ANÁLISIS NUMÉRICO BÁSICO

Un enfoque algorítmico con el soporte de MATLAB®

ISBN 978-9942-922-01-4

Escuela Superior Politécnica del Litoral
Instituto de Ciencias Matemáticas
Guayaquil - Ecuador

Luis Rodríguez Ojeda, MSc.

CONTENIDO

ANÁLISIS NUMÉRICO BÁSICO

Un enfoque algorítmico con el soporte de MATLAB®

Prefacio

Esta obra es una contribución dedicada a los estudiantes que toman el curso de Análisis Numérico en las carreras de ingeniería en la ESPOL. El pre-requisito es haber tomado los cursos de matemáticas del ciclo básico de nivel universitario y alguna experiencia previa con el programa MATLAB para aprovechar el poder de este instrumento computacional y aplicar los métodos numéricos de manera efectiva.

El contenido se basa en la experiencia desarrollada en varios años impartiendo los cursos de Análisis Numérico y Métodos Numéricos en las carreras de ingeniería, sin embargo esta obra solo pretende ser un texto complementario para estas materias. La orientación principal del material es su aplicación computacional, sin descuidar los conceptos matemáticos básicos con los que se fundamentan los métodos numéricos.

Este texto contribuye también a difundir entre los estudiantes el uso del programa MATLAB para cálculos, graficación y manejo matemático simbólico para integrarlo como un soporte común para todos los cursos básicos matemáticos, incluyendo Álgebra Lineal, Cálculo Diferencial e Integral, Ecuaciones Diferenciales, Análisis Numérico y otros.

MATLAB dispone de un amplio repertorio de funciones especiales para su aplicación inmediata en la solución de una gran cantidad de problemas matemáticos, sin embargo sería equivocado usarlas como una receta sin el conocimiento de sus fundamentos matemáticos. En este curso se desarrollan algunas funciones alternativas a las que ofrece MATLAB, y en algunos casos, nuevas funciones cuya programación es instructiva para orientar a los estudiantes en el desarrollo de software para matemáticas e ingeniería.

El segundo objetivo principal de esta obra es contribuir al desarrollo de textos virtuales en la ESPOL de tal manera que puedan ser usados en línea e interactivamente, aunque también puedan imprimirse, pero tratando de reducir al mínimo el uso de papel. Otra ventaja importante de los textos virtuales es que pueden ser actualizados y mejorados continuamente sin costos de impresión. El texto ha sido compilado en formato **pdf**. El tamaño del texto en pantalla es controlable y tiene un índice electrónico para facilitar la búsqueda de temas. Se pueden usar facilidades para resaltar y marcar texto, agregar comentarios, notas, enlaces, revisiones, búsqueda por contenido, etc.

Adjunto a este libro virtual se provee un documento en formato de texto copiable con todas las funciones instrumentadas en MATLAB desarrolladas en este documento, de tal manera que los usuarios puedan tomarlas y pegarlas en la ventana de edición de MATLAB, probarlas inmediatamente y posteriormente realizar cambios y mejoras en las mismas.

También se adjunta un manual de uso del programa MATLAB para los usuarios que deseen adquirir o mejorar su conocimiento de este programa.

Finalmente, debo agradecer a la ESPOL y a sus autoridades por dar las facilidades a sus profesores para que desarrollen su actividad académica.

Luis Rodríguez Ojeda, MSc.
lrodrig@espol.edu.ec
Instituto de Ciencias Matemáticas
Escuela Superior Politécnica del Litoral
2011

ANÁLISIS NUMÉRICO

Un enfoque algorítmico con el soporte de MATLAB®

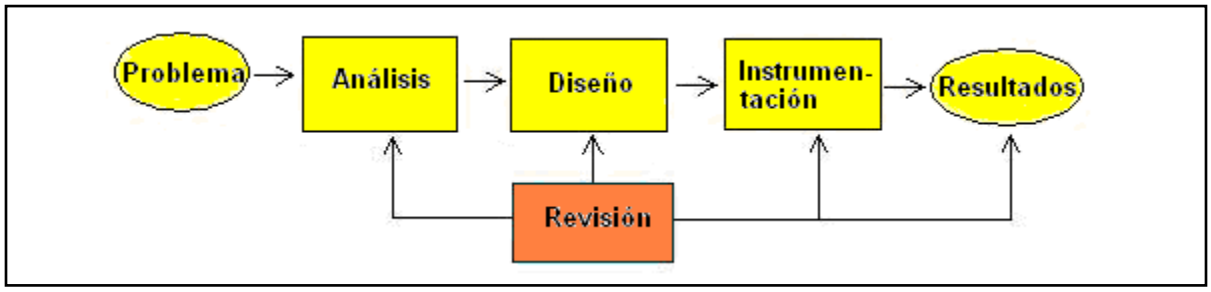
1 INTRODUCCIÓN

Análisis Numérico es una rama de la matemática cuyo objetivo principal es el estudio de métodos para resolver problemas numéricos complejos. El estudio de estos métodos no es reciente, pero actualmente con el apoyo de la computación se los puede usar con mucha eficiencia en la resolución de problemas que antes no era posible.

En este curso se proporciona a los estudiantes el conocimiento matemático básico para proveer soporte y formalidad a cada uno de los métodos estudiados. Se desarrolla la forma algorítmica de los métodos y finalmente se instrumenta su forma computacional usando la capacidad de cálculo, visualización y programación de **MATLAB**. Este componente práctico del curso es desarrollado en un laboratorio de computación.

El estudio de cada método se complementa con el desarrollo de ejemplos y ejercicios que pueden resolverse con la ayuda de una calculadora. Sin embargo, el objetivo principal del curso es la aplicación de los métodos para obtener respuestas con precisión controlada en la resolución de algunos problemas de ingeniería que por su complejidad requieren usar el computador.

1.1 Resolución de problemas con el computador



Suponer un problema que debemos resolver y que está en nuestro ámbito de conocimiento.

En la etapa de **Análisis** es necesario estudiar y entender el problema. Sus características, las variables y los procesos que intervienen. Asimismo, debemos conocer los datos requeridos y el objetivo esperado. En el caso de problemas de tipo numérico, el resultado de esta etapa será un **modelo matemático** que caracteriza al problema. Por ejemplo, un sistema de ecuaciones lineales.

En la etapa de **Diseño** procedemos a elegir el método numérico apropiado para resolver el modelo matemático. Debe suponerse que no se puede, o que sería muy laborioso, obtener la solución exacta mediante métodos analíticos. Los métodos numéricos permiten obtener soluciones aproximadas con simplicidad. El resultado de esta etapa es la formulación matemática del método numérico y la elaboración de un **algoritmo** para usarlo.

En la etapa de **Instrumentación** elegimos el dispositivo de cálculo para la obtención de resultados. En problemas simples, basta una calculadora. Para problemas complejos, requerimos el computador mediante funciones predefinidas y en algunos casos, desarrollando **programas** y **funciones** en un lenguaje computacional. Esta última opción es importante para la comprensión de los métodos.

Este proceso debe complementarse con una **revisión** y retroalimentación. Es preferible invertir más tiempo en las primeras etapas, antes de llegar a la instrumentación.

1.2 Fuentes de error en la resolución de un problema numérico

En el **Análisis** pueden introducirse errores debido a suposiciones inadecuadas, simplificaciones y omisiones al construir el modelo matemático. Estos errores se denominan errores inherentes.

En el **Diseño** se pueden introducir errores en los métodos numéricos utilizados los cuales se construyen mediante fórmulas y procedimientos simplificados para obtener respuestas aproximadas. También se pueden introducir errores al usar algoritmos iterativos. Estos errores se denominan errores de truncamiento.

En la **Instrumentación** se pueden introducir errores en la representación finita de los números reales en los dispositivos de almacenamiento y en los resultados de las operaciones aritméticas. Este tipo de error se denomina error de redondeo. También se pueden introducir errores de redondeo al usar datos imprecisos.

Los errores son independientes y su efecto puede acumularse. En el caso del error de redondeo el efecto puede incrementarse si los valores que se obtienen son usados en forma consecutiva en una secuencia de cálculos.

Debido a que los métodos numéricos en general, permiten obtener únicamente aproximaciones para la respuesta de un problema, es necesario definir alguna medida para cuantificar el error en el resultado obtenido. En general, no es posible determinar exactamente este valor por lo que al menos debe establecerse algún criterio para estimarlo o acotarlo. Esta información es útil para conocer la precisión de los resultados calculados.

1.3 El modelo matemático

Al resolver un problema con el computador, la parte más laboriosa normalmente es el análisis del problema y la obtención del modelo matemático que finalmente se usa para obtener la solución.

El modelo matemático es la descripción matemática del problema que se intenta resolver. Esta formulación requiere conocer el ámbito del problema y los instrumentos matemáticos para su definición.

1.4 Algoritmos numéricos

Un algoritmo es una descripción ordenada de los pasos necesarios para resolver un problema. Para diseñar un algoritmo para resolver un problema numérico es necesario conocer en detalle la formulación matemática, las restricciones de su aplicación, los datos y algún criterio para validar y aceptar los resultados obtenidos.

Esta descripción facilita la instrumentación computacional del método numérico. En problemas simples puede omitirse la elaboración del algoritmo e ir directamente a la codificación computacional.

1.5 Instrumentación computacional

En este curso se usará **MATLAB** para instrumentar los algoritmos correspondientes a los métodos numéricos estudiados. La aplicación computacional puede realizarse usando directamente la funcionalidad disponible en el lenguaje. Sin embargo, para comprender los métodos numéricos, preferible instrumentar el algoritmo desarrollando una función en el lenguaje computacional y tratando de que sea independiente de los datos específicos de un problema particular para facilitar su reutilización. Estas funciones pueden llamarse desde la ventana de comandos o mediante un programa que contendrá los datos del problema que se desea resolver.

Se supondrá que los estudiantes tienen el conocimiento básico del lenguaje y del entorno de **MATLAB**. En este curso se suministra un tutorial para uso de este instrumento computacional.

1.5.1 Elementos de MATLAB

MATLAB es un instrumento computacional simple de usar, versátil y de gran poder para aplicaciones numéricas, simbólicas y gráficas. Contiene una gran cantidad de funciones predefinidas para aplicaciones en áreas de las ciencias e ingeniería. Este instrumento puede usarse en **forma interactiva** mediante **comandos** o mediante instrucciones creando **programas** y **funciones** con las que se puede agregar poder computacional a la plataforma **MATLAB**.

1.5.2 Uso interactivo de MATLAB

Al ingresar al programa **MATLAB** se tiene acceso a la **Ventana de Comandos**. Los comandos son las instrucciones que se escriben para obtener resultados en forma inmediata.

Ejemplo. Para calcular

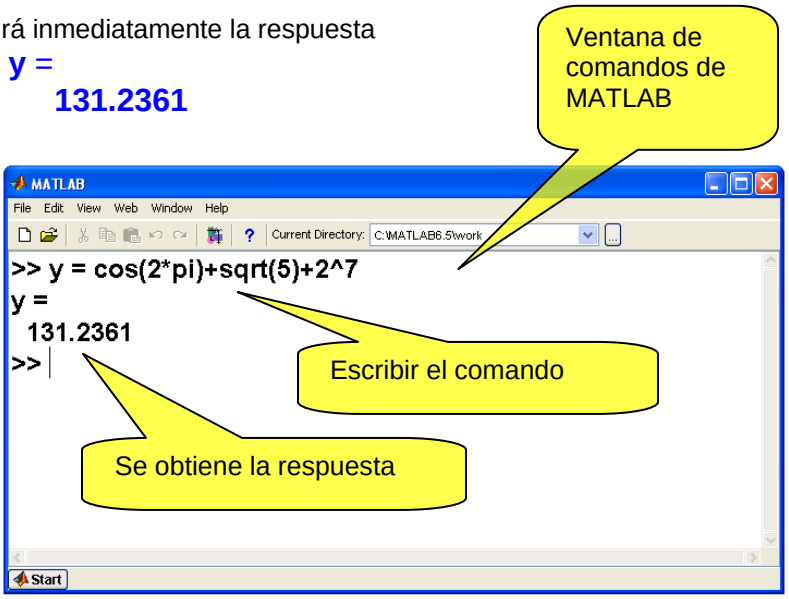
$$y = \cos(2\pi) + \sqrt{5} + 2^7$$

Digite en la ventana de comandos de MATLAB

```
y = cos(2*pi)+sqrt(5)+2^7
```

Obtendrá inmediatamente la respuesta

```
y =
    131.2361
```



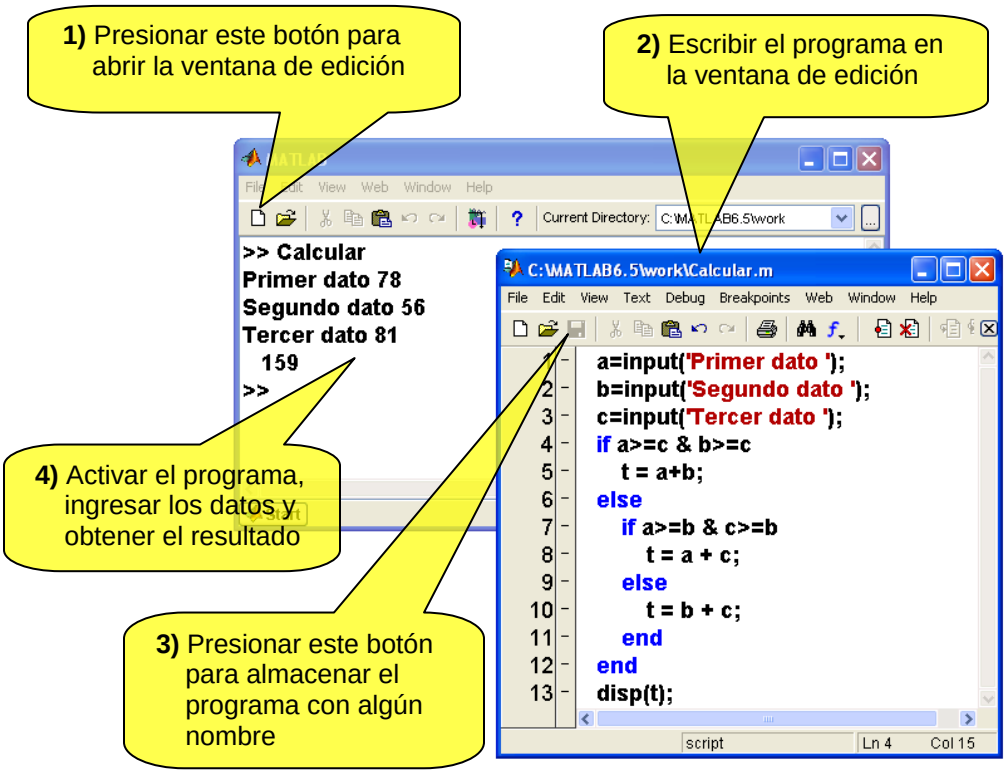
1.5.3 Programación en MATLAB

Junto a este curso se suministra un tutorial de **MATLAB**. Se sugiere usarlo como referencia y adquirir familiaridad con la sintaxis y operatividad de este lenguaje.

Para usar el componente programable de **MATLAB** debe abrir una ventana de edición presionando el botón **New M-File** o **New Script** en la barra del menú de opciones de **MATLAB**.

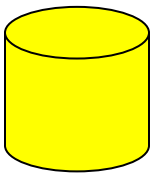
Escriba el programa en la ventana de edición y almacénelo con algún nombre. Finalmente, active el programa escribiendo el nombre en la ventana de comandos. Ingrese los datos y obtenga los resultados.

Ejemplo. Escribir y probar un programa en MATLAB un algoritmo para obtener la suma de las dos mejores calificaciones de las tres obtenidas en un curso.



A continuación se proporciona un ejemplo para seguir el procedimiento descrito para la resolución de un problema

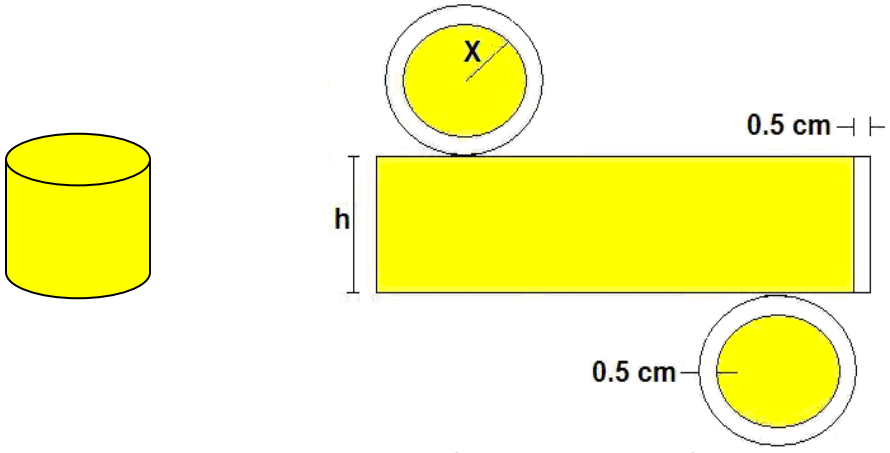
Problema. *Un empresario desea producir recipientes cilíndricos de aluminio de un litro de capacidad. Cada recipiente debe tener un borde de* ***0.5*** *cm. adicionales para sellarlo. Determine las dimensiones del recipiente para que la cantidad de material utilizado en la fabricación sea mínima.*



Análisis

Para formular el modelo matemático del problema, debe entenderse detalladamente. En este ejemplo el dibujo facilita visualizar sus componentes.

Sean: **x, h**: radio y altura del cilindro, respectivamente



Área total: $\mathbf{s = 2\pi(x+0.5)^2 + (2\pi x+0.5)h}$ cm^2 (1)

Dato del volumen requerido: $\mathbf{v = \pi x^2 h = 1000}$ cm^3 (2)

De (2) obtenemos **h**:

$$\mathbf{h = \frac{1000}{\pi x^2}}$$

Sustituimos en (1):

$$\mathbf{s(x) = 2\pi(x+0.5)^2 + \frac{1000(2\pi x+0.5)}{\pi x^2}}$$

Se obtiene una función para la que debe determinarse el valor de la variable **x** que minimice **s**

El Cálculo Diferencial nos proporciona un procedimiento para obtener la respuesta. Se debe resolver la ecuación: **s'(x) = 0**. Este es el modelo matemático del cual se obtendrá la solución.

Algoritmo

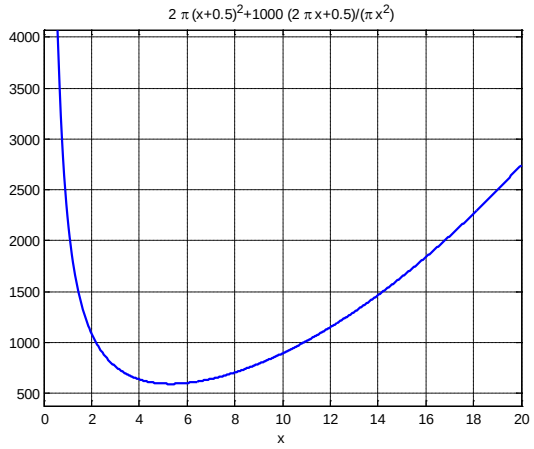
1) Dado el modelo matemático **s(x)**
2) Obtener **s'(x)**
3) Resolver la ecuación **s'(x) = 0**
4) Elegir y validar la respuesta

Instrumentación

Se realizará utilizando funciones existentes en el programa MATLAB

Es útil visualizar la función **s** mediante un gráfico

```
>> s='2*pi*(x+0.5)^2+1000*(2*pi*x+0.5)/(pi*x^2)';
>> ezplot(s,[0,20]),grid on
```



Se observa que hay un valor real positivo que minimiza la función **s(x)**

Obtener la derivada de **s(x)**

```
>> d=diff(s)
d =
   2*pi*(2*x + 1.0) + 2000/x^2 - (2*(2000*pi*x + 500.0))/(pi*x^3)
```

Resolver la ecuación **s'(x)=0**

```
>> x=eval(solve(d))
x =
  5.3112
 -0.1592
 -2.8260 + 4.6881i
 -2.8260 - 4.6881i
```

Por la naturaleza del problema, solamente la primera respuesta **x(1)** es aceptable

```
>> h=1000/(pi*x(1)^2)
h =
   11.2842
```

Solución

Radio: x=5.3112
Altura: h=11.2842

1.6 Preguntas

1. ¿Cual etapa del proceso de resolución de un problema numérico requiere mayor atención?

2. ¿Qué conocimientos son necesarios para formular un modelo matemático?

3. En el ejemplo del recipiente, ¿Cual sería la desventaja de intentar obtener experimentalmente la solución mediante prueba y error en lugar de analizar el modelo matemático?

4. ¿Que es más crítico: el error de truncamiento o el error de redondeo?

5. ¿Cuál es la ventaja de instrumentar computacionalmente un método numérico?

6. ¿Por que es importante validar los resultados obtenidos?

2 TIPOS DE MÉTODOS NUMÉRICOS

Existen dos estrategias para diseñar métodos numéricos y es importante conocer sus características para elegirlos adecuadamente, así como su instrumentación computacional

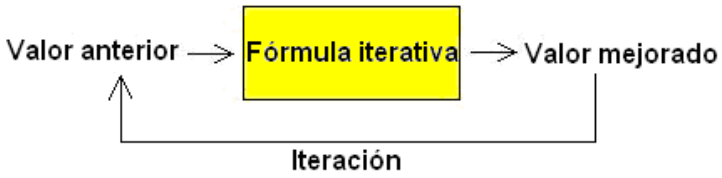
2.1 Métodos iterativos

Los métodos iterativos son procedimientos para acercarse a la respuesta mediante aproximaciones sucesivas. Estos métodos incluyen fórmulas que tienen la propiedad de producir un resultado más cercano a la respuesta a partir de un valor estimado inicial. El resultado obtenido se puede usar nuevamente como valor anterior para continuar mejorando la respuesta.

Se deben considerar algunos aspectos tales como la elección del valor inicial, la propiedad de convergencia de la fórmula y el criterio para terminar las iteraciones.

Estos métodos son auto-correctivos. La precisión de la respuesta está dada por la distancia entre el último valor calculado y la respuesta esperada. Esto constituye el error de truncamiento.

El siguiente gráfico describe la estructura de un método iterativo



Cada ciclo se denomina iteración. Si la fórmula converge, en cada iteración la respuesta estará más cerca del resultado buscado. Aunque en general no es posible llegar a la respuesta exacta, se puede acercar a ella tanto como lo permita la aritmética computacional del dispositivo de cálculo.

Ejemplo. *Instrumentar un método iterativo para calcular la raíz cuadrada* ***r*** *de un número real positivo* ***n*** *mediante operaciones aritméticas básicas*

Método Numérico

Se usará una fórmula que recibe un valor estimado para la raíz cuadrada y produce un valor más cercano a la respuesta. Si se usa repetidamente la fórmula cada resultado tenderá a un valor final que suponemos es la respuesta buscada. La obtención de estas fórmulas se realizará posteriormente.

Sean ***x****: valor estimado para la raíz* ***r***

Fórmula iterativa: $y = \frac{1}{2}(x + \frac{n}{x})$

Algoritmo

1. *Dados* ***n*** *y la precisión requerida* ***E***
2. *Elegir el valor inicial* ***x***
3. *Repetir*
4. *Calcular* $y = \frac{1}{2}(x + \frac{n}{x})$
5. *Asignar* ***x*** *a* ***y****.*
6. *Finalizar si* ***|x-y|*** *es menor que* ***E***
7. *El último valor* ***x*** *será un valor aproximado para la raíz* ***r*** *con precisión* ***E***

Ejemplo. *Calcular* $r = \sqrt{7}$ *con la fórmula iterativa anterior*

Usaremos $x = 3$ *como valor inicial*

Cálculos en MATLAB

```
>> format long
 >> n=7;
 >> x=3;
 >> y=0.5*(x+n/x)
 y =
   2.666666666666667
 >> x=y;
 >> y=0.5*(x+n/x)
 y =
   2.645833333333333
 >> x=y;
 >> y=0.5*(x+n/x)
 y =
   2.645751312335958
 >> x=y;
 >> y=0.5*(x+n/x)
 y =
   2.645751311064591
 >> x=y;
 >> y=0.5*(x+n/x)
 y =
   2.645751311064591
```

El último resultado tiene quince dígitos decimales que no cambian, por lo tanto podemos suponer que la respuesta tiene esa precisión. Se observa la rápida convergencia de la sucesión de números generad. Sin embargo es necesario verificar si la solución es aceptable pues si los números convergen a un valor, no necesariamente es la respuesta correcta

```
>> y^2
 ans =
     7.00000000000000
```

Se comprueba que el último valor calculado es la raíz cuadrada de 7

Ejemplo. *Suponer que se propone la siguiente fórmula iterativa para el ejemplo anterior:*

$$y = 0.4\left(x + \frac{n}{x}\right)$$

```
 >> n=7;
 >> x=3;
 >> y=0.4*(x+n/x)
 y =
   2.133333333333334
 >> x=y;
 >> y=0.4*(x+n/x)
 y =
   2.165833333333333
 >> x=y;
 >> y=0.4*(x+n/x)
```

```
y =
   2.159138258304477
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160468969251076
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160202499208563
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160255780068987
.
.
.
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160246899469289
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160246899469287
>> x=y;
>> y=0.4*(x+n/x)
y =
   2.160246899469287
>> y^2
ans =
   4.66666666666666
```

La fórmula converge pero el resultado final no es la raíz cuadrada de 7. Esto plantea la importancia de verificar la formulación del método numérico y la validación de la respuesta obtenida.

En general, los métodos numéricos se enfocan a resolver una clase o tipo de problemas. El ejemplo anterior es un caso particular del problema general: la solución de ecuaciones no lineales **f(x)=0**

2.1.1 Convergencia de los métodos iterativos

Es la propiedad que tienen las formulas iterativas de un método numérico para producir resultados cada vez más cercanos a la respuesta esperada.

Definición: Convergencia de un método iterativo

Sean **r** : Respuesta del problema (valor desconocido)
X_i : Valor calculado en la iteración **i** (valor aproximado)

Si un método iterativo converge, entonces

$$x_i \underset{i\to\infty}{\to} r \quad .$$

.

2.1.2 Error de truncamiento

La distancia entre la respuesta esperada y cada valor calculado con una fórmula iterativa se denomina error de truncamiento. Si la fórmula iterativa converge, la distancia entre valores consecutivos se debe reducir y se puede usar como una medida para el error de truncamiento.

Definición: Error de truncamiento

Sean **r** : Respuesta del problema (valor desconocido)
X_i : Valor calculado en la iteración **i** (valor aproximado)
X_{i+1}: Valor calculado en la iteración **i + 1** (valor aproximado)
Entonces
E_i = **r – X_i** : Error de truncamiento en la iteración **i**
E_{i+1} = **r – X_{i+1}**: Error de truncamiento en la iteración **i + 1**

2.1.3 Finalización de un proceso iterativo

Con la definición de convergencia se puede establecer un criterio para finalizar el proceso iterativo. Consideremos los resultados de dos iteraciones consecutivas: x_i, x_{i+1}

Si el método converge, $x_i \underset{i\to\infty}{\to} r$ y también $x_{i+1} \underset{i\to\infty}{\to} r$

Restando estas dos expresiones: $x_{i+1} - x_i \underset{i\to\infty}{\to} 0$, se puede establecer un criterio de convergencia

Definición: Criterio para finalizar un proceso iterativo (error absoluto)

Sea **E** algún valor positivo arbitrariamente pequeño.
Si el método converge, se cumplirá que a partir de alguna iteración **i:**

$$|X_{i+1} - X_i| < E \quad .$$

Este valor **E** es el **error de truncamiento absoluto** y puede usarse como una medida para la precisión de la respuesta calculada.

La precisión utilizada en los cálculos aritméticos, debe ser coherente con el error de truncamiento del método numérico y con los errores inherentes en el modelo matemático.

Adicionalmente, es necesario verificar que la respuesta final sea aceptable para el modelo matemático y para el problema que se está resolviendo.

***Ejemplo**. Se desea que la respuesta calculada para un problema con un método iterativo tenga un error absoluto menor que **0.0001**. Entonces el algoritmo deberá terminar cuando se cumpla que $|X_{i+1} - X_i| < 0.0001$. Los cálculos deben realizarse al menos con la misma precisión.*

Para que el criterio del error sea independiente de la magnitud del resultado, conviene usar la definición del error relativo:

Definición: Criterio para finalizar un proceso iterativo (error relativo)

Sea **e** algún valor positivo arbitrariamente pequeño.
Si el método converge, se cumplirá que a partir de alguna iteración **i:**

$$\frac{|x_{i+1} - x_i|}{|x_{i+1}|} < e \quad .$$

Este valor **e** es el **error de truncamiento relativo** y puede usarse como una medida para la precisión de la respuesta calculada, independiente de la magnitud de la respuesta. Para calcular el error relativo se toma el último valor como si fuese exacto.

***Ejemplo**. Se desea que la respuesta calculada para un problema con un método iterativo tenga un error relativo menor que **0.1%.** Entonces el algoritmo deberá terminar cuando se cumpla que*

$$\frac{|x_{i+1} - x_i|}{|x_{i+1}|} < 0.001.$$

2.1.4 Eficiencia de un método iterativo

Sean E_i , E_{i+1} los errores de truncamiento en las iteraciones **i**, **i+1** respectivamente. Se supondrá que los valores del error son pequeños y menores a 1.

Si a partir de alguna iteración **i** esta relación puede especificarse como $|E_{i+1}| = k\,|E_i|$, siendo **k** alguna constante positiva menor que uno, entonces se dice que la convergencia es **lineal** o de **primer orden** y **k** es el factor de convergencia. Se puede usar la notación **O( )** y escribir $E_{i+1} = O(E_i)$ para expresar de una manera simple el orden de esta relación, y se lee "orden de".

Si en un método esta relación es más fuerte tal como $E_{i+1} = O(E_i^2)$ entonces el error se reducirá más rápidamente y se dice que el método tiene convergencia **cuadrática** o de **segundo orden**.

Definición: Orden de convergencia de un método iterativo

Sean E_i , E_{i+1} los errores en las iteraciones consecutivas **i**, **i + 1** respectivamente
Si se pueden relacionar estos errores en la forma:

$$E_{i+1} = O(E_i^n)$$

Entonces se dice que el método iterativo tiene convergencia de orden **n**.

Si un método iterativo tiene convergencia mayor que lineal, entonces si el método converge, lo hará más rápidamente. .

2.1.5 Elección del valor inicial

Los métodos iterativos normalmente requieren que el valor inicial sea elegido apropiadamente. Si es elegido al azar, puede ocurrir que no se produzca la convergencia.

Si el problema es simple, mediante algún análisis previo puede definirse una **región de convergencia** tal que si el valor inicial y los valores calculados en cada iteración permanecen en esta región, el método converge.

2.1.6 Preguntas

Conteste las siguientes preguntas

1. ¿Por que el error de redondeo no debe ser mayor que el error de truncamiento?

2. Una ventaja de los métodos iterativos es que son auto-correctivos, es decir que si se introduce algún error aritmético en una iteración, en las siguientes puede ser corregido. ¿Cuando no ocurriría esta auto-corrección?

3. El ejemplo del método numérico para calcular $\sqrt{7}$ produce una secuencia numérica. ¿Le parece que la convergencia es lineal o cuadrática?

2.2 Métodos directos

Son procedimientos para obtener resultados realizando una secuencia finita de operaciones aritméticas. La cantidad de cálculos aritméticos depende del tamaño del problema. El resultado obtenido será exacto siempre que se puedan conservar en forma exacta los valores calculados en las operaciones aritméticas, caso contrario se introducirán los **errores de redondeo**.

Ejemplo. *Instrumentar un método directo para resolver un sistema triangular inferior de ecuaciones lineales.*

Modelo matemático

$$
\begin{aligned}
& a_{1,1}x_1 && = b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 && = b_2 \\
& a_{3,1}x_1 + a_{3,2}x_2 + a_{3,3}x_3 && = b_3 \\
& \ldots . && \\
& a_{n,1}x_1 + a_{n,2}x_2 + a_{x,3}x_3 + \ldots + a_{n,n}x_n && = b_n
\end{aligned}
$$

Siendo: $a_{i,j}$: *coeficientes (datos)*
b_i : *constantes (datos)*
x_i : *variables (resultados)*

La formulación matemática del método se obtiene despejando sucesivamente x_i *de la ecuación* i

$$x_1 = \frac{1}{a_{1,1}}(b_1)$$

$$x_2 = \frac{1}{a_{2,2}}(b_2 - a_{2,1}x_1)$$

$$x_3 = \frac{1}{a_{3,3}}(b_3 - a_{3,1}x_1 - a_{3,2}x_2)$$

$$\ldots$$

En general:

$$x_i = \frac{1}{a_{i,i}}(b_i - a_{i,1}x_1 - a_{i,2}x_2 - \ldots - a_{i,i-1}x_{i-1}),\ i = 2, 3, \ldots, n,\ a_{i,i} \neq 0$$

Algoritmo

1) *Dados* n, a, b
2) *Calcular* $x_1 = \frac{1}{a_{1,1}}(b_1)$
3) *Para* $i=2,3,\ldots, n$
4) *Calcular* $x_i = \frac{1}{a_{i,i}}(b_i - a_{i,1}x_1 - a_{i,2}x_2 - \ldots - a_{i,i-1}x_{i-1})$
5) *fin*
6) *El vector* x *contendrá la solución exacta*

Este algoritmo es un caso particular del problema general: solución de un sistema de ecuaciones lineales. Los métodos numéricos normalmente se desarrollan para resolver una clase o tipo general de problemas. La instrumentación puede hacerse mediante un programa, pero es más conveniente definirlo como una función en MATLAB para estandarizar la entrada y salida de variables.

Instrumentación computacional en MATLAB

Para que la instrumentación sea general es preferible diseñar cada método numérico independientemente de los datos de un problema particular. La instrumentación del método numérico del ejemplo anterior se hará mediante una función en MATLAB.

El nombre para la función será **"triangular".** La función recibirá como datos la matriz de coeficientes **a** y el vector de constantes **b** y producirá como resultado el vector solución **v** (vector columna)

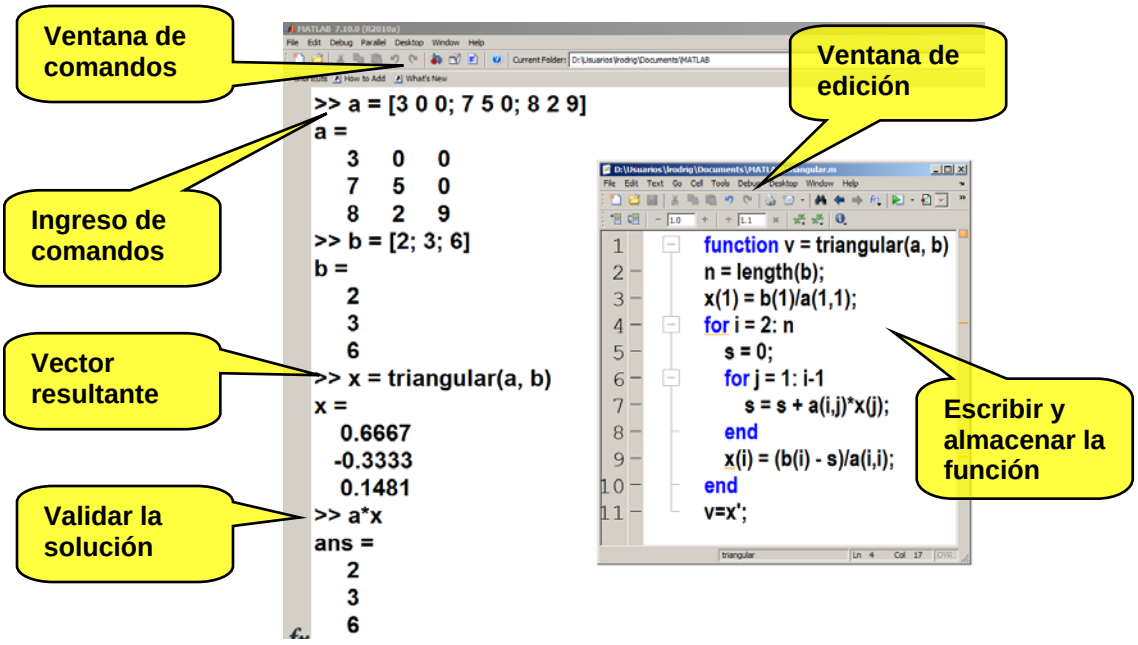
```
function v = triangular(a, b)
n = length(b);
x(1) = b(1)/a(1,1);
for i = 2: n
   s = 0;
   for j = 1: i-1
      s = s + a(i,j)*x(j);
   end
   x(i) = (b(i) - s)/a(i,i);
end
v=x';                                   (se define el vector columna resultante)
```

Ejemplo. *Escribir los comandos para resolver el siguiente problema usando la función anterior*

$$3x_1 = 2$$
$$7x_1 + 5x_2 = 3$$
$$8x_1 + 2x_2 + 9x_3 = 6$$

```
>> a = [3 0 0; 7 5 0; 8 2 9]      Definir la matriz
>> b = [2; 3; 6]                  Definir el vector de constantes
>> x = triangular(a, b)           Uso de la función
```

Interacción con MATLAB



2.2.1 Error de redondeo

Los métodos numéricos operan con datos que pueden ser inexactos y con dispositivos para representar a los números reales. El error de redondeo se atribuye a la imposibilidad de almacenar todas las cifras de estos números y a la imprecisión de los instrumentos de medición con los cuales se obtienen los datos.

Definición: Error de redondeo absoluto

Sean	**X:** Valor exacto	(normalmente desconocido)
	$\overline{\mathbf{X}}$ **:** Valor aproximado	(observado o calculado)
	$\mathbf{E = X - \overline{X}}$	Error de redondeo .

Definición: Error de redondeo relativo

Sean **X:** Valor exacto (normalmente desconocido)
$\overline{\mathbf{X}}$ **:** Valor aproximado (observado o calculado)
E: Error de redondeo

$\mathbf{e} = \frac{\mathbf{E}}{\mathbf{X}} \cong \frac{\mathbf{E}}{\overline{\mathbf{X}}}$: Error de redondeo relativo. (**X,** $\overline{\mathbf{X}}$ diferentes de cero)

Debido a que normalmente no es posible calcular exactamente el valor de **E**, se debe intentar al menos acotar su valor.

2.2.2 Error en la representación de los números reales

Las operaciones aritméticas pueden producir resultados que no se pueden representar exactamente en los dispositivos de almacenamiento. Si estos errores se producen en forma recurrente entonces el error propagado pudiera crecer en forma significativa dependiendo de la cantidad de operaciones requeridas. Esta cantidad de operaciones está determinada por la eficiencia del algoritmo.

Ejemplo. *Suponga que un dispositivo puede almacenar únicamente los cuatro primeros dígitos decimales de cada número real y trunca los restantes (esto se llama redondeo inferior).*
Se requiere almacenar el número:

$\mathbf{X = 536.78}$

Primero expresemos el número en forma normalizada, es decir sin enteros y ajustando su magnitud con potencias de 10:

$\mathbf{X = 0.53678x10^3}$

Ahora descomponemos el número en dos partes

$\mathbf{X = 0.5367x10^3 + 0.00008x10^3}$

El valor almacenado es un valor aproximado

$\overline{\mathbf{X}} \mathbf{= 0.5367x10^3}$

El error de redondeo por la limitación del dispositivo de almacenamiento es

$\mathbf{E = 0.00008x10^3 = 0.8x10^{3-4} = 0.8x10^{-1}}$

En general, si **n** es la cantidad de enteros del número normalizado con potencias de 10, y **m** es la cantidad de cifras decimales que se pueden almacenar en el dispositivo, entonces si se truncan los decimales sin ajustar la cifra anterior, el error de redondeo absoluto está acotado por:

$$|E| < 1 * 10^{n-m}$$

Mientras que el error relativo:

$$|e| < \frac{\max(|E|)}{\min(|X|)} = \frac{1*10^{n-m}}{0.1*10^{n}} = 10*10^{-m}$$ (Solo depende del almacenamiento)

2.2.3 Error de redondeo en las operaciones aritméticas

En los métodos directos debe considerarse el error en las operaciones aritméticas cuando la cantidad de cálculos requeridos es significativo

a) Error de redondeo en la suma

Sean **X, Y :** Valores exactos

$\overline{X}$, $\overline{Y}$: Valores aproximados

Con la definición de error de redondeo

$E_X = X - \overline{X}$, $E_Y = Y - \overline{Y}$

$S = X + Y$

$S = (\overline{X} + E_X) + (\overline{Y} + E_Y) = (\overline{X} + \overline{Y}) + (E_X + E_Y)$

$\overline{S} = \overline{X} + \overline{Y}$ Valor que se almacena

Error de redondeo absoluto en la suma

$E_{X+Y} = E_X + E_Y$

$|E_{X+Y}| \leq |E_X| + |E_Y|$

Error de redondeo relativo en la suma

$$e_{X+Y} = \frac{E_{X+Y}}{\overline{X} + \overline{Y}} = \frac{E_X + E_Y}{\overline{X} + \overline{Y}} = \frac{E_X}{\overline{X} + \overline{Y}} + \frac{E_Y}{\overline{X} + \overline{Y}} = \frac{\overline{X}}{\overline{X} + \overline{Y}} \frac{E_X}{\overline{X}} + \frac{\overline{Y}}{\overline{X} + \overline{Y}} \frac{E_Y}{\overline{Y}}$$

$$e_{X+Y} = \frac{\overline{X}}{\overline{X} + \overline{Y}} e_x + \frac{\overline{Y}}{\overline{X} + \overline{Y}} e_Y$$

$$|e_{X+Y}| \leq \left|\frac{\overline{X}}{\overline{X} + \overline{Y}} e_x\right| + \left|\frac{\overline{Y}}{\overline{X} + \overline{Y}} e_Y\right|$$

b) Error de redondeo en la multiplicación

$P = X\,Y$

$P = (\overline{X} + E_X)(\overline{Y} + E_Y) = \overline{X}\,\overline{Y} + \overline{X}\,E_Y + \overline{Y}\,E_X + E_X E_Y$

$P = \overline{X}\,\overline{Y} + \overline{X}\,E_Y + \overline{Y}\,E_X$ El último término se descarta por ser muy pequeño

$\overline{P} = \overline{X}\,\overline{Y}$ Valor que se almacena

Error de redondeo absoluto en la multiplicación

$E_{XY} = \overline{X}\,E_Y + \overline{Y}\,E_X$

$|E_{XY}| \leq |\overline{X}\,E_Y| + |\overline{Y}\,E_X|$

La magnitud del error de redondeo en la multiplicación puede ser tan grande como la suma de los errores de redondeo de los operandos ponderada por cada uno de sus respectivos valores.

Error de redondeo relativo en la multiplicación

$$e_{XY} = \frac{E_{XY}}{\overline{X}\,\overline{Y}} = \frac{\overline{X}E_Y + \overline{Y}E_X}{\overline{X}\,\overline{Y}} = \frac{\overline{X}E_Y}{\overline{X}\,\overline{Y}} + \frac{\overline{Y}E_X}{\overline{X}\,\overline{Y}} = \frac{E_Y}{\overline{Y}} + \frac{E_X}{\overline{X}}$$

$e_{XY} = e_x + e_Y$

$|e_{XY}| \leq |e_x| + |e_Y|$

En general, si los valores de los operandos tienen ambos el mismo signo se puede concluir que la operación aritmética de multiplicación puede propagar más error de redondeo que la suma. Adicionalmente, si el resultado de cada operación aritmética debe almacenarse, hay que agregar el error de redondeo debido a la limitación del dispositivo de almacenamiento.

2.2.4 Propagación del error de redondeo en las operaciones aritméticas

Algunos casos de interés

a) Suma de números de diferente magnitud

Es suficiente considerar tres números: **X, Y, Z**. Suponer por simplicidad que los datos son valores positivos exactos, por lo tanto $e_X = 0, e_Y = 0, e_z = 0$

$$S = X + Y + Z, \quad \text{con} \quad X > Y > Z$$

Suponer que la suma se realiza en el orden escrito

$$S = (X + Y) + Z$$

$$e_{X+Y} = \frac{\overline{X}}{\overline{X} + \overline{Y}} e_X + \frac{\overline{Y}}{\overline{X} + \overline{Y}} e_Y + r_1 = r_1$$

r_1: error de redondeo al almacenar la suma

$$e_{(X+Y)+Z} = \frac{\overline{X} + \overline{Y}}{\overline{X} + \overline{Y} + \overline{Z}} e_{X+Y} + \frac{\overline{Z}}{\overline{X} + \overline{Y} + \overline{Z}} e_Z + r_2 = \frac{\overline{X} + \overline{Y}}{\overline{X} + \overline{Y} + \overline{Z}} r_1 + r_2 = \frac{(\overline{X} + \overline{Y}) r_1 + (\overline{X} + \overline{Y} + \overline{Z}) r_2}{\overline{X} + \overline{Y} + \overline{Z}}$$

r_2: error de redondeo al almacenar la suma

Si cada resultado se almacena en un dispositivo que tiene **m** cifras su cota de error de redondeo:

$$| r_1, r_2 | < 10 * 10^{-m}$$

$$| e_{(X+Y)+Z} | < \frac{(2\overline{X} + 2\overline{Y} + \overline{Z}) 10 * 10^{-m}}{\overline{X} + \overline{Y} + \overline{Z}}$$

Se puede concluir que la suma de números debe realizarse comenzando con los números de menor magnitud

b) Resta de números con valores muy cercanos

Suponer que se deben restar dos números muy cercanos **X=578.1**, **Y=577.8** con error de redondeo en el segundo decimal: $E_X = E_Y = 0.05$ aproximadamente (ambos errores pueden ser del mismo signo o de diferente signo, depende de la forma como se obtuvieron los datos **X**, **Y**)

$$e_X = \frac{E_X}{\overline{X}} \cong 0.000086 = 0.0086\%, \quad e_Y = \frac{E_Y}{\overline{Y}} \cong 0.000086 = 0.0086\%$$

Cota para el error de redondeo relativo:

$$e_{X-Y} = \frac{E_{X-Y}}{\overline{X} - \overline{Y}} = \frac{E_X - E_Y}{\overline{X} - \overline{Y}} = \frac{E_X}{\overline{X} - \overline{Y}} - \frac{E_Y}{\overline{X} - \overline{Y}}$$

$$| e_{X-Y} | \leq | \frac{E_X}{\overline{X} - \overline{Y}} | + | - \frac{E_Y}{\overline{X} - \overline{Y}} |$$

$$| e_{X-Y} | \leq | \frac{0.05}{578.1 - 577.8} | + | - \frac{0.05}{578.1 - 577.8} | \cong 0.3333 = 33.33\%$$

El aumento en la cota del error en el resultado es muy significativo con respecto a los operandos. Se concluye que se debe evitar restar números cuya magnitud sea muy cercana.

Adicionalmente habría que aumentar el efecto del error de redondeo **r** al almacenar el resultado.

2.2.5 Eficiencia de los métodos directos

La eficiencia de un algoritmo está relacionada con el tiempo necesario para obtener la solución. Este tiempo depende de la cantidad de operaciones aritméticas que se deben realizar. Así, si se tienen dos algoritmos para resolver un mismo problema, es más eficiente el que requiere menos operaciones aritméticas para producir el mismo resultado. Adicionalmente, el algoritmo más eficiente acumulará menos error si los resultados son números reales que no pueden representarse en forma exacta en el dispositivo de cálculo.

Sea **n** el tamaño del problema, y **T(n)** la eficiencia del algoritmo (cantidad de operaciones aritméticas requeridas). Para obtener **T(n)** se pueden realizar pruebas en el computador con diferentes valores de **n** registrando el tiempo de ejecución. Siendo este tiempo proporcional a la cantidad de operaciones aritméticas que se realizaron, se puede estimar la función **T(n)**.

Esta forma experimental para determinar **T(n)** tiene el inconveniente de requerir la instrumentación computacional del algoritmo para realizar las pruebas. Es preferible conocer la eficiencia del algoritmo antes de invertir esfuerzo en la programación computacional.

Para determinar **T(n)** se puede analizar la formulación matemática del método numérico o la estructura del algoritmo.

