UNIVERSIDAD NACIONAL DE MISIONES

LA SOLUCIÓN NUMÉRICA ELEMENTAL

Mario Eugenio Matiauda

Año 2009

EDITORIAL UNIVERSITARIA DE MISIONES

San Luis 1870

Posadas - Misiones – Tel-Fax: (03752) 428601

Correos electrónicos:

edunam-admini@arnet.com.ar

edunam-direccion@arnet.com.ar

edunam-produccion@arnet.com.ar

edunam-ventas@arnet.com.ar

Colección: Cuadernos de Cátedra

Coordinación de la edición: Claudio Oscar Zalazar

ISBN 978-950-579-126-2

Impreso en Argentina

Matiauda, Mario Eugenio
La solución numérica elemental. - 1a ed. - Posadas: EdUNaM - Editorial Universitaria de la Universidad Nacional de Misiones, 2009.
168 p.; 21x30 cm.
ISBN 978-950-579-126-2
1. Matemática. I. Título
CDD 510.711

Fecha de catalogación: 09/03/2009

EL AUTOR

MATIAUDA, Mario Eugenio

Título grado: Ingeniero Químico. FCEQyN-UNaM
Tititulo posgrado:
-Especialista en Celulosa y Papel (FCEQyN)-UNaM.
-Especialista en Vinculación Tecnológica (UNL).
-Magister en Ciencias de la Madera, Celulosa y Papel (FCEQyN)-UNaM.
Como docente se inició en el año 1986 pasando por diversos cargos de la carrera docente. Actualmente es Prof. Adjunto exclusiva (FCEQyN), a cargo de las asignaturas Matemática I-Matemática II, Matemática Aplicada (Analista en Sistemas-FCEQyN-UNaM), y Matemática I, II, III, IV e Investigación Operativa (Licenciatura en sistemas, FCEQyN-UNaM).
Investigación, en la actualidad: integrante del proyecto "Gasificación de aserrín para producción de gas enriquecido en hidrógeno", FCEQyN.
Desde el año 1986 trabaja en investigación siempre relacionado a Ciencia y Tecnología de la madera: Secado de madera, Optimización energética Secadero de madera, Tensiones y Deformaciones en el secado de madera, Combustión.

Publicaciones realizadas (independiente):
Cálculo diferencial e integral, 120 páginas, año 1999, expte 66754 de solicitud de depósito en custodia de obra inédita, Dirección Nacional del derecho de autor, 23/06/00.Único titular.
Acerca de optimización, 106 páginas, -año 1999, expte 66753 de solicitud en depósito en custodia de obra inédita, Dirección Nacional del derecho de autor, 23/06/00. Único titular.
Algebraicas 1.0, 109 páginas, año 2000, expte 82747 de de solicitud de depósito en custodia de obra inédita, Dirección Nacional del derecho de autor, 05/09/00. Único titular.
Programación No Lineal, 1ª. Parte, año 2002, 30 páginas, producción independiente.
Programación No Lineal, 2ª. Parte, año 2002, 30 páginas, producción independiente.
Álgebra lineal (Nociones teóricasy elementales y ejercitación), año 2005, 73 páginas, producción independiente.

ÍNDICE

PRÓLOGO

El objeto de esta publicación es ofrecer una introducción al Análisis Numérico, principalmente a través de los casos más simples de aplicación: solución de ecuaciones univariables, ajuste polinómico y en series de datos, diferenciación e integración numérica, aproximación polinómica y de valores propios para sistemas lineales, solución de ecuaciones diferenciales (hasta segundo orden) ordinarias y en derivadas parciales, con sus asociados problemas de valor inicial y de condición de frontera, ofreciendo la salida efectiva y poderosa de Matlab.

Se estima podrá ser de utilidad para estudiantes de grado de carreras que involucran en su currícula el estudio de las matemáticas (álgebra, análisis) con un carácter simplificado y abriendo la posibilidad de su profundización, tanto en el conocimiento de la disciplina matemática como las surgentes de aplicación del software

UNIDAD 1

1.1. INTRODUCCIÓN: ¿POR QUÉ EL CÁLCULO NUMÉRICO?

Los modelos matemáticos se encargan de representar los fenómenos reales en el campo de la ciencia y tecnología, buscando profundizar el conocimiento del fenómeno y su evolución.

Para encontrar y aplicar las herramientas apropiadas a este tipo de problemas, está la rama de de las matemáticas denominada Matemática aplicada, que muchas veces se ve impedida de aplicar técnicas analíticas y/o exactas debido a ausencia de métodos analíticos solucionantes del problema o son inapropiadas para el modelo planteado o de aplicación muy complicada o conducen a soluciones muy complejas para un post análisis.

Es el momento de inclusión, entonces, de técnicas numéricas, de laboriosidad variable, conducentes a soluciones numéricas lógicamente aproximadas; es justo aclarar que estos métodos son inseparables del empleo del ordenador, cuya evolución actual las hace más seductoras para aplicar estas técnicas.

1.2. CÁLCULO NUMÉRICO Y EL ERROR

Hablar de cálculo numérico es hablar a la vez de error, haciendo imprescindible su evaluación para tener magnitud del grado de aproximación de la solución generada.

El origen de los errores asociados a todo cálculo numérico responden a los propios de la formulación del problema y los surgentes por el método empleado para solucionar el problema.

Entre los primeros están los que resultan de la formulación matemática como una aproximación de la situación física real, que serán despreciables o no según nuestra capacidad formulativa, no ligados de la precisión utilizada por la técnica numérica usada para la solución.

En el segundo grupo se incluyen los derivados en la imprecisión de los datos físicos: constantes físicas y datos empíricos. En el caso de errores en la medida de los datos empíricos y teniendo en cuenta su carácter generalmente aleatorio, su tratamiento analítico es especialmente complejo aunque necesario para enfrentarlos con los obtenidos numéricamente

En lo atinente al error computacional, éste puede abrevar desde: errores en las operaciones (error grueso), el error por el tipo de aproximación, como ser una sumatoria frente a un infinito o un delta por un infinitésimo, caso sencillo: el tomar no términos de un serie expandido para aproximar una función, o las reglas geométricas de aproximación de areas. Englóbase a estos errores como errores de truncado.

Finalmente deben considerarse los derivados, tal vez los más significativos, de la imprecisión del cálculo aritmético, por la operación misma y la representación decimal de los números, en definitiva el redondeo y su error de redondeo.

1.2.1. Breves nociones del error

Para una dada magnitud, siendo $\vec{x}$ su valor y x su valor aproximado, se define:

Error absoluto*:* $e_a(x)=x-\vec{x}$ (1.1)

Error relativo: $e_r(x)=x- \vec{x}/\vec{x}$ (1.2)

Generalmente no se dispone de este valor de error sino sólo se dispone de un $\epsilon(x)$, es decir:

$|e_a(x)| \leq \epsilon_a(x)$ (1.3)

O con el error relativo:

$|e_r(x)| \leq \epsilon_r(x)$ (1.4)

De manera que un número se expresa como:

$$\vec{x} = x \pm \varepsilon_a(x) \quad (1.5)$$

$$\vec{x} = x(1 \pm \varepsilon_r(x)) \quad (1.6)$$

Llamando $x(d)$ a un número redondeado con d decimales y $\varepsilon_r(d)$ su error de redondeo, se deberá verificar:

$$\left|\varepsilon_r(d)\right| = \left|x - x(d)\right| \leq 0.5.10^{-d} \quad (1.7)$$

Tomando 12,37523, redondeado a 4 decimales, se tiene:

x(4)=12,3752 y $\left|\varepsilon_r(4)\right| = \left|12{,}37523 - 12{,}3752\right| = 3.10^{-4} \leq 0.5.10^{-4}$

si se emplea directamente el truncado, descartando los dígitos de menor orden (menor exactitud), con el error asociado dado por:

$$\left|\varepsilon_t(d)\right| = \left|x - x(d)\right| \leq 1.10^{-d} \quad (1.8)$$

¿Cuál es efecto de las operaciones involucradas por las técnicas numéricas sobre el resultado final?
Debe estar perfectamente claro que el error se propaga (se suman las aportes individuales de las variables, ya sean errores absolutos y/o relativos).

En el caso de una función $f(x_1, x_2)$

$$\varepsilon_a(y) = \sum_{i=1}^{2} \left|\frac{\partial}{\partial x_i} f(x_1, x_2)\right| \varepsilon_a(x_i) = \left|\frac{\partial f}{\partial x_1}\varepsilon_a(x_1) + \frac{\partial f}{\partial x_2}\varepsilon_a(x_2)\right| \quad (1.9)$$

Dependiendo de la regla de correspondencia de la función y de las operaciones involucradas, si es menor el número de pasos intermedios que se realicen para lograr la solución, menor será el error cometido.

1.3. ORDENADORES Y SU ARITMÉTICA

Las computadoras no almacenan los números con precisión infinita sino de forma aproximada empleando un número fijo de *bits* (apócope del término inglés *Binary Digit*) o *bytes* (grupos de ocho *bits*). Prácticamente todos los computadores permiten al programador elegir entre varias representaciones o 'tipos de datos'. Los diferentes tipos de datos pueden diferir en el número de bits empleados, pero también (lo que es más importante) en cómo el número representado es almacenado: en formato fijo (también denominado 'entero') o en punto flotante (denominado 'real').

1.3.1 Aritmética de punto fijo

Un entero se puede representar empleando todos los bits de una palabra de computadora, con la salvedad de que se debe reservar un bit para el signo. Por ejemplo, en una máquina con longitud de palabra de 32 bits, los enteros están comprendidos entre $-(2^{31} - 1)$ y $2^{31} - 1 = 2147483647$. Un número representado en formato entero es 'exacto'. Las operaciones aritméticas entre números enteros son también 'exactas' siempre y cuando:

1. La solución no esté fuera del rango del número entero más grande o más pequeño que se puede representar (generalmente con signo). En estos casos se dice que se comete un error de desbordamiento por exceso o por defecto (en inglés: *Overflow* y *Underflow*) y es necesario recurrir a técnicas de escalado para llevar a cabo las operaciones.

2. La división se interpreta que da lugar a un número entero, despreciando cualquier resto. De manera que, excepto en casos elementales, es poco usada la aritmética de punto fijo.

En notación científica normalizada un número real no nulo se representa en la forma:

$$x = \pm r \text{ x } 10^n \qquad (1.10)$$

con r un número comprendido en $\frac{1}{10} < r < 1$, n entero positivo, negativo o cero

Para el sistema binario, la expresión científica sería:

$$x = \pm q \text{ x } 2^m \text{ con } m \text{ entero} \qquad (1.11)$$

q es la mantisa, m es el exponente; en ordenadores binarios q y m se representan como números en base 2, al estar normalizada la mantisa, se tendrá:

$$\frac{1}{2} \le |q| \le 1 \qquad (1.12)$$

¿Cómo representar los números en punto flotante?

Si el ordenador tiene una longitud de palabra de 32 *bits* (los más sencillos), los *bits* se acomodan del siguiente modo:

Signo del número real x:	1 bit
Signo del exponente m:	1 bit
Exponente (entero $\|m\|$):	7 bits
Mantisa (número real $\|q\|$):	23 bits

En la mayoría de los cálculos en punto flotante las mantisas se normalizan, es decir, se toman de forma que el bit más significativo (el primer bit) sea siempre '1'. Por lo tanto, la mantisa q cumple siempre (12).
Dado que la mantisa siempre se representa normalizada, el primer bit en q es siempre 1, por lo que no es necesario almacenarlo proporcionando un bit significativo adicional. Esta forma de almacenar un número en punto flotante se conoce con el nombre de ***técnica del 'bit fantasma'***.
Se dice que un número real expresado como aparece en (11) y que satisface (12) tiene la forma de ***punto flotante normalizado***. Si además puede representarse exactamente con $|m|$ ocupando 7 bits y $|q|$ ocupando 24 bits, entonces es un ***número de máquina*** en el ordenador ejemplificado.

La restricción de que $|m|$ no requiera más de 7 bits significa que:

$$|m| \le (1111111)_2 = 2^7 - 1 = 127$$

como 2^{127} es aproximadamente 10^{38}, el ordenador manejará números tan pequeños como 10^{-38} y tan grandes como 10^{38}, en realidad un intervalo no tan amplio, haciendo menester acudir a programas escritos en ***aritmética de doble precisión*** o de ***precisión extendida***.

Ahora, q se representa hasta con 24 bits entonces los números de máquina tendrán una precisión limitada cercana a las siete cifras decimales, ya que el bit menos significativo de la mantisa representa unidades de 2^{-24}(aprox 10^{-27}), indicando que los números con más de siete dígitos decimales se *aproximarán* cuando se almacenen en el ordenador.

Sea **0.2** representado en punto flotante (para ordenador de longitud de palabra de 32 bits) se almacena en la forma en la memoria del siguiente modo:

$$\frac{1}{5} = 0(\text{sgn } de\ |m|)0(\text{sgn } |q|)0000000(m, 7 \text{ bits})00000000000000000000000(|q|1 + 23bits)$$

Sea el caso de un ordenador cuya notación de punto fijo consiste en palabras de longitud 32 bits repartidas del siguiente modo: 1 bit para el signo, 15 bits para la parte entera y 16 bits para la parte fraccionaria. Represente los números 26.32 en base 2 empleando esta notación de punto fijo y notación de punto flotante con 32 bits. Calcule el error de almacenamiento cometido en cada caso.

El número 26.32 en binario se escribe del siguiente modo:

$$26.32_{10} = 11010.\overline{0101000111101011000}_2$$

Empleando las representaciones comentadas, se obtiene:

26.32_{10} = **Punto fijo :** 0 000000000011010.0101000111101011

Punto flotante : 0 10000101 10100101000111101011100

el error es la diferencia entre el valor y el número realmente almacenado en el ordenador, se obtiene:

$$\varepsilon_a(\text{fix}) = 8 \cdot 10^{-6} \quad \varepsilon_r(\text{fix}) = 3 \cdot 10^{-7}$$
$$\varepsilon_a(\text{flt}) = 1.3 \cdot 10^{-6} \quad \varepsilon_r(\text{flt}) = 1.2 \cdot 10^{-8}$$

Suponiendo que p, el número de bits de la mantisa, sea 24. En el intervalo (1/2,1) (exponente $f = 0$) es posible representar 2^{24} números igualmente espaciados y separados por una distancia $1/2^{24}$. De modo análogo, en cualquier intervalo $(2^f, 2^{f+1})$hay 2^{24} números equiespaciados, pero su densidad en este caso es $2^f/2^{24}$. Por ejemplo, entre $2^{20} = 1048576$ y $2^{21} = 2097152$ hay $2^{24} = 16777216$ números, pero el espaciado entre dos números contiguos es 1/16

Se puede extraer lo siguiente: cuando es necesario comparar dos números en punto flotante relativamente grandes, es siempre preferible comparar la diferencia relativa a la magnitud de los números.

1.4. ARITMÉTICA DE PUNTO FLOTANTE

1.4.1. Números de máquina aproximados

Se estudia cómo obtener el error cometido al aproximar un número real positivo x mediante un número de máquina ordenador 32.

Si se representa el número por:

$$x=(a_1a_2\ldots a_{24}a_{25}a_{26}\ldots)_2 \text{x } 2^m$$

en donde cada a_i es 0 o 1 y el bit principal es $a_1 = 1$. Un número de máquina se puede obtener de dos formas:

- **Truncamiento**: descartando todos los bits excedentes $a_{25}a_{26}$., el número resultante, x' es siempre menor que x (se encuentra a la izquierda de x en la recta real).
- **Redondeo por exceso**: Aumentando en una unidad el último bit remanente a_{24} y después se elimina el exceso de bits como en el caso anterior.

Todo lo anterior, aplicado al caso del ordenador 32, se sintetiza como: si x es un número real distinto de 0 dentro del intervalo de la máquina, entonces el número de máquina x^* más cercano a x satisface la desigualdad:

$$\delta = \left|\frac{x - x^*}{x}\right| \le 2^{-24} \qquad (1.13)$$

que se puede escribir así:

$$x^* = x(1+\delta) \qquad |\delta| \le 2^{-24}$$

Tomando el número 2/3:

El número 2/3 en binario se expresa como:

$$\left(\frac{2}{3}\right)_{10} = \left(0.1\overline{0}\right)_2$$

Los números (24 bits) cercanos son
$x'=(0,101010...1010)_2$
$x''=(0,101010....1011)_2$
x' de truncado y *x''* redondeado en exceso

Entonces:

$x-x'=2/3 \times 10^{-24}$
$x-x'=1/3 \times 10^{-24}$

el más próximo es x'', con los errores de redondeo absoluto y relativo del orden de*:*

$|f(x)-x|=1/3x10^{-24}$
$|f(x)-x/x|=2^{-25}<2^{-24}$

1.4.2. Operaciones

Se quiere encontrar el resultado de operar sobre dos números en punto flotante normalizado de *l*-dígitos de longitud, *x* e *y*, que producen un resultado normalizado de *l*-dígitos, es decir:

$$fl(x\ op\ y)$$

donde *op* es +, -, x o /.

Se asume, en cada caso, la mantisa del resultado es primero normalizada y después redondeada (operación que puede dar lugar a un desbordamiento que requeriría renormalizar el número). El valor de la mantisa redondeada a *p* bits, q_r, se define como (de una forma más rigurosa que en el caso anterior):

$$q_r = \begin{cases} 2^{-p}\left\lfloor 2^p q - \frac{1}{2}\right\rfloor & \\ 2^{-p}\left\lfloor 2^p q + \frac{1}{2}\right\rfloor & q<0 \\ & q>0 \end{cases}$$

en donde la función *redondeo por defecto* $\lfloor x \rfloor$ es el mayor entero menor o igual a *x* y la función *redondeo por exceso* $\lceil x \rceil$ es el menor entero mayor o igual a *x*. Para números enteros, esta función se traduce en la bien conocida regla de sumar 1 en la posición $p + 1$. Teniendo en cuenta solo la mantisa, redondear de este modo da lugar a un intervalo máximo del error de:

$$|\epsilon_a| \leq 2^{-p-1} \qquad (1.14)$$

Y un error relativo máximo en el intervalo

$$|\varepsilon_r| \leq \frac{2^{-p-1}}{1/2} = 2^{-p} \qquad (1.15)$$

Pasando al error que se genera en cada una de las operaciones elementales:

* multiplicar dos números expresados en punto flotante significa sumar los exponentes y multiplicar las mantisas, si la mantisa resultante no está normalizada, se recurre a renormalizar el resultado ajustando adecuadamente el exponente. Luego, es necesario redondear la mantisa a *p* bits.
Suponiendo dos números

$$x = q_x 2^{f_x} \quad \text{y} = q_y 2^{f_y}$$

Al multiplicar:

$$xy = q_x\, q_y\, 2^{f_x + f_y}$$

en donde el valor de la mantisa se encontrará en el intervalo:

$$\frac{1}{4} \leq |q_x q_y| < 1$$

Es decir, la normalización del producto $q_x\ q_y$ implica un desplazamiento a la derecha de, como máximo, una posición. La mantisa redondeada será entonces uno de estos dos posibles valores:

$$x = q_x\, q_y + \epsilon \quad \text{o } x = 2\, q_x\, q_y + \epsilon_{\ y}$$

donde ϵ, el error de redondeo, cumple la ecuación respectiva, teniéndose entonces:

$$fl(x \text{ x y}) = \begin{cases} (q_x q_y + \varepsilon) 2^{f_x + f_y} & |q_x q_y| \geq \frac{1}{2} \\ (2q_x q_y + \varepsilon) 2^{f_x + f_y} & \frac{1}{2} > |q_x q_y| \geq \frac{1}{4} \end{cases}$$

$$= q_x q_y 2^{f_x + f_y} \begin{cases} 1 + \dfrac{\varepsilon}{q_x q_y} & |q_x q_y| \geq \frac{1}{2} \\ 1 + \dfrac{\varepsilon}{2 q_x q_y} & \frac{1}{2} > |q_x q_y| \geq \frac{1}{4} \end{cases}$$

$$= xy(1+\epsilon_q)$$

Es decir que: $|\epsilon_q| \leq 2|\epsilon| \leq 2^{-p}$

la cota del error relativo en la multiplicación es la misma que la que surge por redondear la mantisa.

- Para dividir en punto flotante, se divide la mitad de la mantisa del numerador por la mantisa del denominador (para evitar cocientes mayores de la unidad), mientras que los exponentes se restan:

$$\frac{x}{y} = \frac{q_x / 2}{q_y} 2^{f_x - f_y + 1}$$ estando el cociente limitado por:

$$\frac{1}{4} \le \left| \frac{q_x}{2q_y} \right| < 1$$

Mediante un procedimiento similar al de la multiplicación, se obtiene:

$$fl(x / y) = \begin{cases} \left(q_x / 2q_y + \varepsilon\right) 2^{f_x - f_y + 1} & \left|q_x / 2q_y\right| \ge \frac{1}{2} \\ \left(2q_x q_y + \varepsilon\right) 2^{f_x - f_y} & \frac{1}{2} > \left|q_x / 2q_y\right| \ge \frac{1}{4} \end{cases}$$

$$= q_x / 2q_y \, 2^{f_x - f_y + 1} \begin{cases} 1 + \dfrac{\varepsilon}{q_x / 2q_y} & \left|q_x / 2q_y\right| \ge \frac{1}{2} \\ 1 + \dfrac{\varepsilon}{q_x q_y} & \frac{1}{2} > \left|q_x / 2q_y\right| \ge \frac{1}{4} \end{cases}$$

$$= x/y.(1+\epsilon_d)$$

Entonces

$$|\epsilon_d| \le 2 |\epsilon| \le 2^{-p}$$

Equivale a: la cota máxima del error relativo en la división, como en el caso anterior, es la misma que la que surge por redondear la mantisa.

*** Suma y resta.**

La operación de suma o resta se realiza: se toma la mantisa del operando de menor magnitud (supongamos que es y) y se desplaza $f_x - f_y$ posiciones a la derecha. La mantisa resultante es sumada (o restada) y el resultado se normaliza y después se redondea, o sea:

$$x \pm y = (q_x \pm q_y \, 2^{f_y - f_x}) 2^{f_x}$$

El análisis del error cometido en esta operación es más complejo que los estudiados hasta ahora, obviándose su deducción pero la expresión final indica que la cota máxima del error cometido en la adición y la sustracción será:

$$|\epsilon_a| \le 2 |\epsilon| \le 2^{-p}$$

Resumiendo, en todas las operaciones aritméticas elementales en punto flotante, el error absoluto del resultado es no mayor de 1 en el bit menos significativo de la mantisa.
Respecto a los errores de redondeo: estos se acumulan a medida que aumenta el número de cálculos. Si en el proceso de calcular un valor se llevan a cabo N operaciones aritméticas es posible obtener, en el mejor de los casos, un error de redondeo total del orden de $\sqrt{N}\ \varepsilon_m$

(que coincide con el caso en que los errores de redondeo están aleatoriamente distribuidos, por lo que se produce una cancelación parcial).

Este error puede crecer muy rápidamente por dos motivos:

- Es muy frecuente que la regularidad del cálculo o las peculiaridades del ordenador originen que el error se acumule preferentemente en una dirección; en cuyo caso el error de redondeo se puede aproximar a $N\epsilon_m$.
- En circunstancias especialmente desfavorables pueden existir operaciones que incremente espectacularmente el error de redondeo, por ejemplo cuando se calcula la diferencia entre dos números muy próximos, dando lugar a un resultado en el cual los únicos bits significativos que no se cancelan son los de menor orden (en los únicos en que difieren). Puede parecer que la probabilidad de que se de dicha situación es pequeña, sin embargo, algunas expresiones matemáticas fomentan este fenómeno.

Se toma una situación para la expresión cuadrática $ax^2 + bx + c = 0$, resuelta como

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (16) \quad \text{o} \quad \frac{2c}{-b \pm \sqrt{b^2 - 4ac}} \qquad (1.17)$$

Para *a,c*, o los dos pequeños, el valor del discriminante es cercano a *b*, de manera que la diferencia $|b| - \sqrt{b^2 - 4ac}$ presenta un importante error de redondeo; en (1.16) se evalúa bien la raíz más grande en valor absoluto, en (1.17) la menor.
De allí la necesidad de mejorar la condición, siendo preferible calcular primero

$$q = \frac{1}{2}\left[b + sgb(b)\sqrt{b^2 - 4ac} \right] \qquad (1.18)$$

luego las dos raíces a partir de *b:* $x_1 = q/a \quad x_2 = c/q$ (1.19)

tomando: $1.0\, x^2 + 1{,}343.10^5 x + 3{,}764.10^{-6} = 0$

$$a = 1.0; \quad b = 1.343 \cdot 10^5; \quad 3.764 \cdot 10^{-6}$$

Con (1.16) se obtienen $x_1 = -0{,}13430.10^5 \quad x_2 = 0{,}44676.10^{-3}$ pero con (1.17) $x_1 = \infty$ *(overflow)* $x_2 = -0{,}28027.10^{-10}$; adoptando (1.19) se obtienen $x_1 = -0{,}13430.10^5$, $x_2 = -0{,}28027.10^{-10}$

1.5. DESBORDAMIENTO POR EXCESO Y DESBORDAMIENTO POR DEFECTO

El desbordamiento por exceso de punto flotante (overflow) se da cuando el resultado de una operación de punto flotante tiene una magnitud superior a $M\text{x}2^{F:}$ ($F = 2^7 - 1$; $M = 1 - 2^{24}$).

Si $q=8$, $F = 2^7 - 1 = 127$, se generaría desbordamiento cuando por ejemplo:

$$\left(\tfrac{1}{2} \times 2^{70}\right) \times \left(\tfrac{1}{2} \times 2^{80}\right); \quad \left(\tfrac{1}{2} \times 2^{127}\right) - \left(-\tfrac{1}{2} \times 2^{127}\right)$$

Análogamente, el desbordamiento por defecto (*underflow*) se produce cuando el resultado de una operación en punto flotante es demasiado pequeño, aunque no nulo, el número más pequeño representable suponiendo que siempre trabajamos con mantisas normalizadas es $\frac{1}{2}x2^{-F}$, donde $-F$ es el exponente negativo más grande permitido (generalmente -2^{-q-1}). Por ejemplo, con q=8 resulta $-F$ = -128.

Entonces $\frac{1}{2}x2^{-80} / \frac{1}{2}x2^{50}$ generaría underflow; en el caso de overflow es producto de un error en el cálculo, en underflow posible continuar el cálculo reemplazando el resultado por cero.

1.6. CONDICIONAMIENTO Y ESTABILIDAD

La 'inestabilidad' en un cálculo es un fenómeno que se produce cuando los errores de redondeo individuales se propagan a través del cálculo incrementalmente. La mejor forma de ver este fenómeno es a través de un ejemplo.
Suponiendo el siguiente sistema de ecuaciones diferenciales:
$y'_1=y_2$
$y'_2=y_1$
cuya solución general:
$y_1=a_1e^x+a_2e^{-x}$
$y_2= a_1e^x-a_2e^{-x}$
Para un PVI tal que $y_1(0) = -y_2(0) = 1$ el valor de las constantes a_1, a_2 es: $a_1 = 0$ y $a_2= 1$

Ahora, suponiendo que el sistema de ecuaciones anterior se resuelve empleando un método numérico cualquiera con el fin de calcular los valores de las funciones y_1 y y_2 en una secuencia de puntos $x_1,x_2,...,x_n$ y que el error del método da lugar a un valor de $a_1\neq0$.
Como a_1 multiplica a un exponencial creciente cualquier valor, por pequeño que sea, de a_1 dará lugar a que el término e^x prepondere sobre e^{-x} para valores suficientemente grandes de x
Entonces: no es posible calcular una solución al sistema de ecuaciones diferenciales anterior que, para valores suficientemente grandes de x, no de lugar a un error arbitrariamente grande en relación con la solución exacta (caso de problema inestable o mal condicionado).
El problema anterior se dice que es inherentemente inestable, o empleando una terminología más común en cálculo numérico, se dice que está 'mal condicionado' (*ill-conditioned*).

EJERCICIOS PROPUESTOS PARA UNIDAD 1

1. ¿Con qué exactitud es necesario medir el radio de una esfera para que su volumen sea conocido con un error relativo menor de 0.01%? ¿Cuántos decimales es necesario emplear para el valor de π?

2. Suponiendo una barra de hierro de longitud *l* y sección rectangular a x b fija por uno de sus extremos. Si sobre el extremo libre aplicamos una fuerza *F* perpendicular a la barra, la flexión *s* que rsta experimenta viene dada por la expresión:

$$s = \frac{4}{E}\frac{l^3}{ab^3}F$$

en donde *E* es una constante que depende solo del material denominada módulo de Young. Conociendo que una fuerza de 140 Kp aplicada sobre una barra de 125 cm de longitud y sección cuadrada de 2.5 cm produce una flexión de 1.71 mm, calcular el módulo de Young y el intervalo de error. Suponer que los datos vienen afectados por un error máximo correspondiente al de aproximar por truncamiento las cifras dadas

UNIDAD II: RESOLVER f(x) =0

SOLUCIONES DE ECUACIONES UNIVARIABLES

Para una función $f(x)=0$, el problema más simple de la aproximación numérica será hallar la raíz x que anula f.
Se verán algunas técnicas sencillas para la dilucidación de este problema.

2.1. MÉTODO DE LA BISECCIÓN

Se parte de f definida en I real con extremos a y b pertenecientes a I, bajo el supuesto de que $f(a).f(b)\langle 0$.
De acuerdo al teorema de la media, habrá una p, en (a,b) que garantice $f(p)=0$.
El método radica en dividir a la mitad sucesivamente, los $I_i \subset I$, encontrando en cada paso la mitad que contiene a p.
Haciendo p_i el primer punto medio, se tendrá

$$p_i = \frac{a+b}{2} \qquad (2.1)$$

En el caso de que $f(p_1)=0$, $p=p_i$; de lo contrario $f(p_1)$ coincidirá en signo con $f(a)$ o $f(b)$.
Para la situación que coincidan en signo $f(p_1)$ y $f(a)$ equivaldría a sostener que $p \in (p,b)$, tomando entonces $a_1 = p_1$ y $b_1 = b$.
Si $f(p_1)$ y $f(a)$ difieren en signo, se tendrá $p \in (a,p_1)$, haciendo $a_2 = a$ y $b_1 = q_1$.
Este proceso se repite en el intervalo $[a,b]$.

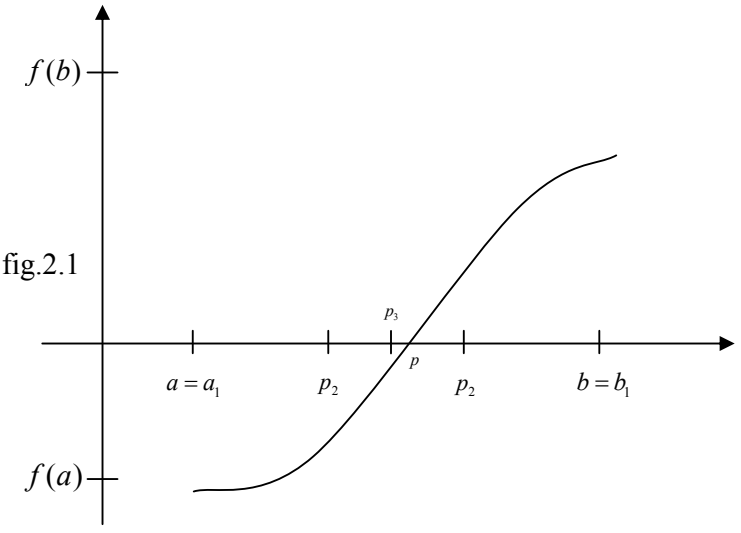


fig.2.1

Lógicamente para un algoritmo iterativo, se deberá pensar en un criterio de detención.
Por ejemplo para determinar el número de iteraciones para resolver $f(x)=0$ con una tolerancia ε elegida $(\rangle 0)$, se puede emplear

$$|p_n - p| \leq \frac{b-a}{2^n} \langle \varepsilon \qquad (2.2)$$

El método presenta lentitud en su convergencia, pudiendo darse que en el proceso de repetición se pierda una aproximación buena sin percatarse de ello.
Pero posee la ventaja de converger siempre a una solución, quedando su uso como arranque para métodos más eficaces.
Ejemplo 2.1
Dada la ecuacion $xe^x - 1 = 0$, se pide:
a) Estudiar gráficamente sus raíces reales y acotarlas.
b) Aplicar el método de la bisección y acotar el error después de siete iteraciones.
Para $x < 0: 1/x < 0$ y $e^x > 0 \rightarrow e^x \neq 1/x$
Para $x > 0$: $f(x) = xe^x - 1 \rightarrow f(0) = -1 < 0$ y $f(+\infty) = +\infty > 0$
y existe, por tanto, un número impar de raíces positivas (al menos una).
La función derivada $f'(x) = xe^x + e^x = (x+1)e^x$ solo se anula para $x = -1$.
Dado que, si existiera más de una raíz positiva, el teorema de Rolle asegura que la función derivada debe anularse en algún punto intermedio y se ha visto que $f'(x)$ no se anula para ningún valor positivo de la variable, se puede asegurar que solo existe una raíz real, positiva y simple, pues su derivada es $\neq 0$.
Dado que $f(1) = e - 1 > 0$, $f(0) = -1 < 0$, se concluye que la raíz estará en $(0, 1)$.
$[a_0, b_0] = [a, b] = [0, 1]$ con
$f(0) = -1 < 0$
$f(1) = e - 1 > 0$

$f(0.5) < 0 \rightarrow [a_1, b_1] = [0.5, 1]$
$f(0.75) > 0 \rightarrow [a_2, b_2] = [0.5, 0.75]$
$f(0.625) > 0 \rightarrow [a_3, b_3] = [0.5, 0.625]$
$f(0.5625) < 0 \rightarrow [a_4, b_4] = [0.5625, 0.625]$
$f(0.59375) > 0 \rightarrow [a_5, b_5] = [0.5625, 0.59375]$
$f(0.578125) > 0 \rightarrow [a_6, b_6] = [0.5625, 0.578125]$
$f(0.5703125) > 0 \rightarrow [a_7, b_7] = [0.5625, 0.5703125]$
Tomando como aproximación a la raíz el punto medio del intervalo
$x_7 = 0.56640625 \rightarrow |\epsilon_7| < 1/2^{7+1} = 0.00390625 \rightarrow |\epsilon_7| < 10^{-2}$
Redondeando a las dos primeras cifras decimales, es decir, tomando
$\tilde{x} = 0.57$, el error acumulado verifica que
$|\epsilon| < |0.57 - 0.56640625| + 0.00390625 = 0.0075 < 10^{-2}$
por lo que puede asegurarse que la solución de la ecuación es 0.57 con las dos cifras decimales exactas.

2.2. MÉTODO DEL PUNTO FIJO

Para una función h determinada, el número q que hace $h(q) = q$ es el conocido como punto fijo de h.
Así el problema de hallar raíces de $h(q) = 0$ es equivalente al de encontrar un punto fijo, definiendo la función h con un punto fijo q en diversas maneras, ya sea $h(x) = x - f(x)$ o $h(x) = x + \alpha f(x)$, $\alpha \in \mathbf{R}$.
En idéntica forma, si h posee a q como punto fijo, $f(x) = x - h(x)$ tendrá una raíz en q.

¿Cómo se establece la existencia y unicidad de un punto fijo?
Si $h \in C[a,b]$ y $h(x) \in [a,b]$ para toda $x \in [a,b]$, h tiene un punto fijo en $[a,b]$.
Si también $g'(x)$ existe en $]a,b[$ con una constante $m\langle 1$, de modo que

$$|h'(x)| \le m\langle 1 \qquad \forall x \in]a,b[\qquad (2.3)$$

que garantiza que el punto fijo es único en $[a,b]$.
La sucesión que se genera $\{q_n\}_{n=0}^{\infty}$ con $q_n = h(q_{n-1})$ para $n \ge 1$ conlleva cotas de error en la aproximación de q_n a q, a través de:

$$|q_n - q| \le m^n \quad \max\{q_0 - a, b - q_0\} \qquad (2.4)$$

Y

$$|q_n - q| \le \frac{m^n}{1-m}|q_0 - q_1| \qquad \forall n \ge 1 \qquad (2.5)$$

La velocidad de convergencia de la sucesión $\{q_n\}$ hacia la cota m depende del factor $\frac{m^n}{1-m}$, haciéndose más rápida para m más pequeña, volviéndose muy lenta para m cercana a 1.

La figura (2.2) muestra como generar una sucesión por el método del punto fijo, en este caso $0<f'(x)<1$

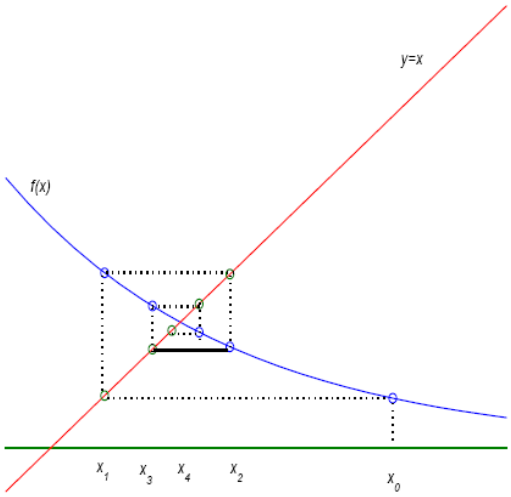


Fig. 2.2

Ejemplo 2.2.- Dada la ecuación $x^3 + 4x^2 - 10 = 0$, estudiarla en el intervalo $[1,2]$ empleando el método del punto fijo.
Se obtendrán diferentes expresiones, para apreciar que el punto fijo de cada una es solución de la ecuación original. Así:

$$a) x = h_1(x) = x - x^3 - 4x^2 + 10$$

$$b) x = h_2(x) = \left(\frac{10}{x} - 4x\right)^{0,5}$$

$$c) x = h_3(x) = 0,5\left(10 - x^3\right)^{0,5}$$

$$d) x = h_4(x) = \left(\frac{10}{4+x}\right)^{0,5}$$

Si se adopta un $q = 1,5$ se podrían obtener los resultados para a), b), c) y d), según la tabla dada:

n	a	b	c	d
0	1,5	1,5	1,5	1,5
1	–0,875	0,8165	1,28695376	1,34839972
2	6,732	2,9969	1,40254080	1,36737637
3	–469,7	$\left(\langle 0\right)^{0,5}$	1,34545837	1,36495701
4	$1,03\times10^8$		1,37517025	1,36526474
5			1,36009419	1,36522559
6			1,36784696	1,36523057
7			1,36388700	1,36522994
8			1,36591673	1,36523002
9			1,36591673	1,36523001

Donde se ve claramente la divergencia en a) y que queda indeterminado en el b) en los reales.
Ahora estudiando en cuanto a las exigencias del teorema del punto fijo para las distintas expresiones de $h(x)$:

a).- la derivada $h'(x) = 1 - 3x^2 - 8x$, entonces resulta que no es posible encontrar un intervalo que contenga a q verificándose $\left|h_1'(x)\right| \langle 1$, no siendo esperable la convergencia en tal caso.

b).- Para $h_2(x) = \left[\frac{10}{x} - 4x\right]^{0,5}$ esta expresión el $[1,2]$ en $[1,2]$ y la sucesión de los q_i no está definida en 1,5 como tampoco $\left|h_2'(x)\right| \langle 1$.

c).- Para $h_3(x)$, su derivada es negativa en $[1,2]$, lo que habla de un comportamiento decreciente, pero en $x = 2$ su valor absoluto es 2,1 aproximadamente, no cumpliendo con $\left|h_3'(x)\right| \langle 1$. Ahora si se restringe el intervalo a $[1,q]$ se verificarán todas las condiciones para su convergencia.

d).- En el caso de $h_4(x)$ la cota del valor absoluto de la derivada $h_4'(x)$ es que la cota de $h_3'(x)$, por lo tanto convergerá más rápido.

2.3. MÉTODO DE NEWTON

Entre las posibilidades para $f(x) = 0$, se presenta el método de Newton.
Considerando $g \in C^2[a,b]$, con $x^* \in [a,b]$ aproximación de p de modo que $g'(x^*) \neq 0$ y $\left|x^* - p\right|$ pequeño.

Desarrollando el polinomio de Taylor de primer grado en derredor de x^*.

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + \frac{(x - x^*)}{2} g''(\xi(x)) \qquad (2.6)$$

Con $x\langle\xi(x)\langle x^*$
Al ser $g(p) = 0$, para $x = p$, se tendrá

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + \frac{(x - x^*)}{2} g''(\xi(x)) \qquad (2.7)$$

Despreciando $\left(p - x^*\right)^2$, se verá que al despejar p:

$$p \approx x^* - \frac{g(x^*)}{g'(x^*)} \qquad (2.8)$$

Entonces, partiendo de un p_0 de arranque, se genera la sucesión $\{p_n\}$ definido a través de:

$$p = p_{n-1} - \frac{g(p_{n-1})}{g'(p_{n-1})} \qquad n \geq 1 \qquad (2.9)$$

Gráficamente se emplean tangentes a la función; a partir de un p_0 se obtendrá la aproximación de p_1 (intersección de la tangente de $g(p_0, g(p_0))$ con el eje x), seguidamente p_2 y así continua
Las pautas de detención del método son similares al de bisección, como ser

$$\left|p_n - p_{N-1}\right| \langle \varepsilon \qquad \varepsilon \rangle 0 \qquad (2.10)$$

$$\frac{\left|p_N - p_{N-1}\right|}{\left|p_N\right|} \langle \varepsilon \qquad p_N \rangle 0 \qquad (2.11)$$

O

$$\left|g(p_N)\right| \langle \varepsilon \qquad (2.12)$$

En definitiva, el método genera funciones $p_N = g(p_{N-1})$ con

$$g(p_N) = p_{N-1} - \frac{g(p_{N-1})}{g'(p_{N-1})} \qquad n \geq 1 \qquad (2.13)$$

Presentando mayor eficacia para la derivada de g alejada de la nulidad y cerca del punto fijo.
Respecto a la convergencia es importante la elección de p_0.

La dificultad del método estriba en que se debe conocer g' en cada aproximación, situación que según sea g puede volver engorrosa la técnica. En síntesis, la idea del método es recorrer las tangentes (fig 2.3).

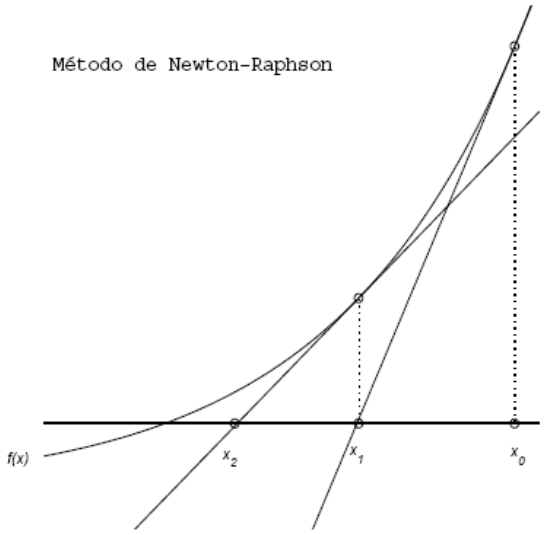


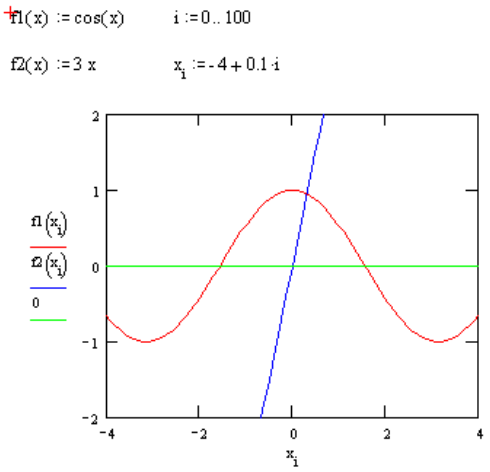
fig. 2.3

Ejemplo 2.3 Se considera la ecuación real $2\cos(2x) + 4x - k = 0$.
Para $k = 3$, probar que posee una única raíz simple en el intervalo [0, 1], y calcularla con 6 cifras decimales exactas utilizando el método de Newton.
Para $k = 3$ se tiene $f(x) = 2\cos(2x) + 4x - 3$ pero sus derivadas son independientes del valor asignado a k.
Como $f(0) = -1$ y $f(1) = 1 + 2\cos 2 = 0.1677... > 0$ la función tiene, al menos una raíz en dicho intervalo
Dado que en [0, 1] se anula la derivada $(f'(\pi/4) = 0)$ interesa reducir el intervalo en el que va a buscar la raíz. Para ello, y dado que $f(0.5) = 2\cos 1 - 1 = 0.0806... > 0$, se restringe al intervalo $[0, 0.5]$ en el que se sabe no se anula la derivada.
Como $f'(x) = -4\sin(2x)+4 > 0 \;\; x \in [0, 0.5]$ y $f''(x) = -8\cos(2x) < 0 \; x \in [0, 0.5]$, la re-gla de Fourier dice que el método de Newton converge con valor inicial $x_0 = 0$.
La fórmula de Newton-Raphson queda de la forma:
$x_{n+1} = x_n - f(x_n)/f'(x_n) = -4x\sin(2x) - 2\cos(2x) + 3/-4\sin(2x) + 4$
entonces

$x_0 = 0$	$\epsilon_0 < 2$
$x1 = 0.25$	$\epsilon_1 < 0.48966975243851$
$x_2 = 0.36757918145023$	$\epsilon_2 < 0.09246835650344$
$x_3 = 0.40268002241238$	$\epsilon_3 < 0.00715318566049$
$x_4 = 0.40588577560341$	$\epsilon_4 < 5.683578640000000 \cdot 10^{-5}$
$x_5 = 0.40591165781801$	$\epsilon_5 < 30688150000000000 \cdot 10^{-9}$

Por tanto x = 0.405912 con un error $\epsilon < 3.4218199 \cdot 10^{-7} + 3.6881500 \cdot 10^{-9} < 10^{-6}$,es decir, con sus seis cifras decimales exactas.

Ejemplo 2.4 Encontrar una raíz real de la expresión: $\cos x - 3x = 0$
Si se genera una gráfica, haciendo variar x



El valor próximo a $\pi/8$ sería raíz, planteando dos funciones equivalentes en términos de *x,* a) *x=cosx-2x* b)*x=cosx/3*

Para a) *f(x)=cosx -3x*
g(x)=cosx 2x
$x_0 = \pi/8$, *k=0…4,* x_0*=0.39270 y* x_{k+1}=g_1(x)

x_k	$g_1(x_k)$	$f(x_k)$
0.39270	0.13848	-0.25422
0.13848	0.71346	0.57498
0.71346	-0.67083	-1.38429
-0.67083	2.12496	2.79579
2.12496	-4.77616	-6.90113

En el caso b)

x_k	$g_2(x_k)$	$f(x_k)$
0.39270	0.30796	0.25422
0.30796	0.31765	0.02907
0.31765	0.31666	0.00298
0.31666	0.31676	0.00031
0.31676	0.31675	0.00003

Mostrando que la alternativa b) presenta convergencia para el método

2.4. MÉTODO DE LA SECANTE

Es una variación del método de Newton.

$$x_n = x_{n-1} - \frac{g(x_{n-1})(x_{n-1} - x_{n-2})}{g(x_{n-1}) - g(x_{n-2})} \qquad (2.14)$$

O sea partiendo de de las aproximaciones de arranque x_0 y x_1, x_2 está dada por la intersección con el eje *x* de la recta que pasa por $(x_0, g(x_0))$ y $(x_1, g(x_1))$; luego x_3 representa la intersección con el eje *x* de la recta pasante por $(x_1, g(x_1))$ y $(x_2, g(x_2))$, continuando en la misma forma.

Este método, de convergencia más lenta que Newton, se suele emplear para ajustar un valor hallado por otra técnica, como la bisección. En la fig. 2.4 se muestra la representación geométrica de las iteraciones al aplicar el método de la secante

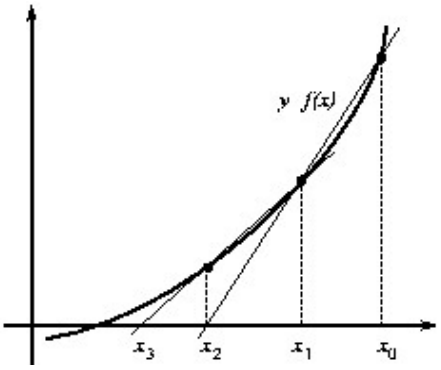
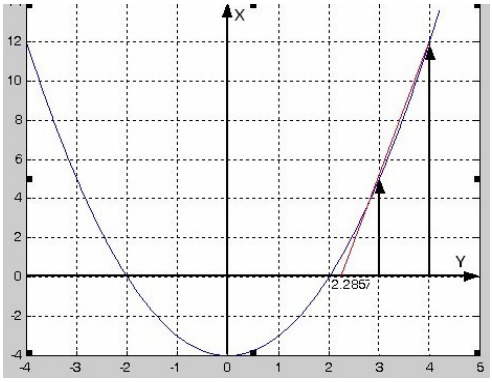


fig. 2.4

La acotación de la raíz no está asegurada a través del método de Newton o de la Secante.

Ejemplo 2.5 Usar el método de la secante para calcular la raíz aproximada de la función $f(x) = x^2 - 4$. Comenzando con $x_0 = 4$, $x_1 = 3$ y hasta que $\epsilon r \leq 1\%$.