Ejemplo. *El siguiente algoritmo calcula la suma de los cubos de los primeros* ***n*** *números naturales. Encontrar* ***T(n)***

Sea ***T*** *la cantidad de sumas que se realizan*

```
. . .
s ← 0
Para i=1, 2, ..., n
  s ← s + i^3
fin
. . .
```

La suma está dentro de una repetición que se realiza ***n*** *veces, por lo tanto,*

$$T(n) = n$$

Ejemplo. *El siguiente algoritmo suma los elementos de una matriz cuadrada* ***a*** *de orden* ***n****. Encontrar* ***T(n)***

Sea ***T*** *la cantidad de sumas*

```
. . .
s←0
Para i=1, 2, ..., n
  Para j=1, 2, ..., n
    s ← s + a_{i,j}
  fin
fin
. . .
```

La suma está incluida en una repetición doble. La variable ***i,*** *cambia* ***n*** *veces y para cada uno de sus valores, la variable* ***j*** *cambia* ***n*** *veces. Por lo tanto.*

$$T(n) = n^2$$

Ejemplo. *El siguiente algoritmo es una modificación del anterior. Suponga que se desea sumar únicamente los elementos de la sub-matriz triangular superior. Obtener* ***T(n)***

```
. . .
s←0
Para i=1, 2, ..., n
  Para j=i, i+1, ..., n
    s ← s + a_{i,j}
  fin
fin
. . .
```

Si no es evidente la forma de ***T(n)****, se puede recorrer el algoritmo y anotar la cantidad de sumas que se realizan*

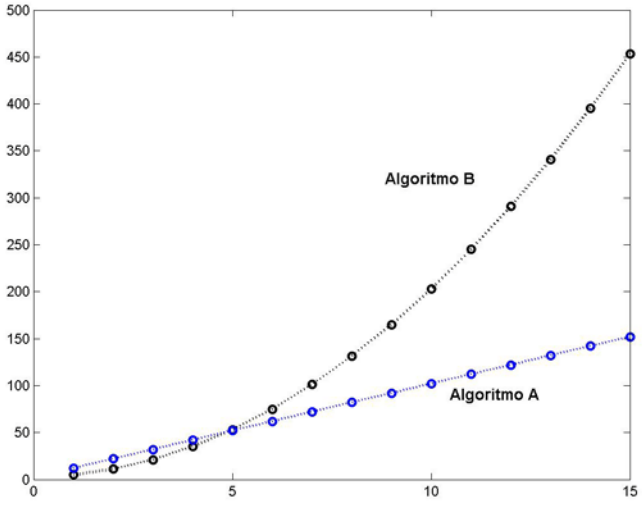
i	***j***
1	*n*
2	*n-1*
3	*n-2*
...	*...*
n-1	*2*
n	*1*

Entonces, $\mathbf{T(n) = 1 + 2 + ... + n = \frac{n}{2}(n+1) = \frac{n^2}{2} + \frac{n}{2}}$ *(suma de una serie aritmética)*

2.2.6 La notación O( )

Supongamos que para resolver un problema se han diseñado dos algoritmos: **A** y **B**, con eficiencias $\mathbf{T_A(n) = 10n+2}$, $\mathbf{T_B(n) = 2n^2 + 3}$, respectivamente. ¿Cual algoritmo es más eficiente?.

Para valores pequeños de **n**, $\mathbf{T_B(n) < T_A(n)}$, pero para valores grandes de **n**, $\mathbf{T_A(n) < T_B(n)}$. Es de interés práctico determinar la eficiencia de los algoritmos para valores grandes de **n**, por lo tanto el algoritmo **A** es más eficiente que el algoritmo **B** como se puede observar en el gráfico:



Si **T(n)** incluye términos con **n** que tienen diferente orden, es suficiente considerar el término de mayor orden pues es el que determina la eficiencia del algoritmo cuando **n** es grande. No es necesario incluir los coeficientes y las constantes.

Ejemplo. *Suponga* ***T(n) = n^2 + n + 10.*** *Evaluar* ***T*** *para algunos valores de* ***n***

T(2) = 4 + 2 + 10
T(5) = 25 + 5 + 10
T(20) = 400 + 20 + 10
T(100) = 10000 + 100 + 10

Se observa que a medida que ***n*** *crece,* ***T*** *depende principalmente de* ***n^2.*** *Este hecho se puede expresar usando la notación* ***O( )*** *la cual indica el "orden" de la eficiencia del algoritmo, y se puede escribir:* ***T(n) = O(n^2)*** *lo cual significa que la eficiencia es proporcional a* ***n^2****, y se dice que el algoritmo tiene eficiencia cuadrática o de segundo orden.*

En general, dado un problema de tamaño **n**, la medida de la eficiencia **T(n)** de un algoritmo se puede expresar con la notación **O(g(n))** siendo **g(n)** alguna expresión tal como: **n, n^2, n^3, ..., log(n), n log(n), ..., 2^n, n!, ...** la cual expresa el orden de la cantidad de operaciones aritméticas que requiere el algoritmo.

Es de interés medir la eficiencia de los algoritmos para problemas de **tamaño grande**, pues para estos valores de **n** se debe conocer la eficiencia. En el siguiente cuadro se ha tabulado **T(n)** con algunos valores de **n** y para algunas funciones típicas **g(n)**.

Tabulación de T(n) con algunos valores de n para algunas funciones típicas g(n)

n	[log(n)]	n	[n log(n)]	n^2	n^3	2^n	n!
1	0	1	0	1	1	2	1
3	1	3	3	9	27	8	6
5	1	5	8	25	125	32	120
7	1	7	13	49	343	128	5040
9	2	9	19	81	729	512	3.62×10^{5}
11	2	11	26	121	1331	2048	3.99×10^{7}
13	2	13	33	169	2197	8192	6.22×10^{9}
15	2	15	40	225	3375	32768	1.30×10^{12}
17	2	17	48	289	4913	1.31×10^{5}	3.55×10^{14}
19	2	19	55	361	6859	5.24×10^{5}	1.21×10^{17}
21	3	21	63	441	9261	2.09×10^{6}	5.10×10^{19}
23	3	23	72	529	12167	8.38×10^{6}	2.58×10^{22}
25	3	25	80	625	15625	3.35×10^{7}	1.55×10^{25}
50	3	50	195	2500	125000	1.12×10^{15}	3.04×10^{64}
100	4	100	460	10000	1000000	1.26×10^{30}	9.33×10^{157}

Los algoritmos en las dos últimas columnas son de tipo **exponencial** y **factorial** respectivamente. Se puede observar que aún con valores relativamente pequeños de **n** el valor de **T(n)** es extremadamente alto. Los algoritmos con este tipo de eficiencia se denominan **no-factibles** pues ningún computador actual pudiera calcular la solución en un tiempo aceptable para valores de **n** grandes

La mayoría de los algoritmos que corresponden a los métodos numéricos son de tipo polinomial con **g(n) = n, n^2, n^3** .

2.2.7 Ejercicios

1. Encuentre la cota del error relativo en la siguiente operación aritmética:

$T = X(Y - Z)$

El error de redondeo relativo de los operandos es respectivamente $\mathbf{e_X}$, $\mathbf{e_Y}$, $\mathbf{e_z}$, y el error de redondeo relativo en el dispositivo de almacenamiento es $\mathbf{r_m}$

a) Primero se realiza la multiplicación y luego la resta
b) Primero se realiza la resta y luego la multiplicación

2. Suponga que tiene tres algoritmos: A, B, C con eficiencia respectivamente:

$T_A(n) = 5n + 50$
$T_B(n) = 10n\ ln(n) + 5$
$T_C(n) = 3n^2 + 1$

a) Determine **n** a partir del cual A es más eficiente que B
b) Determine **n** a partir del cual B es más eficiente que C
c) Coloque los algoritmos ordenados según el criterio de eficiencia establecido

3. Exprese la eficiencia de los algoritmos A, B, C con la notación **O( )**

4. Los computadores actuales pueden realizar 100 millones de operaciones aritméticas en un segundo. Calcule cuanto tiempo tardaría este computador para resolver un problema de tamaño **n=50** si el algoritmo es de tipo:

a) Polinomial de tercer grado
b) Exponencial
c) Factorial

5. Determine la función de eficiencia **T(n)** del algoritmo **triangular** incluido en el capítulo 1 y exprésela con la notación **O( )**. Suponga que es de interés conocer la cantidad total de ciclos que se realizan.

6. Dado el siguiente algoritmo

```
Ingresar n
Mientras n>0 repita
   d ← mod(n, 2)        Produce el residuo entero de la división n/2
   n ← fix(n/2)         Asigna el cociente entero de la división n/2
   Mostrar d
fin
```

a) Recorra el algoritmo con **n = 73**
b) Suponga que **T(n)** representa la cantidad de operaciones aritméticas de división que se realizan para resolver el problema de tamaño **n**. Encuentre **T(n)** y exprésela con la notación **O( )** Para obtener **T(n)** observe el hecho de que en cada ciclo el valor de **n** se reduce aproximadamente a la mitad.

3 RAÍCES REALES DE ECUACIONES NO-LINEALES

Sea **f: R→R**. Dada la ecuación **f(x)** = **0**, se debe encontrar un valor real **r** tal que **f(r)** = **0**. Entonces **r** es una raíz real de la ecuación

Si no es posible obtener la raíz directamente, entonces se debe recurrir a los métodos numéricos iterativos para calcular **r** en forma aproximada con alguna precisión controlada. Se han creado muchos métodos numéricos para resolver este problema clásico, pero con el uso de computadoras para el cálculo, conviene revisar solamente algunos de estos métodos que tengan características significativamente diferentes.

3.1 Método de la bisección

Sea **f: R→R**. Suponer que **f** es continua en **[a, b],** y que además **f(a)** y **f(b)** tienen signos diferentes. Por continuidad, el intervalo **(a, b)** contendrá al menos una raíz real.

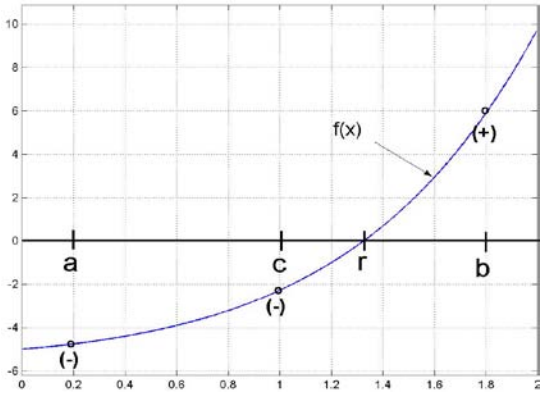
El siguiente teorema establece la existencia de la raíz **r**:

Teorema de Bolzano: Si una función **f** es continua en un intervalo **[a, b]** y **f(a)** tiene signo diferente que **f(b)**, entonces existe por lo menos un punto **r** en **(a, b)** tal que **f(r)=0**.

Si además **f'(x)** no cambia de signo en el intervalo **[a, b]**, entonces la solución es única.

El método de la bisección es un método simple y convergente para calcular **r**. Consiste en calcular el punto medio **c=(a+b)/2** del intervalo **[a, b]** y sustituirlo por el intervalo **[c, b]** ó **[a, c]** dependiendo de cual contiene a la raíz **r**. Este procedimiento se repite hasta que la distancia entre **a** y **b** sea muy pequeña, entonces el último valor calculado **c** estará muy cerca de **r**.

Interpretación gráfica del método de la bisección

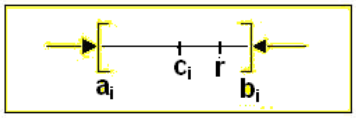


En la figura se puede observar que luego de haber calculado **c,** para la siguiente iteración debe sustituirse el intervalo **[a, b]** por **[c, b]** debido a que **f(a)** y **f(c)** tienen igual signo y por lo tanto la raíz estará en el intervalo **[c, b]**

3.1.1 Convergencia del método de la bisección

Sean a_i, b_i, c_i los valores de **a, b, c** en cada iteración **i=1, 2, 3, . . .** respectivamente

El método de la bisección genera una sucesión de intervalos $[a, b], [a_1, b_1], [a_2, b_2], \ldots, [a_i, b_i]$ tales que $a \le a_1 \le a_2 \ldots \le a_i$ constituyen una sucesión creciente y $b \ge b_1 \ge b_2 \ldots, \ge b_i$ una sucesión decreciente con $a_i < b_i$. Además por definición del método: $c_i, r \in [a_i, b_i]$ en cada iteración **i**



Sean $d_i = b_i - a_i$ longitud del intervalo $[a_i, b_i]$ en la iteración **i=1, 2, 3, . . .**
$d = b - a$ longitud del intervalo inicial

Recorrido de las iteraciones

Iteración	Longitud del intervalo
1	$d_1 = d/2$
2	$d_2 = d_1/2 = d/2^2$
3	$d_3 = d_2/2 = d/2^3$
4	$d_4 = d_3/2 = d/2^4$
. . .	. . .
i	$d_i = d/2^i$

Entonces

$$\frac{d}{2^i} \xrightarrow[i\to\infty]{} 0 \Rightarrow d_i \xrightarrow[i\to\infty]{} 0 \Rightarrow a_i \xrightarrow[i\to\infty]{} b_i \Rightarrow c_i \xrightarrow[i\to\infty]{} r \Rightarrow \exists i>0 \; | c_i - r | < E$$ para algún valor positivo **E**

Suponer que se desea que el último valor calculado c_i tenga precisión **E = 0.001,** entonces si el algoritmo termina cuando $b_i - a_i < E$, se cumplirá que $|c_i - r| < E$ y c_i será una aproximación para **r** con un error menor que 0.0001

Se puede predecir el número de iteraciones que se deben realizar con el método de la Bisección para obtener la respuesta con una precisión requerida **E**:

En la iteración **i**: $d_i = d/2^i$
Se desea terminar cuando: $d_i < E$
Entonces se debe cumplir $d/2^i < E$
De donde se obtiene: $i > \dfrac{\log(d/E)}{\log(2)}$

Ejemplo. *La ecuación* $f(x) = x e^x - \pi = 0$ *tiene una raíz real en el intervalo* ***[0, 2].*** *Determine cuantas iteraciones deben realizarse con el método de la bisección para obtener un resultado con precisión* ***E=0.0001****.*

El número de iteraciones que deberán realizarse es:

$i > \log(2/0.0001)/\log(2) \Rightarrow i > 14.287 \Rightarrow$ ***15 iteraciones***

3.1.2 Algoritmo del método de la bisección

Calcular una raíz **r** real de la ecuación **f(x) = 0** con precisión **E**.
f es contínua en un intervalo **[a, b]** tal que **f(a)** y **f(b)** tienen signos diferentes

1) Defina **f**, el intervalo inicial **[a, b]** y la precisión requerida **E**
2) Calcule el punto central del intervalo: **c=(a+b)/2**
3) Si **f(c)=0**, **c** es la raíz y termine
4) Si la raíz se encuentra en el intervalo **[a, c]**, sustituya **b** por **c**
5) Si la raíz se encuentra en el intervalo **[c, b]** sustituya **a** por **c**
6) Repita los pasos 2), 3), 4), 5) hasta que la longitud del intervalo **[a, b]** sea menor que **E**.

El último valor calculado **c** estará al menos a una distancia **E** de la raíz **r**.

Ejemplo. *Calcule una raíz real de* $f(x) = x e^x - \pi = 0$ *en el intervalo* ***[0, 2] con precisión 0.01***

La función ***f*** *es continua y además* ***f(0)<0***, ***f(2)>0***, *por lo tanto la ecuación* ***f(x)=0*** *debe contener alguna raíz real en el intervalo* ***[0, 2]***

Cantidad de iteraciones

$$i > \frac{\log(d/E)}{\log(2)} = \frac{\log(1/0.01)}{\log(2)} = 7.6439 \quad \Rightarrow \quad \textbf{8 iteraciones}$$

Tabulación de los cálculos para obtener la raíz con el método de la Bisección

iteración	*a*	*b*	*c*	*sign(f(a))*	*sign(f(c))*
inicio	***0***	***2***	***1***	**-**	**-**
1	***1***	***2***	***1.5***	**-**	**+**
2	***1***	***1.5***	***1.25***	**-**	**+**
3	***1***	***1.25***	***1.125***	**-**	**+**
4	***1***	***1.125***	***1.0625***	**-**	**-**
5	***1.0625***	***1.125***	***1.0938***	**-**	**+**
6	***1.0625***	***1.0938***	***1.0781***	**-**	**+**
7	***1.0625***	***1.0781***	***1.0703***	**-**	**-**
8	***1.0703***	***1.0781***	***1.0742***		

En la octava iteración:

$b - a = 1.0781 - 1.0703 = 0.0078 \Rightarrow |r - c| < 0.01$

r = 1.074 con error menor que **0.01**

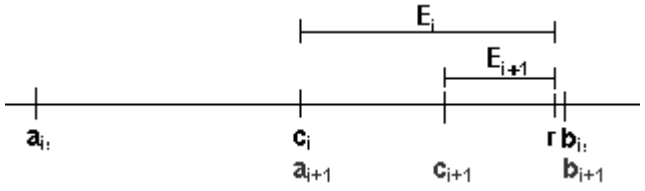
En la última iteración se observa que el intervalo que contiene a la raíz se ha reducido a ***[1.0703, 1.0781]****, por lo tanto el último valor calculado de* ***c = 1.074*** *debe estar cerca de* ***r*** *con una distancia menor que* ***0.01***

3.1.3 Eficiencia del método de la bisección

Suponer el caso más desfavorable, en el que **r** está muy cerca de uno de los extremos del intervalo **[a, b]**:

Sean $E_i = r - c_i$: error en la iteración **i**

$E_{i+1} = r - c_{i+1}$: error en la iteración **i+1**



En cada iteración la magnitud del error se reduce en no más de la mitad respecto del error en la iteración anterior: $E_{i+1} \le \frac{1}{2} E_i$. Esta es una relación lineal. Con la notación **O( )** se puede escribir: $E_{i+1} = O(E_i)$. Entonces, el método de la Bisección tiene **convergencia lineal** o de primer orden.

3.1.4 Instrumentación computacional del método de la bisección

Calcular una raíz **r** real de la ecuación **f(x) = 0**. **f** es contínua en un intervalo **[a, b]** tal que **f(a)** y **f(b)** tienen signos diferentes

Para instrumentar el algoritmo de este método se escribirá una función en MATLAB. El nombre será **bisección**. Recibirá como parámetros **f, a, b**, y entregará **c** como aproximación a la raíz **r**.

Criterio para salir: Terminar cuando la longitud del intervalo sea menor que un valor pequeño **e** especificado como otro parámetro para la función. Entonces el último valor **c** estará aproximadamente a una distancia **e** de la raíz **r**.

```
function c = biseccion(f, a, b, e)
while b-a >= e
  c=(a+b)/2;
  if f(c)==0
    return
  else
    if sign(f(a))==sign(f(c))
      a=c;
    else
      b=c;
    end
 end
end
```

Ejemplo. *Desde la ventana de comandos de MATLAB, use la función* ***bisección*** *para calcular una raíz real de la ecuación* $f(x) = xe^x - \pi = 0$. *Suponer que se desea que el error sea menor que 0.0001.*

Por simple inspección se puede observar que **f** es continua y además **f(0) < 0**, **f(2) > 0**. Por lo tanto se elije como intervalo inicial: **[0, 2].** También se puede previamente graficar **f**.

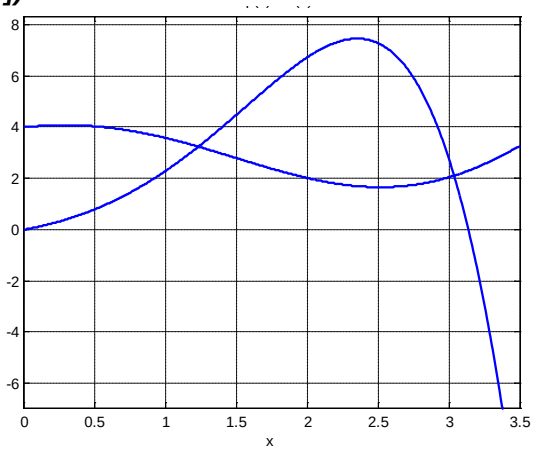
En la ventana de comandos de MATLAB se escribe:

```
>> f = 'x*exp(x)-pi';
>> c = biseccion(inline(f), 0, 2, 0.0001)
 c =
      1.073669433593750          Este es el resultado calculado
>> subs(f,'x',c)                 Al evaluar f(c) se obtiene un valor cercano a 0
 ans =
      6.819373368882609e-005
```

Ejemplo**. Encontrar las intersecciones en el primer cuadrante de los gráficos de las funciones:* ***f(x) = 4 + cos(x+1), g(x)=e^x sen(x).

Primero se grafican las funciones para visualizar las intersecciones:

```
>> f='4+x*cos(x+1)';
>> g='exp(x)*sin(x)';
>> ezplot(f,[0,3.5]),grid on,hold on
>> ezplot(g,[0,3.5])
```



Las intersecciones son las raíces de la ecuación ***h(x) = f(x) – g(x) = 0***
El cálculo de las raíces se realiza con el método de la bisección con un error menor a 0.0001

```
>> h='4+x*cos(x+1)-exp(x)*sin(x)';
>> c=biseccion(inline(h),1,1.5,0.0001)
  c =
    1.233726501464844
>> c=biseccion(inline(h),3,3.2,0.0001)
  c =
    3.040667724609375
```

3.2 Método del punto fijo

Sea **f: R→R.** Dada la ecuación **f(x)=0**, encuentre **r** tal que **f(r)=0**

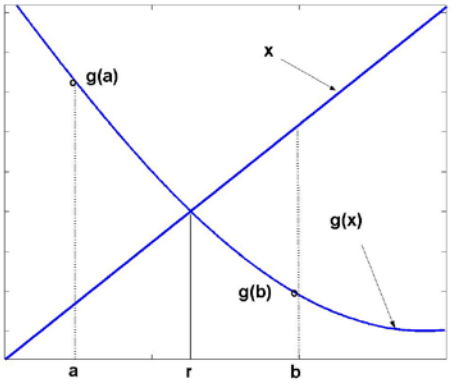
El método del punto fijo, también conocido como método de la Iteración funcional es el fundamento matemático para construir métodos eficientes para el cálculo de raíces reales de ecuaciones no lineales.

Este método consiste en re-escribir la ecuación **f(x) = 0** en la forma **x = g(x).** Esta nueva ecuación debe ser consistente con la ecuación original en el sentido de que debe satisfacerse con la misma raíz:

r = g(r) ⇔ f(r)=0

3.2.1 Existencia de una raíz real con el método del punto fijo

Suponer que la ecuación **f(x) = 0**, se sustituye por la ecuación **x = g(x).** Suponer además que **g** es una función continua en un intervalo **[a, b]** y que **g(a) > a** y **g(b) < b** como se muestra en el siguiente gráfico



Sea **h(x) = g(x) - x** una función, también continua, en el intervalo **[a, b]**
Entonces **h(a) = g(a) - a > 0,**
h(b) = g(b) - b < 0

Por la continuidad de **h**, (Teorema de Bolzano), existe algún valor **r** en el intervalo **[a, b]**, en el cual **h(r)=0**. Entonces **g(r) - r = 0**. Por lo tanto **g(r)=r ⇒ f(r)=0,** y **r** es una raíz real de **f(x)=0**

3.2.2 Algoritmo del punto fijo

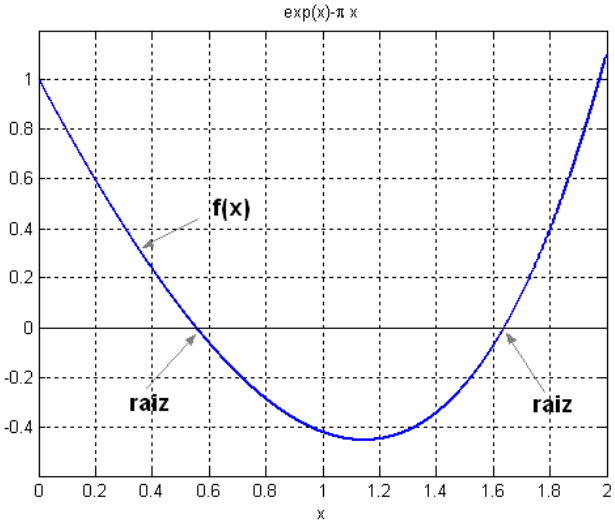
La ecuación **x=g(x)** se usa para construir una fórmula iterativa $x_{i+1} = g(x_i)$, **i = 0, 1, 2, 3, . . .** siendo x_0 el valor inicial, elegido con algún criterio. En la fórmula se usa un índice para numerar los valores calculados.

Con la fórmula iterativa se obtiene una sucesión de valores **x** esperando que tienda a un valor que satisfaga la ecuación **x = g(x)** lo cual implica que la ecuación **f(x) = 0** también se satisface.

1) Elegir un valor inicial x_0
2) Generar la sucesión de valores con la fórmula iterativa:
   $x_{i+1} = g(x_i)$, **i = 0, 1, 2, 3, ...**
3) Si el método converge, la sucesión x_i tenderá hacia un valor fijo que satisface la ecuación **x = g(x)**

Ejemplo. *Calcule una raíz real de* $f(x) = e^x - \pi x = 0$ *con el método del punto fijo.*

Un gráfico de **f** nos muestra que la ecuación tiene dos raíces reales en el intervalo **[0, 2]**



Re-escribir la ecuación en la forma **x = g(x)**.

Tomando directamente de la ecuación (puede haber varias opciones)

a) $x = g(x) = e^x/\pi$

b) $x = g(x) = \ln(\pi x)$

c) $x = g(x) = e^x - \pi x + x,$ etc

Usaremos la primera

Escribir la fórmula iterativa:

$$x_{i+1} = g(x_i) = e^{x_i}/\pi, \quad i = 0, 1, 2, 3, \ldots$$

Elegir un valor inicial $x_0 = 0.6$ (cercano a la primera raíz, tomado del gráfico)

Calcular los siguientes valores

$$x_1 = g(x_0) = e^{x_0}/\pi = e^{0.6}/\pi = 0.5800$$
$$x_2 = g(x_1) = e^{x_1}/\pi = e^{0.5800}/\pi = 0.5685$$
$$x_3 = g(x_2) = e^{0.5685}/\pi = 0.5620$$
$$x_4 = g(x_3) = e^{0.5620}/\pi = 0.5584$$
$$x_5 = g(x_4) = e^{0.5584}/\pi = 0.5564$$
$$x_6 = g(x_5) = e^{0.5564}/\pi = 0.5552$$
$$\ldots$$

La diferencia entre cada par de valores consecutivos se reduce en cada iteración. En los últimos la diferencia está en el orden de los milésimos, por lo tanto podemos considerar que el método converge y el error está en el orden de los milésimos.

Para calcular la segunda raíz, usamos la misma fórmula iterativa:

$$x_{i+1} = g(x_i) = e^{x_i}/\pi, \quad i = 0, 1, 2, 3, \ldots$$

El valor inicial elegido es $x_0 = 1.7$ (cercano a la segunda raíz, tomado del gráfico)

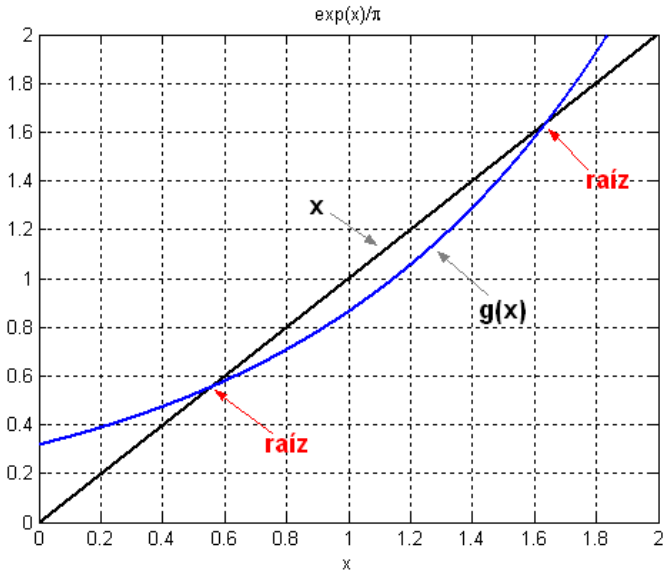
Calcular los siguientes valores

$$x_1 = g(x_0) = e^{x_0}/\pi = e^{1.7}/\pi = 1.7424$$
$$x_2 = g(x_1) = e^{x_1}/\pi = e^{1.7424}/\pi = 1.8179$$
$$x_3 = g(x_2) = e^{1.8179}/\pi = 1.9604$$
$$x_4 = g(x_3) = e^{1.9604}/\pi = 2.2608$$
$$x_5 = g(x_4) = e^{2.2608}/\pi = 3.0528$$
$$x_6 = g(x_5) = e^{3.0528}/\pi = 6.7399$$
$$x_6 = g(x_5) = e^{6.7399}/\pi = 269.1367$$

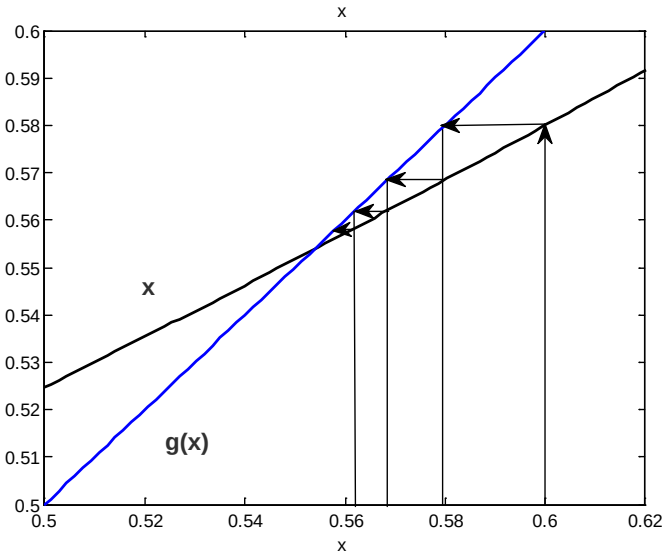
. . .

La diferencia entre cada par de valores consecutivos aumenta en cada iteración. Se concluye que el método no converge.

En el ejemplo anterior, las raíces reales de la ecuación **f(x)=0** son las intersecciones de **f** con el eje horizontal. En el problema equivalente **x=g(x)**, las raíces reales son las intersecciones entre **g** y la recta **x**:



En el cálculo de la primera raíz, la pendiente de **g** es menor que la pendiente de **x** y se observa que la secuencia de valores generados tiende a la raíz. La interpretación gráfica del proceso de cálculo se describe en la siguiente figura.



Para la segunda raíz, la pendiente de **g** es mayor que la pendiente de **x** y se puede constatar que la secuencia de valores generados se aleja de la raíz.

Se puede observar que la convergencia parece relacionada con la pendiente de **g(x).** Esto se prueba mediante el siguiente desarrollo

3.2.3 Convergencia del método del punto fijo

Para el método del punto fijo

f(x) = 0, x = g(x) (ecuaciones equivalentes)

Si **r** es una raíz se debe cumplir

r = g(r) ⇔ f(r) = 0 (se satisfacen con la misma raíz)

Fórmula iterativa

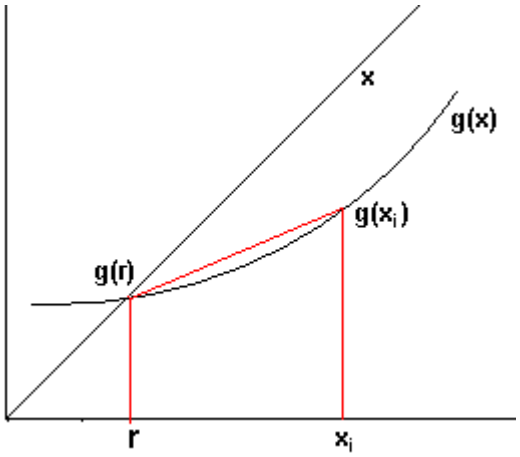
$x_{i+1} = g(x_i)$, **i = 0, 1, 2, 3, . . .**

Definiciones:

$E_i = r - x_i$: Error en la iteración **i**

$E_{i+1} = r - x_{i+1}$: Error en la iteración **i + 1**

Suponer que **g** es continua en el intervalo que incluye a **r** y a cada punto calculado x_i.



Por el Teorema del Valor Medio:

$$g'(z) = \frac{g(x_i) - g(r)}{x_i - r}, \text{ para algún } z \in (r, x_i)$$

Sustituyendo las definiciones anteriores:

$$g'(z) = \frac{x_{i+1} - r}{x_i - r} = \frac{E_{i+1}}{E_i} \quad \Rightarrow \quad E_{i+1} = g'(z)E_i\,, \quad i = 0, 1, 2, 3, \ldots$$

Este resultado indica que si en cada iteración, la magnitud de **g'( )** se mantiene menor que uno, entonces $E_{i+1} \underset{i\to\infty}{\to} 0$ y por lo tanto, $r - x_{i+1} \underset{i\to\infty}{\to} 0 \quad \Rightarrow \quad x_{i+1} \underset{i\to\infty}{\to} r$ (el método converge).
Se puede extender a los casos en los cuales **g** tiene pendiente negativa y deducir en general la condición de convergencia del método del punto fijo: **|g'(x)| < 1**.

Si **g(x)** es simple, se puede construir el intervalo en el cual la fórmula converge:

Ejemplo. *Encuentre el intervalo de convergencia para el ejemplo anterior.*

$f(x) = e^x - \pi x = 0$

$x = g(x) = e^x/\pi$

Condición de convergencia: $|g'(x)| < 1$

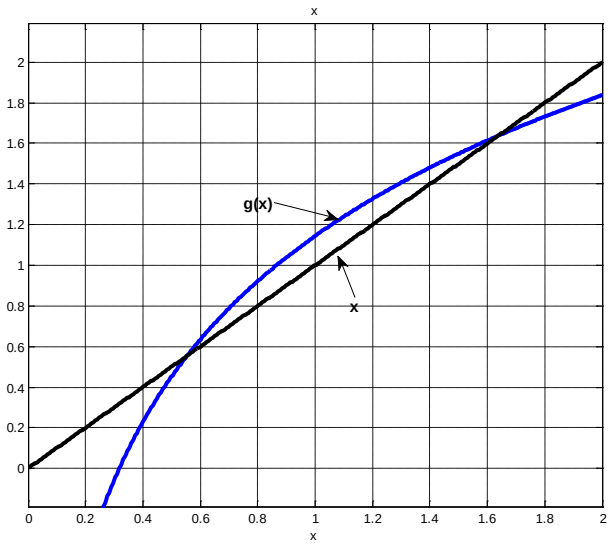
$g'(x) = e^x/\pi \quad \Rightarrow \quad |e^x/\pi| < 1 \quad \Rightarrow \quad x < \ln(\pi)$

Intervalo de convergencia: $(-\infty,\ \ln(\pi))$

Con lo que se concluye que la segunda raíz no se puede calcular con esta fórmula.

Ejemplo. *Calcule una raíz real de la misma ecuación* $f(x) = e^x - \pi x = 0$ *con el método del punto fijo con otra forma de la ecuación de recurrencia* $x=g(x)$.

Para este ejemplo se decide usar la segunda forma: $x = g(x)=\ln(\pi x)$



Según la condición de convergencia establecida, el gráfico muestra que será imposible calcular la primera raíz. La segunda raíz si podrá ser calculada. Se puede verificar numéricamente.

Ejemplo. *Calcular con MATLAB la segunda raíz de la ecuación:* ***f(x)=exp(x)-πx*** *mediante la fórmula de recurrencia:* ***x=g(x)=ln(πx)***

```
>> g=inline('log(pi*x)');
>> ezplot('x',[0,2]),grid on,hold on        Graficar x vs. g(x)
>> ezplot(g,[0,2])                          Produce el gráfico de la página anterior
>> x=1.6;                                   Valor inicial para la segunda raíz
>> x=g(x)
x =
   1.6147
>> x=g(x)                                   Reusar el comando
x =
   1.6239
>> x=g(x)
x =
   1.6296
>> x=g(x)
x =
   1.6330

……

>> x=g(x)                                   Valor final luego de 15 iteraciones
x =
   1.6385
```

3.2.4 Eficiencia del método del punto fijo

El método del punto fijo tiene convergencia lineal o de primer orden debido a que E_i y E_{i+1} están relacionados directamente mediante el factor de convergencia: $E_{i+1} = g'(z)E_i$, por lo tanto este método tiene convergencia lineal $E_{i+1} = O(E_i)$ y $g'(z)$ es el factor de convergencia. Si su magnitud permanece mayor a 1, el método no converge.

3.3 Método de Newton

Sean **f: R→R**, y la ecuación **f(x)=0.** Sea **r** tal que **f(r)=0,** entonces **r** es una raíz real de la ecuación.

El método de Newton, o Newton-Raphson, es una fórmula eficiente para encontrar **r.** Es un caso especial del método del punto fijo en el que la ecuación **f(x) = 0** se re-escribe en la forma **x = g(x)** eligiendo **g** de tal manera que la convergencia sea de segundo orden.

3.3.1 La fórmula de Newton

Suponer que **g** es una función diferenciable en una región que incluye a la raíz **r** y al valor x_i (calculado en la iteración **i**). Desarrollando con la serie de Taylor:

$$g(x_i) = g(r) + (x_i - r)\, g'(r) + (x_i - r)^2 g''(r)/2! + \ldots$$

Con las definiciones del método del punto fijo:

$$r = g(r)$$
$$x_{i+1} = g(x_i),\ i = 0, 1, 2, 3, \ldots$$

Se obtiene:

$$x_{i+1} = r + (x_i - r)\, g'(r) + (x_i - r)^2 g''(r)/2! + \ldots$$

Si se define el error de truncamiento de la siguiente forma:

$E_i = x_i - r$: Error en la iteración **i**

$E_{i+1} = x_{i+1} - r$: Error en la iteración **i +1**

Finalmente se obtiene:

$$E_{i+1} = E_i\, g'(r) + E_i^2\, g''(r)/2! + \ldots$$

Si se hace que $g'(r) = 0$, y si $g''(r) \neq 0$, entonces se tendrá:

$$E_{i+1} = O(E_i^2),$$

Con lo que el método tendrá convergencia cuadrática.

El procedimiento para hacer que **g'(r) = 0**, consiste en elegir una forma apropiada para **g(x):**

$g(x) = x - f(x)\, h(x)$, en donde **h** es alguna función que debe especificarse

Es necesario verificar que la ecuación **x = g(x)** se satisface con la raíz **r** de la ecuación **f(x) = 0**

$$g(r) = r - f(r)\, h(r) = r \Rightarrow g(r) = r$$

Se deriva **g(x)** y se evalúa en **r**

$$g'(x) = 1 - f'(x)\, h(x) - f(x)\, h'(x)$$
$$g'(r) = 1 - f'(r)\, h(r) - f(r)\, h'(r) = 1 - f'(r)\, h(r)$$

Para que la convergencia sea cuadrática se necesita que **g'(r) = 0**

$$g'(r) = 0 \Rightarrow 0 = 1 - f'(r)\, h(r) \Rightarrow h(r) = 1/f'(r) \Rightarrow h(x) = 1/f'(x),\ f'(x) \neq 0$$

Con lo que se puede especificar **h(x)** para que la convergencia sea cuadrática.

Al sustituir en la fórmula propuesta se obtiene **x = g(x) = x - f(x)/f'(x)**, y se puede escribir la fórmula iterativa de Newton:

Definición: Fórmula iterativa de Newton

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad f'(x_i) \neq 0,\ i = 0, 1, 2, \ldots$$

3.3.2 Algoritmo del método de Newton

Para calcular una raíz **r** real de la ecuación **f(x) = 0** con precisión **E** se usa la fórmula iterativa $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$, $f'(x_i) \neq 0$ y se genera una sucesión de valores x_i esperando que tienda a un valor que satisfaga la ecuación.

1) Elegir el valor inicial x_0
2) Generar la sucesión de valores con la fórmula iterativa:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},\ f'(x_i) \neq 0\ ,\ i = 0, 1, 2, 3, \ldots$$

3) Si el método converge, la sucesión x_i tenderá hacia un valor fijo que satisface la ecuación **f(x) = 0**

Ejemplo. *Calcule una raíz real de* $f(x) = e^x - \pi x = 0$ *con la fórmula de Newton*

Suponer el valor inicial: $x_0 = 0.5$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = x_0 - \frac{e^{x_0} - \pi x_0}{e^{x_0} - \pi} = 0.5 - \frac{e^{0.5} - 0.5\pi}{e^{0.5} - \pi} = 0.5522$$

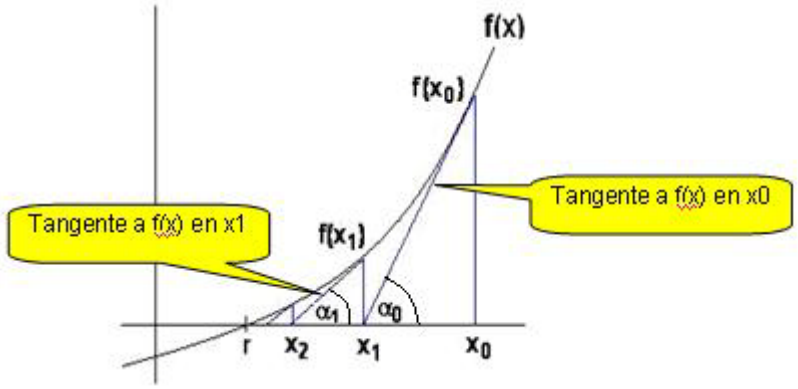
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = x_1 - \frac{e^{x_1} - \pi x_1}{e^{x_1} - \pi} = 0.5522 - \frac{e^{0.5522} - 0.5522\pi}{e^{0.5522} - \pi} = 0.5538$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = x_2 - \frac{e^{x_2} - \pi x_2}{e^{x_2} - \pi} = 0.5538 - \frac{e^{0.5538} - 0.5538\pi}{e^{0.5538} - \pi} = 0.5538$$

En los resultados se observa la rápida convergencia. En la tercera iteración el resultado tiene cuatro decimales que no cambian.

3.3.3 Interpretación gráfica del método de Newton

Suponer que **f** es como se muestra en el siguiente gráfico y x_0 es el valor inicial:



Seguimos el siguiente procedimiento:

Calcular $f(x_0)$
Trazar una tangente a **f** en el punto $(x_0, f(x_0))$ hasta intersecar al eje horizontal
El punto obtenido es x_1
Entonces $\tan(\alpha_0) = f'(x_0) = f(x_0)/(x_0 - x_1)$ de donde se obtiene $x_1 = x_0 - f(x_0)/f'(x_0)$

Calcular $f(x_1)$
Trazar una tangente a **f** en el punto $(x_1, f(x_1))$ hasta intersecar al eje horizontal:
El punto obtenido es x_2
Entonces $\tan(\alpha_1) = f'(x_1) = f(x_1)/(x_1 - x_2)$ de donde se obtiene $x_2 = x_1 - f(x_1)/f'(x_1)$

Con estos dos resultados se puede generalizar:
$\tan(\alpha_i) = f'(x_i) = f(x_i)/(x_i - x_{i+1})$ de donde se obtiene $x_{i+1} = x_i - f(x_i)/f'(x_i)$. Es la fórmula de Newton.

Esta interpretación gráfica permite observar que la secuencia de valores calculados con la fórmula de Newton sigue la trayectoria de las tangentes a **f(x)**. Si hay convergencia, esta secuencia tiende a la raíz **r**. En la siguiente sección se analiza la propiedad de convergencia de éste método.

3.3.4 Convergencia del método de Newton

Se estableció en el método del punto fijo que la ecuación recurrente **x=g(x)** tiene la siguiente propiedad de convergencia: $|E_{i+1}| = |g'(z)||E_i|$, **i = 0, 1, 2, 3, . . .** La convergencia se produce si se cumple que **|g'(x)|<1**

Para el método de Newton se obtuvo la siguiente ecuación recurrente:

$$x = g(x) = x - f(x)/f'(x),$$

Entonces, $$g'(x) = \frac{f(x)f''(x)}{[f'(x)]^2} \quad , \quad f(x_i) \neq 0$$

Si se supone que en alguna región cercana a **r** en la cual se incluyen los valores calculados **x,** se tiene que $f'(x) \neq 0$, y si **r** es una raíz de **f(x)=0**, entonces **f(r) = 0**, y por lo tanto:

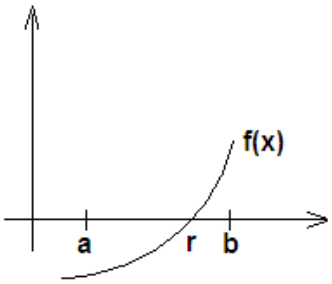
$$g'(r) = \frac{f(r)f''(r)}{[f'(r)]^2} = 0$$

Este resultado ya se estableció, pero demuestra que existe algún intervalo cercano **r** en el que **|g'(x)|<1** siempre que **g** sea continua en ese intervalo La fórmula de Newton converge si los valores calculados se mantienen dentro de este intervalo.

Para obtener el intervalo de convergencia, no es práctico usar la definición anterior pues involucra resolver una desigualdad complicada. Existen otros procedimientos para demostrar la convergencia de esta fórmula como se muestra a continuación.

3.3.5 Una condición de convergencia local para el método de Newton

Dada la ecuación **f(x)=0**. Suponer que **f(x), f'(x), f''(x)** son continuas y limitadas en un intervalo **[a, b]** que contiene a la raíz **r** como se muestra en el siguiente gráfico:



Para este caso, **f(x)** tiene las siguientes propiedades

a) f(x) > 0, $x \in (r, b]$
b) f'(x) > 0, $x \in (r, b]$
c) f''(x) > 0, $x \in (r, b]$

Partiendo de estas premisas se demuestra que la fórmula de Newton converge para cualquier valor inicial x_0 elegido en el intervalo **(r, b]**.

Demostración

1) Las premisas **a)** y **b)** implican que $x_{i+1} < x_i$:
En la fórmula de Newton:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad \Rightarrow \quad x_{i+1} < x_i \qquad (1)$$

El enunciado es válido al suprimir un término positivo pues **f(x), f'(x)** son positivos

2) La premisa **c)** implica que $\mathbf{r < x_{i+1}}$:

Desarrollamos **f(x)** con tres términos de la serie de Taylor alrededor de $\mathbf{x_i}$

$$\mathbf{f(r) = f(x_i) + (r - x_i)f'(x_i) + (r - x_i)^2 f''(z)/2!}$$

$$\mathbf{f(r) > f(x_i) + (r - x_i)f'(x_i) \Rightarrow \; 0 > f(x_i) + (r - x_i)f'(x_i) \; \Rightarrow \; r < x_i - f(x_i)/f'(x_i) \; \Rightarrow \; r < x_{i+1}} \qquad \mathbf{(2)}$$

El enunciado es válido pues el último término que se suprime es positivo, si **f''(x)** es positivo

Combinando los resultados **(1)** y **(2)** se tiene

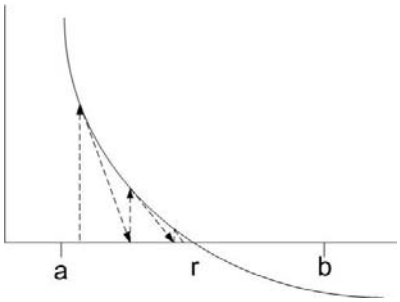
$$\mathbf{r < x_{i+1} < x_i \text{ , } i = 0, 1, 2,...}$$

Este resultado define una sucesión numérica decreciente que tiende a **r** y prueba la convergencia de la fórmula iterativa de Newton: $\mathbf{x_i \underset{i\to\infty}{\to} r}$

Si se dispone del gráfico de **f** es fácil reconocer visualmente si se cumplen las condiciones **a)**, **b)** y **c)** como el caso anterior y se puede definir un intervalo para la convergencia del método.

Si **f** tiene otra forma, igualmente se pueden enunciar y demostrar las condiciones para que se produzca la convergencia en cada caso.

Ejemplo. *Determine la convergencia del método de Newton si* ***f*** *tuviese la siguiente forma*



Su geometría se puede describir con:

a) f(x)>0, x∈[a, r)
b) f'(x)<0, x∈[a, r)
c) f''(x)>0, x∈[a, r)

Con un desarrollo similar al anterior, se puede probar que el método converge para cualquier valor inicial $\mathbf{x_0}$ *elegido en el intervalo* ***[a, r),*** *(a la izquierda de la raíz* ***r).***

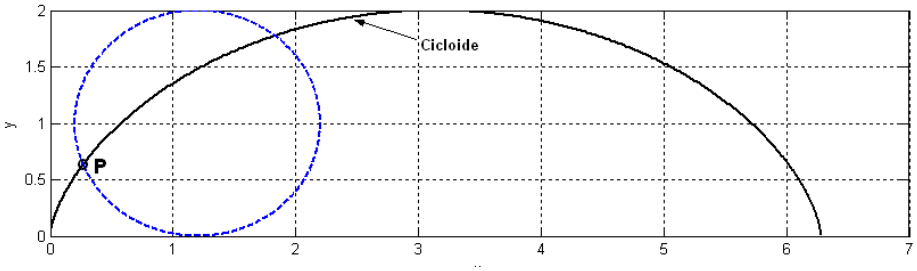
El uso de esta propiedad es simple si se dispone de un gráfico de la ecuación. Se elige como intervalo de convergencia aquella región en la que se observa que la trayectoria de las tangentes produce la convergencia a la raíz. Si se elije un valor inicial arbitrario no se puede asegurar que el método converge.