Primera iteración aplicando el método de la secante de la función $f(x) = x^2 - 4$

Resultados al aplicar el método de la Secante a la función $f(x) = x^2 - 4$. Con $x_0 = 4$ y $x_1 = 3$

i	x_i	x_{i+1}	x_{i+2}	ϵ_a	ϵ_r
0	*4*	*3*	*2.2857*	*0.7143*	*31.25%*
1	*3*	*2.2857*	*2.0541*	*0.2316*	*11.28%*
2	*2.2057*	*2.0541*	*2.0036*	*0.0505*	*2.52%*
3	*2.0541*	*2.0036*	*2.0000*	*0.0036*	*0.18%*

Se termina el proceso iterativo con la encontrada de la raíz para $x_5 = 2.000$

2.5. MÉTODO DE LA POSICIÓN FALSA

Se diferencia del método anterior en que ofrece un modo de control para la raíz acotada entre dos repeticiones consecutivas.

Suponiendo una función $f:[a,b] \rightarrow R$ continua, verificando $f(a)f(b)<0$, pensada como única raíz en el intervalo.

Sea x_1 la intersección de la recta secante L con el eje x, la cual une los puntos $(a,f(a))$ y $(b,f(b))$, su ecuación es:

$$y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a) \qquad (2.15)$$

Como x_1 es el valor de x para $y=0$, se tiene

$$x_1 = a - \frac{f(a)}{f(b)-f(a)}(x-a) = \frac{af(b)-bf(a)}{f(b)-f(a)} \tag{2.16}$$

Si $f(x_1)\neq 0$, entonces $f(a)f(x_1)<0$ o $f(b)f(x_1)<0$

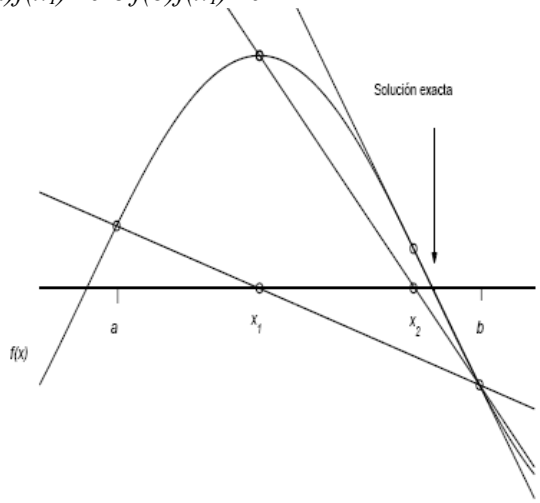


fig. 2.5

En el caso que $f(b)f(x_1)<0$, se define x_2 con el mismo criterio anterior para el intervalo $[x_1,b]=I_1$ y así en adelante. El valor $|I_n|$ puede no tender a 0 pero $x_n \to r$ para toda función continua.

Ejemplo 2.6 Usar el método de la regla falsa para aproximar la raíz de $f(x) = e^{-x} - \ln x$, comenzando en el intervalo $[1,2]$ y hasta que $|\in_a| < 1\%$

$$x_r = x_0 - \frac{f(x_0)[x_1 - x_0]}{f(x_1) - f(x_0)} = 2 - \frac{f(2)\cdot[1-2]}{f(1)-f(2)} = 1.397410482$$

Siguiendo el proceso

$$f(x_r) = e^{-1.397410482} - \ln(1.397410482 = -0.087384509 < 0$$

Estudiando los signos

$f(1)$	$f(1.397410482)$	$f(2)$
+	−	−

La raíz está en [*1, 1.397410482*]

Se calcula la nueva aproximación

$$x_{r_1} = x_0 - \frac{f(x_0)[x_1 - x_0]}{f(x_1) - f(x_0)} = 1.397410482 - \frac{f(1.397410482)\cdot[1-1.397410482]}{f(1)-f(1.397410482)}$$

Con un error

$$|\in_a| = \left|\frac{1.321130513 - 1.397410482}{1.321130513} \times 100\%\right| = 5.77\%$$

Evaluando $f(x_{r_2}) = f(1.321130513) = -0.011654346 < 0$

Se genera la tabla

$f(1)$	$f(1.397410482)$	$f(1.321130513)$
+	−	−

La raíz esta en [1, 1.321130513]

$$x_3 = x_2 - \frac{f(x_2)[x_2 - x_1]}{f(x_2) - f(x_1)} = 1.321130513 - \frac{f(1.321130513)\cdot[1 - 1.321130513]}{f(1) - f(1.321130513)}$$

Y error

$$|\varepsilon_a| = \left|\frac{1.311269556 - 1.321130513}{1.311269556} \times 100\%\right| = 0.75\%$$

Entrando en lo solicitado, de modo que la raíz es **1,311269556**

2.6. ANÁLISIS DEL ERROR

Sea $\{q_n\}_{n=0}^{\infty}$ una sucesión convergente a q. Si existen λ y δ $(\rangle 0)$ de modo que

$$\lim_{n\to\infty} \frac{|q_{n+1} - q|}{|q_n - q|^{\alpha}} = \lambda \qquad (2.17)$$

Se dirá que la sucesión converge a q con orden α y una constante de error asintótico λ.
Globalmente, mayor convergencia de una sucesión implica mayor velocidad de convergencia.
Los más usuales valores de α: 1 y 2, definen un método de convergencia lineal y un método de convergencia cuadrática, respectivamente.
En el caso de los métodos del punto fijo, se puede dar convergencia de orden superior solo cuando $g'(q) = 0$.
Pero si q es una solución de $x = g(x)$, con $g'(q) = 0$, g'' continua y acotada estrictamente por un valor M en un intervalo abierto que contenga a q existirá un $\delta\rangle 0$ tal que para $q_0 \varepsilon [q - \delta, q + \delta]$ la sucesión $g_n = g(q_{n-1})$ para $n \geq 1$ convergerá cuadráticamente a q.a

¿Cómo incrementar la rapidez de convergencia?
Para una secesión que converge linealmente, se menciona una técnica de aceleración de convergencia:

Técnica de Aitken:
Partiendo de que $\{q_n\}_{n=0}^{\infty}$, lineal en convergencia con límite q y una constante de error asintótico inferior a 1.
La técnica considera la solución:

$$\overset{\Delta}{q} = q_n - \frac{(q_{n+1} - q_n)^2}{q_{n+2} - 2q_{n+1} + q} \qquad (2.18)$$

Que converja a mayor velocidad a q que la inicial $\{q_n\}_{n=0}^{\infty}$
Otra técnica conocida para introducir convergencia cuadrática es la de Steffensen.

2.7. POLINOMIOS. RAÍCES. MÉTODOS

Una función expresada por:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \qquad (2.19)$$

Con las a_i constantes, coeficientes $P_j a_n \neq 0$ representa un polinomio de grado n

Recordemos algunas propiedades referidas a polinomios.

a) Si el grado n de P es ≥ 1, $P(x)=0$ tendrá como mínimo una raíz en C. (teorema fundamental del Álgebra).

b) algoritmo de la división

$$P(x)=(x-r)q(x)+R$$

r: número
$q(x)$: $grado(n-1)$
R: residuo

Del cual se extraen

b1) Teorema del resto: $P(r)=R$

b2) Teorema de Factorización: si $p(r)=0 \Rightarrow x-r$ es un factor de $P(x)$

La utilización de polinomios en la aproximación de funciones o en remplazo de una función $f(x)$ se extiende en el Análisis Numérico, pudiendo efectuarse por los polinomios osciladores, de ubicación, cuadrados mínimos, mínimas.

Se verá el método de Horner para ubicar raíces de un polinomio P.

2.7.1. Método de Horner para ubicar raíces de un polinomio P de grado n.

$$P(x)=a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0 \qquad (1.18)$$

Sea $b_n = a_n$

Si $b_k = a_k + b_{k+1}x_0$ para $k = n-1, n-2, ..., 1, 0$. (2.20)

Será $b_0 = P(x_0)$

En el caso de $T(x) = b_n x^n + b_{n-1}x^{n-1} + ... + b_1 x + b_0$ (2.21)

Se tendrá: $P(x) = (x-x_0)T(x) + b_0$ (2.22)

Se utiliza para implementar el método de Newton en la búsqueda de raíces aproximadas de P.

2.7.2. Método de Muller

El método de Muller toma un punto de vista similar a la secante, pero proyecta una parábola a través de tres puntos

El método consiste en obtener los coeficientes de los tres puntos, sustituirlos en la fórmula cuadrática y obtener el punto donde la parábola intercepta el eje x. La aproximación es fácil de escribir, en forma conveniente esta sería:

$$f_2(x) = a(x-x_2)^2 + b(x-x_2) + c \qquad (2.23)$$

Así, se busca esta parábola para intersectar los tres puntos *[x_0, f(x_0)], [x_1, f(x_1)] y [x_2, f(x_2)]*. Los coeficientes de la ecuación anterior se evalúan al sustituir uno de esos tres puntos para dar:

$$f(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c$$
$$f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c$$
$$f(x_2) = a(x_2 - x_2)^2 + b(x_2 - x_2) + c$$

La última ecuación genera que, $f(x_2) = c$, de esta forma, se puede tener un sistema de dos ecuaciones con dos incógnitas:

$$f(x_0) - f(x_2) = a(x_0 - x_2)^2 + b(x_0 - x_2) \quad f(x_1) - f(x_2) = a(x_1 - x_2)^2 + b(x_1 - x_2)$$

Definiendo de esta forma:

$$h_0 = x_1 - x_0 \qquad h_1 = x_2 - x_1$$
$$\delta_0 = \frac{f(x_1) - f(x_2)}{x_1 - x_0} \qquad \delta_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

Sustituyendo en el sistema:

$$(h_0 - h_1)b - (h_0 + h_1)^2 a = h_0\delta_0 + h_1\delta_1$$
$$h_1 b - h_1^2 a = h_1\delta_1$$

Teniendo como resultado los coeficientes:

$$a = \frac{\delta_1 - \delta_0}{h_1 + h_0} \qquad b = ah_1 + \delta_1 \qquad c = f(x_2) \tag{2.24}$$

Hallando la raíz, se implementa la solución convencional, pero debido al error de redondeo potencial, se usará una formulación alternativa:

$$x_3 - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \text{ despejando } x_3 = x_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

La gran ventaja de este método es que se pueden localizar tanto las raíces reales como las imaginarias.

Hallando el error este será:

$$E_a = \left|\frac{x_3 - x_2}{x_3}\right| \cdot 100\ \%$$

Al ser un método de aproximación, este se realiza de forma secuencial e iterativamente, donde *x_1, x_2, x_3* reemplazan los puntos *x_0, x_1, x_2* llevando el error a un valor cercano a cero.

Generalmente el método converge para cualquier aproximación de partida.

Ejemplo 2.7

$f(x) = x^3 - 13x - 12$;h = 0,1; $x_2 = 5$ $x_1 = 5,5$ $x_0 = 4,5$

Con un análisis previo, las raíces son –3, -1 y 4

Solución

$$f(4,5) = 20,625 \qquad f(5,5) = 82,875 \qquad f(5) = 48$$

Calculando

$$h_0 = 5,5 - 4,5 = 1 \quad h_1 = 5 - 5,5 = -0,5$$
$$\delta_0 = \frac{82,875 - 20,625}{5,5 - 4,5} = 62,25 \qquad \delta_1 = \frac{48 - 82,875}{5 - 5,5} = 69,75$$

Hallando los coeficientes

$$a = \frac{69,75 - 62,25}{-0,5 + 1} = 15 \quad b = 15(-0,5) + 69,75 = 62,25 \quad c = 48$$

La raíz cuadrada del discriminante es:

$$\sqrt{62,25^2 - 4 \cdot 15 \cdot 48} = 31,544$$

Así

$$x_3 = 5 + \frac{-2 \cdot 48}{62,25 + 31,544} = 3,9765$$

Y el error estimado

$$E_a = \left| \frac{-1,0235}{x_3} \right| \cdot 100\,\% = 25,74\,\%$$

Ahora

$x_2 = 3,9765$ $x_1 = 5$ $x_0 = 5,5$

realizando diferentes iteraciones:

i	x_r	E_a %
0	5	
1	3,9465	25,740
2	4,0011	0,614
4	4,000	0,026
5	4,000	0,000

2.8. RESOLVIENDO f(x)= 0 USANDO MATLAB

fzero encuentra el cero de una función univariable

Sintaxis

x= fzero(fun,x0)

x = fzero(fun,x0,options)

[x,fval] = fzero(...)

[x,fval,exitflag] = fzero(...)

[x,fval,exitflag,output] = fzero(...)

x = fzero(fun,x0) busca el cero de *fun* cercano a *x0,* si *x0* es un escalar; *fu*n es una function handle o desde un archivo m o function anónima

El valor *x* devuelto por fzero está próximo al punto donde fun cambia de signo, arrojar NaN si la búsqueda.

Ejemplo Hallar el cero de la function escrita como anónima. $f=x^3 - 2x - 5$.

```
>> f = @(x)x.^3-2*x-5;
```

Buscando cerca de 2

```
>> z = fzero(f,2)
z =
   2.0946
```

2.9. EJERCITACIÓN DE UNIDAD II CON MATLAB

I) Dada la función *f(x)= xsenx -1*, hallar una raíz en el intervalo [0,2] por el método de la bisección y una tolerancia de 10^{-3}. Igual para x^3+4x^2-10 en [1,2] y tolerancia de 10^{-4}

```
>>f = @(x)x*sin(x)-1
>> bisect(f,0,2,0.001)
ans =
   1.1138
>>(1.1138)
ans =
 -4.9601e-004

>> bisect(f,1,2,0.0001)
ans =
   1.3652
>>f(1.3652)
ans =
 -4.9562e-004
```

II) Dada la función *f(x)*=$x-x^3+4x^2-10/3x^2+8x$, encontrar la raíz por el método de punto fijo.

```
Para la función y = x - (x.^3 + 4*x.^2 - 10)/(3*x.^2 + 8*x);
>>  fixed_point(1.5, 10)
The procedure was successful after k iterations
k =
    4
The root to the equation is
ans =
   1.3652

Para la función x- x^3-4*x^2+10
>>fixed_point(1.5, 10)
The procedure was unsuccessful
Condition |p(i+1)-p(i)| < tol was not sastified
tol =
  1.0000e-005

Please, examine the sequence of iterates
p =
  1.0e+216 *

   0.0000
  -0.0000
   0.0000
  -0.0000
   0.0000
  -0.0000
   0.0000
  -2.0827
     Inf
     NaN
     NaN
In case you observe convergence, then increase the maximum number of iterations
In case of divergence, try another initial approximation p0 or rewrite g(x)
in such a way that |g'(x)|< 1 near the root
```

Tomando su equivalente

```
y = sqrt(10./x - 4*x);
>> fixed_point(1.5, 10)
The procedure was unsuccessful
Condition |p(i+1)-p(i)| < tol was not sastified
tol =
  1.0000e-005
Please, examine the sequence of iterates
p =
   1.5000
   0.8165
   2.9969
   0 - 2.9412i
   2.7536 + 2.7536i
   1.8150 - 3.5345i
   2.3843 + 3.4344i
   2.1828 - 3.5969i
   2.2970 + 3.5741i
   2.2565 - 3.6066i
   2.2792 + 3.6019i

In case you observe convergence, then increase the maximum number of iterations
In case of divergence, try another initial approximation p0 or rewrite g(x)
in such a way that |g'(x)|< 1 near the root
```

III) Dada la función $f(x)=x-x^3+4x^2-10$, hallar una raíz por el método de Newton con aprox. inicial *1.5*, tolerancia para la raíz de 10^{-4} y 10^{-5} para los valores de *f*, con *10* iteraciones.

```
Sea la función x^3+4*x^2-10
Su derivada es 3*x^2+8*x
>>newton(f,df,1.5,0.0001,0.00001,10)
ans =
    1.3652
>> f(1.3652)
ans =
 -4.9562e-004
```

Efectuar los mismo para *f(x)= cos x-x*, con aprox. inicial *po=π/4*, tol:*0.0*1 y *0.01*, doce iteraciones.

```
f=@ (x)cos(x)-x desde po=π/4
df=@ (x)-1-sin(x)
>> f=@(x)cos(x)-x;
>> df=@(x)-sin(x)-1;
>> newton(f,df,0.25*pi,0.01,0.01,12)
ans =
    0.7395
>> f(0.7395)
ans =
 -6.9439e-004
```

IV) Dada la función $f(x)=cos(x+3)^3+(x+1)^2$, hallar sus raíces, aprox. inicial 1 y 30 iteraciones.

```
Se puede ingresar f como inline:f = inline('cos(x+3).^3+(x-1).^2');.
Como string  f = 'cos(x+3).^3+(x-1).^2'; como anónima:f = @(x) cos(x+3).^3+(x-1).^2;.
>>newtzero('cos(x+3).^3+(x-1).^2',1,30)
ans =
```

0.01246254122870
1.27535795100406

V)a) Para la función *cosx-x*, hallar una raíz con aprox. iniciales *0.5* y $\pi/4$, tolerancia para $p_1=0.01$, para *f* de *0 01* y *10* iteraciones, por el método de la secante.

f=@ (x)cos(x)-x desde po=0.5 y p1=π/4

```
>>secant(f,0.5,pi/4,0.01,0.01,10)
ans =
   0.7364
>>f(0.7364)
ans =
   0.0045
```

En cambio si

```
>>secant(f,0.5,pi/4,0.001,0.001,16)
ans =
   0.7391
>> f(0.7391)
ans =
 -2.4881e-005
```

Repetir el método para x^3+4x^2-10

Para f=@ (x) x^3+4*x^2-10

```
>> secant(f,1.38,1,0.001,0.001,10)
ans =
   1.3652
>>f(1.3652)
ans =
 -4.9562e-004
```

VI) Utilizar la falsa posición para a) x^3+4x^2-10 en *[1.38 1],* tolerancias de 10^{-3} y *10* iteraciones.

Para f=@ (x) x^3+4*x^2-10

```
>> regula(f,1.38,1,0.001,0.001,10)
ans =    1.3652
```

b)repetir para *cosx-x*

Si f=@ (x)cos(x)-x

```
>>regula(f,0.5,pi/4,0.001,0.001,10)
ans =
   0.7391
>> f(0.7391)
ans =
 -2.4881e-005
```

VII) Aplicar el método de Steffensen

Recordando el problema de f(x)=cos x –x con newton

```
>> f=@(x)cos(x)-x;
>> df=@(x)-sin(x)-1;
>> newton(f,df,0.25*pi,0.01,0.01,12)
ans =
   0.7395
>>f(0.7395)
ans =
 -6.9439e-004
```

Planteándolo con steffensen

```
>>steff(f,df,0.25*pi,0.01,0.01,4)  %sólo 4 iteraciones
ans =
   0.7391
>> f(0.7391)
ans =
 -2.4881e-005      % más cerca de cero que -6.9439e-004
```

VIII) Emplear el método de Muller para $cosx - x$ con $p_o=0.5$ $p_1=\pi/3.5$, $p_2=\pi/3.5$

f=@ (x)cos(x)-x desde po=0.5 p1=π/3.5, p2= π/3.5

```
>>muller(f,0.5,pi/3.5,pi/4,0.01,0.001,12)
ans =
   0.7390
```

EJERCICIOS PROPUESTOS UNIDAD 2

2.1. Dada $f(x) = \sqrt{x} - \cos x$ hallar p_3 en el intervalo [0,1] empleando el método de la bisección.

2.2. Para $f(x)=tanx$ hallar una raíz con exactitud de 0.001 en [4,4.5]

2.3. Con una exactitud de 10^{-5}, encuentre una raíz (bisección) para las siguientes funciones, en los intervalos respectivos:

a) $x - 2^{-x} = 0, [0,1]$ b) $e^x - x^2 + 3x - 2 = 0. [0,1]$ c) $2x\cos(2x) - (x+1)^2 = 0, [0.2, 0.3]$

d) $x\cos x - 2x^2 + 3x - 1 = 0,$ en $[0.3, 0.3]$ y $[1.2, 1.3]$

2.4. Dada $x^4-3x^2-3=0$ en [1,2], partiendo de $p_0=1,$ empleando la iteración de punto fijo, halle una solución con exactitud de 0.01.

2.5. Usando la iteración de punto fijo determine una solución para $2sen\pi x+x=0$ en [1,2], con exactitud de 0.01, partiendo de $p_0=1$

2.6. Para exactitud de 10^{-4}, obtener soluciones para las expresiones:

a) $x^3-2x^2-5=0$, en [1,4] b) $x-cosx=0$ en $[0,\pi/2]$ c) $x^3+3x^2-1=0$ en[-3.-2]

d) $x-0.8-0.2senx=0$ en $[0, \pi/2]$

Aplique en cada caso el método de Newton, secante y el de regla falsa.

2.8. Dada $f(x)=cos(x-1), p_0=2$, halle $p_0^{(1)}$ por la regla de Steffensen.

2.9. Dada $x-2^{-x}=0$ en [0,1], para exactitud de 10^{-4}, halle la raíz por Steffensen y compárela con la obtenida por el método de secant.e

2.10. Encuentre las aproximaciones, orden 10^{-4}, de los ceros de las funciones:

a) $f(x)=x^3-2x^2-5$

b) $f(x)=x^3+3x^2-1$

c) $f(x)=x^3-x-1$

por el método de Muller.

UNIDAD 3: INTERPOLACIÓN

3.1. INTERPOLACIÓN. EMPLEO DE POLINOMIOS

Frecuentemente de una serie de datos, como números reales, se busca un ajuste de ellos con posibilidad de predicción representando los polinomios una herramienta apropiada para aproximar funciones continuas, en un intervalo cerrado, según el teorema de aproximación

$$\left|f(x)-P(x)\right|\langle\varepsilon \text{ Con } \varepsilon \text{ mayor } 0 \qquad \forall x\in[a,b] \tag{3.1}$$

El uso de los polinomios tiene consigo ventajas como su derivabilidad e integrabilidad, dando también polinomio.

El planteo del problema será hallar un polinomio interpolante que brinde una aproximación precisa en todo el intervalo, de allí la no adecuación de los polinomios de Taylor que concentran la información alrededor de un punto, dígase x_0.

Entre las técnicas usadas para interpolación y predicción para reemplazar una aproximación polinómicas $P(x)$ por $f(x)$, se citan:

I.- Fórmulas de diferencias centrales como Stirling y Bessel, menos para valores cercanos al inicio o final de la tabla de valores.

II.- Fórmulas de adelanto de Newton para valores cercanos al inicio de la tabla.

III.- Fórmula de retroceso de Newton, para final de la tabla.

IV.- Fórmula de Lagrange, aunque debe elegirse de partida el grado de *P(x)*.

V.- Polinomios de oscilación y el de Taylor pueden emplearse también.

3.2. POLINOMIO DE LAGRANGE

Sean $x_0, x_1, \ldots x_n$, $(n+1)$ puntos y f una función que pasa por esos puntos, entonces habrá un polinomio *P* único de grado máximo *n,* que dispondrá de la propiedad.

$$f(x_k)=P(x_k) \qquad k=0,1,2,\ldots n \tag{3.2}$$

$$P(x)=f(x_0)L_{n,0}(x)+\ldots..+f(x_n)L_{n,n}(x)=\sum_{k=0}^{n}f(x_k)L_{n,k}(x) \tag{3.3}$$

Con

$$L_{n,k}(x)=\frac{(x-x_0)(x-x_1)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)(x_k-x_1)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)} \tag{3.4}$$

$$L_{n,k}(x)=\prod_{\substack{i=0\\ i\neq k}}^{n}\frac{(x-x_i)}{(x_k-x_i)} \qquad k=0,1,2,\ldots n$$

Errores involucrados en la aproximación.

Las posibles causas de error se pueden agrupar en:

I.- Error de ingreso: Cuando los valores de $f(x_k)$ dados son inexactos.

II.- Error de truncado: diferencia $f(x)-P(x)$ en el momento de adoptar la aproximación polinómica:

$$f(x)=P(x)+\frac{f^{n+1}(\zeta)}{(n+1)!}\prod_{i=0}^{n}(x-x_i) \tag{3.5}$$

Para *f* perteneciente a la clase $C^{n+1}[a,b]$, con $\zeta(X)$ un número en $]a,b[$.

III.- Error de redondeo: error ligado al algoritmo y surge de las calculadoras y su empleo para productos, cocientes y la pérdida de dígitos involucrados.

Específicamente el uso de los polinomios lagrangianos conlleva inconvenientes como:

- Es dificultosa la aplicación del término de error, entonces el grado del polinomio requerido para una precisión anhelada no se conoce antes de efectuar las operaciones.
- El trabajo para generar las aproximaciones consecutivas no disminuye.

Ejemplo 3.1- Para los siguientes datos, generar el polinomio de Lagrange.

x	y
5	1
-7	-23
-6	-54
0	-954

Recordando que los coeficientes se obtienen con:

$$l_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

$$l_o(x) = \frac{(x+7)(x+6)x}{(5+7)(5+6)5} \qquad l_1(x) = \frac{(x-5)(x+6)x}{(-7-5)(-7+6)(-7)}$$

$$l_2(x) = \frac{(x-5)(x+7)x}{(-6-5)(-6+7)(-6)} \quad l_3(x) = \frac{(x-5)(x+7)(x+6)}{(0-5)(0+7)(0+6)}$$

Obteniéndose el polinomio de Lagrange

$$p_3(x) = l_0(x) - 23l_1(x) - 54l_2(x) - 954l_3(x)$$

3.2. MÉTODO DE NEVILLE

Técnica para producir repetidamente aproximaciones con polinomios de Lagrange, empleando los cálculos previos.

Si se representa, para *f* definida en $x_0, x_1, ..., x_k$.

Se necesitará un punto inicial y el número de puntos adicionales empleados en la generación de la aproximación.

Si $N_{i,j}$ con $0 \le i \le j$ representa el polinomio interpolante de grado *j* en los $(j+1)$ números

$x_{i-j,}, x_{i-j+1}, ..., x_{i-1}, x_i$

$$N_{i,j} = P_{i-j,i-j+1,...,i-1,i} \qquad (3.6)$$

Se dispondrá de un diagrama

$x_0 \; P_0 = N_{0,0}$

$x_1 \; P_1 = N_{1,0} \; P_{0,1} = N_{1,1}$

$x_2 \; P_2 = N_{2,0} \; P_{1,2} = N_{2,1} \; P_{0,1,2} = N_{2,2}$

$x_3 \; P_3 = N_{30} \; P_{2,3} = N_{3,1} \; P_{1,2,3} = N_{3,2} \; P_{0,1,2,3} = N_{3,3}$

$x_4 \; P_4 = N_{4,0} \; P_{3,4} = N_{4,1} \; P_{2,3,4} = N_{4,2} \; P_{1,2,3,4} = N_{4,3} P_{0,1,2,3,4} = N_{4,4}$

Ejemplo 3.2 Utilizar el algoritmo de Neville para calcular sucesivas aproximaciones de la función $f(x)= ln\ x$ para $x = 2.1$, conocidos sus valores en $x_0=2.0,\ x_1= 2.2$ y $x_2= 2.3$

i	x_i	$x-x_i$	$P_i=lnx_i$	$P_{i-1,i}$	$P_{i-2,i-1,i}$
0	2.0	0.1	$P_0=0,6931$		
1	2.2	0.1	$P_1=0,7885$	$P_{0,1}=(x-x_0)P_1-(x-x_1)P_0/x_1-x_0=0,7410$	$P_{0,1,2}=(x-x_0)P_{1,2}-(x-x_2)P_{0,1}/(x_2-x_0)=0,7420$
2	2.3	0.2	$P_2=0,8329$	$P_{1,2}=(x-x_1)P_2-(x-x_2)P_1/x_2-x_1=0,7441$	

3.3. DIFERENCIAS DIVIDIDAS

Las técnicas de diferencias divididas sirven para producir sucesivamente los polinomios por si mismos.

Tomando los puntos $x_0, x_1, ..., x_k$, se genera la primera diferencia dividida entre x_0, x_1 por:

$$f(x_0, x_1) = \frac{f_1 - f_0}{x_1 - x_0} \tag{3.7}$$

Para otra x_i se emplea modo similar de generación entre parejas.

Una segunda diferencia se conformará a partir de la primera:

$$f(x_0, x_1, x_2) = \frac{f(x_1, x_2) - f(x_0, x_1)}{x_2 - x_0} \tag{3.8}$$

Y para orden n

$$f(x_0, x_1, ..., x_n) = \frac{f(x_1, ..., x_n) - f(x_0, ..., x_{n-1})}{x_n - x_0} \tag{3.9}$$

Por ejemplo se generará una tabla de diferencias:

	1° diferencia	2° diferencia	3° diferencia
$x_0\ \ f_0$			
	$f(x_0, x_1)$		
$x_1\ \ f_1$		$f(x_0, x_1, x_2)$	
	$f(x_1, x_2)$		$f(x_0, x_1, x_2, x_3)$
$x_2\ \ f_2$		$f(x_1, x_2, x_3)$	
	$f(x_2, x_3)$		
$x_3\ \ f_3$			

Si se representa como: $D_i^n(x_i) = (x_i - x_0)(x_i - x_1)...(x_i - x_{i-1})...(x_i - x_n)$

Con Z^+ se podrá generalizar la representación por:

$$f(x_0, x_1, ..., x_n) = \sum_{i=0}^{n} \frac{f_i}{D_i^n(x_i)} \tag{3.10}$$

Una propiedad importante de las diferencias divididas es la simetría: son invariantes para todas las permutaciones de los x_k.

La vinculación entre diferencias divididas y las derivadas, para: $f \in C^n[a,b]$ con $x_0, x_1, ..., x_n$ puntos diferentes en $[a,b]$ viene dada por:

$$f(x_0,x_1,...,x_n)=\frac{f^{(n)}(\zeta)}{n!} \quad \text{con } \zeta\in\left]a,b\right[\tag{3.11}$$

Con la presentación de las diferencias divididas se podrá obtener el polinomio $P(x)$ a través de:

$$P(x)=f\left[x_0\right]+(x-x_0)f\left(x_0,x_1\right)+(x-x_0)(x-x_1)f\left(x_0,x_1,x_2\right)+...+(x-x_0)(x-x_1)...(x-x_{n-1})f(x_0,...,x_n) \tag{3.12}$$

$$P(x)=f\left[x_0\right]+\sum_{k=1}^{n}f\left[x_0,...,x_k\right]\left(x-x_0\right)...\left(x-x_{k-1}\right) \tag{3.13}$$

Esa es la fórmula de diferencias divididas de Newton.

Una simplificación de la fórmula de interpolación de diferencias divididas se obtiene considerando igual espaciado entre los símbolo, $x_0,x_1,...,x_n$

Denotando $h=x_{i+1}-x_i$ para $i=0,1,...,n-1$ con $x=x_0+sh$.

Será $x-x_i=\left(s-i\right)h$, convirtiendo a $P(x)=\sum_{k=o}^{n}s\left(s-1\right)...\left(s-k+1\right)h^k f\left[x_0,x_1,...,x_k\right]$

(3.14) fórmula de Newton de diferencias divididas hacia adelante.

$$P(x)=P\left(x_n+sh\right)=f\left[x_n\right]+shf\left[x_{n-1},x_n\right]+s\left(s+1\right)h^2 f\left[x_{n-2},x_{n-1},x_n\right]+...+s\left(s+1\right)...\left(s+n-1\right)h^n f\left[x_0,x_1,..\right.$$

(3.15) fórmula de Newton de diferencias divididas hacia atrás.

Ahora si se busca aproximar un valor de *x* que se ubica cerca del centro de la tabla, no son adecuadas las expresiones de Newton.

Se emplean las fórmulas de diferencias centradas, según sea el caso, y dada la amplitud de expresiones se mencionan la de Stirling.

Para ella, con x_0 elegido cerca del punto a aproximar, llamando $x_1,x_2...$ los puntos debajo de x_0, y $x_{-1},x_{-2}...$ situados por encima de x_0, la fórmula de Stirling

$$\begin{aligned}
&P(x)=P_{2m+1}\left(x\right)=f\left[x_0\right]+\frac{sh}{2}\left(f\left[x_1,x_0\right]+f\left[x_0,x_1\right]\right)+s^2h^2f\left[x_{-1},x_1,x_0\right]\\
&+\frac{s\left(s^2-1\right)h^3}{2}\left(f\left[x_{-1},x_0,x_1,x_2\right]+f\left[x_{-2},x_{-1},x_0,x_1\right]\right)+...\\
&+s^2\left(s^2-1\right)\left(s^2-4\right)...\left(s^2-\left(m-1\right)^2\right)h^{2m}f\left[x_{-m},....x_m\right]\\
&+\frac{s\left(s^2-1\right)...\left(s^2-m^2\right)h^{2m+1}}{2}\left(f\left[x_{-m},....x_{m+1}\right]+f\left[x_{-m-1},....x_m\right]\right)
\end{aligned} \tag{3.16}$$

Si *n=2m+1* es impar, y si *n=2m* es par se elimina el último término.

Ejemplo 3.3- A manera de ejemplo con una cantidad mayor de puntos, determinemos por el método de diferencias divididas de Newton el polinomio interpolante que pasa por los puntos *(1,4);(3,1),(4.5) (7,3*). El arreglo triangular en este caso toma la forma específica:

1	4			
3	1	−1.5000		
4.5	5	2.6667	1.1905	
7	3	−0.8000	−0.8667	−0.3429

El polinomio de grado tres será:

$$P(x) = 4 - 1.5000(x-1) + 1.1905(x-1)(x-3) - 0.3429(x-1)(x-3)(x-4.5)$$

$$= -0.3429x^3 + 4.1048x^2 - 13.4619x + 13.7000$$

3.4. POLINOMIOS OSCULADORES. POLINOMIOS DE HERMITE.

Avanzando hacia una generalización de los polinomios de Lagrange como de Taylor, se presentan los polinomios osciladores.
Estos polinomios concuerdan con el valor de una función dada en argumentos característicos y sus derivadas hasta un cierto orden se corresponden con las derivadas de tal función, generalmente en idéntico argumentos.
Así para la situación más sencilla se requiere:

$$P(x_k) = f(x_k) \qquad P'(x_k) = f'(x_k) \qquad k = 0,1,...,n \tag{3.17}$$

Cual es el sentido geométrico: será hacer tangente las curvas que representan nuestras dos funciones en los $(n+1)$ puntos.

Osculación de orden superior requerirá
$P''(x_k) = f''(x_k)$ y así sucesivamente, generándose contacto de orden superior entre las curvas correspondientes.
Se presenta básicamente a los polinomios de Hermite, con grado $\leq 2n+1$ y oscilación de primer orden. Sea $f \in C[a,b]$, con $x_0,....,x_n \in [a,b]$ números diferentes, el polinomio único que coincide con f y f' en $x_0,...x_n$ es el polinomio de grado $\leq 2n+1$ que se expresa por

$$P_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) P_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \overline{P}_{n,j}(x) \tag{3.18}$$

Siendo $P_{n,j}(x) = \left[1 - 2(x - x_j) L'_{n,j}(x_j)\right] L^2_{n,j}(x_j)$ (3.19)

$$\overline{P}_{n,j}(x) = (x - x_j) L^2_{n,j}(x_j) \tag{3.20}$$

Con $L_{n,j}$ el j-ésimo polinomio de coeficientes de Lagrange (grado *n*)

Para $f \in C^{2n+2}[a,b]$ se tendrá para el error:

$$f(x) - P_{2n+1}(x) = \frac{(x-x_0)^2 ... (x-x_n)^2}{(2n+2)!} f^{2n+2}(\zeta) \tag{3.21}$$

Con $\zeta \in]a,b[$.

Dado lo laborioso de determinar los polinomios de Lagrange con sus derivadas, alternativamente para aproximar por Hermite se hace uso de la fórmula de diferencia para los polinomios de Lagrange.
Considerando n puntos $x_0, x_1...., x_n$ con sus valores de f y f', se genera una nueva sucesión $w_0, w_1...., w_{2n+1},$ a partir $w_{2i} = w_{2i+1} = x_i$ a para $i = 0,1,...,n$ así se producirá una tabla de diferencias como ya se mencionó para $w_0, w_1...., w_{2n+1},$.
Al ser $w_{2i} = w_{2i+1} = x_i$ para todo *i*, $f[w_{2i}, w_{2i+1}]$ lo sustituimos por $f'(x_i)$ usándolo $f'(x_0), f'(x_1)...f'(x_n)$ en vez de $f[w_0, w_1], f[w_1, w_2], f[w_{2n}, w_{2n+1}]$..

Se seguirá la metodología acostumbrada para el resto de las diferencias divididas, finalizando con el uso de las diferencias divididas adecuadas en las f} fórmulas de interpolación de diferencias divididas de Newton.
Se esquematizan las primeras diferencias.

w $f(w)$	$w_0 = x_0$ $f[w_0] = f[x_0]$		$w_1 = x_0$ $f[w_1] = f[x_0]$		$w_2 = x_1$ $f[w_2] = f[x_1]$		$w_3 = x_1$ $f[w_0, w_1]$ ...
		$f[w_0, w_1] = f'[x_0]$		$f[w_1, w_2] = \frac{f[w_2] - f[w_1]}{w_2 - w_1}$		$f[w_2, w_3] = f'(x_1)$	

3.5. APROXIMACIÓN POLINÓMICA FRAGMENTADA

El camino de aproximar funciones determinadas por los polinomios, considerando el carácter oscilante de los últimos sobre todo en grados elevados, puede inducir que fluctuaciones en una región reducida del dominio conlleve mayores fluctuaciones en el recorrido.
Una alternativa vendrá dada por generar polinomios de aproximación para diferentes particiones del intervalo de estudio: aproximación fragmentada.
El caso más simple será la unión de puntos $\{(x_0, f(x_0))(x_1, f(x_1))...(x_n, f(x_n))\}$ por segmentos rectilíneos, es decir una aproximación lineal, válidas para funciones trigonométricas y logarítmicas, cuando se buscan valores medios de una tabla de valores; pero nada se podrá asegurar sobre el comportamiento en extremo de los intervalos particionados.
El empleo de los polinomios por parte ofrece la ventaja de no requerir conocimiento de las derivadas de la función a aproximar.
El empleo de polinomios cúbicos garantiza la derivabilidad en el intervalo y posee derivadas segundas continuas, sin suponer que coincidan derivadas del polinomio interpelantes con las de la función.

3.6. ADAPTADOR CÚBICO

Se define el adaptador cúbico como:
Para f definida en $[a,b]$ y nodos $a = x_0 \langle x_1 ... \langle x_n = b$, el adaptador cúbico A para f, debe satisfacer.

i) A es un polinomio cúbico, A_j, en el subintervalo para $j = 0,1,...,n-1$

ii) $A(x_j) = f(x_j) \qquad j = 0,1,...,n$

iii) $A_j(x_{j+1}) = A_j(x_{j+1}) \qquad j = 0,1,...,n-2$

iv) $A'_{j+1}(x_{j+1}) = A'_j(x_{j+1}) \qquad j = 0,1,...,n-2$

v) $A''_{j+1}(x_{j+1}) = A''_j(x_{j+1}) \qquad j = 0,1,...,n-2$

vi) Según sea cota natural o fija, debe cumplir las de borde

A. $A''(x_0) = A''(x_n) = 0$ natural

B. $A''(x_0) = f''(x_0) \wedge A'(x_n) = f'(x_n)$ fija

El caso B conducirá a aproximaciones más precisas pero se debe contar con los valores de las derivadas en nodos terminales.

3.6.1. Generación del adaptador cúbico

Para una función *f* dada, aplicando lo definido para interpelantes cúbicos

$$A_j(x) = a_j + b_j\left(x - x_j\right) + c_j\left(x - x_j\right)^2 + d_j\left(x - x_j\right)^3 \qquad (3.22)$$

$$j = 0,1,...,n-1$$

$$A_j\left(x_j\right) = a_j = f\left(x_j\right)$$

Introduciendo iii)

$$a_{j+1} = A_{j+1}(x_{j+1}) = A_j\left(x_{j+1}\right) = a_j + b_j\left(x_{j+1} - x_j\right) + c_j\left(x_{j+1} - x_j\right)^2 + d_j\left(x_{j+1} - x_j\right)^3$$

$$j = 0,1,...,n-2$$

Llamando *h* al paso entre nodos para $j = 0,1,...,n-1$ y $a_n = f\left(x_n\right)$ se tendrá.

$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$ (3.23) se verifica para $j = 0,1,...,n-1$

Análogamente, llamando $b_n = A'\left(x_n\right)$

$$A'_j(x) = b_j + 2c_j\left(x - x_j\right) + 3d_j\left(x - x_j\right)^2$$

Con $A'\left(x_n\right) = b_n$ para $j = 0,1,...,n-1$

Al aplicar iv) $b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$ $\qquad j = 0,1,...,n-1$ (3.24)

Si se hace $c_n = \dfrac{A''\left(x_n\right)}{2}$, aplicando v):

$c_{j+1} = c_j + 3d_j h_j \qquad j = 0,1,...,n-1$ (3.25)

Extrayendo d_j de (3.25) para sustituirlo en (3.23) y (3.24) se presentan las ecuaciones

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3}\left(2c_j + c_{j+1}\right) \qquad (3.26)$$

$$b_{j+1} = b_j + h_j\left(c_j + c_{j+1}\right) \qquad (3.27)$$

De (3.26) de determinará b_j a través de:

$$b_j = \frac{1}{h_j}\left(a_{j+1} - a_j\right) - \frac{h_j}{3}\left(2c_j + c_{j+1}\right) \qquad (3.28)$$

Y para *j*-1 $b_{j-1} = \dfrac{1}{h_{j-1}}\left(a_j - a_{j-1}\right) - \dfrac{h_{j-1}}{3}\left(2c_{j-1} + c_j\right)$ (3.29)

Que llevándolos a (3.27) se genera el sistema de ecuaciones

$$h_{j-1}c_{j-1} + 2\left(h_{j-1} + h_j\right)c_j + h_j c_{j+1} = \frac{3}{h_j}\left(a_{j+1} - a_j\right) - \frac{3}{h_{j-1}}\left(a_j - a_{j-1}\right) \qquad (3.30)$$

para $j = 0,1,...,n-1$

Los valores de los conjuntos $\{h_j\}_{j=0}^{n-1}$ y $\{a_j\}_{j=0}^{n}$ vienen dados por los espaciamientos entre los x_j y los valores de f en los nodos, entonces el conjunto $\{c_j\}_{j=0}^{n}$ serán las incógnitas.
Al conocer los c_j, se podrán hallar los coeficientes b_j de (3.28) y d_j de (3.25), para poder generar los polinomios cúbicos $\{A_j\}_{j=0}^{n-1}$.
Para cualquiera de las condiciones dadas por a) y b), garantizarán poder hallar los c_j a partir de (3.30)

Ejemplo 3.4- La relación liquido-polvo que se debe poner a una la mezcla proporciona la resistencia final que se le quiere dar al producto de mezcla. Se tienen los siguientes datos:

x=liquido/polvo[%]	40	45	50	55	60	65	70
y=Resistencia[kg/cm^2]	390	340	290	250	210	180	160

Efectuar una interpolación por spline cúbico natural

x_i	y_i	h_i	Δ_i
40	390	5	-10
45	340	5	-10
50	290	5	-8
55	250	5	-8
60	210	5	-6
65	180	5	-4
70	160		

Planteando la forma matricial del sistema de ecuaciones

$$\begin{pmatrix} 20 & 5 & 0 & 0 & 0 \\ 5 & 20 & 5 & 0 & 0 \\ 0 & 5 & 20 & 5 & 0 \\ 0 & 0 & 5 & 20 & 5 \\ 0 & 0 & 0 & 5 & 20 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = 6 \begin{pmatrix} 0 \\ 2 \\ 0 \\ 2 \\ 2 \end{pmatrix}$$

obteniendo las soluciones:

$v_1 = -0.181538,\ v_2 = 0.726154,\ v_3 = -0.323077,\ v_4 = 0.566154,\ v_5 = 0.458462$

además $v_0 = 0,\ v_6 = 0$ por tratarse de un spline natural

i	a_i	b_i	c_i	d_i
0	-0.00605128	0	-9.84872	390
1	0.0302564	-0.0907692	-10.3026	340
2	-0.0349744	0.363077	-8.94103	290
3	0.029641	-0.161538	-7.93333	250
4	-0.00358974	0.283077	-7.32564	210
5	-0.0152821	0.229231	-4.7641	180

3.7. INTERPOLANDO CON MATLAB

La función *interp1* realiza una interpolación unidimensional, su forma general es:
>>interp1(x,y,xi,method)
y es un vector conteniendo los valores de una funciónis
x es un vector de la misma longitude con los puntos para los cuales se calcula *y*
xi vector de puntos donde se desea interpolar
method es un string opcional detallando el método de interpolación, entre ellos:'*nearest*', '*linea*r' (es el por default de *interpol1*)
Interpolación por spline (adaptador) cubico('*cubic*'), es similar a 'pchip'(emplea Hermite a trozos)

Interpolación bidimensional
La función *interp2* efectúa la interpolación bidimensional, su forma más general es del tipo:
>>ZI = interp2(X,Y,Z,XI,YI,method)
Z arreglo rectangular con los bidimensionales valores de la function
X e Y arreglos de igual tamaño de los puntos para los valores de Z
XI e *YI* son matrices de puntos a los cuales se interpolan losa datos
method es un string opcional que especifica el método: 'nearest', interpolación bilineal(`linear'), interpolación bicúbica('cubic'), aplicable cuando los datos interpolados y sus derivadas deben ser continuas, siempre requiriendo que X e Y estén siempre creciendo o decreciendo(si están igualmente espaciados se aclara con '*cubic'

3.8. EJERCITACIÓN DE UNIDAD III CON MATLAB

I) Dados los siguientes valores x-y:
x=[1.0 1.3 1.6 1.9 2.2];
y=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
encontrar el polinomio interpolante de Lagrange
```
>> X=[1.0 1.3 1.6 1.9 2.2];
>> Y=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
>> lagran(X,Y)
ans =
   0.0018    0.0553   -0.3430    0.0734    0.9777
```
II) Con los datos de I) generar las diferencias divididas y el polinomio de Newton
```
>> xi=[1.0 1.3 1.6 1.9 2.2];
>> fi=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
```

```
>> divdiff1 ( xi, fi )
ans =
   0.7652   -0.4837   -0.1087    0.0659    0.0018
```
(coef. según fla. hacia delante para el pol. interpolante)

```
>> X=[1.0 1.3 1.6 1.9 2.2];
>> Y=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
>> newpoly(X,Y)
ans =
   0.0018    0.0553   -0.3430    0.0734    0.9777
```
(coef. del pol. interp. de Newton)

III) Con los datos de I) generar el valor de *x=1.4* por Neville
```
>> x=[1.0 1.3 1.6 1.9 2.2];
>>f=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
>>neville1(x, f, 1.4)
ans =
   0.5668
```
Con el algoritmo de neville2, para el mismo punto, grado 4
```
>> x=[1.0 1.3 1.6 1.9 2.2];
>> Q=[0.7651977 0.6200860 0.4554022 0.2818186 0.1103623];
>> neville2(1.4,4,x,Q)
ans =
   0.5668
```

IV) Con los valores de *x,y*, las derivadas de *y*, generar el polinomio de Hermite:
```
x=[1.3 1.6 1.9];  y=[0.6200860 0.4554022 0.2818186];>> yp=[-0.5220232 -0.5698959 -
.5611571]
>> x=[1.3 1.6 1.9];
>> y=[0.6200860 0.4554022 0.2818186];
>> yp=[-0.5220232 -0.5698959 -.5611571]; % derivadas
>> hermite ( x, y, yp )
ans =
  Columns 1 through 5

    0.6145   -4.7291   14.5472  -22.4247   16.8115
  Column 6

   -4.0722                          % coefic. del polinomio
```
Para evaluar el valor del polinomio, ej en x=1.45, se usa polyval
```
>> p=[0.6145   -4.7291   14.5472  -22.4247   16.8115 -4.0722];
>> polyval(p,1.45)
ans =
   0.5393
```
En más puntos:
```
>> polyval(p,[1.38 1.45 1.5])
ans =
   0.5775    0.5393    0.5116
```

V) Dados los valores de *x-f(x)*, encontrar el trazador lineal:
```
 xi=[0.1 0.2 0.3 0.4]; fi=[-0.62049958 -0.2839668 0.00660095 0.24842440];
```

```
>> xi=[0.1 0.2 0.3 0.4];
>> fi=[-0.62049958 -0.2839668 0.00660095 0.24842440];
>> linear_spline ( xi, fi )
   0.1000   -0.6205    3.3653
   0.2000   -0.2840    2.9057
   0.3000    0.0066    2.4182
   0.4000    0.2484         0
```

% la tercera columna da las pendientes de los trozos lineales

Los valores del pol se obtienen con *spline_eval*

```
>> sp=[3.3653 2.9057 2.4182 0];
>> x=[0.1 0.2 0.3 0.4];
>> spline_eval ( sp, x )
-4.9904   -4.7486   -4.5068   -4.2650
```

VI) Dados los valores de *x,f(x)* y las derivadas en extremos, encontrar el trazador cúbico sujeto

x=[0.1 0.2 0.3 0.4]; y=[-0.62049958 -0.2839668 0.00660095 0.24842440];

dx0=3.58502082; dxn=2.16529366.

```
>> X=[0.1 0.2 0.3 0.4];
>> Y=[-0.62049958 -0.2839668 0.00660095 0.24842440];
>> dx0=3.58502082;
>> dxn=2.16529366
>> csfit(X,Y,dx0,dxn)
ans =

  -0.5266   -2.1443    3.5850   -0.6205
  -0.4468   -2.3022    3.1404   -0.2840
  -0.4657   -2.4363    2.6665    0.0066
```

% las filas son los coeficientes de los interpolantes cúbicos

Igual para los siguientes valores

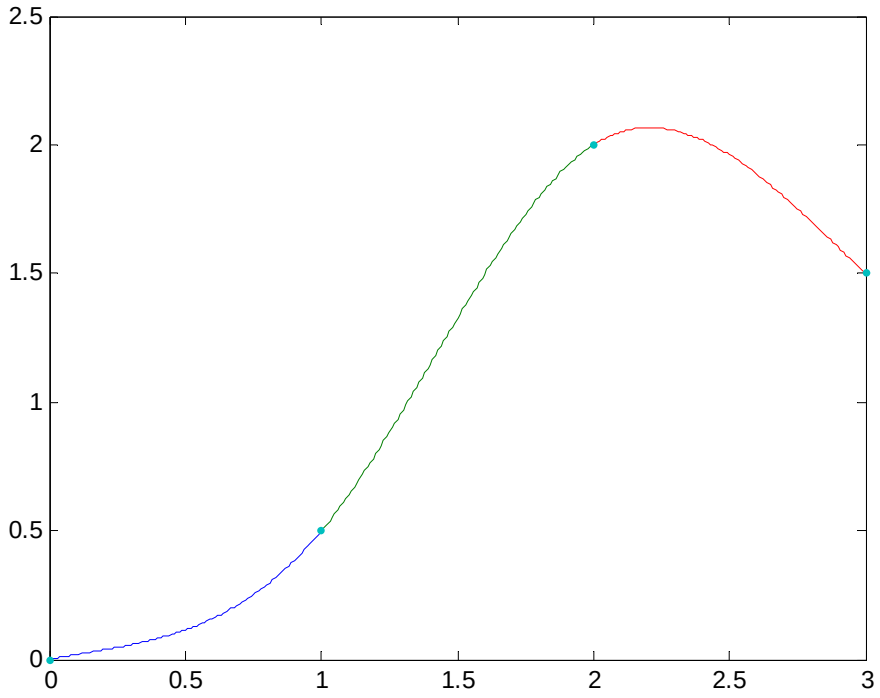
```
> X=[0 1 2 3 ];
>> Y=[0 0.5 2.0 1.5];
>> dx0=0.2;
>> dxn=-1;
>> csfit(X,Y,dx0,dxn)

S =

   0.4800   -0.1800    0.2000         0
  -1.0400    1.2600    1.2800    0.5000
   0.6800   -1.8600    0.6800    2.0000
```

VII) Graficar la aproximación

```
>>  x1=0:.01:1;y1=polyval(S(1,:),x1-X(1));
>>x2=1:.01:2;y2=polyval(S(2,:),x2-X(2));
>> x3=2:.01:3;y3=polyval(S(3,:),x3-X(3));
>> plot(x1,y1,x2,y2,x3,y3,X,Y,'.')
```


2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3

EJERCICIOS PROPUESTOS PARA UNIDAD 3

3.1. Emplee los polinomios de Lagrange de grados uno, dos y tres para aproximar lo siguiente:

a) f(8.4) si f(8,1)=16.94410, f(8.3)=17.56492,f(8.6)=18.50515,f(8.7)=18.82091

b) f(/1/3) si f(-0.75)=-0.07181250, f(-0.5)=-0.02475000, f(-0.25)=0.334993750

3.2. Para los casos del ejercicio anterior emplee la técnica de **Neville.**

3.3. Genere las diferencias divididas para obtener los coefcientes del polinomio interpolante en los casos a y b de 3.1.

3.4. A partir de las siguientes tablas de datos, encuentre el polinomio de Hermite.

a)

x	f(x)	f'(x)
8.3	17.56492	3.116256
8.4	18.50515	3.151762

b)

x	f(x)	f'(x)
-0.5	-0.0247500	0.7510000
-0.25	0.3349375	2.1890000
0	1.1010000	4.002000

3.5. Dados las siguientes tablas de datos, generar el adaptador lineal en cada caso

a)

x	f(x)	f'(x)
-0.5	-0.0247500	0.7510000
-0.25	0.3349375	2.1890000
0	1.1010000	4.002000

b)

x	f(x)	f'(x)
0.1	-0.62049958	3.58502082
0.2	-0-28398668	
0.3	0.00660095	
0.4	0.24842440	2.16529366

3.6. Encuentre el interpolante cúbico libre y el sujeto para los casos a) y b) del ejercicio anterior.

UNIDAD 4: DIFERENCIACIÓN E INTEGRACIÓN NUMÉRICAS

4.1. FÓRMULAS DE APROXIMACIÓN

Recordando que para f en x_0, se define la derivada como

$$\lim_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h} \tag{4.1}$$

Con $x_0 \in]a,b[, f \in C^2[a,b], h \neq 0,$ considerando el punto x_1 con un incremento h *en* x_0 utilizando el polinomio de Lagrange $P_{0,1}$ con su error se tendrá:

$$\begin{aligned} f(x) &= P_{0,1} + \frac{(x-x_0)(x-x)}{2!} f''(\zeta(x)) \quad \text{con } \zeta(x) \text{ en } [a,b] \\ &= \frac{f(x_0)(x-x_0-h)}{-h} + \frac{f(x_0+h)(x-x_0)}{h} + \frac{(x-x_0)(x-x_0-h)}{2} f''(\zeta(x)) \end{aligned} \tag{4.2}$$

Al derivar la expresión

$$f'(x) \approx \frac{f(x_0+h)-f(x_0)}{h} \tag{4.3}$$

Como no se conoce $f'''(\zeta(x))\zeta'(x)$, es decir la derivada de $f''(\zeta(x))$ se hace problemática la determinación del error de truncado, pero en $x = x_0$ su coeficiente se anula, quedando la expresión

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{h}{2} f''(\zeta) \tag{4.4}$$

Ahora el cociente incremental $\frac{[f(x_0+h)-f(x_0)]}{h}$, con h pequeños, permitirá aproximar $f'(x_0)$ con una cota para el error de $\frac{Kh}{2}$, siendo K cota de $f''(x)$, generándose $j = 0,1,...,\ j = 0,1,...,\ j = 0,1,...,$ la expresión conocida como fórmula de diferencias hacia adelante para $h\rangle 0$ (para atrás si $h\langle 0$

Pasando a un plano más general, para un intervalo con $(n+1)$ puntos diferentes $\{x_0, x_1, ..., x_n\}$ con $f \in C^{n+1}$ sobre dicho intervalo

$$f'(x) = \sum_{j=0}^{n} f(x_j) L'_j(x) + \frac{f^{n+1}(\zeta(x))}{(n+1)!} \prod_{\substack{j=0 \\ j\neq 0}}^{n} (x_k - x_j) \tag{4.5}$$

Que combina linealmente $(n+1)$ valores de $f(x)$, para $j = 0,1,...,n$, conocida como fórmula de $(n+1)$ puntos.

Las fórmulas más empleadas involucran 3y 5 puntos, aunque más puntos implican más precisión pero a la vez crece el error de redondeo y la necesidad de evaluar las funciones.

Se avanza en la simplificación de tamaño de paso constante entre nodos, permitiendo las expresiones:

$$f(x)=\frac{1}{2h}\left[-3f(x_0)+4f(x_0+h)-f(x_0+2h)\right]+\frac{h^2}{3}f^3(\zeta_0) \qquad (4.6)$$

Con ζ_0 entre x_0 y x_1+2h, y

$$f(x)=\frac{1}{2h}\left[f(x_0+h)-f(x_0-h)\right]+\frac{h^2}{6}f^{(3)}(\zeta_1) \qquad (4.7)$$

Con ζ_1 entre (x_0-h) y (x_0+h)

Estas dos últimas ecuaciones se conocen como fórmulas de tres puntos, observándose que en (2.7) los datos se consideran a ambos lados de x_0 además de evaluar *f* en solo dos puntos, con un error equivalente a la mitad de (2.6).

Para las expresiones de cinco puntos

$$f'(x_0)=\frac{1}{12h}\left[f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)\right]+\frac{h^4}{30}f^{(5)}(\zeta) \qquad (4.8)$$

y

$$f'(x_0)=\frac{1}{12h}\left[-25f(x_0)+48f(x_0+h)-36f(x_0+2h)+16f(x_0+3h)+3f(x_0+4h)\right]+\frac{h^4}{5}f^{(5)}(\zeta) \qquad (4.9)$$

Con ζ_0 entre x_0 y x_0+4h

Se puede aproximar por izquierda con $h\rangle 0$ o por derecha $h\langle 0$

Las fórmulas presentadas, necesarias para aproximar las soluciones de ecuaciones diferenciales ordinarias o en derivadas parciales, presentan problemas de estabilidad ya que en el compromiso de reducir *h* para disminuir el error de truncado induce el error de redondeo dada la división entre números pequeños.

4.2. TÉCNICA DE RICHARDSON

Esta técnica suele aplicarse si se conoce el error, dependiente de *h,* permitiendo trabajar con buena precisión y fórmulas de orden bajo.

Llamando *R(h)* la expresión que aproxima el valor no conocido R, con un error de truncado $O(h)$, para ciertas constantes $M_1, M_2, M_3...$:

$$R=R(h)+M_1h+M_2h^3+M_3h^3+... \qquad (4.10)$$

Se busca mejorar la expresión de $O(h)$, aunque no es sencilla obtener *R(h)*, para *h* pequeños.

La expresión general será:

$$R=R(h)+\sum_{j=1}^{m-1}M_jh^j+O(h^m)\ \ 2.11\ \text{ para } c/\ j=2,3,...,m \qquad (4.11)$$

Existiendo una aproximación para cada *j* del tipo:

$$R(h)=R_{j-1}+\frac{R_{h-1}(h/2)-R_{j-1}(h)}{2^{j-1}-1} \qquad (4.12)$$

La técnica (extrapolante) es aplicable siempre y cuando el error de truncado para una fórmula sea del tipo:

$$\sum_{j=1}^{m-1}M_jh^{\alpha j}+O(h^{\alpha m}) \qquad (4.13)$$

Para constantes M_j y que se verifique $\alpha_1\langle\alpha_1\langle...\alpha_1\langle\alpha_m$

Las aproximaciones que se van produciendo pueden englobarse en una tabla

$O(h)$ $O(h^2)$ $O(h^3)$ $O(h^4)$

$R_1(h) \equiv R(h)$

. . . .

. . . .

. . . .

Se observa que las columnas que se obtienen luego de la primera se logran calculando las medias, se tenderá a aproximaciones con poco cálculo, error de redondeo y buen orden, pero que debido a que la diferenciación numérica es inestable, el error de redondeo en $R_1(\frac{h}{2}k)$ crecerá con *k*.

Ejemplo 4.1.- Sea la función $f(x) = xe^x$ y sus valores

x	$f(x)$
1,8	10,889365
1,9	12,703199
2,0	14,778112
2,1	17,148957
2,2	19,855030

Encontrar el valor aproximado de $f'(x)$ en $x_0 = 2$

Con la expresión de tres puntos 2.7

$$f'(x_0) = \frac{1}{2h}\left[f(x+h) - f(x_0 - h)\right] - \frac{h^2}{6}f^{(3)}(\xi)$$

Para h=0,1

$$f'(2) = 5\left[f(2,1) - f(1,9)\right] = 22,03310$$

Con la 2.6 para *h*=0,1, se tiene:

$$f'(2) = 5\left[-3f(2) - 4f(2,1) - f(2,2)\right] = 22,03310$$

Usando 4.6 aparece un error de 0,1348 y con 2.7 de -0,0616 (prácticamente la mitad), pues el valor exacto de $f'(x) = (x+1)e^x$ en $x_0 = 2$ es 22,167168.

Recordando 4.13, como condición para aplicar la extrapolación, es decir cuando el error de truncado sea del tipo

$\sum_{j=1}^{m-1} M_j h^{\alpha j} + O(h^{\alpha m})$; La expresión 4.7 para $f'(x_0)$ puede extenderse como:

$$f'(x_0) = \frac{1}{2h}\left[f(x_0+h) - f(x_0-h)\right] + \frac{h^2}{6}f^{(3)}(x_0) + \frac{h^4}{120}f^{(5)}(x_0) + ...$$

Es decir que *h* a potencias pares para el error, así la extrapolación será $\alpha_j = 2j$ pues

$R_1(h) \equiv R(h) = \frac{1}{2h}\left[f(x_0+h) - f(x_0-h)\right]$ Y para cada $j = 2,3,..., O(h^{2j})$, pues

$R_j(h) \equiv R_{j-1}(h/2) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{4^{j-1} - 1}$ pues se están eliminando potencias de h^2 en lugar de *h*. Para *h=0,2,* x_0=2, entonces:

$$R_1(0,2) = R(0,2) = 2,5\left[f(2,2) - f(1,8)\right] = 22,41460$$

$$R_1(0,1) = R(0,1) = 22,228786$$

$$R_1(0,05) = R(0,05) = 22,182564$$

Extrapolando estos datos se obtienen la tabla:

$$R_1(0,2) = 22,414160$$

$$R_1(0,1) = 22,228786$$

$$R_1(0,05) = 22,182564$$

$$R_2(0,2) = R_1(0,1) + \frac{R_1(0,1) - R(0,2)}{3} = 22,166995$$

$$R_2(0,1) = R_1(0,05) + \frac{R_1(0,05) - R(0,1)}{3} = 22,167157$$

$$R_3(0,2) = R_2(0,1) + \frac{R_2(0,1) - R_2(0,2)}{15} = 22,167168$$

4.3. INTEGRACIÓN NUMÉRICA

Es bastante común la necesidad de encontrar la integral definida de una función cuya primitiva no es sencilla de hallar, por lo tanto se debe aproximar $\int_a^b f(x)dx$, empleando sumas del tipo $\sum_{i=0}^{n} \alpha f(x_i)$.

La aproximación polinómica es un recurso básico para un vasto rango de fórmulas de integración, basado en que si *P(x)* aproxima a *f(x)*, se tendrá:

$$\int_a^b P(x)dx \approx \int_a^b f(x)dx \qquad (4.14)$$

4.3.1. Fórmula de Newton

Para un intervalo que comprende a x y x_n, para grado n, se tendrán

$$\int_{x_0}^{x_1} P(x)dx = \frac{h}{2}\left(f(x_0) + f(x_1)\right)$$

$$\int_{x_0}^{x_2} P(x)dx = \frac{h}{3}\left(f(x_0) + 4f(x_1) + f(x_2)\right) \qquad (4.15)$$

$$\int_{x_0}^{x_3} P(x)dx = \left(\frac{3h}{8}\right)\left(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right)$$

Los errores de truncado de estas expresiones son:

$$\int_{x_0}^{x_n} f(x)dx - \int_{x_0}^{x_n} P(x)dx$$

Que pueden adoptar formas del tipo

$$n = 1 \qquad \frac{h^3 f''(\zeta)}{12}$$

$$n = 2 \qquad \frac{h^5 f'''(\zeta)}{90} \quad \text{para } \zeta \text{ entre } x_0 \text{ y } x_n \qquad (4.16)$$

Una extensión de estas fórmulas son las llamadas fórmulas compuestas que aplican las fórmulas simples para intervalos más prolongados, en forma repetida, presentando la ventaja del uso de un polinomio único de grado superior.

4.3.2. Regla del Trapecio

Se expresa por

$$\int_{x_0}^{x_n} f(x)dx = \frac{h}{2}\left[f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n)\right] \quad (4.17)$$

Quizá la más simple de las fórmula compuestas, usando segmentos de recta como aproximación a $f(x)$.

El error de truncado es proporcional a $\frac{(x_n - x_0)h^2 f^2(\zeta)}{12}$ (4.18)

4.3.3. Regla de Simpson

Emplea segmentos parabólicos como aproximación a $f(x)$ y es de extenso uso en integración numérica.

$$\int_{x_0}^{x_n} f(x)dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right] \quad (4.19)$$

Con un error de truncado aproximado de $\frac{(x_n - x_0)h^4 f^4(\zeta)}{180}$

Las fórmulas del trapecio y de Simpson son particulares dentro de una clase de técnicas conocidas como de Newton-Cotes, que presenta formas abiertas y cerradas, en el primer caso incluye nodos del intervalo abierto (x_0, x_n) y en el segundo incluye los puntos extremos del intervalo cerrado $[x_0, x_n]$.

Las técnicas de integración compuesta de Newton-Cotes presentan estabilidad frente al error de redondeo, haciéndose estable a medida que el tamaño de paso tiende a cero, en distinción de las técnicas de diferenciación numéricas vistas anteriormente.

4.3.4. Integración de Romberg

Representa una acción de refinamiento de los métodos vistos hasta aquí, empleando la regla compuesta de los trapecios para una primaria aproximación y posteriormente utiliza la técnica de Richardson para mejoramientos de las aproximaciones.

Si expresamos la regla trapecial compuesta por:

$x_0 = a$ y $x_n = b$

$$\int_{x_0}^{x_n} f(x)dx = \frac{h_k}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{2^{k-1}-1} f(a + ih_k)\right] - \frac{b-a}{12}h_k^2 f^{(2)}(\zeta_k) \quad (4.20)$$

Con $h_k = \frac{x_n - x_0}{m_k} = \frac{b-a}{2^{k-1}}$ y ζ_k esta en $(\mathrm{a,b})$

Denotaremos por $R_{k,1}$ las partes de 4.20 para la aproximación trapecial:

$$R_{1,1} = \frac{h_1}{2}\left[f(a) + f(b)\right] = \frac{(b-a)}{2}\left[f(a) + f(b)\right]$$

$$R_{1,1} = \frac{h_2}{2}\left[f(a) + f(b) + f(a + h_2)\right] = \frac{(b-a)}{4}\left[f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right)\right]$$

$$= \frac{1}{2}\left[R_{1,1} + h_1 f(a + h_2)\right]$$

Generalizando

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1} \sum_{i=1}^{k-2} f(a + (2i-1)h_k) \right] \qquad k = 2,3,...,n \tag{4.21}$$

Para un orden de error $O(h_k^4)$

$$\int_a^b f(x)dx - \left[R_{k+1,1} + \frac{R_{k+1,1} - R_{k,1}}{3} \right] = \sum_{i=2}^{\infty} \frac{E_i}{3} \left(\frac{h_k^{2i}}{4^{i-1}} - h_k^{2i} \right) = \sum_{i=2}^{\infty} \frac{E_i}{3} \left(\frac{1-4^{i-1}}{4^{i-1}} \right) h_k^{2i} \tag{4.22}$$

con E_i para cada *i* no depende de h_k, dependiendo solo de las derivadas de orden (2i-1) en *a* y en *b.*

A esta expresión se le aplica la extrapolación en busca de un orden h_k^6 y así seguidamente.

Para cada $k = 2,3,4,...,n$ y $j = 2,...,k$ se puede reducir la notación expresando

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}$$

Conformando un diagrama característico

$R_{1,1}$

$R_{2,1}$ $R_{2,2}$

$R_{3,1}$ $R_{3,2}$ $R_{3,3}$

.

. O

$R_{n,1}$ L L L $R_{n,n}$

Cada fila completa se obtiene por aplicación de la regla compuesta de los trapecio y utilizando los valores ya obtenidos, se evalúan los restantes términos de la fila, o sea un orden $R_{1,1}, R_{2,1}, R_{2,2}, R_{3,1}$, etc.

Operativamente es procedente fijar una tolerancia de error para aproximar y definir *n*, hasta que los sucesivos valores de $R_{n-1,n-1}$ y $R_{n,n}$ coincidan dentro de una tolerancia, con una cota superior.

Existen otras expresiones más complejas que involucran términos de corrección o incluso el desarrollo del integrando en serie de potencias de Taylor.

Ejemplo 2.2. Aplicar el algoritmo de integración de Romberg a la integral:

$$\int_1^3 \frac{e^x}{x} dx$$

Tomando ϵ_3=0.01%

Se efectúan los cálculos correspondientes a uno, dos, cuatro y ocho subintervalos:

$$I(h_1) = \frac{3-1}{2}\left[\frac{e^1}{1} + \frac{e^3}{3} \right] = 9.413460803$$

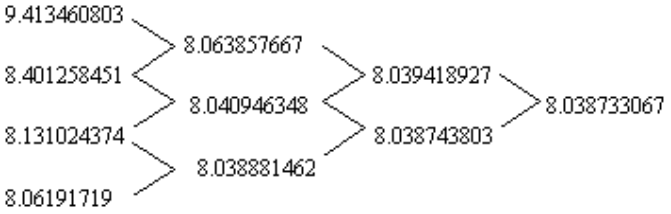
$$I(h_2) = \frac{3-1}{4}\left[\frac{e^1}{1} + 2\left(\frac{e^2}{2} \right) + \frac{e^3}{3} \right] = 8.401258451$$

$$I(h_3) = \frac{3-1}{8}\left[\frac{e^1}{1} + 2\left(\frac{e^{1.5}}{1.5} + \frac{e^2}{2} + \frac{e^{2.5}}{2.5} \right) + \frac{e^3}{3} \right] = 8.131024374$$

$$I(h_4) = \frac{3-1}{16}\left[\frac{e^1}{1}+2\left(\frac{e^{1.25}}{1.25}+\frac{e^{1.5}}{1.5}+\frac{e^{1.75}}{1.75}+\frac{e^2}{2}+\frac{e^{2.25}}{2.25}+\frac{e^{2.5}}{2.5}+\frac{e^{2.75}}{2.75}\right)+\frac{e^3}{3}\right] = 8.06191719$$

Aplicando estos valores se efectúan los cálculos hasta el nivel 4, según la expresión recursiva:

$$I_{j,k} \approx \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1}-1}$$ y para el error $$|\varepsilon_a| = \left|\frac{I_{j,k} - I_{j,k-1}}{I_{j,k}}\right| \times 100\% < \varepsilon_s$$



Haciendo los cálculos de los errores, la aproximación se obtiene hasta el nivel 4, donde $|\varepsilon_a| = 0.008\%$.

Entonces la aproximación buscada es:

$$\int_1^3 \frac{e^x}{x}dx \approx 8.038733067$$

4.4. FÓRMULAS GAUSSIANAS

Hasta ahora las fórmulas presentadas suponen espaciamientos iguales cuando la realidad indica la presencia de funciones que pueden estimarse para cualquier argumento.
Entonces, la presentación general será:

$$\int_a^b \mu(x) f(x)dx \cong \sum_{i=1}^{n} a_i f(x_i) \qquad (4.24)$$

Con $\mu(x)$ una función de peso, que en el caso de $\mu(x) = 1$ se presenta la forma más simple.
Si $f(x)$ es función potencial x^{2n+1}, se generan $2n$ condiciones para encontrar las $2n$ x_i y a_i, es decir

$$a_i = \int_a^b \mu(x) L_i(x)dx \qquad (4.25)$$

con $L_i(x)$ función multiplicadora de Lagrange.
Los $x_1, ..., x_n$ representan los ceros del polinomio $P_n(x)$, de grado n, de una familia que verifica ortogonalidad:

$$\int_a^b \mu(x) P_n(x) P_m(x)dx = 0 \qquad para \qquad m \neq n$$

con los coeficientes dependientes de $\mu(x)$, entonces $\mu(x)$ ejerce influencia

$$\int_{-1}^{1} \frac{P_n(x)}{x-x_k}dx = \frac{-2}{(n+1)P_{n+1}(x_k)}$$

$$e \cong \frac{1}{2n+1}\left[f(1)+f(-1)-Int-\sum_{i=1}^{n} a_i f'(x_i) \right]$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

(4.26)

$$\int_0^{\infty} e^{-x} f(x)dx \cong \sum_{i=1}^{n} a_i f'(x_i)$$

$$\beta(x) = (x-x_1)...(a,b) = (-1,1)$$

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^2-1\right)^n \qquad P_0(x_i) = 1, a_i = \frac{2\left(1-x_i^2\right)}{n^2\left|P_{n-1}(x_i)\right|^2}$$

sobre a_i, x_i aunque no esté taxativamente presente en la fórmula de Gauss.

La fórmula gaussiana posee un error de truncado dado por:

$$\int_a^b \mu(x)f(x)dx - \sum_{i=1}^{n} a_i f(x_i) = \frac{f^{(2n)}(\zeta)}{2n!}\int_a^b \mu(x)\left|\beta(x)\right|^2 dx \qquad (4.27)$$

Con $\beta(x) = (x-x_1)...(x-x_n)$

Para polinomios de grado menores a *2n-1*, las fórmulas tienen gran exactitud pues son proporcionales a $f^{(2n)}$, a diferencia de las expresiones de integración no gaussianas que eran proporcionales a $f^{(n)}$.

Dependiendo de $\mu(x)$, y el intervalo para integrar, se tienen fórmulas específicas gaussianas.

<u>Fórmula de Gauss-Legendre</u>

$\mu(x) = 1$

$(a,b) = (-1,1)$

Los polinomios ortogonales son polinomios de Legendre

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^2-1\right)^n \qquad P_0(x) = 1 \qquad (4.28)$$

x_i raíces del polinomio

Y $a_i = \dfrac{2\left(1-x_i^2\right)}{n^2\left|P_{n-1}(x_i)\right|^2}$ (4.29)

Tanto los x_i como los a_i están tabulados para reemplazar en

$$\int_a^b f(x)dx \cong \sum_{i=1}^{n} a_i f(x_i) \qquad (4.30)$$

Propiedades de los polinomios de Legendre

I) $\int_{-1}^{1} x^k P_n(x)dx = 0 \qquad k = 0,1,...,n-1$ (4.31)

II) $$\int_{-1}^{1} x^n P_n(x)dx = \frac{2^{n+1}(n!)^2}{(2n+1)!} \tag{4.32}$$

III) $$\int_{-1}^{1} \left[P_n(x)\right]^2 dx = \frac{2}{(2n+1)} \tag{4.33}$$

IV) $$\int_{-1}^{1} P_m(x)P_n(x)dx = 0 \qquad con\ m \neq n \tag{4.34}$$

V) $$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \tag{4.35}$$

VI) $$\int_{-1}^{1} \frac{P_n(x)}{x - x_k} dx = \frac{-2}{(n+1)P_{n+1}(x_k)} \tag{4.36}$$

El error de truncado se puede aproximar por

$$e \cong \frac{1}{2n+1}\left[f(1) + f(-1) - Int - \sum_{i=1}^{n} a_i f'(x_i) \right] \tag{4.37}$$

Int= integral aproximada de la formula gaussiana de n puntos.

Fórmula de Gauss-Laguerre

$$\int_0^{\infty} e^{-x} f(x)dx \cong \sum_{i=1}^{n} a_i f'(x_i) \tag{4.38}$$

Con x_i los ceros del polinomio de Laguerre de orden n:

$$L_n(x) = e^x \frac{d^n}{dx^n}\left(e^{-x}x^n\right) \tag{4.39}$$

y

$$a_i = \frac{(n!)^2}{x_i\left[L'(x_i)\right]^2} \tag{4.40}$$

x_i *y* a_i también se disponen en tablas.

Fórmula de Gauss-Chebyshev

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}} dx \cong \left(\frac{\pi}{n}\right)\sum_{i=1}^{n} f(x_i) \tag{4.41}$$

El polinomio n-simo de Chebyshev: $T_n(x) = \cos(n \arccos(x))$

x_i los ceros del polinomio $T_n(x)$.

4.5. INTEGRANDO NUMÉRICAMENTE DESDE MATLAB

Las funciones de cuadratura que emplea Matlab son
quad usa cuadratura adaptativa de Simpson
quadl usa cuadratura adaptativa de Lobatto
dblquad evalúa la integral doble numéricamente
triplequad evalúa la integral triple numéricamente
ejemplo: sea integrar la función definida en el archivo *fun2.m* de 0 a 1
f=sqrt(1+x.^2)-tan(x);

>>quad(@fun2,0,1)
ans =

0.5322

4.6. EJERCITACIÓN DE UNIDAD 4 CON MATLAB

I) Dada la función x^2-3x, hallar su derivada en x=3 con tolerancia 0.01

```
>> f=@(x)x^2-3*x;
>> difflim(f,3,0.01)
ans =
   1.0000    3.0000         0
   0.1000    3.0000    0.0000
   0.0100    3.0000    0.0000
```

% la salida es L=[H' D' E']: H vector de tamaños de paso; D el de deriv. aprox, E error

II) Hallar la integral de la función f(x)=sen x entre 0 y $\pi/3$

i) Trapezoidal compuesta

```
>> f=@(x)sin(x);
>> traprl(f,0,pi/3,6)
ans =
0.4987
```

ii) simpson compuesta

```
>> simprl(f,0,pi/3,6)
ans =
   0.5000
```

iii)trapezoidal recursiva

```
ctrap(f,0,pi/3,5)
ans =
  Columns 1 through 5
    0.4534    0.4885    0.4971    0.4993    0.4998
  Column 6
    0.5000
```

iV) Romberg

```
romber(f,0,pi/3,6,0.001)
ans =
   0.4534         0         0         0         0
   0.4885    0.5002         0         0         0
   0.4971    0.5000    0.5000         0         0
   0.4993    0.5000    0.5000    0.5000         0
   0.4998    0.5000    0.5000    0.5000    0.5000
```

Adaptativas

i) adapt. Con simpson

```
> adapt(f,0,pi/3,0.0001)
ans =
  Columns 1 through 5
         0    1.0472    0.5002    0.5000    0.0000
  Column 6
    0.0001
```

EJERCICIOS PROPUESTOS PARA UNIDAD 4

4.1. Usar la fórmula más precisa para determinar las aproximaciones que completen las tablas

a)

x	f(x)	f'(x)
2.1	-1,709847	
2.2	-1.373823	
2.3	-1.119214	
2.4	-0.9160143	
2.5	-0.7470223	
2.6	-0.6015966	

b)

x	f(x)	f'(x)
-3.0	9.367879	
2.2	8.233241	
2.3	7.180350	
2.4	6.209329	
2.5	5.320305	
2.6	4.513417	

4.2. Aproximar las integrales

a) $\int_1^{1.5} x^2 \ln x dx$ b) $\int_0^1 x^2 e^{-x} dx$ c) $\int_1^{1.6} \frac{2x}{x^2-4} dx$

empleando las técnicas de i) trapezoidal recursiva, ii) trapezoidal compuesta, iii) Simpson compuesta.

4.3. Aproximar las integrales

a) $\int_1^{1.5} x^2 \ln x dx$ b) $\int_0^1 x^2 e^{-x} dx$ c) $\int_3^{3.5} \frac{x}{(x^2-4)^{1/2}} dx$

empleando la técnica de Romberg, halle $R_{3,3}$

4.4. Empleando el método adaptativo de Simpson, resuelva las aproximaciones de los casos a,b y c de 3.3

UNIDAD 5: ECUACIONES DIFERENCIALES ORDINARIAS

5.1. PROBLEMA DE VALOR INICIAL

El modelado de problemas de valores iniciales involucra la necesidad de resolver una ecuación diferencial sujeta a una condición inicial determinada.
En la búsqueda de aproximar una solución, cuando se torna dificultosa arribar a una solución exacta, existen métodos para efectuarla directamente, aunque el alcance de lo que se presentará comprende aproximaciones en determinados puntos.
Recordando primero el concepto de función lipschtziana como $f(t,x)$ para un dominio D_f en $\mathbf{R}^2$.

$$|f(t,y_1)-f(t,y_2)| \le L|y_2-y_1| \qquad (5.1)$$

$(t,y_1)\, y\, (t,y_2) \in D_f$ y L = cte. de Lipschitz.
O la condición de mayor utilidad

$$\text{Si } \left|\frac{df}{dy}(t,y)\right| \le L \qquad \forall (t,y) \in D_f \subset \mathbf{R}^2 \qquad (5.2)$$

Si $L\rangle 0$ f es lipschtziana sobre D_f respecto a la variable y

5.2. EXISTENCIA Y UNICIDAD PARA ECUACIONES DIFERENCIALES DE PRIMER ORDEN

Si $f(t,y)$ es continua en $D=\{(t,y)/a\le t\le b, -\infty\langle y\langle\infty\}$ y f es lipschtziana en el dominio, el problema de valor inicial (P.V.I.).

$$\begin{aligned} y' &= f(t,y) \qquad a\le t\le b \\ y(a) &= \alpha \end{aligned} \qquad (5.3)$$

tendrá solución única $y(t)$ para el intervalo de t.

5.2.1. P.V.I. bien planteado

El P.V.I.

$$\begin{aligned} y' &= f(t,y) \qquad a\le t\le b \\ y(a) &= \alpha \end{aligned} \qquad (5.4)$$

Será un problema bien planteado cuando

i. Exista $y(t)$ única
ii. Para cualquier $\delta\rangle 0$, habrá un $m(\delta)$ constante que cumpla para $|\delta_0|\langle\delta$ y $\varepsilon(t)$ continua con $|\varepsilon(t)|\langle\delta$ sobre $[a,b]$, una solución única al problema.
iii. $\dfrac{dz}{dt} = f(t,z)+\varepsilon(t) \qquad a\le t\le b \qquad z(\alpha)=\alpha+\delta_0$

Existe de modo que

$$|z(t)-y(t)|\langle m(\delta)\delta \qquad \forall t\in[a,b] \qquad (5.5)$$

El problema presentado por 5.5 se denomina problema con perturbación, asociado al P.V.I. original.
Las condiciones que garantizan que el P.V.I. esté bien planteado se pueden condensar en:
Sea $D_f=\{(t,y)/a\le t\le b, -\infty\langle y\langle\infty\}$, si f es continua y verifica una condición de Lipschitz en y sobre el D_f, el P.V.I. dado por

$$\begin{cases} y' = f(t,y) & a \le t \le b \\ y(a) = \alpha \end{cases}$$ esta bien planteado (5.6)

5.3. MÉTODOS PARA RESOLVER P.V.I.

5.3.1. Método de Euler

Utiliza la ecuación de diferencia

$$w_{i+1} = w_i + hf(t_i, w_i) \quad (5.7)$$

h=paso entre nodos consecutivos

$w_i \approx y(t_i)$

En realidad, es un desprendimiento del teorema de existencia pues no es una aproximación muy exacta de $w' = f(t_i, w_i)$, siempre bajo consideraciones adecuadas de $f(t,w)$.

Si M es cota de $y''(t)$, o sea $\left|y''(t)\right| \le M \quad \forall t \in [a,b]$ (5.8)

Con $y(t)$ solución única al P.V.I. dado por

$$y' = f(t,y) \qquad a \le t \le b \qquad y(a) = \alpha$$

Y $w_0, ..., w_N$ las aproximaciones de Euler para $N \in Z^+$, se tendrá:

$$\left|y(t_i) - w_i\right| \le \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right] \quad (5.9)$$

L=cte de Lipschitz

$i = 0,1,2,...,N$

De la expresión de la fórmula de cota para el error se observa que es proporcional a h decreciendo h crecerá el error de redondeo al ser necesarios más cálculos.

Entonces se suele emplear para el método una expresión del tipo:

$\mu_0 = \alpha + \delta_0$

$$\mu_{i+1} = \mu_i + hf(t_i, w_i) + \delta_{i+1} \qquad i = 0,1,2,...,N-1 \quad (5.10)$$

Siendo δ_i error de redondeo asociado con μ_i.

El mínimo valor de $E(h)$ se da para $h = \sqrt{2\delta / M}$

O sea h

Superiores inducirán a incrementar el error total en la aproximación.

Dada la siguiente ecuación diferencial con la condición inicial:

$$y' = 2xy$$
$$y(0) = 1$$

Ejemplo con Euler Aproximar y(0.5), con un tamaño de paso 0.1

Sustituyendo estos datos en la formula de Euler, en el primer paso:

$$\begin{cases} x_1 = x_0 + h = 0.1 \\ y_1 = y_0 + hf(x_0, y_0) = 1 + 0.1[2(0)(1)] = 1 \end{cases}$$

en un segundo paso:

$$\begin{cases} x_2 = x_1 + h = 0.2 \\ y_2 = y_1 + h f(x_1, y_1) = 1 + 0.1[2(0.1)(1)] = 1.02 \end{cases}$$

Sintetizando los valores hasta x=0.5 en la tabla

n	x_n	y_n
0	0	1
1	0.1	1
2	0.2	1.02
3	0.3	1.0608
4	0.4	1.12445
5	0.5	1.2144

El error involucrado será

$$|\varepsilon_r| = \left| \frac{1.28402 - 1.2144}{1.28402} \times 100\% \right| = 5.42\%$$

5.3.2. Serie de Taylor

Reconociendo al error de truncamiento local por

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - \theta(t_i, y_i) \qquad i = 0,1,2,...,N-1 \tag{5.11}$$

Para Euler en el i-esimo paso se tendrá

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - f(t_i, y_i) \qquad i = 0,1,2,...,N-1 \tag{5.12}$$

Con $y_i = y(t_i)$, el valor exacto de la solución en t_i.

Se lo denomina local pues mide la precisión del método en un paso específico, suponiendo exactitud del método en el paso precedente.

En Euler, el error de truncamiento es $O(h)$, lo ideal será encontrar métodos con errores locales $O(h^p)$ con p lo mayor posible.

Dado que Euler se origina del desarrollo de Taylor pone n=1, la convergencia podrá suponerse mejoraría con n mayores.

Así se tendrá, para $\zeta_i \; en \; (t_i, t_{i+1})$

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!} f^{(n)}\left(\zeta_i, y(s_i)\right) \qquad i = 0,1,2,...,N-1 \tag{5.13}$$

En el caso que $y \in C^{n+1}[a,b], y^{(n+1)}(t) = f^{(n)}(t, y(t))$ está acotada en $[a,b]$ y $\tau_i = O(h^n)$ para $i = 0,1,2,...,N$.

5.3.3. Métodos de Runge-Kutta

Las desventajas de las técnicas de Taylor estriban en la necesidad del cálculo de derivadas de orden alto, hacia esto se apunta con la introducción de los métodos de Runge-Kutta aunque los errores no son tan fáciles de visualizar.

Así en reemplazo de las derivadas se emplean valores extras de $f(t, y)$.

Las fórmulas más difundidas son

$$\kappa_1 = hf(t, y)$$

$$\kappa_2 = hf\left(t + \frac{h}{2}, y + \frac{\kappa_1}{2}\right) \qquad O(h^2)$$

$$\kappa_3 = hf\left(t + \frac{h}{2}, y + \frac{\kappa_2}{2}\right) \qquad O(h^3) \qquad (5.14)$$

$$\kappa_4 = hf\left(t + h, y + \kappa_3\right) \qquad O(h^4)$$

$$y(t+h) \approx y(t) + \frac{1}{6}\left(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4\right) \qquad \textit{cuarto orden}$$

Existiendo muchas variaciones según el orden y tipo de algoritmos.

Ejemplo 5.1. Determine *y (0.5)* utilizando el método de Runge-Kutta de cuarto orden, en el intervalo de interés *[0, 0.5],* en 5 nodos, para el siguiente PVI:

y' =4e$^{0.8x}$ – 0.5y ; y(0) =2 ; y(0.5) =?

el paso será: h =0.5 / 5=0.1

por lo tanto $x_0 = 0$, $x_1 = 0.1$, $x_2 = 0.3$, $x_4 = 0.4$, $x_5 = 0.5$

ITERACIÓN I i =0 ; x_0 =0 ; y_0 =2

*K_1 =f [0, 2] =4e$^{(0.8*0)}$ – (0.5 * 2)*

K_1 =3

*K_2 =f [0 +0.1/2, 2 +(0.1 *3) /2] =f [0.05, 2.15] =4e$^{(0.8*0.05)}$ – (0.5 * 2.15)*

K_2 =3.088243

*K_3 =f [0 +0.1/2, 2 +(0.1 *3.088243) /2] =f [0.05, 2.154412]*

*K_3 =4e$^{(0.8*0.05)}$ – (0.5 * 2.154412)*

K_3 =3.086037

K_4 =f [0 +0.1, 2 +(0.1 *3.086037)] =f [0.1, 2.308603]

K_4 =4e$^{(0.8*0.1)}$ – (0.5 * 2.308603)

K_4 =3.178846

*y_1(0.1) =2 +{0.1 /6 [3 +(2 *3.088243) +(2 *3.086037) +3.178846]}*

y_1(0.1) =2.308790

ITERACIÓN II $i = 1 ; x_1 = 0.1 ; y_1 = 2.308790$

$K_1 = f[0.1, 2.308790] = 4e^{(0.8*0.1)} - (0.5 * 2.308790)$

$K_1 = 3.178753$

$K_2 = f[0.1 + 0.1/2, 2.308790 + (0.1 * 3.178753)/2] = f[0.15, 2.467727]$

$K_2 = 4e^{(0.8*0.15)} - (0.5 * 2.467727)$

$K_2 = 3.276123$

$K_3 = f[0.1 + 0.1/2, 2.308790 + (0.1 * 3.276123)/2] = f[0.15, 2.472596]$

$K_3 = 4e^{(0.8*0.15)} - (0.5 * 2.472596)$

$K_3 = 3.273689$

$K_4 = f[0.1 + 0.1, 2.308790 + (0.1 * 3.273689)] = f[0.2, 2.636158]$

$K_4 = 4e^{(0.8*0.2)} - (0.5 * 2.636158)$

$K_4 = 3.375964$

$y_2(0.2) = 2.308790 + \{0.1/6 [3.178753 + (2 * 3.276123) + (2 * 3.273689) + 3.375964]\}$

$y_2(0.2) = 2.636362$

ITERACIÓN III $i = 2 ; x_2 = 0.2 ; y_2 = 2.636362$

$K_1 = f[0.2, 2.636362] = 4e^{(0.8*0.2)} - (0.5 * 2.636362)$

$K_1 = 3.375862$

$K_2 = f[0.2 + 0.1/2, 2.6366362 + (0.1 * 3.375862)/2] = f[0.25, 2.805155]$

$K_2 = 4e^{(0.8*0.25)} - (0.5 * 2.805155)$

$K_2 = 3.483033$

$K_3 = f[0.2 + 0.1/2, 2.636362 + (0.1 * 3.483033)/2] = f[0.25, 2.810513]$

$K_3 = 4e^{(0.8*0.25)} - (0.5 * 2.810513)$

$K_3 = 3.480354$

$K_4 = f[0.2 + 0.1, 2.636362 + (0.1 * 3.480354)] = f[0.3, 2.984397]$

$K_4 = 4e^{(0.8*0.3)} - (0.5 * 2.984397)$

$K_4 = 3.592798$

$y_3(0.3) = 2.636362 + \{0.1/6 [3.375862 + (2 * 3.483033) + (2 * 3.480354) + 3.592798]\}$

$y_2(0.3) = 2.984619$

ITERACIÓN IV $i = 3 ; x_3 = 0.3 ; y_3 = 2.984619$

$K_1 = f[0.3, 2.984619] = 4e^{(0.8*0.3)} - (0.5 * 2.984619)$

$K_1 = 3.592687$

$K_2 = f[0.3 + 0.1/2, 2.984619 + (0.1 * 3.592687)/2] = f[0.35, 3.164253]$

$K_2 = 4e^{(0.8*0.35)} - (0.5 * 3.164253)$

$K_2 = 3.710392$

$K_3 = f[0.3 + 0.1/2, 2.984619 + (0.1 * 3.710392)/2] = f[0.35, 3.170138]$

$K_3 = 4e^{(0.8*0.35)} - (0.5 * 3.170138)$

$K_3 = 3.707450$

$K_4 = f[0.3 + 0.1, 2.984619 + (0.1 * 3.707450)] = f[0.4, 3.355364]$

$K_4 = 4e^{(0.8*0.4)} - (0.5 * 3.355364)$

$K_4 = 3.830829$

$y_4(0.4) = 2.984619 + \{0.1/6 [3.592687 + (2 * 3.710392) + (2 * 3.707450) + 3.830829]\}$

$y_2(0.4) = 3.355606$

ITERACIÓN V $\quad i = 4 ; x_4 = 0.4 ; y_4 = 3.355606$

$K_1 = f[0.4, 3.355606] = 4e^{(0.8*0.4)} - (0.5 * 3.355606)$

$K_1 = 3.830708$

$K_2 = f[0.4 + 0.1/2, 3.355606 + (0.1 * 3.830708)/2] = f[0.45, 3.547141]$

$K_2 = 4e^{(0.8*0.45)} - (0.5 * 3.547141)$

$K_2 = 3.959747$

$K_3 = f[0.4 + 0.1/2, 3.355606 + (0.1 * 3.959747)/2] = f[0.45, 3.553593]$

$K_3 = 4e^{(0.8*0.45)} - (0.5 * 3.553593)$

$K_3 = 3.956521$

$K_4 = f[0.4 + 0.1, 3.355606 + (0.1 * 3.956521)] = f[0.5, 3.751258]$

$K_4 = 4e^{(0.8*0.5)} - (0.5 * 3.751258)$

$K_4 = 4.091669$

$y_5(0.5) = 3.355606 + \{0.1/6 [3.830708 + (2 * 3.959747) + (2 * 3.956521) + 4.091669]\}$

La **solución requerida es** $\quad$ **$y_5(0.5) = 3.751521$**

Si se hubiera empleado Taylor, los resultados son semejantes pues ambos reproducen la serie de h^4.

El mayor trabajo en los métodos R-K es el cálculo de *f;* así en los de segundo orden $O(h^2)$ se requieren dos evaluaciones de *f* por paso, en el de cuarto orden habrá cuatro evaluaciones por paso.

5.3.4. MÉTODOS MULTIPASOS

Las técnicas vistas hasta aquí emplean únicamente la información lograda en el intervalo que se está aproximando. A las técnicas que emplean la aproximación en más de uno de los puntos anteriores de malla, para lograr en el punto siguiente, se conocen como multipasos.
Se define de la siguiente manera:
Dado el P.V.I. $y' = f(t, y) \quad a \le t \le b \qquad y(a) = b$ (5.15)
La ecuación de diferencias w_{i+1} en t_{i+1} $\left(m \in Z^+, y \rangle 1\right)$ será:

$$w_{i+1} = c_{m-1}w_i + c_{m-2}w_{i-1} + ... + c_0 w_{i+1-m} + h\left[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + ... + b_0 f(t_{i+1-m}, w_{i+1-m})\right] \quad (5.16)$$

$i = m-1, m, ..., N-1$
Valores iniciales dados $w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, ..., w_{m-1} = \alpha_{m-1}$
Paso $h = \dfrac{b-a}{N}$
Si $b_m \neq 0, w_{i+1}$ se presenta en ambos lados de 5.16, o sea implícitamente, y para $b_m = 0, w_{i+1}$ está en función de valores definidos, o sea explícitamente.
Dentro de la variedad de expresiones para este tipo de técnicas se presentarán algunas de ellas.

5.3.4.1. Métodos de Adams-Bashforth

A. De dos pasos

$$w_0 = \alpha \qquad w_1 = \alpha_1$$

$$w_{i+1} = w_i + \frac{h}{2}\left[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\right] \qquad i = 1, 2, ..., N \quad (5.17)$$

$$\tau_{i+1}(h) = \frac{5}{2} y^3(\mu_i) h^2 \quad \text{con } \mu_i \text{ entre } t_{i-1}, t_{i+1}$$

B. de tres pasos

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, w_3 = \alpha_3$$

$$w_{i+1} = w_i + \frac{h}{24}\left[55f(t_{i+1}, w_{i+1}) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) + 9f(t_{i-3}, w_{i-3})\right] \quad (5.18)$$

$$\tau_{i+1}(h) = \frac{251}{720} y^{(5)}(\mu_i) h^4$$

C. De cinco pasos

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, w_3 = \alpha_3, w_4 = \alpha_4$$

$$w_{i+1} = w_i + \frac{h}{720}\left[1901f(t_i, w_i) - 2774f(t_{i-1}, w_{i-1}) + 2616f(t_{i-2}, w_{i-2}) - 1274f(t_{i-3}, w_{i-3}) + 251f(t_{i-4}, w_{i-4})\right] \quad (5.19)$$

$i = 4, 5, ..., N-1$

Para los implícitos se debe usar $\left(t_{i+1}, f(t_{i+1}, y(t_{i+1}))\right)$ como un nodo de interpolación extra aproximar la integral $\int_{t_i}^{t_{i+1}} f(t, y(t))dt$.

Se presentan algunos de ellos (Adams-Moulton)

5.3.4.2. Métodos de Adams-Moulton

A. De dos pasos

$$w_0 = \alpha \qquad w_1 = \alpha_1$$

$$w_{i+1} = w_i + \frac{h}{12}\left[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\right]$$

$$i = 1, 2, ..., N-1$$

$$\tau_{i+1}(h) = -\frac{1}{24}y^{(4)}(\mu_i)h^3 \qquad con\ t_{i-1} \langle \mu_i \langle t_{i+1}$$

B. De tres pasos

$$w_0 = \alpha \qquad w_1 = \alpha_1 \qquad w_2 = \alpha_2 \tag{5.20}$$

$$\tau_{i+1}(h) = -\frac{19}{720}y^{(5)}(\mu_i)h^4 \qquad con\ t_{i-2} \langle \mu_i \langle t_{i+1}$$

C. De cuatro pasos

$$w_0 = \alpha \qquad w_1 = \alpha_1 \qquad w_2 = \alpha_2 \qquad w_3 = \alpha_3$$

$$w_{i+1} = w_i + \frac{h}{720}\left[251f(t_{i+1}, w_{i+1}) + 646f(t_i, w_i) - 264f(t_{i-1}, w_{i-1}) + 106f(t_{i-2}, w_{i-2}) - 19f(t_{i-3}, w_{i-3})\right]$$

$$i = 3, 4, ..., N-1$$

$$\tau_{i+1}(h) = -\frac{3}{160}y^{(6)}(\mu_i)h^5 \qquad con\ t_{i-3} \langle \mu_i \langle t_{i+1} \tag{5.21}$$

Los métodos implícitos por lo general son más estables con menores errores de redondeo, pero presentan la dificultad de requerir una manipulación algebraica para obtener una expresión explícita de w_{i+1}.