Ejemplo.- *Si un círculo de radio* ***a*** *rueda en el plano a lo largo del eje horizontal, un punto* ***P*** *de la circunferencia trazará una curva denominada cicloide. Esta curva puede expresarse mediante las siguientes ecuaciones paramétricas*

$$x(t) = a(t - \operatorname{sen} t), \qquad y(t) = a(1 - \cos t)$$

Suponga que el radio es ***1*** *metro, si* ***(x, y)*** *se miden en metros y* ***t*** *representa tiempo en segundos, determine el primer instante en el que la magnitud de la velocidad es 0.5 m/s. Use el método de Newton,* ***E=0.0001***

Gráfico de la cicloide



Su trayectoria:

$$u(t) = (x(t), y(t)) = (t - \operatorname{sen} t,\ 1 - \cos t)$$

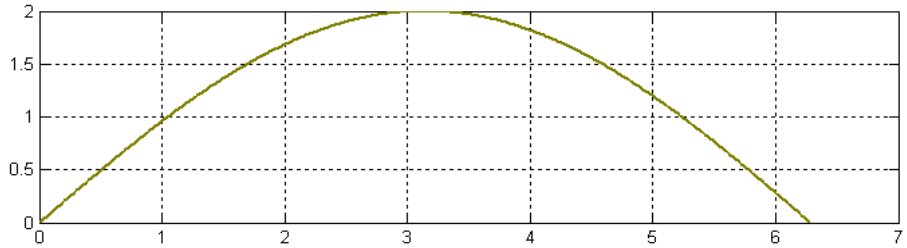
Su velocidad:

$$u'(t) = (1 - \cos t, \operatorname{sen} t)$$

Magnitud de la velocidad:

$$\|u'(t)\| = \sqrt{(1-\cos t)^2 + (\operatorname{sen} t)^2}$$

Gráfico de la magnitud de la velocidad



Dato especificado:

$$\sqrt{(1-\cos t)^2 + (\operatorname{sen} t)^2} = 0.5 \quad \Rightarrow \quad f(t) = (1 - \cos t)^2 + (\operatorname{sen} t)^2 - 0.25 = 0$$

Método de Newton

$$t_{i+1} = t_i - \frac{f(t_i)}{f'(t_i)} = t_i - \frac{(1-\cos t_i)^2 + (\operatorname{sen} t_i)^2 - 0.25}{2(1-\cos t_i)(\operatorname{sen} t_i) + 2(\operatorname{sen} t_i)(\cos t_i)}$$ *(fórmula iterativa)*

$t_0 = 0.5$ *(del gráfico)*

$t_1 = \ldots = 0.505386$ *(iteraciones)*
$t_2 = \ldots = 0.505360$
$t_3 = \ldots = 0.505360$

3.3.6 Práctica computacional

En esta sección se describe el uso de MATLAB para usar el método de Newton. Se lo hará directamente en la ventana de comandos.

Para calcular una raíz debe elegirse un valor inicial cercano a la respuesta esperada de acuerdo a la propiedad de convergencia estudiada para este método.

Para realizar los cálculos se usa la fórmula de Newton en notación algorítmica:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i=0, 1, 2, 3, \ldots$$

x_0 es el valor inicial

$x_1, x_2, x_3, \ldots$ son los valores calculados

En los lenguajes computacionales como MATLAB no se requieren índices para indicar que el valor de una variable a la izquierda es el resultado de la evaluación de una expresión a la derecha con un valor anterior de la misma variable.

La ecuación se puede definir como una cadena de caracteres entre **comillas** simples. La derivada se obtiene con la función **diff** de MATLAB y con la función **eval** se evalúan las expresiones matemáticas. Opcionalmente se puede usar el tipo **syms** para definir variables simbólicas o la función **sym** para operar algebraicamente con expresiones matemáticas definidas en formato texto.

Forma computacional de la fórmula de Newton en MATLAB:

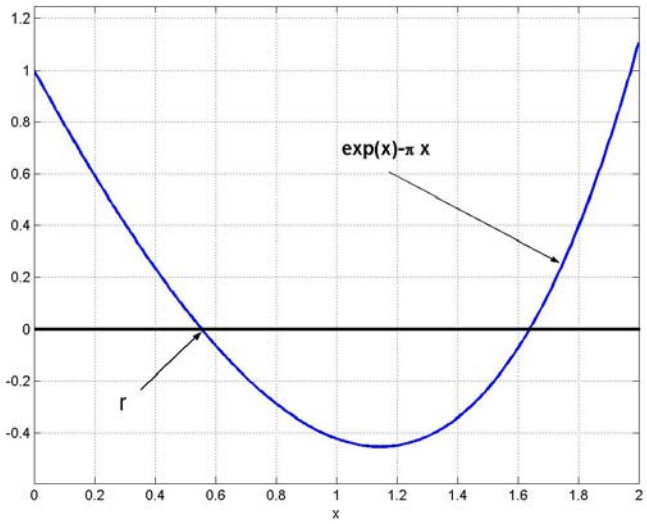
```
x=x-eval(f)/eval(diff(f))
```

Presionando repetidamente la tecla del cursor ↑ se obtienen resultados sucesivos. La convergencia o divergencia se puede observar directamente en los resultados.

Ejemplo. *Calcule con MATLAB las raíces reales de* $f(x) = e^x - \pi x = 0$ *con la fórmula de Newton.*

Es conveniente graficar la ecuación mediante los siguientes comandos de MATLAB. También se puede editar el dibujo directamente en la ventana de graficación:

```
>> syms x
>> f=exp(x)-pi*x;
>> ezplot(f,[0,2]),grid on
```



A continuación se utliza la fórmula de Newton en MATLAB eligiendo del gráfico un valor inicial. Reutilizando este comando se obtiene una secuencia de aproximaciones:

```
>> format long
>> x=0.5;
>> x=x-eval(f)/eval(diff(f))
x =
   0.552198029112459
>> x=x-eval(f)/eval(diff(f))
x =
   0.553825394773978
>> x=x-eval(f)/eval(diff(f))
x =
   0.553827036642841
>> x=x-eval(f)/eval(diff(f))
x =
   0.553827036644514
>> x=x-eval(f)/eval(diff(f))
x =
   0.553827036644514
```

El último resultado tiene quince decimales fijos.

Se puede observar la rapidez con la que el método se acerca a la respuesta duplicando aproximadamente, la precisión en cada iteración. Esto concuerda con la propiedad de convergencia cuadrática.

Finalmente, es necesario verificar que este resultado satisface a la ecuación:

```
>> eval(f)
ans =
    0
```

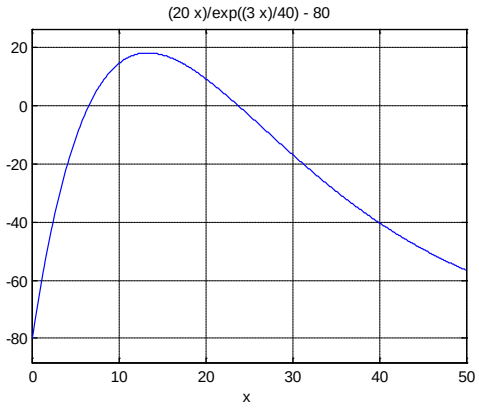
Ejemplo. *Se propone el siguiente modelo para describir la demanda de un producto, en donde* ***x*** *es tiempo en meses:*

$$\mathbf{d(x) = 20x\ e^{-0.075x}}$$

Encuentre el valor de ***x*** *para el cual la demanda alcanza el valor de 80 unidades. Use el método de Newton para los cálculos. Elija el valor inicial del gráfico y muestre los valores intermedios.*

La ecuación a resolver es: $\mathbf{f(x) = 20x\ e^{-0.075x} - 80 = 0}$

```
>> syms x
>> f=20*x*exp(-0.075*x)-80;
>> ezplot(f,[0,50]),grid on
>> x=5;
>> x=x-eval(f)/eval(diff(f))
x =
   6.311945053556490
>> x=x-eval(f)/eval(diff(f))
x =
   6.520455024943885
>> x=x-eval(f)/eval(diff(f))
x =
   6.525360358429755
>> x=x-eval(f)/eval(diff(f))
x =
   6.525363029068742
>> x=x-eval(f)/eval(diff(f))
x =
   6.525363029069534
>> x=x-eval(f)/eval(diff(f))
x =
   6.525363029069534
```



Ejemplo. *Una partícula se mueve en el plano* ***X-Y*** *de acuerdo con las ecuaciones paramétricas siguientes, donde* ***t*** *es tiempo, entre* ***0*** *y* ***1****:*

x(t)=t*exp(t)

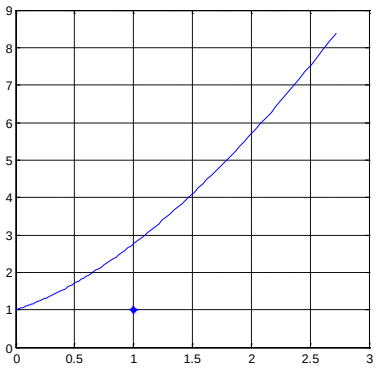
y(t)=1+t*exp(2t)

Con la fórmula de Newton calcule el tiempo en el que la partícula está más cerca del punto ***(1,1)***

Distancia de un punto ***(x, y)*** *al punto* ***(1, 1)*** *:* $\mathbf{d = \sqrt{(x(t)-1)^2 + (y(t)-1)^2}}$

Para encontrar la menor distancia, debe resolverse la ecuación: ***f(t) = d'(t) = 0***

```
>> t=[0:0.01:1];          Puntos para evaluar las ecuaciones paramétricas
>> x=t.*exp(t);
>> y=1+t.*exp(2*t);
>> plot(x,y)              Gráfico del recorrido
```



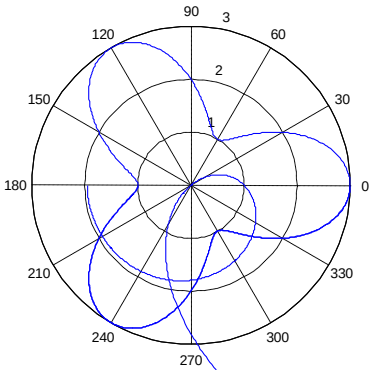

```
>> syms t
>> x=t*exp(t);
>> y=1+t*exp(2*t);
>> d=sqrt((x-1)^2+(y-1)^2)                    (para operar algebraicamente)
    d =
       (t^2*exp(4*t) + (t*exp(t) - 1)^2)^(1/2)
>> f=diff(d);
>> t=0.5;
>> t=t-eval(f)/eval(diff(f))
t =
   0.278246639067713
>> t=t-eval(f)/eval(diff(f))
t =
   0.258310527656699
>> t=t-eval(f)/eval(diff(f))
t =
   0.256777599742604
>> t=t-eval(f)/eval(diff(f))
t =
   0.256768238259669
>> t=t-eval(f)/eval(diff(f))
t =
   0.256768237910400
>> t=t-eval(f)/eval(diff(f))
t =
   0.256768237910400                          (tiempo para la menor distancia)
>> eval(d)
ans =
    0.794004939848305                         (es la menor distancia)
```

Ejemplo. *Encuentre una intersección de las siguientes ecuaciones en coordenadas polares*

$r=2+\cos(3t)$
$r=2-e^{t}$

Ecuación a resolver: $f(t) = 2+\cos(3t) - (2-e^{t})$

```
>> t=[-pi:0.01:2*pi];                Puntos para evaluar las ecuaciones
>> r=2+cos(3*t);
>> polar(t,r),hold on
>> r=2- exp(t);
>> polar(t,r)                        Gráfico en coordenadas polares
```




```
>> syms t
>> f=2+cos(3*t)-(2-exp(t));
>> t=-1;                             (valor inicial)
>> t=t-eval(f)/eval(diff(f))
t =
  -0.213748703557153
>> t=t-eval(f)/eval(diff(f))
t =
  -0.832049609116596
>> t=t-eval(f)/eval(diff(f))
t =
  -0.669680711112045
>> t=t-eval(f)/eval(diff(f))
t =
  -0.696790503081824
>> t=t-eval(f)/eval(diff(f))
t =
  -0.697328890705191
>> t=t-eval(f)/eval(diff(f))
t =
  -0.697329123134159
>> t=t-eval(f)/eval(diff(f))
t =
  -0.697329123134202
>> t=t-eval(f)/eval(diff(f))
t =
  -0.697329123134202                (ángulo: -39.95…. grados)
>> r=2+cos(3*t)
r =
   1.502086605214547                (radio)
```

3.3.7 Instrumentación computacional del método de Newton

Para evitar iterar desde la ventana de comandos, se puede instrumentar una función que reciba la ecuación a resolver **f**, la variable independiente **v** definida como símbolo matemático y el valor inicial **u**. Adicionalmente se puede enviar un parámetro **e** para controlar la precisión requerida y otro parámetro **m** para el máximo de iteraciones.

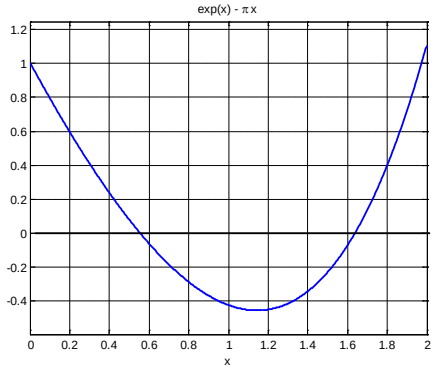
La función entrega la solución calculada **u** y el número de iteraciones realizadas **k**. Si el método no converge en el máximo de iteraciones previsto, **u** contendrá un valor nulo y el número de iteraciones **k** será igual al máximo **m**.

```
function [u,k]=newton(f,v,u,e,m)
t=u;
for k=1:m
   u=u-subs(f,v,u)/subs(diff(f,v),v,u);          %fórmula de Newton
   if abs(t-u)<e
      return
   end
   t=u;
end
u=[ ];
k=m;
```

Ejemplo. *Calcule las dos raíces reales de* $f(x) = e^x - \pi x = 0$ *con la función newton. Los valores iniciales son tomados del gráfico.*

```
>> syms x
>> f=exp(x)-pi*x;
>> ezplot(f,[0,2]),grid on
```

forma alternativa de definir expresiones matemáticas con el tipo simbólico: ***syms*** *(también se pueden escribir con comillas)*




```
>> [u,k]=newton(f, x, 0.5, 0.000001, 10)
u =
   0.553827036644514                          Primera raíz
k =
    4
>> [u,k]=newton(f, x, 1.6, 0.000001, 10)
u =
   1.638528419970363                          Segunda raíz
k =
    4
```

Si no es de interés el número de iteraciones, la función puede llamarse con un solo parámetro

```
>> u=newton(f, x, 0.5, 0.000001, 10)
u =
  0.553827036644514
```

3.3.8 Uso de funciones especiales de MATLAB

MATLAB dispone de funciones especiales para calcular raíces de ecuaciones no lineales.

La función fzero

Es equivalente a usar los métodos de la bisección y de Newton

Ejemplo. *Resuelva la ecuación* $f(x) = e^x - \pi x = 0$ *usando* ***fzero***

```
>> f='exp(x)-pi*x';
>> r=fzero(f, [0.4, 0.8])          (Para usar como el método de la bisección
   r =                              se especifica el intervalo que contiene a la raíz)
      0.55382703664451


>> f='exp(x)-pi*x';
>> r=fzero(f, 0.4)                 (Para usar como el método de Newton
   r =                               se especifica un valor cercano a la raíz)
      0.553827036644514
>> r=fzero(f, 1.6)
   r =
      1.638528419970363
```

La función solve

Se usa resolver ecuaciones en forma simbólica exacta. En algunos casos la solución queda expresada mediante símbolos predefinidos en MATLAB. Con la función **eval** se convierten las soluciones al formato numérico decimal. Esta función no siempre entrega todas las soluciones de una ecuación.

Ejemplo. *Resuelva la ecuación* $f(x) = e^x - \pi x = 0$ *usando* ***solve***

```
>> f='exp(x)-pi*x';
>> r=eval(solve(f))
   r =
      0.553827036644514          (Raíz real en forma numérica decimal)
```

La función roots

Sirve para calcular todas las **raíces de ecuaciones polinómicas.** Los coeficientes del polinomio deben almacenarse en un vector.

Ejemplo. *Encuentre todas las raíces del polinomio* $2x^3 - 5x^2 - 4x - 7 = 0$ *usando* ***roots***

```
>> a=[2, -5, -4, -7];              (Coeficientes del polinomio)
>> r=roots(a)
r =
   3.39334562071304               (Una raíz real y dos raíces complejas)
  -0.44667281035652 + 0.91209311493838i
  -0.44667281035652 - 0.91209311493838i
```

3.4 Ejercicios y problemas de ecuaciones no-lineales

1. El producto de dos números reales positivos es **5**. Mientras que al sumar el cubo del primero más el cuadrado del segundo se obtiene **40**. Encuentre estos dos números.

2. El producto de las edades en años de dos personas es 677.35 y si se suman los cubos de ambas edades se obtiene 36594.38 Encuentre cuales son estas edades.

3. Una empresa produce semanalmente una cantidad de artículos. El costo de producción semanal tiene un costo fijo de **250** y un coso de **2.50** por cada artículo producido. El ingreso semanal por venta tiene un valor de **3.50** por cada artículo vendido, más un costo de oportunidad que se ha estimado directamente proporcional a la cantidad de artículos producidos, multiplicado por el logaritmo natural. Encuentre el punto de equilibrio para este modelo económico.

4. En una empresa de fertilizantes en cada mes, el ingreso por ventas en miles de dólares se describe con el modelo **v(x) = 0.4x (30 - x)** mientras que el costo de producción en miles de dólares es **c(x) = 5 + 10 ln(x)**, siendo **x** la cantidad producida en toneladas, **1<x<30**. ¿Que cantidad mensual debe producir para obtener el máximo beneficio económico?

5. El costo semanal fijo por uso de un local para venta de cierto tipo de artículo es $50. Producir cada Kg. del artículo cuesta $2.5. El ingreso por venta es **3x + 2 ln(x)** en donde **x** representa la cantidad de kg vendidos en cada semana. Determine la cantidad de Kg. que se debe vender semanalmente a partir de la cual la empresa obtiene ganancias.

6. En una planta de abastecimiento de combustible se tiene un tanque de forma esférica. El volumen del líquido almacenado en el tanque se puede calcular mediante la siguiente fórmula: $\mathbf{v(h) = \pi h^2 (R - h/3)}$,donde h representa la altura del líquido dentro del tanque medida desde la parte inferior del tanque y R su radio ($0 \leq h \leq 2R$). Suponga que el tanque tiene un radio de 2 m. Calcule la altura que debe tener el líquido para que el tanque contenga 27 m^3. Calcule el resultado con una tolerancia $E=10^{-4}$.

7. Encuentre el punto de la curva dada por $\mathbf{y = 2x^5 - 3xe^{-x} - 10}$ ubicado en el tercer cuadrante, donde su recta tangente sea paralela al eje **X**.

8. Determine la raíz real de la ecuación: **sin(x) = ln(x)**

9. Determine la segunda raíz real positiva, con 4 decimales exactos de la ecuación:
5 cos(πx) = tan(πx)

10. Calcule una raíz real positiva de la ecuación **sin(πx) + 1 = x.**

.

11. Para que **f** sea una función de probabilidad se tiene que cumplir que su integral en el dominio de **f** debe tener un valor igual a **1**. Encuentre el valor de **b** para que la función $\mathbf{f(x)=2x^2 + x}$ sea una función de probabilidad en el dominio **[0, b]**.

12. Calcule con cuatro decimales exactos la intersección de la ecuación $\mathbf{(y - 1)^2 + x^3 = 6}$, con la recta que incluye a los puntos **(-1, -1), (1, 1)** en el plano.

13. Encuentre una fórmula iterativa de convergencia cuadrática y defina un intervalo de convergencia apropiado para calcular la raíz real n-ésima de un número real. El algoritmo solamente debe incluir operaciones aritméticas elementales.

14. Un ingeniero desea tener una cantidad de dólares acumulada en su cuenta de ahorros para su retiro luego de una cantidad de años de trabajo. Para este objetivo planea depositar un valor mensualmente. Suponga que el banco acumula el capital mensualmente mediante la siguiente fórmula:

$$\mathbf{A = P\left[\frac{(1+x)^n - 1}{x}\right]}, \text{ en donde}$$

A: Valor acumulado
P: Valor de cada depósito mensual
n: Cantidad de depósitos mensuales
x: Tasa de interés mensual

Determine la tasa de interés anual que debe pagarle el banco si desea reunir 200000 en 25 años depositando cuotas mensuales de 350

15. Una empresa compra una máquina en 20000 dólares pagando 5000 dólares cada año durante los próximos 5 años. La siguiente fórmula relaciona el costo de la máquina **P**, el pago anual **A**, el número de años **n** y el interés anual **x**: $\mathbf{A = P\frac{x(1+x)^n}{(1+x)^n - 1}}$

Determine la tasa de interés anual **x** que se debe pagar.

16. Una empresa vende un vehículo en **P**=$34000 con una entrada de **E**=$7000 y pagos mensuales de **M**=$800 durante cinco años. Determine el interés mensual **x** que la empresa está cobrando. Use la siguiente fórmula:

$\mathbf{P = E + \frac{M}{x}[1 - \frac{1}{(1+x)^n}]}$, en donde **n** es el número total de pagos mensuales

17. Un modelo de crecimiento poblacional está dado por: $\mathbf{f(t) = 5t + 2e^{0.1t}}$, en donde n es el número de habitantes, t es tiempo en años.

a) Calcule el número de habitantes que habrán en el año 25
b) Encuentre el tiempo para el cual la población es 200

18. Un modelo de crecimiento poblacional está dado por. $\mathbf{f(x) = kx + 2e^{0.1x}}$, siendo **k** una constante que debe determinarse y **x** tiempo en años. Se conoce que **f(10)=50**.

a) Determine la población en el año 25
b) Determine el año en el que la población alcanzará el valor 1000.

19. Un modelo de crecimiento poblacional está dado por $\mathbf{f(t) = k_1 t + k_2 e^{0.1t}}$

Siendo $\mathbf{k_1}$ y $\mathbf{k_2}$ constantes, y **t** tiempo en años.
Se conoce que f(10)=25, f(20)=650.

a) Determine la población en el año 25
b) Determine el año en el que la población alcanzará el valor 5000.

20. La concentración de bacterias contaminantes **C** en un lago decrece de acuerdo con la relación: $\mathbf{C = 70e^{-1.5t} + 25e^{-0.075t}}$.
Se necesita determinar el tiempo para que la concentración de bacterias se reduzca a 15 unidades o menos.
a) Determine un intervalo de existencia de la raíz de la ecuación. Use un gráfico
b) Aproxime la raíz indicando la cota del error.

21. En un modelo de probabilidad se usa la siguiente fórmula para calcular la probabilidad **f(k)** que en el intento número k se obtenga el primer resultado favorable:

$$f(k) = p(1-p)^{k-1}, 0 \le p \le 1, k = 0, 1, 2, 3, \ldots$$

a) Si en una prueba se obtuvo que **f(5) = 0.0733**, encuentre cuales son los dos posibles valores de **p** posibles en la fórmula.
b) Con el menor valor obtenido para **p** encuentre el menor valor de **k** para el que **f(k)<0.1**

22. Para simular la trayectoria de un cohete se usará el siguiente modelo:

$$y(t) = 6 + 2.13t^2 - 0.0013t^4$$

En donde **y** es la altura alcanzada, en metros y **t** es tiempo en segundos. El cohete está colocado verticalmente sobre la tierra.

a) Encuentre el tiempo de vuelo.
b) Encuentre la altura máxima del recorrido.

23. El movimiento de una partícula en el plano, se encuentra representado por las ecuaciones paramétricas:

$$x(t) = 3\,\mathrm{sen}^3(t) - 1; \quad y(t) = 4\,\mathrm{sen}(t)\cos(t); \quad t \ge 0$$

Donde **x, y** son las coordenadas de la posición expresadas en cm, **t** se expresa en seg.

a) Demuestre que existe un instante $t \in [0, \pi/2]$ tal que sus coordenadas **x** e **y** coinciden.
b) Aproxime con una precisión de 10^{-5} en qué instante de tiempo las dos coordenadas serán iguales en el intervalo dado en a).

24. Los polinomios de Chebyshev $T_n(x)$ son utilizados en algunos métodos numéricos. Estos polinomios están definidos recursivamente por la siguiente formulación:

$$T_o(x) = 1, \quad T_1(x) = x$$
$$T_n(x) = 2.x.T_{n-1}(x) - T_{n-2}(x), \; n = 0, 1, 2, 3, \ldots$$

Calcule todas las raíces reales positivas de $T_7(x)$.

25. La posición del ángulo central θ en el día t de la luna alrededor de un planeta si su período de revolución es P días y su excentricidad es e, se describe con la ecuación de Kepler:

$$2\pi t - P\theta + P e \,\mathrm{sen}\,\theta = 0$$

Encuentre la posición de la luna (ángulo central) en el día 30, sabiendo que el período de revolución es 100 días y la excentricidad 0.5.

26. Una partícula se mueve en el plano **XY** (la escala está en metros) con una trayectoria descrita por la función $u(t) = (x(t), y(t)) = \left(2t\,e^t + \sqrt[4]{t},\ 2t^3\right)$, $t \in [0,1]$, **t** medido en horas.

a) Grafique la trayectoria **u(t)**
b) Encuentre la posición de la partícula cuando **x=3.**
c) En que el instante la partícula se encuentra a una distancia de **4** metros del origen.

27. La siguiente ecuación relaciona el factor de fricción **f** y el número de Reynolds **Re** para flujo turbulento que circula en un tubo liso: $1/f = -0.4 + 1.74 \ln(Re\, f)$
Calcule el valor de **f** para **Re = 20000**

28. En una región se instalan **100** personas y su tasa de crecimiento es $\mathbf{e^{0.2x}}$, en donde **x** es tiempo en años. La cantidad inicial de recursos disponibles abastece a **120** personas. El incremento de recursos disponibles puede abastecer a una tasa de **10x** personas, en donde **x** es tiempo en años. Se desea conocer cuando los recursos no serán suficientes para abastecer a toda la población. Calcule la solución con cuatro dígitos de precisión y determine el año, mes y día en que se producirá este evento.

29. Suponga que el precio de un producto **f(x)** depende del tiempo **x** en el que se lo ofrece al mercado con la siguiente relación **f(x) = 25x exp(-0.1x), 0≤x≤12,** en donde **x** es tiempo en meses. Se desea determinar el día en el que el precio sube a **80**.
a) Evalúe **f** con **x** en meses hasta que localice una raíz real (cambio de signo) y trace la forma aproximada de **f(x)**
b) Calcule la respuesta (mes) con E=$\mathbf{10^{-4}}$. Exprese esta respuesta en días (1mes = 30 días)
c) Encuentre el día en el cual el precio será máximo. E=10^{-4}

30. Una partícula sigue una trayectoria elíptica centrada en el origen (eje mayor 4 y eje menor 3) comenzando en el punto más alto, y otra partícula sigue una trayectoria parabólica ascendente hacia la derecha comenzando en el origen (distancia focal 5). El recorrido se inicia en el mismo instante.

a) Encuentre el punto de intersección de las trayectorias.

b) Si la primera partícula tiene velocidad uniforme y pasa por el punto más alto cada minuto, determine el instante en el cual debe lanzarse la segunda partícula con aceleración 10 m/s^2 para que intercepte a la primera partícula.

31. La velocidad **V** de un paracaidista está dada por la fórmula:

$$\mathbf{V = \frac{gx}{c}(1 - e^{-\frac{ct}{x}})}$$

En donde **g**=**9.81** m/s^2 (gravedad terrestre), **c**=**14** kg/s (coeficiente de arrastre)
t: tiempo en segundos, **x**: masa del paracaidista en kg.

Cuando transcurrieron **7** segundos se detectó que la velocidad es **35** m/s. Determine la masa del paracaidista. E=0.001

32. Se desea dividir un pastel circular de 35 cm. de diámetro, mediante dos cortes paralelos, de tal manera que las tres porciones obtenidas tengan igual cantidad. Formule el modelo matemático (una ecuación no lineal con una incógnita**:** altura de la perpendicular del centro a cada línea de corte) y obtenga el ancho de cada uno de los tres cortes.

33. Una esfera de densidad 0.4 y radio 5 flota parcialmente sumergida en el agua. Encuentre la profundidad h que se encuentra sumergida la esfera.
Nota: requiere conocer la densidad del agua y el volumen de un segmento esférico.

3.5 Raíces reales de sistemas de ecuaciones no-lineales

En general este es un problema difícil, por lo que conviene intentar reducir el número de ecuaciones y en caso de llegar a una ecuación, poder aplicar alguno de los métodos conocidos.

Si no es posible reducir el sistema, entonces se intenta resolverlo con métodos especiales para sistemas de ecuaciones no-lineales.

Debido a que el estudio de la convergencia de estos métodos es complicado, se prefiere utilizar algún método eficiente, de tal manera que numéricamente pueda determinarse la convergencia o divergencia con los resultados obtenidos.

Una buena estrategia consiste en extender el método de Newton, cuya convergencia es de segundo orden, al caso de sistemas de ecuaciones no lineales. En esta sección se describe la fórmula para resolver un sistema de n ecuaciones no lineales y se la aplica a la solución de un sistema de dos ecuaciones. Al final de este capítulo se propone una demostración más formal de esta fórmula.

3.5.1 Fórmula iterativa de segundo orden para calcular raíces reales de sistemas de ecuaciones no-lineales

Sean **F: $f_1, f_2, \ldots, f_n$** sistema de ecuaciones no lineales con variables **X: $x_1, x_2, \ldots, x_n$**. Se requiere calcular un vector real que satisfaga al sistema **F**

En el caso de que **F** contenga una sola ecuación **f** con una variable **x**, la conocida fórmula iterativa de Newton puede escribirse de la siguiente manera:

$$x^{(k+1)} = x^{(k)} - \left(\frac{df^{(k)}}{dx}\right)^{-1} f^{(k)} \quad , k=0, 1, 2, \ldots \qquad \text{(iteraciones)}$$

Si **F** contiene **n** ecuaciones, la fórmula se puede extender, siempre que las derivadas existan:

$$X^{(k+1)} = X^{(k)} - \left(\frac{\partial F^{(k)}}{\partial X}\right)^{-1} F^{(k)} = X^{(k)} - (J^{(k)})^{-1} F^{(k)}$$

En donde:

$$X^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \ldots \\ x_n^{(k+1)} \end{bmatrix}, \quad X^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \ldots \\ x_n^{(k)} \end{bmatrix}, \quad F^{(k)} = \begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \\ \ldots \\ f_n^{(k)} \end{bmatrix}, \quad J^{(k)} = \begin{bmatrix} \frac{\partial f_1^{(k)}}{\partial x_1} & \frac{\partial f_1^{(k)}}{\partial x_2} & \ldots & \frac{\partial f_1^{(k)}}{\partial x_n} \\ \frac{\partial f_2^{(k)}}{\partial x_1} & \frac{\partial f_2^{(k)}}{\partial x_2} & \ldots & \frac{\partial f_2^{(k)}}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial f_n^{(k)}}{\partial x_1} & \frac{\partial f_n^{(k)}}{\partial x_2} & \ldots & \frac{\partial f_n^{(k)}}{\partial x_n} \end{bmatrix}$$

J es la matriz jacobiana

Esta ecuación de recurrencia se puede usar iterativamente con **k=0, 1, 2, …** partiendo de un vector inicial $X^{(0)}$ generando vectores de aproximación: $X^{(1)}$, $X^{(2)}$, $X^{(3)}$, …

3.5.2 Convergencia del método de Newton para sistemas de ecuaciones no lineales

En forma general la convergencia de este método para sistemas no lineales requiere que:

a) **$f_1, f_2, \ldots f_n$** así como sus derivadas sean continuas en la región de aplicación.
b) El determinante del Jacobiano no se anule en esta región
c) El valor inicial y los valores calculados pertenezcan a esta región, la cual incluye a la raíz que se intenta calcular

3.5.3 Algoritmo del método de Newton para sistemas de ecuaciones no lineales

Dado un sistema de ecuaciones **F = 0**, sea **J** su matriz Jacobiana. El siguiente algoritmo genera una sucesión de vectores que se espera tienda al vector solución:

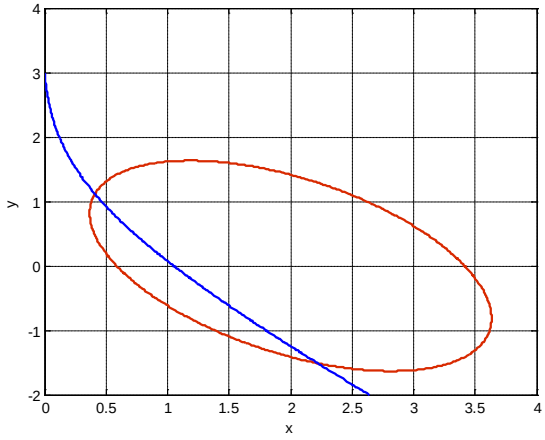
1) Elegir el vector inicial $\mathbf{X}^{(0)}$
2) Generar la sucesión de vectores con la fórmula iterativa:
$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - (\mathbf{J}^{(k)})^{-1}\mathbf{F}^{(k)}, \quad k=0, 1, 2, \ldots$$
3) Si el método converge, la sucesión de vectores $\mathbf{X}^{(k)}$ tenderá hacia un vector que satisface al sistema de ecuaciones **F = 0**

Ejemplo. *Encuentre las raíces reales del sistema:*

$$f_1(x,y) = (x-2)^2 + (y-1)^2 + xy - 3 = 0$$
$$f_2(x,y) = xe^{x+y} + y - 3 = 0$$

En el caso de dos ecuaciones con dos variables, sus gráficos pueden visualizarse en el plano. Las raíces reales son las intersecciones.

La siguiente figura obtenida con MATLAB muestra el gráfico de las dos ecuaciones.



El gráfico se obtuvo con los siguientes comandos de MATLAB y el editor de gráficos:

```
>> syms x y
>> f1=(x-2)^2 + (y-1)^2+x*y-3;
>> f2=x*exp(x+y)+y-3;
>> ezplot(f1,[0,4,-2,4]),grid on,hold on
>> ezplot(f2,[0,4,-2,4])
```

No es posible reducir el sistema a una ecuación, por lo que se debe utilizar un método para resolverlo simultáneamente con la fórmula propuesta:

Obtención de la solución con el método de Newton (sistema de dos ecuaciones no lineales)

$$f_1(x,y) = (x-2)^2 + (y-1)^2 + xy - 3 = 0$$
$$f_2(x,y) = xe^{x+y} + y - 3 = 0$$

Comenzar con el vector inicial $\mathbf{X}^{(0)} = \begin{bmatrix} x^{(0)} \\ y^{(0)} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}$ *tomado del gráfico*

Matriz jacobiana y vectores:

$$J=\begin{bmatrix}\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y}\\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y}\end{bmatrix}=\begin{bmatrix}2x+y-4 & x+2y-2\\ e^{x+y}(1+x) & xe^{x+y}+1\end{bmatrix}$$

\\

$$X=\begin{bmatrix}x\\ y\end{bmatrix},\quad F=\begin{bmatrix}f_1\\ f_2\end{bmatrix}=\begin{bmatrix}(x-2)^2+(y-1)^2+xy-3\\ xe^{x+y}+y-3\end{bmatrix}$$

Ecuación de recurrencia

$$X^{(k+1)}=X^{(k)}-(J^{(k)})^{-1}F^{(k)}$$

Primera iteración: ***k=0***

$$X^{(1)}=X^{(0)}-(J^{(0)})^{-1}F^{(0)}$$

$$\begin{bmatrix}x^{(1)}\\ y^{(1)}\end{bmatrix}=\begin{bmatrix}0.5\\ 1.0\end{bmatrix}-\begin{bmatrix}2(0.5)+1-4 & 0.5+2(1)-2\\ e^{0.5+1}(1+0.5) & 0.5e^{0.5+1}+1\end{bmatrix}^{-1}\begin{bmatrix}(0.5-2)^2+(1-1)^2+0.5(1)-3\\ 0.5e^{0.5+1}+1-3\end{bmatrix}$$

$$\begin{bmatrix}x^{(1)}\\ y^{(1)}\end{bmatrix}=\begin{bmatrix}0.5\\ 1.0\end{bmatrix}-\begin{bmatrix}-2 & 0.5\\ 6.7225 & 3.2408\end{bmatrix}^{-1}\begin{bmatrix}-0.25\\ 0.2408\end{bmatrix}$$

$$\begin{bmatrix}x^{(1)}\\ y^{(1)}\end{bmatrix}=\begin{bmatrix}0.5\\ 1.0\end{bmatrix}-\begin{bmatrix}-0.3293 & 0.0508\\ 0.6830 & 0.2032\end{bmatrix}\begin{bmatrix}-0.25\\ 0.2408\end{bmatrix}=\begin{bmatrix}0.4055\\ 1.1218\end{bmatrix}$$

3.5.4 Práctica computacional

Obtención de las raíces de las ecuaciones para el ejemplo anterior calculando directamente en la ventana de comandos de MATLAB mediante la ecuación de recurrencia:

$$X^{(k+1)}=X^{(k)}-(J^{(k)})^{-1}F^{(k)}$$

```
>> syms x y
>> f1=(x-2)^2 + (y-1)^2+x*y-3;
>> f2=x*exp(x+y)+y-3;
>> J=[diff(f1,x) diff(f1,y); diff(f2,x) diff(f2,y)]
J =
[          2*x + y - 4,                x + 2*y - 2]
[ exp(x + y) + x*exp(x + y), x*exp(x + y) + 1]
>> F=[f1; f2];
>> X=[x;y];
>> x=0.5; y=1;
>> X=eval(X)                                  Valores iniciales
X =
  0.500000000000000
  1.000000000000000
```

```
>> X=X-inv(eval(J))*eval(F)          Primera iteración
X =
   0.405451836483295
   1.121807345933181
>> x=X(1); y=X(2);
>> X=X-inv(eval(J))*eval(F)          Segunda iteración
X =
   0.409618877363502
   1.116191209478471
>> x=X(1); y=X(2);
>> X=X-inv(eval(J))*eval(F)
X =
   0.409627787030011
   1.116180137991844
>> x=X(1); y=X(2);
>> X=X-inv(eval(J))*eval(F)
X =
   0.409627787064807
   1.116180137942813
>> x=X(1); y=X(2);
>> X=X-inv(eval(J))*eval(F)
X =
   0.409627787064807
   1.116180137942814

>> eval(f1)                          Verificar la solución
ans =
   -4.440892098500626e-016
>> eval(f2)
ans =
    4.440892098500626e-016
```

3.5.5 Instrumentación computacional del método de Newton para un sistema de n ecuaciones no-lineales.

Sea **F:** $f_1, f_2, \ldots, f_n$ ecuaciones con variables independientes **X:** $x_1, x_2, \ldots, x_n$.

Ecuación de recurrencia:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - (\mathbf{J}^{(k)})^{-1}\mathbf{F}^{(k)}, \quad k=0, 1, 2, \ldots$$

En donde **J** es la matriz jacobiana del sistema

Entrada

- **f**: Vector con las ecuaciones
- **v**: Vector con las variables independientes
- **u**: Vector con valores iniciales para las variables

Salida

- **u**: Vector con los nuevos valores calculados para las variables

Nota: La convergencia será controlada interactivamente reusando la función esde la ventana de comandos. Por las propiedades de este método, la convergencia o divergencia será muy rápida.

Alternativamente, se puede incorporar a la instrumentación un ciclo con un máximo de iteraciones para que las iteraciones se realicen dentro de la función.

Las derivadas parciales se obtienen con la función **diff** y la sustitución de los valores de u en las variables se realiza con la función **subs**. La solución se la obtiene con la inversa de la matriz de las derivadas parciales J.

```
function  u = snewton(f, v, u)    %Sistemas no lineales
n=length(f);
for i=1:n                         %Obtención de la matriz jacobiana  J
   for j=1:n
      J(i,j)=diff(f(i),v(j));
   end
end
for i=1:n                         %Sustitución del vector  u  en  J
   for j=1:n
      for k=1:n
         if findstr(char(J(i,j)),char(v(k)))>0
              J(i,j)=subs(J(i,j),v(k),u(k));
         end
      end
   end
end

for i=1:n
   for j=1:n
      f(i)=subs(f(i),v(j),u(j));       %Sustitución del vector  u  en el vector  f
   end
end

u=u-inv(eval(J))*eval(f);          %Obtención de la nueva aproximación  u
```

Ejemplo. *Use la función* ***snewton*** *para encontrar una raíz real del sistema*

$$f_1(x,y) = (x-2)^2 + (y-1)^2 + xy - 3 = 0$$

$$f_2(x,y) = xe^{x+y} + y - 3 = 0$$

```
>> syms x y
>> f1=(x-2)^2 + (y-1)^2+x*y-3;
>> f2=x*exp(x+y)+y-3;
>> f=[f1; f2];
>> v=[x; y];
>> u=[0.5; 1];                          Valores iniciales tomados del gráfico
>> u=snewton(f, v, u)
u =
  0.405451836483295
  1.121807345933181
>> u=snewton(f, v, u)
u =
  0.409618877363502
  1.116191209478472
>> u=snewton(f, v, u)
u =
  0.409627787030011
  1.116180137991845
```

```
>> u=snewton(f, v, u)
u =
  0.409627787064807
  1.116180137942814

>> u=snewton(f, v, u)
u =
  0.409627787064807
  1.116180137942814
```

Se observa la rápida convergencia.
Para verificar que son raíces reales de las ecuaciones debe evaluarse ***f***

```
>> subs(f1,{x,y},{u(1),u(2)})
ans =
   4.440892098500626e-016
>> subs(f2,{x,y},{u(1),u(2)})
ans =
    0
```

Los valores obtenidos son muy pequeños, por lo cual se aceptan las raíces calculadas

Para calcular la otra raíz, tomamos del gráfico los valores iniciales cercanos a esta raíz.

```
>> u=[2.4; -1.5];
>> u=snewton(f, v, u)
u =
  2.261842718297048
 -1.535880731849205
>> u=snewton(f, v, u)
u =
  2.221421001369104
 -1.512304705819129
>> u=snewton(f, v, u)
u =
  2.220410814294533
 -1.511478104887419
>> u=snewton(f, v, u)
u =
  2.220410327256473
 -1.511477608846960
>> u=snewton(f, v, u)
u =
  2.220410327256368
 -1.511477608846834
>> u=snewton(f, v, u)
u =
  2.220410327256368
 -1.511477608846835

>> subs(f1,{x,y},{u(1),u(2)})          (Comprobar si es una solución del sistema)
ans =
  -8.881784197001252e-016
>> subs(f2,{x,y},{u(1),u(2)})
ans =
   8.881784197001252e-016
```

3.5.6 Uso de funciones de MATLAB para resolver sistemas no-lineales

La función **solve** de MATLAB se puede usar para resolver sistemas no lineales como el ejemplo anterior:

```
>> syms x y
>> f1=(x-2)^2 + (y-1)^2+x*y-3;
>> f2=x*exp(x+y)+y-3;
>> f=[f1;f2];
>> [x,y]=solve(f)
x =
   0.40962778706480689876647619089358
y =
   1.11618013794281356257169823456 5
```

*El método **solve** de MATLAB proporciona solamente una de las dos soluciones. Con esto concluimos que no siempre los programas computacionales disponibles producen todas las respuestas esperadas.*

3.5.7 Obtención de la fórmula iterativa de segundo orden para calcular raíces reales de sistemas de ecuaciones no lineales

Se considera el caso de dos ecuaciones y luego se generaliza a más ecuaciones

Sean $\mathbf{f_1(x_1, x_2) = 0}$, $\mathbf{f_2(x_1, x_2) = 0}$ dos ecuaciones no-lineales con variables $\mathbf{x_1, x_2}$.

Sean $\mathbf{r_1, r_2}$ valores reales tales que $\mathbf{f_1(r_1, r_2) = 0}$, $\mathbf{f_2(r_1, r_2) = 0}$, entonces $\mathbf{(r_1, r_2)}$ constituye una raíz real del sistema y es de interés calcularla.

Suponer que $\mathbf{f_1, f_2}$ son funciones diferenciables en alguna región cercana al punto $\mathbf{(r_1, r_2)}$

Con el desarrollo de la serie de Taylor expandimos $\mathbf{f_1, f_2}$ desde el punto $\mathbf{(x_1^{(k)}, x_2^{(k)})}$ al punto $\mathbf{(x_1^{(k+1)}, x_2^{(k+1)})}$

$$\mathbf{f_1^{(k+1)} = f_1^{(k)} + (x_1^{(k+1)} - x_1^{(k)})\frac{\partial f_1^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)})\frac{\partial f_1^{(k)}}{\partial x_2} + O(x_1^{(k+1)} - x_1^{(k)})^2 + O(x_2^{(k+1)} - x_2^{(k)})^2}$$

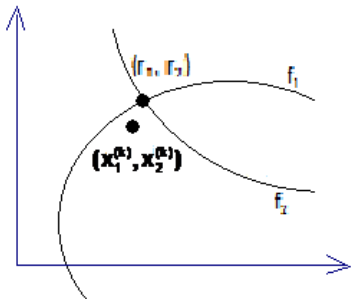
$$\mathbf{f_2^{(k+1)} = f_2^{(k)} + (x_1^{(k+1)} - x_1^{(k)})\frac{\partial f_2^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)})\frac{\partial f_2^{(k)}}{\partial x_2} + O(x_1^{(k+1)} - x_1^{(k)})^2 + O(x_2^{(k+1)} - x_2^{(k)})^2}$$

Por simplicidad se ha usado la notación: $\mathbf{f_1^{(k)} = f_1(x_1^{(k)}, x_2^{(k)})}$, $\mathbf{f_1^{(k+1)} = f_1(x_1^{(k+1)}, x_2^{(k+1)})}$, etc.

En los últimos términos de ambos desarrollos se han escrito únicamente los componentes de interés, usando la notación **O( )**.

Las siguientes suposiciones, son aceptables en una región muy cercana a $\mathbf{(r_1, r_2)}$:

$(x_1^{(k)}, x_2^{(k)})$ cercano a la raíz (r_1, r_2)



Si el método converge cuadráticamente entonces $(x_1^{(k+1)}, x_2^{(k+1)})$ estará muy cercano a (r_1, r_2)

Por lo tanto se puede aproximar:

$$f_1(x_1^{(k+1)}, x_2^{(k+1)}) \approx 0$$
$$f_2(x_1^{(k+1)}, x_2^{(k+1)}) \approx 0$$

Por otra parte, si $(x_1^{(k)}, x_2^{(k)})$ es cercano a $(x_1^{(k+1)}, x_2^{(k+1)})$, las diferencias serán pequeñas y al elevarse al cuadrado se obtendrán valores más pequeños y se los omite.

Sustituyendo en el desarrollo propuesto se obtiene como aproximación el sistema lineal:

$$0 = f_1^{(k)} + (x_1^{(k+1)} - x_1^{(k)})\frac{\partial f_1^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)})\frac{\partial f_1^{(k)}}{\partial x_2}$$
$$0 = f_2^{(k)} + (x_1^{(k+1)} - x_1^{(k)})\frac{\partial f_2^{(k)}}{\partial x_1} + (x_2^{(k+1)} - x_2^{(k)})\frac{\partial f_2^{(k)}}{\partial x_2}$$

En notación matricial:

$$-F^{(k)} = J^{(k)}(X^{(k+1)} - X^{(k)})$$

Siendo

$$F^{(k)} = \begin{bmatrix} f_1^{(k)} \\ f_2^{(k)} \end{bmatrix}, \quad X^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix}, \quad X^{(k+1)} = \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{bmatrix}, \quad J^{(k)} = \begin{bmatrix} \frac{\partial f_1^{(k)}}{\partial x_1} & \frac{\partial f_1^{(k)}}{\partial x_2} \\ \frac{\partial f_2^{(k)}}{\partial x_1} & \frac{\partial f_2^{(k)}}{\partial x_2} \end{bmatrix}$$

$$J^{(k)}X^{(k+1)} = J^{(k)}X^{(k)} - F^{(k)}$$

$$X^{(k+1)} = X^{(k)} - (J^{(k)})^{-1}F^{(k)}, \quad |J^{(k)}| \neq 0$$

Es la ecuación de recurrencia que se puede usar iterativamente con **k=0, 1, 2, …** partiendo de un vector inicial $X^{(0)}$ generando vectores de aproximación: $X^{(1)}$, $X^{(2)}$, $X^{(3)}$, …

La notación matricial y la ecuación de recurrencia se extienden directamente a sistemas de **n** ecuaciones no lineales $f_1, f_2, \ldots, f_n$ con variables $x_1, x_2, \ldots, x_n$. La matriz de las derivadas parciales **J** se denomina **jacobiano**. La ecuación de recurrencia se reduce a la fórmula de Newton si se tiene una sola ecuación.

3.5.8 Ejercicios y problemas de sistemas de ecuaciones no-lineales

1. Encuentre las soluciones del sistema de ecuaciones dado:

$$\sin(x) + e^{y} - xy = 5$$
$$x^2 + y^3 - 3xy = 7$$

a) Grafique las ecuaciones en el intervalo [-4, 4, -4, 4] y observe que hay dos raíces reales. Elija del gráfico, valores aproximados para cada raíz.
b) Use iterativamente la función **snewton**:
c) Compruebe que las soluciones calculadas satisfacen a las ecuaciones
d) Calcule las soluciones con la función **solve** de MATLAB y compare

2. Encuentre las soluciones del sistema de ecuaciones dado:

$$\cos(x+y) + xy = 3$$
$$3(x - 2)^2 - 2(y - 3)^2 = 5xy$$

a) Grafique las ecuaciones en el intervalo [-6, 6, -6, 6] y observe que hay dos raíces reales. Elija del gráfico, valores aproximados para cada raíz.
b) Use iterativamente la función **snewton**:
c) Compruebe que las soluciones calculadas satisfacen a las ecuaciones
d) Calcule las soluciones con la función **solve** de MATLAB y compare

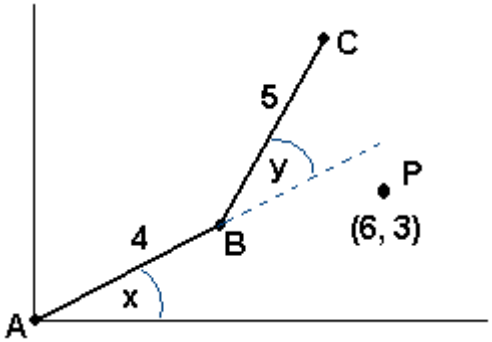
3. Encuentre las soluciones del sistema de ecuaciones dado:

$$\exp(x+y) + x - y = 3$$
$$\text{sen}(x+y) - 2x + y = 1$$

a) Grafique las ecuaciones en el intervalo [-3, 3, -3, 3] y observe que hay una raíz real. Elija del gráfico, valores aproximados para la raíz.
b) Use iterativamente la función **snewton**:
c) Compruebe que la solución calculada satisface a las ecuaciones
d) Calcule la solución con la función **solve** de MATLAB y compare.

4. El siguiente gráfico representa un mecanismo compuesto de dos brazos articulados en los puntos **A** y **B**. Las longitudes de los brazos son **4** y **5**. Determine los ángulos **X** y **Y** para que el extremo **C** coincida con el punto **P** de coordenadas **(6,3).**

El modelo matemático es un sistema de dos ecuaciones no lineales que debe resolver con el método de Newton para sistemas de ecuaciones no lineales (**snewton**).



4 MÉTODOS DIRECTOS PARA RESOLVER SISTEMAS DE ECUACIONES LINEALES

En este capítulo se estudia el componente algorítmico y computacional de los métodos directos para resolver sistemas de ecuaciones lineales.

Ejemplo. *Un comerciante compra tres productos: **A, B, C**, pero en las facturas únicamente consta la cantidad comprada y el valor total de la compra. Se necesita determinar el precio unitario de cada producto. Para esto dispone de tres facturas con los siguientes datos:*

Factura	*Cantidad de **A***	*Cantidad de **B***	*Cantidad de **C***	*Valor pagado*
1	*4*	*2*	*5*	*$18.00*
2	*2*	*5*	*8*	*$27.30*
3	*2*	*4*	*3*	*$16.20*

Análisis

Sean $\mathbf{x_1, x_2, x_3}$ *variables que representan al precio unitario de cada producto. Entonces, se puede escribir:*

$$\mathbf{4x_1 + 2x_2 + 5x_3 = 18.00}$$
$$\mathbf{2x_1 + 5x_2 + 8x_3 = 27.30}$$
$$\mathbf{2x_1 + 4x_2 + 3x_3 = 16.20}$$

El modelo matemático resultante es un sistema lineal de tres ecuaciones con tres variables.

En general, se desea resolver un sistema de **n** ecuaciones lineales con **n** variables

$$\mathbf{a_{1,1}x_1 + a_{1,2}x_1 + \ldots + a_{1,n}x_n = b_1}$$
$$\mathbf{a_{2,1}x_1 + a_{2,2}x_1 + \ldots + a_{2,n}x_n = b_2}$$
$$\mathbf{\ldots}$$
$$\mathbf{a_{n,1}x_1 + a_{n,2}x_1 + \ldots + a_{n,n}x_n = b_n}$$

En donde

$\mathbf{a_{i,j}} \in \Re$: Coeficientes

$\mathbf{b_i} \in \Re$: Constantes

$\mathbf{x_i} \in \Re$: Variables cuyo valor debe determinarse

En notación matricial:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \ldots & & & \ldots \\ a_{n,1} & a_{n,1} & \ldots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}$$

Simbólicamente

$$\mathbf{AX = B}$$

Siendo

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \ldots & & & \ldots \\ a_{n,1} & a_{n,1} & \ldots & a_{n,n} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}; \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}$$

4.1 Determinantes y sistemas de ecuaciones lineales

Sea **A** la matriz de coeficientes del sistema **AX = B.** Sea $\mathbf{A^{-1}}$ su inversa y **|A|** su determinante. La relación entre **|A|** y la existencia de la solución **X** se establece con la siguiente definición:

$$\mathbf{A^{-1}} = \frac{\mathbf{[adj(A)]^t}}{\mathbf{|A|}},$$

En donde $\mathbf{[adj(A)]^t}$ es la transpuesta de la adjunta de la matriz **A**.

Si $\mathbf{|A| \neq 0}$, entonces $\mathbf{A^{-1}}$ existe, y se puede escribir:

$$\mathbf{AX = B} \quad \Rightarrow \quad \mathbf{A^{-1}AX = A^{-1}B} \quad \Rightarrow \quad \mathbf{IX = A^{-1}B} \quad \Rightarrow \quad \mathbf{X = A^{-1}B}$$

En donde **I** es la matriz identidad. En resumen, si $\mathbf{|A| \neq 0}$ entonces **X** existe y además es único.

4.2 Método de Gauss - Jordan

La estrategia de este método consiste en transformar la matriz **A** del sistema **AX = B** y reducirla a la matriz identidad **I**. Según el enunciado anterior, esto es posible si **| A | ≠ 0.** Aplicando simultáneamente las mismas transformaciones al vector **B**, este se convertirá en el vector solución $\mathbf{A^{-1}B}$.

En caso de que esta solución exista, el procedimiento debe transformar las ecuaciones mediante operaciones lineales que no modifiquen la solución del sistema original, estas pueden ser**:**

a) Intercambiar ecuaciones
b) Multiplicar ecuaciones por alguna constante no nula
c) Sumar alguna ecuación a otra ecuación

Ejemplo. *Con el Método de Gauss-Jordan resuelva el siguiente sistema de ecuaciones lineales correspondiente al problema planteado al inicio del capítulo*

$$\begin{aligned} 4x_1 + 2x_2 + 5x_3 &= 18.00 \\ 2x_1 + 5x_2 + 8x_3 &= 27.30 \\ 2x_1 + 4x_2 + 3x_3 &= 16.20 \end{aligned}$$

Solución: *Se define la matriz aumentada* ***A | B*** *para transformar simultáneamente* ***A*** *y* ***B****:*

$$\mathbf{A \mid B} = \left[\begin{array}{ccc|c} 4 & 2 & 5 & 18.00 \\ 2 & 5 & 8 & 27.30 \\ 2 & 4 & 3 & 16.20 \end{array}\right]$$

Las transformaciones sucesivas de la matriz aumentada se describen en los siguientes pasos:

Dividir fila 1 para 4

1.0000	***0.5000***	***1.2500***	***4.5000***
2.0000	*5.0000*	*8.0000*	*27.3000*
2.0000	*4.0000*	*3.0000*	*16.2000*

Restar de cada fila, la fila 1 multiplicada por el elemento de la columna 1

1.0000	*0.5000*	*1.2500*	*4.5000*
0	***4.0000***	***5.5000***	***18.3000***
0	***3.0000***	***0.5000***	***7.2000***

Dividir fila 2 para 4

1.0000	*0.5000*	*1.2500*	*4.5000*
0	***1.0000***	***1.3750***	***4.5750***
0	*3.0000*	*0.5000*	*7.2000*

Restar de cada fila, la fila 2 multiplicada por el elemento de la columna 2

1.0000	*0*	***0.5625***	***2.2125***
0	*1.0000*	*1.3750*	*4.5750*
0	*0*	***-3.6250***	***-6.5250***

Dividir fila 3 para -3.625

1.0000	*0*	*0.5625*	*2.2125*
0	*1.0000*	*1.3750*	*4.5750*
0	*0*	***1.0000***	***1.8000***

Restar de cada fila, la fila 3 multiplicada por el elemento de la columna 3

1.0000	*0*	*0*	*1.2000*
0	*1.0000*	*0*	*2.1000*
0	*0*	*1.0000*	*1.8000*

La matriz de los coeficientes ha sido transformada a la ***matriz identidad****.*

Simultáneamente, las mismas transformaciones han convertido a la última columna en el ***vector solución****:*

$$\mathbf{X} = \begin{bmatrix} \mathbf{1.2} \\ \mathbf{2.1} \\ \mathbf{1.8} \end{bmatrix}$$

Como antes, la solución debe verificarse en el sistema

4.2.1 Práctica computacional

Resolver el ejemplo anterior en la ventana de comandos con la notación matricial

```
>> a=[4 2 5; 2 5 8; 2 4 3]                    Definición de la matriz
a =
    4    2    5
    2    5    8
    2    4    3
>> b=[18.0; 27.3; 16.2]                       Vector de constantes
b =
  18.0000
  27.3000
  16.2000
>> a=[a, b]                                   Matriz aumentada
a =
   4.0000   2.0000   5.0000  18.0000
   2.0000   5.0000   8.0000  27.3000
   2.0000   4.0000   3.0000  16.2000
>> a(1,1:4)=a(1,1:4)/a(1,1)                   Normalizar fila 1
a =
   1.0000   0.5000   1.2500   4.5000
   2.0000   5.0000   8.0000  27.3000
   2.0000   4.0000   3.0000  16.2000
>> a(2,1:4)=a(2,1:4)-a(2,1)*a(1,1:4)          Reducir fila 2
a =
   1.0000   0.5000   1.2500   4.5000
        0   4.0000   5.5000  18.3000
   2.0000   4.0000   3.0000  16.2000
>> a(3,1:4)=a(3,1:4)-a(3,1)*a(1,1:4)          Reducir fila 3
a =
   1.0000   0.5000   1.2500   4.5000
        0   4.0000   5.5000  18.3000
        0   3.0000   0.5000   7.2000
>> a(2,2:4)=a(2,2:4)/a(2,2)                   Normalizar fila 2
a =
   1.0000   0.5000   1.2500   4.5000
        0   1.0000   1.3750   4.5750
        0   3.0000   0.5000   7.2000
>> a(1,2:4)=a(1,2:4)-a(1,2)*a(2,2:4)          Reducir fila 1
a =
   1.0000         0   0.5625   2.2125
        0   1.0000   1.3750   4.5750
        0   3.0000   0.5000   7.2000
>> a(3,2:4)=a(3,2:4)-a(3,2)*a(2,2:4)          Reducir fila 3
a =
   1.0000         0   0.5625   2.2125
        0   1.0000   1.3750   4.5750
        0         0  -3.6250  -6.5250
>> a(3,3:4)=a(3,3:4)/a(3,3)                   Normalizar fila 3
a =
   1.0000       0   0.5625   2.2125
      0   1.0000   1.3750   4.5750
      0         0   1.0000   1.8000
```

```
>> a(1,3:4)=a(1,3:4)-a(1,3)*a(3,3:4)                    Reducir fila 1
a =
   1.0000         0         0    1.2000
        0    1.0000    1.3750    4.5750
        0         0    1.0000    1.8000
>> a(2,3:4)=a(2,3:4)-a(2,3)*a(3,3:4)                    Reducir fila 2
a =
   1.0000         0         0    1.2000
        0    1.0000         0    2.1000
        0         0    1.0000    1.8000
>> x=a(1:3,4)                                           Vector solución
x =
   1.2000
   2.1000
   1.8000

>> a*x                                                  Verificar la solución
ans =
  18.0000
  27.3000
  16.2000
```

4.2.2 Formulación del método de Gauss-Jordan y algoritmo

Para establecer la descripción algorítmica, conviene definir la matriz aumentada **A** con el vector **B** pues deben realizarse simultáneamente las mismas operaciones**:**

$$\mathbf{A \mid B} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix}$$

En donde se ha agregado la columna n+1 con el vector de las constantes:

$$a_{i,n+1} = b_i, \quad i = 1, 2, 3, \dots, n$$ (columna **n+1** de la matriz aumentada)

El objetivo es transformar esta matriz y llevarla a la forma de la matriz identidad **I**:

$$\mathbf{A \mid B} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix} \rightarrow \dots \rightarrow \left[\begin{array}{cccc|c} 1 & 0 & \dots & 0 & a_{1,n+1} \\ 0 & 1 & \dots & 0 & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{n,n+1} \end{array}\right]$$

Si es posible realizar esta transformación, entonces los valores que quedan en la última columna constituirán el vector solución **X**

Las transformaciones deben ser realizadas en forma sistemática en **n** etapas, obteniendo sucesivamente en cada etapa, cada columna de la matriz identidad, de izquierda a derecha.

En cada etapa, primero se hará que el elemento en la diagonal tome el valor **1**. Luego se hará que los demás elementos de la columna tomen el valor **0.**

$$\begin{bmatrix} 1 & 0 & \dots & 0 & a_{1,n+1} \\ 0 & 1 & \dots & 0 & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{n,n+1} \end{bmatrix}$$

Etapa 1

Etapa 2

Etapa n

Etapa 1

Normalizar la fila 1: (colocar **1** en el lugar del elemento $a_{1,1}$

$a_{1,j} \leftarrow a_{1,j} / a_{1,1}$ **j=1, 2, ..., n+1;** supones que $a_{1,1} \neq 0$

Reducir las otras filas: (colocar **0** en los otros elementos de la columna **1**)

$a_{i,j} \leftarrow a_{i,j} - a_{i,1}a_{1,j}$, **j=1, 2, ..., n+1; i=2, 3, ..., n**

$$A \mid B = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & a_{2,n+1} \\ \dots & & & \dots & \dots \\ a_{n,1} & a_{n,1} & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,n} & a_{1,n+1} \\ 0 & a_{2,2} & \dots & a_{2,n} & a_{2,n+1} \\ \dots & & & \dots & \dots \\ 0 & a_{n,1} & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix}$$

Valores transformados

Etapa 2

Normalizar la fila 2: (colocar **1** en el lugar del elemento $a_{2,2}$

$a_{2,j} \leftarrow a_{2,j} / a_{2,2}$ **j=2, 3, ..., n+1;** $a_{2,2} \neq 0$

Reducir las otras filas: (colocar **0** en los otros elementos de la columna **2**

$a_{i,j} \leftarrow a_{i,j} - a_{i,2}a_{2,j}$, **j=2, 3, ..., n+1; i=1, 3, ..., n**

$$\begin{bmatrix} 1 & a_{1,2} & \dots & a_{1,n} & a_{1,n+1} \\ 0 & a_{2,2} & \dots & a_{2,n} & a_{2,n+1} \\ \dots & & & \dots & \dots \\ 0 & a_{n,1} & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \dots & a_{1,n} & a_{1,n+1} \\ 0 & 1 & \dots & a_{2,n} & a_{2,n+1} \\ \dots & & & \dots & \dots \\ 0 & 0 & \dots & a_{n,n} & a_{n,n+1} \end{bmatrix}$$

Valores transformados

La última columna contendrá el vector solución

La formulación obtenida en estas dos etapas se puede generalizar y con ella construir el algoritmo:

ALGORITMO BÁSICO DE GAUSS-JORDAN

a: matriz de coeficientes, aumentada con el vector **b** del sistema de **n** ecuaciones lineales

```
Para  e = 1, 2, . . ., n
    Para j=e, e+1, ..., n+1
        a_{e,j} ← a_{e,j} / a_{e,e}          Normalizar la fila e ( a_{e,e} ≠ 0 )
    Fin
    Para i=1, 2, …, i -1, i +1,… n
        Para j=e, e+1, ..., n+1
            a_{i,j} ← a_{i,j} – a_{i,e} a_{e,j}     Reducir las otras filas
        Fin
    Fin
Fin
Para i=1,2,...,n
    x_i ← a_{i,n+1}                          La última columna contendrá la solución
Fin
```

4.2.3 Eficiencia del método de Gauss-Jordan

El método de Gauss-Jordan es un método directo. Los métodos directos pueden estar afectados por el error de redondeo, es decir los errores en la representación de los números que se producen en las operaciones aritméticas. Para cuantificar la magnitud del error de redondeo se define la función de eficiencia del método.

Sea **n** el tamaño del problema y **T(n)** la cantidad de operaciones aritméticas que se realizan

En la normalización: $T(n) = O(n^2)$ (Dos ciclos anidados)

En la reducción: $T(n) = O(n^3)$ (Tres ciclos anidados)

Por lo tanto, este método es de tercer orden: **$T(n) = O(n^3)$**

Mediante un conteo recorriendo los ciclos del algoritmo, se puede determinar la función de eficiencia para este método directo:

e	**i**	**j**
1	n-1	n+1
2	n-1	n
.	.	.
.	.	.
.	.	.
n-1	n-1	3
n	n-1	2

T(n) = (n-1)(2 + 3 + n + (n+1)) = (n-1) (3 + n) (n/2) = **$n^3/2 + 2n^2/2 + 3n/2$**

4.2.4 Instrumentación computacional

En esta primera versión del algoritmo se supondrá que el determinante de la matriz es diferente de cero y que no se requiere intercambiar filas.

La codificación en MATLAB sigue directamente la formulación matemática descrita anteriormente. Se usa notación compacta para manejo de matrices

```
function x=gaussjordan(a,b)
n=length(b);
a=[a,b];                                        %matriz aumentada
for e=1:n
    a(e,e:n+1)=a(e,e:n+1)/a(e,e);               %normalizar fila e
    for i=1:n
        if i~=e
            a(i,e:n+1)=a(i,e:n+1)-a(i,e)*a(e,e:n+1);   %reducir otras filas
        end
    end
end
x=a(1:n,n+1);                                   %vector solución
```

Ejemplo. *Desde la ventana de comandos de MATLAB, use la función Gauss-Jordan para resolver el sistema:*

$$\begin{bmatrix} 2 & 3 & 7 \\ -2 & 5 & 6 \\ 8 & 9 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 8 \end{bmatrix}$$

Escriba en la ventana de comandos de MATLAB

```
>> a=[2, 3, 7; -2, 5, 6; 8, 9, 4];      Matriz de coeficientes
>> b=[3; 5; 8];                         Vector de constantes
>> x=gaussjordan(a,b)                   Llamada a la función
x =
   -0.0556                              Solución proporcionada por MATLAB
    0.9150
    0.0523
>> a*x                                  Verificar la solución
ans =
    3.0000                              La solución satisface al sistema
    5.0000
    8.0000
```

Cuando se dispone de la instrumentación computacional de un algoritmo, se puede obtener experimentalmente su eficiencia registrando, para diferentes valores de **n**, el tiempo de ejecución del algoritmo. Este tiempo depende de la velocidad del procesador del dispositivo computacional, pero es proporcional a T(n).

MATLAB dispone de las funciones **tic, toc** para registrar tiempo de ejecución, mientras que para las pruebas se pueden generar matrices y vectores con números aleatorios. Se presentan algunos resultados con obtenidos con un procesador intel core i5 y la versión 7.01 de MATLAB:

n=100, t=0.0781 seg.
n=200, t=0.3859 seg.
n=300, t=1.0336 seg.
n=400, t=2.0758 seg.

Se observa que T(n) tiene crecimiento tipo potencial

4.2.5 Obtención de la inversa de una matriz

Para encontrar la matriz inversa se puede usar el método de Gauss-Jordan.

Sea $\mathbf{A}$ una matriz cuadrada cuyo determinante es diferente de cero.
Sean $\mathbf{t_1, t_2, \ldots, t_{m-1}, t_m}$ las transformaciones lineales del método de Gauss-Jordan que transforman la matriz **A** en la matriz identidad $\mathbf{I}$ incluyendo intercambios de filas

$$\mathbf{t_m\ t_{m-1} \ldots t_2\ t_1\ A = I}$$

Entonces se puede escribir

$$\mathbf{t_m\ t_{m-1} \ldots t_2\ t_1\ A^{-1}\ A = A^{-1} I} \quad \Rightarrow \quad \mathbf{t_m\ t_{m-1} \ldots t_2\ t_1\ I = A^{-1}}$$

Lo cual significa que las mismas transformaciones que convierten $\mathbf{A}$ en la matriz $\mathbf{I}$, convertirán la matriz $\mathbf{I}$ en la matriz $\mathbf{A^{-1}}$.

Para aplicar este algoritmo, suponiendo que se desea conocer la matriz $\mathbf{A^{-1}}$, se debe aumentar la matriz anterior con la matriz $\mathbf{I}$: $\mathbf{A \mid B \mid I}$
Las transformaciones aplicadas simultáneamente proporcionarán finalmente el vector solución **X** y la matriz identidad $\mathbf{A^{-1}}$

Ejemplo. *Con el Método de Gauss-Jordan resuelva el sistema de ecuaciones siguiente y simultáneamente obtenga la matriz inversa:*

$$\begin{aligned} 4x_1 + 2x_2 + 5x_3 &= 18.00 \\ 2x_1 + 5x_2 + 8x_3 &= 27.30 \\ 2x_1 + 4x_2 + 3x_3 &= 16.20 \end{aligned}$$

Solución. *La matriz aumentada es:*

$$\mathbf{A \mid B} = \left[\begin{array}{ccc|c|ccc} 4 & 2 & 5 & 18.00 & 1 & 0 & 0 \\ 2 & 5 & 8 & 27.30 & 0 & 1 & 1 \\ 2 & 4 & 3 & 16.20 & 0 & 0 & 1 \end{array}\right]$$

Cálculos

Normalizar fila 1 y reducir filas 2 y 3

1.0000	0.5000	1.2500	4.5000	0.2500	0	0
0	4.0000	5.5000	18.3000	-0.5000	1.0000	0
0	3.0000	0.5000	7.2000	-0.5000	0	1.0000

Normalizar fila 2 y reducir filas 1 y 3

1.0000	0	0.5625	2.2125	0.3125	-0.1250	0
0	1.0000	1.3750	4.5750	-0.1250	0.2500	0
0	0	-3.6250	-6.5250	-0.1250	-0.7500	1.0000

Normalizar fila 3 y reducir filas 1 y 2

1.0000	0	0	1.2000	0.2931	-0.2414	0.1552
0	1.0000	0	2.1000	-0.1724	-0.0345	0.3793
0	0	1.0000	1.8000	0.0345	0.2069	-0.2759

Solución del sistema

$$\mathbf{X} = \begin{bmatrix} 1.2 \\ 2.1 \\ 1.8 \end{bmatrix}$$

Matriz inversa

$$\mathbf{A}^{-1} = \begin{bmatrix} 0.2931 & -0.2414 & 0.1552 \\ -0.1724 & -0.0345 & 0.3793 \\ 0.0345 & 0.2069 & -0.2759 \end{bmatrix}$$

4.3 Método de Gauss

El método de Gauss es similar al método de Gauss-Jordan. Aquí se trata de transformar la matriz del sistema a una forma triangular superior**.** Si esto es posible entonces la solución se puede obtener resolviendo el sistema triangular resultante.

Ejemplo. *Con el Método de Gauss resuelva el sistema de ecuaciones lineales del problema planteado al inicio de este capítulo*

$$4x_1 + 2x_2 + 5x_3 = 18.00$$
$$2x_1 + 5x_2 + 8x_3 = 27.30$$
$$2x_1 + 4x_2 + 3x_3 = 16.20$$

Solución: *Se define la matriz aumentada* ***A | B*** *para transformar simultáneamente* ***A*** *y* ***B****:*

$$A \mid B = \left[\begin{array}{ccc|c} 4 & 2 & 5 & 18.00 \\ 2 & 5 & 8 & 27.30 \\ 2 & 4 & 3 & 16.20 \end{array}\right]$$

Las transformaciones sucesivas de la matriz aumentada se describen en los siguientes cuadros

Dividir fila 1 para 4

1.0000	*0.5000*	*1.2500*	*4.5000*
2.0000	*5.0000*	*8.0000*	*27.3000*
2.0000	*4.0000*	*3.0000*	*16.2000*

Restar de cada fila, la fila 1 multiplicada por el elemento de la columna 1

1.0000	*0.5000*	*1.2500*	*4.5000*
0	*4.0000*	*5.5000*	*18.3000*
0	*3.0000*	*0.5000*	*7.2000*

Dividir fila 2 para 4

1.0000	*0.5000*	*1.2500*	*4.5000*
0	*1.0000*	*1.3750*	*4.5750*
0	*3.0000*	*0.5000*	*7.2000*

Restar de la fila, la fila 2 multiplicada por el elemento de la columna 2

1.0000	*0.5000*	*1.2500*	*4.5000*
0	*1.0000*	*1.3750*	*4.5750*
0	*0*	*-3.6250*	*-6.5250*

Dividir fila 3 para -3.625

1.0000	*0.5000*	*1.2500*	*4.5000*
0	*1.0000*	*1.3750*	*4.5750*
0	*0*	*1.0000*	*1.8000*

La matriz de los coeficientes ha sido transformada a la forma triangular superior

De este sistema se obtiene la solución mediante una sustitución directa comenzando por el final:

$$x_3 = 1.8$$
$$x_2 = 4.575 - 1.375(1.8) = 2.1$$
$$x_1 = 4.5 - 0.5(2.1) - 1.25(1.8) = 1.2$$

4.3.1 Formulación del método de Gauss y algoritmo

Para unificar la descripción algorítmica, es conveniente aumentar la matriz **A** con el vector **B** pues deben realizarse las mismas operaciones simultáneamente:

$$\mathbf{A \mid B} = \left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & a_{2,n+1} \\ \cdots & & & \cdots & \cdots \\ a_{n,1} & a_{n,1} & \cdots & a_{n,n} & a_{n,n+1} \end{array}\right]$$

En donde la columna de los coeficientes se define:

$$a_{i,n+1} = b_i \,, \quad i=1, 2, 3, \ldots, n$$

La formulación se obtiene directamente del método de Gauss-Jordan en la que la reducción de las filas únicamente se realiza en la sub-matriz triangular inferior. Las transformaciones convierten la matriz aumentada en la forma triangular superior:

$$\mathbf{A \mid B} = \left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & a_{2,n+1} \\ \cdots & & & \cdots & \cdots \\ a_{n,1} & a_{n,1} & \cdots & a_{n,n} & a_{n,n+1} \end{array}\right] \rightarrow \ldots \rightarrow \left[\begin{array}{cccc|c} 1 & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} \\ 0 & 1 & \cdots & a_{2,n} & a_{2,n+1} \\ \cdots & & & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & a_{n,n+1} \end{array}\right]$$

De sistema triangular se puede obtener directamente la solución. Para facilitar la notación expandimos la forma triangular final obtenida:

$$\left[\begin{array}{cccccc|c} 1 & a_{1,2} & \cdots & a_{1,n-2} & a_{1,n-1} & a_{1,n} & a_{1,n+1} \\ 0 & 1 & \cdots & a_{2,n-2} & a_{2,n-1} & a_{2,n} & a_{2,n+1} \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & a_{n-2,n-1} & a_{n-2,n} & a_{n-2,n+1} \\ 0 & 0 & \cdots & 0 & 1 & a_{n-1,n} & a_{n-1,n+1} \\ 0 & 0 & \cdots & 0 & 0 & 1 & a_{n,n+1} \end{array}\right]$$

La solución se obtiene del sistema triangular superior comenzando desde el final:

$$x_n \leftarrow a_{n,\,n+1}$$
$$x_{n-1} \leftarrow a_{n-1,\,n+1} - a_{n-1,\,n}x_n$$
$$x_{n-2} \leftarrow a_{n-2,\,n+1} - (a_{n-2,\,n-1}x_{n-1} - a_{n-2,\,n}x_n)$$
$$\ldots \text{ etc}$$

Con la formulación del método anterior (Gauss-Jordan) modificando el índice de las filas para reducir solamente debajo de la diagonal y con la formulación para resolver el sistema triangular resultante, se define el algoritmo para el método de Gauss:

ALGORITMO BÁSICO DE GAUSS

a: matriz de coeficientes, aumentada con el vector **b** del sistema de **n** ecuaciones lineales

```
Para  e = 1, 2, . . ., n
    Para j=e, e+1, ..., n+1
        a_{e,j} ← a_{e,j} / a_{e,e}          Normalizar la fila e  ( a_{e,e} ≠ 0 )
    Fin
    Para i=e+1, e+2, … n
        Para j=e, e+1, ..., n+1
            a_{i,j} ← a_{i,j} – a_{i,e} a_{e,j}     Reducir la filas debajo de la fila e
        Fin
    Fin
Fin
x_n ← a_{n, n+1}                             Resolver el sistema triangular superior
Para i=n-1, n-2, ..., 1
    s ← 0
    Para j=i+1, i+2, ..., n
        s ← s + a_{i,j} x_j
    Fin
    x_i ← a_{i,n+1} – s
Fin
```

4.3.2 Eficiencia del método de Gauss

Sea **n** el tamaño del problema y **T(n)** la cantidad de operaciones aritméticas que se realizan

En la normalización:	$T(n) = O(n^2)$	(dos ciclos anidados)
En la reducción:	$T(n) = O(n^3)$	(tres ciclos anidados)
En la obtención de la solución:	$T(n) = O(n^2)$	(dos ciclos anidados)

Por lo tanto, este método es de tercer orden: $\mathbf{T(n) = O(n^3)}$

Mediante un recorrido de los ciclos del algoritmo, se puede determinar en forma más precisa: $\mathbf{T(n) = n^3/3 + O(n^2)}$ con lo que se puede concluir que el método de Gauss es más eficiente que el método de Gauss-Jordan. Se supone **n** grande. Esta diferencia se la puede constatar experimentalmente resolviendo sistemas grandes y registrando el tiempo de ejecución.

4.3.3 Instrumentación computacional

En esta primera versión del algoritmo se supondrá que el determinante de la matriz es diferente de cero y que no se requiere intercambiar filas. La codificación en MATLAB sigue directamente la formulación matemática descrita anteriormente. Se usa notación compacta para manejo de matrices.

```
function x=gauss1(a,b)
n=length(b);
a=[a,b];                                  %Matriz aumentada
for e=1:n
   a(e,e:n+1)=a(e,e:n+1)/a(e,e);          %Normalizar la fila e
   for i=e+1:n                            %Reducir otras filas
      a(i,e:n+1)=a(i,e:n+1)-a(i,e)*a(e,e:n+1);
  end
end
x(n,1)=a(n,n+1);                          %Solución del sistema triangular
for i=n-1:-1:1
  x(i,1)=a(i,n+1)-a(i,i+1:n)*x(i+1:n,1);
end
```

4.3.4 Estrategia de pivoteo

Al examinar la eficiencia de los métodos directos para resolver sistemas de ecuaciones lineales se observa que la operación de multiplicación está en la sección crítica del algoritmo con eficiencia **$O(n^3)$**.

Formulación del método de Gauss:

Etapa **e = 1, 2, . . ., n**

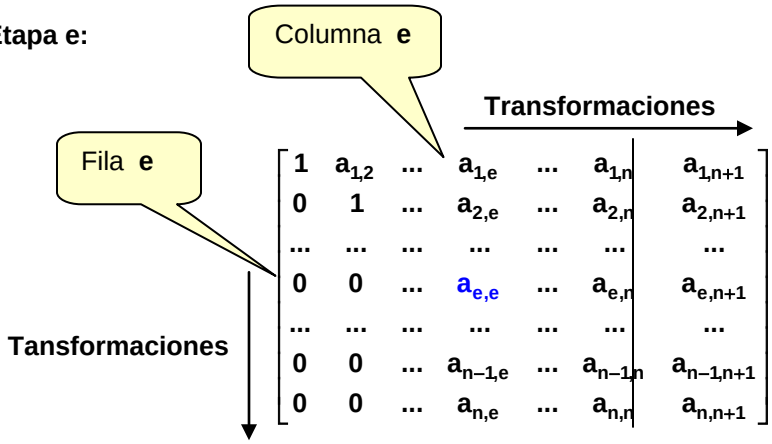
Normalizar la fila **e:**

$$a_{e,j} \leftarrow a_{e,j} / a_{e,e} \ , \qquad j=e, e+1, ..., n+1; \qquad a_{e,e} \neq 0$$

Reducir las otras filas**:**

$$a_{i,j} \leftarrow a_{i,j} - a_{i,e} a_{e,j} \ , \qquad j=e, e+1, ..., n+1; \qquad i=e+1, e+2, ..., n$$

Etapa e:



Recordando la definición del error de redondeo propagado en la multiplicación:

$$E_{XY} = \overline{X} E_Y + \overline{Y} E_X$$

Una estrategia para disminuir el error de redondeo consiste en reducir el valor de los operandos que intervienen en la multiplicación.

En la estrategia de **"Pivoteo Parcial",** antes de normalizar la fila **e** se busca en la columna **e** de cada fila **i = e, e+1, . . ., n** cual es el elemento con mayor magnitud. Si se usa este elemento como divisor para la fila **e**, el cociente $a_{e,j}$ tendrá el menor valor. Este factor permite disminuir el error cuando se realiza la etapa de **reducción** de las otras filas.

Por otra parte, si en esta estrategia de búsqueda, el valor elegido como el mayor divisor no es diferente de cero, se concluye que en el sistema existen ecuaciones redundantes o incompatibles, entonces el sistema no tiene solución única y el algoritmo debe terminar

4.3.5 Instrumentación computacional del método de Gauss con pivoteo

La siguiente instrumentación en MATLAB del método de eliminación de Gauss incluye la formulación descrita y la estrategia de "pivoteo parcial" vista anteriormente, en notación matricial compacta de MATLAB. En esta instrumentación final se incluye un chequeo del divisor para prevenir el caso de que el sistema sea singular aunque, por los errores de redondeo, no sea exactamente igual a cero. También se verifica que sea una matriz cuadrada.

```
function x=gauss(a,b)
[n,m]=size(a);
 if n~=m                                  %Verificar si la matriz es cuadrada
    x=[ ];
    return;
 end
a=[a,b];                                  %Matriz aumentada
for e=1:n
  [z, p]=max(abs(a(e:n,e)));              %Pivoteo por filas
  p=p+e-1;
  t=a(e,e:n+1);                           %Intercambio de filas
  a(e,e:n+1)=a(p,e:n+1);
  a(p,e:n+1)=t;
  if abs(a(e,e))<1.0e-10                  %Si el divisor es ~cero, no hay solución
    x=[ ];
    return;
  end
  a(e,e:n+1)=a(e,e:n+1)/a(e,e);           %Normalizar la fila e
  for i=e+1:n                             %Reducir otras filas
      a(i,e:n+1)=a(i,e:n+1)-a(i,e)*a(e,e:n+1);
  end
end
x(n,1)=a(n,n+1);                          %Solución del sistema triangular
for i=n-1:-1:1
  x(i,1)=a(i,n+1)-a(i,i+1:n)*x(i+1:n,1);
end
```

La función **[z,p]=max(v)** de MATLAB entrega en **z** el mayor valor, y en **p** el número de la fila en la que está ubicado este valor en el vector **v**

Ejemplo. *Desde la ventana de comandos de MATLAB, use la función Gauss para resolver el sistema:*

$$\begin{bmatrix} 2 & 3 & 7 \\ -2 & 5 & 6 \\ 8 & 9 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 8 \end{bmatrix}$$

Escriba en la ventana de comandos de MATLAB

```
>> a=[2, 3, 7; -2, 5, 6; 8, 9, 4];      Matriz de coeficientes
>> b=[3; 5; 8];                         Vector de constantes
>> x=gauss(a,b)                         Llamada a la función
  x =
    -0.0556                             Solución calculada
     0.9150
     0.0523
```

4.3.6 Funciones de MATLAB para sistemas de ecuaciones lineales

MATLAB tiene un soporte muy potente para resolver sistemas de ecuaciones lineales. Sugerimos entrar al sistema de ayuda de MATLAB y revisar la amplia información relacionada con este tema.

La forma más simple de resolver un sistema lineal, si la matriz de coeficientes es cuadrada y no-singular, es usando la definición de inversa de una matriz MATLAB.

Ejemplo. *Resuelva el ejemplo anterior con la función* ***inv*** *de MATLAB*

```
>> a=[2, 3, 7; -2, 5, 6; 8, 9, 4];      Matriz de coeficientes
>> b=[3; 5; 8];                         Vector de constantes
>> x=inv(a)*b                           Invertir la matriz de coeficientes
   x =                                  Solución calculada por MATLAB
      -0.0556
       0.9150
       0.0523
```

Una forma más general para resolver sistemas lineales, incluyendo sistemas singulares se puede hacer con la función **rref** de MATLAB. Esta función reduce una matriz a su forma escalonada con 1's en la diagonal.

Ejemplo. *Resuelva el ejemplo anterior con la función* ***rref*** *de MATLAB*

```
>> a=[2 3 7;-2 5 6;8 9 4];
>> b=[3;5;8];
>> a=[a, b];                                   Matriz aumentada
>> c=rref(a)
 c =
    1.0000         0         0   -0.0556
         0    1.0000         0    0.9150
         0         0    1.0000    0.0523
```

La última columna de la matriz aumentada resultante contiene la solución.

4.3.7 Cálculo del determinante de una matriz

El algoritmo de Gauss transforma la matriz cuadrada de los coeficientes a la forma triangular superior. En una matriz triangular, el determinante es el producto de los coeficientes de la diagonal principal. Por lo tanto, el determinante se puede calcular multiplicando los divisores colocados en la diagonal principal, considerando además el número de cambios de fila que se hayan realizado en la estrategia de pivoteo.

Sean

- **A:** matriz cuadrada
- **T:** Matriz triangular superior obtenida con el algoritmo de Gauss
- $\mathbf{a_{i,i}}$ **:** Elementos en la diagonal de la matriz **T**. Son los divisores
- **k:** Número de cambios de fila realizados
- **det(A):** Determinante de la matriz **A**

Entonces

$$\mathbf{det(A)} = (-1)^k \prod_{i=1}^{n} a_{i,i}$$

4.3.8 Instrumentación computacional para calcular el determinante

```
function d=determinante(a)
%Determinante de una matriz cuadrada
%Mediante reducción a una matriz triangular
%El determinante es el producto de los pivotes
[n,m]=size(a);
if n~=m                                  %La matriz debe ser cuadrada
   d=0;
   return
end
signo=1;                                 %cambios de fila
d=1;
for e=1:n
  [z, p]=max(abs(a(e:n,e)));             %Pivoteo por filas
  p=p+e-1;
  t=a(e,e:n);                            %Intercambio de filas
  a(e,e:n)=a(p,e:n);
  a(p,e:n)=t;
  if e~=p                                %Cambio de fila = cambio de signo
     signo=signo*(-1);
  end
  if abs(a(e,e))<1.0e-10                 %Divisor cercano a 0: matriz singular
    d=0;
    return;
  end
  d=d*a(e,e);                            %Multiplicación de pivotes
  a(e,e:n)=a(e,e:n)/a(e,e);              %Normalizar la fila e
  for i=e+1:n                            %Reducir otras filas
      a(i,e:n)=a(i,e:n)-a(i,e)*a(e,e:n);
  end
end
d=d*signo;                               %Determinante
```

Ejemplo

```
>> a=[5 3 7; 2 9 8; 5 8 2]
a =
    5    3    7
    2    9    8
    5    8    2
>> d=determinante(a)
d =
  -325
```

4.4 Sistemas mal condicionados

Al resolver un sistema de ecuaciones lineales usando un método directo, es necesario analizar si el resultado calculado es confiable. En esta sección se estudia el caso especial de sistemas que son muy sensibles a los errores en los datos o en los cálculos y que al resolverlos producen resultados con mucha variabilidad.

Para describir estos sistemas se considera el siguiente ejemplo:

Ejemplo *Una empresa compra tres materiales A, B, C en cantidades en kg. como se indica en el cuadro. Se dispone de tres facturas en las que consta el total pagado en dólares.*

Factura	*A*	*B*	*C*	*Total*
1	*2.0*	*4.0*	*5.0*	*220*
2	*6.0*	*9.0*	*8.0*	*490*
3	*4.1*	*5.0*	*3.0*	*274*

Con esta información debe determinarse el precio por kg. de cada material.

Sean x_1, x_2, x_3 *los precios por kg. que deben determinarse. Entonces se pueden plantear las ecuaciones:*

$$2.0x_1 + 4.0x_2 + 5.0x_3 = 220$$
$$6.0x_1 + 9.0x_2 + 8.0x_3 = 490$$
$$4.1x_1 + 5.0x_2 + 3.0x_3 = 274$$

En notación matricial

$$\begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.1 & 5.0 & 3.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 220 \\ 490 \\ 274 \end{bmatrix}$$

Si se resuelve este sistema con un método directo se obtiene:

$$\mathbf{X} = \begin{bmatrix} 40.00 \\ 10.00 \\ 20.00 \end{bmatrix}$$

Supondremos ahora que el digitador se equivocó al ingresar los datos en la matriz y registró ***4.1*** *en lugar del valor correcto* ***4.2***

$$\begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.2 & 5.0 & 3.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 220 \\ 490 \\ 274 \end{bmatrix}$$

Si se resuelve este sistema nuevamente con un método directo se obtiene:

$$\mathbf{X} = \begin{bmatrix} 20.00 \\ 31.53 \\ 10.76 \end{bmatrix}$$

Un cambio menor en un coeficiente produjo un cambio muy significativo en la solución. El resultado fue afectado fuertemente por este error. Esto es un indicio de que el sistema es de un tipo especial denominado **mal condicionado**. Los resultados obtenidos con estos sistemas no son confiables para tomar decisiones.

Esta situación se origina en el hecho de que la tercera ecuación es “casi linealmente dependiente” de las otras dos ecuaciones, por lo tanto, la solución puede variar mucho al cambiar algunos coeficientes.

Es conveniente detectar si un sistema es mal condicionado. Se debe cambiar ligeramente el valor de algún coeficiente y observar el cambio en el vector solución. Si la solución cambia significativamente, entonces es un sistema mal condicionado y debe revisarse la elaboración del modelo matemático.

En esta sección se establece una medida para cuantificar el nivel de mal condicionamiento de un sistema de ecuaciones lineales.

4.4.1 Definiciones

La norma de un vector o de una matriz es una manera de expresar la magnitud de sus componentes

Sean **X**: vector de n componentes
A: matriz de nxn componentes

Algunas definiciones comunes para la norma:

$$\|\mathbf{X}\| = \sum_{i=1}^{n} |x_i|$$

$$\|\mathbf{X}\| = \max |x_i|, i = 1,2,...,n$$

$$\|\mathbf{X}\| = \left(\sum_{i=1}^{n} x^2\right)^{1/2}$$

$$\|\mathbf{A}\| = \max \sum_{i=1}^{n} |a_{i,j}|, j = 1,2,...,n$$

$$\|\mathbf{A}\| = \max \sum_{j=1}^{n} |a_{i,j}|, i = 1,2,...,n$$

$$\|\mathbf{A}\| = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j}^2\right)^{1/2}$$

Las dos primeras se denominan **norma 1** y **norma infinito**, tanto para vectores como para matrices. La tercera es la norma euclideana.

Ejemplo. *Dada la siguiente matriz*

$$\mathbf{A} = \begin{bmatrix} 5 & -3 & 2 \\ 4 & 8 & -4 \\ 2 & 6 & 1 \end{bmatrix}$$

Calcule la norma infinito (norma por fila).

Esta norma es el mayor valor de la suma de las magnitudes de los componentes de cada fila

Fila 1: |5| + |-3| + |2| = 10
Fila 2: |4| + |8| + |-4| = 16
Fila 3: |2| + |6| + |1| = 9

Por lo tanto, la norma por fila de la matriz es 16

4.4.2 Algunas propiedades de normas

Sea **A**: matriz de nxn componentes. (También se aplican a vectores)

a) $\|\mathbf{A}\| \geq \mathbf{0}$

b) $\|\mathbf{kA}\| = |\mathbf{k}| \|\mathbf{A}\|$, $\mathbf{k} \in \Re$

c) $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$

d) $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$

e) $\|(\mathbf{kA})^{-1}\| = \|1/\mathbf{k}\ \mathbf{A}^{-1}\|$, $\mathbf{k} \in \Re$, $\mathbf{k} \neq 0$

4.4.3 Número de condición

El número de condición de una matriz se usa para cuantificar su nivel de mal condicionamiento.

Definición: Número de condición

> Sea **AX = B** un sistema de ecuaciones lineales, entonces
> **cond(A) = || A || || A^{-1} ||** es el **número de condición** de la matriz **A**.

Cota para el número de condición:

$$\mathbf{cond(A) = \| A \| \| A^{-1} \| \geq \| A\ A^{-1} \| = \| I \| = 1 \Rightarrow cond(A) \geq 1}$$

El número de condición no cambia si la matriz es multiplicada por alguna constante:

$$\mathbf{cond(kA) = \|kA\| \|(kA)^{-1}\| = k \|A\| \|1/k\ A^{-1}\| = k \|A\| 1/k \|A^{-1}\| = \|A\| \|A^{-1}\| = cond(A)}$$

Ejemplo. $\mathbf{A} = \begin{bmatrix} 0.010 & 0.005 \\ 0.025 & 0.032 \end{bmatrix}$; $\mathbf{B} = 1000\mathbf{A} = \begin{bmatrix} 10 & 5 \\ 25 & 32 \end{bmatrix}$

	A	*B*
Determinante	*0.000195*	*195*
Norma$_1$ de la matriz	*0.0370*	*37*
Norma$_1$ de la inversa	*292.3077*	*0.2923*
Número de condición	***10.8154***	***10.8154***

Si la matriz tiene filas "casi linealmente dependientes", su determinante tomará un valor muy pequeño y su inversa tendrá valores muy grandes, siendo esto un indicio de que la matriz es mal condicionada o es "casi singular". Este valor interviene en el número de condición de la matriz.

Por otra parte, si la matriz tiene valores muy pequeños, su determinante será muy pequeño y su inversa contendrá valores grandes aunque la matriz no sea mal condicionada.

Si el número de condición solo dependiera de la norma de la matriz inversa, esta norma tendría un valor grande en ambos casos. Por esto, y usando la propiedad anotada anteriormente, es necesario multiplicar la norma de la matriz inversa por la norma de la matriz original para que el número de condición sea grande únicamente si la matriz es mal condicionada.

***Ejemplo**. Calcule el número de condición de la matriz del ejemplo inicial*

$$\mathbf{A} = \begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.1 & 5.0 & 3.0 \end{bmatrix}$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 10.0000 & -10.0000 & 10.0000 \\ -11.3846 & 11.1538 & -10.7692 \\ 5.3077 & -4.9231 & 4.6154 \end{bmatrix}$$

cond(**A**) = $\| \mathbf{A} \| \| \mathbf{A}^{-1} \| = 766.07$

Es un valor alto, respecto al valor mínimo que es 1

Una matriz puede considerarse mal condicionada si una ligera perturbación, error o cambio, en la matriz de coeficientes produce un cambio muy significativo en el vector solución.