De allí que su uso se dirige para mejorar aproximaciones obtenidas por las técnicas explicitas, combinación que configuran las técnicas denominadas de predicción-corrección, empleando el explícito para predecir la aproximación y la implícita para corregirla.

Otros métodos multipasos se generan por integración o interpolación de polinomios en intervalos $\left[t_j, t_{i+1}\right]$ con $j \le i-1$ parar aproximar a $y(t_{i+1})$.

Así está el método de Milne, entre $\left[t_{i-3}, t_{i+1}\right]$:

Milne $$w_{i+1} = w_{i-3} + \frac{4h}{3}\left[2f(t_i, w_i) - f(t_{i-1}, w_{i-1}) + 2f(t_{i-2}, w_{i-2})\right] \tag{5.22}$$

Como dupla de predicción y corrección (Simpson), con un error de truncado proporcional a $y^{(5)}$ y h^4.

Es menos empleado que los de Adams-Basforth-Moulton por sus inconvenientes de estabilidad.

5.3.5. USO DE LA EXTRAPOLACIÓN

Su empleo en el P.V.I. está ligado al método de la media

$w_{i+1} = w_{i-1} + 2hf(t_i, w_i) \qquad i \geq 1 .$ (5.23)

Para hallar w_2 se necesita $w_0 \, y \, w_1$, se emplea la condición inicial $w_0 = \alpha = y(a)$ y para w_1 se utiliza la técnica de Euler. De allí en más se emplea 5.25

Al llegar al valor *t*, se corrige el punto extremo que comprende las dos aproximaciones del punto medio, generando una aproximación $w(t,h)$ a $y(t)$ dada por.

$$y(t) = w(t,h) + \sum_{k=1}^{\infty} \gamma_{kh}{}^{2k}$$

Con γ_k ctes., ligadas a las derivadas de $y(t)$, pero es independiente de *h.*

5.3.6. ESTUDIO DEL ERROR

Los algoritmos de Runge-Kutta, Adams y Milne tienen un error de truncado dependiente de la derivada quinta de $y(t)$, o sea tienen equivalencias en la exactitud con polinomios de Taylor de cuarto grado.

Para cualquier método se desea siempre que haya convergencia a la solución exacta de la ecuación diferencial. Por la serie de Taylor, si las $f(t_{i+1}, y_{i+1})$ a partir del polinomio de Taylor basado en t_i, es decir que el polinomio cambie en cada paso, la solución aproximada se puede acercar tanto como se quiera a la solución exacta, seleccionando los t_i cercanos entre sí.

Las técnicas de Runge-Kutta son convergentes bajo requisitos parecidos al de Taylor, los predoctores-correctores requieren que $f(t,y)$ sean lipschitzianos.

Respecto a la estabilidad: una técnica es relativamente estable si un error que se produzca en la aplicación $y' = Ay$ se reproduzca en la solución exacta.

Los de Taylor y Adams presentan estabilidad relativa, no así el de Milne que para $A\langle 0$ los errores individuales crecen exponencialmente mientras la solución exacta desciende.

- Para ecuaciones de un paso:

El método es consistente si $\lim_{h\to 0} \max_{1\leq i\leq n} |\tau_i(h)| = 0$

$\tau_i(h)$ error de truncado en el paso i.

El método es convergente si $\lim_{h\to 0} \max_{1\leq i\leq n} |w_{ji} - y(t_i)| = 0$

Con $y(t_i)$, el valor exacto de la ecuación diferencial.

Es decir, la consistencia se dará cuando $h \to 0$ pues $\tau_i(h) \to 0$, en el caso de convergencia ocurre algo parecido pues $w_i \to y(t_i)$ cuando $h \to 0$.

Englobando, para una P.V.I. dado por:

$y' = f(t,y) \qquad a \leq t \leq b \qquad y(a) = \alpha$

Que se aproxima por $w_0 = \alpha$

$$w_{i+1} = w_{i-1} + h\theta(t_i, w_i, h)$$

Con un número $h_0 \rangle 0$, $\theta(t_i, w_i, h)$ continua y lipschitziana para ternas (t,w,h), con $a \leq t \leq b$, $-\infty\langle w\langle\infty, 0 \leq h \leq h_0$.

La técnica será:

i. Estable

ii. Convergente $\Leftrightarrow$ es consistente, o sea si $\theta(t,w,o) = f(t,y) \qquad \forall t \in [a,b]$

iii. Si para $i = 1,2,...,N$

$$|\tau_i(h)| \le \tau(h) \quad \text{para } 0 \le h \le h_0$$

Se tendrá $|y(t_i) - w_i| \le \frac{\tau(h)}{L} e^{L(t_i - a)}$ L=cte. de Lipschitz

Así Euler tiene una convergencia de $O(h^2)$

Parar los métodos multipasos presentados, en un P.V.I, como

$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, ..., w_{m-1} = \alpha_{m-1}$

$$w_{i+1} = c_{m-1}w_i + c_{m-2}w_{i-1} + ... + c_0 w_{i+1-m} + hF[t_i h, w_{i+1}, w_i, ..., w_{i+1-m})] \qquad (5.24)$$

Para $i = m-1, m, ..., N-1$ con $c_0, c_1, ..., c_{m+1}$ ctes.

$t_i = \alpha + ih$

5.24 posee un polinomio característico vinculado, que se expresa por:

$$P(\gamma) = \gamma^m - c_{m-1}\gamma^{m-1} - c_{m-2}\gamma^{m-2} - ... - c\gamma - c_0 \qquad (5.25)$$

Las raíces de $P(\gamma)$ quedarán ligados a la estabilidad respecto del error de redondeo.

Sean $\gamma_1, \gamma_2, ..., \gamma_m$ las raíces, incluso no todas distintas, de la ecuación de $P(\gamma)$=0 asociado a 5.25.

Si $|\gamma_i| \le 1$ para $i = 1,2,...,m$ y todas las raíces con valor absoluto 1, son raíces simples, el método satisface la condición de raíz.

Se infiere

a) Las técnicas que satisfacen la condición de raíz y $\gamma = 1$ es la única raíz de la ecuación característica de magnitud uno, son fuertemente estable.
b) Si satisfacen la condición de raíz y con más de una raíz distinta de 1, son débilmente estables.
c) Las técnicas que no satisfacen la condición de raíz son inestables.

En las técnicas multipasos, convergencia y consistencia están estrechamente vinculadas a la estabilidad del redondeo, lo que se puede sintetizar a través del lema:

Para una técnica

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, ..., w_{m-1} = \alpha_{m-1}$$

Y

$$w_{i+1} = c_{m-1}w_i + c_{m-2}w_{i-1} + ... + c_0 w_{i+1-m} + hF[t_i h, w_{i+1}, w_i, ..., w_{i+1-m})]$$

Se dará consistencia siempre y cuando se verifica la condición de raíz. Por otro lado, si el método de diferencias es consistente con la ecuación diferencial, implican que el método será estable si es convergente.

La resolución de las ecuaciones diferenciales parciales (en este caso PVI) encuentran un amplio espacio de cobertura en soft especializados. Se brinda aquí una muy breve descripción desde Matlab y su Toolbox específico

5.4. BREVE DESCRIPCIÓN Y USO DEL SOLVER DE EDO DE MATLAB

Las denominadas *ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb* resuelven problemas de valor inicial para EDO.

Sintaxis

[t,Y] = solver(odefun,tspan,y0)
[t,Y] = solver(odefun,tspan,y0,options)
[t,Y,TE,YE,IE] = solver(odefun,tspan,y0,options)
sol = solver(odefun,[t0 tf],y0...)

odefun: función que evalúa el lado derecho de la ecuación diferencial; todos los solvers resuelven sistemas de ecuaciones de la forma *y'=f(t,y)* o problemas que involucran una matriz de masa *M(t,y)y'=f(t,y*) La 23s solo con *M=cte*, la *15s* y *23t* con una *M* singular, caso ecuaciones diferenciales.

tspan vector que especifica el intervalo de integración *[t0,tf*]. El solver impone las condiciones iniciales en *tspan*(1) e integra de *tspan*(1) a *tspan*(end), si se desea a tiempos particulares se usa *tspan = [t0,t1,...,tf]*.

Si *tspan* tiene dos elementos *[t0 tf*], el solver devuelve la solución evaluados en cada etapa de integración, si tiene más de dos elementos, la solución evaluada a los tiempos dados, ubicados en orden creciente o decreciente.

y0 vector de condiciones iniciales

options estructura de parámetros iniciales que cambia las propiedades de integración por default, con *odeset* se generan estas opciones

Los argumentos de salida serán:

t vector de columna de los tiempos

y arreglo de la solución, cada fila es la solución al tiempo correspondiente

solver	Tipo de problema	Orden de exactitud	Cuando usar
Ode45	No stiff(rigido)	medio	Es el más común
Ode 23	No stiff	bajo	Para tol pobres y en stiff moderados
Ode 113	No stiff	Bajo a alto	Para tol estrictas
Ode15s	stiff	Bajo a moderado	Reemplaza a 45 para stiff
Ode23s	stiff	bajo	Para M cte y tol pobres en sistemas
Ode23t	moderado stiff	bajo	Para stiffs moderados
Ode23tb	stiff	bajo	Para sistemas stiffs con tol pobre

Estructura *options*

parámetros	Ode45	Ode23	Ode113	Ode15s	Ode23s	Ode23t	Ode23tb
RelTol, AbsTol, NormControl	xx	xx	xx	xx	xx	xx	xx
OutputFcn, OutputSel, Refine, Stats	xx	xx	xx	xx	xx	xx	xx
Events	xx	xx	xx	xx	xx	xx	xx
Jacobian, JPattern, Vectorized	xx	xx	xx	xx	xx	xx	xx
Mass MStateDependence MvPattern MassSingular	-- xx --- ---	---- xx --- ---	--- xx --- ---	xx xx --- xx	xx ---- ---- ----	Xx Xx Xx xx	x xx xx ----
InitialSlope	---	---	---	xx	---	xx	---
MaxOrder, BDF	---	---	---	xx	---	---	--
MaxStep, InitialStep	xx	xx	xx	xx	xx	xx	xx

Ejemplos

Considerando el movimiento de un cuerpo rígido sin fuerzas externas, dado por el sistema no rígido de primer orden

$y_1' = y_2 y_3,\ y_1(0) = 0$

$y_2' = -y_1 y_3,\ y_2(0) = 1$

$y_3' = -0.51 y_1 y_2,\ y_3(0) = 1$

Para similar este sistema, se crea una función rigid que contiene las ecuaciones

function dy = rigid(t,y)

dy = zeros(3,1); % a column vector

*dy(1) = y(2) * y(3);*

*dy(2) = -y(1) * y(3);*

*dy(3) = -0.51 * y(1) * y(2)*;

mediante *odeset* se modifica la tolerancia y se resuelve para un intervalo de tiempo *[0 12]* con un vector de condición inicial *[0 1 1]* al tiempo *0*

options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);

[t,Y] = ode45(@rigid,[0 12],[0 1 1],options);

Graficando las columnas del arreglo retornado y-t muestran la solución

t =

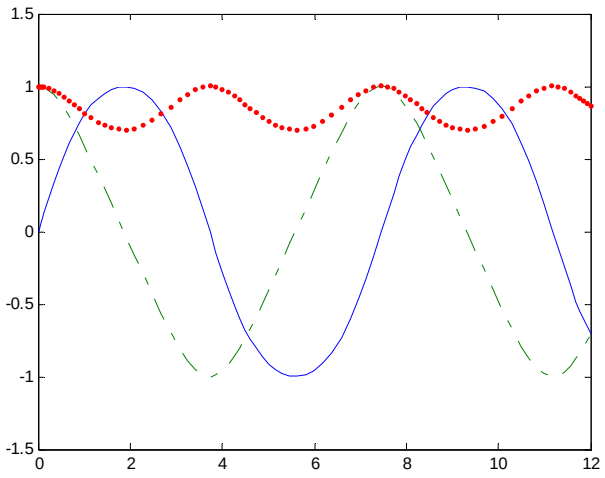
```
       0
  0.0317
  0.0634
  0.0951
  0.1268
  0.2356
  .......
  .
 11.9237
 12.0000
```

y=

```
       0    1.0000    1.0000
  0.0317    0.9995    0.9997
  0.0633    0.9980    0.9990
  0.0949    0.9955    0.9977
  0.1263    0.9920    0.9959
  0.2324    0.9726    0.9861
  ......    ......    ......
 -0.6570   -0.7542    0.8833
 -0.7058   -0.7087    0.8639
```

Graficando las columnas del arreglo retornado y-t muestran la solución

>>plot(t,Y(:,1),'-',t,Y(:,2),'-.',t,Y(:,3),'.')



Algoritmos

Ode45 se basa en Runge-Kutta explícita, el par Dormand-Prince. Es un solver de una etapa, en calcular *y(tn)*, usando solo *y(tn-1)*..

En general, *ode45* es la que más se aplica como "primera aproximación".

La *ode23* es un Runge-Kutta explícita, par de Bogacki – Shampine, preferible a la *ode45* para tolerancia gruesa y con rigidez moderada; al igual que la *ode45*, es de un solver de una etapa.

La *ode113* es un solver Adams-Bashforth-Moulton (predictor-corrector) de orden variable, puede ser más eficiente que la *ode45* a tolerancias estrictas y cuando la función del archivo ODE es costosa para su evaluación.

La *ode113* es un solver multietapasos.

La *ode15s* es un solver de orden variable basado en fórmulas de diferenciación numérica, opcionalmente usa el método de Gear, menos eficiente.

Al igual que las *ode113*, *ode15s* es un solver multietapas. Puede intentarse la *ode15s* cuando la *ode45* falla es muy ineficiente o se sospecha que el problema es rígido o cuando se resuelve un problema algebraico diferencial

La *ode23s* se basa en una fórmula de orden dos de Rosenbrock, de una etapa, puede funcionar mejor que las *15s* para tolerancias gruesas, resolviendo ciertos problemas de rigidez donde las *15s* no sean efectivas

La *ode23t* es una adecuación de la regla trapezoidal con un interpolante libre, aplicable a problemas solo de rigidez moderada, puede resolver ecuaciones diferenciales algebraicas, implementa un Runge Kutta implícito, de una primera etapa trapezoidal y una segunda de fórmulas de diferenciación regresivas de orden dos.

5.5. EJERCITACIÓN DE UNIDAD 5 EMPLEANDO MATLAB

I) Dada la ecuación diferencial en PVI

$y' = 3t+3y$

$y(0) = 1$, en [0,1]

sn exacta: y(t) =4/3 e^{3t} t .

i) por Heun:

i) crear el archivof1.m:

```
function z = f(t,y)
z = 3*y+3*t;
```

ii) correr el método

```
>>H1 = heun(@f1,0,1,1,5)   % se uso M=5
H1 =
         0    1.0000
    0.2000    1.8400
    0.4000    3.4912
    0.6000    6.5863
    0.8000   12.2517
    1.0000   22.4920
```

Para M=10

```
H1 = heun(@f1,0,1,1,10)
H1 =
         0    1.0000
    0.1000    1.3600
    0.2000    1.8787
    0.3000    2.6109
    0.4000    3.6301
    0.5000    5.0355
    0.6000    6.9602
    0.7000    9.5835
    0.8000   13.1463
    0.9000   17.9728
    1.0000   24.4989
```

ii) Empleando Runge Kutta(orden 4)

```
>>rk4(@f1,0,1,1,10)
ans =
         0    1.0000
    0.1000    1.3664
    0.2000    1.8961
    0.3000    2.6460
    0.4000    3.6932
    0.5000    5.1418
    0.6000    7.1321
    0.7000    9.8537
    0.8000   13.5624
    0.9000   18.6035
    1.0000   25.4432
```

iii) Uso de métodos de predictor-corrector

Sea la ecuación y'=0.5*(t-y) con y(0)=1 en [0,3], se plantean los métodos de Adams Basforth-Moulton(4 pasos);Milne- Simpson (3 pasos), Hamming(3 pasos); todos

requieren el cálculo previo de coordenadas T e Y (por ej con Runge Kutta de cuarto orden).
Seleccionando un paso de 0.25

```
>> rk4(@f1,0,3,1,12)
ans =
                0   1.00000000000000
   0.25000000000000   0.89749145507813
   0.50000000000000   0.83640366823723
   0.75000000000000   0.81186958242375
   1.00000000000000   0.81959403365079
   1.25000000000000   0.85578655193387
   1.50000000000000   0.91710205830810
   1.75000000000000   1.00058853011477
   2.00000000000000   1.10364081576586
   2.25000000000000   1.22395987640518
   2.50000000000000   1.35951681679056
   2.75000000000000   1.50852114264965
   3.00000000000000   1.66939274788702
```

Desde la ventana de comandos

```
>> T=zeros(1,13);
>> Y=zeros(1,13);
>> T=0:1/4:3;
>> Y(1:4)=[1 0.8975 0.8364 0.8119];
>>
abm(@f1,T,Y)

ans =

         0    1.0000
    0.2500    0.8975
    0.5000    0.8364
    0.7500    0.8119
    1.0000    0.8196
    1.2500    0.8558
    1.5000    0.9171
    1.7500    1.0006
    2.0000    1.1036
    2.2500    1.2240
    2.5000    1.3595
    2.7500    1.5085
    3.0000    1.6694




>>hamming(@f1,T,Y)
ans =

         0    1.0000
    0.2500    0.8975
    0.5000    0.8364
```

```
    0.7500    0.8119
    1.0000    0.8196
    1.2500    0.8558
    1.5000    0.9171
    1.7500    1.0006
    2.0000    1.1037
    2.2500    1.2240
    2.5000    1.3595
    2.7500    1.5085
    3.0000    1.6694

>>milne(@f1,T,Y)

ans =
         0    1.0000
    0.2500    0.8975
    0.5000    0.8364
    0.7500    0.8119
    1.0000    0.8196
    1.2500    0.8558
    1.5000    0.9171
    1.7500    1.0006
    2.0000    1.1036
    2.2500    1.2240
    2.5000    1.3595
    2.7500    1.5085
    3.0000    1.6694
```

EJERCICIOS PROPUESTOS PARA UNIDAD 5 –P.V.I.

- Dados los problemas de valor inicial

a) $y' = te^{t} - 2y$, *en* $0 \leq t \leq 1$ *con* $y(0) =$, $h = 0.2$ *Sn exacta:* $0.2te^{3t} - 0.04(e^{3t} - e^{-2t})$

b) $y' = 1 + (t-y)^2$, *en* $0 \leq t \leq 3$ *con* $y(2) = 1$, $h = 0.2$; *sn exacta:* $t + (1/1-t)$

c) $y' = 1 + y/t$, *en* $1 \leq t \leq 2$ *con* $y(1) = 2$, $h = 0.2$; *sn exacta:* $tlnt + 2t$

d) $y' = (y/t) - (y/t)^2$ *en* $1 \leq t \leq 2$ *con* $y(1) = -1$, $h = 0.1$; *sn exacta:* $1/1 + lnt$

e) $y' = 1 + (y/t) + (y/t)^2$ *en* $1 \leq t \leq 3$ *con* $y(1) = 0$, $h = 0.2$, *sn exacta:* $\tan(lnt)$

resolverlos empleando las técnicas de

i)Heun

ii) Runge Kutta

iii)Runge Kutta Fehlberg

iv) ABM

v)Milne-Simpson

vi) Hamming

UNIDAD N° 6: SISTEMAS LINEALES

Los sistemas de ecuaciones lineales se presentan en variados problemas científicos y tecnológicos e incluso en la vinculación de la matemática a disciplinas sociales y económicas.

Planteado generalmente como

$$\begin{aligned}
&R_1 \quad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&R_2 \quad a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
&\vdots \\
&\vdots \\
&R_n \quad a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n
\end{aligned} \qquad (6.1)$$

Sistemas de nxn, n ecuaciones con $x_1, x_2, \ldots, x_n$ con las a_{ij} que para cada *i,j=1,2,…,n* y las b_i con $i = 1, 2, \ldots, n$.

Se presentan primeramente consideraciones del Algebra lineal.

6.1. OPERACIONES PERMITIDAS DE REDUCCIÓN

a) El renglón R_i puede multiplicarse por una constante $\lambda \neq 0$, empleando λR_i en vez de R_i:
$(\lambda R_i \to R_i)$.

b) R_i multiplicado por λ y sumado a R_i puede investigarse en lugar de $R_i : (R_i + \lambda R_i \to R_i)$

c) Se pueden intercambiar R_i *con* $R_j : (R_i \leftrightarrow R_j)$

El empleo de estas operaciones puede originar un sistema más sencillo con igual conjunto solución.

6.2. REPRESENTACIÓN DE UN SISTEMA LINEAL. SUSTITUCIÓN HACIA ATRÁS

Una matriz de *n* por *(n+1)* puede representar un sistema lineal.

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
&\vdots \\
&\vdots \\
&a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n
\end{aligned} \qquad (6.2)$$

Con

$$A = \begin{pmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} \cdots & a_{nn} \end{pmatrix} \qquad \overline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \overline{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Matriz de coeficientes Matriz columna Matriz de independientes

Considerando la matriz ampliada $\left[A,\bar{b}\right]$, se aplican las operaciones permitidas descriptas en 6.1, transformándose en un sistema lineal y obtener las soluciones para $x_1, x_2, ..., x_n$, generándose una matriz triangular.

$$\bar{A} = \begin{bmatrix} a_{11} & & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & \ddots & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & & & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & a_{nn} & a_{n,n+1} \end{bmatrix} \quad \text{con} \quad a_{i,n+1} = b_i \quad \text{para}$$

$c/i = 1, 2, ..., n$

Permitiendo la sustitución hacia atrás:

Así $x_n = \dfrac{a_{n,n+1}}{a_{nn}}$

Y para el i-ésimo

$$x_i = \frac{a_{n,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}} \tag{6.3}$$

Este método se conoce como eliminación de Gauss con sustitución hacia atrás.
Salta a la vista el gran número de operaciones de producto-división y suma-resta que involucrará un algoritmo para este método, pasando de 17 producto-divisiones y 11 sumas restas para *n=3,*a 44.150 producto divisiones y 42875 sumas-restas para n=50; o sea $\left(\frac{n^3}{3} + n^2 - \frac{n}{3}\right)$producto divisiones y $\left(\frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}\right)$sumas y restas.

6.3. TÉCNICA DE GAUSS-JORDAN

Usala i-esima ecuación para eliminar x_i de $R_{i+1}, R_{i+2}, ..., R_n$ *y de* $R_1, R_2, ..., R_{i-1}$ reduciendo $\left[A,\bar{b}\right]$ a:

$$\begin{matrix} a_{11}^{(1)} & 0 & \cdots & 0 & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & \ddots & \vdots & a_{2,n+1}^{(2)} \\ \vdots & \ddots & & 0 & \vdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} & a_{n,n+1}^{(n)} \end{matrix} \tag{6.4}$$

La solución se obtiene de

$$x_i = \frac{a_{i,n+1}^{(i)}}{a_{ii}^{(i)}} \quad \text{para } c/i = 1, 2, ..., n \tag{6.5}$$

Evitando la sustitución hacia atrás de la eliminación gaussiana.

Requiere $\left(\frac{n^3}{2} + n^2 - \frac{n}{2}\right)$ producto-divisiones y $\left(\frac{n^3}{2} - \frac{n}{2}\right)$ sumas y restas, es decir más operaciones que la sustitución hacia atrás.

6.4. PIVOTEO

Si uno de los elementos pivote $a_{kk}^{(k)}$ es nulo, se hace necesario un intercambio de renglones $(R_k) \leftrightarrow (R_p)$ con p el menor entero mayor que $k \left| a_{pk}^{(k)} \neq 0 \right.$; muchas veces se requiere el intercambio de renglones para bajar el error de redondeo, aún para pivotes $\neq 0$.

Ya que si $a_{kk}^{(k)}$ es $\square$ $a_{jk}^{(k)}$, el factor $m_{j,k}$ $m_{j,k} = \dfrac{a_{j,k}^{(k)}}{a_{k,k}^{(k)}} \square$ 1.

Lo mismo ocurrirá en la sustitución hacia atrás para $x_k = \dfrac{a_{k,n+1}^{(k)} - \sum_{j=k+1}^{n} a_{k,j}^{(k)} x_j}{a_{k,k}^{(k)}}$ pues un error que se tenga en el numerador, al dividirlos por un $a_{k,k}$ pequeño, crecerá significativamente.

Entonces, el pivoteo se efectúa eligiendo el mayor $a_{p,q}^{(k)}$ como pivote y cambiando los renglones *k y p q* más el cambio de las columnas *k y q* si es necesario.

Una pista para seleccionar es tomar el menor $p \geq k \left|\right.$.

$$\left|a_{p,k}^{(k)}\right| = \max_{k \leq i \leq h} \left|a_{i,k}^{(k)}\right| \qquad \text{y efectuar } (R_k) \leftrightarrow (R_p) \qquad (6.6)$$

(técnica parcial de pivoteo)

Otra técnica es la de reescalonamiento de columnas. Se intercambian renglones para colocar ceros en la primera columna, seleccionando el menor *k* que cumpla:

$$\delta_i = \max_{j=1,2,\ldots,n} \left|a_{i,j}\right| \neq 0 \qquad \frac{|ak_1|}{s_k} = \max_{j=1,2,\ldots,n} \frac{\left|a_{i,j}\right|}{s_j}$$

Y se realiza $(R_1) \leftrightarrow (R_k)$, con lo que se asegura que el mayor elemento de cada R_i tiene un valor relativo de 1 antes de efectuar la comparación para el intercambio de renglones.

Ejemplo 6.1:

El escalado implica usar un factor de escala para cada fila, que se define

$$s_k = \max_{1 \leq j \leq n} |a_{kj}|$$

Ahora, antes de eliminar la variable *x i* el intercambio de filas Fila_i <-> Fila_p se hace tomando el primer entero $p \neq i$, de modo que

$$\frac{|a_{pi}|}{s_p} = \max_{i \leq k \leq n} \frac{|a_{ki}|}{s_k}$$

El pivoteo parcial agrega $3/2(n^2-n)$ comparaciones y $(n^2+n)/2$ -1 divisiones, por lo que para lograr una precisión mayor hace falta más poder de cómputo

Sea

$$\begin{aligned} 2.11x_1 - 4.21x_2 + 0.921x_3 &= 2.01 \\ 4.01x_1 + 10.2x_2 - 1.12x_3 &= -3.09 \\ 1.09x_1 + 0.987x_2 + 0.832x_3 &= 4.21 \end{aligned}$$

En su forma aumentada

$$\left(\begin{array}{ccc|c} 2.11 & -4.21 & 0.921 & 2.01 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 1.09 & 0.987 & 0.831 & 4.21 \end{array}\right)$$

$s_1$máximo de la fila $1 = \max\limits_{1 \le j \le n} |a_{1j}| = 4.21$

$s_2$máximo de la fila $2 = \max\limits_{1 \le j \le n} |a_{2j}| = 10.2$

$s_3$máximo de la fila $3 = \max\limits_{1 \le j \le n} |a_{3j}| = 1.09$

Luego queda buscar el pivote, el $\max\limits_{1 \le k \ n} \frac{|a_{k1}|}{s_k}$

$$\frac{|a_{11}|}{s_1} = \frac{2.11}{4.21} = 0.501, \quad \frac{|a_{21}|}{s_2} = \frac{4.01}{10.2} = 0.393 \quad \text{y} \quad \frac{|a_{31}|}{s_3} = \frac{1.09}{1.09} = 1$$

De ahí se toma el mayor que es $\frac{|a_{31}|}{s_3}$ y se hace el intercambio Fila$_1$ <-> Fila$_3$. La matriz pasa a quedar

$$\left(\begin{array}{ccc|c} 1.09 & 0.987 & 0.831 & 4.21 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 2.11 & -4.21 & 0.921 & 2.01 \end{array}\right)$$

$m_{21} = 3.68, m_{31} = 1.94$ son los mutiplicadores, se elimina:

$$\left(\begin{array}{ccc|c} 1.09 & 0.987 & 0.831 & 4.21 \\ 0 & 6.57 & -4.18 & -18.6 \\ 0 & -6.12 & -0.689 & -6.16 \end{array}\right)$$

Reiterando la búsqueda de valores máximos:

$$\frac{|a_{22}|}{s_2} = \frac{6.57}{10.2} = 0.644 < \frac{|a_{32}|}{s_3} = \frac{6.12}{4.21} = 1.45$$

El pivote pasa a ser el de la fila 3, por lo que se intercambian Fila$_2$ <-> Fila$_3$ y queda

$$\left(\begin{array}{ccc|c} 1.09 & 0.987 & 0.831 & 4.21 \\ 0 & -6.12 & -0.689 & -6.16 \\ 0 & 6.57 & -4.18 & -18.6 \end{array}\right)$$

$m_{32} = \frac{6.57}{-6.12} = -1.07$

Elimando, queda:

$$\left(\begin{array}{rrr|r} 1.09 & 0.987 & 0.831 & 4.21 \\ 0 & -6.12 & -0.689 & -6.16 \\ 0 & 0 & -4.92 & -25.2 \end{array}\right)$$

Reemplazando hacia atrás $x = (-0.431\ 0.430\ 5.12)$

6.5. ALGUNAS MATRICES MÁS COMUNES Y SUS OPERACIONES

1. una matriz triangular superior $(nxn): S = \left(s_{i,j}\right)$ tiene para cada $j = 1, 2, ..., n$ los elementos $s_{i,j}$ =0 para cada $i = j+1, j+2, ..., n$.
2. una matriz triangular inferior $I = \left(I_{ij}\right)$ tiene para cada $i = 1, 2, ..., n$ los elementos
   $I_{ij} = 0 \qquad i = 1, 2, ..., j-1$
3. Dadas las matrices $A(nxn), B(nxn), C(kxp), D(mxk)$ y $\lambda \in \mathbf{R}$

Se verifican:

i. $A(BC)=(AB)C$
ii. $A(B+D)=AB+AD$
iii. $U_m B = B \wedge BU_k = B$ U=identidad.
iv. $\lambda(AB) = (\lambda A)B = A(\lambda B)$

4. una matriz A es no singular si existe una matriz $A^{-1}(nxn)$ que verifica $AA^{-1} = A^{-1}A = U$, con A^{-1} la inversa de A.

Si A no posee inversa se denomina Singular
Para la inversa de A (si existe):

Considerando la columna j de $B(nxn)$: $B_j = \begin{bmatrix} b_{1,j} \\ b_{2,j} \\ \vdots \\ b_{n,j} \end{bmatrix}$

Haciendo $AB=C$, la columna j de C vendrá dada por el producto:

$$\begin{bmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \\ c_{n,j} \end{bmatrix} = C_j = AB_j = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} b_{1,j} \\ b_{2,j} \\ \vdots \\ b_{n,j} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n} a_{1k} b_{kj} \\ \sum_{k=1}^{n} a_{2k} b_{kj} \\ \vdots \\ \sum_{k=1}^{n} a_{nk} b_{kj} \end{bmatrix}$$

Bajo la asunción de existencia de A^{-1} y como $A^{-1}=B=(b_{i,j}) \Rightarrow AB = U$, con lo cual

$$AB_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \text{con 1 en el renglón } j$$

Para hallar B se presentarán n sistemas donde la columna j de la inversa es la solución del sistema cuyo segundo miembro es la columna j de U.
Se prefiere arreglar la ampliada más grande

$$[A \vdots U]$$

Realizando la eliminación gaussiana se obtendrá una matriz ampliada del tipo

$$[S \vdots Y]$$

S= triangular superior
Y= obtenida al hacer las mismas operaciones en U que se realizaron para transformar A en S.

Ejemplo 6.2.- sea $A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 1 & 2 \end{bmatrix}$ tomando una matriz B 3x3, cualquiera

$$AB = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} =$$

$$\begin{bmatrix} b_{11}+2b_{21}-b_{31} & b_{12}+2b_{22}-b_{31} & b_{13}+2b_{23}-b_{33} \\ 2b_{11}+b_{21} & 2b_{12}+b_{22} & 2b_{13}+b_{23} \\ -b_{11}+b_{21}+2b_{31} & -b_{12}+b_{22}+2b_{31} & -b_{13}+b_{23}+2b_{33} \end{bmatrix}$$

Suponiendo que $B = A^{-1}$, su producto será U (identidad), de donde:

$$b_{11}+2b_{21}-b_{31}=1 \quad b_{12}+2b_{22}-b_{31}=0 \quad b_{13}+2b_{23}-b_{33}=0$$
$$2b_{11}+b_{21}=0 \quad 2b_{12}+b_{22}=1 \quad 2b_{13}+b_{23}=0$$
$$-b_{11}+b_{21}+2b_{31}=0 \quad -b_{12}+b_{22}+2b_{31}=0 \quad -b_{13}+b_{23}+2b_{33}=1$$

Efectuando la eliminación gaussiana sobre la matriz ampliada más grande, generada por la combinación de las matrices de cada sistema:

$$\left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 2 & 0 & 0 & 1 \end{array}\right]$$

Haciendo $R_2 - 2R_1 \to E_1; R_3 + R_1 \to R_3 \; y \; R_3 + E_2 \to E,$, quedará:

$$\left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & -3 & 2 & -2 & 1 & 0 \\ 0 & 3 & 1 & 1 & 0 & 1 \end{array}\right] \text{ y } \left[\begin{array}{ccc:ccc} 1 & 2 & -1 & 1 & 0 & 0 \\ 0 & -3 & 2 & -2 & 1 & 0 \\ 0 & 0 & 3 & -1 & 1 & 1 \end{array}\right]$$

Con sustitución regresiva en cada matriz ampliada

$$\left[\begin{array}{cccc:c} 1 & 2 & -1 & \vdots & 1 \\ 0 & -3 & 2 & \vdots & -2 \\ 0 & 0 & 3 & \vdots & -1 \end{array}\right], \left[\begin{array}{cccc:c} 1 & 2 & -1 & \vdots & 0 \\ 0 & -3 & 2 & \vdots & 1 \\ 0 & 0 & 3 & \vdots & 1 \end{array}\right], \left[\begin{array}{cccc:c} 1 & 2 & -1 & \vdots & 0 \\ 0 & -3 & 2 & \vdots & 0 \\ 0 & 0 & 3 & \vdots & 1 \end{array}\right]$$

Hallando

$b_{11} = -2/9, b_{21} 4/9, b_{31} = -1/3, b_{12} = 5/9, b_{22} = -1/9, b_{31} = 1/3,$

$b_{13} = -1/9, b_{23} = 2/9, b_{33} = 1/3$

$$A^{-1} = \begin{bmatrix} -2/9 & 5/9 & -1/9 \\ 4/9 & -1/9 & 2/9 \\ -1/3 & 1/3 & 1/3 \end{bmatrix}$$

5. La transpuesta de $A(nxn) = (a_{ij})$ es $A^t(mxn) = (a_{ji})$

Si A es cuadrada, será simétrica si $A^t = A$

Propiedades de la transpuesta

i. $(A^t)^t = A$

ii. $(AB)^t = B^t A^t$

iii. $(A+B)^t = A^t + B^t$

iv. $(A^{-1})^t = (A^t)^{-1}$ siempre que exista A^{-1}

6. Si $A = (a_{ij}), nxn$, triangular o diagonal, su det$all = \prod_{i=1}^{n} a_{ij}$

7. para cualquier $A(nxn)$
   i. $Ax=0$ admite la solución única $\vec{x} = 0$
   ii. $Ax=b$ admite la solución única para cualquier vector columna n-dimensional $\vec{b}$.
   iii. La eliminación gaussiana con intercambio de filas en $Ax=b$ se puede efectuar para cualquier vector columna n-dimensional $\vec{b}$.

6.6. FACTORIZACION MATRICIAL

Si A admite el producto SI (triangular superior por triangular inferior), la factorización es de gran provecho, para la resolución de sistemas lineales.

Se parte del supuesto que en el sistema $A\vec{x} = \vec{b}$ puede practicarse la eliminación gaussiana sin intercambios de filas (no hay pivotes nulos $a_{ii}^{(i)}$ para cada $i = 1, 2, ..., n$)

Para cada $j = 1, 2, ..., n$, se realiza el paso dado por:

$$(P_j - m_{j,1} E_1) \rightarrow (P_j) \qquad \text{con } m_{j,1} = \frac{a_{j,1}^{(1)}}{a_{11}^{(1)}} \qquad (6.7)$$

Para llevar a un sistema con todos los elementos de la primera columna, debajo de la diagonal, nulos.

Esto también puede hacerse multiplicando A por $M^{(1)}$ igual a:

$$M^{(1)}=\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -M^{(1)}{}_{21} & 1 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -M^{(1)}{}_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix}$$ primera matriz de Gauss

Representando

$M^{(1)}A\vec{x}=A^{(2)}\vec{x}=M^{(1)}\vec{b}=\vec{b}^{(2)}$

Luego se forma $M^{(2)}$, la matriz identidad con los elementos debajo de la diagonal de la 2° columna reemplazados por los negativos de $m_{j,2}=\dfrac{a_{j,2}^2}{a_{22}^2}$

$M^{(2)}$ por $A^{(2)}$ tiene ceros debajo de la diagonal en la 1° y 2° columna:

$A^{(3)}\vec{x}=M^{(2)}A^{(2)}\vec{x}=M^{(2)}M^{(1)}A\vec{x}=M^{(2)}M^{(1)}\vec{b}=\vec{b}^{(3)}$

Así $A^{(k)}\vec{x}=\vec{b}^{(k)}$ se deberá multiplicar por la k-matriz de Gauss.

$$M^{(k)}=\begin{bmatrix} 1 & & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ & \ddots & & \ddots & & & & \vdots \\ 0 & & \ddots & & \ddots & & & \vdots \\ \vdots & \ddots & & \ddots & & \ddots & & \vdots \\ \vdots & & 0 & & \ddots & & \ddots & \vdots \\ \vdots & & \vdots & & & \ddots & & \ddots \\ \vdots & & \vdots & -m_{k+1,k} & 0 & & \ddots & 0 \\ \vdots & & \vdots & & & \ddots & & \ddots \\ 0 & \cdots & 0 & -m_{n,k} & 0 & \cdots & 0 & 1 \end{bmatrix}$$

$$A^{(k+1)}\vec{x}=M^{(k)}A^{(k)}\vec{x}=M^{(k)}\vec{b}^{(k)}=\vec{b}^{(k+1)}=M^{(k)}...M^{(1)}\vec{b} \qquad (6.9)$$

Terminando con la formación de $A^{(n)}\vec{x}=\vec{b}^{(n)}$ con

$$A^{(n)}=\begin{bmatrix} a_{11}^{(1)} & & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \ddots & a_{22}^{(2)} & \ddots & \vdots \\ \vdots & \ddots & & \ddots & a_{n-1,n}^{(n-1)} \\ 0 & \cdots & 0 & & a_{n,n}^{(n)} \end{bmatrix}$$ triangular superior

Dados por $A^{(n)}=M^{(n-1)}M^{(n-2)}...M^{(1)}A$

Hasta acá se efectúo una media factorización de $A=IS$ (S la triangular superior de $A^{(n)}$).

Para hallar I, se usa de nuevo $A^{(k)}\vec{x}=\vec{b}^{(k)}$ efectuada por $M^{(k)}$.

$A^{(k-1)}\vec{x}=M^{(k)}A^{(k)}\vec{x}=M^{(k)}\vec{b}^{(k)}=\vec{b}^{(k-1)}$ (6.10)

Con $M^{(k)}$ a través de $\left(R_j-m_{j,k}R_k\right)\rightarrow\left(R_j\right)$ $\qquad j=k+1,...,n$

Como se desea invertir el proceso, para llegar a $A^{(k)}$ los $\left(R_j+m_{j,k}R_k\right)\rightarrow\left(R_j\right)$ se realizaron para cada _ $j=k+1,...,n$, lo que significa hacer el producto $\left[M^{(k)}\right]^{-1}$.

$$I^{(k)} = \left[M^{(k)} \right]^{-1} == \begin{bmatrix} 1 & & 0 & \cdots & & \cdots & \cdots & \cdots & 0 \\ & \ddots & & \ddots & & & & & \vdots \\ 0 & & \ddots & & \ddots & & & & \vdots \\ \vdots & \ddots & & \ddots & & \ddots & & & \vdots \\ \vdots & & 0 & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & & \ddots & & \ddots & \vdots \\ \vdots & & \vdots & -m_{k+1,k} & 0 & & \ddots & & 0 \\ \vdots & & \vdots & & & \ddots & & \ddots & \\ 0 & \cdots & 0 & -m_{n,k} & 0 & \cdots & 0 & & 1 \end{bmatrix} \qquad (6.11)$$

Pero *I,* en el factoreo de *A,* representa el producto de los $I^{(k)}$.

$$I = I^{(1)} = I^{(2)} = ... = I^{(n+1)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n1} & \cdots & m_{n,n-1} & 1 \end{bmatrix}$$

Ya que $IS = M^{(n-1)}...M^{(2)}M^{(1)}A$ dará

$$\begin{aligned} IS &= I^{(1)}I^{(2)}...I^{(n-2)}I^{(n-1)}M^{(n-1)}M^{(n-2)}...M^{(2)}M^{(1)}A \\ &= I^{(1)}I^{(2)}...I^{(n-2)}UM^{(n-2)}M^{(n-3)}...M^{(2)}M^{(1)}A \\ &= I^{(1)}I^{(2)}...I^{(n-2)}M^{(n-2)}M^{(n-3)}...M^{(2)}M^{(1)}A \\ &= I^{(1)}I^{(2)}...I^{(n-3)}UM^{(n-3)}...M^{(2)}M^{(1)}A \end{aligned}$$

A partir de esta deducción se infiere el lema:

Si la eliminación gaussiana para $A\vec{x} = \vec{b}$ se efectúa sin cambios de filas, *A* podrá factorizarse como el producto *IS.*

$$A = IS$$

con

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

$$\mathrm{S} = \begin{bmatrix} a_{11}^{(1)} & & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \ddots & a_{22}^{(2)} & \ddots & \vdots \\ \vdots & \ddots & & \ddots & a_{n-1,n}^{(n-1)} \\ 0 & \cdots & 0 & & a_{n,n}^{(n)} \end{bmatrix}$$

Una ves que se factorizó *A*, $\vec{x}$ se obtendrá en dos etapas: primero se hace $\vec{y} = S\vec{x}$ resolviendo el sistema $I\vec{Y} = \vec{b}$ para hallar *y*; como *I* es triangular no es muy complicado. Hallado $\vec{y}$, el sistema triangular $\vec{y} = S\vec{x}$ se resuelve para encontrar $\vec{x}$ por sustitución hacia atrás.

Según determinadas variaciones para la técnica se conocen los métodos:

- Doolittle: requiere que $i_{11} = i_{22} = ... = i_{nn} = 1$
- Crout: requiere que los elementos de la diagonal principal de *S* sean todos 1.
- Choleski: requiere $i_{ii} = s_{ii}$ para cada *i.*

Es válido remarcar que este tipo de factorización presentado es ventajoso cuando no se requiere el intercambio de filas para ajustar el error de redondeo.
Para reordenar o permutar renglones de una determinada matriz es útil la matriz de permutación.
Llamada *P(nxn)* tiene un elemento igual a 1 en cada columna y en cada fila, el resto son todos ceros.

Para una (3x3), P será = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ que al multiplicar por *A(3x3)* por izquierda intercambia el segundo y tercer renglón de *A*, y por derecha, se cambien la segunda y tercera columna de *A*.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{13} & a_{12} \\ a_{21} & a_{23} & a_{22} \\ a_{31} & a_{33} & a_{32} \end{bmatrix}$$

- Si *P* es matriz de permutación tiene inversa y $P^{-1} = P^t$.
- Para cualquier *A*, no singular, habrá una *P* tal que $PA\vec{x} = P\vec{b}$ se resuelve sin intercambiar filas, con lo que *PA* se puede factorizar como *PA=IS*

Y al ser $P^{-1} = P^t \Rightarrow A = \left(P^t I\right) S$

$\left(P^b I\right)$ no será triangular inferior (salvo que *P=U*).

6.7. MATRICES CARACTERISTICAS

a. La *A(nxn)* es diagonalmente estricta dominante si

$$|a_{ii}| \rangle \sum_{i=1}^{n} |a_{ij}| \qquad (6.12)$$

b. Una matriz diagonalmente estricta dominante es no singular, con lo que la eliminación gaussiana es factible para $A\vec{x} = \vec{b}$ sin intercambios de filas o columnas, presentando estabilidad frente a los errores de redondeo.

c. La matriz *A* es definida positiva si es simétrica y si $\vec{x}^t A\vec{x} \rangle 0$ para cualquier vector columna n-dimensional $\vec{x} \neq 0$.

No es el criterio más sencillo para verificar la positividad de la matriz.

Se presentan dos lemas de utilidad

$L_2 = para\ A(nxn)$ definida positiva

i. *A* es no singular

ii. $\max_{1 \le k,j \le n} |a_{kj}| \le \max_{1 \le i \le n} |a_{ii}|$ (6.13)

iii. $a_{ii} \rangle 0$ para cada $i = 1, 2, ..., n$

iv. $(a_{ii})^2 \langle a_{ii} a_{ij} \quad con \quad i \neq j$

$L_3 = A$ simétrica será definida positiva siempre y cuando pueda realizarse la eliminación gaussiana sin intercambio de renglones para $A\vec{x} = \vec{b}$ con los pivotes todos positivos.

De L_3 se desprenden consecuencias como.

a) A es definida positiva si y solo sí A puede factorizarse como IDI^t, con *I* con $a_{ii} = 1$ y *D* diagonal con $a_{ii} \rangle 0$.

b) A es definida positiva si y solo sí A puede factorizarse como II^t, con los a_{ii} de I no nulos (Método de Choleski).

c) Si *A(nxn)* es simétrica con eliminación gaussiana sin intercambio de filas, A puede factorizarse en IDI^t, I posee $a_{ii}=1$ y D es la matriz con $a_{11}^{(1)},...,a_{nn}^{(n)}$ en su diagonal.

II^t de Choleski para una definida positiva requiere menos operaciones que IDI^t pero implica hallar n raíces cuadradas.

d) Matrices de banda.

Una matriz *(nxn)* es de banda si se encuentran p y q (enteros) con $p\rangle 1, q\langle n$ tal que $a_{ij}=0$ siempre que $i+p\le j$ o $j+q\le i$, definiéndose el ancho de banda como $r=p+q-1$.

Estas matrices dispondrán sus elementos no nulos en derredor de la diagonal.

Las mas comunes, $p=q=2$, son tridiagonales y los de $p=q=4$, pentadiagonales.

Las primeras, asociadas a aproximaciones de splines cúbicos o aproximaciones lineales por parte en problemas de valores de contorno, los pentadiagonales de utilidad para problemas de valores de frontera.

En forma general

$$A=\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & & \ddots & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & 0 & a_{n.n-1} & \ddots & a_{nn} \end{bmatrix} \qquad (6.14)$$

El trabajar con matrices de banda facilita los algoritmos de factorización (por el n° de operaciones) ya sea de Doolittle o de Crout.