4.4.4 El número de condición y el error de redondeo

Dado un sistema de ecuaciones lineales $\mathbf{AX} = \mathbf{B}$ cuya solución existe y es $\mathbf{X}$

Suponer que debido a errores de medición, la matriz **A** de los coeficientes tiene un error **E**. Sea $\overline{\mathbf{A}} = \mathbf{A} + \mathbf{E}$, la matriz con los errores de medición. Suponer que el vector **B** es exacto

Entonces, al resolver el sistema con la matriz $\overline{\mathbf{A}}$ se tendrá una solución $\overline{\mathbf{X}}$ diferente a la solución **X** del sistema con la matriz **A**. Esta solución $\overline{\mathbf{X}}$ satisface al sistema: $\overline{\mathbf{A}}\ \overline{\mathbf{X}} = \mathbf{B}$

Es importante determinar la magnitud de la diferencia entre ambas soluciones: $\mathbf{X} - \overline{\mathbf{X}}$

Sustituyendo $\overline{\mathbf{A}}\ \overline{\mathbf{X}} = \mathbf{B}$ en el sistema original $\mathbf{AX} = \mathbf{B}$:

$$\begin{aligned} \mathbf{X} &= \mathbf{A}^{-1}\mathbf{B} \\ &= \mathbf{A}^{-1}(\overline{\mathbf{A}}\overline{\mathbf{X}}) \\ &= \mathbf{A}^{-1}(\mathbf{A} + \mathbf{E})\overline{\mathbf{X}} \\ &= \mathbf{A}^{-1}\mathbf{A}\ \overline{\mathbf{X}} + \mathbf{A}^{-1}\mathbf{E}\overline{\mathbf{X}} \\ &= \mathbf{I}\ \overline{\mathbf{X}} + \mathbf{A}^{-1}\mathbf{E}\overline{\mathbf{X}} \\ &= \overline{\mathbf{X}} + \mathbf{A}^{-1}\mathbf{E}\overline{\mathbf{X}} \end{aligned}$$

$$\Rightarrow \quad \mathbf{X} - \overline{\mathbf{X}} = \mathbf{A}^{-1}\mathbf{E}\overline{\mathbf{X}} \quad \Rightarrow \quad \left\| \mathbf{X} - \overline{\mathbf{X}} \right\| \le \left\| \mathbf{A}^{-1} \right\| \left\| \mathbf{E} \right\| \left\| \overline{\mathbf{X}} \right\| \quad \Rightarrow \quad \left\| \mathbf{X} - \overline{\mathbf{X}} \right\| \le \left\| \mathbf{A}^{-1} \right\| \left\| \mathbf{A} \right\| \frac{\left\| \mathbf{E} \right\|}{\left\| \mathbf{A} \right\|} \left\| \overline{\mathbf{X}} \right\|$$

De donde se puede escribir, sustituyendo **E** y el número de condición de **A**:

Definición: Cota para el error relativo de la solución

$$\frac{\| \mathbf{X} - \overline{\mathbf{X}} \|}{\| \overline{\mathbf{X}} \|} \le \mathbf{cond(A)} \frac{\| \overline{\mathbf{A}} - \mathbf{A} \|}{\| \mathbf{A} \|}$$

$\mathbf{e_X} \le \mathbf{cond(A)}\,(\mathbf{e_A})$ Cota para el error relativo de la solución

X es el vector solución calculado con la matriz inicial **A**
$\overline{\mathbf{X}}$ es el vector solución calculado con la matriz modificada $\overline{\mathbf{A}}$
$\mathbf{E} = \overline{\mathbf{A}} - \mathbf{A}$ es la matriz con la variación de los datos de la matriz.

$\mathbf{e_X}$ es el error relativo de la solución
$\mathbf{e_A}$ es el error relativo de la matriz

La expresión establece que la magnitud del error relativo de la solución está relacionada con el error relativo de la matriz del sistema, ponderada por el número de condición. El número de condición es un factor que amplifica el error en la matriz **A** aumentando la dispersión y la incertidumbre de la solución calculada $\overline{\mathbf{X}}$

Ejemplo. *Encuentre una cota para el error en la solución del ejemplo inicial*

Matriz original $\mathbf{A} = \begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.1 & 5.0 & 3.0 \end{bmatrix}$

Matriz modificada $\overline{\mathbf{A}} = \begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.2 & 5.0 & 3.0 \end{bmatrix}$

Error en la matriz: $\mathbf{E_A} = \overline{\mathbf{A}} - \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.1 & 0 & 0 \end{bmatrix}$

Norma del error relativo de la matriz:

$$\mathbf{e_A} = \frac{\| \mathbf{E_A} \|}{\| \mathbf{A} \|} = \frac{0.1}{23} = 0.0043 = 0.43\%$$

Número de condición:

$$\mathbf{cond(A) = 766.07}$$

Cota para el error relativo de la solución:

$$\mathbf{e_X \leq cond(A)\ (e_A) = 766.07\ (0.0043) = 3.29 = 329\%}$$

*Indica que la magnitud del error relativo de la solución puede variar hasta en **329%**, por lo tanto no se puede confiar en ninguno de los dígitos de la respuesta calculada.*

Ejemplo. *Encuentre el error relativo de la solución en el ejemplo inicial y compare con el error relativo de la matriz de los coeficientes.*

Sistema original: $\begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.1 & 5.0 & 3.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 220 \\ 490 \\ 274 \end{bmatrix}$ *Solución:* $\mathbf{X} = \begin{bmatrix} 40.00 \\ 10.00 \\ 20.00 \end{bmatrix}$

Sistema modificado: $\begin{bmatrix} 2.0 & 4.0 & 5.0 \\ 6.0 & 9.0 & 8.0 \\ 4.2 & 5.0 & 3.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 220 \\ 490 \\ 274 \end{bmatrix}$ *Solución:* $\overline{\mathbf{X}} = \begin{bmatrix} 20.00 \\ 31.53 \\ 10.76 \end{bmatrix}$

Error en la solución: $\mathbf{E_X} = \overline{\mathbf{X}} - \mathbf{X} = \begin{bmatrix} 20.00 \\ 31.53 \\ 10.76 \end{bmatrix} - \begin{bmatrix} 40.00 \\ 10.00 \\ 20.00 \end{bmatrix} = \begin{bmatrix} -20.00 \\ 21.53 \\ -9.23 \end{bmatrix}$

Norma del error relativo de la solución:

$$\mathbf{e_X} = \frac{\| \mathbf{E_X} \|}{\| \overline{\mathbf{X}} \|} = \frac{21.53}{31.53} = 0.6828 = 68.28\%$$

Norma del error relativo de la matriz:

$$\mathbf{e_A} = \frac{\| \mathbf{E_A} \|}{\| \mathbf{A} \|} = \frac{0.1}{23} = 0.0043 = 0.43\%$$

La variación en el vector solución es muy superior a la variación de la matriz de coeficientes. Se concluye que es un sistema mal condicionado.

4.4.5 Funciones de MATLAB para normas y número de condición

Cálculo de normas de vectores y matrices en MATLAB

Sea **a** un vector o una matriz

norm(a, 1)	para obtener la norma 1 (norma de columna)
norm(a, inf)	para obtener la norma infinito (norma de fila)
cond(a, 1)	número de condición con la norma 1
cond(a, inf)	número de condición con la norma infinito

Ejemplo. *Calcule el número de condición de la matriz* $\mathbf{A} = \begin{bmatrix} 4 & 5 \\ 4.1 & 5 \end{bmatrix}$

Escribimos en la pantalla de comandos de MATLAB:

```
>> a=[4, 5; 4.1, 5];                    (Matriz)
>> norm(a,inf)                          (Norma de fila
    ans =
      9.5
>> inv(a)                               (Matriz inversa)
    ans =
     -10.0000   10.0000
       8.2000   -8.0000

>> cond(a,inf)                          (Número de condición)
    ans =
      182.0000                          (Matriz mal condicionada)
```

4.5 Sistemas singulares

En esta sección se describe un método directo para intentar resolver un sistema lineal propuesto de **n** ecuaciones con **m** variables, **n<m**. Estos sistemas también se obtienen como resultado de la reducción de un sistema dado originalmente con **m** ecuaciones y **m** variables, en los que algunas ecuaciones no son independientes. En ambos casos la matriz de coeficientes contendrá una o más filas nulas y por lo tanto **no tienen inversa** y se dice que la **matriz es singular**, en este caso también diremos que el **sistema es singular**. Estos sistemas no admiten una solución única.

Sin embargo, si el sistema es un modelo que representa algún problema de interés, es importante detectar si el sistema es incompatible para el cual no existe solución, o se trata de un sistema incompleto para el cual existe infinidad de soluciones. Más aún, es útil reducirlo a una forma en la cual se facilite determinar las variables libres, a las que se pueden asignar valores arbitrarios analizar las soluciones resultantes en términos de éstas variables y su relación con el problema.

Para facilitar el análisis de estos sistemas, es conveniente convertirlo en una forma más simple. La estrategia que usaremos es llevarlo a la forma de la matriz identidad hasta donde sea posible

4.5.1 Formulación matemática y algoritmo

Se desea resolver el sistema de **n** ecuaciones lineales con **m** variables, siendo **n≤m**

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_1 + \ldots + a_{1,m}x_m &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_1 + \ldots + a_{2,m}x_m &= b_2 \\ &\ldots \\ a_{n,1}x_1 + a_{n,2}x_1 + \ldots + a_{n,m}x_m &= b_n \end{aligned}$$

En donde

$a_{i,j} \in \Re$: Coeficientes

$b_i \in \Re$: Constantes

$x_i \in \Re$: Variables cuyo valor debe determinarse

En notación matricial:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,m} \\ \ldots & & & \ldots \\ a_{n,1} & a_{n,1} & \ldots & a_{n,m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}$$

Simbólicamente: **AX = B**, en donde

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,m} \\ \ldots & & & \ldots \\ a_{n,1} & a_{n,1} & \ldots & a_{n,m} \end{bmatrix}; \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}; \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_m \end{bmatrix}$$

Para unificar la descripción algorítmica, es conveniente aumentar la matriz **A** con el vector **B** pues en ambos deben realizarse simultáneamente las mismas operaciones:

$$A \mid B = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} & a_{1,m+1} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,m} & a_{2,m+1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,m} & a_{n,m+1} \end{bmatrix}$$

Siendo $a_{i,m+1} = b_i$, **i = 1, 2, 3, ..., n**

El procedimiento consiste en transformar la matriz aumentada, de manera similar al método de Gauss-Jordan. El objetivo es reducir la matriz aumentada a una forma escalonada con **1's** en la diagonal mediante intercambios de filas, hasta donde sea posible hacerlo.

Si **n<m** la matriz transformada finalmente tendrá la siguiente forma

$$\mathbf{A \mid B} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} & a_{1,m+1} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} & a_{2,m+1} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} & a_{n,m+1} \end{bmatrix} \rightarrow \dots \rightarrow \left[\begin{array}{ccccccc|c} 1 & 0 & \dots & 0 & a_{1,n+1} & \dots & a_{1,m} & a_{1,m+1} \\ 0 & 1 & \dots & 0 & a_{2,n+1} & \dots & a_{2,m} & a_{2,m+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{n,n+1} & \dots & a_{n,m} & a_{n,m+1} \end{array}\right]$$

En el sistema resultante, las variables $x_{n+1}, x_{n+2}, \dots, x_m$ se denominan **variables libres** y pueden tomar valores arbitrarios, normalmente asociados al problema que se analiza, mientras que las otras variables $x_1, x_2, \dots, x_n$ pueden tomar valores dependientes de las variables libres. Los valores $a_{i,j}$ que aparecen en la matriz reducida, son los valores resultantes de las transformaciones aplicadas. La forma final facilita asignar valores a estas variables.

En cada etapa, primero se hará que el elemento en la diagonal tome el valor 1. Luego se hará que los demás elementos de la columna correspondiente, tomen el valor 0. Realizando previamente intercambios de filas para colocar como divisor el elemento de mayor magnitud. Esta estrategia se denomina pivoteo parcial y se usa para determinar si el sistema es singular.

La formulación es similar al método de Gauss-Jordan descrito en una sección anterior, pero el algoritmo incluye el registro de variables libres.

ALGORITMO DE GAUSS-JORDAN PARA SISTEMAS SINGULARES

a: matriz aumentada del sistema de n ecuaciones lineales
v: vector de variables libres (inicialmente nulo)

```
Para  e = 1, 2, . . ., n
    Elegir el valor de mayor magnitud de la columna e  en las filas  e, e+1, ..., n
    Si este valor es cero
        agregar  e  al vector  v  de las variable libres
        avanzar a la siguiente etapa  e
    Sino   (continuar con la transformación matricial)
        Para j=e, e+1, ..., n+1
            a(e,j) ← a(e,j) / a(e,e)          Normalizar la fila  e  ( a(e,e) ≠ 0 )
        Fin
        Para i=1, 2, …, i -1, i +1,… n
            Para j=e, e+1, ..., n+1
                a(i,j) ← a(i,j) – a(i,e) a(e,j)          Reducir las otras filas
            Fin
        Fin
    Fin
Fin
Si el vector  v  no contiene variables libres
  Entregar en x el vector solución almacenado en la última columna de la matriz a
Sino
  Entregar un vector x nulo
  Entregar la matriz a reducida
Fin
```

Ejemplo. *Una empresa produce cuatro productos:* ***P1, P2, P3, P4*** *usando tres tipos de materiales* ***M1, M2, M3****. Cada Kg. de producto requiere la siguiente cantidad de cada material, en Kg.:*

	P1	***P2***	***P3***	***P4***
M1	*0.2*	*0.5*	*0.4*	*0.2*
M2	*0.3*	*0*	*0.5*	*0.6*
M3	*0.5*	*0.5*	*0.1*	*0.2*

La cantidad disponible de cada material es: ***10, 12, 15*** *Kg. respectivamente, los cuales deben usarse completamente. Se quiere analizar la estrategia de producción factible.*

Sean x_1, x_2, x_3, x_4 *cantidades en Kg. producidas de* ***P1, P2, P3, P4****, respectivamente*
Se obtienen las ecuaciones:

$$0.2x_1 + 0.5x_2 + 0.4x_3 + 0.2x_4 = 10$$
$$0.3x_1 + 0.5x_3 + 0.6x_4 = 12$$
$$0.5x_1 + 0.5x_2 + 0.1x_3 + 0.2x_4 = 15$$

Es un sistema de tres ecuaciones y cuatro variables. En notación matricial

$$\begin{bmatrix} 0.2 & 0.5 & 0.4 & 0.2 \\ 0.3 & 0 & 0.5 & 0.6 \\ 0.5 & 0.5 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 12 \\ 15 \end{bmatrix}$$

Reducción con el método de Gauss-Jordan usando pivoteo

$$\begin{bmatrix} 1.00 & 1.00 & 0.20 & 0.40 & 30.00 \\ 0 & -0.30 & 0.44 & 0.48 & 3.00 \\ 0 & 0.30 & 0.36 & 0.12 & 4.00 \end{bmatrix}$$

$$\begin{bmatrix} 1.00 & 0 & 1.66 & 2.00 & 40.00 \\ 0 & 1.00 & -1.46 & -1.60 & -10.00 \\ 0 & 0 & 0.80 & 0.60 & 7.00 \end{bmatrix}$$

$$\begin{bmatrix} 1.00 & 0 & 0 & 0.75 & 25.41 \\ 0 & 1.00 & 0 & -0.50 & 2.83 \\ 0 & 0 & 1.00 & 0.75 & 8.75 \end{bmatrix}$$

Sistema equivalente reducido:

$$x_1 + 0.75x_4 = 25.41$$
$$x_2 - 0.50x_4 = 2.83$$
$$x_3 + 0.75x_4 = 8.75$$

La variable x_4 *queda como variable libre:*

Sea $x_4 = t,\ t \geq 0,\ t \in \Re$ *(variable libre)*

Conjunto solución: $X = [25.41 - 0.75\,t,\ 2.83 + 0.50\,t,\ 8.75 - 0.75\,t,\ t]^T$

Rango para la variable libre:

$x_3 = 8.75 - 0.75\,t \geq 0 \Rightarrow t \leq 11.66$ *(la producción no puede tener valores negativos)*
$x_2 = 2.83 + 0.50\,t \geq 0 \Rightarrow t \geq 0$
$x_1 = 25.41 - 0.75\,t \geq 0 \Rightarrow t \leq 33.88$

Se concluye que $0 \le t \le 11.66$

Rango para las variables

$0 \le x_4 \le 11.66$

$x_3 = 8.75 - 0.75\,t \quad \Rightarrow 0 \le x_3 \le 8.75$

$x_2 = 2.83 + 0.50\,t \quad \Rightarrow 2.83 \le x_2 \le 8.66$

$x_1 = 25.41 - 0.75\,t \quad \Rightarrow 16.66 \le x_1 \le 25.41$

Esta información puede ser útil para decidir la cantidad que debe producirse de cada artículo usando todos los recursos disponibles.

Si se decide producir **10** *Kg del producto* P_4*, entonces para que no sobren materiales, la producción será:*

$x_4=10 \Rightarrow t = x_4,\ x_3=1.25,\ x_2=7.83,\ x_1=17.91$

Si se decide producir **20** *Kg del producto* P_1*, entonces para que no sobren materiales, la producción será:*

$x_1=20 \Rightarrow t=x_4=7.21,\ x_3= 3.34,\ x_2=6.43$

4.5.2 Instrumentación computacional

La instrumentación se realiza mediante una función con el nombre **slin** utilizando la notación compacta de matrices de MATLAB. En la codificación se incluyen algunos comentarios ilustrativos. La matriz es estandarizada dividiendo por el elemento de mayor magnitud para reducir el error de truncamiento y detectar en forma consistente si el sistema es singular.

Por los errores de redondeo se considerará que el elemento de la diagonal es nulo si su valor tiene una magnitud menor que 10^{-10}.

Parámetros de entrada
a: matriz de coeficientes
b: vector de constantes

Parámetros de salida
x: vector solución
a: matriz de coeficientes reducida a la forma escalonada

Puede llamarse a la función **slin** especificando únicamente un parámetro de salida, el vector solución:

```
>> x=slin(a,b)
```

Si se obtiene como respuesta un vector nulo

```
x =
   [ ]
```

Entonces se puede llamar usando los dos parámetros de salida para analizar el resultado de la transformación matricial, como se indicará en los ejemplos posteriores:

```
>> [x,c]=slin(a,b)
```

Si el vector solución no es un vector nulo, entonces el sistema es consistente y la solución obtenida es única y la matriz se habrá reducido a la matriz identidad.

Si el vector solución es un vector nulo, entonces el sistema es singular y la matriz de coeficientes reducida permitirá determinar si es un sistema redundante o incompatible.

```
function [x,a]=slin(a,b)
[n,m]=size(a);
z=max(max(a));
v=[n+1:m];                          %Vector inicial de variables libres
a(1:n,m+1)=b;                       %Matriz aumentada
if n>m                              %Mas ecuaciones que variables
   x=[ ];
   a=[ ];
   return;
end
a=a/z;                              %Estandarizar la matriz para reducir error
for e=1:n                           %Ciclo para n etapas
      [z,p]=max(abs(a(e:n,e)));     %Pivoteo por filas
      p=p+e-1;
      t=a(e,e:m+1);                 %Cambiar filas
      a(e,e:m+1)=a(p,e:m+1);
      a(p,e:m+1)=t;
      if abs(a(e,e))<1.0e-10        %Sistema singular
         v=[v, e];                  %Agregar variable libre y continuar
      else
         a(e,e:m+1)=a(e,e:m+1)/a(e,e);   %Normalizar la fila e
         for i=1:n                  %Reducir otras filas
            if i~=e
               a(i,e:m+1)=a(i,e:m+1)-a(i,e)*a(e,e:m+1);
            end
         end
      end
end
x=[ ];
if length(v)==0;                    %Sistema consistente. Solución única
   x=a(1:n,m+1);                    %El vector X es la última columna de A
   a(:,m+1)=[ ];                    %Eliminar la última columna de A
   return;
end
```

En los siguientes ejemplos se utiliza una **notación informal** para identificar cada tipo de sistema que se resuelve. Los resultados obtenidos deben interpretarse según lo descrito en la instrumentación computacional de la función **slin.**

Sistema completo:	Tiene **n** ecuaciones y **n** variables
Sistema incompleto:	Tiene **n** ecuaciones y **m** variables, **n<m**. Puede ser dado inicialmente o puede ser resultado del proceso de transformación matricial en el que algunas ecuaciones desaparecen pues son linealmente dependientes.
Sistema consistente:	Tiene una solución única
Sistema redundante:	Tiene variables libres y por lo tanto, infinidad de soluciones
Sistema incompatible:	Contiene ecuaciones incompatibles. La transformación matricial reduce el sistema a uno conteniendo proposiciones falsas

Las variables libres se reconocen porque no están diagonalizadas, es decir que no contienen 1 en la diagonal principal de la matriz transformada.

1) Sistema consistente

$$x_1 + 2x_3 + 4x_4 = 1$$
$$x_2 + 2x_3 = 0$$
$$x_1 + 2x_2 + x_3 = 0$$
$$x_1 + x_2 + 2x_4 = 2$$

```
>> a = [1 0 2 4; 0 1 2 0; 1 2 1 0; 1 1 0 2]
a =
    1    0    2    4
    0    1    2    0
    1    2    1    0
    1    1    0    2
>> b=[1; 0; 0; 2]
b =
    1
    0
    0
    2
>> x=slin(a,b)
x =
   -3.0000
    2.0000
   -1.0000
    1.5000
```

2) Sistema completo reducido a incompleto redundante

```
>> a=[1 1 2 2;1 2 2 4;2 4 2 4 ;1 3 0 2]
a =
    1    1    2    2
    1    2    2    4
    2    4    2    4
    1    3    0    2
>> b=[1; 2; 2; 1]
b =
    1
    2
    2
    1
>> x=slin(a,b)
x =
    []
>> [x,c]=slin(a,b)
x =
    []
c =
    1    0    0   -4   -2
    0    1    0    2    1
    0    0    1    2    1
    0    0    0    0    0
```

Se obtiene un sistema equivalente. Las soluciones se asignan mediante la variable libre x_4 *:*

$$x_1 - 4x_4 = -2 \qquad x_1 = 4x_4 - 2$$
$$x_2 - 2x_4 = 1 \quad \Rightarrow \quad x_2 = 2x_4 + 1$$
$$x_3 + 2x_4 = 1 \qquad x_3 = -2x_4 + 1$$

Sea $x_4 = t,\ t \in \mathbb{R}$*, entonces el conjunto solución es* $\{4t-2,\ 2t+1,\ -2t+1\}$

3) Sistema incompleto

```
>> a=[1 0 2 4;4 1 2 4;2 4 5 2]
a =
   1    0    2    4
   4    1    2    4
   2    4    5    2
>> b=[1; 0; 2]
b =
   1
   0
   2
>> x=slin(a,b)
x =
   [ ]

>> [x,c]=slin(a,b)

x =
   []
c =
   1.0000        0        0   0.6400   -0.2800
        0   1.0000        0  -1.9200   -0.1600
        0        0   1.0000   1.6800    0.6400
```

Se obtiene un sistema equivalente. Las soluciones se asignan mediante la variable libre x_4*:*

$$x_1 + 0.64x_4 = -0.28$$
$$x_2 - 1.92x_4 = -0.16$$
$$x_3 + 1.68x_4 = 0.64$$

4) Sistema completo reducido a incompleto incompatible

```
>> a=[1 1 2 2;1 2 2 4;2 4 2 4 ;1 3 0 2]
a =
   1    1    2    2
   1    2    2    4
   2    4    2    4
   1    3    0    2
>> b=[1; 2; 2; 4]
b =
   1
   2
   2
   4
>> x=slin(a,b)
x =
   []
>> [x,c]=slin(a,b)
x =
   []
c =
   1    0    0   -4   -2
   0    1    0    2    1
   0    0    1    2    1
   0    0    0    0    3
```

Sistema equivalente:

$$x_1 - 4x_4 = -2$$
$$x_2 - 2x_4 = 1$$
$$x_3 + 2x_4 = 1$$
$$0x_4 = 3$$

4.5.3 Uso de funciones de MATLAB

Comparación de la función **slin** con la función **rref** de **MATLAB** resolviendo un sistema singular.

Resolver el sistema incompleto:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 2 \\ 30 \\ -4 \\ 10 \\ 50 \end{bmatrix}$$

```
>> a=[1 0 -1 0 0 0; 0 0 0 1 0 -1; 0 0 1 0 -1 0;0 1 -1 0 0 0; 0 0 0 1 0 0];   (matriz 5x6)
>> b=[2; 30; -4; 10; 50];                                                    (vector 5x1)
>> [x,c]=slin(a,b)
x =
   []
c =
    1    0    0    0   -1    0   -2
    0    1    0    0   -1    0    6
    0    0    1    0   -1    0   -4
    0    0    0    1    0   -1   30
    0    0    0    0    0    1   20

>> c=rref([a, b])
c =
    1    0    0    0   -1    0   -2
    0    1    0    0   -1    0    6
    0    0    1    0   -1    0   -4
    0    0    0    1    0    0   50
    0    0    0    0    0    1   20
```

Los resultados son equivalentes. La variable libre es x_5

De ambos sistemas reducidos se puede obtener:

$$x_6 = 20$$
$$x_4 = 50$$

$$x_5 = t, \quad t \geq 0 \qquad \text{(variable libre)}$$
$$x_3 = -4 + t$$
$$x_2 = 6 + t$$
$$x_1 = -2 + t$$

Los resultados obtenidos también establecen que $t \geq 4$

4.6 Sistemas tridiagonales

En un sistema tridiagonal la matriz de los coeficientes contiene todos sus componentes iguales a cero excepto en las tres diagonales principales. Estos sistemas se presentan en la aplicación de cierto tipo de métodos numéricos como el caso de los trazadores cúbicos y en la solución de ecuaciones diferenciales con diferencias finitas.

Se puede diseñar un método directo para resolver un sistema tridiagonal con eficiencia de primer orden: **T(n)=O(n)** lo cual representa una enorme mejora respecto a los métodos directos generales para resolver sistemas de ecuaciones lineales, cuya eficiencia es **$T(n)=O(n^3)$**.

4.6.1 Formulación matemática y algoritmo

Un sistema tridiagonal de **n** ecuaciones expresado en notación matricial:

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \dots & \dots & \dots \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \dots \\ d_n \end{bmatrix}$$

Para obtener la formulación se puede considerar únicamente un sistema de tres ecuaciones para luego extenderla al caso general. Las transformaciones son aplicadas a la matriz aumentada:

$$\begin{bmatrix} b_1 & c_1 & & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ & a_3 & b_3 & d_3 \end{bmatrix}$$

1) Sea $w_1 = b_1$. Dividir la primera fila para w_1

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & \frac{d_1}{w_1} \\ a_2 & b_2 & c_2 & d_2 \\ & a_3 & b_3 & d_3 \\ & & & \end{bmatrix}$$

2) Sea $g_1 = \frac{d_1}{w_1}$. Restar de la segunda fila, la primera multiplicada por a_2

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & g_1 \\ 0 & b_2 - a_2\frac{c_1}{w_1} & c_2 & d_2 - a_2 g_1 \\ & a_3 & b_3 & d_3 \\ & & & \end{bmatrix}$$

3) Sea $w_2 = b_2 - a_2\frac{c_1}{w_1}$. Dividir la segunda fila para w_2

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & g_1 \\ 0 & 1 & \frac{c_2}{w_2} & \frac{d_2 - a_2 g_1}{w_2} \\ & a_3 & b_3 & d_3 \\ & & & \end{bmatrix}$$

4) Sea $g_2 = \dfrac{d_2 - a_2 g_1}{w_2}$. Restar de la tercera fila, la segunda multiplicada por a_3

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & g_1 \\ 0 & 1 & \frac{c_2}{w_2} & g_2 \\ & 0 & b_3 - a_3\frac{c_2}{w_2} & d_3 - a_3 g_2 \end{bmatrix}$$

5) Sea $w_3 = b_3 - a_3\dfrac{c_2}{w_2}$. Dividir la tercera fila para w_3

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & g_1 \\ 0 & 1 & \frac{c_2}{w_2} & g_2 \\ & 0 & 1 & \frac{d_3 - a_3 g_2}{w_3} \end{bmatrix}$$

6) Sea $g_3 = \dfrac{d_3 - a_3 g_2}{w_3}$. Finalmente se obtiene:

$$\begin{bmatrix} 1 & \frac{c_1}{w_1} & & g_1 \\ 0 & 1 & \frac{c_2}{w_2} & g_2 \\ & 0 & 1 & g_3 \end{bmatrix}$$

De donde se puede hallar directamente la solución

$$x_3 = g_3$$

$$x_2 = g_2 - \frac{c_2}{w_2} x_3$$

$$x_1 = g_1 - \frac{c_1}{w_1} x_2$$

La formulación se extiende al caso general [(2)]

Transformación matricial de un sistema tridiagonal de **n** ecuaciones lineales

$$w_1 = b_1$$

$$g_1 = \frac{d_1}{w_1}$$

$$w_i = b_i - \frac{a_i c_{i-1}}{w_{i-1}}, \qquad i = 2,3,\ldots,n$$

$$g_i = \frac{d_i - a_i g_{i-1}}{w_i}, \qquad i = 2,3,\ldots,n$$

Obtención de la solución

$$x_n = g_n$$

$$x_i = g_i - \frac{c_i x_{i+1}}{w_i}, \qquad i = n-1,\ n-2,\ldots,2,1$$

[(2)] **Algoritmo de Thomas**

El algoritmo incluye un ciclo dependiente del tamaño del problema **n** para reducir la matriz y otro ciclo externo dependiente de **n** para obtener la solución. Por lo tanto es un algoritmo con eficiencia **T(n)=O(n).**

4.6.2 Instrumentación computacional

Con la formulación anterior se escribe una función para resolver un sistema tridiagonal de **n** ecuaciones lineales. La función recibe los coeficientes y las constantes en los vectores **a, b, c, d**. La solución es entregada en el vector **x**

```
function x = tridiagonal(a, b, c, d)
n = length(d);
w(1) = b(1);
g(1) = d(1)/w(1);
for i = 2:n                                   %Transformación matricial
   w(i) = b(i) - a(i)*c(i-1)/w(i-1);
   g(i) = (d(i) - a(i)*g(i-1))/w(i);
end
x(n) = g(n);
for i = n-1:-1:1                              %Obtención de la solución
   x(i) = g(i) - c(i)*x(i+1)/w(i);
end
```

Ejemplo. *Resuelva el siguiente sistema tridiagonal de ecuaciones lineales usando la función anterior*

$$\begin{bmatrix} 7 & 5 & & \\ 2 & -8 & 1 & \\ & 6 & 4 & 3 \\ & & 9 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 7 \\ 8 \end{bmatrix}$$

```
>> a = [0  2  6  9];                    Vectores con las diagonales (completas)
>> b = [7  -8  4  8];
>> c = [5  1  3  0];
>> d = [6  5  7  8];
>> x = tridiagonal(a, b, c, d)
x =                                     Solución calculada
    0.7402
    0.1637
    4.8288
   -4.4324
```

5 Métodos iterativos para resolver sistemas de ecuaciones lineales

La resolución de sistemas de ecuaciones lineales también puede hacerse con fórmulas iterativas que permiten acercarse a la respuesta mediante aproximaciones sucesivas, sin embargo desde el punto de vista práctico es preferible usar métodos directos que con el soporte computacional actual resuelven grandes sistemas en forma eficiente y con mucha precisión, a diferencia de los sistemas de ecuaciones no-lineales cuya solución no se puede obtener mediante métodos directos.

Las fórmulas iterativas no siempre convergen, su análisis puede ser complicado, la convergencia es de primer orden y se requiere elegir algún vector inicial para comenzar el proceso iterativo. En la época actual el estudio de estos métodos iterativos se puede considerar principalmente como de interés teórico matemático, excepto para resolver grandes sistemas de ecuaciones lineales con matrices esparcidas y cuya convergencia se puede determinar fácilmente.

Para definir un método iterativo, se expresa el sistema **AX = B** en la forma **X = G(X)** con el mismo fundamento descrito en el método del Punto Fijo para resolver ecuaciones no lineales.

5.1 Método de Jacobi

5.1.1 Formulación matemática

Dado un sistema de ecuaciones lineales

$$a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1$$
$$a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = b_2$$
. . . .
$$a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n = b_n$$

Expresado abreviadamente en notación matricial: **AX = B**

Una forma simple para obtener la forma **X = G(X)** consiste en re-escribir el sistema despejando de la ecuación **i** la variable x_i a condición de que $a_{i,i}$ sea diferente de **0:**

$$x_1 = 1/a_{1,1}\,(b_1 - a_{1,2}x_2 - a_{1,3}x_3 - \ldots - a_{1,n}x_n)$$
$$x_2 = 1/a_{2,2}\,(b_2 - a_{2,1}x_1 - a_{2,3}x_3 - \ldots - a_{2,n}x_n)$$
. . . .
$$x_n = 1/a_{n,n}\,(b_n - a_{n,1}x_1 - a_{n,2}x_2 - \ldots - a_{n,n-1}x_{n-1})$$

El cual puede escribirse con la notación de sumatoria:

$$x_i = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j - \sum_{j=i+1}^{n} a_{i,j}x_j\right) = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{i,j}x_j\right); \quad i = 1, 2, \ldots, n;$$

El sistema está en la forma **X = G(X)** la cual sugiere su uso iterativo.

Utilizamos un índice para indicar iteración:

$$X^{(k+1)} = G(X^{(k)}), \quad k=0, 1, 2, \ldots.$$ (iteraciones)

Fórmula iterativa de Jacobi:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{i,j}x_j^{(k)}\right); \quad i = 1, 2, \ldots, n; \quad k = 0, 1, 2, \ldots$$

$X^{(0)}$ es el vector inicial. A partir de este vector se obtienen sucesivamente los vectores $X^{(1)}, X^{(2)}, \ldots$ Si el método converge, entonces $X^{(k)}$ tiende a la solución **X** a medida que **k** crece:

$$X^{(k)} \underset{k\to\infty}{\rightarrow} X$$

Ejemplo. *Dadas las ecuaciones:*

$$5x_1 - 3x_2 + x_3 = 5$$
$$2x_1 + 4x_2 - x_3 = 6$$
$$2x_1 - 3x_2 + 8x_3 = 4$$

Formule un sistema iterativo con el método de Jacobi:

$$x_1^{(k+1)} = 1/5\,(5 + 3x_2^{(k)} - x_3^{(k)})$$
$$x_2^{(k+1)} = 1/4\,(6 - 2x_1^{(k)} + x_3^{(k)})$$
$$x_3^{(k+1)} = 1/8\,(4 - 2x_1^{(k)} + 3x_2^{(k)}) \qquad k = 0, 1, 2, \ldots$$

Realice dos iteraciones, comenzando con los valores iniciales: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 1$

k=0:

$$x_1^{(1)} = 1/5\,(5 + 3x_2^{(0)} - x_3^{(0)}) = 1/5\,(5 + 3(1) - (1)) = 1/5\,(7) = 1.4$$
$$x_2^{(1)} = 1/4\,(6 - 2x_1^{(0)} + x_3^{(0)}) = 1/4\,(6 - 2(1) + (1)) = 1/4\,(5) = 1.25$$
$$x_3^{(1)} = 1/8\,(4 - 2x_1^{(0)} + 3x_2^{(0)}) = 1/8\,(4 - 2(1) + 3(1)) = 1/8\,(5) = 0.625$$

k=1:

$$x_1^{(2)} = 1/5\,(5 + 3x_2^{(1)} - x_3^{(1)}) = 1/5\,(5 + 3(1.25) - (0.625)) = 1.6250$$
$$x_2^{(2)} = 1/4\,(6 - 2x_1^{(1)} + x_3^{(1)}) = 1/4\,(6 - 2(1.4) + (0.625)) = 0.9563$$
$$x_3^{(2)} = 1/8\,(4 - 2x_1^{(1)} + 3x_2^{(1)}) = 1/8\,(4 - 2(1.4) + 3(1.25)) = 0.6188$$

5.1.2 Manejo computacional de la fórmula de Jacobi

La siguiente función en MATLAB recibe un vector **X** y entrega un nuevo vector **X** calculado en cada iteración

```
function x = jacobi(a,b,x)
n=length(x);
t=x;                                   % t es asignado con el vector X ingresado
for i=1:n
    s=a(i,1:i-1)*t(1:i-1)+a(i,i+1:n)*t(i+1:n);
    x(i) = (b(i) - s)/a(i,i);
end
```

Uso de la función Jacobi para el ejemplo anterior:

```
>> a= [5, -3, 1; 2, 4, -1;2, -3, 8];
>> b= [5; 6; 4];
>> x= [1; 1; 1];                       Vector inicial
>> x=jacobi(a,b,x)                     Repita este comando y observe la convergencia
x =
   1.4000
   1.2500
   0.6250
>> x=jacobi(a,b,x)
x =
   1.6250
   0.9563
   0.6188
>> x=jacobi(a,b,x)
x =
   1.4500
   0.8422
   0.4523                              etc.
```

5.1.3 Algoritmo de Jacobi

a: matriz de coeficientes, **b**: vector de constantes del sistema de **n** ecuaciones lineales
e: estimación del error para la solución, **m**: máximo de iteraciones permitidas
x: vector inicial para la solución, **k** es el conteo de iteraciones
t ← x
Para **k = 1, 2, . . ., m**
  Calcular el vector **x** con la fórmula de Jacobi
  Si la norma del vector **x – t** es menor que **e**
    x es el vector solución con precisión **e**
    Terminar
  sino
    t ← x
  fin
fin
Al exceder el máximo de iteraciones entregar un vector **x** nulo

5.1.4 Instrumentación computacional del método de Jacobi

La siguiente función en MATLAB recibe la matriz de coeficientes **a** y el vector de constantes **b** de un sistema lineal. Adicionalmente recibe un vector inicial **x**, la estimación del error **e** y el máximo de iteraciones permitidas **m**. La función entrega el vector **x** calculado y el número de iteraciones realizadas **k**. Si el método no converge, **x** contendrá un vector nulo y el número de iteraciones **k** será igual al máximo **m**.

```
function [x,k] = jacobim(a,b,x,e,m)
n=length(x);
for k=1:m
   t=x;
   for i=1:n
      s=a(i,1:i-1)*t(1:i-1)+a(i,i+1:n)*t(i+1:n);
      x(i) = (b(i) - s)/a(i,i);
   end
   if norm((x-t),inf)<e
      return
   end
end
x=[ ];
k=m;
```

Ejemplo

*Use la función **jacobim** para encontrar el vector solución del ejemplo anterior con una precisión de **0.0001** Determine cuántas iteraciones se realizaron. Comenzar con el vector inicial **x=[1; 1; 1]***

```
>> a=[5, -3, 1; 2, 4, -1; 2, -3, 8];
>> b=[5; 6; 4];
>> x=[1; 1; 1];
>> [x,k]=jacobim(a,b,x,0.0001,20)
x =
   1.4432
   0.8973
   0.4757
k =
    12
```

5.1.5 Forma matricial del método de Jacobi

Dado el sistema de ecuaciones lineales

AX = B

La matriz **A** se re-escribe como la suma de tres matrices:

A = L + D + U

D es una matriz diagonal con elementos iguales a los elementos de la diagonal principal de **A**

L es una matriz triangular inferior con ceros en la diagonal principal y los otros elementos iguales a los elementos respectivos de la matriz **A**

U es una matriz triangular superior con ceros en la diagonal principal y los otros elementos iguales a los elementos respectivos de la matriz **A**

En forma desarrollada:

$$\mathbf{A}=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} = \mathbf{L}+\mathbf{D}+\mathbf{U} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{2,1} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & 0 \end{bmatrix} + \begin{bmatrix} a_{1,1} & 0 & 0 & 0 \\ 0 & a_{2,2} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & a_{n,n} \end{bmatrix} + \begin{bmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ 0 & 0 & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Sustituyendo en la ecuación:

(L + D + U)X = B

LX + DX + UX = B

$X = D^{-1}B - D^{-1}LX - D^{-1}UX$, Siempre que $\mathbf{D^{-1}}$ exista

Ecuación recurrente del método de Jacobi según el método del Punto Fijo **X = G(X)**

$X = D^{-1}B - D^{-1}(L + U)X$

En donde

$$\mathbf{D^{-1}} = \left[\frac{1}{a_{i,i}}\right]_{nxn}$$

Ecuación recurrente desarrollada

$$\begin{bmatrix} x_1 \\ x_2 \\ : \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1/a_{1,1} \\ b_2/a_{2,2} \\ : \\ b_n/a_{n,n} \end{bmatrix} - \begin{bmatrix} 0 & a_{1,2}/a_{1,1} & a_{1,3}/a_{1,1} & \cdots & a_{1,n}/a_{1,1} \\ a_{2,1}/a_{2,2} & 0 & a_{2,3}/a_{2,2} & \cdots & a_{2,n}/a_{2,2} \\ : & : & : & \cdots & : \\ a_{n,1}/a_{n,n} & a_{n,2}/a_{n,n} & a_{n,3}/a_{n,n} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix}$$

Fórmula recurrente iterativa:

$$\mathbf{X^{(k+1)} = D^{-1}B - D^{-1}(L+U)X^{(k)}}, \quad \mathbf{k = 0,1,2,\ldots} \qquad \text{(iteraciones)}$$

Fórmula iterativa con las matrices desarrolladas

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ : \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} b_1/a_{1,1} \\ b_2/a_{2,2} \\ : \\ b_n/a_{n,n} \end{bmatrix} - \begin{bmatrix} 0 & a_{1,2}/a_{1,1} & a_{1,3}/a_{1,1} & \cdots & a_{1,n}/a_{1,1} \\ a_{2,1}/a_{2,2} & 0 & a_{2,3}/a_{2,2} & \cdots & a_{2,n}/a_{2,2} \\ : & : & : & \cdots & : \\ a_{n,1}/a_{n,n} & a_{n,2}/a_{n,n} & a_{n,3}/a_{n,n} & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ : \\ x_n^{(k)} \end{bmatrix}$$

$\mathbf{X^{(0)}}$ es el vector inicial. A partir de este vector se obtienen sucesivamente los vectores $\mathbf{X^{(1)}}$, $\mathbf{X^{(2)}}$, ...

Si el método converge, entonces la sucesión { $\mathbf{X}^{(k)}$ }$_{k=0,1,2,..}$ tiende al vector solución **X**

5.1.6 Práctica computacional con la notación matricial

Resuelva el ejemplo anterior usando la ecuación de recurrencia matricial directamente desde la ventana de commandos de MATLAB

```
>> a=[5 -3 1; 2 4 -1; 2 -3 8]
a =
    5   -3    1
    2    4   -1
    2   -3    8
>> b=[5;6;4]
b =
    5
    6
    4
>> d=[5 0 0; 0 4 0; 0 0 8]                    Matriz diagonal
d =
    5    0    0
    0    4    0
    0    0    8
>> l=[0 0 0; 2 0 0; 2 -3 0]                   Matriz triangular inferior con ceros en la diagonal
l =
    0    0    0
    2    0    0
    2   -3    0
>> u=[0 -3 1; 0 0 -1; 0 0 0]                  Matriz triangular superior con ceros en la diagonal
u =
    0   -3    1
    0    0   -1
    0    0    0
>> x=[1;1;1]                                  Vector inicial
x =
    1
    1
    1
>> x=inv(d)*b-inv(d)*(l+u)*x                  Formula iterativa de Jacobi en forma matricial
x =
   1.4000
   1.2500
   0.6250
>> x=inv(d)*b-inv(d)*(l+u)*x
x =
   1.6250
   0.9562
   0.6187
>> x=inv(d)*b-inv(d)*(l+u)*x
x =
   1.4500
   0.8422
   0.4523
```

Etc

5.2 Método de Gauss-Seidel

Se diferencia del método anterior al usar los valores más recientes del vector **X**, es decir aquellos que ya están calculados, en lugar de los valores de la iteración anterior como en el método de Jacobi. Por este motivo, podemos suponer que el método de Gauss-Seidel en general converge o diverge más rápidamente que el método de Jacobi.

5.2.1 Formulación matemática

La fórmula de Gauss-Seidel se la obtiene directamente de la fórmula de Jacobi separando la sumatoria en dos partes: los componentes que aún no han sido calculados se los toma de la iteración anterior k, mientras que los que ya están calculados, se los toma de la iteración k+1:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k)}\right); \quad i = 1, 2, ..., n; \quad k = 0, 1, 2, ...$$

En general, el método de Gauss-Seidel requiere menos iteraciones que el método de Jacobi en caso de que converja

***Ejemplo**. Dadas las ecuaciones:*

$$5x_1 - 3x_2 + x_3 = 5$$
$$2x_1 + 4x_2 - x_3 = 6$$
$$2x_1 - 3x_2 + 8x_3 = 4$$

Formule un sistema iterativo con el método de Gauss-Seidel:

$$x_1 = 1/5\ (5 + 3x_2 - x_3)$$
$$x_2 = 1/4\ (6 - 2x_1 + x_3)$$
$$x_3 = 1/8\ (4 - 2x_1 + 3x_2)$$

Fórmula iterativa:

$$x_1^{(k+1)} = 1/5\ (5 + 3x_2^{(k)} - x_3^{(k)})$$
$$x_2^{(k+1)} = 1/4\ (6 - 2x_1^{(k+1)} + x_3^{(k)})$$
$$x_3^{(k+1)} = 1/8\ (4 - 2x_1^{(k+1)} + 3x_2^{(k+1)})$$

Comenzando con el vector inicial: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 1$, realice dos iteraciones:

k=0:

$$x_1^{(1)} = 1/5\ (5 + 3x_2^{(0)} - x_3^{(0)}) = 1/5\ (5 + 3(1) - (1)) = 1.4$$
$$x_2^{(1)} = 1/4\ (6 - 2x_1^{(1)} + x_3^{(0)}) = 1/4\ (6 - 2(1.4) + (1)) = 1.05$$
$$x_3^{(1)} = 1/8\ (4 - 2x_1^{(1)} + 3x_2^{(1)}) = 1/8\ (4 - 2(1.4) + 3(1.05)) = 0.5438$$

k=1:

$$x_1^{(2)} = 1/5\ (5 + 3x_2^{(1)} - x_3^{(1)}) = 1/5\ (5 + 3(1.05) - (0.5438)) = 1.5212$$
$$x_2^{(2)} = 1/4\ (6 - 2x_1^{(2)} + x_3^{(1)}) = 1/4\ (6 - 2(1.5212) + (0.5438)) = 0.8753$$
$$x_3^{(2)} = 1/8\ (4 - 2x_1^{(2)} + 3x_2^{(2)}) = 1/8\ (4 - 2(1.5212) + 3(0.8753)) = 0.4479$$

5.2.2 Manejo computacional de la fórmula de Gauss-Seidel

La siguiente función en MATLAB recibe un vector **X** y entrega un nuevo vector **X** calculado en cada iteración

```
function x = gaussseidel(a,b,x)
n=length(x);
for i=1:n
   s=a(i,1:i-1)*x(1:i-1)+a(i,i+1:n)*x(i+1:n);          %Usa el vector x actualizado
   x(i) = (b(i) - s)/a(i,i);
end
```

Resuelva el ejemplo anterior usando la función gaussseidel:

```
>> a = [5, -3, 1; 2, 4, -1;2, -3, 8];
>> b = [5; 6; 4];
>> x = [1; 1; 1];                    Vector inicial
>> x=gaussseidel(a,b,x)              Repetir este comando  y observar la convergencia
x =
   1.4000
   1.0500
   0.5438
>> x=gaussseidel(a,b,x)
x =
   1.5213
   0.8753
   0.4479
>> x=gaussseidel(a,b,x)
x =
   1.4356
   0.8942
   0.4764
>> x=gaussseidel(a,b,x)
x =
   1.4412
   0.8985
   0.4766                            etc.
```

En general, la convergencia es más rápida que con el método de Jacobi

5.2.3 Instrumentación computacional del método de Gauss-Seidel

Similar al método de Jacobi , conviene instrumentar el método incluyendo el control de iteraciones dentro de la función, suministrando los parámetros apropiados.

Los datos para la función son la matriz de coeficientes **a** y el vector de constantes **b** de un sistema lineal. Adicionalmente recibe un vector inicial **x**, el criterio de error **e** y el máximo de iteraciones permitidas **m**. La función entrega el vector **x** calculado y el número de iteraciones realizadas **k**. Si el método no converge, **x** contendrá un vector nulo y el número de iteraciones **k** será igual al máximo **m**.

```
function [x,k] = gaussseidelm(a,b,x,e,m)
n=length(x);
for k=1:m
   t=x;
   for i=1:n
      s=a(i,1:i-1)*x(1:i-1)+a(i,i+1:n)*x(i+1:n);
      x(i) = (b(i) - s)/a(i,i);
   end
   if norm((x-t),inf)<e
      return
   end
end
x=[ ];
k=m;
```

Ejemplo. *Use la función* ***gaussseidelm*** *para encontrar el vector solución del ejemplo anterior con una precisión de* ***0.0001*** *y determine cuántas iteraciones se realizaron si se comienza con el vector inicial* ***x=[1; 1; 1]***

```
>> a=[5, -3, 1; 2, 4, -1; 2, -3, 8];
>> b=[5; 6; 4];
>> x=[1; 1; 1];
>> [x,k]=gaussseidelm(a,b,x,0.0001,20)
x =
   1.4432
   0.8973
   0.4757
k =
    7
```

5.2.4 Forma matricial del método de Gauss-Seidel

Dado el sistema de ecuaciones lineales

AX = B

La matriz **A** se re-escribe como la suma de tres matrices:

A = L + D + U

D es una matriz diagonal con elementos iguales a los elementos de la diagonal principal de **A**

L es una matriz triangular inferior con ceros en la diagonal principal y los otros elementos iguales a los elementos respectivos de la matriz **A**

U es una matriz triangular superior con ceros en la diagonal principal y los otros elementos iguales a los elementos respectivos de la matriz **A**

Sustituyendo en la ecuación:

(L + D + U)X = B

LX + DX + UX = B

$X = D^{-1}B - D^{-1}LX - D^{-1}UX$, Siempre que D^{-1} exista

Ecuación recurrente con las matrices desarrolladas

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 / a_{1,1} \\ b_2 / a_{2,2} \\ \vdots \\ b_n / a_{n,n} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{2,1} / a_{2,2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{n,1} / a_{n,n} & a_{n,2} / a_{n,n} & a_{n,3} / a_{n,n} & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} 0 & a_{1,2} / a_{1,1} & a_{1,3} / a_{1,1} & \dots & a_{1,n} / a_{1,1} \\ 0 & 0 & a_{2,3} / a_{2,2} & \dots & a_{2,n} / a_{2,2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

El sistema está en la forma recurrente del punto fijo **X = G(X)** que sugiere su uso iterativo. En el método de Gauss-Seidel se utilizan los valores recientemente calculados del vector **X**.

Fórmula iterativa

$$X^{(k+1)} = D^{-1}B - D^{-1}LX^{(k+1)} - D^{-1}UX^{(k)}, \quad k = 0,1,2,\dots$$

$X^{(0)}$ es el vector inicial. A partir de este vector se obtienen sucesivamente los vectores $X^{(1)}, X^{(2)}, \dots$

Si el método converge, entonces la sucesión $\{ X^{(k)} \}_{k=0,1,2,..}$ tiende al vector solución **X**

Fórmula matricial iterativa del método de Gauss-Seidel:

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} b_1 / a_{1,1} \\ b_2 / a_{2,2} \\ \vdots \\ b_n / a_{n,n} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{2,1} / a_{2,2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{n,1} / a_{n,n} & a_{n,2} / a_{n,n} & a_{n,3} / a_{n,n} & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} - \begin{bmatrix} 0 & a_{1,2} / a_{1,1} & a_{1,3} / a_{1,1} & \dots & a_{1,n} / a_{1,1} \\ 0 & 0 & a_{2,3} / a_{2,2} & \dots & a_{2,n} / a_{2,2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}$$

5.3 Método de relajación

Es un dispositivo para acelerar la convergencia (o divergencia) de los métodos iterativos

5.3.1 Formulación matemática

Se la obtiene de la fórmula de Gauss-Seidel incluyendo un factor de convergencia

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i}^{n} a_{i,j} x_j^{(k)} \right); \quad i = 1, 2, ..., n; \quad k = 0, 1, 2, ...$$

$0 < \omega < 2$ es el factor de relajación

Si $\omega = 1$ la fórmula se reduce a la fórmula iterativa de Gauss-Seidel

***Ejemplo**. Dadas las ecuaciones:*

$$5x_1 - 3x_2 + x_3 = 5$$
$$2x_1 + 4x_2 - x_3 = 6$$
$$2x_1 - 3x_2 + 8x_3 = 4$$

Formule un sistema iterativo con el método de Relajación:

$$x_1^{(k+1)} = x_1^{(k)} + \omega/5 \, (5 - 5x_1^{(k)} + 3x_2^{(k)} - x_3^{(k)})$$
$$x_2^{(k+1)} = x_2^{(k)} + \omega/4 \, (6 - 2x_1^{(k+1)} - 4x_2^{(k)} - x_3^{(k)})$$
$$x_3^{(k+1)} = x_3^{(k)} + \omega/8 \, (4 - 2x_1^{(k+1)} + 3x_2^{(k+1)} - 8x_3^{(k)})$$

Realice una iteración con el vector inicial: $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 1$, *y con* $\omega = 1.1$:

k=0:

$x_1^{(1)} = x_1^{(0)} + 1.1/5\,(5 - 5x_1^{(0)} + 3x_2^{(0)} - x_3^{(0)}) = 1 + 1.1/5\,(5 - 5(1) + 3(1) - 1) = 1.4400;$

$x_2^{(1)} = x_2^{(0)} + 1.1/4\,(6 - 2x_1^{(1)} - 4x_2^{(0)} + x_3^{(0)}) = 1 + 1.1/4\,(6 - 2(1.6) - 4(1) + 1) = 1.0330$

$x_3^{(1)} = x_3^{(0)} + 1.1/8\,(4 - 2x_1^{(1)} + 3x_2^{(1)} - 8x_3^{(0)}) = 1 + 1.1/8\,(4 - 2(1.6) - 3(0.925) - 8(1)) = 0.4801$

5.3.2 Manejo computacional de la fórmula de relajación

La siguiente función en MATLAB recibe un vector **X** y el factor de relajación **w**. Entrega un nuevo vector **X** calculado en cada iteración

```
function x = relajacion(a,b,x,w)
n=length(x);
for i=1:n
    s=a(i,1:n)*x(1:n);                    %Usa el vector x actualizado
    x(i)=x(i)+w*(b(i)-s)/a(i,i);
end
```

Resuelva el ejemplo anterior usando la función **relajación** con **k=1.2**:

```
>> a = [5, -3, 1; 2, 4, -1;2, -3, 8];
>> b = [5; 6; 4];
>> x = [1; 1; 1];                     Vector inicial
>> x=relajación(a,b,x,1.2)            Repita este comando y observe la convergencia
x =
   1.4800
   1.0120
   0.4114
>> x=relajacion(a,b,x,1.2)
x =
   1.5339
   0.8007
   0.4179
```

5.3.3 Forma matricial del método de relajación

Dado el sistema de ecuaciones lineales

$$AX = B$$

La matriz **A** se re-escribe como la suma de las siguientes matrices:

$$A = L + D + S - D$$

D es una matriz diagonal con elementos iguales a los elementos de la diagonal principal de **A**

L es una matriz triangular inferior con ceros en la diagonal principal y los otros elementos iguales a los elementos respectivos de la matriz **A**

S es una matriz triangular superior con todos sus elementos iguales a los elementos respectivos de la matriz **A**

Sustituyendo en la ecuación:

$$(L + D + S - D)X = B$$

$$LX + DX + SX - D\,X = B$$

$X = D^{-1}B - D^{-1}LX - D^{-1}SX + D^{-1}DX,$ Siempre que D^{-1} exista

$$X = X + D^{-1}B - D^{-1}LX - D^{-1}SX$$

Ecuación recurrente con las matrices desarrolladas

$$\begin{bmatrix} x_1 \\ x_2 \\ : \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1/a_{1,1} \\ b_2/a_{2,2} \\ : \\ b_n/a_{n,n} \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{2,1}/a_{2,2} & 0 & 0 & \dots & 0 \\ : & : & : & \dots & : \\ a_{n,1}/a_{n,n} & a_{n,2}/a_{n,n} & a_{n,3}/a_{n,n} & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix} - \begin{bmatrix} 1 & a_{1,2}/a_{1,1} & a_{1,3}/a_{1,1} & \dots & a_{1,n}/a_{1,1} \\ 0 & 1 & a_{2,3}/a_{2,2} & \dots & a_{2,n}/a_{2,2} \\ : & : & : & \dots & : \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix}$$

El sistema está en la forma recurrente del punto fijo **X = G(X)** que sugiere su uso iterativo. En el método de Relajación se agrega un factor ω

Fórmula iterativa en forma matricial

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \omega(\mathbf{D}^{-1}\mathbf{B} - \mathbf{D}^{-1}\mathbf{L}\mathbf{X}^{(k+1)} - \mathbf{D}^{-1}\mathbf{S}\mathbf{X}^{(k)}), \quad k = 0,1,2,\dots$$

ω es el factor de relajación. Este factor modifica al residual tratando de reducirlo a cero con mayor rapidez que el método de Gauss-Seidel.

Si $\omega = 1$ la fórmula iterativa se reduce a la fórmula del método de Gauss-Seidel
Si $\omega \in (0,1)$ se denomina método de subrelajación.
Si $\omega \in (1,2)$ se denomina método de sobrerelajación.

$\mathbf{X}^{(0)}$ es el vector inicial. A partir de este vector se obtienen sucesivamente los vectores $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots$

5.4 Convergencia de los métodos iterativos para sistemas lineales

Dado el sistema de ecuaciones lineales

$$\mathbf{AX} = \mathbf{B}$$

Se puede re-escribir en un sistema equivalente recurrente como en el Punto Fijo: **X = G(X)**

$$\mathbf{X} = \mathbf{C} + \mathbf{TX}$$

En donde **C** es un vector y **T** se denomina matriz de transición

Forma iterativa de la ecuación de recurrencia:

$$\mathbf{X}^{(k+1)} = \mathbf{C} + \mathbf{T}\mathbf{X}^{(k)}, \quad k = 0,1,2,\dots$$

De la resta de las ecuaciones se obtiene

$$\mathbf{X} - \mathbf{X}^{(k+1)} = \mathbf{T}\,(\mathbf{X} - \mathbf{X}^{(k)})$$

Si se define el error de truncamiento vectorialmente entre dos iteraciones consecutivas

$$\mathbf{E}^{(k)} = \mathbf{X} - \mathbf{X}^{(k)}$$

$$\mathbf{E}^{(k+1)} = \mathbf{X} - \mathbf{X}^{(k+1)}$$

Sustituyendo en la ecuación anterior

$$\mathbf{E}^{(k+1)} = \mathbf{T}\,\mathbf{E}^{(k)}$$

Definición

Convergencia del error de truncamiento

$$\mathbf{E}^{(k+1)} = \mathbf{T}\,\mathbf{E}^{(k)}$$

En donde **T** es la matriz de transición

Definición

Si **A** es una matriz, entonces su radio espectral se define como

$$\rho(\mathbf{A}) = \max_{1 \le i \le n} \left\{ |\lambda_i| \mid \lambda_i \text{ es un valor característico de } \mathbf{A} \right\}$$

Teorema general de convergencia

La sucesión $\{ \mathbf{X}^{(k)} \}_{k=0,1,2,..}$ definida con la fórmula iterativa $\mathbf{X}^{(k+1)} = \mathbf{C} + \mathbf{T}\mathbf{X}^{(k)}, \quad k = 0,1,2,...$ converge con cualquier vector inicial $\mathbf{X}^{(0)} \in \mathbf{R}^n$ al vector solución $\mathbf{X}$ si y solo si $\rho(\mathbf{T}) < 1$

5.4.1 Matriz de transición para los métodos iterativos

Forma general recurrente de los métodos iterativos

$$\mathbf{X}^{(k+1)} = \mathbf{C} + \mathbf{T}\mathbf{X}^{(k)}, \quad k = 0,1,2,...$$

Para obtener **T** se usará la definición de convergencia con el error de truncamiento

$$\mathbf{E}^{(k+1)} = \mathbf{T}\,\mathbf{E}^{(k)}$$

Matriz de transición para el método de Jacobi

Sistema de ecuaciones lineales

$$\mathbf{AX} = \mathbf{B}$$

Ecuación recurrente equivalente sustituyendo **A = L + D + U**

$$\mathbf{X} = \mathbf{D}^{-1}\mathbf{B} - \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{X}, \qquad \text{Siempre que } \mathbf{D}^{-1} \text{ exista}$$

Ecuación recurrente iterativa del método de Jacobi

$$\mathbf{X}^{(k+1)} = \mathbf{D}^{-1}\mathbf{B} - \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{X}^{(k)}, \quad k = 0,1,2,...$$

Restar las ecuaciones para aplicar la definición de convergencia: $\mathbf{E}^{(k+1)} = \mathbf{T}\,\mathbf{E}^{(k)}$

$$\mathbf{X} - \mathbf{X}^{(k+1)} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})(\mathbf{X} - \mathbf{X}^{(k)})$$
$$\mathbf{E}^{(k+1)} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\mathbf{E}^{(k)}$$

Matriz de transición:

$$\mathbf{T} = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}) = -\begin{bmatrix} 0 & a_{1,2}/a_{1,1} & a_{1,3}/a_{1,1} & \cdots & a_{1,n}/a_{1,1} \\ a_{2,1}/a_{2,2} & 0 & a_{2,3}/a_{2,2} & \cdots & a_{2,n}/a_{2,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n,1}/a_{n,n} & a_{n,2}/a_{n,n} & a_{n,3}/a_{n,n} & \cdots & 0 \end{bmatrix}$$

En la matriz de transición del método de Jacobi se puede establecer una condición suficiente de convergencia. Más débil que la condición general de convergencia:

Si la matriz de coeficientes **A** es de tipo diagonal dominante, entonces el método de Jacobi converge a la solución **X** para cualquier vector inicia $\mathbf{X}^{(0)} \in \mathbf{R}^n$

Convergencia definida con el error de truncamiento

$$\mathbf{E}^{(k+1)} = \mathbf{T}\,\mathbf{E}^{(k)}$$

Su norma:

$$\| \mathbf{E}^{(k+1)} \| \le \| \mathbf{T} \| \, \| \mathbf{E}^{(k)} \|, \quad k = 0, 1, 2, ...$$

Esta relación define una condición suficiente para la convergencia del método de Jacobi mediante la norma de la matriz de transición: $\| T \|_\infty < 1$

Se conoce también la relación $\rho(A) \leq \| T \|_\infty$

La forma de la matriz **T** establece que si en cada fila de la matriz **A** la magnitud de cada elemento en la diagonal es mayor que la suma de la magnitud de los otros elementos de la fila respectiva, entonces $\| T \| < 1$ usando la norma de fila. Si la matriz **A** cumple esta propiedad se dice que es **"diagonal dominante"** y constituye una condición suficiente para la convergencia

$$\forall i \left(|a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|\right) \Rightarrow \| T \| < 1$$

Matriz de transición para el método de Gauss-Seidel

Sistema de ecuaciones lineales

$$AX = B$$

Ecuación recurrente equivalente sustituyendo $A = L + D + U$

$$X = D^{-1}B - D^{-1}LX - D^{-1}UX,$$ Siempre que D^{-1} exista

Ecuación recurrente iterativa del Método de Gauss-Seidel

$$X^{(k+1)} = D^{-1}B - D^{-1}LX^{(k+1)} - D^{-1}UX^{(k)}$$

Restar las ecuaciones para aplicar la definición de convergencia: $E^{(k+1)} = T E^{(k)}$

$$X - X^{(k+1)} = -D^{-1}L(X - X^{(k+1)}) - D^{-1}U(X - X^{(k)})$$

$$E^{(k+1)} = -D^{-1}LE^{(k+1)} - D^{-1}UE^{(k)}$$

$$E^{(k+1)}(I + D^{-1}L) = -D^{-1}UE^{(k)}$$

$$E^{(k+1)} = (I + D^{-1}L)^{-1}(-D^{-1}U)E^{(k)}$$

Matriz de transición:

$$T = (I + D^{-1}L)^{-1}(-D^{-1}U)$$

Matriz de transición para el método de Relajación

Sistema de ecuaciones lineales

$$AX = B$$

Ecuación recurrente equivalente sustituyendo $A = L + D + S - D$ incluyendo el factor ω

$$X = X + \omega D^{-1}B - \omega D^{-1}LX - \omega D^{-1}SX,$$ Siempre que D^{-1} exista

Ecuación recurrente iterativa del Método de Relajación

$$X^{(k+1)} = X^{(k)} + \omega D^{-1}B - \omega D^{-1}LX^{(k+1)} - \omega D^{-1}SX^{(k)}$$

Restar las ecuaciones para aplicar la definición de convergencia: $E^{(k+1)} = T E^{(k)}$

$$X - X^{(k+1)} = X - X^{(k)} - \omega D^{-1}L(X - X^{(k+1)}) - \omega D^{-1}S(X - X^{(k)})$$

$$E^{(k+1)} = E^{(k)} - \omega D^{-1}LE^{(k+1)} - \omega D^{-1}SE^{(k)}$$

$$E^{(k+1)}(I + \omega D^{-1}L) = E^{(k)}(I - \omega D^{-1}S)$$

$$E^{(k+1)} = (I + \omega D^{-1}L)^{-1}(I - \omega D^{-1}S)E^{(k)}$$

Matriz de transición:

$$T = (I + \omega D^{-1}L)^{-1}(I - \omega D^{-1}S)$$

5.5 Eficiencia de los métodos iterativos

La fórmula del error de truncamiento expresa que la convergencia de los métodos iterativos es de primer orden:

$$E^{(k+1)} = O(E^{(k)})$$

Cada iteración requiere multiplicar la matriz de transición por un vector, por lo tanto la cantidad de operaciones aritméticas realizadas T en cada iteración es de segundo orden: $T(n)=O(n^2)$

Si **k** representa la cantidad de iteraciones que se realizan hasta obtener la precisión requerida, entonces la eficiencia de cálculo de los métodos iterativos es: $T(n) = k\, O(n^2)$

5.6 Finalización de un proceso iterativo

Si la fórmula iterativa converge, se puede escribir:

$$X^{(k)} \to X, \text{ si } k \to \infty$$

$$X^{(k+1)} \to X, \text{ si } k \to \infty$$

$$\Rightarrow \quad \| X^{(k+1)} - X^{(k)} \| \to 0, \text{ si } k \to \infty$$

Entonces, si el método converge, se tendrá que para cualquier valor positivo ε arbitrariamente pequeño, en alguna iteración **k**:

$$\| X^{(k+1)} - X^{(k)} \| < \varepsilon$$

Se dice que el vector calculado tiene precisión ε

Para que el error se independiente de la magnitud de los resultados se puede usar la definición de error relativo:

$$\frac{\| X^{(k+1)} - X^{(k)} \|}{\| X^{(k+1)} \|} < \varepsilon,$$ en donde ε puede expresarse como porcentaje

5.7 Práctica computacional con los métodos iterativos

Ejemplo. *Dado el sistema de ecuaciones:*

$$9x_1 + 3x_2 + 7x_3 = 5$$
$$2x_1 + 5x_2 + 7x_3 = 6$$
$$6x_1 + 2x_2 + 8x_3 = 4$$

Determine la convergencia y resuelva con los métodos iterativos anteriores

```
>> a=[9 3 7; 2 5 7; 6 2 8]
a =                                          No es diagonal dominante
    9    3    7                              pero pudiera ser que converja
    2    5    7
    6    2    8
>> b=[5;6;4]
b =
    5
    6
    4
>> d=diag(diag(a))                           matriz diagonal
d =
    9    0    0
    0    5    0
    0    0    8
>> l=tril(a)-d                               matriz triangular inferior
l =
    0    0    0
    2    0    0
    6    2    0
>> u=triu(a)-d                               matriz triangular superior
u =
    0    3    7
    0    0    7
    0    0    0
>> t=-inv(d)*(l+u)                           Método de Jacobi
t =                                          Matriz de transición
         0   -0.3333   -0.7778
   -0.4000         0   -1.4000
   -0.7500   -0.2500         0
>> e=eig(t)                                  valores característicos
e =
  -1.1937
   0.5969 + 0.0458i
   0.5969 - 0.0458i
>> r=norm(e,inf)                             mayor valor característico
r =
    1.1937                                   No converge
>> x=[1;1;1]
x =
    1
    1
    1
>> x=jacobi(a,b,x)
x =
   -0.5556
   -0.6000
   -0.5000
```

```
>> t=inv((eye(3)+inv(d)*l))*(-inv(d)*u)        Método de Gauss-Seidel
t =                                            Matriz de transición
         0   -0.3333   -0.7778
         0    0.1333   -1.0889
         0    0.2167    0.8556
>> r=norm(eig(t),inf)
r =
    0.5916                                     Si converge
>> x=[1;1;1]
x =
     1
     1
     1
>> x=gaussseidel(a,b,x)
x =
   -0.5556
    0.0222
    0.9111
>> x=gaussseidel(a,b,x)
x =
   -0.1605
   -0.0114
    0.6232
>> x=gaussseidel(a,b,x)
x =
    0.0746
    0.2977
    0.3696

>> s=triu(a)                                   Método de Relajación con w=0.9
s =
     9     3     7
     0     5     7
     0     0     8
>> w=0.9;
>> t=inv(eye(3)+w*inv(d)*l)*(eye(3)-w*inv(d)*s)   Matriz de transición
t =
    0.1000   -0.3000   -0.7000
   -0.0360    0.2080   -1.0080
   -0.0594    0.1557    0.7993
>> r=norm(eig(t),inf)
r =
    0.6071                                     Si converge
>> x=[1;1;1]
x =
     1
     1
     1
>> x=relajacion(a,b,x,0.9)
x =
   -0.4000
    0.0640
    0.8056
```

```
>> x=relajacion(a,b,x,0.9)
x =
   -0.1231
    0.1157
    0.5876

>> w=1.1;                                   Método de relajación (w=1.1)
>> t=inv(eye(3)+w*inv(d)*l)*(eye(3)-w*inv(d)*s)
t =                                         Matriz de transición
   -0.1000   -0.3667   -0.8556
    0.0440    0.0613   -1.1636
    0.0704    0.2856    0.9258
>> r=norm(eig(t),inf)
r =
    0.6076                                  Si converge

>> w=1.5;                                   Método de relajación (w=1.5)
>> t=inv(eye(3)+k*inv(d)*l)*(eye(3)-k*inv(d)*s)
t =                                         Matriz de transición
   -0.5000   -0.5000   -1.1667
    0.3000   -0.2000   -1.4000
    0.4500    0.6375    1.3375
>> r=norm(eig(t),inf)
r=
    0.9204                                   Si converge

>> w=1.6;                                   Método de relajación (w=1.6)
>> t=inv(eye(3)+w*inv(d)*l)*(eye(3)-w*inv(d)*s)
t =                                         Matriz de transición
   -0.6000   -0.5333   -1.2444
    0.3840   -0.2587   -1.4436
    0.5664    0.7435    1.4708
>> r=norm(eig(t),inf)
r =
    1.0220                                  No converge
```

Se puede comprobar la convergencia o divergencia llamando repetidamente a la función ***relajación*** *escrita previamente en MATLAB. Se la puede modificar incluyendo una condición para que se repita hasta que el método converja y un conteo de iteraciones para comparar con diferentes casos, en forma similar a la función* ***gaussseidelm***

Ejemplo

La siguiente matriz es del tipo que aparece frecuentemente en Análisis Numérico

$$a = \begin{bmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{bmatrix}$$

Use el Teorema general de convergencia: $\rho(T) < 1$*, para determinar cuál es el método iterativo más favorable para realizar los cálculos con esta matriz.*

Los cálculos se realizan con MATLAB:

```
 >> a=[4 -1 -1 -1; -1 4 -1 -1;-1 -1 4 -1; -1 -1 -1 4]
a =
    4   -1   -1   -1
   -1    4   -1   -1
   -1   -1    4   -1
   -1   -1   -1    4

>> d=diag(diag(a));
>> l=tril(a)-d;
>> u=triu(a)-d;
>> s=triu(a);

>> t=-inv(d)*(l+u);                                  Método de Jacobi
>> rjacobi=norm(eig(t),inf)
rjacobi =
   0.7500

>> t=inv((eye(4)+inv(d)*l))*(-inv(d)*u);             Método de Gauss-Seidel
>> rgaussseidel=norm(eig(t),inf)
rgaussseidel =
   0.5699

>> w=0.9;
>> t=inv(eye(4)+w*inv(d)*l)*(eye(4)-w*inv(d)*s);     Método de Relajación
>> rrelajacion=norm(eig(t),inf)
rrelajacion =
   0.6438

>> w=1.1;
>> t=inv(eye(4)+w*inv(d)*l)*(eye(4)-w*inv(d)*s);
>> rrelajacion=norm(eig(t),inf)
rrelajacion =
   0.4754

>> w=1.2;
>> t=inv(eye(4)+w*inv(d)*l)*(eye(4)-w*inv(d)*s);
>> rrelajacion=norm(eig(t),inf)
rrelajacion =
   0.3312

>> w=1.3;
>> t=inv(eye(4)+w*inv(d)*l)*(eye(4)-w*inv(d)*s);
>> rrelajacion=norm(eig(t),inf)
rrelajacion =
   0.3740

>> w=1.4;
>> t=inv(eye(4)+w*inv(d)*l)*(eye(4)-w*inv(d)*s);
>> rrelajacion=norm(eig(t),inf)
rrelajacion =
   0.4682
```

*Estos resultados muestran que para esta matriz, los tres métodos convergerían. La convergencia será más rápida si se usa el **método de relajación** con **w = 1.2**. Se puede verificar realizando las iteraciones con las funciones respectivas escritas en MATLAB*

5.8 Ejercicios y problemas con sistemas de ecuaciones lineales

Métodos directos

1. Dado el sistema lineal de ecuaciones:

$$3x_1 - x_3 = 5$$
$$\alpha x_1 + 2x_2 - x_3 = 2$$
$$-x_1 + x_2 + (\alpha + 1)x_3 = 1$$

Indique para cuales valores de α el sistema tiene una solución

2. Dado el sistema $[a_{i,j}]\, x = [b_i]$, i, j = 1, 2, 3
Siendo $a_{i,j} = i/(i + j)$, $b_i = 2i$

a) Escriba el sistema de ecuaciones lineales correspondiente
b) Resuelva el sistema con el Método de Gauss-Jordan

3. Los puntos (x, y): (1, 3), (2,5), (4,2), (5, 4) pertenecen a la siguiente función:
$f(x) = a_1x^2 + a_2\, e^{0.1x} + a_3\, x + a_4$

a) Escriba el sistema de ecuaciones con los puntos dados,
b) Resuelva el sistema con el Método de Gauss usando la estrategia de pivoteo con 4 decimales

4. Demuestre mediante un conteo que la cantidad de multiplicaciones que requiere el método de directo de Gauss-Jordan para resolver un sistema de n ecuaciones lineales es $n^3/2 + O(n^2)$ y que para el Método de Gauss es $n^3/3 + O(n^2)$

Sistemas mal condicionados

1. Considere la matriz de los coeficientes del ejercicio 3 de la sección anterior
a) Use el método de Gauss-Jordan para encontrar la matriz inversa
b) Calcule el número de condición. ¿Es una matriz mal condicionada?

2. Dado el siguiente sistema de ecuaciones

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/4 & 1/5 & 1/6 \\ 1/7 & 1/8 & 1/9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

a) Resuelva el sistema usando el método de Gauss-Jordan. Simultáneamente encuentre la inversa de la matriz

b) Modifique la matriz de coeficientes sustituyendo el valor de elemento $\mathbf{a_{1,1}}$ con el valor 0.9
Resuelva nuevamente el sistema. Encuentre la variación en la solución calculada.
c) Obtenga el número de condición de la matriz original.
d) Suponga que el error en los coeficientes no excede a 0.01. Use la definición indicada para encontrar una cota para el error en la solución

3. Dado el siguiente sistema de ecuaciones

$$\begin{bmatrix} 2 & 5 & 4 \\ 3 & 9 & 8 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 35 \\ 65 \\ 17 \end{bmatrix}$$

a) Obtenga la solución con un método directo
b) En la matriz de coeficientes sustituya 5 por 5.1 y obtenga nuevamente la solución con un método directo y compare con la solución del sistema original
c) Encuentre el error relativo de la solución y compare con el error relativo de la matriz. Comente acerca del tipo de sistema

Sistemas singulares

1. Una empresa produce semanalmente cuatro productos: **A, B, C, D** usando tres tipos de materiales **M1, M2, M3**. Cada Kg. de producto requiere la siguiente cantidad de cada material, en Kg.:

	P1	**P2**	**P3**	**P4**
M1	0.1	0.3	0.6	0.4
M2	0.2	0.6	0.3	0.4
M3	0.7	0.1	0.1	0.2

La cantidad disponible semanal de cada material es: **100, 120, 150** Kg. respectivamente, los cuales **deben usarse completamente**. Se quiere analizar la estrategia de producción.

a) Formule un sistema de ecuaciones lineales siendo x_1, x_2, x_3, x_4 cantidades en Kg. producidas semanalmente de los productos **A, B, C, D** respectivamente
b) Obtenga una solución con la función **slin** y re-escriba el sistema de ecuaciones resultante.
c) Escriba el conjunto solución expresado mediante la variable libre.
d) Encuentre el rango factible para la variable libre
e) Encuentre el rango factible para las otras variables
f) Defina cuatro casos de producción eligiendo un valor para cada una de las cuatro variables y estableciendo el nivel de producción para las restantes variables.

2. Use la función **slin** para resolver el siguiente sistema. Identifique las variables libres. Escriba el conjunto solución en términos de la variable libre. Asigne un valor a la variable libre y determine el valor de cada una de las otras variables:

$$\begin{bmatrix} 8 & 2 & 9 & 7 & 6 & 7 \\ 5 & 5 & 3 & 2 & 2 & 4 \\ 1 & 3 & 2 & 6 & 4 & 2 \\ 9 & 9 & 1 & 3 & 3 & 1 \\ 6 & 8 & 5 & 8 & 6 & 6 \\ 2 & 9 & 9 & 9 & 1 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 4 \\ 6 \\ 5 \end{bmatrix}$$

Métodos iterativos

1. Determine si el sistema de ecuaciones lineales obtenido con la siguiente fórmula converge a la solución con el Método de Jacobi

$$3kX_{k-1} - 6kX_k + 2kX_{k+1} = 2k - 1$$
$$X_0 = -3, \quad X_n = 4, \quad k=1, 2, 3, ..., n-1, n>3$$

2. Sea el sistema lineal de ecuaciones: $\begin{cases} 2X_1 - X_3 = 1 \\ \beta X_1 + 2X_2 - X_3 = 2 \\ -X_1 + X_2 + \alpha X_3 = 1 \end{cases}$

a) ¿Para que valores de α y β se puede asegurar la convergencia con el método de Jacobi?

b) Realice 3 iteraciones del método de Jacobi, con $\boldsymbol{X^{(0)} = [1, 2, 3]^T}$ usando valores de α y β que aseguren la convergencia.

3. Dado el sistema siguiente. Reordene las ecuaciones tratando de acercarlo a la forma "diagonal dominante".

$$4x_1 + 2x_2 + 5x_3 = 18.00$$
$$2x_1 + 5x_2 + x_3 = 27.30$$
$$2x_1 + 4x_2 + 3x_3 = 16.20$$

a) Escriba la matriz de transición del método de Jacobi y determine si se cumple la condición suficiente de convergencia al ser de tipo "diagonal dominante".
b) Determine si la matriz de transición de Jacobi cumple la condición general de convergencia. Puede calcular los valores característicos con la función **eig** de MATLAB
c) Escriba la matriz de transición del método de Gauss-Seidel y verifique que cumple la condición general de convergencia. Puede calcular los valores característicos con la función **eig** de MATLAB
b) Use la función Gauss-Seidel para realizar 15 iteraciones. Comente los resultados obtenidos.

PROBLEMAS

En cada problema plantee un sistema de ecuaciones lineales para determinar los coeficientes y resuelva el sistema lineal con un método directo. Muestre las transformaciones matriciales.

1. Para estudiar las propiedades de tres de tipos de materiales: X, Y, Z se combinarán tres ingredientes: A, B, C. de los cuales se tienen 4.9, 6.0, 4.8 Kg. respectivamente.

Cada Kg. del material X usa 0.3, 0.2, 0.5 Kg de ingredientes A, B, C respectivamente
Cada Kg. del material Y usa 0.4, 0.5, 0.1 Kg de ingredientes A, B, C respectivamente
Cada Kg. del material Z usa 0.25, 0.4, 0.35 Kg de los ingredientes A, B, C respectivamente

Encuentre la cantidad en Kg que se obtendrá de cada tipo de material, estableciéndose como requisito que debe usarse toda la cantidad disponible de los tres ingredientes en las mezclas.

2. Un comerciante compra tres productos: A, B, C. Estos productos se venden por peso en Kg. pero en las facturas únicamente consta el total que debe pagar. El valor incluye el impuesto a las ventas y supondremos, por simplicidad que es 10%. El comerciante desea conocer el precio unitario de cada artículo, para lo cual dispone de tres facturas con los siguientes datos:

Factura	Kg. de A	Kg. de B	Kg. de C	Valor pagado
1	4	2	5	$19.80
2	2	5	8	$30.03
3	2	4	3	$17.82

3. La distribución de dinero a 16 comunidades se describe en el siguiente cuadro. No fue posible contactar a cinco comunidades X_1, X_2, X_3, X_4, X_5 por lo que se decidió asignar a ellas el promedio del valor asignado a las comunidades que están a su alrededor, por ejemplo, X_1 recibirá el promedio de 30 + X_2 + X_3 + 45 + 50.
Determine cuales son los valores que serán asignados a estas cinco comunidades.

50	X_1	30	40
45	X_3	X_2	25
60	X_4	X_5	10
55	20	15	35

4. Suponga que en el siguiente modelo **f(x)** describe la cantidad de personas que son infectadas por un virus, en donde **x** es tiempo en días:

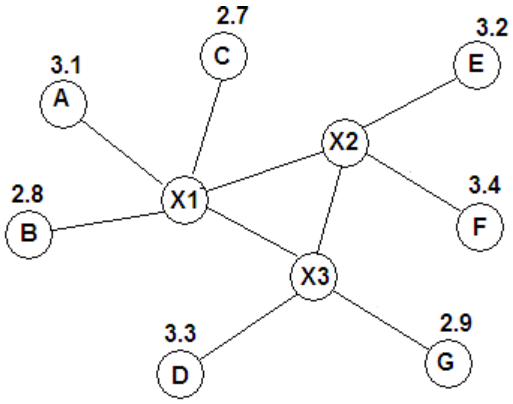
$$\mathbf{f(x) = k_1\, x + k_2\, x^2 + k_3\, e^{0.15X}}$$

En el modelo $\mathbf{k_1}$, $\mathbf{k_2}$ y $\mathbf{k_3}$ son coeficientes que deben determinarse.
Se conoce la cantidad de personas infectadas en los días 10, 15 y 20:

f(10)=25, f(15)=130, f(20)=650

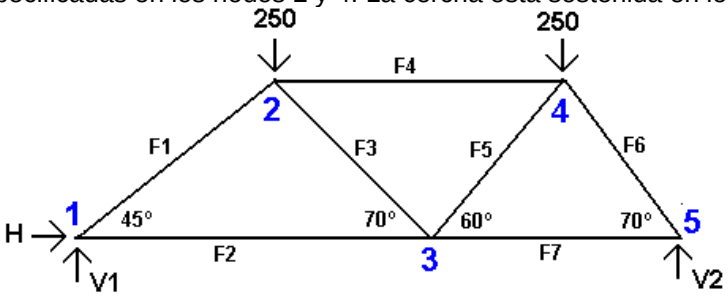
Plantee un sistema de ecuaciones lineales para determinar los coeficientes y use la solución para definir el modelo **f(x)** para determinar en cual día la cantidad de personas infectadas por el virus será 10000. Muestre el gráfico de la ecuación y los valores intermedios calculados.

5. En una región se desean instalar tres nuevos distribuidores X1, X2, X3 de un producto. En las cercanías ya existen otros distribuidores: A, B, C, D, E, F, G del mismo producto. En el gráfico, los círculos indican el precio de venta del producto en cada distribuidor. Las líneas indican con que otros distribuidores están directamente conectados. Determine el precio de venta que deben establecer los distribuidores x1, x2, x3, de tal manera que sean el promedio de los precios de los distribuidores con los que están directamente conectados.



6. Las cerchas son estructuras reticuladas con elementos triangulares, usualmente metálicas, que se utilizan para soportar grandes cargas. Es importante conocer las fuerzas que actúan en cada nodo. Para ello se plantean ecuaciones de equilibrio de fuerzas verticales y horizontales para cada nodo, las cuales conforman un sistema de ecuaciones lineales.

Determine las fuerzas que actúan en cada nodo en la siguiente cercha con las cargas verticales, en Kg, especificadas en los nodos 2 y 4. La cercha está sostenida en los nodos 1 y 5:



F1, F2, F3, F4, F5, F6, F7 son las fuerzas en los elementos que actúan para mantener la estructura unida, juntando los nodos y evitando que se separen. Sus valores son desconocidos. Se convendrá que si las fuerzas en cada nodo actúan hacia la derecha y hacia arriba tienen signo positivo. **H** es la fuerza horizontal y **V1, V2** son las fuerzas verticales que soportan la estructura. Sus valores son desconocidos.

Ecuaciones de equilibrio en cada nodo:

Nodo 1: $V1 + F1\ sen(45º) = 0$
$H + F1\ cos(45º) + F2 = 0$

Nodo 2: $-F1\ sen(45º) – F3\ sen(70º) – 250 = 0$
$-F1\ cos(45º) + F4 + F3\ cos(70º) = 0$

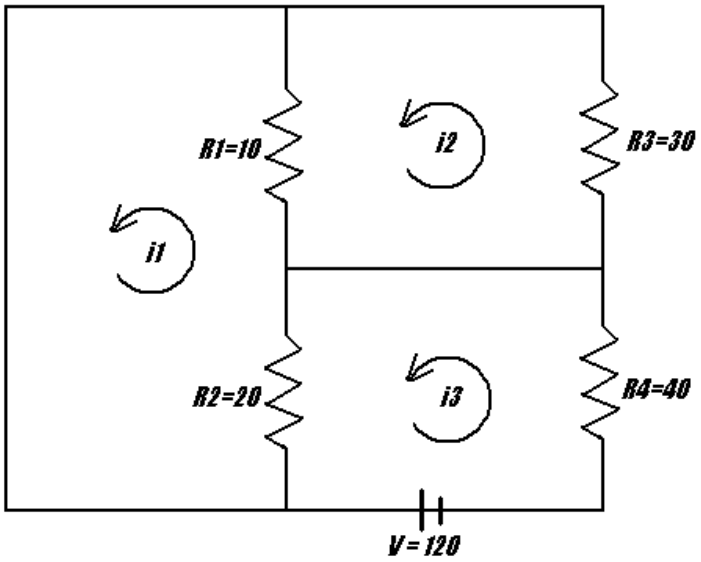
Nodo 3: $F3\ sen(70º) + F5\ sen(60º) = 0$
$-F2 – F3\ cos(70º) + F5\ cos(60º) + F7 = 0$

Nodo 4: $-F5\ sen(60º) – F6\ sen(70º) – 250 = 0$
$-F4 - F5\ cos(60º) + F6\ cos(70º) = 0$

Nodo 5: $V2 + F6\ sen(70º) = 0$
$-F7 – F6\ cos(70º) = 0$

Use una de las funciones instrumentadas en MATLAB para obtener la solución.

7. Se desea conocer la corriente i que fluye en un circuito que incluye una fuente de voltaje V y resistencias R como se indica en el gráfico:



El análisis se basa en leyes básicas de la física:
a) La suma de la caída de voltaje en un circuito cerrado es cero
b) El voltaje a través de una resistencia es el producto de su magnitud multiplicado por el valor de la corriente.

Para determinar el valor de la corriente en cada uno de los ciclos cerrados se conviene que la corriente es positiva si el sentido es opuesto al sentido del reloj.

Circuito cerrado a la izquierda: $\mathbf{10\ i_1 + 20\ i_1 - 10\ i_2 - 20\ i_3 = 0}$
Circuito cerrado arriba a la derecha: $\mathbf{30\ i_2 + 20\ i_2 - 10\ i_1 = 0}$
Circuito cerrado abajo a la derecha: $\mathbf{20\ i_3 + 40\ i_3 - 20\ i_2 = 120}$

Obtenga la solución con el método de Gauss

6 INTERPOLACIÓN

Para introducir este capítulo se analiza un problema para el cual se requiere como modelo un polinomio de interpolación.

Problema. *La inversión en la promoción para cierto producto en una región ha producido los siguientes resultados:*

(x, f): (5, 1.2), (10, 2.5), (15, 4.3), (20, 4.0)

*En donde **x:** tiempo invertido en la promoción del producto en horas*
***f:** porcentaje de incremento en las ventas del producto*

Se desea usar los datos para determinar la cantidad óptima de horas deberían invertirse en regiones similares para maximizar la eficiencia de las ventas.

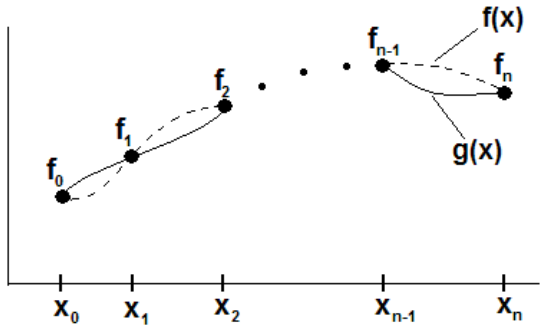
Análisis

Con los datos obtenidos se debe construir alguna función que permita obtener la respuesta y también estimar el error en el resultado.

6.1 El Polinomio de Interpolación

La interpolación es una técnica matemática que permite describir un conjunto de puntos mediante alguna función. Adicionalmente, tiene utilidad cuando se desea aproximar una función por otra función matemáticamente más simple.

Dados los puntos (x_i, f_i), $i = 0, 1, 2, \ldots, n$ que pertenecen a alguna función **f** que supondremos desconocida pero diferenciable, se debe encontrar alguna función **g** para aproximarla.



La función **g** debe incluir a los puntos dados: $g(x_i) = f_i$, $i = 0, 1, 2, \ldots, n$

Por simplicidad se elegirá como función **g**, un polinomio de grado no mayor a **n**:

$$g(x) = p_n(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n$$

Este polinomio **p** se denomina polinomio de interpolación

Primero se establecerán algunos fundamentos básicos relacionados con el polinomio de interpolación.

6.1.1 Existencia del polinomio de interpolación

El polinomio de interpolación **p** debe incluir a cada punto dado:

$$x=x_0: \quad p_n(x_0) = a_0x_0^n+a_1x_0^{n-1}+\ldots+a_{n-1}x_0+a_n = f_0$$
$$x=x_1: \quad p_n(x_1) = a_0x_1^n+a_1x_1^{n-1}+\ldots+a_{n-1}x_1+a_n = f_1$$
$$\ldots$$
$$x=x_n: \quad p_n(x_n) = a_0x_n^n+a_1x_n^{n-1}+\ldots+a_{n-1}x_n+a_n = f_n$$

Expresado en notación matricial

$$\begin{bmatrix} x_0^n & x_0^{n-1} & \ldots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_n^n & x_n^{n-1} & \ldots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \ldots \\ f_n \end{bmatrix}$$

La matriz de los coeficientes es muy conocida y tiene nombre propio: **Matriz de Vandermonde**. Su determinante se lo puede calcular con la siguiente fórmula. La demostración puede ser hecha con inducción matemática.

Sea $D = \begin{bmatrix} x_0^n & x_0^{n-1} & \ldots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \ldots & x_1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_n^n & x_n^{n-1} & \ldots & x_n & 1 \end{bmatrix}$ entonces $|D| = \prod_{j=0, j<i}^{n} (x_j - x_i)$

Ejemplo. *Calcule el determinante de la matriz de Vandermonde con* ***n=2***

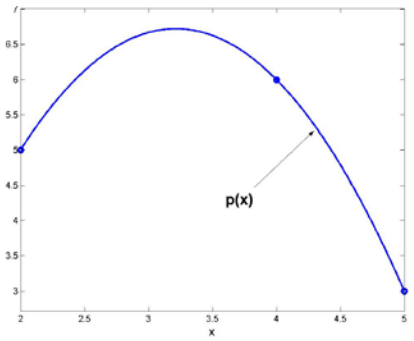
$$D = \begin{bmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{bmatrix} \Rightarrow |D| = \prod_{j=0, j<i}^{2} (x_j - x_i) = (x_0 - x_1)(x_0 - x_2)(x_1 - x_2)$$

De la definición anterior, se concluye que el determinante de esta matriz será diferente de cero si los valores de **X** dados no están repetidos. Por lo tanto, una condición necesaria para la existencia del polinomio de interpolación es que las abscisas de los datos dados sean diferentes entre si.

Ejemplo. *Dados los siguientes puntos:* ***(2, 5), (4, 6), (5, 3)***
a) Encuentre y grafique el polinomio de interpolación que los incluye.

Con la definición anterior y usando un método directo se obtiene:

$$\begin{bmatrix} 2^2 & 2 & 1 \\ 4^2 & 4 & 1 \\ 5^2 & 5 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -1.1666 \\ 7.5 \\ -5.3333 \end{bmatrix}$$



b) Calcule el número de condición de la matriz.

$$\mathbf{D}=\begin{bmatrix} 2^2 & 2 & 1 \\ 4^2 & 4 & 1 \\ 5^2 & 5 & 1 \end{bmatrix} \Rightarrow \mathbf{cond(D) = 341} \Rightarrow \mathbf{D} \text{ es una matriz mal condicionada}$$

En general, la matriz de Vandermonde es una matriz mal condicionada, por lo tanto la solución obtenida es muy sensible a los errores en los datos y los cálculos involucran el uso de algoritmos de tercer orden: $\mathbf{T(n)=O(n^3)}$

Existen métodos para encontrar el polinomio de interpolación que no utilizan la matriz anterior y tienen mejor eficiencia.

6.1.2 Unicidad del polinomio de interpolación con diferentes métodos

Dados los datos $\mathbf{(x_i, f_i), i = 0, 1, 2, . . ., n,}$ el polinomio de interpolación que incluye a todos los puntos es único.

Demostración
Suponer que usando los mismos datos y con métodos diferentes se han obtenido dos polinomios de interpolación: **p(x),** y **q(x)**. Ambos polinomios deben incluir a los puntos dados:

$$\mathbf{p(x_i) = f_i \text{ , } i=0, 1, 2, . . ., n}$$
$$\mathbf{q(x_i) = f_i \text{ , } i=0, 1, 2, . . ., n}$$

Sea la función **h(x) = p(x) - q(x).** Esta función también debe ser un polinomio y de grado no mayor a **n**. Al evaluar la función **h** en los puntos dados se tiene:

$$\mathbf{h(x_i) = p(x_i) - q(x_i) = f_i - f_i = 0, i=0, 1, 2, . . ., n}$$

Esto significaría que el polinomio **h** cuyo grado no es mayor a **n** tendría **n+1** raíces, contradiciendo el teorema fundamental del álgebra. La única posibilidad es que la función **h** sea la función cero, por lo tanto, $\mathbf{\forall x \in \Re[h(x)=0] \Rightarrow \forall x \in \Re[p(x)-q(x)=0] \Rightarrow \forall x \in \Re[p(x) = q(x)]}$

6.2 El polinomio de interpolación de Lagrange

Dados los datos (x_i, f_i), $i = 0, 1, 2, \ldots, n$. La siguiente fórmula permite obtener el polinomio de interpolación:

Definición: Polinomio de Interpolación de Lagrange

$$p_n(x) = \sum_{i=0}^{n} f_i \, L_i(x)$$

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}, \quad i = 0, 1, \ldots, n$$

Para probar que esta definición proporciona el polinomio de interpolación, la expresamos en forma desarrollada:

$$p_n(x) = f_0L_0(x) + f_1L_1(x) + \ldots + f_{i-1}L_{i-1}(x) + f_iL_i(x) + f_{i+1}L_{i+1}(x) + \ldots + f_nL_n(x) \qquad (1)$$

$$L_i(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}, \quad i=0,1,\ldots,n \qquad (2)$$

De la definición (2):

$L_i(x_i) = 1$, $i=0,1,\ldots,n$ (Por simplificación directa)

$L_i(x_k) = 0$, $k=0,1,\ldots,n$; $k \neq i$, (Contiene un factor nulo para algún $j = 0, 1, \ldots, n$; $j\neq i$)

Sustituyendo en la definición (1) de $p_n(x)$ se obtiene directamente:

$$p_n(x_i) = f_i \; ; \quad i = 0, 1, \ldots, n$$

Por otra parte, cada factor $L_i(x)$ es un polinomio de grado n, por lo tanto $p_n(x)$ también será un polinomio de grado n con lo cual la demostración está completa.

Ejemplo. *Dados los siguientes puntos:* ***(2, 5), (4, 6), (5, 3)***

Con la fórmula de Lagrange, encuentre el polinomio de interpolación que incluye a estos puntos.

$$p_2(x) = \sum_{i=0}^{2} f_iL_i(x) = f_0L_0(x) + f_1L_1(x) + f_2L_2(x) = 5L_0(x) + 6L_1(x) + 3L_2(x)$$

$$L_i(x) = \prod_{j=0, j\neq i}^{2} \frac{(x-x_j)}{(x_i-x_j)}, \quad i=0, 1, 2$$

$$L_0(x) = \prod_{j=0, j\neq 0}^{2} \frac{(x-x_j)}{(x_0-x_j)} = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-4)(x-5)}{(2-4)(2-5)} = \frac{x^2-9x+20}{6}$$

$$L_1(x) = \prod_{j=0, j\neq 1}^{2} \frac{(x-x_j)}{(x_1-x_j)} = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-2)(x-5)}{(4-2)(4-5)} = \frac{x^2-7x+10}{-2}$$

$$L_2(x) = \prod_{j=0, j\neq 2}^{2} \frac{(x-x_j)}{(x_2-x_j)} = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-2)(x-4)}{(5-2)(5-4)} = \frac{x^2-6x+8}{3}$$

Sustituir en el polinomio y simplificar:

$$p_2(x) = 5\left(\frac{x^2-9x+20}{6}\right) + 6\left(\frac{x^2-7x+10}{-2}\right) + 3\left(\frac{x^2-6x+8}{3}\right) = -\frac{7}{6}x^2 + \frac{15}{2}x - \frac{16}{3}$$

Se puede verificar que este polinomio incluye a los tres puntos dados.