Ejemplo 6.3-

Encontrar la factorización LU de la matriz

$$A=\begin{pmatrix} 2 & 3 & 0 & 1 \\ 4 & 5 & 3 & 3 \\ -2 & -6 & 7 & 7 \\ 8 & 9 & 5 & 21 \end{pmatrix}$$

Primero se convierte en cero todos los elementos debajo del primer elemento diagonal de A. Para esto, se suma (-2) veces la primera fila de A a la segunda fila de A. Luego sumar la primera fila de A a la tercera fila de A, y por último se suma (-4) veces la primera fila de A a la cuarta fila de A, obteniendo la siguiente matriz

$$U_1=\begin{pmatrix} 2 & 3 & 0 & 1 \\ 0 & -1 & 3 & 1 \\ 0 & -3 & 7 & 8 \\ 0 & -3 & 5 & 17 \end{pmatrix}$$

Mientras tanto, comenzar la construcción de una matriz triangular inferior, L_1,con unos en la diagonal principal. Para ello, colocar los opuestos de los multiplicadores utilizados en las operaciones de fila en la primera columna de L_1 debajo del primer elemento diagonal de L_1, obteniendo

$$L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & ? & 1 & 0 \\ 4 & ? & ? & 1 \end{pmatrix}$$

Ahora sumar (-3) veces la segunda fila de U_1 a la tercera fila de U_1 y (-1) veces la tercera fila de U_1 a la cuarta fila de U_1. Colocando los opuestos de los multiplicadores debajo del segundo elemento diagonal deL_1, obteniendo

$$U_2 = \begin{pmatrix} 2 & 3 & 0 & 1 \\ 0 & -1 & 3 & 1 \\ 0 & 0 & -2 & 5 \\ 0 & 0 & -2 & 8 \end{pmatrix} \text{ y } L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 3 & 1 & 0 \\ 4 & 1 & ? & 1 \end{pmatrix}$$

finalmente sumar (-1) veces la tercera fila de U_2 a la cuarta fila de U_2. Luego colocar el opuesto de este multiplicador debajo del tercer elemento diagonal de L_2, obteniendo

$$U_3 = \begin{pmatrix} 2 & 3 & 0 & 1 \\ 0 & -1 & 3 & 1 \\ 0 & 0 & -2 & 5 \\ 0 & 0 & 0 & 4 \end{pmatrix} \text{ y } L_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 3 & 1 & 0 \\ 4 & 1 & 1 & 1 \end{pmatrix}$$

Las matrices U_3 y L_3 componen una factorización LU para la matriz A

Ejemplo 6.4- Usar el método de Gauss-Jordan para calcular la matriz inversa de:

$$A = \begin{pmatrix} 2 & -4 & 0 \\ -0.5 & 1.2 & -0.1 \\ -0.3125 & -0.625 & 3.125 \end{pmatrix}$$

Se ve que el primer elemento pivote $a_{11}=2$ está bien colocado y se procede a hacer ceros debajo de este elemento, multiplicando el renglón *1* por $\frac{0.5}{2}$ y lo sumar al renglón *2* y multiplicar el mismo renglón *1* por $\frac{0.3125}{2}$ sumado al renglón *3*.

$$\begin{pmatrix} 2 & -4 & 0 & 1 & 0 & 0 \\ 0 & 0.2 & -0.1 & 0.25 & 1 & 0 \\ 0 & -1.25 & 3.125 & 0.15625 & 0 & 1 \end{pmatrix}$$

Para el segundo pívot se escoge el elemento mayor (con valor absoluto) entre $a_{22}=0.2$ y $a_{32}=-1.25$, entonces se intercambia el renglón *2* y el renglón *3*, quedando:

$$\begin{pmatrix} 2 & -4 & 0 & 1 & 0 & 0 \\ 0 & -1.25 & 3.125 & 0.15625 & 0 & 1 \\ 0 & 0.2 & -0.1 & 0.25 & 1 & 0 \end{pmatrix}$$

Se hacen ceros arriba y abajo del segundo elemento pívot, multiplicando el renglón *2* por $-\frac{4}{1.25}$ y sumado el renglón *1*, como multiplicar el mismo renglón *2* por $\frac{0.2}{1.25}$ y sumado al renglón *3*, quedando:

$$\begin{pmatrix} 2 & 0 & -10 & 0.5 & 0 & -3.2 \\ 0 & -1.25 & 3.125 & 0.15625 & 0 & 1 \\ 0 & 0 & 0.4 & 0.275 & 1 & 0.16 \end{pmatrix}$$

el tercer elemento pivot es $a_{33}=0.4$, para hacer ceros arriba de este elemento, multiplicar el renglón *3* por $-\frac{3.125}{0.4}$ y sumado renglón *2*, como multiplicar el mismo renglón *3* por $\frac{10}{0.4}$ y sumado al renglón *1*, quedando:

$$\begin{pmatrix} 2 & 0 & 0 & 7.375 & 25 & 0.8 \\ 0 & -1.25 & 0 & -1.9921875 & -7.8125 & -0.25 \\ 0 & 0 & 0.4 & 0.275 & 1 & 0.16 \end{pmatrix}$$

Por último se hacen unos en la diagonal principal, multiplicando el renglón *1, 2* y *3* por $\frac{1}{2}$, $-\frac{1}{1.25}$ y $\frac{1}{0.4}$, respectivamente, resultando la matriz

$$\begin{pmatrix} 1 & 0 & 0 & 3.6875 & 12.5 & 0.4 \\ 0 & 1 & 0 & 1.59375 & 6.25 & 0.2 \\ 0 & 0 & 1 & 0.6875 & 2.5 & 0.4 \end{pmatrix}$$

la matriz inversa de *A* es:

$$A^{-1} = \begin{pmatrix} 3.6875 & 12.5 & 0.4 \\ 1.59375 & 6.25 & 0.2 \\ 0.6875 & 2.5 & 0.4 \end{pmatrix}$$

6.8. NORMA DE VECTORES Y MATRICES

6.8.1. Norma Matricial

Como los vectores en $\mathbf{R}^n$ son vectores columnas, se usará la simbología de la transpuesta para representarlo en sus componentes

$$\vec{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \text{se notará } \vec{X} = (x_1, x_2, ..., x_n)^t$$

Las normas para $\vec{X} = (x_1, x_2, ..., x_n)^t$ se definen por

$$\|X\| = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2} \qquad \wedge \qquad \|X\| = \max_{1 \le i \le n} |x_i| \tag{6.15}$$

La primera (norma euclideana) da el concepto clásico de distancia al origen para $X \in R, R^2, o, R^3$.

Como la norma de un vector proporciona una medida de la distancia de un vector cualquiera y el vector origen, la distancia entre dos vectores puede asimilarse a la norma de la diferencia de ellos.

Así, para $\overline{X} = (x_1, x_2, ..., x_n)^t$ e $\overline{Y} = (y_1, y_2, ..., y_n)^t \in R^n$

Las distancias entre $\overline{X}$ e $\overline{Y}$ se definen:

$$\left\|\bar{X}-\bar{Y}\right\| = \left\{\sum_{i=1}^{n}(x_i-y_i)^2\right\}^{1/2} \qquad (6.16)$$

$$\left\|\bar{X}-\bar{Y}\right\| = \max_{1\le i\le n}\left|x_i-y_i\right|$$

Con ello se puede definir el límite de una sucesión de vectores en R^n:

- Una sucesión $\left\{\bar{X}^{(k)}\right\}_{k=1}^{\infty}$ de vectores de R^n converge en $\bar{X}$ respecto a la norma $\|\ \|$, si $\forall\varepsilon\rangle 0.$, existe un entero $z(\varepsilon)$ tal que $\left\|\bar{X}^{(k)}-\bar{X}\right\|\langle\varepsilon$ para $\forall k \ge z(\varepsilon)$.
- Una sucesión $\left\{\bar{X}^{(k)}\right\}$ converge a $\bar{X}$ en R^n respecto a la norma $\|\ \|$ si para $\lim_{k\to\infty} X_i^{(k)} = x_i, \qquad i = 1,2,...,n$
- Se define en el conjunto de las matrices *(nxn)*como una función de reales que verifica para *A,B(nxn)* y cualquier $\beta \in R$:
  - i. $\|A\|\rangle 0$
  - ii. $\|A\| = 0 \Leftrightarrow A = 0$
  - iii. $\|\alpha A\| = |\alpha|\|A\|$
  - iv. $\|A+B\| \le \|A\| + \|B\|$
  - v. $\|AB\| \le \|A\|\|B\|$

Se considerarán solo las normas $\|\ \|_e$ y $\|\ \|$ (vectoriales).

Si $\|\ \|$ es norma vectorial en R^n se dará

$$\|A\| = \max_{\|\bar{x}\|=1}\left|A\vec{x}\right|$$

Norma matricial natural

Para $A = \left(a_{ij}\right), nxn,$ se tiene

$$\|A\| = \max_{1\le i\le n}\sum_{j=1}^{n}\left|a_{ij}\right|$$

6.9. VALORES Y VECTORES PROPIOS

6.9.1. Polinomio propio

Si *A* es matriz cuadrada, el polinomio propio de *A* será:

$\rho(\lambda) = \det(A-\lambda U)$ U= matriz identidad (6.17)

Si λ es raíz de ρ, el sistema lineal $(A-\lambda U)\vec{x} = \vec{0}$ tiene soluciones diferentes a las triviales.

Si ρ es el polinomio propio de *A*, las raíces de ρ se llaman valores propios de la matriz *A*.

Si λ es el valor propio de *A*, con $\vec{x} \ne \vec{0}$, y se hace $(A-\lambda U)\vec{x} = \vec{0}$, $\vec{x}$ es el vector propio de *A* correspondiente a λ.

Al ser $\bar{X}$ el vector propio asociado a λ, $A\bar{X} = \lambda\bar{X}$, lo que significa que *A,* transforma $\bar{X}$ en un múltiplo de $\bar{X}$.

Si λ (real)$\rangle 1$, *A* alarga en λ a $\bar{X}$; si $0\langle\lambda\langle 1$, *A* contrae $\bar{X}$ en un factor λ e idénticamente para $-1\langle\lambda\langle 0$.

6.9.2. Radio espectral de una mMatriz

Se define $\rho(A)$ como

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

Se vincula con la norma de una matriz mediante

a) $\left[\rho(AA^t)\right]^{1/2} = \|A\|_e$

b) $\rho(A) \le \|A\|$ para toda norma natural.

6.9.3. Convergencia de una matriz

Se plantean enunciados equivalentes

a) A es matriz convergente

b) $\lim_{n\to\infty} \|A^n\| = 0$ para alguna norma natural

c) $\lim_{n\to\infty} \|A^n\| = 0$ para todas las normas naturales (6.18)

d) $\rho(A)\langle 1$

e) $\lim_{n\to\infty} A^n\vec{x} = \vec{0}$ para cualquier $\vec{x}$

6.10. TÉCNICAS DE REPETICIÓN PARA SISTEMAS LINEALES

Básicamente consisten, para una sistema $A\vec{x} = \vec{b}$, partir de un vector de arranque $\vec{x}$ aproximación de la solución $\vec{x}$ para ir configurando una $\left\{\vec{x}^{(k)}\right\}_0^\infty$ convergente a $\vec{x}$.

Transformar el sistema original en uno equivalente $\vec{x} = W\vec{x} + \vec{c}$ para cierta matriz W y vector $\vec{c}$.

La sucesión se va calculando como $\vec{x}^{(k)} = W\vec{x}^{(k-1)} + \vec{c}$ $\quad k = 1,2,3...,$ (6.19)

Empleadas para sistemas lineales grandes, con uso en el cálculo numérico de problemas de borde y ecuaciones en derivadas parciales.

6.10.1. Técnica de Jacobi

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

Se despeja $\vec{x}_i$ (de la ecuación i-esima), para $a_{ii} \neq 0$.

$$x_i = \sum_{\substack{j=1 \\ j\neq 1}}^{n} \left(\frac{-a_{ij}x_j}{a_{ii}} \right) + \frac{b_{ii}}{a_{ii}} \qquad i = 1,2,...,n \qquad (6.20)$$

Generando para cada $x_i^{(k)}$, a partir de $x_i^{(k-1)}$, con $k \ge 1$, como

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j\neq 1}}^{n} \left(a_{ij}x_j^{(k-1)} \right) + b_i}{a_{ii}} \qquad i = 1,2,...,n \qquad (6.21)$$

La técnica se expresa como $\vec{x}^{(k)} = W\vec{x}^{(k-1)} + \vec{c}$, desglosando A en su parte diagonal y la no diagonal.

Es decir, sea D la matriz diagonal coincidente con la diagonal de A, -I la triangular inferior de A y –S la triangular superior de A:

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}=\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix}-\begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}--\begin{bmatrix} 0 & a_{ii} & \cdots & a_{ii} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{ii} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

(D) (I) (s)

$A=D-I-S \Rightarrow A\vec{x}=\vec{b}=(D-I-S)\vec{x}=\vec{b}$

$D\vec{x}=(I+S)\vec{x}+\vec{b} \Rightarrow D^{-1}(I+S)\vec{x}+D^{-1}\vec{b}$ (6.22)

Matricialmente

$\vec{x}^{(k)}=D^{-1}(i+s)\vec{x}^{(k-1)}+D^{-1}\vec{b} \qquad k=1,2,3...$

Para detener el proceso se puede emplear:

$$\frac{\left\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\right\|}{\left\|\vec{x}^{(k)}\right\|}\langle\varepsilon \qquad (6.23)$$

Usando la norma natural.

6.10.2. Técnica de Gauss-Seidel

Si para el calculo de los $\vec{x}_i^{(k)}$, en ves de usar $\vec{x}^{(k-1)}$ y como ya se obtuvieron $\vec{x}_1^{(k)},...,\vec{x}_{i-1}^{(k)}$ de 6.21, representando aproximaciones superiores a la solución $\vec{x}_1,...,\vec{x}_{i-1}$, seria prudente calcular $\vec{x}_i^{(k)}$ con los valores más próximos a través de

$$\vec{x}_i^{(k)}=\frac{-\sum_{j=1}^{i-1}\left(a_{ij}\vec{x}_j^{(k)}\right)-\sum_{j=1}^{i-1}\left(a_{ij}\vec{x}_j^{(k-1)}\right)+b_i}{a_{ii}} \qquad i=1,2,...,n \qquad (6.24)$$

Constituyendo la técnica Gauss-Seidel

Matricialmente se expresa por

$(D-I)^{-1}S\vec{X}^{(k-1)}+(D-I)^{-1}\vec{b} \qquad k=1,2,...$ (6.25)

$(D-I)$ debe ser no singular o sea $a_{ii}\neq 0$ para $\not{c}\, i=1,2,...,n$

6.10.3. Convergencia de las técnicas

Lema 1: para cualquier $\vec{x}_0$ de R^n, la sucesión $\left\{\vec{x}^{(k)}\right\}_0^\infty$ de

$\vec{x}^{(k)}=W\vec{x}^{(k-1)}+\vec{c} \qquad k\geq 1, \vec{c}\neq 0$ (6.26)

Converge a $\vec{x}=W\vec{x}+\vec{c} \Leftrightarrow \rho(W)\langle 1$

Lema 2(Expresa condiciones de suficiencia para convergencia de las dos técnicas vistas)

Si en $A\vec{x}=\vec{b}$, A es estrictamente dominante, para cualquier $\vec{x}^{(0)}$ las sucesiones $\left\{\vec{x}^{(k)}\right\}_0^\infty$ convergen a la solución única de $A\vec{x}=\vec{b}$.

Siempre se buscará una técnica que posea una matriz asociada con radio espectral mínimo, expresado por el teorema de Stein-Rosenberg.

- Si $a_{ij}\leq 0$ para cada $i\neq j$ y $a_{ii}\rangle 0$ para cada $i=1,2,...,n$, se cumplirá una y solo una de las proposiciones:
  i. $0\leq\rho(W_g)\langle\rho(W_j)\langle 1$
  ii. $1\langle\rho(W_j)\langle\rho(W_g)$ (6.27)

iii. $\rho(W_j) = \rho(W_g) = 0$

iv. $\rho(W_j) = \rho(W_g) = 1$

v.

6.11. TÉCNICAS DE RELAJACIÓN

Se llamará vector resto, para un $\vec{x} \in R^n$ que aproxima el sistema lineal $A\vec{x} = \vec{b}$, el definido por $\vec{r} = \vec{b} - A\vec{x}$

En los métodos vistos habrá un

$\vec{r}_i^{(k)} = \left(\vec{r}_{1i}^{(k)}, \vec{r}_{2i}^{(k)}, ..., \vec{r}_{ni}^{(k)}\right)^t$

Que se busca converja a cero.

En el caso de la técnica G-S, se buscará seleccionar $\vec{x}_i^{(k)}$ tal que:

$$\vec{x}_i^{(k)} = \vec{x}_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}} \tag{6.28}$$

Considerando ahora el vector residuo $r_{i+1}^{(k)}$ asociado al vector

$$\vec{x}_{i+1}^{(k)} = \left(\vec{x}_1^{(k)}, ..., \vec{x}_i^{(k)}, \vec{x}_{i+1}^{(k-1)}, ..., \vec{x}_n^{(k-1)}\right)^t \tag{6.29}$$

Para la componente *i* de $r_{i+1}^{(k)}$ se tiene

$$\begin{aligned} r_{i,i+1}^{(k)} &= b_i - \sum_{j=1}^{i}\left(a_{ij}\vec{x}_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}\vec{x}_j^{(k-1)}\right) \\ &= b_i - \sum_{j=1}^{i}\left(a_{ij}\vec{x}_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}\vec{x}_j^{(k-1)}\right) - a_{ii}\vec{x}_i^{(k)} \end{aligned} \tag{6.30}$$

La técnica G-S requiere que sea nula la componente $i = \vec{r}_{i+1}$, pero este proceso no es el más sencillo en la búsqueda de decrecer la norma $\vec{r}_{i+1}^{(k)}$.

Por ello, se presenta 6.28 como

$$\vec{x}_i^{(k)} = \vec{x}_i^{(k-1)} + q\frac{r_{ii}^{(k)}}{a_{ii}} \quad \text{métodos de relajación} \tag{6.31}$$

- Si $0\langle q\langle 1$, técnicas de subrelajación, para sistemas no convergentes por G-S.
- $q\rangle 1$, para acelerar convergencia de sistemas convergentes por G-S (métodos SOR o sobrerelajación)

Para el cálculo se usará:

$$\vec{x}_i^{(k)} = (1-q)\vec{x}_i^{(k-1)} + \frac{q}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1}\left(a_{ij}\vec{x}_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}\vec{x}_j^{(k-1)}\right)\right] \tag{6.32}$$

Matricialmente

$$\vec{x}^{(k)} = (D - qI)^{-1}\left[(1-q)D + qS\right]\vec{x}^{(k-1)} + q(D - qI)^{-1}\vec{b} \tag{6.33}$$

Ejemplo 6.5-

Aplicar el método de Jacobi para resolver el sistema:

$10x_1 - x_2 + 2x_3 = 6$

$-x_1 + 11x_2 - x_3 + 3x_4 = 25$

$2x_1 - x_2 + 10x_3 - x_4 = -11$

$3x_2 - x_3 + 8x_4 = 15$

cuya solución única es $x = (1, 2, -1, 1)^t$, a partir de la aproximación inicial $x(0) = (0, 0, 0, 0)^t$ y con una tolerancia $TOL = 5 \times 10^{-4}$.

Las ecuaciones del proceso iterativo son

$$x_1^{k+1} = \frac{1}{10}x_2^k - \frac{1}{5}x_3^k + \frac{3}{5}$$

$$x_2^{k+1} = \frac{1}{11}x_1^k + \frac{1}{11}x_3^k - \frac{3}{11}x_4^k + \frac{25}{11}$$

$$x_3^{k+1} = -\frac{1}{5}x_1^k + \frac{1}{10}x_2^k + \frac{1}{10}x_4^k - \frac{11}{10}$$

$$x_4^{k+1} = -\frac{3}{8}x_2^k + \frac{1}{8}x_3^k + \frac{15}{8}$$

Cuyos resultados se dan en la tabla

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0	2.2727	1.7159	2.0530	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
$x_4^{(k)}$	0	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	.9998

en la iteración se tiene:
$||x_{(10)} - x_{(9)}||/||x(10)|| = 0.327 \times 10-4 < TOL$

Empleando el método de Gauss-Seidel

$$x_1^{k+1} = \frac{1}{10}x_2^k - \frac{1}{5}x_3^k + \frac{3}{5}$$

$$x_2^{k+1} = \frac{1}{11}x_1^{k+1} + \frac{1}{11}x_3^k - \frac{3}{11}x_4^k + \frac{25}{11}$$

$$x_3^{k+1} = -\frac{1}{5}x_1^{k+1} + \frac{1}{10}x_2^{k+1} + \frac{1}{10}x_4^k - \frac{11}{10}$$

$$x_4^{k+1} = -\frac{3}{8}x_2^{k+1} + \frac{1}{8}x_3^{k+1} + \frac{15}{8}$$

Generándose la tabla

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0.6000	1.0302	1.0066	1.0009	1.0001
$x_2^{(k)}$	0	2.3273	2.0369	2.0036	2.0003	2.000
$x_3^{(k)}$	0	-0.9873	-1.0145	-1.0025	-1.0003	-1.000
$x_4^{(k)}$	0	0.8789	0.9843	0.9984	0.9998	1.000

como $||x_{(5)} - x_{(4)}||/|x_{(5)}|| = 2.09 \times 10-4 < TOL$, solamente se efectuaron cinco iteraciones

6.11.1. Condicionamiento de una matriz

Si $\tilde{x}$ aproxima la solución de $A\vec{x} = \vec{b}$, A es no singular y $\vec{r}$ es el vector resto de $\vec{x}$, parar cualquier norma natural.

$$\|\vec{x} - \tilde{x}\| \le \|\vec{r}\|\|A^{-1}\|$$

Con $\frac{\|\vec{x}-\tilde{x}\|}{\|\vec{x}\|} \le \|A\|\|A^{-1}\|\left\|\frac{\vec{r}}{\vec{b}}\right\|$ $\vec{x} \neq \vec{0}$ y $\vec{b} \neq \vec{0}$ (6.34)

Para una matriz no singular se define la condición relativa a la norma natural $\|\ \|$ como $C(A) = \|A\|\|A^{-1}\|$

Entonces el error relativo se representará por

$$\frac{\|\vec{x}-\tilde{x}\|}{\|\vec{x}\|} \le C(A)\left\|\frac{\vec{r}}{\vec{b}}\right\| \quad (6.35)$$

Dándose para cualquier matriz A no singular y norma natural $\|\ \|$

$\|I\| = \|AA^{-1}\| \le \|A\|\|A^{-1}\| = C(A)$

Si $C(A)$ esta próximo a 1 esta bien condicionada

Si $C(A)$ es bastante mayor que 1, esta mal condicionada.

Si se establece la estimación $\tilde{\vec{y}} \approx \vec{x} - \tilde{x}$ con $\tilde{\vec{y}}$ solución aproximada del sistema $A\vec{y} = \vec{r}$, se tendrá que $\tilde{\vec{x}} + \tilde{\vec{y}}$ precisará más la aproximación a la solución $A\vec{x} = \vec{b}$ que lo hiciera $\tilde{\vec{x}}$, constituyendo el método de refinamiento, o sea efectuar iteraciones en el sistema cuyo miembro derecho es el vector resto para las siguientes aproximaciones.

6.12. GRADIENTE CONJUGADO

El método del gradiente conjugado es un algoritmo para la solución numérica de sistemas particulares de ecuaciones lineales, principalmente aquellos cuya matriz es simétrica y definida positiva, y el cual tiene buena recepción pues aprovecha muy bien la estructura de la matriz, además de tener muy buenas propiedades de estabilidad numérica. Es un método iterativo, de manera que puede ser aplicado a sistemas dispersos (sparse) que sean excesivamente grandes como para ser resueltos mediante métodos directos como la descomposición

Tales sistemas surgen regularmente cuando se hallan soluciones numéricas a ecuaciones diferenciales parciales. El método del gradiente conjugado puede ser también utilizado para resolver problemas de optimización no restringida. Variantes de este m_etodo son los del Gradiente Biconjugado, los cuales proporcionan una generalización a matrices no simétricas. Los métodos de gradiente conjugado no lineal buscan el mínimo a ecuaciones no lineales.

6.12.1. CGM como método directo

Suponiendo que se desea resolver el siguiente sistema de ecuaciones lineales:

$$Ax = B \quad (6.36)$$

donde la matriz A de n x n es simétrica (es decir, $A^T = A$), definida positiva ($\mathbf{x}^T A\mathbf{x} > 0$), para todos los vectores $\mathbf{x}$ en R^n (diferentes de cero), y *real*. Se denotará la solución única a este sistema como $\mathbf{x}^*$

Se dice que dos vectores no nulos (diferentes de cero) **u** y **v** son conjugados (con respecto a A) si $\mathbf{u}^T A\mathbf{v} = 0$ (6.37)

Dado que A es simétrica y definida positiva, el lado izquierdo de la anterior relación define un producto interno:

$\langle \mathbf{u},\mathbf{v} \rangle_A := \langle A\mathbf{u},\mathbf{v} \rangle = \langle \mathbf{u},A\mathbf{v} \rangle = \mathbf{u}^T A\mathbf{v}$ (6.38)

Estos dos vectores son conjugados si son ortogonales con respecto a este producto interno. El ser conjugado es una relación simétrica: si **u** es conjugado a v, entonces **v** es conjugado a **u.** Suponiendo que $\{p_k\}$ es una secuencia de n direcciones mutuamente conjugadas.

Entonces la forma p_k forma una base en R^n, de manera que la solución **x*** de $Ax = B$ se puede expandir en esta base:

$\mathbf{x^*} = \alpha_1 p_1 + \ldots + \alpha_n p_n$ (6.39)

En este caso los coeficientes están dados por:

$$\alpha_k = \frac{p_k^T b}{p_k^T A p_k} = \frac{< p_k, b >}{< p_k, p_k >_A} = \frac{< p_k, b >}{\|p_k\|_A^2} \quad (6.40)$$

Este resultado es tal vez más explicativo claro al considerar el producto interno definido anteriormente, obteniéndose un método para resolver el sistema $Ax = B$: se encuentra una secuencia de n direcciones conjugadas, y luego se calculan los coeficientes α_k.

6.12.2. CGM como un método iterativo

El método del gradiente conjugado es también un método iterativo que a partir de un valor inicial va calculando sucesivos valores que se van acercando a la solución exacta del sistema lineal. El valor $k+1$ será la solución, si la diferencia entre él y el anterior k es menor que un cierto valor de tolerancia. Si se eligen los vectores conjugados p_k adecuadamente, pueden no necesitarse todos ellos para obtener una muy buena aproximación a la solución **x***, lo cul permite resolver sistemas donde n es tan grande que el método directo implicaría mucho tiempo.

Se denota una solución de prueba inicial para **x** como $\mathbf{x}^0$. Se puede asumir, sin pérrdida de generalidad que $\mathbf{x}^0 = 0$ (equivalente a considerar al sistema $Az = b\text{-}Ax_0$). Note que la solución **x*** es también el único minimizador de:

$$f(x) = \frac{1}{2}\vec{x}^T Ax - b^T \vec{x},\ \vec{x} \varepsilon R^n \quad (6.41)$$

Esto sugiere tomar como primer vector base p_1 al gradiente de f en $x = x_0$, el cual es igual a $-b$. Los otros vectores en la base serán los conjugados al gradiente Si se denota al residuo en la *k-esima* iteración como r_k

$r_k = b\text{-}Ax_k$ (6.42)

se puede ver que este término r_k es el negativo del gradiente de f en $x = x_k$, con lo que se puede mostrar, ya que las direcciones p_k son conjugadas entre sí, que una expresión para tales direcciones es:

$$p_{k+1} = r_k - \frac{p_k^T A r_k}{p_k^T A p_k} p_k \quad (6.43)$$

relación a partir de la cual se hallan los coeficientes, mediante (6.40), para así encontrar la solución ***x****, tal y como se plantea en (6.39), para el sistema lineal.

6.13. LAS FUNCIONES DE MATLAB PARA ALGEBRA LINEAL

Se listan las funciones de MATLAB para algebra lineal
cond número de condición respecto a la inversión
condeig número de condición respecto a los eigenvalues
det Determinante
norm norma de vector o matriz
normest Estima 2-norma de la matriz
null espacio nulo
orth ortogonalización
rank rango de una matriz

rcond estima el número de condición recíproco de la matriz
rref forma escalonda reducida por filas
subspace ángulo entre dos subespacios
trace suma de los elemntos de la diagonal
chol factorización de Cholesky
cholinc factorización incomplete de Cholesky
condest estima el número de condición de1-norma
funm función matriz general
inv matriz inversa
linsolve resuelve un sistema lineal de ecuaciones
lscov solución cuadrados mínimos con covarianza conocida
lsqnonneg mínimos cuadrados no negativos
lu factorización LU de una matriz
luinc factorización incompletaLU
pinv pseudo inversa Moore-Penrose de una matriz
qr descomposición triangular- ortogonal
balance mejora exactitud de los eigenvalues calculados
cdf2rdf Convierte forma diagonal compleja a forma diagonal en bloques real
eig autovalores y autovectores
eigs autovalores y autovectores de matriz dispersa
gsvd descomposición de valor singular generalizada
hess forma de Hessenberg de una matriz
poly polinomio con raíces específicas
polyeig polinomio del problema de autovalor
qz factorización QZ para autovalores generalizados
*rsf2csf c*onvierte forma real de Schur a la forma compleja de Schur
schur descomposición de Schur
svd descomposición de valor singular
svds vectores y valores singulares de matriz dispersa
expm exponencial de una matriz
logm logaritmo de una matriz
*sqrtm m*atrix raíz cuadrada
planerot plano rotación de Givens
qrdelete borra columna o fila de la factorización QR
qrinsert Inserta columna o fila en la factorización QR
qz factorización QZ para autovalores generalizados

6.14. EJERCITACIÓN UNIDAD 6 CON MATLAB

I) Dada una matriz A, efectuar el proceso de scalonamiento y reducción,
sea A= [4 3 1; 3 4 -1;0 -1 4]
>>rrefstep(A)
We start by determining an Upper triangular form.
<><><><> Upper Triangular Form STEP-BY-STEP <><><><>
A- Your original matrix is:

```
   4    3    0
   3    4   -1
   0   -1    4
```

Press enter to begin ROW REDUCTION

Making the pivot in row 1 equal to 1
in matrix

```
   4.0000    3.0000         0
        0    1.0000   -0.5714
        0         0    1.0000
```

```
  1. PERFORM this step.                    <3> Turn on rational display.
  2. EXPLAIN this step then PERFORM it.    <4> Turn off rational display.

  0. Quit
     Your choice ==>  1
```

……..hasta
We obtain:

```
     1.0000    0.7500         0
          0    1.0000   -0.5714
          0         0    1.0000
```

II) Obtner la inversa de A

```
>> invert(A)
ans =
    0.6250   -0.5000   -0.1250
   -0.5000    0.6667    0.1667
   -0.1250    0.1667    0.2917
```

III) Generar un conjunto de vectores linealmente independientes, dada una matriz A
Para hallar un subconjunto de vectores linealmente independientes, si se ingresa ‘r’ como segundo argumento los vectores son las columnas de, si se ingresa ‘c’ las filas de A.

```
>> lisub(A,'r')

ans =

     4     3     0
     3     4    -1
     0    -1     4
```

IV)Hallar la solución general de un sistema homogéneo de ecuaciones, devolviendo un conjunto de vectores de la base del espacio nulo de Ax = 0.

```
>> homsoln(A)

ans =

     0
     0
     0
```

V) Encontrar el menor de la matriz A, fila 2 columna 1.

```
>> Cminor(A,2,1)
ans =
    12
```

VI) Realizar la eliminación gaussiana de la matriz A
>> Gelim(A)

Rational numbers? y/n: y
Count of operations? y/n: y
All steps? y/n: y

initial matrix
```
     4      3      0
     3      4     -1
     0     -1      4
```

[press Enter at each step to continue]

normalize
```
     1    3/4      0
     3      4     -1
     0     -1      4
```
multiplications:
```
   2
```
---
create zero
```
     1    3/4      0
     0    7/4     -1
     0     -1      4
```
additions, multiplications:
```
   2   2
```
---
normalize
```
     1    3/4      0
     0      1   -4/7
     0     -1      4
```
multiplications:
```
   1
```
---
create zero
```
     1    3/4      0
     0      1   -4/7
     0      0   24/7
```
additions, multiplications:
```
   1   1
```
---
normalize
```
     1    3/4      0
     0      1   -4/7
     0      0      1
```
multiplications:
```
   0
```

-echelon form-

```
Total additions, multiplications, element-swaps:
    3     6     0
--------------------
create zero
       1           3/4           0
       0             1          -4/7
       0             0           1
additions, multiplications:
    0     0
--------------------
create zero
       1           3/4           0
       0             1           0
       0             0           1
additions, multiplications:
    0     0
--------------------
create zero
       1             0           0
       0             1           0
       0             0           1
additions, multiplications:
    0     0
--------------------
-reduced echelon form-
       1             0           0
       0             1           0
       0             0           1
Total additions, multiplications, element-swaps:
    3     6     0
--------------------
```

VII) Hallar el determinante de la matriz A empleando Gauss Jordan

```
>> Gjdet(A)

Rational numbers? y/n: y
All steps? y/n: y

initial matrix
       4             3           0
       3             4          -1
       0            -1           4
[press Enter at each step to continue]

create zero
       4             3           0
       0           7/4          -1
       0            -1           4
--------------------
```

```
create zero
      4          3          0
      0         7/4        -1
      0          0         24/7
--------------------
-final reduced form-
      4          3          0
      0         7/4        -1
      0          0         24/7

number of row swaps
    0

determinant =
     24
```

VIII) Reducir A empleando la eliminación de Gauss Jordan

```
>> Gjelim(A)

Rational numbers? y/n: y
Count of operations? y/n: y
All steps? y/n: y

initial matrix
      4          3          0
      3          4         -1
      0         -1          4
[press Enter at each step to continue]

normalize
      1         3/4         0
      3          4         -1
      0         -1          4
multiplications:
    2
--------------------
create zero
      1         3/4         0
      0         7/4        -1
      0         -1          4
additions, multiplications:
    2     2
--------------------
normalize
      1         3/4         0
      0          1         -4/7
      0         -1          4
multiplications:
    1
--------------------
```

```
create zero
      1           0          3/7
      0           1         -4/7
      0          -1           4
additions, multiplications:
    1     1
--------------------
create zero
      1           0          3/7
      0           1         -4/7
      0           0         24/7
additions, multiplications:
    1     1
--------------------
normalize
      1           0          3/7
      0           1         -4/7
      0           0           1
multiplications:
    0
--------------------
create zero
      1           0           0
      0           1         -4/7
      0           0           1
additions, multiplications:
    0     0
--------------------
create zero
      1           0           0
      0           1           0
      0           0           1
additions, multiplications:
    0     0
--------------------
-reduced echelon form-
      1           0           0
      0           1           0
      0           0           1
Total additions, multiplications, element-swaps:
    4     7     0
--------------------
```

IX) Dada P=[4 3 0;3 4 -1;0 -1 4]; graficar las líneas en la eliminación de Gauss-Jordan, para sistema de dos variables, la matriz debe tener tres columnas, es aumentada.

```
>> P=[4 3 0;3 4 -1;0 -1 4];
>> Gjpic(P)
```

```
    4     3     0
    3     4    -1
    0    -1     4
--------------------
   1.0000    0.7500         0
   3.0000    4.0000   -1.0000
        0   -1.0000    4.0000
--------------------
   1.0000    0.7500         0
        0    1.7500   -1.0000
        0   -1.0000    4.0000
--------------------
   1.0000    0.7500         0
        0    1.0000   -0.5714
        0   -1.0000    4.0000
--------------------
   1.0000         0    0.4286
        0    1.0000   -0.5714
        0         0    3.4286
--------------------
   1.0000         0    0.4286
        0    1.0000   -0.5714
        0         0    1.0000
--------------------
    1     0     0
    0     1     0
    0     0     1
--------------------
   -reduced echelon form-
```

X) Encontrar la la matriz adjunta de A

```
>> Adj(A)

ans =

   15   -12    -3
  -12    16     4
   -3     4     7
```

XI) Resolver un sistema triangular (cuadrada), dada la matriz de coefcientes A y el vector independiente B

```
>>A=[4 -1 2 3;0 -2 7 -4;0 0 6 5;0 0 0 3];
>>B=[20;-7;4;6];
>> backsub(A,B)
ans =
     3
    -4
    -1
     2
```

XII) Resolver un sistema triangular (cuadrada), dada la matriz de coefcientes A y el vector independiente B, empleando triangularización superior más sustitución hacia atrás

```
>>A=[1 2 2 4;2 0 4 3;4 2 2 1;-3 1 3 2]  % no singular
>> B=[13 28 20 6]';
>> uptrbk(A,B)
ans =
    3.0000
   -1.3333
    5.0000
    0.6667
```

XIII) Resolver un sistema triangular (cuadrada), dada la matriz de coefcientes A y el vector independiente B,con factorización con pivoteo

```
>>A=[1 2  4 1;2 8 6 4;3 10 8 8;4 12 10 6];
>>B=[21 52 79 82];
>> lufact(A,B)
ans =
    3.0000
   -1.3333
    5.0000
    0.6667
```

XIV) Resolución por técnicas iterativas, dada la matriz de coeficientes y el vector independiente

```
i) >> A=[1 -5 -1;4 1 -1;2 -1 -6];
>> B=[-8 13 -2]';
>> P=[0 0 0]';
>> jacobi(A,B,P,0.001, 10)
ans =
  1.0e+006 *
    9.1711
    5.8774
   -0.8071
>>jacobi(A,B,P,0.001, 20)
ans =
  1.0e+013 *
   -3.0646
   -1.4564
    0.2279

>> gseid(A,B,P,0.001,10)
ans =
  1.0e+013 *
    0.3856
   -1.5624
    0.3889

>> gseid(A,B,P,0.001,18)
ans =
  1.0e+023 *
    0.7280
   -2.9500
```

0.7343

XV) Utilizar el método del Gradiente conjugado(directo no es tan bueno como elim. Gauss con pivoteo, sí para iterativos)

```
>> A=[4 3 0;3 4 -1;0 -1 4]; % debe ser simétrica
>> b=[24 30 -24]';
>> conjgrad(A,b,0.001)
ans =
    3.0000
    4.0000
   -5.0000
```

Para asegurar el condicionamiento

Gradiente conjugado precondicionado

pcgnull

EJERCICIOS PROPUESTOS PARA UNIDAD 6

6.1. Dados los sistemas

a) $\begin{aligned} x_1 - x_2 + 3x_3 &= 2 \\ 3x_1 - 3x_2 + x_3 &= -1 \\ x_1 + x_2 + &= 3 \end{aligned}$ b) $\begin{aligned} 2x_1 - 1.5x_2 + 3x_3 &= 1 \\ -x_1 + 2x_3 &= 3 \\ 4x_1 - 4.5x_2 + 5x_3 &= 1 \end{aligned}$

c) $\begin{aligned} 2x_1 - &= 3 \\ x_1 + 1.5x_2 + &= 4.5 \\ -3x_2 + 0.5x_3 &= -6.6 \\ 2x_1 - 2x_2 + 3x_3 + x_4 &= 0.8 \end{aligned}$

Resolverlos empleando i) la sustitución hacia atrás,ii) triangularización superior más sustitución hacia atrás iii) factorizándolo LU con pivoteo

6.2- Halle las inversas de las siguientes matrices

a) $\begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 2 & -4 & -2 \\ 2 & 1 & 1 & 5 \\ -1 & 0 & -2 & -4 \end{bmatrix}$ b) $\begin{bmatrix} 2 & 0 & 1 & 2 \\ 1 & 1 & 0 & 2 \\ 2 & -1 & 3 & 1 \\ 3 & -1 & 4 & 3 \end{bmatrix}$ por Gauss Jordan

6.3- factorice en la forma LU las matrices

a) $\begin{bmatrix} 1.012 & -2.132 & 3.104 \\ -2.132 & 4.906 & -7.013 \\ 3.104 & -7.013 & 0.014 \end{bmatrix}$ b) $\begin{bmatrix} 2 & -1 & 1 \\ 3 & 3 & 9 \\ 3 & 3 & 5 \end{bmatrix}$

6.4 Dadas las matrices de 4.2, cuáles son definidas positivas?

6.5- Dados los siguientes sistemas lineales de ecuaciones:

a) $\begin{aligned} 3x_1 - x_2 + x_3 &= 1 \\ -x_1 + 6x_2 + 2x_3 &= 0 \\ x_1 + 2x_2 + 7x_3 &= 4 \end{aligned}$ b) $\begin{aligned} 10x_1 - x_2 &= 9 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ -2x_2 + 10x_3 &= 6 \end{aligned}$

c) $\begin{aligned} 10x_1 + 5x_2 &= 6 \\ 5x_1 + 10x_2 - 4x_3 &= 25 \\ -4x_2 + 8x_3 - x_4 &= -11 \\ -x_3 + 5x_4 &= -11 \end{aligned}$

i) resuélvalos por los métodos de Jacobi y Gauss Seidel (tres primeras iteraciones), partiendo del vector cero

ii) con $C=C^{-1}=I$ efectúe dos primeros pasos del método del gradiente conjugado

UNIDAD 7: APROXIMACIÓN

7.1. Aproximación polinómicas por mínimos cuadrados

Se basa en aproximar a través de un polinomio *P(x)* una función dada *y(x)* minimizando los cuadrados de los errores.

Para datos discretos, se minimiza la suma:

$$S = \sum_{i=0}^{N} \left[y_i - a_0 - a_1 x_i - ... - a_1 x_i^{\,m} \right]^2 \qquad (7.1)$$

Con datos x_i e y_i, $\quad m \le N$

Pero $p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m$ no podrá ubicarse a todos los puntos N al ser $m \le N$, con lo cual S no será cero pero si se podrá hacerlo lo más pequeño posible.

Las ecuaciones normales para hallar los a_j son:

$$\begin{aligned} &s_0 a_0 + s_1 a_1 + ... + s_m a_m = t \\ &s_1 a_0 + s_2 a_1 + ... + s_{m+1} a_m = t_1 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &s_m a_0 + s_{m+1} a_1 + ... + s_{2m} a_m = t_m \end{aligned} \qquad (7.2)$$

$$S_k = \sum_{i=0}^{N} x_i^{(k)}, \quad t_k = \sum_{i=0}^{N} y_i x_i^{(k)}$$

El caso más simple es que p(x) represente una recta del tipo.

P(x)=mx+h, con

$$m = \frac{s_0 t_1 - s_1 t_0}{s_0 s_2 + s_1^{\,2}}, \; h = \frac{s_2 t_0 - s_1 t_1}{s_0 s_2 + s_1^{\,2}}$$

cuando el grado del polinomio crece, el sistema 5.2 puede resultar imposible de despejar las a_j, volviéndose necesario introducir polinomios ortogonales (o sea una base ortogonal del espacio vectorial).

Si se trabaja con datos discretos, se tendrán polinomios de grado $m = 0,1,...P_{m,n}(t)$ que cumplan:

$$\sum_{t=0}^{N} P_{m,n}(t) P_{n,n}(t) = 0 \qquad \text{ortogonalidad} \qquad (7.3)$$

Que se puede escribir explícitamente

$$P_{m,n}(t) = \sum_{i=0}^{m} (-1)^i \binom{m}{i} \binom{m+i}{i} \frac{t^{(i)}}{N^{(i)}} \qquad (7.4)$$

En forma alternativa se escribirá el polinomio de mínimos cuadrados

$$p(t) = \sum_{k=0}^{m} a_{k,n}(t) \;\; a_k \qquad (7.5)$$

Los $a_k = \sum_{k=0}^{N} y_1(t) P_{k(n)}(t) \Big/ \sum_{k=0}^{m} P^2_{k_{(n)}(t)}$

$$\sum_{i=0}^{m} (-1)^i \binom{m}{i} \binom{m+i}{i} \frac{t^{(i)}}{N^{(i)}}$$

(7.6) con a_k minimizando la suma S del error

$$S_{\min} = \sum_{t=0}^{N} y_t^{\,2} - \sum_{k=0}^{m} W_k(t) a_k^2$$

(7.7) W_k =suma del divisor de 7.6

Dentro del área de datos discretos, los polinomios de cuadrados mínimos se emplean para suavizar datos, con el grado de *p(x),* según el caso, aunque es muy usada la parábola de cinco puntos y cuadrados mínimos para puntos (x_i, y_i) con $i = k-2, k-1, ..., k+2$.}

$$y(x_k) = p(x_k) = y_k - \frac{3}{35}\partial^4 y_k$$

El error raíz cuadrada de la media de los cuadrados de un conjunto de aproximaciones A_i respecto a los verdaderos V_i será RMS= $S = \left[\sum_{t=0}^{N}(V_i - A_i)^2 / N\right]^{1/2}$

Que dará cierta estimación de la eficiencia del suavizado.
También se puede emplear para diferenciación.
La parábola de cinco puntos lleva a la formula

$$y'(x_k) = p'(x_x) = \frac{1}{10h}(-2y_{k-2} - y_{k-1} + y_{k+1} + 2y_{k+2})$$

Generando resultados mejores a los polinomios ya vistos.
En el caso de datos continuos, representados por *y(x)*, se puede minimizar la integral:

$$I = \int_{-1}^{1}\left[y(x) - a_0 p_0(x) - ... - a_m p_m(x)\right]^2 dx \qquad (7.7)$$

Con los $p_j(x)$ polinomios de Legendre, para *y(x)* integrable.

Es decir que el polinomio de mínimos cuadrados *p(x)* se representara en función de polinomios ortogonales, como:

$$p(x) = a_0 p_0(x) + ... + a_m p_m(x) \qquad (7.8)$$

Los $a_k = \frac{2k+1}{2}\int_{-1}^{1} y(x)p_k(x)dx$ (7.9)

Para el uso de los polinomios de Legendre a veces es mejor considerar el intervalo (0,1) que presentará cambio en la variable del polinomio, conformando lo polinomios de Legendre modificados.
Las integrales de 7.9 deben obtenerse por aproximación o discretizar el conjunto de argumentos continuos de partida para resolver la integral como suma.
También el suavizado y la diferenciación aproximada de funciones de datos continuos *y(x)* son aplicaciones del *p(x)*.
Generalizando se plantea minimizar la integral:

$$J = \int_{a}^{b}\mu(x)\left[y(x) - a_0 Q_0(x) - ... - a_m Q_m(x)\right]^2 dx \qquad (7.10)$$

Con $\mu(x)$ función de peso no negativa.
Los $Q(x)$ representan polinomios ortogonales verificantes de:

$$\int_{a}^{b}\mu(x)Q_j(x)Q_k(x)dx = 0 \qquad j \neq k \qquad (7.11)$$

$$a_k = \frac{\int_{a}^{b}\mu(x)y(x)Q_k(x)dx}{\int_{a}^{b}\mu(x)Q_j^{\,2}(x)dx} \qquad (7.12)$$

Entonces $J_{\min} = \int_{a}^{b}\mu(x)y^{\,2}(x)dx - \sum_{k=0}^{m} W_k a_k^2$ (7.13)

Lo que conduce a la inecuación de Bessel:

$$\sum_{k=0}^{m} W_k a_k^2 \le \int_a^b \mu(x) y^{\,2}(x) dx \qquad (7.14)$$

El error de aproximación $\rightarrow$o para mayor grado de *p(x)* con *y(x)* suficientemente suave.

7.2. POLINOMIOS DE CHEBYSCHEV

Del método general de los cuadrados mínimos con $\mu(x) = \left(1 - x^2\right)^{-1/2}$ constituye el caso especifico que emplea polinomios de Chebyshev para aproximar. Los ortogonales $Q_k(x)$ son los polinomios de Chebyshev

$$H_k(x) = \cos(k \arccos x) \qquad (7.15)$$

Que para los primeros tendrá la forma

$$H_0(x) = 1, H_1(x) = x, H_2(x) = 2x^2 - 1, H_3(x) = 4x^3 - 3x$$

Estos polinomios son propietarios de cualidades como:

$$H_{n+1}(x) = 2xH_n(x) - H_{n-1}(x)$$

$$\int_{-1}^{1} \frac{H_m(x) H_n(x)}{\sqrt{1-x^2}} dx = 0 \text{ si } m \neq n \text{ , } \pi \text{ si } m=n \neq 0; \pi \text{ si } m=n=0$$

$$H_n(x) = 0 = \text{ en el caso de } x = \cos\left[\left(2i+1\right)\pi / 2n\right] \quad con \quad i = 0,1,....n-1 \qquad (7.16)$$

$$H_n(x) = \left(-1\right)^i \text{ en el caso de } x = \cos\left[\pi i / n\right] \quad i = 0,1,....n-1$$

Además, entre ± 1, los polinomios de Chebyshev oscilan de tal forma que alcanzan estos extremos *(n+1)* argumentos dentro de *(-1,1)*,

Haciendo que el error $y(x) - p(x)$ oscile comúnmente entre un máximo y un mínimo de $\pm e$, lo que acarrea que la aproximación es casi infinitamente exacta en todo el intervalo.

Se podrán representar las potencias de *x* en términos de los polinomios de Chebyshev, a través de:

$$1 = H_0(x), x = H_1(x), x^2 = \frac{1}{2}\left[H_0 + H_2\right], x^3 = \frac{3H_1 + H_3}{4}$$

Que se traduce en polinomios economizados donde cada potencia de *x* en un polinomio, se sustituye por una combinación pertinente de polinomios de Chebyshev.

En el caso de que se busque minimizar la suma:

$$\sum_{i=0}^{N-1} \left[y(x_i) - a_0 H_0(x_i) - ... - a_m H_m(x)\right]^2 \qquad (7.17)$$

Se expresan los argumentos x_i $x_i = \cos\left[(2i+1)\pi / 2N\right]$, los ceros de $H_N(x)$

En base a la ortogonalidad de los polinomios de Chebyshev:

$$\sum_{i=0}^{N-1} H_m(x_i) H_n(x) = 0, (n \neq m), N/2 (m = n \neq 0), N (m = n = 0) \qquad (7.18)$$

Los coeficientes serán:

$$a_0 = \frac{1}{N}\sum_{t=0}^{N} y(x_i) \quad a_k = \frac{1}{N}\sum_{t=0}^{N} y(x_i) H_k(x_i) \qquad (7.19)$$

Con lo que el polinomio de aproximación será:

$$p(x) = a_0 H_0(x) + ... + a_m H_m(x)$$

7.3. APROXIMACIÓN MÍNIMO – MÁXIMO (O DE CHEBYSHEV)

También se puede trazar la distinción entre datos discretos y continuos.
En el primer caso, para datos $\left(x_i, y_i\right)$ *(i=1,...N), p(x)*de grado menor o igual que *n* y $h_i = p(x) - y_i$, las desviaciones.
Llámese T al mayor de ellas, el polinomio min-max es el *P(x)* para el cual T es menor.
Posee la propiedad de igual error: llamando *P(x)* al min-max y el error máximo

$$E = \max\left|P(x) - y(x_i)\right| \qquad (7.20)$$

P(x) es el único para el cual la diferencia en 5.20 adopta los valores $\pm E(n+2)$ como mínimo con signo alternante.
El algoritmo de intercambio permite encontrar *P(x)* a través de la propiedad de igual error. Para ello se elige un subconjunto de arranque de *(n+2)* valores x_i y se busca un polinomio de igual error para esos argumentos.
Si el $E_{\max}$ de ese polinomio coincide con T (Máximo total), se esta frente al *P(x)*
En caso contrario, se intercambia algún valor del subconjunto por uno exterior al subconjunto, iterando el procedimiento; habrá convergencia hacia *P(x)*.
Los aspectos rescatables de las técnicas min-máx se parecen a los explicitados en el caso discreto, a saber:

i. La aproximación min-máx a *y(x)* entre todos los polinomios de grado $\leq n$, minimiza el $\max\left|p(x) - y(x)\right|$ en *(a,b)*.
ii. *P(x)* existe y es único.
iii. Posee la propiedad de igual error, con $p(x) - y(x)$ alcanzando valores extremos $\pm E$ en *(a,b)*. Se plantea la recta min-max con (*y''(x) positiva*.)

Así será:

$$P(x) = mx + c$$

$$m = \frac{y(b) - y(a)}{b - a} \qquad c = \frac{y(a) - y(x_2)}{2} - \frac{(a + x_2)\left[y(b) - y(b)\right]}{2(b - a)}$$

A partir de $y'(x_2)$ $y'(x_2) = \left[y(b) - y(b)\right] / b - a$ se determina x_2, con *a,x y b*, los puntos extremos.

iv. los polinomios de serie truncadas de Chebyshev generan aproximaciones casi min-máx, en muchos casos.

7.4. APROXIMACIÓN POR FUNCIONES RACIONALES

Dada la gran variedad de funciones racionales, éstas representan una llave importante para aproximar, por ejemplo, como los polinomios carecen de singularidades se estaría impedido frente a funciones con polos. Se mencionan dos clases de aproximaciones, empleándose las fracciones continuas y mutuas de diferencias.

Las fracciones continuas adoptan la forma:

$$y(x) = y_1 + \cfrac{x - x_1}{\beta_1 + \cfrac{x - x_2}{\beta_2 - y_1 \cfrac{x - x_3}{\beta_3 - \beta_1 + \cfrac{x - x_4}{\beta_4 - \beta_1}}}} \qquad (7.21)$$

Que pueden ampliarse si fuese necesario; en cualquier caso 7.21 tiene la expresión de cociente de polinomios o funciones racionales. Los β se denominan diferencias mutuas y deben escogerse adecuadamente. Para 7.21 serán:

$$\beta_1 = \frac{x_2 - x_1}{y_2 - y_1} \qquad \beta_2 - y_1 = \frac{x_3 - x_2}{\frac{x_3 - x_1}{y_3 - y_1} - \frac{x_2 - x_1}{y_2 - y_1}}$$ así se puede seguir para β_3 y β_4

7.5. APROXIMACIÓN DE FUNCIONES CIRCULARES

De amplio uso, senos y cosenos, por sus propiedades de ortogonalidad y periodicidad que carecen los polinomios.

Para una función de datos en *2L+1* argumentos preestablecidos, se puede obtener una suma trigonométrica de colocación como:

$$y(x) = 0,5 + \sum_{k=1}^{L}\left(a_k \cos\left(\frac{2\pi}{L+1}\right)kx + b_k sen\left(\frac{2\pi}{2L+1}\right)kx \right) \tag{7.22}$$

Pero los senos y cosenos, por su ortogonalidad:

$$\sum_{x=0}^{N} sen\frac{2\pi}{N+1} jx sen\frac{2\pi}{N+1} kx = 0 \qquad (j \neq k); N+1/2(j=k=0) \tag{7.23}$$

$$\sum_{x=0}^{N} sen\frac{2\pi}{N+1} jx \cos\frac{2\pi}{N+1} kx = 0 \tag{7.24}$$

$$\sum_{x=0}^{N} sen\frac{2\pi}{N+1} jx \cos\frac{2\pi}{N+1} kx = 0 \tag{7.25}$$

O para $(j \neq k); N+1/2(j=k \neq 0, N+1); N+1(j=k=0, N+1)$ a partir de las cuales, los coeficientes se obtienen de:

$$a_k = \frac{2}{2L+1}\sum_{x=0}^{2L} y(x) \cos\frac{2\pi}{L+1} kx, k = 0,1,...,L$$

A (7.26)y (7.27)

$$b_k = \frac{2}{2L+1}\sum_{x=0}^{2L} y(x) \cos\frac{2\pi}{L+1} kx, k = 1,2,...,L$$

Que proporcionan la única función de colocación en las condiciones preestablecidas.

Para número par de argumentos (2L)

$$y(x) = \frac{a_0}{2} + \sum_{k=1}^{L-1}\left(a_k \cos\left(\frac{\pi}{L}\right)kx + b_k sen\left(\frac{\pi}{L}\right)kx \right) + a_{L/2} \cos \pi x \tag{7.28}$$

$$a_k = \frac{2}{L}\sum_{x=0}^{2L-1} y(x) \cos\frac{\pi}{L} kx, k = 0,1,...,L$$

A (7.29) y (7.30)

$$b_k = \frac{1}{L}\sum_{x=0}^{2L-1} y(x) sen\frac{\pi}{L} kx, k = 1,2,...,L-1$$

Lo que se minimiza es.

$$S = \sum_{x=0}^{2L}\left[y(x) - H_m(x) \right]^2 \qquad \text{para 2L+1 valores} \tag{7.31}$$

Con $H_m(x) = A_0/2 + \sum_{k=1}^{M}\left(A_k \cos\left(\frac{2\pi}{L+1}\right)kx + B_k sen\left(\frac{2\pi}{2L+1}\right)kx\right)$ para $M \langle L$ (7.32)

Lo que significa adoptar, en el proceso de minimización, $A_k = a_k, B_k = b_k$.

Así, $S_{\min} = \frac{2L+1}{2}\sum_{k=M+1}\left(a_k^2 + b_k^2\right)$ (7.33)

Casos particulares se presentan para funciones pares e impares.
Si es par $y(-x) = y(x)$, si su periodo es *2L* (caso coseno(x))

$a_k = \frac{2}{P}\left[y(0) + y(L)\cos k\pi\right] + \frac{4}{P}\sum_{x=1} y(x)\cos\frac{2\pi}{P}kx \;\; b_k = 0$

Si fuera impar, Ej. $sen(x)$, $y(-x) = -y(x)$, con periodo 2L

$a_k = 0$ y $b_k = \frac{4}{P}\sum_{x=1}^{L-1} y(x) sen\frac{2\pi}{P}kx$

7.6. SERIES DE FOURIER

Para un conjunto de datos continuos, las series de Fourier sustituyen a las sumas finitas trigonometricas, aunque permanecen muchas similitudes.

Sea $y(x)$ sobre $(0, 2\pi)$, la serie adopta la forma $\frac{\alpha_0}{2} + \sum_{k=1}^{\infty}\left(\alpha_k \cos kt + \beta_k senkt\right)$ (7.34)

Los coeficientes $\alpha_k \; y \; \beta_k$ se determinan considerando la ortogonalidad de:

$\int_0^{2\pi} sen(jt)sen(kt)dt = 0 \quad (j \neq k); \qquad \pi \; si \; (j = k \neq 0)$ (7.35)

$\int_0^{2\pi} sen(jt)\cos(kt)dt = 0$ (7.36)

$\int_0^{2\pi} \cos(jt)\cos(kt)dt = 0\,(j \neq k); \qquad \pi; (j = k \neq 0),\; 2\pi(j = k = 0)$ (7.37)

$\alpha_k = \frac{1}{\pi}\int_0^{2\pi} y(t)\cos(kt)dt \qquad \beta_k = \frac{1}{\pi}\int_0^{2\pi} y(t)sen(kt)dt$ (7.38)

El uso se restringe al intervalo $[0, 2\pi]$ pues su periodo es 2π: las expresiones de $\alpha_k \; y \; \beta_k$ se simplifican según sean funciones pares e impares.
Las aproximaciones por mínimos cuadrados; para datos continuos, se obtienen truncando la serie de Fourier, lo minimizará la integral:

$J = \int_0^{2\pi}\left[y(t) - H_m(t)\right]^2 dt$ (7.39)

Con $H_m(t) = \frac{A_0}{2} + \sum_{k=1}^{M}\left(A_0 \cos kt + B_k senkt\right)$ (7.40)

Lo que equivale a hacer $A_k = \alpha_k; B_k = \beta_k$, para minimizar *J*

$J_{\min} = \pi\sum_{k=M+1}^{\infty}\left(\alpha_k^2 + \beta_k^2\right)$ (7.41)

Si $M \to 0; \qquad J_{\min}$ tiene límite cero. Las aplicaciones más extendidas son la suavización de datos y diferenciación por aproximación.

7.8. APROXIMACIÓN POR VALORES PROPIOS

Si se presentarán teoremas y definiciones de utilidad para el desarrollo del tema.

T_1: Si A es una matriz y $\lambda_1,...,\lambda_k$ son valores propios bien definidos de A con sus vectores propios asociados $\vec{x}^{(1)},\vec{x}^{(1)},...,\vec{x}^{(k)}$, el conjunto $\left\{\vec{x}^{(1)},\vec{x}^{(1)},...,\vec{x}^{(k)}\right\}$ es linealmente independiente.

- Un conjunto de vectores $\left\{\vec{u}^{(1)},\vec{u}^{(1)},...,\vec{u}^{(n)}\right\}$ será ortogonal si $\left(\vec{u}^{(i)^t}\right)\vec{u}^{(j)}=0$ con $i\neq j$; en el caso adicional $\left(\vec{u}^{(i)^t}\right)\vec{u}^{(j)}=1$ para *i=1,2,...,n,* el conjunto será ortonormal 5.41

O también si: $\left\|\vec{u}^{(i)}\right\|_e$=1, *i=1,2,...,n* *(* 7.42)

T_2: Si un conjunto de vectores, que no contiene al vector nulo, es ortogonal, los vectores son linealmente independientes.

- para una matriz A(nxn) si $A^{-1}=A^t$, la matriz es ortogonal (las matrices de permutación P son ortogonales).

Similiridad: dos matrices A y B, ambas nxn, son similares si existe una matriz S no singular que verifique $A=S^{-1}BS$.

T_3: si *A* y *B* (nxn) son similares, λ un valor propio asociado a un vector propio $\vec{x}$ será λ un valor propio de b y además si $A=S^{-1}BS$, $S\vec{x}$ es un vector propio asociado a λ y a la matriz *B*.

A partir que a través de $\prod_{i=1}^{n}\left(a_{ii}-\lambda\right)=0$ se puede hallar un valor propio de una matriz triangular, se establece la relación entre matrices cualesquiera y las triangulares:

Sea *A*(nxn), existirá una V no singular que cumple $T=V^{-1}AU$ (transformación de similaridad) 5.43.

Con *T* (triangular superior) cuyos elementos de la diagonal son los valores propios de A.

Si se reduce la matriz de transformación a una ortogonal, se hace más sencillo determinar la transformación de similitud.

T_4: si *A* es simétrica (nxn) y D es una diagonal cuyos elementos son los valores propios de A, Existirá una matriz ortogonal *P/D*= $A=P^{-1}AP=P^tAP$. Puede inferirse de T_4 que para *A* simétrica de nxn los valores propios de A son números reales y existen n vectores propios de *A* que conforman un conjunto ortonormal.

T_5: una matriz simétrica *A* es definida positiva si y solo sí todos los valores propios de *A* son positivos, lo que genera un criterio más sencillo que $\vec{x}^tA\vec{x}\rangle 0$ (visto en unidad 3).

Cotas para valores propios.

Dada *A*(nxn) y R_i el círculo en el plano complejo con a_{ii} como centro y de radio $\sum_{\substack{j=1\\ j\neq i\neq}}^{n}\left|a_{ij}\right|$, es

decir, $R_i=\left\{z\in C/\left|z-a_{ii}\right|\leq\sum_{\substack{j=1\\ j\neq i\neq}}^{n}\left|a_{ij}\right|\right\}$ (7.44)

Los autovalores de A están dentro de R=$\bigcup_{i=1}^{n}R_i$ (círculo de Gerschgorin)

Ejemplo 7.1
Para A, acotar los autovalores empleando los círculos de Gerschgorin

$$A = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 2 & 1 \\ -2 & 0 & 9 \end{pmatrix}$$

$$R_1 = \{z \in C / |z-4| \le 2\}$$

$$R_1 = \{z \in C / |z-2| \le 1\}$$

$$R_1 = \{z \in C / |z-9| \le 2\}$$

Como R_1 y R_2 no tienen que ver con R_3, habrá dos valores característicos dentro de $R_1 \, UR_2$ y uno en R_3; como $\rho(A)=máx_{1\le j\le 3} \mid \lambda_i \mid$ se tiene $7 \le \rho(A) \le 11$

7.8.1. Técnicas de Aproximación

7.8.1.1. Métodos de Potencias

A debe ser (nxn), con n valores propios $\lambda_1, \lambda_2, ..., \lambda_k$ con un conjunto asociado de vectores propios $\{\vec{v}^{(1)}, \vec{v}^{(2)}, ..., \vec{v}^{(n)}\}$ linealmente independientes y uno de los λ debe ser el más grande.

Estableciendo $|\lambda_1| \ge |\lambda_2| \ge ... \ge |\lambda_n| \ge 0$, $\vec{x}$ vector en R^n, al ser $\{\vec{v}^{(1)}, \vec{v}^{(2)}, ..., \vec{v}^{(n)}\}$ linealmente independiente, existirán las constantes $\alpha_1, \alpha_2, ..., \alpha_n$ que verifiquen:

$$\vec{x} = \sum_{j=1}^{n} \alpha_j \vec{v}^{(j)}$$

Si se va haciendo el producto de los A^k,

$$A\vec{x} = \sum_{j=1}^{n} \alpha_j A\vec{v} = \sum_{j=1}^{n} \alpha_j \lambda_j \vec{v}^{(j)}$$

$$\vdots$$

$$\vdots$$

$$A^k\vec{x} = \sum_{j=1}^{n} \alpha_j \lambda^k{}_j \vec{v}^{(j)}$$

Al sacar factor común $\lambda^k{}_1$ del segundo miembro

$$A^k\vec{x} = \lambda^k{}_1 j = \lambda^k{}_1 \sum_{j=1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^k \vec{v}^{(j)}$$

Pero como $|\lambda_1| \ge |\lambda_j|$ para $j = 2, 3, ..., n$ el $\lim_{k\to\infty} \left(\frac{\lambda_j}{\lambda_1}\right)^k = 0$ quedando:

$$\lim_{k\to\infty} A\vec{x} = \lim_{k\to\infty} \alpha_j \lambda_1^k \vec{v}^{(1)}$$

Sucesión que converge a cero para $|\lambda_1| < 1$ pero diverge si $|\lambda_1| \ge 1$ (para $\alpha_1 \ne 0$) (7.45)

Practicando reescalado en 5.45, tomando $\vec{x}$ como un vector unitario $\vec{x}^{(0)}$, llamada $\vec{x}^{(0)} = 1 = \|\vec{x}^{(0)}\|$.

Notando $\vec{y}^{(1)} = A\vec{x}^{(0)}$ y definiendo $\mu^{(1)} = \vec{y}^{(0)}$, quedará:

$$\mu^{(1)} = \vec{y}_{p,0}^{(1)} = \frac{\vec{y}_{p,0}^{(1)}}{\vec{x}_{p,0}^{(1)}} = \frac{\alpha_1\lambda_1 v^1{}_{p,0} + \sum_{j=2}^{n}\alpha_j\lambda_j\vec{v}_{p,0}{}^{(j)}}{\alpha_1 v^{(1)}{}_{p,0} + \sum_{j=2}^{n}\alpha_j\vec{v}_{p,0}{}^{(j)}} = \frac{\alpha_1 v^1{}_{p,0} + \sum_{j=2}^{n}\alpha_j\left(\lambda_j/\lambda_1\right)\vec{v}_{p,0}{}^{(j)}}{\alpha_1 v^{(1)}{}_{p,0} + \sum_{j=2}^{n}\alpha_1\vec{v}_{p,0}{}^{(j)}}\lambda_1$$

Tomando p_1, como el menor entero que cumple $\left|\vec{y}_{p,0}^{(1)}\right| = \left\|\vec{y}^{(1)}\right\|$ y definiendo $\vec{x}^{(1)}$ como:

$$\vec{x}^{(1)} = \frac{1}{\vec{y}_{p,1}^{(1)}}\vec{y}^{(1)} = \frac{1}{\vec{y}_{p,1}^{(1)}}A\vec{x}^{(0)}$$

Será $\vec{x}_{p,1}^{(1)} = 1 = \left\|\vec{x}^{(1)}\right\|$

Ahora par $\vec{y}^{(2)} = A\vec{x}^{(1)} = \frac{1}{\vec{y}_{p,1}^{(1)}}A^2\vec{x}^{(0)}$

Con

$$\mu^2 = \vec{y}_{p,1}^{(2)} = \frac{\vec{y}_{p,1}^{(2)}}{\vec{x}_{p,0}^{(1)}} = \frac{\left[\alpha_1\lambda_1{}^2 v^1{}_{p,1} + \sum_{j=2}^{n}\alpha_j\lambda_j^2\vec{v}_{p,1}{}^{(j)}\right]/\vec{y}_{p,1}^{(1)}}{\left[\alpha_1\lambda_1 v^{(1)}{}_{p,1} + \sum_{j=2}^{n}\alpha_j\lambda_j\vec{v}_{p,1}{}^{(j)}\right]/\vec{y}_{p,1}^{(1)}} = \frac{\alpha_1 v^1{}_{p,1} + \sum_{j=2}^{n}\alpha_j\left(\lambda_j/\lambda_1\right)^2\vec{v}_{p,1}{}^{2(j)}}{\alpha_1 v^{(1)}{}_{p,1} + \sum_{j=2}^{n}\alpha_1\left(\lambda_j/\lambda_1\right)\vec{v}_{p,1}{}^{(j)}}\lambda_1$$

Ahora para p_2 menor entero $/\left|\vec{y}_{p,2}^{(2)}\right| = \left\|\vec{y}^{(2)}\right\|$ se define

$$\vec{x}^{(2)} = \frac{\vec{y}^{(2)}}{\vec{y}_{p,2}^{(2)}} = \frac{A\vec{x}^{(1)}}{\vec{y}_{p,2}^{(2)}} = \frac{A^2\vec{x}^{(0)}}{\vec{y}_{p,2}^{(2)}} = \frac{A^2\vec{x}^{(0)}}{\vec{y}_{p,2}^{(2)}\vec{y}^{(1)}}$$

Inductivamente se generarán sucesiones $\left\{\vec{x}^{(m)}\right\}_{m=0}^{\infty}, \left\{\vec{y}^{(m)}\right\}_{m=1}^{\infty}$ y de escalares $\left\{u^{(m)}\right\}_{m=1}^{\infty}$, a través de:

$$\mu_m = \vec{y}_{p_{m-1}}^{(m)} = \frac{\vec{y}_{p,1}^{(2)}}{\vec{x}_{p,0}^{(1)}} = \lambda_1\left[\frac{\alpha_1 v^1{}_{p_{m-1}} + \sum_{j=2}^{n}\left(\lambda_j/\lambda_1\right)^m\alpha_j\vec{v}_{p_{m-1}}{}^{(j)}}{\alpha_1\vec{v}_{p_{m-1}}{}^{(j)} + \sum_{j=2}^{n}\left(\lambda_j/\lambda_1\right)\alpha_j\vec{v}_{p_{m-1}}{}^{(j)}}\right],\ \vec{x}^{(m)} = \frac{\vec{y}^{(m)}}{\vec{y}_{pm}^{(m)}} = \frac{A\vec{x}^{(1)}}{\vec{y}_{p,2}^{(2)}} = \frac{A^m\vec{x}^{(0)}}{\prod_{k=1}^{m}\vec{y}_{p_k}^{(k)}}$$

(7.46)

En cada etapa se emplea p_m como menor entero para $\left|\vec{y}_{pm}^{(m)}\right| = \left\|\vec{y}^{(m)}\right\|_\infty$. Como $\left|\lambda_j/\lambda_1\right| < 1$ para cada *j=2,3,...* $\lim_{\infty}\mu^{(m)} = \lambda_1$ siempre que se elija $\vec{x}^{(0)}$ tal que $\alpha_i \neq 0$ y $\left\{\vec{x}^{(m)}\right\}_{m=0}^{\infty}$ convergerá a un vector propio de norma 1 asociado a λ_1. Los inconvenientes del método pasan por no saber, previamente, si la matriz tiene valor propio dominante y como elegir $\vec{x}^{(0)}$. Para acelerar la convergencia, puede emplearse el método de Aitken.

Ejemplo 7.2

Utilizar el método de la potencia para calcular el autovalor dominante y el correspondiente autovector para la matriz

$$A = \begin{pmatrix} -4 & 14 & 0 \\ -4 & 13 & 0 \\ 1 & 0 & 2 \end{pmatrix}$$

utilizando la aproximación inicial $x_{(0)} = \left(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}\right)^t$ y una tolerancia TOL =

10^{-3}. El resultado exacto es: $\lambda_1 = 6$ y

$$v = \frac{1}{\sqrt{1233}}(28, 20, -7) \approx (0.797400, 0.569572, -0.199350)$$

Empleando
$y^{(k+1)} = Ax^{(k)}$, $x^{(k+1)} = y^{(k+1)}/||y^{(k+1)}||$, se genera la tabla

k	$\lambda(k)$	$x_1(k)$	$x_2(k)$	$x_3(k)$
1	6.933334	0.778499	0.622799	0.077850
2	6.475747	0.796858	0.597644	-0.088540
3	6.230470	0.798244	0.583332	-0.150097
4	6.112702	0.797990	0.576326	-0.176237
5	6.055632	0.797721	0.572909	-0.188194
6	6.027620	0.797564	0.571228	-0.193884
7	6.013756	0.797482	0.570396	-0.196649
8	6.006863	0.797441	0.569983	-0.198009
9	6.003428	0.797420	0.569777	-0.198683

Deteniéndose en la iteración nueve pues $||x_{(9)} - x_{(8)}|| = 7.04 \times 10{-}4 < TOL$

7.8.1.2. Método simétrico de Potencias

Si A es simétrica, modificando la elección de los vectores $\vec{x}^{(m)}, \vec{y}^{(m)}$ y los escalares $\mu^{(m)}$ se puede incrementar la velocidad de convergencia de la sucesión de escalares $\{u^{(m)}\}_{m=1}^{\infty}$ hacia el λ_1 dominante.

El método general de potencias presenta una velocidad de convergencia del orden $O\left(|\lambda_2 / \lambda_1|^m\right)$, el método simétrico lo hará con $O\left(|\lambda_2 / \lambda_1|^{2m}\right)$.

Incluso el método inverso de potencias es una mejora en la convergencia respecto al método normal empleándose para precisar el valore propio de A más próximo a uno específico q(por el círculo de Gerschgorin).

Se calcula q de una aproximación inicial al vector propio $\vec{x}^{(0)}$: $q = \dfrac{\vec{x}^{(0)t} A \vec{x}^{(0)}}{\vec{x}^{(0)t} \vec{x}^{(0)}}$

Para elegir q se considera que si $\vec{x}$ es vector propio de A respecto al valor propio λ, se dará que $A\vec{x} = \lambda\vec{x}$, entonces $\vec{x}^t A\vec{x} = \lambda \vec{x}^t \vec{x}$ y

$$\lambda = \frac{\vec{x}^t A \vec{x}}{\vec{x}^t \vec{x}}$$

En el caso que q este próximo a un valor propio, la convergencia será rápida.

7.8.1.3. Técnicas de Deflación

Se emplean para la determinación de aproximaciones a los otros valores propios de una matriz después de haber encontrado la aproximación al valor propio dominante.
Consiste en generar una nueva matriz B, con valores propios iguales a los de A, salvo que el valor propio dominante de A se sustituya por el valor propio 0 en B.
La técnica es muy sensible a errores de redondeo.

7.8.1.4. Técnica de Householder

Se aplica para hallar una matriz tridiagonal simétrica B, similar a la A simétrica dada.

A es similar a una matriz diagonal D ya que existe una matriz ortogonal Q tal que $D = Q^{-1}AQ = Q^tAQ$ pero el calculo de Q no es sencillo, lo que se salva por la transformación de Householder para posteriormente emplear algoritmos para aproximar los valores propios de la matriz simétrica tridiagonal que resulta (caso algoritmo QR).
La matriz nxn dada por $P = U - \vec{w}\vec{w}^t$ con $\vec{w} \in R^n$, se denomina matriz de Householder.
Mediante la técnica, se generan bloques de elementos iguales a cero en vectores o columnas de matrices tal que se estabilice respecto al error de redondeo.

Ejemplo 7.3

Tomando la matriz *A*, aplicar Householder

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix} \text{ simétrica,}$$

$$\alpha = -\text{sgn}(a^{(k)}{}_{k+1,k})\left(\sum_{j=k+1}^{n} \left(a_{jk}{}^{(k)}\right)^2 \right)^{1/2}$$

$$r = \left(\frac{1}{2}\alpha^2 - \frac{1}{2}\alpha\alpha_{k+1,k}^{(k)} \right)^{1/2}$$

Como $w_1^{(k)} = w_2^{(k)} = ... = w_k^{(k)} = 0$

$$w_{k+1}^{(k)} = \frac{a_{k+1,k}^{(k)} - \alpha}{2r}$$

$$w_j^{(k)} = \frac{\alpha_{jk}^{(k)}}{2r}, \text{ para cada } j = k+2, k+3, ..., n$$

$P^{(k)}=I-2\mathbf{w}^{(k)}.(\mathbf{w}^{(k)})^t$ y $A^{(k+1)}=P^{(k)}A^{(k)}P^{(k)}$

$$\alpha = -(1)\left(\sum_{j=2}^{4} \left(a_{j1}{}^{(2)}\right) \right)^{1/2} = -3$$

$$r = \left(\frac{1}{2}(-3)^2 - \frac{1}{2}(1)(-3) \right)^{1/2} = \sqrt{6}$$

$$w = \left(0, \frac{\sqrt{6}}{3}, -\frac{\sqrt{6}}{6}, \frac{\sqrt{6}}{6} \right),$$

$$P^{(1)}= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - 2\left(\frac{\sqrt{6}}{6} \right)^2 \begin{bmatrix} 0 \\ 2 \\ -1 \\ 1 \end{bmatrix} \bullet (0, 2, -1, 1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1/3 & 2/3 & -2/3 \\ 0 & 2/3 & 2/3 & 1/3 \\ 0 & 2/3 & 1/3 & 2/3 \end{bmatrix}$$

y

$$A^{(2)} = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & 10/3 & 1 & 4/3 \\ 0 & 1 & 5/3 & -4/3 \\ 0 & 4/3 & -4/3 & -1 \end{bmatrix}$$

La segunda iteración conduce a

$$\alpha = -5/3; r = 2\sqrt{5}/3, w = \left(0, 0, 2\sqrt{5}, \sqrt{5}/5\right)$$

$$P^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -3/5 & -4/5 \\ 0 & 0 & -4/5 & 3/5 \end{bmatrix}$$

La matriz tridiagonal simétrica será:

$$A^{(2)} = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & 10/3 & -5/3 & 0 \\ 0 & -5/3 & -23/25 & 68/75 \\ 0 & 0 & 68/75 & 149/75 \end{bmatrix}$$

7.8.1.5. Algoritmo QR

Una vez presentada la matriz tridiagonal, $A^{(i)}$ se factoriza como el producto: $A^{(i)} = Q^{(i)} R^{(i)}$ con $Q^{(i)}$ matriz ortogonal y $R^{(i)}$ una triangular superior.

Luego $A^{(i+1)}$ se define por el producto $R^{(i)} \vec{y} Q^{(i)}$ en dirección opuesta, o sea $A^{(i+1)} = R^{(i)} Q^{(i)}$.

Al ser $Q^{(i)}$ ortogonal, $A^{(i+1)} = R^{(i)} Q^{(i)} = \left(Q^{(i)t} A^{(i)}\right) Q^{(i)} = Q^{(i)t} A^{(i)} Q^{(i)}$

(7.47)

Y $A^{(i+1)}$ es simétrica y tridiagonal con iguales valores propios que $A^{(i)}$. Inductivamente $A^{(i+1)}$ tiene iguales valores propios que $A^{(i)}$, con $A^{(i+1)}$ tendiendo a una matriz diagonal con los valores propios de A en su diagonal.

7.9. Ortogonalización de Gram-Schdmidt

Carta blanca.lnk El método de ortogonalización de Gram–Schmidt es un método de *ortogonalizar* un conjunto de vectores en un espacio euclídeo R^n. Ortogonalizar equivale a partir de vectores $\boldsymbol{v_1,\ldots,v_k}$, linealmente independientes y se desea hallar mutuamente vectores ortogonales $\boldsymbol{u_1,\ldots,u_k}$, que generan el mismo subespacio que los $\boldsymbol{v_1,\ldots,v_k}$

Definiendo el operador proyección como

$$\operatorname{proj}_{\mathbf{u}} \mathbf{v} = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}.$$

Que proyecta ortogonalmente **v** sobre **u**

Los dos primeros pasos del método de Gram–Schmidt

$$\mathbf{u}_1 = \mathbf{v}_1,$$
$$\mathbf{e}_1 = \frac{\mathbf{u}_1}{||\mathbf{u}_1||}$$
$$\mathbf{u}_2 = \mathbf{v}_2 - \operatorname{proj}_{\mathbf{u}_1} \mathbf{v}_2,$$
$$\mathbf{e}_2 = \frac{\mathbf{u}_2}{||\mathbf{u}_2||}$$
$$\mathbf{u}_3 = \mathbf{v}_3 - \operatorname{proj}_{\mathbf{u}_1} \mathbf{v}_3 - \operatorname{proj}_{\mathbf{u}_2} \mathbf{v}_3,$$
$$\mathbf{e}_3 = \frac{\mathbf{u}_3}{||\mathbf{u}_3||}$$

$$\mathbf{u}_k = \mathbf{v}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j} \mathbf{v}_k,$$

$$\mathbf{e}_k = \frac{\mathbf{u}_k}{||\mathbf{u}_k||}$$

Se debe ir verificando que su producto escaalr sea nulo (ortogonalidad) y finalmente dividir por su norma

Ejemplo 7.4

Sean $\boldsymbol{v_1}=(3\ 1)^t$,$\boldsymbol{v_2}=(2\ 2)^t$; se aplica G-SCh para halllar los ortogonales

$\boldsymbol{u_1}=\boldsymbol{v_1}=(\ 3\ 1)^t$

$\boldsymbol{u_2}=\boldsymbol{v_2}$-proy$_{\boldsymbol{u1}}\boldsymbol{v_2}$=$(2\ 2)^t$-proy$_{(3\ 1)t}(2\ 2)^t$=$(\ -2/5\ 6/5)^t$

Haciendo $<\boldsymbol{u_1}\ \boldsymbol{u_2}>=0$

Se normaliza dividiendo por su longitud, obteniéndose

$$\mathbf{e}_1 = \frac{1}{\sqrt{10}}\begin{pmatrix}3\\1\end{pmatrix}$$

$$\mathbf{e}_2 = \frac{1}{\sqrt{\frac{40}{25}}}\begin{pmatrix}-2/5\\6/5\end{pmatrix} = \frac{1}{\sqrt{10}}\begin{pmatrix}-1\\3\end{pmatrix}.$$

7.10. EJERCITACIÓN UNIDAD 7 CON MATLAB

I) Dada A=[0 11 -5;-2 17 -7;-4 26 -10]; hallar el autovalor y autovector dominante

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];
>> X=[1 1 1]';
>> power1(A,X,0.001,10)
ans =
   4.0016
```

Por el método de inverso de potencias

```
>> X=[1 1 1]';
>> power1(A,X,0.001,10)
ans =
   4.0016
>> alpha=4.1;
>> invpow(A,X,alpha,0.001,10)
ans =
   4.2000

>> alpha=2.1;
>> invpow(A,X,alpha,0.001,10)
ans =
   2.2000
```

Alpha=dada aproximación al valor buscado

II) Para A simétrica, hallar sus autovalores y autovectores

```
>>A =

    8   -1    3   -1
   -1    6    2    0
    3    2    9    1
   -1    0    1    7
```

```
>> jacobi1(A,0.001)
ans =
   0.5288   -0.5730    0.5822    0.2306
   0.5920    0.4721    0.1761   -0.6291
  -0.5360    0.2820    0.7925   -0.0710
   0.2875    0.6077    0.0443    0.7390
```

La salida es una matriz 4x4 de eigenvectores y diagonal de eigenvalores

III) Reducir una matriz simétrica a la forma tridiagonal(por transformación de Householder)

```
>> house (A)
ans =

   8.0000    3.3166         0         0
   3.3166    6.9091   -2.4663         0
        0   -2.4663    8.2431   -0.7931
        0         0   -0.7931    6.8478
```

IV) Aproximar los autovalores con el método QR

```
>> qr1(A,0.001)
ans =

   3.2979
   8.3877
   7.8773
  10.4371
```

O con

```
>> qr2(A,0.001)
ans =

  11.7043
   3.2957
   8.4077
   6.5923
```

% A tridiagonal simétrica cuadrada

V) Dados la matriz x, y la fila v aplicar la ortogonalización de Gram Schdmit Grams Schmidt

```
> x=[4 3 0;3 4 -1;0 -1 4];v=[1 2 2];
>> gschmidt(x,v)
ans =

   4.0000   -0.8400   -0.9730
   3.0000    1.1200    1.2973
        0   -1.0000    2.2703
(The Gram-Schmidt process on the columns in matrix
%       x. The orthonormal basis appears in the columns of y
%       unless there is a second argument in which case y
%       contains only an orthogonal basis. The second argument
%       can have any value)
```

EJERCICIOS PROPUESTOS PARA UNIDAD 7

7.1. Dadas las matrices

a) $\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & -1 & 2 \end{bmatrix}$ b) $\begin{bmatrix} 4.75 & 2.25 & -0.25 \\ 2.25 & 4.75 & 1,25 \\ -0.25 & 1.25 & 4.75 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 0 & -1 & 1 \\ 2 & 2 & -1 & 1 \\ 0 & 1 & 3 & -2 \\ 1 & 0 & 1 & 4 \end{bmatrix}$

Determinar las cotas de los autovalores mediante el método de Gerschgorin

7.2. Dadas las matrices

a) $\begin{bmatrix} 1 & -1 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 2 \end{bmatrix}$ $\vec{x}^{(0)} = (-1,2,1)^t$

b) $\begin{bmatrix} 5 & -2 & -0.5 & 1.5 \\ -2 & 5 & 1.5 & -0.5 \\ -0.5 & 1.5 & 5 & -2 \\ 1.5 & -0.5 & -2 & 5 \end{bmatrix}$ $\vec{x}^{(0)} = (1,1,0,-3)^t$

Halle las tres primeras iteraciones aplicando

i) método de potencia

ii) método de potencia inversa

7.3. Para las matrices

a) $\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$ b) $\begin{bmatrix} 5 & -2 & -0.5 & 1.5 \\ -2 & 5 & 1.5 & -0.5 \\ -0.5 & 1.5 & 5 & -2 \\ 1.5 & -0.5 & -2 & 5 \end{bmatrix}$

c) $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

Ponerlas en forma tridiagonal empleando el método de Householder

7.4. Para las matrices

i) a) $\begin{bmatrix} 4 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$ b) $\begin{bmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix}$ efectúe las dos primeras iteraciones del algoritmo QR

ii)Con una exactitud de 0.001, determine los autovalores de las matrices con el algoritmo QR

$$a)\begin{bmatrix} 5 & -1 & 0 & 0 & 0 \\ -1 & 4.5 & 0.2 & 0 & 0 \\ 0 & 0.2 & 1 & -0.4 & 0 \\ 0 & 0 & -0.4 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix} \quad b)\begin{bmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

UNIDAD N° 8: SISTEMAS DE ECUACIONES NO LINEALES

8.1. MÉTODO DE NEWTON

Un sistema de ecuaciones no lineal puede presentarse como:

$$\begin{aligned} &f_1(x_1, x_2, ..., x_n) = o \\ &f_2(x_1, x_2, ..., x_n) = o \\ &\quad\vdots \\ &\quad\vdots \\ &f_n(x_1, x_2, ..., x_n) = o \end{aligned} \qquad (8.1)$$

Con las f_i mapeando un vector $\vec{x} = (x_1, x_2, ..., x_n)^t$ de R^n en R.

Si se asume una función $\overline{F}$ de R^n en R^n por:

$\overline{F}(x_1, x_2, ..., x_n) = \left[f_1(x_1, x_2, ..., x_n), f_2(x_1, x_2, ..., x_n), ..., f_3(x_1, x_2, ..., x_n) \right]^t$ o $\overline{F}(\vec{x}) = \vec{O}$

(8.2)

Recordando el concepto de continuidad de funciones multivariables en un punto, empleando derivadas parciales:

Para $f : D \subset R^n \to R^n$ con $\vec{x}_o \in D$

Si existen constantes $\gamma \rangle 0$ y $k \rangle 0$ tales que:

$\left| \dfrac{\partial f(\vec{x})}{\partial x_j} \right| \leq K, \quad j = 1, 2, ..., n$ siempre que $\left\| \vec{x} - \vec{x}_0 \right\| \langle \gamma \quad y \quad \vec{x} \in D$, f será continua en $\vec{x}_0$ para funciones de $R^n \to R^n$

Sea la función $\overline{F}$ en $D \subset R^n \to R^n$, tendrá punto fijo en $\vec{q}$ si $\overrightarrow{G(\vec{q})} = \vec{q}$ (8.3)

Si $\overline{F}$ tiene derivadas parciales continúas y existe una constante $k \langle 1$ de modo que:

$$\left| \frac{\partial f_i(\vec{x})}{\partial x_j} \right| \leq \frac{K}{n}, \quad \vec{x} \in D \quad j = 1, 2, ..., n$$

La sucesión $\left\{ \vec{x}^{(k)} \right\}_{k=0}^{\infty}$ definida por una $\vec{x}^{(0)}$ en D, arbitraria, y que se obtiene de $\vec{x}^{(k)} = \overline{F}(\vec{x}^{(k-1)})$; $k \geq 1$, converge al punto fijo $\vec{q} \in D$, verificándose $\left\| \vec{x}^{(k)} - \vec{q} \right\| \leq \dfrac{K^k}{1-K} \left\| \vec{x}^{(1)} - \vec{x}^{(0)} \right\|$

Para la situación de n dimensiones, se presenta la matriz:

$$A(\vec{x}) = \begin{bmatrix} a_{11}(x) & a_{12}(x) & \cdots & \cdots & \cdots & a_{1n}(x) \\ a_{21}(x) & a_{22}(x) & \cdots & \cdots & \cdots & a_{2n}(x) \\ \vdots & \vdots & & & & \\ \vdots & \vdots & & & & \\ \vdots & \vdots & & & & \\ a_{n1}(x) & a_{n2}(x) & \cdots & \cdots & \cdots & a_{nn}(x) \end{bmatrix} \qquad (8.4)$$

Con los $a_{ij}(x)$ funciones de R^n en R.

Entonces se deberá encontrar $A(\vec{x})$ que verifique:

$F(\vec{x}) = \vec{x} - A(\vec{x})^{(-1)} G(\vec{x})$

y proporcione convergencia cuadrática a la solución $G(\vec{x})=0$, partiendo que $A(\vec{x})$ es no singular en un punto fijo $\vec{q}$ de $\overline{F}$.

Será útil introducir el siguiente teorema:

T_1**.-**Sea $\vec{q}$ una solución de $\overline{F}(\vec{x})=\vec{x}$ para $\overline{F}=(f_1,f_2,...,f_n)^t$ de R^n en R. si existe una constante $\gamma\rangle 0$ tal que:

i. $\partial f_i/\partial x_j$ es continua en $S_\gamma=\left\{\vec{x}/\left\|\vec{x}-\vec{q}\right\|\langle\gamma,\quad i=1,2,...,n\right\}\qquad y\ j=1,2,...,n$

ii. $\partial^2 f_i(\vec{x})/\partial x_j\partial x_k$ es continua y $\left|\partial^2 f_i(\vec{x})/\partial x_j\partial x_k\right|\le M$ (cte) siempre que $\vec{x}\in S_\gamma$ para $i=1,2,...,n;\quad j=1,2,...,n;\quad k=1,2,...,n$

iii. $\partial f_i(\vec{q})/\partial x_k\partial x_k=0,\quad i=1,2,...,n;\quad k=1,2,...,n$

existirá una $\vec{\gamma}\le\gamma$ de forma que la sucesión $\vec{x}^{(k)}=F(\vec{x}^{(k-1)})$ converge cuadraticamente en $\vec{q}$ para cualquier $\vec{x}^{(0)}$ siempre que $\left\|\vec{x}^{(0)}-\vec{q}\right\|\le\frac{h^2M}{2}\left\|\vec{x}^{(k-1)}-\vec{q}\right\|^2,\ k\ge 1$

Trabajando con lo expresado en el T_1 y suponiendo que $A(\vec{x})$ no singular cerca de la solución $\vec{q}$ de $G(\vec{x})=0$ y sea $b_{ij}(\vec{x})$ el elemento de $A(\vec{x})^{-1}$ en la fila *i* y columna *j*.

$$F(\vec{x})=\vec{x}-A(\vec{x})^{(-1)}G(\vec{x})\quad f_i(\vec{x})=x-\sum_{j=1}^{n}b_{ij}(\vec{x})g_j(\vec{x})$$

$$\text{Y }\partial f_i(\vec{x})/\partial x_k=\begin{cases}1-\sum_{j=1}^{n}\left(b_{ij}(\vec{x})\right)\frac{\partial g_j}{\partial x_k}(\vec{x})+\frac{\partial b_{ij}}{\partial x_k}(\vec{x})g_j(\vec{x}),\ i=k\\ -\sum_{j=1}^{n}\left(b_{ij}(\vec{x})\right)\frac{\partial g_j}{\partial x_k}(\vec{x})+\frac{\partial b_{ij}}{\partial x_k}(\vec{x})g_j(\vec{x}),\quad i\ne k\end{cases}$$

Como se requiere que $\partial f_i(\vec{q})/\partial x_k=0,\qquad i=1,2,...,n;\quad k=1,2,...,n$ en el caso *i=k*

$$\left\{1=\sum_{j=1}^{n}\left(b_{ij}(\vec{x})\right)\frac{\partial g_j}{\partial x_k}(\vec{q})\right\}\tag{8.5}$$

En el caso $i\ne k$

$$\left\{0=\sum_{j=1}^{n}\left(b_{ij}(\vec{x})\right)\frac{\partial g_j}{\partial x_k}(\vec{q})\right\}\tag{8.6}$$

Recordando la expresión de la matriz jacobiana J:

$$J(\vec{x})=\begin{bmatrix}\frac{\partial g_1(\vec{x})}{\partial x_1} & \frac{\partial g_1(\vec{x})}{\partial x_2} & \cdots & \frac{\partial g_1(\vec{x})}{\partial x_n}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial g_n(\vec{x})}{\partial x_1} & \frac{\partial g_n(\vec{x})}{\partial x_2} & \cdots & \frac{\partial g_n(\vec{x})}{\partial x_n}\end{bmatrix}\tag{8.7}$$

De 8.5 y 8.6

$A(\vec{q})^{(-1)}J(\vec{q})=U$, con U la matriz Identidad

O sea $A(\vec{q})=J(\vec{q})$, lo que equivale a elegir $A(\vec{x})$ como la jacobiana de $J(\vec{x})$

Definiendo $F(\vec{x})=\vec{x}-J(\vec{x})^{(-1)}G(\vec{x})$, seleccionando $\vec{x}^{(0)}$ e iterando, se generará para $k\ge 1\qquad \vec{x}^{(k)}=F(\vec{x}^{(k-1)})=\vec{x}^{(k-1)}-J(\vec{x}^{(k-1)})^{(-1)}G(\vec{x}^{(k-1)})\tag{8.8}$

constituyendo el método de Newton para sistemas no lineales, esperando convergencia cuadrática, si se conoce $\vec{x}^{(0)}$ preciso y si se puede hallar $J(\vec{q})^{-1}$.

Para facilita la obtención de $J(\vec{x})^{-1}$ se opera en dos pasos:

1. hallar un vector $\vec{y}$ que satisfaga $J(\vec{x}^{(k-1)})\vec{y} = -G(\vec{x}^{(k-1)})$
2. luego la nueva aproximación $\vec{x}^{(k)}$ se consigue sumando $\vec{y}$ a $\left(\vec{x}^{(k-1)}\right)$

Ejemplo 8.1- Resolver aproximadamente el sistema no lineal

$$3x_1 - \cos(x_2 x_3 -) - \frac{1}{2} = 0$$

$$x_1^2 - 81(x_2 + 0.1)^2 + sen x_3 + 1.06 = 0\,3 \quad \text{con } x^{(0)}=(0.1,0.1,-0.1)^t$$

$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

*F(**x**)=(f1(**x**),f2(**x**),f3(**x**))*

La matriz jacobiana J(x) para este sistema es

$$J(x_1, x_2, x_3) = \begin{bmatrix} 3 & x_3 sen x_2 x_3 & x_2 sen x_2 x_3 \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

$$\begin{bmatrix} x_1^k \\ x_2^k \\ x_3^k \end{bmatrix} = \begin{bmatrix} x_1^{k-1} \\ x_2^{k-1} \\ x_3^{k-1} \end{bmatrix} + \begin{bmatrix} y_1^{k-1} \\ y_2^{k-1} \\ y_3^{k-1} \end{bmatrix}$$

Con

$$\begin{bmatrix} y_1^{k-1} \\ y_2^{k-1} \\ y_3^{k-1} \end{bmatrix} = -(J(x_1^{k-1}, x_2^{k-1}, x_3^{k-1}))^{-1} F(x_1^{k-1}, x_2^{k-1}, x_3^{k-1})$$

Haciendo necesario resolver en el paso *k* el sistema lineal $(J(\vec{x}^{k-1})(\vec{y}^{k-1}) = -F(\vec{x}^{k-1})$ con:

$$J(\vec{x}^k)^1 = \begin{bmatrix} 3 & x_3{}^{k-1} sen x_2{}^{k-1} x_3{}^{k-1} & x_2{}^{k-1} sen x_2{}^{k-1} x_3{}^{k-1} \\ 2x_1{}^{k-1} & -162(x_2{}^{k-1} + 0.1) & \cos x_3{}^{k-1} \\ -x_2{}^{k-1} e^{-x_1{}^{k-1} x_2{}^{k-1}} & -x_1{}^{k-1} e^{-x_1{}^{k-1} x_2 k-1} & 20 \end{bmatrix}$$

$$\vec{y}^{k-1} = \begin{bmatrix} y_1^{k-1} \\ y_2^{k-1} \\ y_3^{k-1} \end{bmatrix}$$

$$F(\vec{x}^k)^1 = \begin{bmatrix} 3x_1{}^{k-1} - \cos x_2{}^{k-1} x_3{}^{k-1} - 1/2 \\ (x_1{}^{k-1})^2 - 81(x_2{}^{k-1} + 0.1)^2 + sen x_3{}^{k-1} + 1.06 \\ e^{-x_1{}^{k-1} x_2 k-1} + 20x_3{}^{k-1} + \frac{10\pi - 3}{3} \end{bmatrix}$$

Efectuando el proceso iterativo se obtiene

k	x_1^k	x_2^k	x_3^k	$\|\|x^k-x^{k-1}\|\|_\infty$
0	0.10000000	0.10000000	-0.10000000	
1	0.50003702	0.01946686	-0.52152047	0.422
2	0.50004593	0.00155859	-0.52355711	$1.79\text{x}10^{-2}$
3	0.50000034	0.00001244	-0.52359845	$1.58\text{x}10^{-3}$
4	0.50000000	0.00000000	-0.52359877	$1.24\text{x}10^{-5}$
5	0.50000000	0.00000000	-0.52359877	0

8.2. TÉCNICAS CUASI-NEWTON

Dada la complejidad de calcular matrices jacobianas y resolver sistemas lineales nxn que ellas involucran, se vuelve tedioso el método de Newton.
Se presenta una generalización del método de la secante para sistemas no lineales (técnicas de Broyden), que requiere menos evaluaciones de funciones y cálculos algebraicos.
Reemplazan la matriz jacobiana (del método de Newton) por una matriz de aproximación que se ajusta en cada repetición, aunque su convergencia es superlineal y no tienen instancias de autocorrección.
Sea disponer la aproximación inicial $\vec{x}^{(0)}$ a la solución $\vec{q}$ de $G(\vec{x})=0$, se calcula $\vec{x}^{(1)}$ por el jacobiano $J(\vec{x}^{(0)})$ o la ecuación en diferencias:

$$\frac{\partial g_j}{\partial x_k}(\vec{x}^{(i)}) = \frac{g_j(\vec{x}^{(i)}+\vec{u}_k h)-g_j(\vec{x}^{(i)})}{h} \qquad (8.9)$$

$\vec{u}_k$ vector con un solo elemento no nulo en la coordenada k.

Para $\vec{x}^{(2)}$ se parte del método de Newton y se observa el método de la secante para una ecuación no lineal, aproximando:

$g'(\vec{x}_1) = \frac{g(\vec{x}_1)-g(\vec{x}_0)}{\vec{x}_1-\vec{x}_0}$ en vez de $g'(x)$ del método de Newton.

En sistemas no lineales, $\vec{x}^{(1)}-\vec{x}^{(0)}$ es un vector o sea que el cociente no esta definido, pero reemplazando $J(\vec{x}^{(1)})$ por una A_i que cumpla: $A_i(\vec{x}^{(1)}-\vec{x}^{(0)}) = G(\vec{x}^{(1)})-G(\vec{x}^{(0)})$ (8.10)

Como no se conoce como varía G en dirección ortogonal a $\vec{x}^{(1)}-\vec{x}^{(0)}$, A_i debe cumplir además

$A_i\vec{z} = J(\vec{x}^{(1)})\vec{z}$ (8.11)

Si $(\vec{x}^{(1)}-\vec{x}^{(0)})^t\vec{z}=0$

Cuyo significado es que cualquier vector ortogonal a $\vec{x}^{(1)}-\vec{x}^{(0)}$ no se ve influido por la actualización de $J(\vec{x}^{(1)})$ empleada para hallar $\vec{x}^{(2)}$.

Ahora 8.10 y 8.11 definen una unicidad a A_1, a través de:

$$A_1 = A_1 = J(\vec{x}^{(0)}) + \frac{\left[\vec{G}(\vec{x}^{(1)})-\vec{G}(\vec{x}^{(1)})-J(\vec{x}^{(1)})\left(\vec{x}^{(1)}-\vec{x}^{(0)}\right)\right]\left(\vec{x}^{(1)}-\vec{x}^{(0)}\right)^t}{\left\|\vec{x}^{(1)}-\vec{x}^{(0)}\right\|_e^z}$$

A_1 reemplaza a $J(\vec{x}^{(1)})$ para hallar $\vec{x}^{(2)}: \vec{x}^{(2)} = \vec{x}^{(1)} - A_1 G(\vec{x}^{(1)})$, repitiéndose para $\vec{x}^{(3)}$, usando A_1 en vez de $J(\vec{x}^{(0)})$ y $\vec{x}^{(2)}$ y $\vec{x}^{(1)}$ en vez de $\vec{x}^{(1)}$ y $\vec{x}^{(0)}$.

O sea $A_i = A_{i-1} + \frac{\vec{y}_i - A_{i-1}S_i}{\left\|\vec{S}_i\right\|_e^2} S_i$ (8.12)

Y $\vec{x}^{(i+1)} = \vec{x}^{(i)} - A_i^{-1}G(\vec{x}^{(i)})$ (8.13)

Con $\vec{y}_i = G(\vec{x}^{(i)}) - G(\vec{x}^{(i-1)}); \qquad S_i = \vec{x}^{(i)} - \vec{x}^{(i-1)}$

Ejemplo 8.2- resolver por Broyden el mismo problema

Reemplazar el jacobiano por una matriz de aproximación que se actualiza en cada paso

$$F(\vec{x}^0) = \begin{bmatrix} -1.199950 \\ -2.269833 \\ 8.462025 \end{bmatrix}$$

Tomando

$$A_0 = J(\vec{x}^0) = \begin{bmatrix} 3 & 9.9998333x10^{-4} & -9.9998333x10^{-4} \\ 0.2 & -32.4 & 0.9950042 \\ 9.900498x10^{-2} & -9.900498x10^{-2} & 20 \end{bmatrix}$$

Hallando su inversa (eliminación gaussiana)

$$A_0^{-1} = J(\vec{x}^0)^{-1} = \begin{bmatrix} 0.3333332 & 1.023852x10^{-5} & 1.615701x10^{-5} \\ 2.108607x10^{-3} & -3.086883x10^{-2} & 1.535836.10^{-3} \\ 1.660520x10^{-3} & -1.527577x10^{-4} & 5.000768x10^{-2} \end{bmatrix}$$

Actualizando

$$\vec{x}^1 = \vec{x} - A_0^{-1}F(\vec{x}^0) = \begin{bmatrix} 0.4998697 \\ 1.946685x10^{-2} \\ -0.5215205 \end{bmatrix} \text{ entonces}$$

$$F(\vec{x}^1) = \begin{bmatrix} -3.3944465x10^{-4} \\ -0.3443879 \\ 3.188238x10^{-2} \end{bmatrix} \quad \vec{y}_1 = F(\vec{x}^1) - F(\vec{x}^0) = \begin{bmatrix} 1.199611 \\ 1.925445 \\ -8.4300143 \end{bmatrix}$$

$$s_1 = \begin{bmatrix} 0.3998697 \\ -8.053315x10^{-2} \\ -0.4215204 \end{bmatrix}$$

$$\vec{s}_1{}^t A_0^{-1}\vec{y}_i = 0.3424604$$

$$A_1^{-1} = A_0^{-1} + (1/0.3424604)\left[(\vec{s}_1 - A_1^{-1}\vec{y}_1)\vec{s}_1^t A_0^{-1}\right]$$

$$= \begin{bmatrix} 0.3333781 & 1.11050x10^{-5} & 8.967344x10^{-6} \\ -2.021270x10^{-3} & -3.094849x10^{-2} & 2.196906.10^{-3} \\ 1.022214x10^{-3} & -1.650709x10^{-4} & 5.010986x10^{-2} \end{bmatrix}$$

$$\vec{x}^2 = \vec{x}^1 - A_1^{-1}F(\vec{x}^1) = \begin{bmatrix} 0.499863 \\ 8.737833x10^{-3} \\ -0.5231746 \end{bmatrix}$$

Continuando las iteraciones, descriptas en la tabla

k	x_1^k	x_2^k	x_3^k	$\|\|x^k - x^{k-1}\|\|_\infty$
3	0.5000066	$8.672157x10^{-4}$	-0.5236918	$7.88x10^{-3}$
4	0.5000003	$6.083352x10^{-2}$	-0.5235954	$8.12x10^{-4}$
5	0.5000000	$-1.448889x10^{-6}$	-0.5235989	$6.24x10^{-5}$
6	0.5000000	$6.059030x10^{-9}$	-0.5235988	$1.50x10^{-6}$

8.3. TÉCNICAS DE MAYOR PENDIENTE

Los métodos de Newton o cuasi-Newton tienen la dificultad de necesitar $\vec{x}^{(0)}$ preciso para garantizar convergencia, instancia que se supera con el método de mayor pendiente aunque presente convergencia lineal, lo que permitirá usarlo previamente a las técnicas de Newton. Este método busca un mínimo local para $g : R^n \to R$, mediante:

i. Evaluar g en una aproximación inicial $\vec{x}^{(0)} = (x_1^0, x_2^0, ..., x_n^0)^t$

ii. Hallar la dirección a partir de $\vec{x}^{(0)}$ que genere un descenso en g.

iii. Ver cuanto se debe mover en esa dirección, obteniendo $\vec{x}^{(1)}$.

iv. Iterar i a iii sustituyendo $\vec{x}^{(0)}$ por $\vec{x}^{(1)}$.

La dirección del mayor descenso de g en $\vec{x}$ esta dada por $-\vec{\nabla} g(\vec{x})$, entonces para $\vec{x}^{(1)}$ será:

$$\vec{x}^{(1)} = \vec{x}^{(0)} - a\vec{\nabla} g(\vec{x}^{(0)}),\ a\rangle 0 \qquad (8.14)$$

La clave del método radica en encontrar un valor de *a* que garantice que $g(\vec{x})$ se aproxime a cero, o que $g(\vec{x}^{(1)})$ □ $g(\vec{x}^{(0)})$.

Si al segundo miembro de 8.14 se engloba como una función *t(a),* a través de un polinomio cuadrático *P* y tres números $a_1, a_2\ y\ a_3$, próximos a *t*.

Si el mínimo absoluto de *P* es *a* en el menor intervalo cerrado que contenga a $a_1, a_2\ y\ a_3$, usando *P(a)* para aproximar el mínimo valor de *t(a)*.

Con ese *a* se efectúa otra iteración para aproximar el mínimo de $\vec{x}^{(1)} = \vec{x}^{(0)} - a\vec{\nabla} g(\vec{x}^{(0)})$,

Como $g(\vec{x}^{(0)})$ es conocido, se toma $a_1 = 0$ luego se halla un a_3 tal que $t(a_3)\ \langle t(a_1)$ y se escoge $a_2 = \frac{a_3}{2}$. El mínimo *a* de *P* en $[a_1, a_3]$ estará en el único punto crítico de *P* o a la derecha de a_3 si $P(a_3) = t(a_3) \langle t(a_1) = P(a_1)$ (no es difícil hallar un punto crítico para un polinomio de segundo grado).

Es decir, se parte de $\overrightarrow{a_1} = 0$ y $\overrightarrow{a_3} = 1$. Si $t(a_3) \geq t(a_1)$ se divide sucesivamente por dos y se reasigna el valor de $\overrightarrow{a_3}$ hasta que $t(a_3)\ \langle t(a_1)$ y $\overrightarrow{a_3} = 2^k$ para algún k.

La función g a usar es $\sum_{j=1}^{n} f_i^2$, con f_i los componentes de *F* de R^n en *R*.

8.4. PROBLEMA DE VALOR DE FRONTERA PARA ECUACIONES DIFERENCIALES ORDINARIAS Y EN DERIVADAS PARCIALES

Situaciones físicas dependientes del espacio se representan muchas veces por ecuaciones diferenciales con condiciones dadas en más de un punto.

Así para dos puntos y una ecuación de 2° orden, se tendrá

$$y'' = f(x, y, y') \qquad en \qquad [a, b]$$

Sujeta a $y(a) = \alpha \wedge y(b) = \beta$ (8.15)

8.5. EXISTENCIA Y UNICIDAD DE LA SOLUCIÓN PARA 8.15

Si en $y'' = f(x, y, y')$ *en* $[a,b]$

$$y(a) = \alpha$$
$$y(b) = \beta \qquad (8.16)$$

f es continua en el conjunto $D = \{(x, y, y') / a \le x \le b, -\infty < y < \infty, -\infty < y' < \infty\}$

con $\partial f / \partial y$ y $\partial f / \partial y'$, continuas en D

si además

a) $\dfrac{\partial f}{\partial y}(x, y, y') > 0 \quad \forall (x, y, y') \in D$

b) existe una C cota $\left|\dfrac{\partial f}{\partial y'}(x, y, y')\right| < c \qquad \forall (x, y, y') \in D$

el problema 8.16 tiene solución única.

8.5.1 Problema lineal

Si el problema de frontera lineal

$y'' = p(x)y' + q(x)y + r(x)$ *en* $[a,b]$ (8.17)

Con $y(a) = \alpha$

$y(b) = \beta$

Satisface

1. $p(x), q(x), y\, r(x)$ continuas en $[a,b]$
2. $q(x) > 0$ en $[a,b]$

El problema 8.17 tiene solución única

8.6. TÉCNICA DE DISPARO

$$y'' = p(x)y' + q(x)y + r(x) \quad \text{en} \quad [a,b] \qquad (8.18)$$

Con $y(a) = \alpha$

$y(b) = \beta$

E

$$y'' = p(x)y' + q(x)y + r(x) \quad \text{en} \quad [a,b] \qquad (8.19)$$
$$y(a) = 0$$
$$y'(a) = 1$$

Sea $y_1(x)$ solución de (8.18) e $y_2(x)$ la solución de (8.19), se puede plantear que

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x) \qquad (8.20)$$

Para $y_2(b) \neq 0$ será la solución única de los P.V.F.

El método de disparo para expresiones lineales reemplaza el problema lineal de valores de frontera por dos problemas de valores iniciales (P.V.I), dados por (8.18) y (8.19)

Los $y_1(x)$ e $y_2(x)$ se pueden aproximar por algunas técnicas vistas para los P.V.I y con 8.20 se aproxima la solución del problema de valor de frontera.

Ejemplo 8.3 Resolver el siguiente PVF

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{sen(\ln x)}{x^2},\ \ 1 \le x \le 2,\ y(1) = 1, y(2) = 2$$

Presentando los dos PVI

$$y_1'' = -\frac{2}{x}y'_1 + \frac{2}{x^2}y_1 + \frac{sen(\ln x)}{x^2},\ \ 1 \le x \le 2,\ y_1(1) = 1, y'_1(1) = 0$$

$$y_2'' = -\frac{2}{x}y'_2 + \frac{2}{x^2}y_2 + \frac{sen(\ln x)}{x^2},\ \ 1 \le x \le 2,\ y_2(1) = 0, y_2(1) = 1$$

Llamando $u_{1,i}$ a la aproximación de $y_1(x_i)$, $v_{1,i}$ a la de $y_2(x_i)$, se denomina w_i a la aproximación de $y(x_i) = y_1(x_i) + \frac{2 - y_1(2)}{y_2(2)} y_2(x_i)$

La sn exacta es: **$y=1.1392070132x-0.03920701320/x^2-0.3sen(lnx)-0.1cos(lnx)$**

Representando los resultados en la tabla, para 10 nodos igualmente espaciados

x_i	$u_{1,i}$	$v_{1,}$	w_i	$y(x_i)$	$\lvert y(x_i) - w_i \rvert$
1.0	1.00000000	0.00000000	1.00000000	1.00000000	
1.1	1.00896058	0.09117986	1.09262917	1.0926930	$1.43\text{x}10^{-7}$
1.2	1.03245472	0.16851175	1.18708471	1.18708484	$1.34\text{x}10^{-7}$
1.3	1.06674375	0.23608704	1,28338227	1.28338236	$9.78\text{x}10^{-8}$
1.4	1.10928795	0.29659067	1.38144589	1.38144595	$6.02\text{x}10^{-8}$
1.5	1.15830000	0.35184379	1,48115939	1.48115942	$3.06\text{x}10^{-8}$
1.6	1.21248372	0.40311695	1.58239245	1.58239246	$1.08\text{x}10^{-8}$
1.7	1.27087454	0.45131840	1.68501396	1.68501396	$5.43\text{x}10^{-10}$
1.8	1.33273851	0.49711137	1.78889854	1.78889853	$5.05\text{x}10^{-9}$
1.9	1.39750618	0.54098928	1.89392951	1.89392951	$4.41\text{x}10^{-9}$
2.0	1.46472815	0.58332538	2.00000000	2.00000000	

<u>Situación no lineal</u>:

y''=f(x,y,y'), a≤x≤b, y(a)=, y'(a)=t

donde se eligen los parámetros $t=t_k/lim_{\infty}y(b.t_k)=y_b=\beta$

la solución en *x* como en *t*, requerirá la forma de los PVI como:

- *y''(x,t)=f(x,y(x,t),y'(x,t)), a≤x≤b, y(a,t)=, y'(a,t)=t* y
- *z''(x,t)=(∂f/∂y)(x,y,y')z(x,t)+(∂f/∂y')(x,y,y')z'(x,t), a≤x≤b, z(a,t)=0, z'(a,t)=1*

simbolizando *z(x,t)* a *(∂y/∂t)(x,t)*

entonces en cada iteración se deberá resolver ambos PVI, con

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{z(b, t_{k-1})} \qquad (8.21)$$

Ejemplo 8.4

Sea $y''' = \frac{1}{8}\left(32 + 2x^3 yy'\right),\ \ 1 \le x \le 3,\ y(1) = 17, y(3) = 43/3$

Para utilizar el método del disparo, se plantean los PVI

$y''' = \frac{1}{8}\left(32 + 2x^3 yy'\right),\ \ 1 \le x \le 3,\ y(1) = 17, y'(1) = t_k$ y

$z'' = \frac{\partial f}{\partial y} z + \frac{\partial f}{\partial y'} z' = \frac{1}{8}\left(y'z + yz'\right),\ \ 1 \le x \le 3,\ \mathrm{z}(1) = 0, z'(1) = 1$ que deben aproximarse en cada paso de la iteración, la sn exacta:**$y(x)=x^2+16/x$**

x_i	$w_{1,i}$	$y(x_i)$	$\lvert w_{1,i} - y(x_i) \rvert$
1.00000000	17.000000	17.000000	
1.1	15.755495	15.755455	$4.06\text{x}10^{-5}$
1.2	14.773389	14.773333	$5.60\text{x}10^{-5}$
1.3	13.997752	13.997692	$5.94\text{x}10^{-5}$
1.4	13.388629	13.388571	$5.71\text{x}10^{-5}$
1.5	12.919719	12.916667	$5.23\text{x}10^{-5}$
1.6	12.560046	12.560000	$4.64\text{x}10^{-5}$
1.7	12.301805	12.301765	$4.02\text{x}10^{-5}$
1.8	12.128923	12.128889	$3.14\text{x}10^{-5}$
1.9	12.031081	12.031053	$2.84\text{x}10^{-5}$
2.0	12.000023	12.000000	$2.32\text{x}10^{-5}$
2.1	12.029066	12.029048	$1.84\text{x}10^{-5}$
2.2	12.112741	12.112727	$1.40\text{x}10^{-5}$
2.3	12.246532	12.246522	$1.01\text{x}10^{-5}$
2.4	12.426673	12.426667	$6.68\text{x}10^{-6}$
2.5	12.650004	12.650000	$3.61\text{x}10^{-6}$
2.6	12.913847	12.913847	$9.17\text{x}10^{-7}$
2.7	13.215924	13.215926	$1.43\text{x}10^{-6}$
2.8	13.554282	13.554286	$3.46\text{x}10^{-6}$
2.9	13.927236	13.927241	$5.21\text{x}10^{-6}$
3.0	14.333327	14.333333	$6.69\text{x}10^{-6}$

8.7. EJERCITACION UNIDAD 8 EMPLEANDO MATLAB

I) Dado el PVF x''(t)=(2t/1+t^2)x'(t)- (2/1+t^2)x(t)+1
Con x(0)=1.25 y x(4)= -0.95 en [0,4]
Se usa un RK 4 para para resolver los PVI asociados(será un sistema de EDO).
Entonces crear un archivo.m, ej. F para guardar el PVI