Si únicamente se desea evaluar el polinomio de interpolación, entonces no es necesario obtener las expresiones algebraicas $L_i(x)$. Conviene sustituir desde el inicio el valor de **x** para obtener directamente el resultado numérico.

Ejemplo. *Dados los siguientes puntos:* ***(2, 5), (4, 6), (5, 3)***

Con la fórmula de Lagrange, evalúe en x = 3, el polinomio de interpolación que incluye a estos tres puntos dados.

$$p_2(3) = \sum_{i=0}^{2} f_i L_i(3) = f_0L_0(3) + f_1L_1(3) + f_2L_2(3) = 5L_0(3) + 6L_1(3) + 3L_2(3)$$

$$L_i(3) = \prod_{j=0, j\neq i}^{2} \frac{(3-x_j)}{(x_i-x_j)}, \quad i=0, 1, 2$$

$$L_0(3) = \prod_{j=0, j\neq 0}^{2} \frac{(3-x_j)}{(x_0-x_j)} = \frac{(3-x_1)(3-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(3-4)(3-5)}{(2-4)(2-5)} = 1/3$$

$$L_1(3) = \prod_{j=0, j\neq 1}^{2} \frac{(3-x_j)}{(x_1-x_j)} = \frac{(3-x_0)(3-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(3-2)(3-5)}{(4-2)(4-5)} = 1$$

$$L_2(3) = \prod_{j=0, j\neq 2}^{2} \frac{(3-x_j)}{(x_2-x_j)} = \frac{(3-x_0)(3-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(3-2)(3-4)}{(5-2)(5-4)} = -1/3$$

Finalmente se sustituyen los valores dados de ***f***

$$p_2(3) = 5(1/3) + 6(1) + 3(-1/3) = 20/3$$

6.2.1 Eficiencia del método de Lagrange

La fórmula de Lagrange involucra dos ciclos anidados que dependen del número de datos **n**:

$$p_n(x) = \sum_{i=0}^{n} f_i \, L_i(x)$$

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}, \quad i = 0, 1, \ldots, n$$

La evaluación de esta fórmula es un método directo que incluye un ciclo para sumar **n+1** términos. Cada factor $L_i(x)$ de esta suma requiere un ciclo con **2n** multiplicaciones, por lo tanto, la eficiencia de este algoritmo es $T(n) = 2n(n+1) = O(n^2)$, significativamente mejor que el método matricial que involucra la resolución de un sistema lineal cuya eficiencia es $T(n) = O(n^3)$.

6.2.2 Instrumentación computacional

La siguiente función recibe los puntos dados y entrega el polinomio de interpolación en forma algebraica usando la fórmula de Lagrange. Opcionalmente, si la función recibe como parámetro adicional el valor a interpolar, el resultado entregado es un valor numérico, resultado de la interpolación.

x, f: puntos base para la interpolación
v: valor para interpolar (parámetro opcional). Puede ser un vector

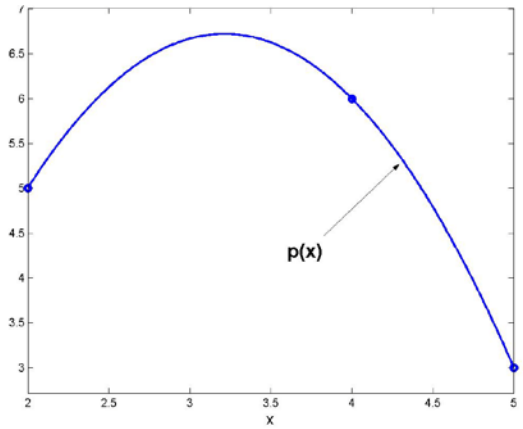
```
function p = lagrange(x, f, v)
n=length(x);
syms t;                                  %Variable para el polinomio
p=0;
for i=1:n
   L=1;
   for j=1:n
      if i ~= j
          L=L*(t-x(j))/(x(i)-x(j));
      end
   end
   p=p+L*f(i);                           %entrega p(t) en forma simbólica
end
p=expand(p);                             %simplificación algebraica
if nargin==3                             %verifica si existe un parámetro adicional
   t=v;
   p=eval(p);                            % entrega el resultado de p evaluado en v
end
```

Esta instrumentación es una oportunidad para explorar algunas de las características interesantes de MATLAB paral manejo matemático simbólico:

Ejemplo. *Use la función Lagrange para el ejemplo anterior:*

```
>> x = [2, 4, 5];                        Datos
>> f = [5, 6, 3];
>> p=lagrange(x, f)                      Obtención del polinomio de interpolación
 p =
   -7/6*t^2+15/2*t-16/3

>> r=lagrange(x, f, 4)                   Evaluar p en un punto dado
  r =
       6
>> r=lagrange(x, f, 4.25)                Evaluar p en un punto desconocido
  r =
     5.4687
>> plot(x, f, 'o'), grid on              Graficar los puntos
>> hold on, ezplot(p, [2, 5])            Graficar el polinomio sobre los puntos
```



Encuentre el valor de x para el cual p(x) = 4:

```
>> g=p-4                          (ecuación que debe resolverse: g(x) = p(x)-4 = 0)
g =
-7/6*t^2+15/2*t-28/3
>> s=eval(solve(g))               (obtener la solución con un método MATLAB)
s =
   4.7413
   1.6873
```

Encuentre el valor máximo de p(x):

```
>> p=lagrange(x,f)
p =
-7/6*t^2+15/2*t-16/3
>> g=diff(p)                      (ecuación que debe resolverse: g(x)=p'(x)=0)
g =
-7/3*t+15/2
>> t=eval(solve(g))               (obtener la solución con un método MATLAB)
t =
   3.2143
>> r=lagrange(x,f,t)
r =
   6.7202                         (coordenadas del máximo (t, r))
```

6.3 Interpolación múltiple

Se puede extender la interpolación a funciones de más variables. El procedimiento consiste en interpolar en una variable, fijando los valores de las otras variables y luego combinar los resultados. En esta sección se usará el polinomio de Lagrange en un ejemplo que contiene datos de una función que depende de dos variables. No es de interés encontrar la forma analítica del polinomio de interpolación que tendría términos con más de una variable.

Ejemplo. *Se tienen tabulados los siguientes datos* ***f(x,y)*** *de una función* ***f*** *que depende de las variables independientes* ***x***, ***y***. *Se deben usar* ***todos*** *los datos disponibles para estimar mediante interpolación polinomial el valor de* ***f(3,12)***

y / x	5	10	15	20
2	3.7	4.2	5.8	7.1
4	4.1	5.3	6.1	7.9
6	5.6	6.7	7.4	8.2

Primero interpolamos para **x=3** con los datos de cada columna **y = 5, 10, 15, 20**. Debe usarse un polinomio de segundo grado pues hay tres datos en la dirección **x**:

$$p_2(x) = \sum_{i=0}^{2} f_i L_i(x) = f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x)\ ;$$

$$L_i(x) = \prod_{j=0, j\neq i}^{2} \frac{(x-x_j)}{(x_i-x_j)}\ ,\quad i=0,\ 1,\ 2$$

No se requiere la forma algebraica. Se sustituye directamente el valor para interpolar **x=3.**

$$L_0(3) = \prod_{j=0, j\neq 0}^{2} \frac{(3-x_j)}{(x_0-x_j)} = \frac{(3-x_1)(3-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(3-4)(3-6)}{(2-4))(2-6)} = 3/8$$

$$L_1(3) = \prod_{j=0, j\neq 1}^{2} \frac{(3-x_j)}{(x_1-x_j)} = \frac{(3-x_0)(3-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(3-2)(3-6)}{(4-2))(4-6)} = 3/4$$

$$L_2(3) = \prod_{j=0, j\neq 2}^{2} \frac{(3-x_j)}{(x_1-x_j)} = \frac{(3-x_0)(3-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(3-2)(3-4)}{(6-2))(6-4)} = -1/8$$

Polinomio de interpolación para cada columna **y = 5, 10, 15, 20**:

$$p_2(3) = f_0 L_0(3) + f_1 L_1(3) + f_2 L_2(3) = f_0(3/8) + f_1(3/4) + f_2(-1/8)$$

Los valores de $L_i(3)$ son los mismos para cada columna **y**:

Se sustituyen los valores de cada columna:

y=5: $p_2(3) = 3.7(3/8) + 4.1(3/4) + 5.6(-1/8) = 3.7625$

y=10: $p_2(3) = 4.2(3/8) + 5.3(3/4) + 6.7(-1/8) = 4.7125$

y=15: $p_2(3) = 5.8(3/8) + 6.1(3/4) + 7.4(-1/8) = 5.8250$

y=20: $p_2(3) = 7.1(3/8) + 7.9(3/4) + 8.2(-1/8) = 7.5625$

Con los cuatro resultados se interpola en **y = 12** con un polinomio de tercer grado:

x \ y	5	10	15	20
3	3.7625	4.7125	5.8250	7.5625

$$p_3(y) = \sum_{i=0}^{3} f_i L_i(y) = f_0 L_0(y) + f_1 L_1(y) + f_2 L_2(y) + f_3 L_3(y);$$

$$L_i(y) = \prod_{j=0, j\neq i}^{3} \frac{(y-y_j)}{(y_i-y_j)}, \quad i=0, 1, 2, 3$$

Se sustituye directamente el valor para interpolar con la otra variable: **y = 12**

$$L_0(12) = \prod_{j=0, j\neq 0}^{3} \frac{(12-y_j)}{(y_0-y_j)} = \frac{(12-y_1)(12-y_2)(12-y_3)}{(y_0-y_1)(y_0-y_2)(y_0-y_3)} = \frac{(12-10)(12-15)(12-20)}{(5-10)(5-15)(5-20)} = -8/125$$

$$L_1(12) = \prod_{j=0, j\neq 1}^{3} \frac{(12-y_j)}{(y_1-y_j)} = \frac{(12-y_0)(12-y_2)(12-y_3)}{(y_1-y_0)(y_1-y_2)(y_1-y_3)} = \frac{(12-5)(12-15)(12-20)}{(10-5)(10-15)(10-20)} = 84/125$$

$$L_2(12) = \prod_{j=0, j\neq 2}^{3} \frac{(12-y_j)}{(y_2-y_j)} = \frac{(12-y_0)(12-y_1)(12-y_3)}{(y_2-y_0)(y_2-y_1)(y_0-y_3)} = \frac{(12-5)(12-10)(12-20)}{(15-5)(15-10)(15-20)} = 56/125$$

$$L_3(12) = \prod_{j=0, j\neq 3}^{3} \frac{(12-y_j)}{(y_2-y_j)} = \frac{(12-y_0)(12-y_1)(12-y_2)}{(y_3-y_0)(y_3-y_1)(y_3-y_2)} = \frac{(12-5)(12-10)(12-15)}{(20-5)(20-10)(20-15)} = -7/125$$

Resultado final:

y=12: $p_3(12) = f_0 L_0(12) + f_1 L_1(12) + f_2 L_2(12) + f_3 L_3(12)$

$= (3.7625)(-8/125) + (4.7125)(84/125) + (5.8250)(56/125) + (7.5625)(-7/125) = 5.1121$

$f(3, 12) \cong 5.1121$

6.3.1 Instrumentación computacional

Para interpolar en dos o más variable, se puede usar la función **lagrange** para interpolar en una variable. Al aplicarla en cada dirección se obtienen los resultados parciales. Interpolando con estos resultados producirá el resultado final.

Para el ejemplo anterior:

```
>> x = [2,  4,  6];                Interpolaciones parciales en x para cada columna de y
>> f = [3.7,  4.1,  5.6];
>> r1 = lagrange(x, f, 3);
>> f = [4.2,  5.3,  6.7];
>> r2 = lagrange(x, f, 3);
>> f = [5.8,  6.1,  7.4];
>> r3 = lagrange(x, f, 3);
>> f = [7.1,  7.9,  8.2];
>> r4 = lagrange(x, f, 3);

>> y = [5, 10, 15, 20];            Interpolación en y con los resultados parciales
>> f = [r1, r2, r3, r4];
>> p = lagrange(y, f, 12)
p =                                Resultado final
   5.1121
```

Si se planea usar este tipo de interpolación con frecuencia, conviene instrumentar una función en MATLAB para interpolar en dos dimensiones con la fórmula de Lagrange.

Esta función se llamará **lagrange2**, y usará internamente a la función **lagrange** para interpolar con una variable, y con los resultados realizar la interpolación con la otra variable.

Sean:
x,y: vectores con valores de las variables independientes **X, Y**
f: matriz con los datos de la variable dependiente organizados en filas
u,v: valores para los cuales se realizará la interpolación en **x** e **y** respectivamente

```
function p=lagrange2(x, y, f, u, v)
[n,m]=size(f);
for i=1:m
   r(i)=lagrange(x, f(:, i), u);        % cada columnas es enviada (resultados parciales)
end
p=lagrange(y, r, v);                    % interpolación final en la otra dirección
```

Ejemplo. *Use la función* ***lagrange2*** *para encontrar la respuesta en el ejemplo anterior*

```
>> x=[2,  4,  6];
>> y=[5,  10,  15,  20];
>> f=[3.7,  4.2,  5.8,  7.1;  4.1,  5.3,  6.1,  7.9;  5.6,  6.7,  7.4,  8.2];
>> p=lagrange2(x, y, f, 3, 12)
p =
   5.1121                               Resultado final
```

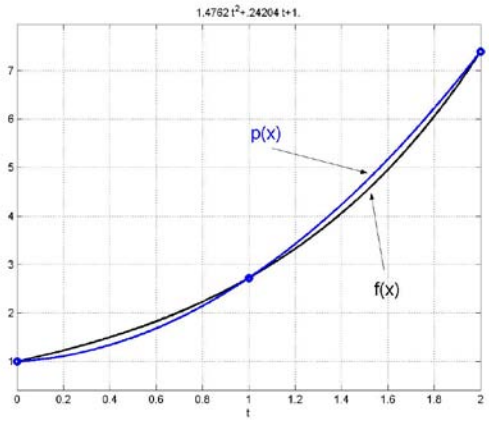
6.4 Error en la interpolación

Para entender este concepto usaremos un polinomio de interpolación para aproximar a una función conocida. Así puede determinarse en forma exacta el error en la interpolación.

Ejemplo. *Suponga que se desea aproximar la función* $f(x)=e^x$, $0 \le x \le 2$, *con un polinomio de segundo grado.*

Para obtener este polinomio tomamos tres puntos de la función **f:** $(0, e^0), (1, e^1), (2, e^2)$ y usamos la fórmula de Lagrange para obtener el polinomio de interpolación, con cinco decimales:

$$p_2(x) = 1.4762x^2 + 0.24204x+1$$



a) Encuentre el error en la aproximación cuando $x = 0.5$

Si se aproxima f(x) con $p_2(x)$ se introduce un error cuyo valor es $f(x) - p_2(x)$

$$f(0.5) - p_2(0.5) = e^{0.5} - 1.4762(0.5)^2 - 0.24204(0.5) - 1 = 0.1587$$

b) Encuentre el máximo error en la aproximación

Si se desea conocer cual es el máximo error en la aproximación, se debe resolver la ecuación

$$\frac{d}{dx}(f(x) - p_2(x)) = 0 \Rightarrow e^x - 2.9524x - 0.2420 = 0$$

Con un método numérico se encuentra que el máximo ocurre cuando $x = 1.6064$.

Entonces el máximo valor del error es: $f(1.6064) - p_2(1.6064) = -0.2133$

En general, dados los puntos (x_i, f_i), $i=0, 1, ..., n$, siendo **f** desconocida

Sea $p_n(x)$ el polinomio de interpolación, es decir el polinomio tal que $p_n(x_i) = f_i$, $i=0, 1, ..., n$

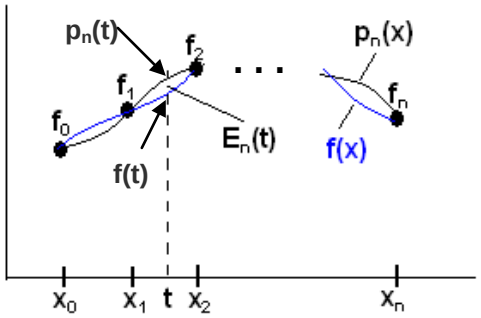
Suponer que se desea evaluar **f** en un punto **t** usando p_n como una aproximación:

$$f(t) \cong p_n(t), \quad t \neq x_i, \ i=0, 1, ..., n$$

Definición. Error en la interpolación

$$E_n(t) = f(t) - p_n(t)$$

Representación gráfica del error en la interpolación



Siendo **f** desconocida, no es posible conocer el error pues el valor exacto **f(t)** es desconocido. Únicamente se tiene el valor aproximado $\mathbf{p_n(t)}$. Pero es importante establecer al menos alguna expresión para estimar o acotar el valor del error $\mathbf{E_n(t)}$. En los puntos dados $\mathbf{E_n(x_i) = 0,\ i=0,1,\ ...,\ n}$

6.4.1 Una fórmula para estimar el error en la interpolación

En las aplicaciones comunes, únicamente se conocen puntos de la función **f**, siendo igualmente importante estimar la magnitud del error al usar el polinomio de interpolación. A continuación se desarrolla un procedimiento para estimar el error

Sean $g(x) = \prod_{i=0}^{n} (x - x_i) = (x-x_0)(x-x_1) \ldots (x-x_n)$, g es un polinomio de grado n+1

$h(x) = f(x) - p_n(x) - g(x)E_n(t)/g(t)$. h es una función con las siguientes propiedades

1) h es diferenciable si suponemos que **f** es diferenciable
2) $h(t) = 0$
3) $h(x_i) = 0,\ i = 0, 1, \ldots, n$

Por lo tanto, h es una función diferenciable y la ecuación

$h(x) = 0$ tiene n+2 ceros en el intervalo $[x_0, x_n]$

Aplicando sucesivamente el Teorema de Rolle:

$h^{(n+1)}(x) = 0$ tiene al menos 1 cero en el intervalo $[x_0, x_n]$

Sea $z \in [x_0, x_n]$ el valor de x tal que $h^{(n+1)}(z) = 0$

Derivando formalmente la función h:

$h^{(n+1)}(x) = f^{(n+1)}(x) - 0 - (n+1)!\ E_n(t)/g(t)$

Al evaluar esta función con x=z:

$h^{(n+1)}(z) = f^{(n+1)}(z) - 0 - (n+1)!\ E_n(t)/g(t) = 0$

Se obtiene finalmente:

$E_n(t) = g(t)\ f^{(n+1)}(z)/(n+1)!,\ \ t \neq x_i,\ \ z \in [x_0, x_n]$

Definición. Fórmula para estimar el error en el polinomio de interpolación

$$\mathbf{E_n(x) = g(x)\ f^{(n+1)}(z)/(n+1)!,\ \ x \neq x_i,\ \ z \in [x_0, x_n]}$$

$$\mathbf{Siendo\ g(x) = \prod_{i=0}^{n} (x - x_i) = (x-x_0)(x-x_1) \ldots (x-x_n)}$$

Para utilizar esta fórmula es necesario poder estimar el valor de $\mathbf{f^{(n+1)}(z)}$.

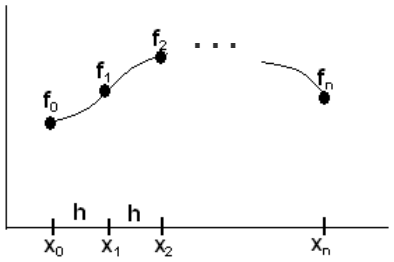
En el siguiente capítulo se introduce una técnica para estimar $\mathbf{f^{(n+1)}(z)}$ usando los puntos de **f**.

6.5 Diferencias finitas

Las siguientes definiciones establecen algunas relaciones simples entre los puntos dados. Estas definiciones tienen varias aplicaciones en análisis numérico.

Suponer que se tienen los puntos **(x_i, f_i), i=0, 1, ..., n,** tales que las abscisas están espaciadas regularmente en una distancia **h:**

$$x_{i+1} - x_i = h, \quad i=0, 1, ..., n-1$$



Definiciones

Primera diferencia finita avanzada: $\Delta^1 f_i = f_{i+1} - f_i$, $i=0,1,2, ...$

$\Delta^1 f_0 = f_1 - f_0$
$\Delta^1 f_1 = f_2 - f_1$, etc

Segunda diferencia finita avanzada: $\Delta^2 f_i = \Delta^1 f_{i+1} - \Delta^1 f_i$, $i=0,1,2, ...$

$\Delta^2 f_0 = \Delta^1 f_1 - \Delta^1 f_0$
$\Delta^2 f_1 = \Delta^1 f_2 - \Delta^1 f_1$, etc

Las diferencias finitas se pueden expresar con los puntos dados:

$$\Delta^2 f_0 = \Delta^1 f_1 - \Delta^1 f_0 = (f_2 - f_1) - (f_1 - f_0) = f_2 - 2f_1 + f_0$$

K-ésima diferencia finita avanzada:

$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$, $i=0,1,2, ...$, $k=1, 2, 3, ...$ con $\Delta^0 f_i = f_i$

Es útil tabular las diferencias finitas en un cuadro como se muestra a continuación:

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	x_0	f_0	$\Delta^1 f_0$	$\Delta^2 f_0$	$\Delta^3 f_0$	...
1	x_1	f_1	$\Delta^1 f_1$	$\Delta^2 f_1$	...	...
2	x_2	f_2	$\Delta^1 f_2$	...	...	...
3	x_3	f_3	...	...	...	...
...	...	...	...	...	...	...

Cada diferencia finita se obtiene restando los dos valores consecutivos de la columna anterior.

Ejemplo. *Tabule las diferencias finitas correspondientes a los siguientes datos*

(1.0, 5), (1.5, 7), (2.0, 10), (2.5, 8)

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$
0	1.0	5	2	1	-6
1	1.5	7	3	-5	
2	2.0	10	-2		
3	2.5	8			

Cálculo de diferencias finitas en MATLAB:

La función **diff** de MATLAB proporciona las diferencias finitas de orden sucesivo:

```
>> f=[5 7 10 8];
>> d1=diff(f)
d1 =
    2    3   -2
>> d2=diff(d1)
d2 =
    1   -5
>> d3=diff(d2)
d3 =
   -6
```

6.5.1 Relación entre derivadas y diferencias finitas

Desarrollo de la serie de Taylor de una función que suponemos diferenciable, **f**, alrededor de un punto x_0 a una distancia **h**, con un término:

$$f(x_0+h) = f_1 = f_0 + h\, f'(z), \text{ para algún } z \in [x_0, x_1]$$

De donde:

$$f'(z) = \frac{f_1 - f_0}{h} = \frac{\Delta^1 f_0}{h}, \text{ para algún } z \in [x_0, x_1]$$

Es el Teorema del Valor Medio. Este teorema de existencia, con alguna precaución, se usa como una aproximación para la primera derivada en el intervalo especificado:

$\frac{\Delta^1 f_0}{h}$ es una aproximación para **f'** en el intervalo $[x_0, x_1]$.

Desarrollo de la serie de Taylor de la función **f**, que suponemos diferenciable, alrededor del punto x_1 hacia ambos a una distancia **h**, con dos términos:

(1) $$f_2 = f_1 + hf'_1 + \frac{h^2}{2!} f''(z_1), \text{ para algún } z_1 \in [x_1, x_2]$$

(2) $$f_0 = f_1 - hf'_1 + \frac{h^2}{2!} f''(z_2), \text{ para algún } z_2 \in [x_0, x_1]$$

Sumando (1) y (2), sustituyendo la suma $f''(z_1) + f''(z_2)$ por un valor promedio $2f''(z)$ y despejando:

$$f''(z) = \frac{f_2 - 2f_1 + f_0}{h^2} = \frac{\Delta^2 f_0}{h^2}, \text{ para algún } z \in [x_0, x_2]$$

$\frac{\Delta^2 f_0}{h^2}$ es una aproximación para **f''** en el intervalo $[x_0, x_2]$

En general,

Definición. Relación entre derivadas y diferencias finitas

$$f^{(n)}(z) = \frac{\Delta^n f_0}{h^n}, \text{ para algún } z \text{ en el intervalo } [x_0, x_n].$$

$\frac{\Delta^n f_0}{h^n}$ es una aproximación para $f^{(n)}$ en el intervalo $[x_0, x_n]$.

Si suponemos que **f** tiene derivadas y que no cambian significativamente en el intervalo considerado, esta fórmula debe usarse con precaución para aproximar sus derivadas. El valor de **h** debe ser pequeño, pero si es muy pequeño puede aparecer el error de redondeo al restar números muy cercanos y la estimación de las derivadas de orden alto no será aceptable. Si se usa esta aproximación para acotar el error, se debería tomar el mayor valor de la diferencia finita tabulada y del mismo orden.

6.5.2 Diferencias finitas de un polinomio

Si la función **f** de donde provienen los datos es un polinomio de grado n, entonces la n-ésima diferencia finita será constante y las siguientes diferencias finitas se anularán.

Demostración:

Sea f un polinomio de grado n: $f(x) = a_0x^n + a_1x^{n-1} + \ldots + a_n$

Su n-ésima derivada es una constante: $f^{(n)}(x) = n!\, a_0$

Por lo tanto, la n-ésima diferencia finita también será constante: $\Delta^n f_i = h^n f^{(n)}(x) = h^n\, n!\, a_0$

Ejemplo

Tabule las diferencias finitas de $f(x) = 2x^2 + x + 1$, para $x = -2, -1, 0, 1, 2, 3$

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	-2	7	-5	4	0	0
1	-1	2	-1	4	0	0
2	0	1	3	4	0	
3	1	4	7	4		
4	2	11	11			
5	3	22				

El polinomio de interpolación coincidirá con la función, por la propiedad de unicidad.

Ejemplo. *Verifique si la siguiente función puede expresarse mediante un polinomio en el mismo dominio.*

$$f(x) = 2x + \sum_{i=1}^{x} i^2 \quad , x = 1, 2, 3, \ldots$$

Solución

La función con el sumatorio expresado en forma desarrollada:

$$f(x) = 2x + (1^2 + 2^2 + 3^2 + \ldots + x^2), \quad x = 1, 2, 3, \ldots$$

Tomando algunos puntos de esta función, tabulamos las diferencias finitas:

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	1	3	6	5	2	0
1	2	9	11	7	2	0
2	3	20	18	9	2	
3	4	38	27	11		
4	5	65	38			
5	6	103				

Siendo f(x) una expresión algebraica, y observado que la tercera diferencia finita es constante, se puede concluir que es un polinomio de grado tres. Para encontrar el polinomio de interpolación podemos usar la conocida fórmula de Lagrange. También se puede obtener el polinomio de interpolación con métodos basados en los valores tabulados de diferencias finitas.

6.6 El polinomio de interpolación de diferencias finitas avanzadas

Dado un conjunto de puntos (x_i, f_i), **i=0, 1,…, n** espaciados en forma regular en una distancia **h** y que provienen de una función desconocida **f(x),** pero supuestamente diferenciable, se desea obtener el polinomio de interpolación. Si se tienen tabuladas las diferencias finitas se puede obtener el polinomio de interpolación mediante un procedimiento de recurrencia y generalización.

Suponer que se tienen dos puntos (x_0, f_0), (x_1, f_1) con los cuales se debe obtener el polinomio de primer grado, es decir, la ecuación de una recta:

$$p_1(x) = a_0 + a_1(x-x_0)$$

Sustituyendo los dos puntos y despejando a_0 y a_1 se obtienen, agregando la notación usual:

$$a_0 = f_0$$

$$a_1 = \frac{f_1 - f_0}{x_1 - x_0} = \frac{\Delta^1 f_0}{h}$$

$$p_1(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0)$$

En el caso de tener tres puntos (x_0, f_0), (x_1, f_1), (x_2, f_2), se debe obtener el polinomio de segundo grado. Se propone la siguiente forma para este polinomio:

$$p_2(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)$$

Sustituyendo los tres puntos y despejando a_0, a_1 y a_2 se obtienen, incluyendo la notación usual:

$$a_0 = f_0$$

$$a_1 = \frac{f_1 - f_0}{x_1 - x_0} = \frac{\Delta^1 f_0}{h}$$

$$a_2 = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = \frac{\frac{\Delta^1 f_1}{h} - \frac{\Delta^1 f_0}{h}}{2h} = \frac{\Delta^1 f_1 - \Delta^1 f_0}{2h^2} = \frac{\Delta^2 f_0}{2h^2}$$

$$p_2(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2h^2}(x - x_0)(x - x_1)$$

Se puede generalizar para un conjunto de puntos (x_i, f_i), $i=0, 1,\ldots, n$:

Definición. Polinomio de diferencias finitas avanzadas o polinomio de Newton

$$p_n(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots + \frac{\Delta^n f_0}{n!h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Si las diferencias finitas están tabuladas en forma de un triángulo, los coeficientes del polinomio de interpolación se toman directamente de la primera fila del cuadro de datos (fila sombreada):

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$		$\Delta^n f_i$
0	x_0	f_0	$\Delta^1 f_0$	$\Delta^2 f_0$	$\Delta^3 f_0$	…	$\Delta^n f_0$
1	x_1	f_1	$\Delta^1 f_1$	$\Delta^2 f_1$	$\Delta^3 f_n$	…	…
2	x_2	f_2	$\Delta^1 f_2$	$\Delta^2 f_2$	…	…	…
3	x_3	f_3	$\Delta^1 f_3$	…	…	…	…
…	…	…	…	…	…	…	…
n	x_n	f_n	…	…	…	…	…

Ejemplo. *Dados los siguientes puntos:* ***(2, 5), (3, 6), (4, 3), (5, 2),*** *encuentre el polinomio de interpolación que incluye a los cuatro datos usando el método de diferencias finitas*

El método de diferencias finitas es aplicable pues **h** es constante e igual a **1**

Tabla de diferencias finitas:

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$
0	2	5	1	-4	6
1	3	6	-3	2	
2	4	3	-1		
3	5	2			

Polinomio de tercer grado de diferencias finitas:

$$p_3(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2)$$

Reemplazando los coeficientes, tomados de la primera fila, y simplificando:

$$p_3(x) = 5 + \frac{1}{1}(x - 2) + \frac{-4}{2(1^2)}(x - 2)(x - 3) + \frac{6}{3!(1^3)}(x - 2)(x - 3)(x - 4)$$

$$= x^3 - 11x^2 + 37x - 33$$

6.6.1 Práctica computacional

Obtención del polinomio de diferencias finitas mediante comandos de MATLAB

Datos: **(2, 5), (3, 6), (4, 3), (5, 2)**

```
>> x=[2 3 4 5];
>> f=[5 6 3 2];
```

Cálculo de diferencias finitas

```
>> d1=diff(f)
d1 =
     1          -3          -1
>> d2=diff(d1)
d2 =
    -4           2
>> d3=diff(d2)
d3 =
     6
```

Obtención del polinomio de interpolación mediante sustitución directa

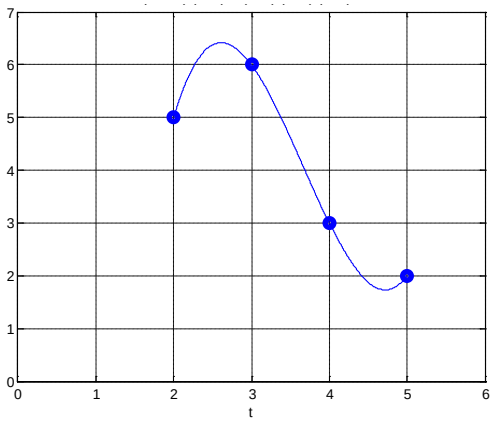
```
>> syms t
>> p=5+1/1*(t-2) + (-4)/(2*1^2)*(t-2)*(t-3) + 6/(6*1^3)*(t-2)*(t-3)*(t-4)
p =
    t - (2*t - 4)*(t - 3) + (t - 2)*(t - 3)*(t - 4) + 3
```

Simplificación algebraica

```
>> p=expand(p)
p =
   t^3 - 11*t^2 + 37*t – 33
```

Graficación de los puntos y del polinomio

```
>> plot(x,f,'o'),grid on,hold on
>> ezplot(p,[2,5])
>> axis([0,5,0,7])
```



Interpolación: evaluar **p(2.5)**

```
>> q=inline(p);
```

```
>> y=q(2.5)
y =
   6.3750
```

Interpolación inversa: hallar **t** tal que **p(t)=4**, con el método de Newton

```
>> h=p-4;
>> g=inline(t-h/diff(h));
>> t=3.6;
>> t=g(t)
t =
   3.6892
>> t=g(t)
t =
   3.6889
>> t=g(t)
t =
   3.6889
```

6.6.2 Eficiencia del polinomio de interpolación de diferencias finitas

Forma original del polinomio de interpolación de diferencias finitas avanzadas:

$$p_n(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots$$
$$+ \frac{\Delta^n f_0}{n!h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

La evaluación de este polinomio es un método directo en el que se suman **n+1** términos. Cada término de esta suma requiere una cantidad variable de multiplicaciones en las que aparece el valor x que se interpola. Entonces la eficiencia de este algoritmo es: **$T(n) = O(n^2)$,** similar **a** la fórmula de Lagrange, y significativamente mejor que el método matricial que involucra la resolución de un sistema de ecuaciones lineales cuya eficiencia es **$T(n) = O(n^3)$**.

Sin embargo, el polinomio de diferencias finitas puede escribirse en forma recurrente:

$$p_n(x) = f_0 + \frac{(x - x_0)}{h}(\Delta^1 f_0 + \frac{(x - x_1)}{2h}(\Delta^2 f_0 + \frac{(x - x_2)}{3h}(\Delta^3 f_0 + \ldots + \frac{(x - x_{n-2})}{(n-1)h}(\Delta^{n-1} f_0 + \frac{(x - x_{n-1})}{nh}\Delta^n f_0)\ldots)))$$

Entonces, el polinomio puede expresarse mediante un algoritmo recursivo:

$$p_0 = \Delta^n f_0$$

$$p_1 = \Delta^{n-1} f_0 + \frac{(x - x_{n-1})}{nh} p_0$$

$$p_2 = \Delta^{n-2} f_0 + \frac{(x - x_{n-2})}{(n-1)h} p_1$$

.

.

$$p_{n-1} = \Delta^1 f_0 + \frac{(x - x_1)}{2h} p_{n-2}$$

$$p_n = \Delta^0 f_0 + \frac{(x - x_0)}{h} p_{n-1}, \qquad \text{siendo } \Delta^0 f_0 = f_0$$

La evaluación del polinomio con este procedimiento requiere **2n** sumas y restas y **3n** multiplicaciones y divisiones: **$T(n) = O(n)$**. Esto constituye una mejora significativa con respecto a los métodos anteriores. Sin embargo, la tabulación de las diferencias finitas tiene eficiencia **$O(n^2)$** pero únicamente contiene restas y debe calcularse una sola vez para interpolar en otros puntos.

Ejemplo. *Dados los siguientes puntos:* ***(1.2, 5), (1.4, 6), (1.6, 3), (1.8, 2),*** *use el algoritmo recursivo anterior para evaluar en* ***x=1.5*** *el polinomio de interpolación que incluye a los cuatro datos.*

Tabla de diferencias finitas:

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$
0	1.2	5	1	-4	6
1	1.4	6	-3	2	
2	1.6	3	-1		
3	1.8	2			

$$p_0 = \Delta^3 f_0 \qquad p_0 = 6$$

$$p_1 = \Delta^2 f_0 + \frac{(x - x_2)}{3h} p_0 \qquad p_1 = -4 + \frac{(1.5 - 1.6)}{3(0.2)} 6 = -5$$

$$p_2 = \Delta^1 f_0 + \frac{(x - x_1)}{2h} p_1 \quad \Rightarrow \quad p_2 = 1 + \frac{(1.5 - 1.4)}{2(0.2)} (-5) = -0.25$$

$$p_3 = \Delta^0 f_0 + \frac{(x - x_0)}{h} p_2 \qquad p_3 = 5 + \frac{(1.5 - 1.2)}{0.2} (-0.25) = 4.625$$

6.6.3 El error en el polinomio de interpolación de diferencias finitas

Polinomio de interpolación de diferencias finitas avanzadas:

$$p_n(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots$$
$$+ \frac{\Delta^n f_0}{n!h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Se tiene el error en el polinomio de interpolación:

$$E_n(x) = g(x)\frac{f^{(n+1)}(z)}{(n+1)!}, \quad x \neq x_i, \quad z \in [x_0, x_n], \quad \text{siendo} \quad g(x) = (x\text{-}x_0)(x\text{-}x_1)\ldots(x\text{-}x_n)$$

Si los puntos están regularmente espaciados a una distancia **h**, se estableció una relación entre las derivadas de **f** y las diferencias finitas:

$$f^{(n)}(t) = \frac{\Delta^n f_0}{h^n}, \quad \text{para algún } t \text{ en el dominio de los datos dados}$$

Es una aproximación para la derivada si no cambia significativamente y **h** es pequeño.

Extendiendo esta relación a la siguiente derivada y sustituyendo en la fórmula del error en la interpolación

$$E_n(x) = g(x)\frac{f^{(n+1)}(z)}{(n+1)!} \cong g(x)\frac{\Delta^{n+1} f_0}{h^{n+1}(n+1)!} = (x - x_0)(x - x_1)\ldots(x - x_n)\frac{\Delta^{n+1} f_0}{h^{n+1}(n+1)!}$$

Si se compara con el polinomio de diferencias finitas avanzadas se observa que el error en la interpolación es aproximadamente igual al siguiente término que no es incluido en el polinomio de interpolación.

Definición. Estimación del error en el polinomio de interpolación de diferencias finitas

$$E_n(x) \cong \frac{\Delta^{n+1} f_0}{h^{n+1}(n+1)!}(x-x_0)(x-x_1)\ldots(x-x_n)$$

Si se toman **n+1** puntos para construir el polinomio de interpolación de grado **n**, y los puntos provienen de un polinomio de grado **n**, entonces el $f^{(n+1)}(\ )$ es cero y también $\Delta^n f$, por lo tanto $E_n(x)$ también es cero, en forma consistente con la propiedad de unicidad del polinomio de interpolación.

Ejemplo. *Para aplicar esta definición se usan los siguientes datos*

i	x_i	f_i	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	0.0	1.000000	0.110517	0.023247	0.003666	0.000516
1	0.1	1.110517	0.133764	0.026913	0.004182	
2	0.2	1.244281	0.160677	0.031095		
3	0.3	1.404958	0.191772			
4	0.4	1.596730				

En la tabla puede observarse que las diferencias finitas tienden a reducir su valor, entonces un polinomio de interpolación es una aproximación adecuada para esta función. Adicionalmente, las diferencias finitas en cada columna tienen valores de similar magnitud, por lo tanto se pueden usar para estimar a las derivadas.

El grado del polinomio de interpolación depende del error que toleramos y su valor está relacionado directamente con el orden de la diferencia finita incluida.

Supongamos que deseamos evaluar el polinomio de interpolación en **x = 0.08** usando el polinomio de diferencias finitas avanzadas de segundo grado.

$$f(x) \cong p_2(x) = f_0 + \frac{\Delta^1 f_0}{h}(x-x_0) + \frac{\Delta^2 f_0}{2!h^2}(x-x_0)(x-x_1)$$

$$f(0.08) \cong p_2(0.08) = 1 + \frac{0.110517}{0.1}(0.08-0) + \frac{0.023247}{2(0.1)^2}(0.08-0)(0.08-0.1) = 1.086554$$

Estimar el error en la interpolación:

$$E_2(x) \cong \frac{\Delta^3 f_0}{3!h^3}(x-x_0)(x-x_1)(x-x_2)$$

$$E_2(0.08) \cong \frac{0.003666}{3!0.1^3}(0.08-0)(0.08-0.1)(0.08-0.2) = 0.0001173$$

Para comparar, calculemos el valor exacto de **f(0.08)** con la función de la cual fueron tomados los datos: $f(x) = x\,e^x + 1$

$$f(0.08) = 0.08\,e^{0.08} + 1 = 1.086663\ldots$$

El error exacto es

$$1.083287 - 1.086554 = 0.0001089$$

El valor calculado con el polinomio de interpolación de segundo grado concuerda muy bien con el valor exacto.

6.6.4 Forma estándar del polinomio de interpolación de diferencias finitas

El polinomio de interpolación de diferencias finitas avanzadas:

$$p_n(x) = f_0 + \frac{\Delta^1 f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots$$
$$+ \frac{\Delta^n f_0}{n!h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Puede re-escribirse usando la siguiente sustitución: $S = \frac{x - x_0}{h}$

$$\frac{\Delta^1 f_0}{h}(x - x_0) = \Delta^1 f_0 S$$
$$\frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) = \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - (x_0 + h)) = \frac{\Delta^2 f_0}{2!h^2}\frac{(x - x_0)}{h}\frac{(x - x_0 - h)}{h} = \frac{\Delta^2 f_0}{2!}S(S-1)$$

. . . (sucesivamente)

Mediante recurrencia se puede generalizar:

$$p_n(S) = f_0 + \Delta^1 f_0 S + \frac{\Delta^2 f_0}{2!}S(S-1) + \frac{\Delta^3 f_0}{3!}S(S-1)(S-2) + \ldots + \frac{\Delta^n f_0}{n!}S(S-1)(S-2)\ldots(S-n+1)$$

Con la definición del coeficiente binomial

$$\binom{S}{i} = \frac{S(S-1)(S-2)\ldots(S-i+1)}{i!}$$

Se obtiene una forma compacta para el polinomio de interpolación de diferencias finitas:

Definición. Forma estándar del polinomio de interpolación de diferencias finitas

$$p_n(S) = f_0 + \binom{S}{1}\Delta^1 f_0 + \binom{S}{2}\Delta^2 f_0 + \binom{S}{3}\Delta^3 f_0 + \ldots + \binom{S}{n}\Delta^n f_0 = \sum_{i=0}^{n}\binom{S}{i}\Delta^i f_0$$

También se puede expresar con esta notación el error en la interpolación:

$$E_n(x) \cong \frac{\Delta^{n+1} f_0}{h^{n+1}(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n)$$

Sustituyendo la definición $S = \frac{x - x_0}{h}$:

$$E_n(S) \cong S(S-1)(S-2)\ldots(S-n)\frac{\Delta^{n+1} f_0}{(n+1)!}$$

Que finalmente se puede expresar de la siguiente forma

Definición. Estimación del error en el polinomio de interpolación de diferencias finitas

$$E_n(S) \cong \binom{S}{n+1}\Delta^{n+1} f_0, \quad S = \frac{x - x_0}{h}, \quad x \neq x_i$$

6.7 El polinomio de interpolación de diferencias divididas

Dado un conjunto de puntos (x_i, f_i), i=0, 1,…, n espaciados en forma arbitraria y que provienen de una función desconocida **f(x)** pero supuestamente diferenciable, se desea obtener el polinomio de interpolación.

Una forma alternativa al método de Lagrange es el polinomio de diferencias divididas cuya fórmula se la puede obtener mediante un procedimiento de recurrencia generalizado.

Suponer que se tienen dos puntos $(x_0, f_0), (x_1, f_1)$ con los cuales se debe obtener el polinomio de primer grado, es decir, la ecuación de una recta:

$$p_1(x) = a_0 + a_1(x-x_0)$$

Sustituyendo los dos puntos y despejando a_0 y a_1 se obtienen:

$$a_0 = f_0 = f[x_0]$$
$$a_1 = \frac{f_1 - f_0}{x_1 - x_0} = f[x_{0:1}]$$

$$p_1(x) = f[x_0] + f[x_{0:1}](x - x_0)$$

La notación $f[x_{0:1}] = \frac{f_1 - f_0}{x_1 - x_0}$ denota la diferencia dividida en el rango $[x_0, x_1]$

La diferencia dividida $f[x_{0:1}]$ es una aproximación para $f'( )$ en el intervalo $[x_0, x_1]$

En el caso de tener tres puntos $(x_0, f_0), (x_1, f_1), (x_2, f_2)$, se debe obtener un polinomio de segundo grado. Se propone la siguiente forma para este polinomio:

$$p_2(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)$$

Sustituyendo los tres puntos y despejando a_0, a_1 y a_2 se obtienen:

$$a_0 = f_0 = f[x_0]$$
$$a_1 = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f[x_0] - f[x_1]}{x_1 - x_0} = f[x_{0:1}]$$
$$a_2 = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = \frac{f[x_{1:2}] - f[x_{0:1}]}{x_2 - x_0} = f[x_{0:2}]$$
$$p_2(x) = f[x_0] + f[x_{0:1}](x - x_0) + f[x_{0:2}](x - x_0)(x - x_1)$$

La notación $f[x_{0:2}] = \frac{f[x_{1:2}] - f[x_{0:1}]}{x_2 - x_0}$ denota la diferencia dividida en el rango $[x_0, x_2]$

Al extender la recurrencia y generalizar al conjunto de puntos (x_i, f_i), $i=0, 1,\ldots, n$ se tiene:

$$f[x_{0:n}] = \frac{f[x_{1:n}] - f[x_{0:n-1}]}{x_n - x_0}$$

Definición. Polinomio de diferencias divididas

$$p_n(x) = f[x_0] + f[x_{0:1}](x - x_0) + f[x_{0:2}](x - x_0)(x - x_1) + f[x_{0:3}](x - x_0)(x - x_1)(x - x_2) + \ldots$$
$$+ f[x_{0:n}](x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Es conveniente tabular las diferencias divididas en forma de un triángulo en el que cada columna a la derecha se obtiene de la resta de los dos valores inmediatos de la columna anterior. Este resultado debe dividirse para la longitud del rango de los datos incluidos.

Los coeficientes del polinomio de interpolación serán los valores resultantes colocados en la primera fila del cuadro tabulado (fila sombreada):

i	x_i	$f[x_i]$	$f[x_{i:i+1}]$	$f[x_{i:i+2}]$	$f[x_{i:i+3}]$	$f[x_{i:i+4}]$
0	x_0	$f[x_0]$	$f[x_{0:1}]$	$f[x_{0:2}]$	$f[x_{0:3}]$	$f[x_{0:4}]$
1	x_1	$f[x_1]$	$f[x_{1:2}]$	$f[x_{1:3}]$	$f[x_{1:4}]$	
2	x_2	$f[x_2]$	$f[x_{2:3}]$	$f[x_{2:4}]$		
3	x_3	$f[x_3]$	$f[x_{3:4}]$			
4	x_4	$f[x_4]$				
…	…	…				

Ejemplo. *Dados los siguientes puntos:* ***(2, 5), (4, 6), (5, 3), (7, 2),*** *encuentre y grafique el polinomio de interpolación que incluye a los cuatro datos, usando el método de diferencias divididas:*

Tabla de diferencias divididas:

i	x_i	$f[x_i]$	$f[x_{i:i+1}]$	$f[x_{i:i+2}]$	$f[x_{i:i+3}]$
0	2	5	$\frac{6-5}{4-2} = \frac{1}{2}$	$\frac{-3-1/2}{5-2} = -\frac{7}{6}$	$\frac{5/6-(-7/6)}{7-2} = \frac{2}{5}$
1	4	6	$\frac{3-6}{5-4} = -3$	$\frac{-1/2-(-3)}{7-4} = \frac{5}{6}$	
2	5	3	$\frac{2-3}{7-5} = -\frac{1}{2}$		
3	7	2			

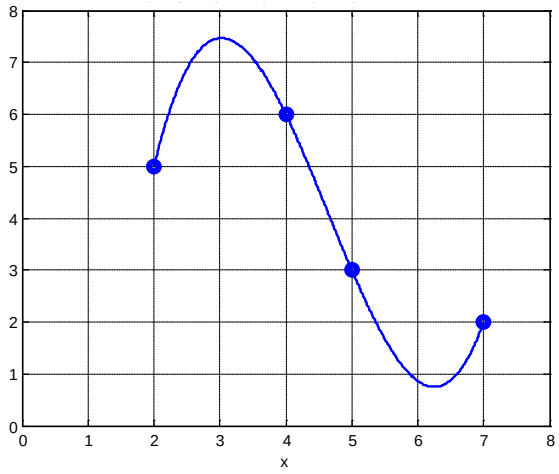
Polinomio de tercer grado de diferencias divididas:

$$p_n(x) = f[x_0] + f[x_{0:1}](x - x_0) + f[x_{0:2}](x - x_0)(x - x_1) + f[x_{0:3}](x - x_0)(x - x_1)(x - x_2)$$

Reemplazando los coeficientes, tomados de la primera fila, y simplificando:

$$p_3(x) = 5 + (1/2)(x-2) + (-7/6)(x-2)(x-4) + (2/5)(x-2)(x-4)(x-5)$$
$$= \frac{2}{5}x^3 - \frac{167}{30}x^2 + \frac{227}{10}x - \frac{64}{3}$$

Gráfico del polinomio:



6.7.1 El error en el polinomio de interpolación de diferencias divididas

Polinomio de diferencias divididas

$$p_n(x) = f[x_0] + f[x_{0:1}](x - x_0) + f[x_{0:2}](x - x_0)(x - x_1) + f[x_{0:3}](x - x_0)(x - x_1)(x - x_2) + \ldots$$
$$+ f[x_{0:n}](x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Error en el polinomio de interpolación:

$$E_n(x) = g(x)\frac{f^{(n+1)}(z)}{(n+1)!}, \quad x \neq x_i, \quad z \in [x_0, x_n], \text{ siendo } \quad g(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$$

Si los puntos estuviesen igualmente espaciados en una distancia **h** se tendría

$$f[x_{0:n}] = \frac{f[x_{1:n}] - f[x_{0:n-1}]}{x_n - x_0} = \frac{\Delta^n f_0}{n!h^n} \cong \frac{f^{(n)}(z)}{n!}$$

En el polinomio de diferencias divididas, **h** sería el cociente promedio de las **n** distancias entre las abscisas de los datos.