```
function Z=F1(t,Z)
x=Z(1);y=Z(2);
Z=[y,2*t*y/(1+t^2)-2*x/(1+t^2)+1];
y
function Z=F2(t,Z)
x=Z(1);y=Z(2);
Z=[y,2*t*y/(1+t^2)-2*x/(1+t^2)];
```

Entonces se corre

```
>> linsht(F1,F2,a,b,alpha,beta,M)
a=0,b=4;alpha=1.25,beta=-0.95,M020 para h=0.2
```

tomando el ejemplo
II)x''=2x'-x+t^2-1 en[0,1],con x(0)=5 y x(1)=10; se emplea el shoot(lineal)
(la sn exacta es t^2+4t+5)
En el prompt de Matlab se tipea

```
>> shoot
```

Aparece la interface para

Scheme: euler, euler mejorado,RK(de tercer y cuarto orden) y los cuadros para ingresar p=2,q=-1,r=x^2-1;a=0;b=1;y(a)=5;y(b)=10;h=0.1;y'(a)=5

Se elige plot y mostrar resultados
>>

s	Y(b)	y(b)	Error
00005.000000	00008.791688	00010.000000	-0001.208312

III) por diferencias finitas
para la ecuación general de orden dos.x''=p(t)x'(t)+q(t)x(t)+r(t) en [a,b] con alpha=x(a), beta=x(b) y N el número de etapas, se emplea
findiff(p,q,r,a,b,alpha,beta,N)
para el caso PVF $x''(t)=(2t/1+t^2)x'(t)- (2/1+t^2)x(t)+1$
Con x(0)=1.25 y x(4)= -0.95 en [0,4]
Se tendrá:

```
>> p=@(x)(2*x/(1+x^2));
>> q=@(x)(-2/(1+x^2));
>>r=@ (x)0*x+1;
>> findiff(p,q,r,0,4,1.25,-0.95,20)
```

EJERCICIOS PROPUESTOS PARA UNIDAD 8

1) Dados los sistemas de ecuaciones diferenciales de primer orden

a) $\begin{array}{l} u_1' = 3u_1 + 2u_2 - (2t^2+1)e^{2t} \quad 0 \le t \le 1 \quad u_1(0) = 1; \\ u_2' = 4u_1 + u_2 + (t^2+2t-4)e^{2t} \quad 0 \le t \le 1 \quad u_2(0) = 1; \\ h = 0.2 \end{array}$

sn exacta: $u_1(t)=(1/3)e^{3t}-(1/3)e^{-t}+e^{2t} \quad u_2(t)=(1/3)^{e5t}+(2/3)e^{-t}+t^2e^{2t}$

b) $\begin{array}{l} u_1' = -4u_1 - 2u_2 + \cos t + 4sent \quad 0 \le t \le 2 \quad u_1(0) = 0; \\ u_2' = 3u_1 + u_2 \ - 3sent \quad 0 \le t \le 2 \quad u_2(0) = -1; \\ h = 0.1 \end{array}$

sn exacta: $u_1(t)=2e^{-t}-2e^{-2t}+sent \quad u_2(t)=-3e^{-t}+2e^{-2t}$

2) Sea la ecuación diferencial

$y``-2y'+4y=te^t-t \quad en \quad 0 \le t \le 1 \quad con \ y(0)=y'(0)=0 \quad h=0.1$

resuélvela empleando Runge Kutta para sistemas

Sn exacta: $(1/6)t^3e^t-te^t+2e^t-t-2$

3) Empleando la técnica del disparo lineal aproxime la solución de los siguientes problemas de borde

a) $y''=-3y'+2y+2x+3$ en $0 \le x \le 1$ $y(0)=2, y(1)=1, h=0.1$

b) $y''=-(4/x)y'+(2/x)y-(2x^2)\ln x$ en $1 \le x \le 2$, $y(1)=-0.5, y(2)=\ln 2$, $h=0.05$

c) $y''=-(x+1)y'+2y+(1-x^2)e^{-x}$ en $0 \le x \le 1$ $y(0)=-1, y(1)=0, h=0.1$

4) Usando el algoritmo de diferencias finitas aproxime la solución de los siguientes problemas de frontera

a) $y''=-3y'+2y+2x+3$ en $0 \le x \le 1$ $y(0)=2, y(1)=1, h=0.1$

b) $y''=-(4/x)y'+(2/x)y-(2x^2)\ln x$ en $1 \le x \le 2$, $y(1)=-0.5, y(2)=\ln 2$, $h=0.05$

c) $y''=-(x+1)y'+2y+(1-x^2)e^{-x}$ en $0 \le x \le 1$ $y(0)=-1, y(1)=0, h=0.1$

UNIDAD 9: ECUACIONES EN DERIVADAS PARCIALES

9.1. ECUACIONES EN DERIVADAS PARCIALES

Cuando el problema físico a resolver involucra más de una variable, la expresión adecuada a través de derivadas parciales.
Los métodos de aproximación analítica a la solución de EDP, proporcionan frecuentemente información útil acerca del comportamiento de la solución en valores críticos de la variable dependiente, pero tienden a ser más difíciles de aplicar que los métodos numéricos. Entre las consideraciones que justifican el uso de métodos numéricos para solucionar ciertos tipos de ecuaciones diferenciales ordinarias y en derivadas parciales se encuentran: 1) Los datos de los problemas reales presentan siempre errores de medición, y el trabajo aritmético para la solución está limitado a un número finito de cifras significativas que resultan en errores de redondeo. Por lo tanto, incluso los métodos analíticos proporcionan resultados que son aproximaciones numéricas; 2) La evaluación numérica de las soluciones analíticas es a menudo una tarea laboriosa y computacionalmente ineficiente, mientras que los métodos numéricos generalmente proporcionan soluciones numéricas adecuadas, de manera más simple y eficiente. De los métodos de aproximación numérica disponibles para resolver ecuaciones diferenciales, los más utilizados son el método de diferencias finitas y el método de elementos finitos

Se considerarán los casos reducidos a dos variables.
Básicamente se presentarán tres formulaciones típicas, como ser: la ecuación diferencial parcial elíptica, hiperbólica y principalmente la parabólica (cuyos nombres provienen de la resolvente cuadrática) y las aproximaciones por diferencias finitas.

9.2. ECUACIÓN GENERAL

$$\left[Af_{xx} + 2Bf_{xy} + Cf_{yy}\right] + Df_x + Ef_y + Ff = G \tag{9.1}$$

con $A, B, ..., G$ funciones reales de x e y, en base a su analogía con el discriminante de secciones cónicas:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{9.2}$$

se las describirá como hiperbólicas, elípticas y parabólicas.
la clasificación de una EDP está ligada a las *caracteristicas* de la EDP.
caracteristicas: son hipersuperficies (n-1) dimensionales en hiperespacios n dimensionales (n= n° de variables independientes).
Híper: espacios que pueden ser de más de tres dimensiones, y curvas y superficies en esos espacios.
En n=2, las *caracteristicas* son líneas curvas en el dominio solución, a lo largo de las cuales las señales o información propaga.
Las *caracteristicas* pueden ser *R* o *C*, la presencia o ausencia de *caracteristicas* tiene un impacto significativo sobre la solución de la EDP (analítica o numérica).
Tomando $Af_{xx} + Bf_{xy} + Cf_{yy} + Df_x + Ef_y + Ff = G$ (9.3)
Aplicando la regla de la cadena para hallar las derivadas totales de f_x, f_y:

$d(f_x) = f_{xx}dx + f_{xy}dy$ (9.4a)
$d(f_y) = f_{yx}dx + f_{yy}dx$ (9.4b) poniendo (9.4a) y (9.4b) como matrices

$$\begin{bmatrix} A & B & C \\ dx & dy & 0 \\ 0 & dx & dy \end{bmatrix} \begin{bmatrix} f_{xx} \\ f_{xy} \\ f_{yy} \end{bmatrix} = \begin{bmatrix} -Df_x - Ef_y - F + G \\ df(x) \\ df(y) \end{bmatrix} \tag{9.5}$$

Si el determinante de coeficientes es nulo $\Rightarrow$ las derivadas segundas de $f(x,y)$ son ∞ (sin sentido físico) o indeterminadas.

Del determinante nulo $\Rightarrow A(dy)^2 + B(dx)(dy) + C(dx)^2 = 0$ (9.6)

Esta es la ecuación *caracteristica* de (9.3), que por la fórmula cuadrática dará:

$$dy/dx = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} \tag{9.7}$$

ecuación diferencial para dos familias de curvas en el plano $xy(\pm)$.

A lo largo de estas dos familias de curvas, las derivadas segundas de $f(x,y)$ pueden ser multivalores o discontinuas.

Estas dos familias de curvas, si existen, son las *curvas caracteristicas* de (9.3), pueden ser *R*, *C* (diferentes o repetidas), según $B^2 - 4AC$:

$B^2 - 4AC = 0 \Rightarrow$ reales y repetidas (parabólicas)

Para $B^2 - 4AC > 0 \Rightarrow$ reales y distintas (hiperbólicas)

$B^2 - 4AC < 0 \Rightarrow$ complejas (elípticas)

9.2.1. Significado físico (Clasificación)

Cada uno de ellos tiene rasgos propios, su forma particular de EPP gobernante y su solución numérica.

Problemas de equilibrio: problemas de estado estacionario en dominios cerrados $D(x,y)$, donde la solución $f(x,y)$ es gobernada por una EDP elíptica, sujeta a CF especificadas en cada punto de la frontera *B* del dominio.

Ejemplo: conducción de calor en estado estacionario en sólido: $\nabla^2 T = 0(Laplace)$, T representa la temperatura n este caso

Problema de valor propio: donde la solución existe sólo para valores especiales (λ) de un parámetro del problema.

Problemas de propagación: problemas de VI en dominios abiertos (respecto a una variable) donde $f(x,t)$ en $D(x,t)$ arranca del estado inicial guiado y modificado por CF. Son parabólicas o hiperbólicas. La mayoría de los de propagación son de estado no estacionario.

Los valores de T deben especificarse a lo largo de $T(x,t) = F(x)$ y en ambas fronteras $(f(t), g(t))$

Tabla resumen:

	Elíptica	Parabólica	Hiperbólica
Problema fís.	Equilibrio	propagación	propagación
Características	Compleja	Real repetida	Real distinta
Velocidad de Propag.señal	Indefinida	infinita	finita
Dominio de dependencia	Cerrado	abierto	Abierto .
Met.Numérico	Relajación	Marching	Marching

9.2.2. Condiciones de frontera y valores iniciales

Para problemas de propagación estacionarios o no estacionarios, las condiciones auxiliares consisten en una CI (o condiciones iniciales) a lo largo de la frontera tiempo, y CF (borde o frontera)sobre las fronteras físicas del dominio de solución.

Condiciones auxiliares no pueden aplicarse sobre la frontera abierta en la dirección del tiempo.

Una CI a lo largo de la frontera tiempo:

$f(x,y,z,o) = F(x,y,z)$ sobre la frontera tiempo

Las CF sobre las fronteras físicas pueden ser:

-tipo Dirichlet: se especifica el valor de la función f .

-condición de frontera de Neumann: el valor de la derivada parcial normal en la frontera, se especifica: $\partial f / \partial n$.

-condición mixta: $af + \partial f / \partial n$.(o de Robin)

Tener en cuenta que en diferentes porciones de la frontera pueden especificarse diferentes tipos de CF.

Según **Hadamard**: un problema físico está bien planteado si su solución existe, es unica y depende continuamente de los datos de frontera y (para los de propagación) de los datos iniciales.

Así, por ej., para parabólicas:

- $D(x,t) =$ abierta en la dirección del tiempo.
- Datos iniciales deben especificarse en la frontera tiempo.

CF continuas deben especificarse en las fronteras físicas
Lógicamente, estas EDP pueden resolverse analíticamente, tomando el caso de las parabólicas se podría plantear la separación de variables de posición y tiempo, y aplicando el principio de superposición presentar la solución como una serie expandidadel producto de la solución temporal y de posición:

$$T(x,\ t) = \sum_{k=1}^{\infty} \alpha_k \exp[-\lambda_k t] M_k(x)$$

Pero, el objetivo es buscar un método numérico de aproximación de la solución exacta**.**
Se formularán las EDP clásicas en base a su aproximación por diferencias finitas

9.3. ELÍPTICA O DE POISSON

$$\frac{\partial^2 f}{\partial x^2}(x,y) + \frac{\partial^2 f}{\partial y^2}(x,y) = g(x,y) \qquad (9.8)$$

Por ejemplo para la distribución estacionaria de calor en una región plana, régimen estacionario de mecánica de fluidos, etc.
Si $g(x,y) \equiv 0$ se obtiene la ecuación de Laplace.
Si la función representa un campo de temperaturas, las restricciones del problema a través de la distribución de temperatura en el borde de la región, son las condiciones de frontera de Dirichlet, dadas por:
$f(x,y) = r(x,y)$ $\forall (x,y)$ en S, frontera de la región U.

$U = \{(x,y) / a\langle x\langle b, c\langle y\rangle d\}$

Se eligen $n\, y\, m \in Z$ tamaño paso h y k, con $h = \frac{b-a}{n}$, $k = \frac{d-c}{m}$ con lo cual el intervalo $[a,b]$ se compondrá de n partes de ancho h y $[c,d]$ de m partes de ancho k y se genera una malla (x_i, y_i) con

$x_i = a + ih \qquad i = 0,1,...,n$

$y_i = c + jk \qquad j = 0,1,...,n$

Los cortes x_i e y_i son los puntos de la malla, para los cuales se usa la fórmula de Taylor en x alrededor de x_i para diferenciales centrados.

$$\frac{\partial^2 f}{\partial x^2}(x_i, y_i) = \frac{f(x_{i+1}, y_j) - 2f(x_i, y_j) + f(x_{i-1}, y_j)}{h^2} - \frac{h^2}{12}\frac{\partial^4 f}{\partial x^4}(\zeta_i, y_j) \qquad (9.9)$$

Con $\zeta_i \in (x_{i-1}, x_{i+1})$

Y para y en derredor de y_j

$$\frac{\partial^2 f}{\partial y^2}(x_i, y_i) = \frac{f(x_i, y_{j+1}) - 2f(x_i, y_j) + f(x_{i-1}, y_{j-1})}{k^2} - \frac{h^2}{12}\frac{\partial^4 f}{\partial y^4}(x_i, \eta_j) \qquad (9.10)$$

Con $\eta_j \in (y_{j-1}, y_{j+1})$

Llevando 9.10 y 9.9 a 9.8

$$\frac{f(x_{i+1}, y_j) - 2f(x_i, y_j) + f(x_{i-1}, y_j)}{h^2} + \frac{f(x_i, y_{j+1}) - 2f(x_i, y_j) + f(x_{i-1}, y_{j-1})}{k^2} -$$

$$= g(x_i, y_j) + \frac{h^2}{12}\frac{\partial^4 f}{\partial x^4}(\zeta_i, y_j) + \frac{h^2}{12}\frac{\partial^4 f}{\partial y^4}(x_i, \eta_j) \qquad (9.11)$$

$i = 1,2,...(n-1)$ $j = 1,2,...(m-1)$

Para condiciones de frontera:

$f(x_0, y_j) = r(x_0, y_j) \quad j = 0,1,...,m$

$f(x_n, y_j) = r(x_n, y_j) \quad j = 0,1,...,m$

$f(x_i, y_0) = r(x_i, y_0) \quad i = 1,2,...,n-1$

$f(x_i, y_m) = r(x_i, y_m) \quad i = 1,2,...,n-1$

Para un error de truncado del orden $O(h^2 + k^2)$, se tendrá:

Con $\mu_{i,j}$ aproximando a $f(x_i, y_j)$

$$2\left[\left(\frac{h}{k}\right)^2 + 1\right]\mu_{i,j} - (\mu_{i+1,j} + \mu_{i-1,j}) - \left(\frac{h}{k}\right)^2(\mu_{i,j+1} + \mu_{i,j-1}) = -h^2 g(x_i, y_j) \qquad (9.12)$$

(diferencias centradas)

$i = 1,2,...,n-1 \qquad j = 0,1,...,m$

$\mu_{0,j} = r(x_0, y_j) \qquad j = 0,1,...,m$

$\mu_{n,j} = r(x_n, y_j) \qquad j = 0,1,...,m$

$\mu_{0,j} = r(x_i, y_0) \qquad i = 1,2,...,n-1$

$\mu_{0,j} = r(x_i, y_m) \qquad i = 1,2,...,n-1$

A través de 9.12 se aproxima a $f(x,y)$ en los puntos $(x_{i-1},y_j),(x_i,y_j),(x_{i+1},y_j),(x_i,y_{j-1})\,y\,(x_i,y_{j+1})$.

Con los datos de frontera se obtendrá un sistema lineal de (*n-1)(m-1*) por *(n-1)(m-1)* siendo $\mu_{i,j}$ las incógnitas

Ejemplo 9.1

Se desea hallar la distribución de calor en estado estacionario en una placa cuadrada de 0.5 por 0.5 de espesor despreciable; dos fronteras adyacentes se mantienen a 0ºC, en las otras dos fronteras existe un incremento lineal de 0 a 100ºC en la esquina de intersección.
Se está en presencia de un caso particular de la ecuación de Poisson, *f(x,y)=0* (Laplace)

Entonces. $\nabla^2 u(x,y) \equiv \dfrac{\partial^2 u}{\partial x^2}(x,y)+\dfrac{\partial^2 u}{\partial y^2}(x,y)=0$ en $0<x<0.5, 0<y<0.5$

CF: *u(0,y)=0, u(x,0)=0, u(x,0.5)=200x, u(0.5,y)=200y*
Para una red de *n=m=4*, la ecuación en diferencias quedará:

$4w_{ij}-\left(w_{i+1,j}+w_{i-1,j}\right)-\left(w_{i,j-1}+w_{i,j+1}\right)=0$

Para toda *i=1,2,3* y *j= 1,23*, se genera el sistema

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \end{bmatrix} = \begin{bmatrix} 25 \\ 50 \\ 150 \\ 0 \\ 0 \\ 50 \\ 0 \\ 0 \\ 25 \end{bmatrix}$$

Se ha empleado la notación $w_l=wij$, siendo *l=i+(m-1-j)(n-1)* para toda *i=1,2,…,n-1* y *j=1,2,…,m-1* (en lugar de doble indización)
Mediante una técnica numérica se resuelve el sistema matricial de ecuaciones, sujeto a las CF: $w_{1,0}=w_{2.0}=w_{3,0}=w_{0,1}=w_{0,2}=w_{0,3}=0$
$w_{1,4}=w_{4,1}=25$, $w_{2,4}=w_{4,2}=50$ y $w_{3,4}=w_{4,3}=75$

i	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
w_i	*18.75*	*37.50*	*56.25*	*12.50*	*25.00*	*37.50*	*6.25*	*12.50*	*18.75*

9.4. ECUACIONES PARABÓLICAS

Para la ecuación de **Calor**(parabólica)- unidimensional

$\dfrac{\partial u}{\partial t}=\alpha^2\dfrac{\partial^2 u}{\partial x^2}$ su fórmula en diferencias será

$$=\frac{u_{i,j+1}-u_{i,j}}{k}=\alpha^2\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2} \qquad (9.13)$$

donde *u* es la solución exacta a las ecuaciones aproximadas, $x_i = ih$, *(i =*

0, 1, 2,m − 1...), $t_i = jk$, (j = 0, 1, 2,...) y *m = l/h*. La ecuación (9.13) puede arreglarse:

$u_{i,j+1} = \lambda u_{i-1,j} + (1 - 2\lambda\_)u_{i,}j + u_{i+1,j}$ (9.14)

con $\lambda = \_^2(k/h^2)$.

Dado que la condición inicial *u(x, 0) = f(x)*, para todo $0 \le x \le l$, implica que $u_{i,0} = f(x_i)$, para toda *i = 0, 1, 2,...,m*, se pueden usar estos valores en la ecuación (9.14) para calcular el valor de $u_{i,1}$ para toda *i = 1, 2,...,m − 1*. Las condiciones iniciales *u(0, t) = 0* y *u(l, t) = 0* implican que $u_{0,1} = u_{m,1} = 0$ y, por tanto, se pueden determinar todos los elementos de la forma $u_{i,1}$. Ya conocidas todas las aproximaciones $u_{i,1}$ se pueden obtener,siguiendo un procedimiento semejante, los valores $u_{i,2}$, $u_{i,3}$,...

Haciendo $\boldsymbol{u}^{(0)} = (f(x_1), f(x_2),..., f(x_{m-1}))^t$ y $\boldsymbol{u}^{(j)} = (u_{1j}, u_{2j},..., u_{m-1,j})^t$,para todo *j = 1, 2,...*, se puede plantear matricialmente este método de solución como:

$\boldsymbol{u}^{(j)} = A\boldsymbol{u}^{(j-1)}$, para todo *j = 1, 2,...*donde *A* es la siguiente matriz tridiagonal

$$A = \begin{bmatrix} 1-2\lambda & \lambda & 0 & \dots & \dots & 0 \\ \lambda & 1-2\lambda & \lambda & \ddots & & \vdots \\ 0 & & & & & \vdots \\ \vdots & \ddots & & & & 0 \\ \vdots & & \ddots & \lambda & 1-2\lambda & \lambda \\ 0 & \dots & \dots & 0 & \lambda & 1-2\lambda \end{bmatrix}$$

$w^{(j)}$ se obtiene para $w^{(j-1)}$ por una multiplicación simple de matrices. A esto se le conoce con el nombre de método de diferencias progresivas, pero es inestable para $\lambda > 1/2$.

Empleando otra fórmula en diferencias, la regresiva para la derivada temporal

$$\frac{u_{i,j} - u_{i,j-1}}{k} = \alpha^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \qquad (9.15)$$

expresable como

$(1+2\lambda)\ u_{i,j} - \lambda u_{i+1,j} - \lambda\_)u_{i-1,j} = u_{i,j-1}$ n (9.16)

Como $w_{i,0} = f(x_i)$ para toda *i = 1, 2,...,m − 1* y $w_{m,j} = w_{0,j} = 0$ para toda *j = 1, 2,...*, este método de diferencias tiene la representación matricial

$$\begin{bmatrix} 1+2\lambda & -\lambda & 0 & \dots & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & \ddots & & \vdots \\ 0 & & & & & \vdots \\ \vdots & \ddots & & & & 0 \\ \vdots & & \ddots & -\lambda & 1+2\lambda & -\lambda \\ 0 & \dots & \dots & 0 & -\lambda & 1+2\lambda \end{bmatrix} \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{m-1,j} \end{bmatrix} = \begin{bmatrix} w_{1,j-1} \\ w_{2,j-1} \\ \vdots \\ w_{m-1,j-1} \end{bmatrix} \qquad (9.17)$$

O sea $A\boldsymbol{w}^{(j)} = \boldsymbol{w}^{(j+1)}$ para toda *j = 1, 2,*.

Indicando la necesidad de resolver un sistema lineal para hallar $w^{(j)}$ a partir de $w^{(j-1)}$, siendo estable para cualquier λ.

Dado que $\lambda > 0$, la matriz *A* es definida positiva y con dominancia diagonal estricta, tridiagonal, pudiendo resolverse por algoritmos ya vistos (caso Crout)

Para evitar la falta de precisión, generada por el error de truncado, se requiere que los intervalos de tiempo sean mucho más pequeños que los de espacio ($h >> k$), haciendo necesario un método que permita tomar valores similares para *h* y *k*, y estable para todo λ. Para ello se puede emplear:

- hasta el paso **j** en **t**: $= \frac{u_{i,j+1} - u_{i,j}}{k} = \alpha^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$

- regresiva en el **j**+1 en ***t***: $\frac{u_{i,j+1}-u_{i,j}}{k}=\alpha^2\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2}$

quedando (esquema de Crack- Nicolson)

$$\frac{u_{i,j+1}-u_{i,j}}{k}=\frac{\alpha^2}{2}\left(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{h^2}\right) \quad (9.18)$$

Arreglado como:

$$-(\lambda/2)u_{i-1,j+1}+(1+\lambda)\,u_{i,j+1}-(\lambda/2)u_{i+1,j+1}=(\lambda/2)u_{i-1,j}+(1-\lambda)\,u_{i,j}+(\lambda/2)u_{i+1,j} \quad (9.19)$$

Generando $A\mathbf{w}^{(j+1)}=B\mathbf{w}^{(j)}$ (9.20)

con

$$A=\begin{bmatrix} 1+\lambda & -\lambda/2 & 0 & \cdots & \cdots & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & \ddots & & \vdots \\ 0 & & & & & \vdots \\ \vdots & \ddots & & & & 0 \\ \vdots & & \ddots & -\lambda/2 & 1+\lambda & -\lambda/2 \\ 0 & \cdots & \cdots & 0 & -\lambda/2 & 1+\lambda \end{bmatrix}$$

$$B=\begin{bmatrix} 1-\lambda & -\lambda/2 & 0 & \cdots & \cdots & 0 \\ -\lambda/2 & 1-\lambda & -\lambda/2 & \ddots & & \vdots \\ 0 & & & & & \vdots \\ \vdots & \ddots & & & & 0 \\ \vdots & & \ddots & -\lambda/2 & 1-\lambda & -\lambda/2 \\ 0 & \cdots & \cdots & 0 & -\lambda/2 & 1-\lambda \end{bmatrix}$$

Para obtener $\mathbf{w}^{j+1}$ a partir de $\mathbf{w}^{j+1}$, se debe resolver el sistema planteado en (9.19).La matriz *A* es definida positiva, diagonal estrictamente dominante y tridiagonal. Para resolver este sistema, se puede usar la factorización LU de Crout para sistemas lineales tridiagonales.

Ejemplo 9.2:

Suponiendo que se quiere determinar la distribución de temperatura en cualquier instante *t* de una barra de cobre ($\kappa \approx 1$) con $l = 1$, que presenta una función de temperatura inicial dada por $f(x) = \sin(\pi x)$,tomar $m = 10, N = 100, T = 10$

Sn exacta: $u(x, t) = e^{-\pi^2 t}\sin(\pi x)$

Se presentan los valores para $t=0.25$

x_i	$u(x_i, 0{,}25) \times 10^{-2}$	Dif. Progresivas $u_{pi,500} \times 10^{-2}$	error relativo $\lvert u_p - u\rvert/u \times 10^{-2}$
0.0	0	0	
0.1	2.621	2.658	1.432
0.2	4.985	5.056	1.432
0.3	6.861	6.959	1.432
0.4	8.065	8.181	1.432
0.5	8.480	8.602	1.432
0.6	8.065	8.181	1.432
0.7	6.861	6.959	1.432
0.8	4.985	5.056	1.432
0.9	2.621	2.658	1.432
1.0	0	0	

Valores de la distribución de temperatura de la barra en t = 0,25,calculados por el método de sustitución progresiva(h = 0,1, k = 0,0005, λ = 0,05

x_i	$u(x_i, 0{,}25) \times 10^{-2}$	Dif. Regr. $u_{ri,25} \times 10^{-2}$	Error relativo $\lvert ur - u\rvert/u \times 10^{-2}$
0.0	0	0	
0.1	2.621	2.993	14.19
0.2	4.985	5.692	14.18
0.3	6.861	7.835	14.20
0.4	8.065	9.210	14.20
0.5	8.480	9.684	14.20
0.6	8.065	9.210	14.20
0.7	6.861	7.835	14.20
0.8	4.985	5.692	14.18
0.9	2.621	2.993	14.19
1.0	0	0	

Valores de la distribución de temperatura de la barra en t = 0,25 calculados por el método de sustitución regresiva (h = 0,1, k = 0,01, λ = 1)

xi	$u(x_i, 0{,}25) \times 10^{-2}$	dif.C-N $u_{ci,25} \times 10^{-2}$	eror relativo $\lvert uc - u\rvert/u \times 10^{-2}$
0.0	0	0	
0.1	2.621	2.669	1.844
0.2	4.985	5.077	1.844
0.3	6.861	6.987	1.844
0.4	8.065	8.214	1.844
0.5	8.480	8.637	1.844
0.6	8.065	8.214	1.844
0.7	6.861	6.987	1.844
0.8	4.985	5.077	1.844
0.9	2.621	2.669	1.844
1.0	0	0	

: Valores de la distribución de temperatura de la barra en t = 0,25 calculados por el método de Crank-Nicolson(h = 0,1, k = 0,01, λ= 1)

9.5. ECUACIÓN HIPERBÓLICA

Se presenta la ecuación hiperbólica a través de una situación problema

Sea la EDP $\dfrac{\partial^2 u}{\partial t^2} = 4\dfrac{\partial^2 u}{\partial x^2}$

con las condiciones de frontera

$u(0, t) = 0 = u(1, t),\ 0 < t < 1$

y con las condiciones iniciales

$u(x, 0) = sen(\pi x),\ 0 \le x \le 1$ y $u_t(x, 0) = 0,\ 0 \le x \le 1$

Sn exacta: $u(x, t) = sen(\pi x)cos(2\pi t)$

La división rectangular del dominio (x,t):: $0 \le x \le a,\ 0 \le t \le b$, denominando

$u(x, t) = u_{i,j}$

$u(x + h, t) = u_{i+1,j}$

$u(x - h, t)) = u_{i-1,j}$

$u(x, t + k)) = ui_{j+1}$

$u(x, t - k)) = u_{i,j-1}$

la ecuación en diferencias será

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = c^2\,\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} \qquad (9.21)$$

Haciendo $\sigma = ck/h$

$u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = \sigma^2(\,u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1})$

Colocando en términos de $u_{i,j+1}$ y reordenando, se tiene

$$u_{i,j+1} = (2-2\alpha^2)u_{i,j} + \alpha^2(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1} \quad (9.22)$$

La ecuación (9.22) es aplicable para $i = 2, 3,..., n-1$ y $j = 2, 3,...,m-1$, con el paso $(j+1)$ de tiempo requiriendo de los j-ésimo y los $(j-1)$ ésimo pasos; los valores del $(j-1)$ésimo paso vienen dados por las condiciones iniciales $u_{i,1} = f(x_i)$ pero, los valores del j ésimo paso no se suelen proporcionar, por lo que se usa $g(x)$ para conseguir los valores de este paso.
Para asegurar la estabilidad de este método es necesario que $\lambda = ck/h \leq 1$.

Se desarrolla el polinomio de Taylor de orden uno para $u(x, t)$ alrededor de $(x_i, 0)$.

$$u(x_i, k) = u(x_i, 0) + u_t(x_i, 0)k + O(k_2) \quad (9.23)$$

Se aplica el hecho de que $u(x_i, 0) = f(xi) = f_i$ y $u_t(x_i, 0) = g(x_i) = g_i$ en (8),para obtener aproximaciones numéricas en el j-ésimo paso (se sabe que $t_2 = k$)

$$u_{i,2} = f_i + kg_i \text{ para } i = 2, 3,..., n-1 \quad (9.24)$$

Ya que la fórmula (9.24) es una aproximación lineal a los valores del j ésimo paso, el error de truncado involucrado es importante, entonces para evitar que los valores $u_{i,2}$ calculados con (9.24) acarreen un error significativo, se elegirá un tamaño de paso k menor que h
Si $f(x)$ pertenece a una C^2 en$[0,a]$, se podría obtener una mejor aproximación a $u_{i,2}$.
Como $u_{xx}(x, 0) = f''(x)$,se desarrolla la expresión de Taylor de orden dos para lograr una aproximación mejor de los valores del j-ésimo paso. Si se toma como $x = x_i$ y $t = 0$ en la ecuación diferencial parcial de onda, se obtiene

$$u_{tt}(x_i, 0) = c^2 u_{xx}(x_i, 0) = c^2 f''(x_i) = c^2 (f_{i+1} - 2f_i + f_{i-1}/h^2) + O(h^2) \quad (9.25)$$

donde se empleó la condición $f''(x_i) = f_{i+1} - 2f_i + f_{i-1}/h^2$
Ahora se desarrolla el polinomio de Taylor de grado dos

$$u(x, k) = u(x, 0) + u_t(x, 0)k + \tfrac{1}{2} u_{tt}(x, 0)k^2 + O(k^3) \quad (9.26)$$

Al hacer $x = x_i$ en (9.26), como $u_{i,2} = f_i + kg_i$ y considerando(9.26)

$$u(x_i, k) = f_i + kg_i + c^2k^2/2h^2 (f_{i+1} - 2f_i + f_{i-1}) + O(h^2)O(k^2) + O(k^3) \quad (9.27)$$

Ya que $\lambda = ck/h$, (9.27) quedará para el paso j-ésimo.

$$u_{i,2} = (1-\lambda^2)f_i + kg_i + \lambda^2/2(f_i+1 + f_{i-1}) \text{ para } i = 2, 3,..., n-1. \quad (9.28)$$

Volviendo al ejemplo,se emplea el algoritmo desarrollado en la sección anterior con $n = 11$, $m = 21$, o sea $h = 0,1$, $k = 0,05$ y $\lambda = 1$. Se presenta la comparación numérica-analítica para $t=1$.

x_i	$u_{i,21}$	E
0.0	0	0
0.1	0.3090169944	5.55e-017
0.2	0.5877852523	2.22e-016
0.3	0.8090169944	0.0000000
0.4	0.9510565163	3.33e-016
0.5	1.0000000000	0.0000000
0.6	0.9510565163	1.11e-016
0.7	0.8090169944	0.0000000
0.8	0.5877852523	0.0000000
0.9	0.3090169943	5.55e-017
1.0	0	0

9.6. CONSISTENCIA, ESTABILIDAD Y CONVERGENCIA

CONSISTENCIA: Un esquema en diferencias se dice consistente si la ecuación discretizada tiende a la ecuación diferencial cuando Δx, Δy, Δt, etc. tienden a cero.
Esto es relativamente fácil de verificar, dado que plantear el desarrollo en serie de Taylor es siempre posible.

ESTABILIDAD: El esquema se dice estable si la diferencia entre la solución exacta y la numérica permanece acotada $\forall n\ \Delta t$, con Δt fijo, y siendo además la cota independiente de n.
Es decir,

$$\forall n \left| u_i^n - \bar{u}(x_i, n\,\Delta t) \right| \le k \quad \Delta t \text{ fijo}$$

Esta condición garantiza que los errores (por ejemplo, los iniciales) no se amplifican con el tiempo. Es una condición para el esquema, y no para la ecuación diferencial. Nótese que esto es muy parecido a decir que las soluciones de la EDP homogénea son acotadas en el tiempo.
En realidad, lo que interesa asegurar es que la solución exacta y la aproximada se parezcan; por ello se define un nuevo concepto, el de convergencia.

CONVERGENCIA: El esquema será convergente, si $\forall x, t$ (fijos)

$$\lim_{\substack{\Delta x \to 0 \\ \Delta t \to 0 \\ i,n \to +\infty}} \left| u_I^n - \bar{u}(i\Delta x, n\Delta t) \right| = 0 \quad \text{para } x = i\Delta x,\ t = n\Delta t \quad \text{fijos}$$

Se pueden vincular las dos primeras condiciones con la tercera, mediante el así llamado Teorema de equivalencia de Lax. "Para un problema de valor inicial bien planteado con un esquema de discretización consistente, la estabilidad es CN y S para la convergencia".

Por "bien planteado" se entiende un problema en el cual la solución en todo punto del dominio depende en forma continua de las condiciones iniciales y de borde; ello implica que pequeñas perturbaciones en éstas, producen pequeñas discrepancias en la solución.

9.7. ¿QUÉ ES PDE TOOLBOX?

La ecuación básica del PDE Toolbox es la EDP expresada como:
$-\nabla(c\nabla u) + au = f$ (elíptica) en Ω, es un dominio acotado en el plano, *c, a, f*, y la incógnita u son funciones escalares definidas en Ω, c puede ser una función matricial 2x2 definida en Ω.
El toolbox también puede manejar la ecuación parabólica

$$d\frac{\partial u}{\partial t} - \nabla(c\nabla u) + au = f$$

La hiperbólica

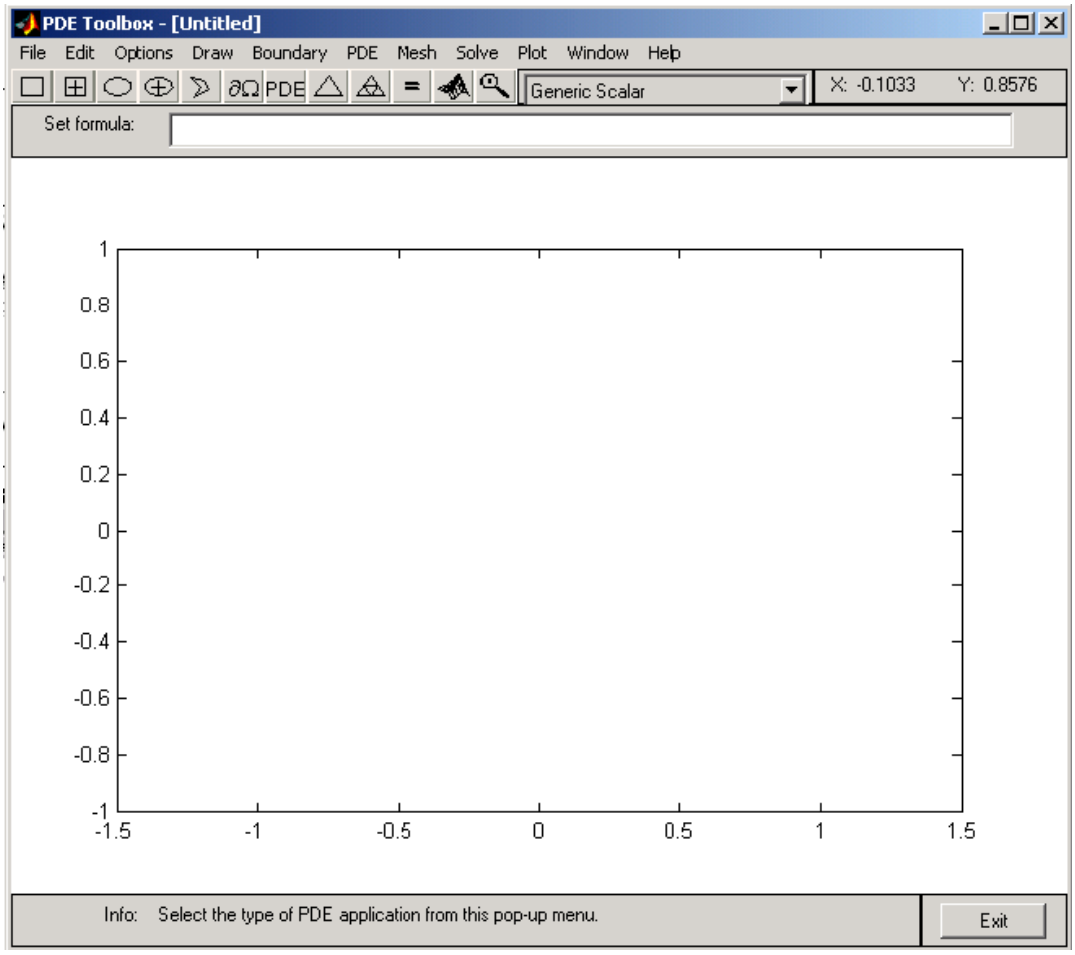
$$d\frac{\partial^2 u}{\partial t^2} - \nabla(c\nabla u) + au = f$$

y el problema de autovalores

$$-\nabla(c\nabla u) + au = \lambda du$$

Para las parabólica e hiperbólica los coeficientes *c, a, f, d* pueden variar con el tiempo
También existe un solver para las elípticas no lineales
$\nabla(c(u)\nabla u) + a(u)u = f(u)$
Las condiciones de frontera se definen para el escalar u
Dirichlet: $hu = r$ sobre la frontera $\partial\Omega$
Generalizada de Neumann: $\vec{n}(c\nabla u) + qu = g$ sobre $\partial\Omega$, con $\vec{n}$ normal unitaria exterior,. *g, q, h*, y *r* son funciones complejas definidas en $\partial\Omega$

Dispone de ocho interfaces de aplicación, a saber: mecánica estructural, tensiones mecánicas en el plano, electroestática, magnetoestática, transferencia de calor, difusión.



Escribiendo

>>pdetool

Existen un total de 11 menús desplegables en la interfaz gráfica. Brevemente, la funcionalidad de cada uno de ellos es la siguiente:

• Menú **File**. Como es habitual, desde este menú pueden abrirse y salvarse ficheros .m que contienen los modelos en ecuaciones en derivadas parciales en los que se esté trabajando. También pueden imprimirse las gráficas activas en ese momento y salir de la interfaz.

• Menú **Edit**. Capacidades de edición habituales: copiar, cortar, borrar y pegar objetos, así como opciones de seleccionar todo.

• Menú **Options**. Contiene opciones como cambiar el rango y espaciado de los ejes, obligar a que en la fase de dibujo las formas se anclen a los puntos de la rejilla, zoom, etc.

• Menú **Draw**. Desde este menú se pueden seleccionar los objetos sólidos básicos como círculos o polígonos que se emplearán en la definición de la geometría, y a continuación dibujarlos en el área de trabajo mediante el uso del ratón. Se recomienda el uso de la barra de herramientas para este fin.

• Menú **Boundary**. Desde este menú se accede al cuadro de diálogo donde se definen las condiciones de frontera. Adicionalmente se pueden poner etiquetas a los bordes y a los subdominios, borrar bordes entre subdominios y exportar la geometría descompuesta y las condiciones de fronteras al espacio de trabajo de Matlab.

• Menú **PDE**. Este menú proporciona un cuadro de diálogo para especificar la EDP, y opciones para etiquetar subdominios y exportar los coeficientes de la ecuación al espacio de trabajo.
• Menú **Mesh**. Desde este menú se crea y se modifica la malla triangular. Puede inicializarse la malla, refinarla, reorganizarla, deshacer cambios previos en la malla, etiquetar los nodos y los triángulos, visualizar la calidad de la malla y exportar la misma al espacio de trabajo.
• Menú **Solve**. Para resolver la EDP. También abre un cuadro de diálogo donde pueden ajustarse los parámetros involucrados en la resolución, y exportar la solución al espacio de trabajo.
• Menú **Plot**. Desde este menú se puede dibujar correctamente la solución a la EDP. Un cuadro de diálogo permite seleccionar que propiedad va a visualizarse, en qué estilo y otros tipos de parámetros. Si se ha generado una animación en tiempo de la solución, también puede exportarse al espacio de trabajo.
• Menú **Window**. Básicamente para seleccionar cual de las ventanas de figuras de Matlab es la activa entre las abiertas en ese momento.
• Menú **Help**. Breve ayuda sobre ciertos comandos y funcionalidades **Barra de herramientas**
La barra de herramientas colocada debajo del menú principal (fig. 2) en la parte superior de la interfaz gráfica contiene botones con iconos que proporcionan un acceso fácil y rápido a algunas de las funcionalidades más importantes de la pdetool

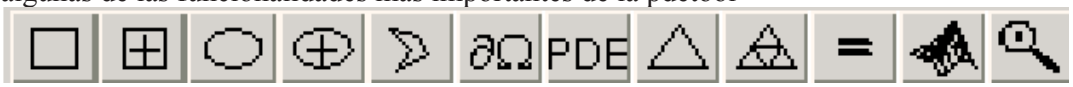
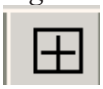


Fig. 2 Barra de herramientas de la pdetool *de* Matlab

 Dibuja un rectángulo/cuadrado comenzando en el centro.

 Dibuja una elipse/círculo comenzando en el perímetro.

 Dibuja una elipse/círculo comenzando en el centro.

.

 Dibuja un polígono. Pincha y arrastra para crear los lados del polígono. El polígono puede cerrarse haciendo click con el botón derecho del ratón o pinchando sobre el vértice inicial.
El hacer doble-click sobre uno de ellos fija como esa herramienta como activa, pudiendo seguir dibujando objetos del mismo tipo hasta que vuelva a pulsarse el botón. Usando el botón derecho del ratón, o bien Control+click, se restringen las herramientas a dibujar cuadrados o círculos en vez
Otra serie de botones

 Entra en el modo para especificar condiciones de frontera

 Abre el cuadro de diálogo para especificar la EDP a resolver.

 Inicializa la malla triangular.

 Refina la malla triangular.

 Resuelve la EDP

 Abre el cuadro de diálogo para representar los resultados.

 Zoom on/off.

Las geometrías complicadas pueden generarse a partir de dibujar objetos sólidos básicos (rectángulos/cuadrados, elipses/círculos y polígonos) que se solapen, parcial o totalmente. La interfaz gráfica asigna automáticamente un nombre a cada objeto sólido que se cree: R1, R2,... para los rectángulos; SQ1, SQ2,... para cuadrados;
E1, E2,... para elipses; C1, C2,... en el caso de los círculos; y P1, P2,... para nombrar los polígonos. Obviamente, estos nombres pueden ser modificados por el usuario haciendo doble-click sobre los mismos, lo que abre el cuadro de diálogo de las propiedades del objeto (Fig..4). Este cuadro de diálogo también nos posibilita el modificar otras características de la forma básica, como la posición de su centro, dimensiones, etc.

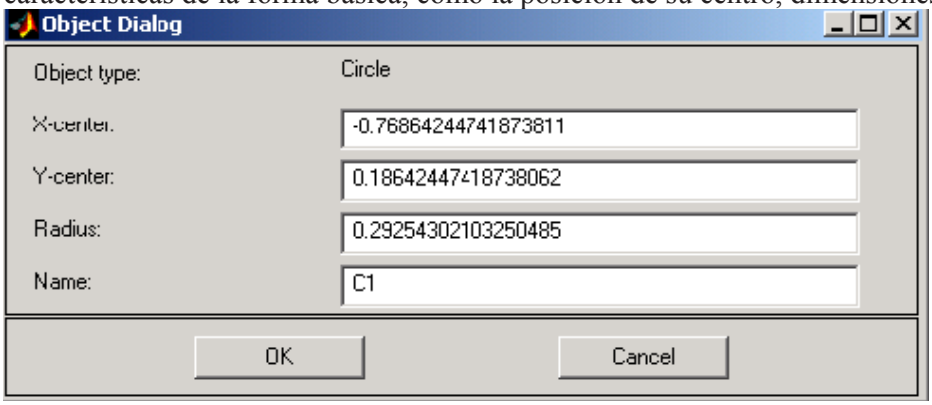


Figura 4: *Cuadro de diálogo con las propiedades de un círculo*

Una vez que los objetos básicos han sido dibujados, la geometría final se crea mediante la introducción, en la línea situada debajo de la barra de herramientas, de

una fórmula que emplee operaciones del álgebra de conjuntos, +, $*$ y −. De todos ellos, el operador de mayor precedencia es el operador diferencia, −, mientras que los operadores unión e intersección, + y $*$, poseen igual prioridad. Sin embargo, este orden de precedencia puede controlarse mediante el uso de paréntesis. El modelo geométrico final, Ω, es el conjunto de todos los puntos para los cuales la fórmula introducida puede evaluarse como verdadera. El proceso general puede entenderse más fácilmente a partir del siguiente ejemplo, la creación de una placa con esquinas redondeadas.

Para esquinas redondeadas

Se inicia la interfaz gráfica y activa la propiedad de la rejilla "snap-to-grid" localizada dentro del menú *Options*. Además, cambia el espaciado a -1.5:0.1:1.5 para el eje-*x* y -1:0.1:1 para el eje-*y*.

Se selecciona el icono para crear rectángulos y usando el ratón dibuja uno de anchura 2 y altura 1, comenzando en el punto (-1,0.5). Para crear las esquinas redondeadas añadir círculos, uno en cada esquina. Los círculos deben tener radio 0.2 y centros a una distancia de 0.2 unidades de las fronteras izquierda/derecha y superior/inferior del rectángulo ((-0.8,-0.3), (-0.8,0.3), (0.8,-0.3) y (0.8,0.3)). Para dibujar círculos en vez de elipses, usar el botón derecho del ratón o mantener la tecla ctrl. pulsada mientras se realiza el dibujo. Para finalizar, se dibuja en cada una de las esquinas un pequeño cuadrado de lado 0.2. Los objetos dibujados deberían presentar el aspecto mostrado en la parte izquierda de la Fig.5.

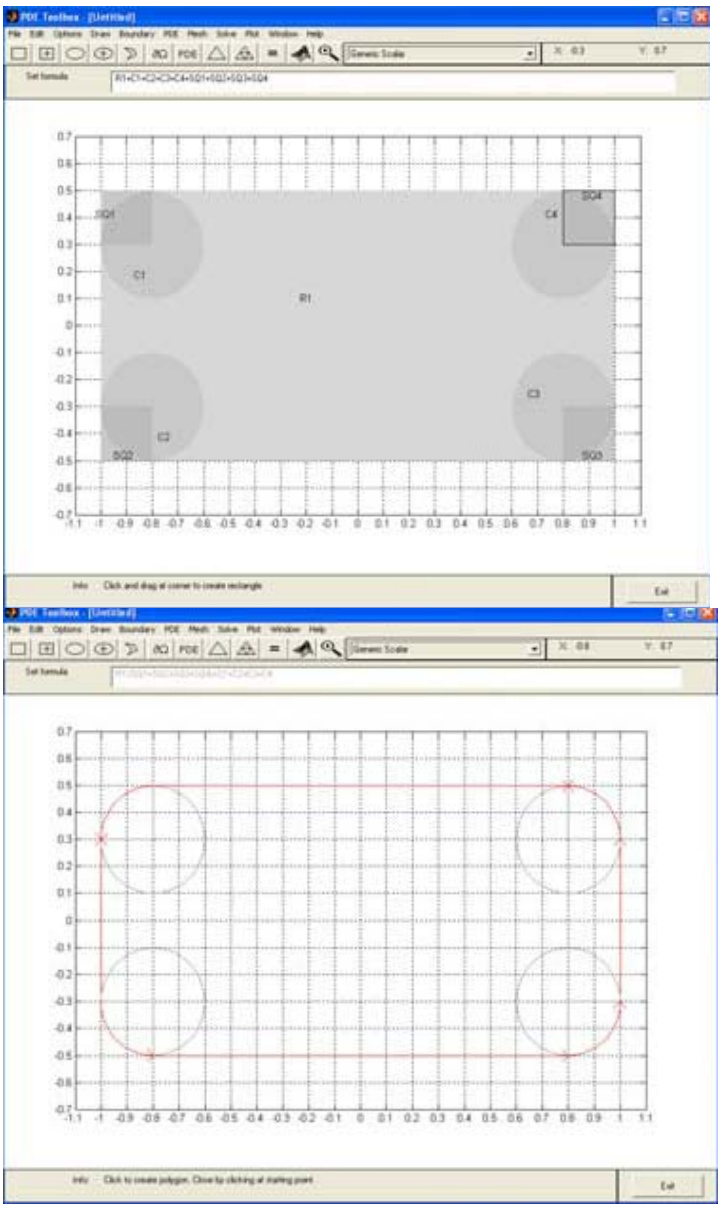


Figura.5: *(arriba) Objetos básicos a partir de los cuales sería definida la geometría definitiva. (abajo) Modelo geométrico final,* Ω.

Ahora se debe editar la fórmula que define la geometría. Para conseguir las esquinas redondeadas, se restan los cuadrados pequeños del rectángulo y se suma a continuación los círculos. En forma de expresión de conjuntos como:

R1 - (SQ1 + SQ2 + SQ3 + SQ4) + C1 + C2 + C3 + C4

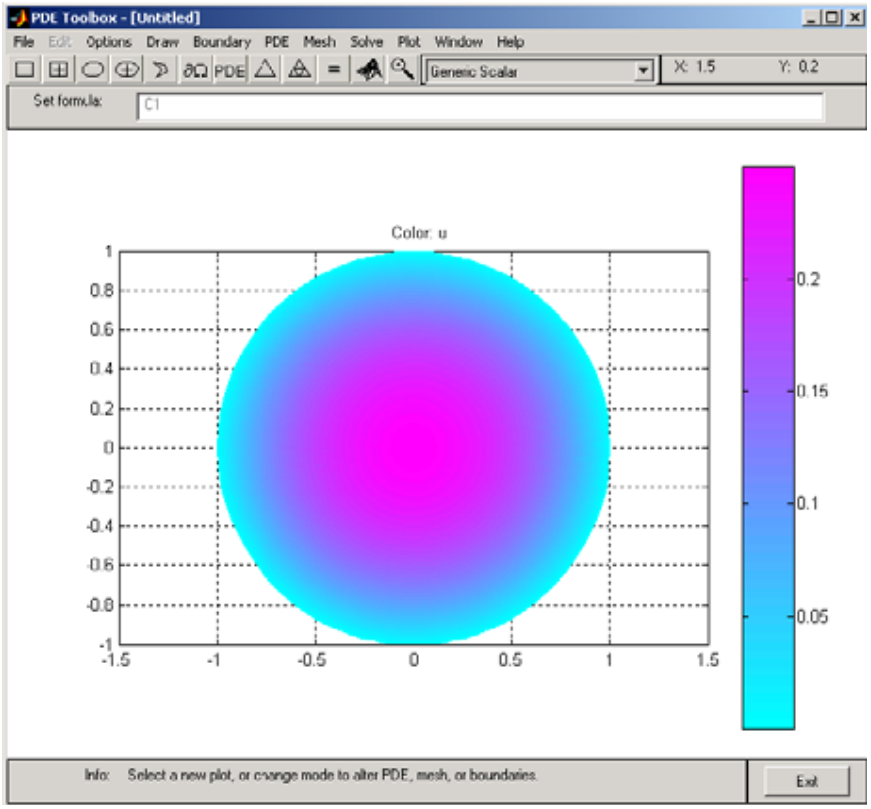
Presionando el botón **∂Ω** se puede entrar en el modo *Boundary* y ver las fronteras de la geometría final (Fig.5, abajo). Puede observarse que aún existen dentro de la placa algunas de las fronteras provenientes de los subdominios originales. Si se supone que la placa es homogénea, se pueden borrarlos. Para ello, se selecciona la opción "*Remove All Subdomain Borders*" del menú *Boundary*. Ahora el modelo de la placa está completo.

Pasos para el modelado

La siguiente secuencia de acciones cubre todos los pasos de una sesión normal empleando la *pdetoo*l:

1. Usar la pdetool como herramienta de dibujo para realizar el dibujo de la geometria 2-D en la que se quiere resolver la EDP, haciendo uso de los objetos básicos y de la característica de "fijar a rejilla". Combina los objetos sólidos mediante las fórmulas de álgebra de conjuntos para crear la geometría definitiva.
2. Guardar la geometría a un fichero de modelo (un fichero.m), de manera que se pueda seguir empleándola en futuras sesiones de trabajo. Si se guarda el fichero más adelante a lo largo del proceso de resolución, el fichero del modelo también incluirá ciertos comandos para recrear las condiciones de frontera, los coeficientes de la EDP y la malla.
3. Pasar a especificar las condiciones de frontera presionando el botón **∂Ω**. Si las fronteras no son las correctas, se puede volver a editar la geometría volviendo al modo de dibujo (*Draw mode*). Si durante la definición de la geometría han quedado bordes de subdominios no deseados se puede borrarlos mediante las opciones del menú *Boundary* ("*Remove Subdomain Border*" o "*Remove All Subdomain Borders"*). A continuación se puede fijar las condiciones de cada una de las fronteras haciendo doble-click sobre cada una de ellas.
4. Usar el botón **PDE** para especificar la EDP a resolver. En el caso en el que los coeficientes de la EDP dependan del material, estos son introducidos entrando en el modo *PDE* y haciendo doble-click en cada uno de los subdominios.
5. Inicializar la malla triangular mediante el botón **Δ**. Normalmente, los parámetros por defecto del algoritmo de generación de la malla producen buenos resultados, aunque en caso necesario pueden modificarse desde la opción "*Parameters*" del menú *Mesh*.
6. Si es necesario, refinar la malla mediante el botón Δ . En cada refinamiento, el número de triángulos aumenta en un factor cuatro. Tener presente que cuando más fina sea la malla mayor será el tiempo requerido para calcular la solución.
Otra opción es reordenar la triangulación de la malla para mejorar su calidad mediante la opción "*Jiggle Mesh*" del menú *Mesh.*
7. Resolver la EDP presionando el botón =.
8. Visualizar las propiedades de la solución en las que se esté interesado mediante el botón  .
También se puede exportar la solución y/o la malla al espacio de trabajo principal de Matlab para un análisis en mayor detalle.
Ejemplo
la *ecuación de Poisson* en un disco unitario con condiciones de frontera Dirichlet homogéneas. La formulación del problema es la siguiente (*Δ*es derivada parcial segunda espacial)
$-\Delta u = 1$ en Ω, $u = 0$ en $\delta\Omega$, donde Ω es el disco unitario y $\delta\Omega$ su frontera.
Se selecciona modo **Generic Scalar** de la lista desplegable de modos disponible que se encuentra localizada a la derecha de la barra de herramientas. A continuación se listan los pasosa realizar con la *pdetool* para resolver el problema descrito.
1. Dibujar un círculo de radio unidad centrado en el origen usando las herramientas de dibujo. Si se dibuja una elipse en vez de un círculo, o el mismo no esáa perfectamente centrado, hacer doble-click en el mismo y cambia sus propiedades en el cuadro de diálogo.
Seleccionar  , dibujando se obtiene el círculo1.m, se guarda.
2. Imposición de las condiciones de frontera. Para ello primero pulsar el botón **∂Ω** y a continuación hacer doble-click sobre las fronteras de la región. Seleccionar para todas ellas condiciones de frontera Dirichlet iguales a cero (aunque éstas son las condiciones de frontera por defecto).
Haciendo doble clic sobre el contorno del círculo, se abre un cuadro de diálogo: en este caso se selcciona la condición de Dirichlet, h=1,r=0
3. Definir la EDP presionando el botón **PDE**. Esto abre un cuadro de diálogo donde aparece activa el tipo *elliptic* y pueden introducirse los coeficientes de la ecuación, c, a y f. En este caso tan sencillo todos son constantes: c = 1, f = 1 y a = 0. En el caso de que

dependan de la posición, también pueden introducirse con la notación habitual de Matlab en forma de productos de vectores (por ejemplo, c = x.^2 +y.^2).
4. Inicializar la malla haciendo uso del botón **Δ,** refinándola con Δ , apareciendo el círculo mallado
5. Resolver la ecuación presionando =. Emplear también si se quiere cambiar las propiedades del gráfico que la pdetool devuelve por defecto, como por ejemplo el mapa de colores.



problemas hiperbólicos: la ecuación de ondas
Como ejemplo de una EDP hiperbólica, resolverla *ecuación de ondas*
$\partial^2 u/\partial t^2 - \Delta u = 0$
para las vibraciones transversales de una membrana en un cuadrado con esquinas en $(-1,-1)$, $(-1, 1)$, $(1,-1)$ y $(1, 1)$. La membrana está sujeta ($u = 0$) en los lados izquierdo y derecho, y se encuentra libre ($\partial u/\partial n = 0$) en los lados superior e inferior. Adicionalmente se necesita especificar los valores iniciales para $u(t0)$ y $\partial u(t0)/\partial t$.
Para este ejemplo se emplean como condiciones iniciales
$u(0) = \arctan(\cos(\pi/2)\, x)$ y
$\partial u(0)/\partial t = 3\,\mathrm{sen}(\pi x) e^{\mathrm{sin}(\pi/2\, y)}$
que son valores iniciales que satisfacen las condiciones de frontera. La razón de escoger las funciones arcotangente y exponencial es tan sólo para introducir más modos en la solución y hacerla de esta forma más atractiva.
La secuencia de pasos a realizar para resolver el problema por medio de la pdetool de Matlab son:
1. Asegurarse que se encuentra seleccionado el modo **Generic Scalar** en la lista de los posibles modos de solución.
2. Dibujar la geometría de interés, el cuadrado con esquinas en *(−1,−1), (−1, 1), (1,−1) y (1, 1).*

3. Imponerlas condiciones de frontera. Para ello primero pulsar el botón **∂Ω** y a continuación doble-click sobre las fronteras de la región. Para las frontera izquierda y derecha se introduce la condición Dirichlet $u = 0$ y para las superior e inferior condiciones Neumann homogéneas $\partial u/\partial n = 0$.
4. Introducir los coeficientes que definen la EDP presionando el botón **PDE**. Para este caso,*d = 1,c = 1,a = 0 y f = 0*. También se deben introducir la condiciones iniciales y el rango de tiempo en el que se quiere resolver el problema [0:0.5:5]. Para ello, seleccionar dentro del menú Solve la opción "**Parameter**s".
En el cuadro de diálogo emergente introduce *linspace(0,5,31*) como tiempos en los que resolver el problema, *atan(cos(pi/2*x))* como condición inicial para *u* y para *∂u/∂t* introduce *3*sin(pi*x).*exp(sin(pi/2*y))*, asegurándose que sea la *hyperbolic*
5. Inicializar la malla haciendo uso del botón **Δ,** refinándola con Δ
6. Resolver la ecuación presionando =. Presionando se pueden cambiar las propiedades de visualización. Como sugerencia, la mejor forma de ver el movimiento de las ondas es en forma de una animación, aunque la misma puede serrelativamente costosa en términos de tiempo y memoria, haciéndolo activando la opción **animation** en **plot selection; contour** es una alternativa para diferenciar las curvas nivel

9.7.1. Opción numérica:
pdepe resuelve sistemas de EDP en una variable especial y en el tiempo, del tipo

$$c(x,t,u,\frac{\partial u}{\partial x})\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}(x^m f(x,t,u,\frac{\partial u}{\partial x})) + s(x,t,u,\frac{\partial u}{\partial x}) \quad (1)$$

El intervalo[a,b] finito, m(0,1,2) según sea placa, cilíndrica o esférica,si m>0, ⊚≥0.El término $(x^m f(x,t,u,\frac{\partial u}{\partial x})$ es un término de flujo, $s(x,t,u,\frac{\partial u}{\partial x})$ es un término de fuente
El acoplamiento de las derivadas parciales respecto al tiempo está restringido a la multiplicación por una matriz diagonal $(x,t,u,\frac{\partial u}{\partial x})$;los elementos de esta diagonal pueden ser ceros o positivos:en el caso cero es elíptica de lo contrario es parabólica.
Una por lo menos debe ser parabólica
Ejemplo: EDP simple

$\pi^2\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}$ para [0,1],t>0

CI: *t=0, u(x,0)=sen(πx)*
CF: *x=0, u(0,t)=0*
x=1, πe^{-t}+(∂ u/∂x)(1,t)=0
1- reescribir la ecuación según la forma (1)
para este caso, se tiene:

$\pi^2\frac{\partial u}{\partial t} = x^0\frac{\partial}{\partial x}(x^0\frac{\partial u}{\partial x}) + 0$ con m=0 y los términos

$c(x,t,u,\frac{\partial u}{\partial x}) = \pi^2$

$f(x,t,u,\frac{\partial u}{\partial x}) = \partial u/\partial x$

$s(x,t,u,\frac{\partial u}{\partial x}) = 0$

2-Codificar la EDP
La función debe tener la forma

[c,f,s] = pdefun(x,t,u,dudx)
Donde c,f y s son los de (1)
Para el ejemplo
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;
f = DuDx;
s = 0;
3- codificar las funciones de las condiciones iniciales
Debe ser una función de la forma
u = icfun(x)
para el ejemplo
function u0 = pdex1ic(x)
*u0 = sin(pi*x);*
4- codificar las condiciones de frontera
Usar funciones del tipo
[pl,ql,pr,qr] = bcfun(xl,ul,xr,ur,t)
Como la forma general responde en $x=a$ y $x=b$ a:
$p(x,t,u)+q(x,t)f(x,t,u,\partial u/\partial x)=0$
en este caso serán:
$u(0,t)+0.\frac{\partial u}{\partial x}u(0,t)=0$ en $x=0$ y
$\pi e^{-t}+1(\partial u/\partial x)(1,t)=0$ en $x=1$
para evaluar $p(x,t,u)$ y $q(x,t)$ de las CF en la función *pdex1ic(x)*
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul;
ql = 0;
*pr = pi * exp(-t);*
qr = 1;
En la función *pdex1bc, pl* y *ql* corresponden a las condiciones de frontera izquierda ($x=0$) y pr y qr corresponden a la derecha ($x=1$)
5- seleccionar los puntos de malla parra la solución
Antes de usar el MATLAB PDE solver, se necesita especificar los puntos de malla *(t,x)*donde se quiere que *pdepe* evalue la solución; se deben especificar como vectores *t* y *x*
En el ejemplo se especifica 20 puntos igualmente separados en [0,1] y cinco valores de tiempo en [0,2].Así
>x = linspace(0,1,20);
>>t = linspace(0,2,5).
6- aplicar el solver
En este caso se llama a *pdepe* con *m = 0*, las funciones *pdex1pde, pdex1ic* y *pdex1bc*, y la malla definida por *x* y *t* donde *pdepe* evalúa la solución.
La function *pdepe* devuelve la solución numérica en un *arreglo ·D*, donde sol(i,j,k) aproxima la componente k de la solución,u_k, evaluada a t(i) y x(j)
>> m = 0;
>>sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
7- Ver los resultados.

En definitiva en la ventana de comandos
>> x = linspace(0,1,20);
>> t = linspace(0,2,5);m = 0;
>> sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t)

sol =

Columns 1 through 6

0	0.1646	0.3247	0.4759	0.6142	0.7357
0	0.0999	0.1972	0.2890	0.3729	0.4467
0	0.0607	0.1197	0.1754	0.2264	0.2711
0	0.0368	0.0726	0.1064	0.1373	0.1644
0	0.0223	0.0440	0.0645	0.0832	0.0997

Columns 7 through 12

0.8372	0.9158	0.9694	0.9966	0.9966	0.9694
0.5083	0.5560	0.5885	0.6050	0.6049	0.5883
0.3085	0.3374	0.3571	0.3670	0.3669	0.3568
0.1871	0.2046	0.2165	0.2225	0.2224	0.2162
0.1134	0.1240	0.1312	0.1348	0.1348	0.1310

Columns 13 through 18

0.9158	0.8372	0.7357	0.6142	0.4759	0.3247
0.5556	0.5078	0.4460	0.3720	0.2877	0.1956
0.3368	0.3077	0.2701	0.2252	0.1740	0.1181
0.2041	0.1864	0.1636	0.1363	0.1052	0.0713
0.1236	0.1129	0.0990	0.0824	0.0636	0.0430

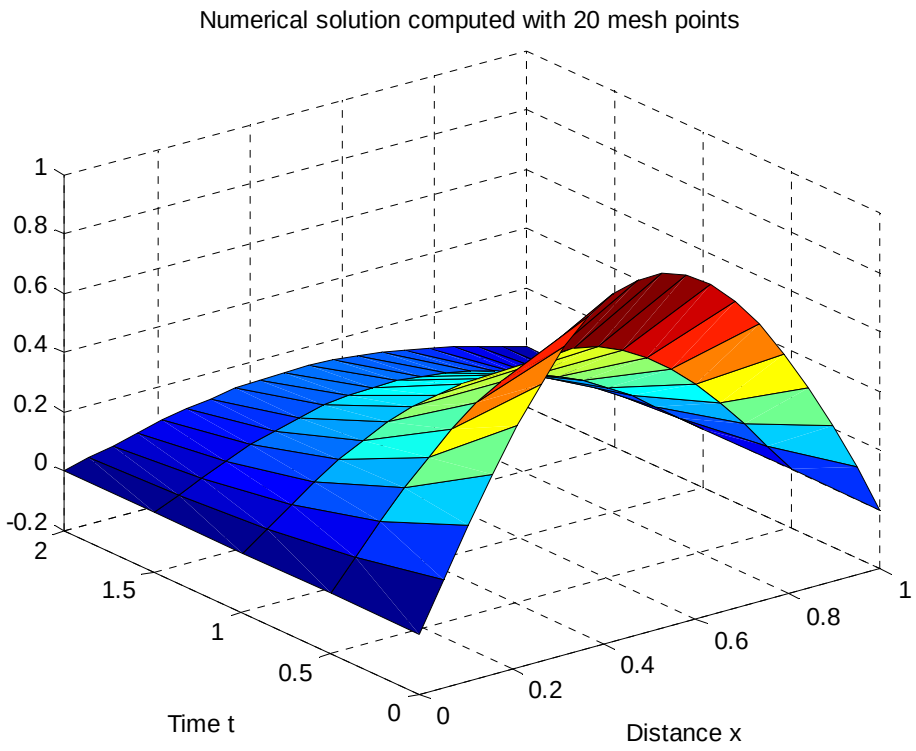
Columns 19 through 20

0.1646	0.0000
0.0981	-0.0022
0.0589	-0.0020
0.0354	-0.0015
0.0212	-0.0012

Extrayendo del arreglo 3D

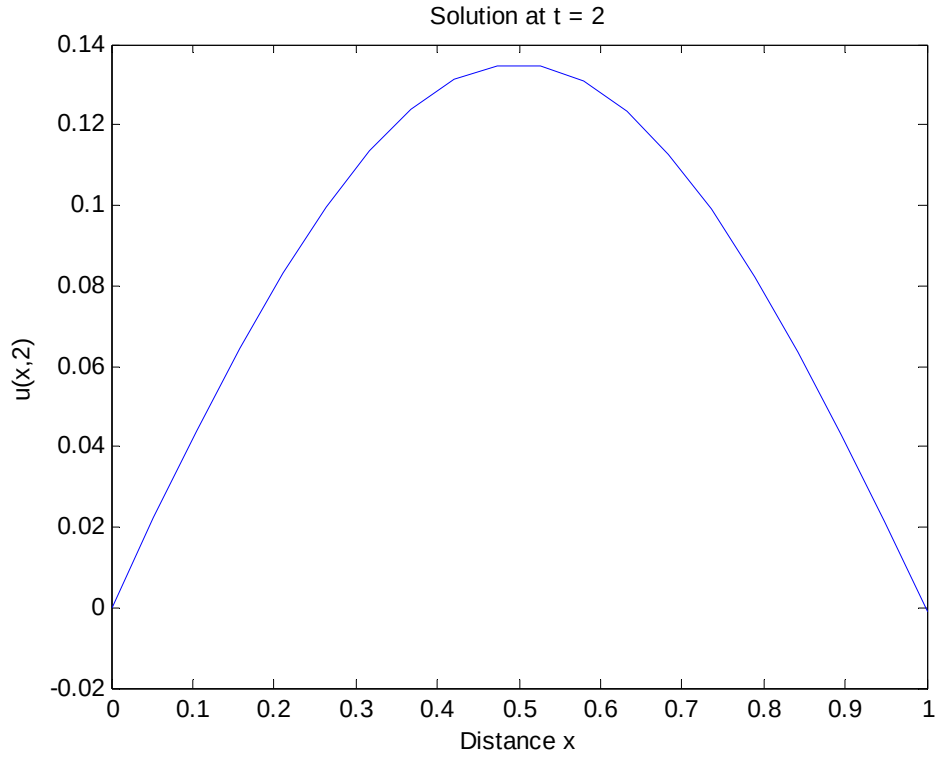
```
>>u = sol(:,:,1);

   surf(x,t,u)
   title('Numerical solution computed with 20 mesh points')
   xlabel('Distance x')
   ylabel('Time t')
```



Si se desa conocer el perfil de la solución al tiempo final,t=2

```
>>figure
   plot(x,u(end,:))
   title('Solution at t = 2')
   xlabel('Distance x')
   ylabel('u(x,2)')
```



Además, empleando los vectores *x* y *u(j,:)*, y la función de ayuda *pdeva*l, se puede evaluar la solución *u* y su derivada en cualquier conjunto de puntos *xou*t, así
[uout,DuoutDx] = pdeval(m,x,u(j,:),xout)
donde
m:0,1 o 2. This is the first input argument used in the call to *pdep*e
xmesh: un vector *[x0, x1,..., xn]* que especifica los puntos donde se calcularon los elementos de ui ; es el mismo vector con el cual se llamó a *pdepe*.
*u*i; un vector *sol(j,:,i*) que aproxima la componente i de la solución al tiempo t_f y puntos de malla *xmesh,* donde sol es la solución devuelta por pdepe.
xout. Un vector de puntos del intervalo*[x0,xn*] a los cuales la solución interpolado se requiere.
por ejemplo
>>pdeval(0,x,u,[0,0.5])

ans =

0 0.1735

9.8. EJERCITACION UNIDAD 9 CON MATLAB

I) Ecuación de onda
$(\partial^2 u(x,t)/\partial t^2)=c^2(\partial^2 u(x,t)/\partial x^2)$ en $0<x<1$ y $0<t<0.5$
$u(0,t)=0, u(1,t)=0$ son las CF
$u(x,0)= u(x,0)=1.5- 1.5x$ en $0\leq x\leq 1$
$u_t(x,0)=0$ en $0<x<1$

```
>> f=@(x)1.5-1.5*x;
>> g=@(x)0*x;a=1;b=0.5;c=2;n=10;m=10;
>> finedif(f,g,a,b,c,n,m)
ans =
  Columns 1 through 6

    0    1.3333    1.1667    1.0000    0.8333    0.6667
    0    1.3333    1.1667    1.0000    0.8333    0.6667
    0   -0.1667    1.1667    1.0000    0.8333    0.6667
    0   -0.1667   -0.3333    1.0000    0.8333    0.6667
    0   -0.1667   -0.3333   -0.5000    0.8333    0.6667
    0   -0.1667   -0.3333   -0.5000   -0.6667    0.6667
    0   -0.1667   -0.3333   -0.5000   -0.6667   -0.8333
    0   -0.1667   -0.3333   -0.5000   -0.6667   -0.8333
    0   -0.1667   -0.3333   -0.5000   -0.6667   -0.8333
    0   -0.1667   -0.3333   -0.5000   -0.6667   -0.8333

  Columns 7 through 10

    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
    0.5000    0.3333    0.1667         0
```

-1.0000	0.3333	0.1667	0
-1.0000	-1.1667	0.1667	0
-1.0000	-1.1667	-1.3333	0

Valores de u en los 10 nodos de x y en los de tiempo

II) Ecuación de calor

$(\partial u(x,t)/\partial t)=c^2(\partial^2 u(x,t)/\partial x^2)$ en $0<x<1$ y $0<t<0.1$
$u(0,t)=0, u(1,t)=0$ son las CF para $0\leq t\leq 0.1$
$u(x,0)=\sin(\pi x)+\sin(3\pi x)$ a $t=0$ y $0\leq x\leq 1$
h=0.1,k=0.01,r=1, generando n=11 y m=11

```
>> f=@(x)1.5-1.5*x;
>> g=@(x)0*x;a=1;b=0.5;c=2;n=10;m=10
>> f=@(x)sin(pi*x)+sin(3*pi*x);c1=0;c2=0;c=1;n=11;m=11;
>> forwdif(f,c1,c2,a,b,c,n,m)

ans =

  1.0e+004 *

  Columns 1 through 7

         0    0.0001    0.0002    0.0001    0.0000         0    0.0000
         0   -0.0002   -0.0003   -0.0001    0.0002    0.0004    0.0002
         0    0.0008    0.0009    0.0003   -0.0005   -0.0009   -0.0005
         0   -0.0025   -0.0029   -0.0009    0.0018    0.0031    0.0018
         0    0.0077    0.0090    0.0029   -0.0056   -0.0095   -0.0056
         0   -0.0240   -0.0282   -0.0092    0.0174    0.0297    0.0174
         0    0.0749    0.0881    0.0286   -0.0544   -0.0926   -0.0544
         0   -0.2340   -0.2750   -0.0894    0.1700    0.2892    0.1700
         0    0.7304    0.8587    0.2790   -0.5307   -0.9029   -0.5307
         0   -2.2805   -2.6809   -0.8711    1.6569    2.8189    1.6569
         0    7.1202    8.3703    2.7197   -5.1731   -8.8010   -5.1731

  Columns 8 through 11

    0.0001    0.0002    0.0001         0
   -0.0001   -0.0003   -0.0002         0
    0.0003    0.0009    0.0008         0
   -0.0009   -0.0029   -0.0025         0
    0.0029    0.0090    0.0077         0
   -0.0092   -0.0282   -0.0240         0
    0.0286    0.0881    0.0749         0
   -0.0894   -0.2750   -0.2340         0
    0.2790    0.8587    0.7304         0
   -0.8711   -2.6809   -2.2805         0
    2.7197    8.3703    7.1202         0
```

Empleando Crank-Nicholson

```
>> crnich(f,c1,c2,a,b,c,n,m)
```

U =

Columns 1 through 7

0	1.1180	1.5388	1.1180	0.3633	0	0.3633
0	-0.0929	0.0270	0.3838	0.7808	0.9534	0.7808
0	0.2110	0.3307	0.3350	0.2795	0.2480	0.2795
0	0.0353	0.0917	0.1679	0.2370	0.2651	0.2370
0	0.0536	0.0934	0.1141	0.1204	0.1211	0.1204
0	0.0214	0.0436	0.0650	0.0812	0.0873	0.0812
0	0.0168	0.0310	0.0409	0.0465	0.0482	0.0465
0	0.0089	0.0172	0.0243	0.0292	0.0309	0.0292
0	0.0058	0.0110	0.0149	0.0174	0.0182	0.0174
0	0.0034	0.0065	0.0090	0.0106	0.0112	0.0106
0	0.0021	0.0040	0.0055	0.0064	0.0067	0.0064

Columns 8 through 11

1.1180	1.5388	1.1180	0
0.3838	0.0270	-0.0929	0
0.3350	0.3307	0.2110	0
0.1679	0.0917	0.0353	0
0.1141	0.0934	0.0536	0
0.0650	0.0436	0.0214	0
0.0409	0.0310	0.0168	0
0.0243	0.0172	0.0089	0
0.0149	0.0110	0.0058	0
0.0090	0.0065	0.0034	0
0.0055	0.0040	0.0021	0

III) Elíptica
Se plantea la ecuación de Laplace, div(div)=0 en R={(x,y):0≤x≤4,0≤y≤4}con las CF dadas por:
u(x,0)=10 y u(x,4)=100 para 0<x<4 y
u(0,y)=50 y u(4,y)=0 para 0<y<4
se toma h=0.5 para x e y, o sea 64 cuadrados en la malla

```
>> f1=@(x)0*x+10;
>> f2=@(x)0*x+100;
>> f3=@(x)0*x+50;
>> f4=@(x)0*x;
>> a=4;b=4;h=0.5;tol=0.001;max1=20;
>> dirich(f1,f2,f3,f4,a,b,h,tol,max1)
```

ans =

Columns 1 through 7

75.0000	100.0000	100.0000	100.0000	100.0000	100.0000	100.0000
50.0000	72.5622	79.8894	81.6479	80.5245	76.6881	68.3084
50.0000	60.3593	65.3473	66.1776	63.7618	57.9197	47.2421
50.0000	53.5276	54.9632	53.9535	50.4254	43.9867	33.8348

50.0000	48.7879	47.0243	44.2476	39.9998	33.7667	25.0342
50.0000	44.5998	40.0985	36.0129	31.5594	26.0462	18.9704
50.0000	39.5133	32.7575	28.1462	24.1790	19.8883	14.6525
50.0000	30.6963	23.2722	19.6352	17.1224	14.6755	11.6914
30.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000

Columns 8 through 9

100.0000	50.0000
49.3035	0
28.9055	0
19.0763	0
13.5648	0
10.1487	0
8.0597	0
7.4378	0
10.0000	5.0000

EJERCICIOS PROPUESTOS PARA UNIDAD 9

Ecuaciones diferenciales parciales

Aproximar las soluciones de las siguientes EDP

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=(x^2+y^2)e^{xy} \qquad 0<x<2;0<y<1;$$

1. $u(0,y)=1,\ \ u(2,y)=e^{2y},\quad 0\le y\le 1;$ sn. exacta=$u(x,y)=e^{xy}$

   $u(x,0)=1,\ \ u(x,1)=e^{x},\quad 0\le x\le 2$

   $h=0.2,k=0.1,$

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=(\frac{x}{y}+\frac{y}{x}) \qquad 1<x<2;1<y<2;$$

2. $u(1,y)=y\ln y,\ \ u(2,y)=2y\ln(2y),\quad 1\le y\le 2;$ sn exacta: $u(x,y)=xyln(xy)$

   $u(x,1)=x\ln x,\ \ u(x,2)=x\ln(4x^2),\quad 1\le x\le 2;$

   $h=k=0.1,$

$$\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=0 \qquad 0<x<\pi;0<t;$$

3. $u(0,t)=u(\pi,t)=0,\quad 0<t;$ sn exacta: $u(x,t)=e^{-t}senx$

   $u(x,0)=senx,\quad 0\le x\le\pi$

   $h=\pi/10,k=0.05.$

$$\frac{\partial u}{\partial t}=\frac{4}{\pi^2}\frac{\partial^2 u}{\partial x^2} \qquad 0<x<4;0<t;$$

4. $u(0,t)=u(4,t)=0,\quad 0<t;$ sn. exacta: $e^{-t}sen(\pi/2)x+: e^{-t/4}sen(\pi/4)x$

   $u(x,0)=sen\frac{\pi}{4}x(1+2\cos\frac{\pi}{4}x),\quad 0\le x\le 4$

   $h=0.2,k=0.04.$

$$\frac{\partial u}{\partial t^2}=\frac{\partial^2 u}{\partial x^2} \qquad 0<x<1;0<t;$$

$u(0,t)=u(1,t)=0,\quad 0<t;$

5. $u(x,0)=\begin{Bmatrix}1 \quad \text{en } 0\le x\le 0.5\\ -1 \text{ en } 0.5\le x\le 1\end{Bmatrix}$

   $\frac{\partial u}{\partial t}(x,0)=0,\quad 0\le x\le 1$

   $h=0.1,k=0.1.$

$\frac{\partial u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad 0 < x < 1; 0 < t;$

$u(0,t) = u(1,t) = 0, \quad 0<t;$

6. $u(x,0) = sen2\pi x \quad 0 \le x \le 1$ sn exacta:$sen2\pi x(cos2\pi t+sen2\pi t)$

$\frac{\partial u}{\partial t}(x,0) = 2\pi sen2\pi x \quad 0 \le x \le 1$

$h = 0.1, k = 0.1.$

BIBLIOGRAFÍA

BURDEN, R.; FAIRES, D.
Análisis Numérico, cuarta edición, grupo Editorial Iberoamericana.

BURGOS, J.
Álgebra Lineal, McGraw Hill.

FROBERG, C. E.
Introducción al Análisis Numérico, Vicens Universidad.

GROSSMAN, S.
Álgebra Lineal, quinta edición, McGraw Hill.

HOFFMANN, J.
Numerical Methods for engineers and Scienctists, McGraw Hill.

MATLAB, R.
13. 2007. Demos y Ayuda.

ZAUDERER, E.
Partial Diferential Equations of Applied Mathematics, John Wiley y Sons.

ZILL, D.
Cálculo con Geometría Analítica, Grupo Editorial Iberoamericana.