Extendiendo esta relación a la siguiente derivada y sustituyendo en la fórmula del error en la interpolación

$$E_n(x) = g(x)\frac{f^{(n+1)}(z)}{(n+1)!} \cong g(x)f[x_{0:n+1}] = (x - x_0)(x - x_1)\ldots(x - x_n)f[x_{0:n+1}]$$

Si se compara con el polinomio de diferencias divididas se observa que el error en el polinomio de interpolación es aproximadamente igual al siguiente término del polinomio que no es incluido.

6.8 El polinomio de mínimos cuadrados

Dados los puntos (x_i, f_i), $i = 1, 2,..., n$, que corresponden a observaciones o mediciones. Si se considera que estos datos contienen errores y que es de interés modelar únicamente su tendencia, entonces el polinomio de interpolación no es una buena opción. Una alternativa es el polinomio de mínimos cuadrados. Estos polinomios tienen mejores propiedades para realizar predicciones o extrapolaciones.

Si los datos tienen una tendencia lineal, la mejor recta será aquella que se coloca entre los puntos de tal manera que se minimizan las distancias de los puntos dados a esta recta, la cual se denomina recta de mínimos cuadrados y se la obtiene con el siguiente procedimiento:

Para cada valor x_i se tiene el dato f_i y el valor $p(x_i)$ obtenido con la recta de mínimos cuadrados.

Sea $p(x) = a_1 + a_2x$ la recta de mínimos cuadrados

Si $e_i = f_i - p(x_i)$ es la diferencia entre el dato y el punto de la recta de mínimos cuadrados., entonces, el objetivo es minimizar e_i^2 para todos los puntos. El cuadrado es para considerar distancia, no importa si el punto está sobre o debajo de la recta

Criterio de para obtener la recta de mínimos cuadrados

Minimizar $$S = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (f_i - p(x_i))^2 = \sum_{i=1}^{n} (f_i - a_1 - a_2x)^2 \quad .$$

Para minimizar esta función **S** cuyas variables son a_1, a_2 se debe derivar con respecto a cada variable e igualar a cero:

$$\frac{\partial S}{\partial a_1} = 0 : \quad \frac{\partial}{\partial a_1} \sum_{i=1}^{n} \left(f_i - a_1 - a_2x_i\right)^2 = 0 \Rightarrow na_1 + a_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} f_i$$

$$\frac{\partial S}{\partial a_2} = 0 : \quad \frac{\partial}{\partial a_2} \sum_{i=1}^{n} \left(f_i - a_1 - a_2x_i\right)^2 = 0 \Rightarrow a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i f_i$$

De esta manera se obtiene el sistema de ecuaciones lineales:

$$a_1 n + a_2 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} f_i$$

$$a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i f_i$$

De donde se obtienen los coeficientes a_1 y a_2 para la recta de mínimos cuadrados:

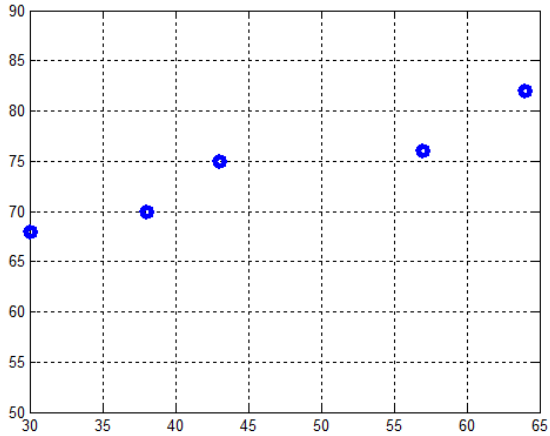
$$p(x) = a_1 + a_2 x$$

Ejemplo

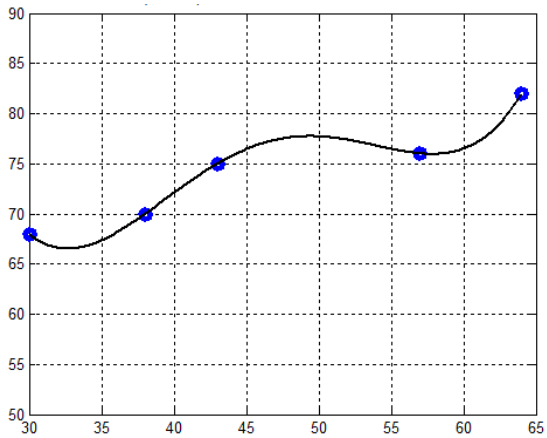
Los siguientes datos corresponden a una muestra de 5 estudiantes que han tomado cierta materia. Los datos incluyen la calificación parcial y la calificación final. Se pretende encontrar un modelo que permita predecir la calificación final que obtendría un estudiante dada su calificación parcial.

Estudiante	*Nota Parcial*	*Nota final*
1	*43*	*75*
2	*64*	*82*
3	*38*	*70*
4	*57*	*76*
5	*30*	*68*

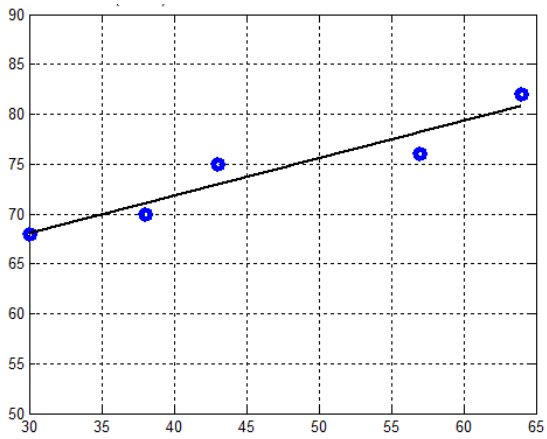
Representación de los datos en un diagrama



Se observa que los datos tienen aproximadamente una tendencia lineal



El polinomio de interpolación es exacto pero no muestra de forma simple la tendencia de los datos



La recta de mínimos cuadrados representa mejor la tendencia de los datos considerando además que no son observaciones exactas. Las interpolaciones o predicciones serían más apropiadas

El estudio detallado de este modelo se denomina Regresión Lineal y permite establecer criterios acerca de la calidad de la representación de los datos con la recta de mínimos cuadrados. Adicionalmente, mediante una transformación se pueden representar mediante la recta de mínimos cuadrados, conjuntos de datos que tienen otro tipo de tendencia.

6.8.1 Práctica computacional

Obtención de la recta de mínimos cuadrados mediante comandos directos de MATLAB

```
>> x=[43 64 38 57 30];
>> f=[75 82 70 76 68];
>> plot(x,f,'o'),grid on,hold on                    Gráfico de los datos
>> p=lagrange(x,f);
>> ezplot(p,[30,64])                               El polinomio de interpolación
>> d=[length(x) sum(x); sum(x) sum(x.^2)]
d =
     5     232
   232   11538
>> c=[sum(f); sum(x.*f)]
c =
   371
 17505
>> a=inv(d)*c
a =
  56.7610
   0.3758
>> syms t
>> p1=a(1)+a(2)*t;                                 La recta de mínimos cuadrados
>> ezplot(p1,[30,64])
```

6.9 Ejercicios y problemas con el polinomio de interpolación

Fórmula de Lagrange

1. Los siguientes datos son observaciones de los ingresos **f** en base al monto de inversión **x** realizada en cierto negocio, en miles de dólares: **(5.5, 83.0), (8.2, 94.5), (12.4, 105.0), (19.0, 92.0).**

a) Encuentre el polinomio de interpolación de tercer grado con la fórmula de Lagrange

Con este polinomio determine:

b) La ganancia que se obtiene si la inversión fuese 15.0

c) Cuanto habría que invertir si se desea una ganancia de 100.0

d) Para que valor de inversión se obtiene la máxima ganancia.

2. Encuentre un polinomio de interpolación para expresar en forma exacta la suma de los cuadrados de los primeros **x** números impares:

$$s(x) = 1^2 + 3^2 + 5^2 + \ldots + (2x-1)^2$$

3. Los siguientes datos pertenecen a la curva de Lorentz, la cual relaciona el porcentaje de ingreso económico global de la población en función del porcentaje de la población:

% de población	% de ingreso global
25	**10**
50	**25**
75	**70**
100	**100**

Ej. El 25% de la población tiene el 10% del ingreso económico global.

a) Use los cuatro datos para construir un polinomio para expresar esta relación

b) Con el polinomio determine el porcentaje de ingreso económico que le corresponde al 60% de la población

c) Con el polinomio determine a que porcentaje de la población le corresponde el 60% de ingreso económico global. Resuelva la ecuación resultante con el método de Newton

Error en la Interpolación

1. La función de variable real **$f(x)=\cos(x)e^x + 1$,** $0 \leq x \leq \pi$, será aproximada con el polinomio de segundo grado p(x) que incluye a los tres puntos **f(0), f($\pi/2$), f(π).**

a) Determine el error en la aproximación si $x = \pi/4$

b) Encuentre la magnitud del máximo error $E(x)=f(x)-p(x)$, que se produciría al usar p(x) como una aproximación a f(x). Resuelva la ecuación no lineal resultante con la fórmula de Newton con un error máximo de 0.0001

Polinomio de Diferencias Finitas

1. Luego de efectuarse un experimento se anotaron los resultados:**(x_i, f_i)** y se tabularon las diferencias finitas. Accidentalmente se borraron algunos valores quedando únicamente lo que se muestra a continuación:

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
1.3	3.534	--------	0.192	0.053	0.002
1.5	--------	--------	--------	--------	
1.7	--------	--------	--------		
1.9	--------	--------			
2.1	--------				

También se había hecho una interpolación lineal con el polinomio de diferencias finitas en **x = 1.4** obteniéndose como resultado de la interpolación el valor **4.0755**

a) Reconstruya la tabla de diferencias finitas
b) Encuentre el valor de **f(1.62)** con un polinomio de interpolación de Diferencias Finitas Avanzadas de tercer grado, y estime el error en la interpolación.
c) Encuentre el valor de **x** tal que **f(x) = 5.4** con un polinomio de interpolación de diferencias finitas avanzadas de tercer grado. Para obtener la respuesta debe resolver una ecuación cúbica. Use el Método de Newton y obtenga el resultado con cuatro decimales exactos. Previamente encuentre un intervalo de convergencia.

2. La suma de los cuadrados de los primeros k números pares:

$$s(k) = 2^2 + 4^2 + 6^2 + \ldots + (2k)^2$$

Se puede expresar exactamente mediante un polinomio de interpolación.
a) Encuentre el polinomio de interpolación con el polinomio de diferencias finitas
b) Calcule s(100) usando el polinomio.

3. Dados los puntos **(x,f(x))** de una función:

x :	1.0000	1.1000	1.2000	1.3000	1.4000	1.5000	1.6000
f(x) :	2.2874	2.7726	3.2768	3.7979	4.3327	4.8759	5.4209

a) Encuentre el valor de f(1.05) con un polinomio de tercer grado.
b) Estime el error en el resultado obtenido. Use la fórmula del error en la interpolación.

Interpolación con funciones de más variables

1. Se registraron los siguientes datos de la cantidad de producto obtenido experimentalmente en parcelas de cultivo en las que se suministraron tres cantidades diferentes de fertilizante tipo 1 y cuatro cantidades diferentes de fertilizante tipo 2:

	Fertilizante 2			
Fert. 1	**1.2**	**1.4**	**1.6**	**1.8**
1.0	7.2	7.8	7.5	7.3
1.5	8.2	8.6	9.2	9.0
2.0	9.5	9.6	9.3	8.6

Use todos los datos dados para determinar mediante una interpolación polinomial con el método de Lagrange, la cantidad de producto que se obtendría si se usaran **1.2** de fertilizante 1 y **1.5** de fertilizante 2.

2. Una empresa que vende cierto producto ha observado que su demanda depende del precio al que lo vende (P en $/unidad) y también del precio al que la competencia vende un producto de similares características (Q en $/unidad). Recopilando información histórica respecto a lo que ha sucedido en el pasado se observó que la demanda diaria (unidades vendidas por día) de este producto fueron de:

		P		
		1	1.1	1.2
Q	1	100	91	83
	1.1	110	100	92
	1.2	120	109	100
	1.3	130	118	108

Use **todos los datos dados** y el polinomio de interpolación de Lagrange para estimar los **ingresos mensuales** de la empresa por la venta de este producto si decide venderlo a $1.15 por unidad y conoce que la competencia estableció un precio de $1.25 por unidad.

6.10 El trazador cúbico

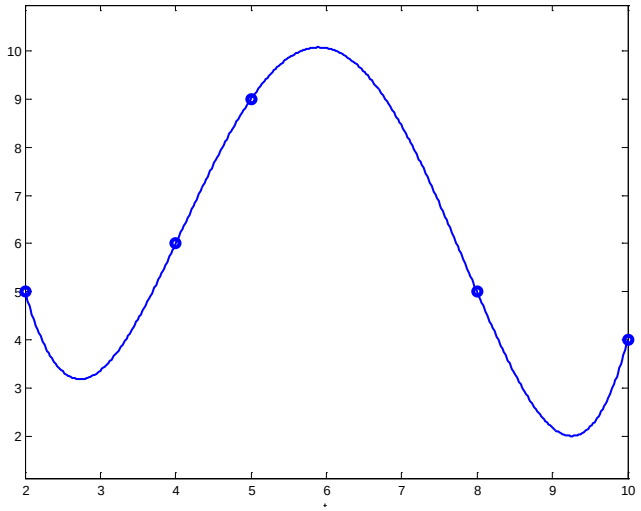
El polinomio de interpolación es útil si se usan pocos datos y que además tengan un comportamiento polinomial, así su representación es un polinomio de grado bajo y adecuado. Si no se cumplen estas condiciones, el polinomio puede tomar una forma inaceptable para representar a los datos, como se muestra en el siguiente ejemplo:

Ejemplo. *Encuentre el polinomio de interpolación que incluye a los siguientes datos*
(2, 5), (4,6), (5,9), (8,5), (10,4)

Con el método de Lagrange obtenemos el polinomio

$$\mathbf{p(x) = 19/288\ x^4 - 151/96\ x^3 + 1823/144\ x^2 - 118/3\ x + 401/9}$$

El gráfico se muestra a continuación



Se observa que en los intervalos **(2, 4),** y **(8, 10)** la forma del polinomio no es apropiada para expresar la tendencia de los datos.

Una opción pudiera ser colocar polinomios de interpolación en tramos. Por ejemplo un polinomio de segundo grado con los puntos **(2, 5), (4,6), (5,9)**, y otro polinomio de segundo grado con los puntos **(5,9), (8,5), (10,4)**. Sin embargo, en el punto intermedio **(5, 9)** en el que se unirían ambos polinomios de segundo grado se tendría un cambio de pendiente inaceptable.

Una mejor opción consiste en usar el Trazador Cúbico. Este dispositivo matemático equivale a la regla flexible que usan algunos dibujantes y que permite acomodarla para seguir de una manera suave la trayectoria de los puntos sobre un plano.

6.10.1 El trazador cúbico natural

Dados los puntos $\mathbf{(x_i, y_i)}$, **i = 1, 2, ..., n**, el **trazador cúbico natural** es un conjunto de **n-1** polinomios de grado tres colocados uno a uno entre cada par de puntos consecutivos, de tal manera que haya continuidad, manteniendo igual pendiente y curvatura con los polinomios de intervalos adyacentes.

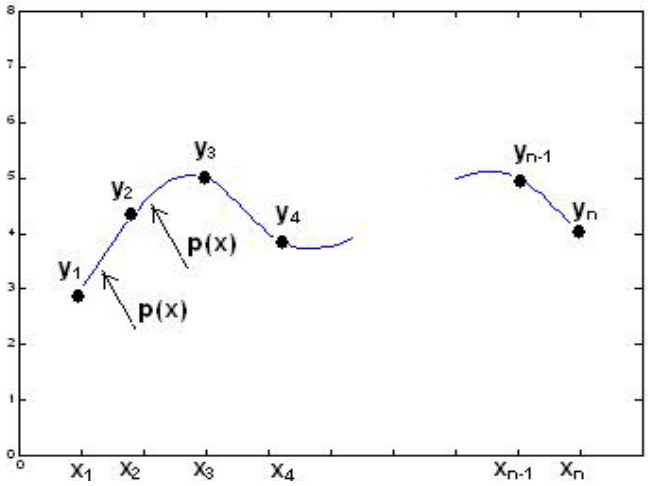
Definición: **Trazador cúbico natural T(x)**

$$\mathbf{T(x)} = \begin{cases} a_1(x-x_1)^3 + b_1(x-x_1)^2 + c_1(x-x_1) + d_1, & x_1 \le x \le x_2 \\ a_2(x-x_2)^3 + b_2(x-x_2)^2 + c_2(x-x_2) + d_2, & x_2 \le x \le x_3 \\ \quad \cdots & \\ a_{n-1}(x-x_{n-1})^3 + b_{n-1}(x-x_{n-1})^2 + c_{n-1}(x-x_{n-1}) + d_{n-1}, & x_{n-1} \le x \le x_n \end{cases}$$

Formulación para el trazador cúbico natural

Polinomio para cada intervalo:

$$y = p(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \qquad (0)$$
$$x_i \le x \le x_{i+1}, \qquad i = 1,2,\ldots,n-1$$



Para cada uno de estos polinomios deben determinarse los coeficientes
a_i, b_i, c_i, d_i, $i = 1, 2, \ldots, n-1$

Los puntos no necesariamente están espaciados en forma regular por lo que conviene asignar un nombre a cada una de las distancias entre puntos consecutivos:
$h_i = x_{i+1} - x_i$, $i = 1, 2, \ldots, n-1$

El siguiente desarrollo basado en las condiciones requeridas para **p(x)** permite obtener los coeficientes del polinomio.

Sea **y=p(x)** el polinomio en cualquier intervalo **i**, $i=1, 2, \ldots, n-1$

Este polinomio debe incluir a los extremos de cada intervalo **i:**

$x=x_i$: $y_i = a_i(x_I - x_i)^3 + b_i(x_I - x_i)^2 + c_i(x_I - x_i) + d_i = d_i \Rightarrow d_i = y_i$ (1)

$x=x_{i+1}$: $y_{i+1} = a_i(x_{I+1} - x_i)^3 + b_i(x_{I+1} - x_i)^2 + c_i(x_{I+1} - x_i) + d_i$

$= a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i$ (2)

Las dos primeras derivadas de $y = p(x)$

$y' = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$ (3)

$y'' = 6a_i(x - x_i) + 2b_i$ (4)

Por simplicidad se usa la siguiente notación para la segunda derivada

$y'' = S = 6a_i(x - x_i) + 2b_i$

Evaluamos la segunda derivada en los extremos del intervalo **i:**

$x=x_i$: $y''_i = S_i = 6a_i(x_i - x_i) + 2b_i = 2b_i \Rightarrow b_i = \dfrac{S_i}{2}$ (5)

$x=x_{i+1}$: $y''_{i+1} = S_{i+1} = 6a_i(x_{i+1} - x_i) + 2b_i = 6a_i h_i + 2b_i$

De donde se obtiene $a_i = \dfrac{S_{i+1} - S_i}{6h_i}$ (6)

Sustituimos **(1), (5),** y **(6)** en **(2)**

$$y_{i+1} = \frac{S_{i+1} - S_i}{6h_i} h_i^3 + \frac{S_i}{2} h_i^2 + c_i h_i + y_i$$

De donde se obtiene**:**

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \qquad (7)$$

Con lo que los coeficientes de **p(x)** quedan expresados mediante los datos dados y los valores de las segundas derivadas **S**

Coeficientes del trazador cúbico

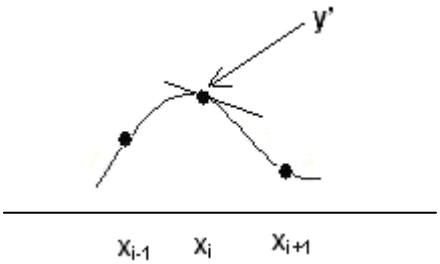
$$a_i = \frac{S_{i+1} - S_i}{6h_i}$$

$$b_i = \frac{S_i}{2} \qquad (8)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}$$

$$d_i = y_i \qquad i = 1, 2, ..., n-1$$

En el punto intermedio entre dos intervalos adyacentes, la pendiente de los polinomios debe ser igual:



Pendiente en el intervalo $[x_i, x_{i+1}]$, de **(3):**

$$y' = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

Evaluamos en el extremo izquierdo

$x=x_i$**:** $\quad y'_i = 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i$

Pendiente en el intervalo $[x_{i-1}, x_i]$, de **(3):**

$$y' = 3a_{i-1}(x - x_{i-1})^2 + 2b_{i-1}(x - x_{i-1}) + c_{i-1}$$

Evaluamos en el extremo derecho

$x=x_i$**:** $\quad y'_i = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1} = 3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1}$

En el punto x_i ambas pendientes deben tener el mismo valor:

$$c_i = 3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1}$$

Finalmente, se sustituyen las definiciones de c_i, a_{i-1}, b_{i-1}, c_{i-1}

$$\frac{y_{i+1} - y_i}{6h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} = 3\left(\frac{S_i - S_{i-1}}{6h_{i-1}}\right) h_{i-1}^2 + 2\left(\frac{S_{i-1}}{2}\right) h_{i-1} + \frac{y_i - y_{i-1}}{h_i} - \frac{2h_{i-1} S_{i-1} + h_{i-1} S_i}{6}$$

Después de simplificar se obtiene:

$$h_{i-1}S_{i-1} + 2(h_{i-1} + h_i)S_i + h_i S_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right), \quad i = 2, 3, ..., n-1 \qquad (9)$$

Esta ecuación debe evaluarse con los datos dados, con lo que se obtiene un sistema de **n – 2** ecuaciones lineales con las **n** variables: $S_1, S_2, ..., S_n$

Para obtener dos datos adicionales se considera que en el **Trazador Cúbico Natural** los puntos extremos inicial y final están sueltos por lo que no tienen curvatura. Con esta suposición el valor de la segunda derivada tiene un valor nulo en los extremos, y se puede escribir:

$$S_1 = 0, \quad S_n = 0 \qquad (10)$$

6.10.2 Algoritmo del trazador cúbico natural

Dados los puntos: (x_i, y_i), $i = 1, 2, ..., n$

1. Con la ecuación **(9)** y reemplazando los valores dados en **(10)** obtenga un sistema de **n-2** ecuaciones lineales con las incógnitas $S_2, S_3, ..., S_{n-1}$, (Sistema tridiagonal de ecuaciones lineales)
2. Resuelva el sistema y obtenga los valores de $S_2, S_3, ..., S_{n-1}$
3. Con las definiciones dadas en **(8)** obtenga los coeficientes para el trazador cúbico.
4. Sustituya los coeficientes en la definición dada en **(0)** y obtenga el polinomio del trazador cúbico en cada uno de los intervalos.

$$h_{i-1}S_{i-1} + 2(h_{i-1} + h_i)S_i + h_iS_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right), \quad i = 2, 3, ..., n-1 \qquad (9)$$

$$S_1 = 0, \quad S_n = 0 \qquad (10)$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i}$$

$$b_i = \frac{S_i}{2}$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_iS_i + h_iS_{i+1}}{6}$$

$$d_i = y_i \qquad i = 1, 2, ..., n-1 \qquad (8)$$

$$y = p(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$
$$x_i \le x \le x_{i+1}, \qquad i = 1, 2, ..., n-1 \qquad (0)$$

Ejemplo. *Encuentre el trazador cúbico natural para los datos* ***(2, 5), (4,6), (5,9), (8,5), (10,4)***

Solución:
Anotamos los datos en la terminología del trazador cúbico

n = 5

i	x_i	y_i	$h_i = x_{i+1} - x_i$
1	2	5	2
2	4	6	1
3	5	9	3
4	8	5	2
5	10	4	

$S_1 = 0$, $S_5 = 0$, de acuerdo a la definición **(10)**

Al sustituir en **(9)** se obtiene un sistema de ecuaciones lineales

$$\mathbf{h_{i-1}S_{i-1} + 2(h_{i-1} + h_i)S_i + h_iS_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right),\ i = 2, 3, 4}$$

i = 2: $h_1S_1 + 2(h_1 + h_2)S_2 + h_2S_3 = 6\left(\frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1}\right)$

$$2(0) + 2(2 + 1)S_2 + 1S_3 = 6\left(\frac{9-6}{1} - \frac{6-5}{2}\right) \quad \Rightarrow\ 6S_2 + S_3 = 15$$

i = 3: $h_2S_2 + 2(h_2 + h_3)S_3 + h_3S_4 = 6\left(\frac{y_4 - y_3}{h_3} - \frac{y_3 - y_2}{h_2}\right)$

$$1S_2 + 2(1 + 3)S_3 + 3S_4 = 6\left(\frac{5-9}{3} - \frac{9-6}{1}\right) \quad \Rightarrow\ S_2 + 8S_3 + 3S_4 = -26$$

i = 4: $h_3S_3 + 2(h_3 + h_4)S_4 + h_4S_5 = 6\left(\frac{y_5 - y_4}{h_4} - \frac{y_4 - y_3}{h_3}\right)$

$$3S_3 + 2(3 + 2)S_4 + 2(0) = 6\left(\frac{5-9}{3} - \frac{9-6}{1}\right) \quad \Rightarrow\ S_3 + 10S_4 = 5$$

$$\begin{bmatrix} 6 & 1 & 0 \\ 1 & 8 & 3 \\ 0 & 3 & 10 \end{bmatrix}\begin{bmatrix} S_2 \\ S_3 \\ S_4 \end{bmatrix} = \begin{bmatrix} 15 \\ -26 \\ 5 \end{bmatrix}$$ Resolviendo este sistema resulta $$\begin{bmatrix} S_2 \\ S_3 \\ S_4 \end{bmatrix} = \begin{bmatrix} 3.2212 \\ -4.3269 \\ 1.7981 \end{bmatrix}$$

Sustituimos estos valores en las definiciones **(8)** y se obtienen los coeficientes:

a_i	b_i	c_i	d_i
0.2684	0	-0.5737	5
-1.2580	1.6106	2.6474	6
0.3403	-2.1635	2.0946	9
-0.1498	0.8990	-1.6987	5

Los coeficientes corresponden a los cuatro polinomios segmentarios del trazador cúbico natural según la definición inicial en la ecuación **(0)**:

$$\mathbf{y = p(x) = a_i(x_I - x_i)^3 + b_i(x_I - x_i)^2 + c_i(x_I - x_i) + d_i\ ,\quad i = 1, 2, 3, 4}$$

i=1: $\mathbf{y = p(x) = a_1(x_I - x_i)^3 + b_1(x_I - x_i)^2 + c_1(x_I - x_i) + d_1\ ,}$
$\mathbf{= 0.2684(x - 2)^3 + 0(x - 2)^2 - 0.5737(x - 2) + 5,}$ $\mathbf{2 \le x \le 4}$

i=2: $\mathbf{y = p(x) = -1.2580(x - 4)^3 + 1.6106(x - 4)^2 + 2.6474(x - 4) + 6,}$ $\mathbf{4 \le x \le 5}$

i=3: $\mathbf{y = p(x) = 0.3403(x - 5)^3 - 2.1635(x - 5)^2 + 2.0946(x - 5) + 9,}$ $\mathbf{5 \le x \le 8}$

i=4: $\mathbf{y = p(x) = -0.1498(x - 8)^3 + 0.8990(x - 8)^2 - 1.6987(x - 8) + 6,}$ $\mathbf{8 \le x \le 10}$

6.10.3 Instrumentación computacional del trazador cúbico natural

La formulación del trazador cúbico natural se ha instrumentado en MATLAB mediante una función denominada **trazador** la cual proporciona un vector con puntos del trazador o los vectores con los coeficientes: **a, b, c, d.** Existe una versión propia de MATLAB equivalente a esta última función y se denomina **spline**

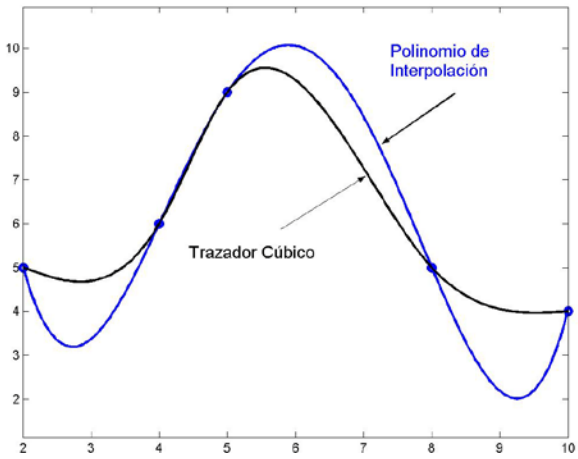
Debido a que el sistema resultante es de tipo **tridiagonal**, se usa un método específico muy eficiente para resolver estos sistemas. Debe haber por lo menos 4 puntos datos.

```
function [a,b,c,d]=trazador(x,y,z)
% Trazador Cúbico Natural: a(i)(x-x(i))^3+b(i)(x-x(i))^2+c(i)(x-x(i))+d(i),   n>3
% z es opcional: es el vector de puntos para evaluar al trazador
% Entrega puntos del trazador o los coeficientes de los polinomios segmentarios
n=length(x);
clear A B C D;
if n<4
  return
end
for i=1:n-1
  h(i)=x(i+1)-x(i);
end
s(1)=0;
s(n)=0;
B(1)=2*(h(1)+h(2));
C(1)=h(2);
D(1)=6*((y(3)-y(2))/h(2)-(y(2)-y(1))/h(1))-h(1)*s(1);
for i=2:n-3                                  % Sistema tridiagonal para obtener S
  A(i)=h(i);
  B(i)=2*(h(i)+h(i+1));
  C(i)=h(i+1);
  D(i)=6*((y(i+2)-y(i+1))/h(i+1)-(y(i+1)-y(i))/h(i));
end
A(n-2)=h(n-2);
B(n-2)=2*(h(n-2)+h(n-1));
D(n-2)=6*((y(n)-y(n-1))/h(n-1)-(y(n-1)-y(n-2))/h(n-2))-h(n-1)*s(n);
u=tridiagonal(A,B,C,D);
for i=2:n-1
  s(i)=u(i-1);
end
for i=1:n-1                                  % Coeficientes del trazador cúbico natural
  a(i)=(s(i+1)-s(i))/(6*h(i));
  b(i)=s(i)/2;
  c(i)=(y(i+1)-y(i))/h(i)-(2*h(i)*s(i)+h(i)*s(i+1))/6;
  d(i)=y(i);
end
if nargin==3                                 % Puntos del trazador cúbico natural
   p=[ ];
   m=length(z);
   for k=1:m
     t=z(k);
     for i=1:n-1
       if t>=x(i) & t<=x(i+1)
         p(k)=a(i)*(t-x(i))^3+b(i)*(t-x(i))^2+c(i)*(t-x(i))+d(i);
       end
     end
   end
   if m>1
      k=m;i=n-1;
      p(k)=a(i)*(t-x(i))^3+b(i)*(t-x(i))^2+c(i)*(t-x(i))+d(i);
   end
   clear a b c d;
   a=p;
end
```

Ejemplo. *Encuentre el trazador cúbico natural usando la función anterior para los siguientes puntos dados.* ***(2, 5), (4,6), (5,9), (8,5), (10,4)***

```
>> x = [2 4 5 8 10];
>> y = [5 6 9 5 4];
>> z = [2: 0.01: 10];          Puntos para evaluar el trazador
>> p = trazador(x, y, z);      Puntos del trazador
>> plot(x, y, 'o');            Gráfico de los puntos
>> hold on
>> plot(z, p)                  Gráfico del trazador
```

Gráfico del trazador cúbico natural junto con los puntos dados y el polinomio de interpolación:



Puede observarse la notable mejora en la representación de los datos dados.
El trazador cúbico mantiene una curvatura suave y continua y sigue mejor la tendencia de los puntos.

Coeficientes del trazador cúbico

```
>> [a, b, c, d] = trazador(x, y);

a =
    0.2684   -1.2580    0.3403   -0.1498
b =
         0    1.6106   -2.1635    0.8990
c =
   -0.5737    2.6474    2.0946   -1.6987
d =
    5.0000    6.0000    9.0000    5.0000
```

6.10.4 El trazador cúbico sujeto

En esta versión del trazador cúbico, los extremos ya no están sueltos sino sujetos y con alguna inclinación especificada. Por lo tanto, ya no se aplica la definición anterior **(10)**: $S_1 = 0,\ S_n = 0$

Dados los puntos (x_i, y_i), $i = 1, 2, ..., n$.

Adicionalmente se especifica como datos, la inclinación del trazador en los extremos:

$y'(x_1) = u$

$y'(x_n) = v$

Utilizamos la expresión **(3)** del análisis anterior:

$$y' = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$$

Sustituimos los datos dados para los polinomios en el primero y último intervalo:

En el primer intervalo:

$$x = x_1: \quad y'(x_1) = u = 3a_1(x_1 - x_1)^2 + 2b_1(x_1 - x_1) + c_1 = c_1$$

Se sustituye la definición de c_1 y se tiene

$$u = \frac{y_2 - y_1}{h_1} - \frac{2h_1S_1 + h_1S_2}{6}$$

Finalmente,

$$-\frac{1}{3}h_1S_1 - \frac{1}{6}h_1S_2 = u - \frac{y_2 - y_1}{h_1} \qquad (11)$$

En el último intervalo:

$$x = x_n: \quad y'(x_n) = v = 3a_{n-1}(x_n - x_{n-1})^2 + 2b_{n-1}(x_n - x_{n-1}) + c_{n-1}$$

$$v = 3a_{n-1}\, h_{n-1}^2 + 2b_{n-1}\, h_{n-1} + c_{n-1}$$

Se sustituyen las definiciones de los coeficientes:

$$v = 3\left(\frac{S_n - S_{n-1}}{6h_{n-1}}\right)h_{n-1}^2 + 2\left(\frac{S_{n-1}}{2}\right)h_{n-1} + \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{2h_{n-1}S_{n-1} + h_{n-1}S_n}{6}$$

De donde se tiene

$$v = \left(\frac{S_n - S_{n-1}}{2}\right)h_{n-1} + S_{n-1}h_{n-1} + \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{2h_{n-1}S_{n-1} + h_{n-1}S_n}{6}$$

Finalmente:

$$\frac{1}{6}h_{n-1}S_{n-1} + \frac{1}{3}h_{n-1}S_n = v - \frac{y_n - y_{n-1}}{h_{n-1}} \qquad (12)$$

Las ecuaciones **(11) y (12)** junto con las ecuaciones que se obtienen de **(9)** conforman un sistema tridiagonal de **n** ecuaciones lineales con las **n** variables $S_1, S_2, ..., S_n$

El resto del procedimiento es similar al que corresponde al trazador cúbico natural.

6.10.5 Algoritmo del trazador cúbico sujeto

Dados los puntos: (x_i, y_i), $i = 1, 2, ..., n$

1. Con las ecuaciones **(9), (11)** y **(12)** obtenga un sistema de **n** ecuaciones lineales con las incógnitas $S_1, S_2, ..., S_n$, (Sistema tridiagonal de ecuaciones lineales)
2. Resuelva el sistema y obtenga los valores de $S_1, S_2, ..., S_n$
3. Con las definiciones dadas en **(8)** obtenga los coeficientes para el trazador cúbico.
4. Sustituya los coeficientes en la definición dada en **(0)** y obtenga el polinomio del trazador cúbico en cada uno de los intervalos.

$$-\frac{1}{3}h_1S_1 - \frac{1}{6}h_1S_2 = u - \frac{y_2 - y_1}{h_1} \qquad (11)$$

$$h_{i-1}S_{i-1} + 2(h_{i-1} + h_i)S_i + h_iS_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right), \quad i = 2, 3, ..., n-1 \qquad (9)$$

$$\frac{1}{6}h_{n-1}S_{n-1} + \frac{1}{3}h_{n-1}S_n = v - \frac{y_n - y_{n-1}}{h_{n-1}} \qquad (12)$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i}$$

$$b_i = \frac{S_i}{2} \qquad (8)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_iS_i + h_iS_{i+1}}{6}$$

$$d_i = y_i \qquad i = 1, 2, ..., n-1$$

$$y = p(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \qquad (0)$$

$$x_i \le x \le x_{i+1}, \qquad i = 1, 2, ..., n-1$$

6.10.6 Instrumentación computacional del trazador cúbico sujeto

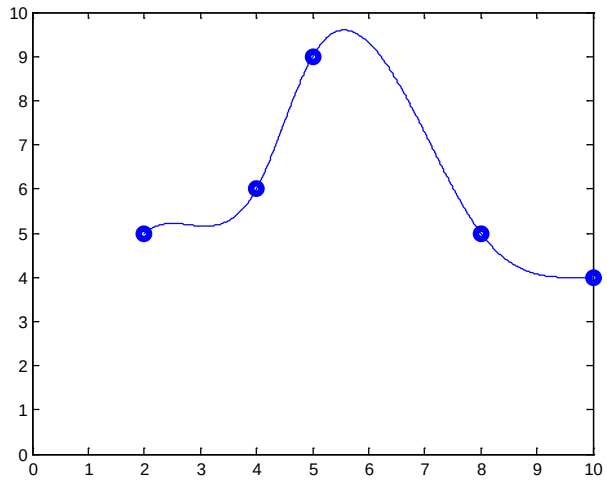
La formulación del trazador cúbico sujeto se ha instrumentado en MATLAB mediante una función denominada **trazadorsujeto** la cual proporciona un vector con puntos del trazador o los vectores con los coeficientes: **a, b, c, d.**

```
function [a,b,c,d]=trazadorsujeto(x,y,u,v,z)
% Trazador Cúbico Sujeto
% u,v son las pendientes en los extremos
% a(i)(z-x(i))^3+b(i)(z-x(i))^2+c(i)(z-x(i))+d(i),   n>3
% z es opcional: es el vector de puntos para evaluar al trazador
% Entrega puntos del trazador o los coeficientes de los polinomios segmentarios
n=length(x);
clear A B C D;
if n<4
  return
end
for i=1:n-1
  h(i)=x(i+1)-x(i);
end
B(1)=-2*h(1)/6;
C(1)=-h(1)/6;
D(1)=u-(y(2)-y(1))/h(1);
for i=2:n-1
  A(i)=h(i-1);
  B(i)=2*(h(i-1)+h(i));
  C(i)=h(i);
  D(i)=6*((y(i+1)-y(i))/h(i)-(y(i)-y(i-1))/h(i-1));
end
A(n)=h(n-1)/6;
B(n)=h(n-1)/3;
D(n)=v-(y(n)-y(n-1))/h(n-1);
s=tridiagonal(A,B,C,D);
for i=1:n-1                          % Coeficientes del trazador cúbico sujeto
  a(i)=(s(i+1)-s(i))/(6*h(i));
  b(i)=s(i)/2;
  c(i)=(y(i+1)-y(i))/h(i)-(2*h(i)*s(i)+h(i)*s(i+1))/6;
  d(i)=y(i);
end
if nargin==5                         % Puntos del trazador cúbico sujeto
   p=[ ];
   m=length(z);
   for k=1:m
     t=z(k);
     for i=1:n-1
       if t>=x(i) & t<=x(i+1)
         p(k)=a(i)*(t-x(i))^3+b(i)*(t-x(i))^2+c(i)*(t-x(i))+d(i);
       end
     end
   end
   if m>1
      k=m;i=n-1;
      p(k)=a(i)*(t-x(i))^3+b(i)*(t-x(i))^2+c(i)*(t-x(i))+d(i);
   end
   clear a b c d;
   a=p;
end
```

Ejemplo. *Encuentre el trazador cúbico sujeto usando la función anterior para los siguientes puntos dados.* ***(2, 5), (4,6), (5,9), (8,5), (10,4).*** *En el extremo izquierdo debe inclinarse 45^0 y debe terminar horizontal en el extremo derecho.*

```
>> x = [2 4 5 8 10];
>> y = [5 6 9 5 4];
>> z = [2: 0.01: 10];                    Puntos para evaluar el trazador
>> p = trazadorsujeto(x, y, 1, 0, z);    Puntos del trazador
>> plot(x, y, 'o');                      Gráfico de los puntos
>> hold on
>> plot(z, p)                            Gráfico del trazador
>> axis([0,10,0,10]);
```

Gráfico del trazador cúbico sujeto junto con los puntos dados:



Coeficientes de los polinomios segmentarios del trazador cúbico sujeto

```
>> [a, b, c, d] = trazadorsujeto(x, y, 1, 0)

a =
   0.5870   -1.4461    0.3535   -0.1728
b =
  -1.4240    2.0980   -2.2402    0.9412
c =
   1.0000    2.3480    2.2059   -1.6912
d =
    5     6     9     5
```

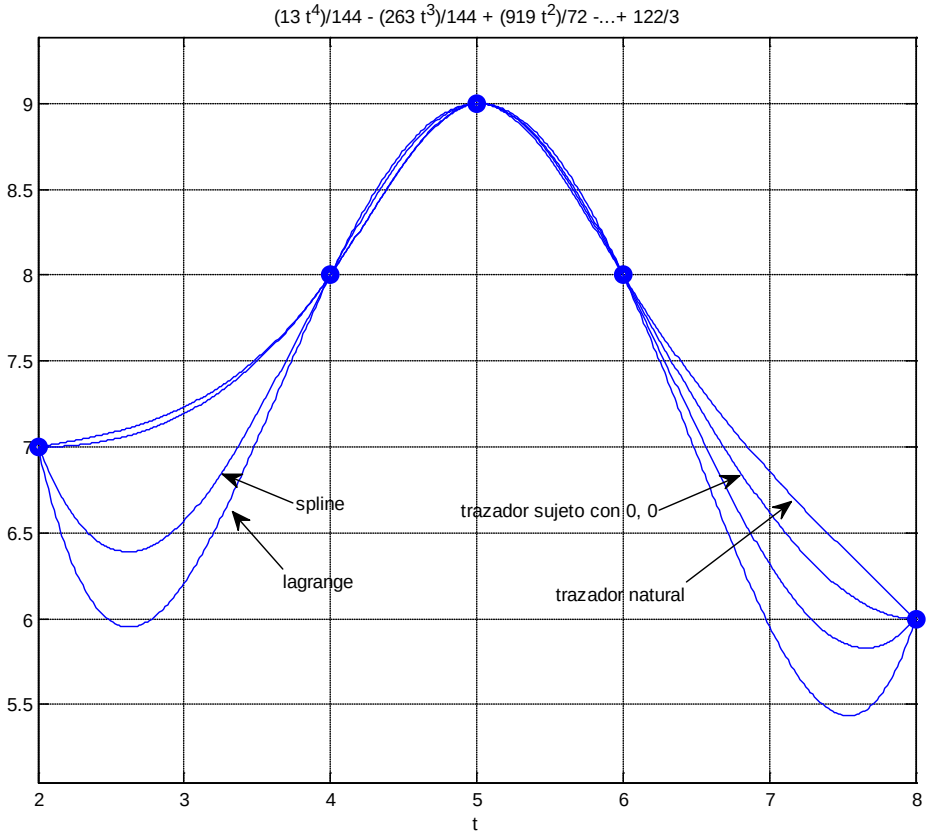
Ejemplo comparativo

Comparación gráfica del polinomio de interpolación y los trazadores cúbicos, incluyendo el trazador cúbico proporcionado con la función spline de MATLAB

Datos: (2, 7), (4, 8), (5, 9), (6, 8), (8,6)

```
>> x=[2 4 5 6 8];
>> y=[7 8 9 8 6];
>> plot(x,y,'o'),grid on,hold on
>> p=lagrange(x,y);
>> ezplot(p,[2,8])
>> z=[2:0.01:8];
```

```
>> v=trazador(x,y,z);
>> plot(z,v)
>> w=spline(x,y,z);
>> plot(z,w)
>> t=trazadorsujeto(x,y,0,0,z);
>> plot(z,t)
```



6.10.7 Ejercicios con el trazador cúbico

1. Con los siguientes datos (x, y): **(1.2, 4.6), (1.5, 5.3), (2.4, 6.0), (3.0, 4.8), (3.8, 3.1)**

a) Encuentre el trazador **cúbico natural**
b) Encuentre el valor interpolado para **x=2.25**

2. Con los siguientes datos (x, y): **y'(1.2) = 1, (1.5, 5.3), (2.4, 6.0), (3.0, 4.8), y'(3.8) = -1**

a) Encuentre el trazador **cúbico sujeto**
b) Encuentre el valor interpolado para **x=2.25**

Use las funciones instrumentadas en MATLAB para comprobar sus resultados

7 INTEGRACIÓN NUMÉRICA

Introducimos este capítulo mediante un problema de interés práctico en el que el modelo matemático resultante es la evaluación de un integral. El objetivo es evaluar numéricamente un integral y estimar la precisión del resultado.

Problema.

La siguiente función es fundamental en estudios estadísticos y se denomina función de densidad de la distribución Normal Estándar: Esta función permite calcular la probabilidad, ***P****, que la variable* ***Z*** *pueda tomar algún valor en un intervalo* ***[a, b]*** *según la siguiente definición:*

$$P(a \le Z \le b) = \int_a^b f(z)dz$$

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$$

$$z \in \Re$$

Debido a que esta función no es integrable analíticamente, es necesario utilizar métodos numéricos para estimar el valor de ***P****.*

Sea **f** una función integrable, definida en un intervalo cerrado y acotado **[a, b] con a<b∈R**. Es de interés calcular el valor de **A**:

$$A = \int_a^b f(x)dx$$

En general hay dos situaciones en las que son útiles los métodos numéricos:

1) El integral existe pero es muy difícil o no se puede evaluar analíticamente.

2) Únicamente se conocen puntos de **f(x)** pero se requiere calcular en forma aproximada el integral debajo de la curva descrita por los puntos dados

En ambos casos se trata de sustituir **f(x)** por alguna función más simple, siendo importante además estimar la precisión del resultado obtenido.

7.1 Fórmulas de Newton-Cotes

El enfoque básico para obtener fórmulas de integración numérica consiste en aproximar la función a ser integrada por el polinomio de interpolación. Las fórmulas así obtenidas se denominan de Newton-Cotes.

Si los puntos están espaciados regularmente, se puede usar el conocido polinomio de diferencias finitas o de Newton y para estimar el error se incluye el término del error del polinomio de interpolación:

$$f(x) = p_n(s) + E_n(s), \qquad s = \frac{x - x_0}{h}$$

$$p_n(s) = f_0 + \Delta f_0 s + \frac{1}{2!}\Delta^2 f_0\, s(s-1) + \frac{1}{3!}\Delta^3 f_0\, s(s-1)(s-2) + \ldots + \frac{1}{n!}\Delta^n f_0 s(s-1)(s-2)\ldots(s-n+1)$$

$$E_n(s) = \binom{s}{n+1} h^{n+1} f^{(n+1)}(z),\ x_0 < x < x_n$$

El uso de polinomios de diferente grado para aproximar a **f** genera diferentes fórmulas de integración.

7.1.1 Fórmula de los trapecios

Esta fórmula usa como aproximación para **f** un polinomio de primer grado, es decir una recta:

$$f(x) \cong p_1(s) = f_0 + \Delta f_0\, s$$

En general, la aproximación mediante una sola recta en el intervalo **[a, b]** tendría poca precisión, por lo que conviene dividir el intervalo **[a, b]** en **m** sub-intervalos y colocar en cada uno, una recta cuyos extremos coinciden con **f(x)**.

La figura geométrica en cada intervalo es un trapecio. Sea $\mathbf{A_i}$ el área del trapecio **i** y sea $\mathbf{T_i}$ el error de truncamiento respectivo, es decir la diferencia entre el área debajo de **f(x)** y el área de cada trapecio **i, i=1, 2, 3, …, m**. El área se puede aproximar con:

$$A = \int_a^b f(x)dx \approx A_1 + A_2 + A_3 + \ldots + A_m = \sum_{i=1}^{m} A_i$$

Mientras que el error de truncamiento total será:

$$T = T_1 + T_2 + T_3 + \ldots + T_m$$

Entonces si no hay puntos singulares ni discontinuidades en el intervalo **[a, b]** , es claro que

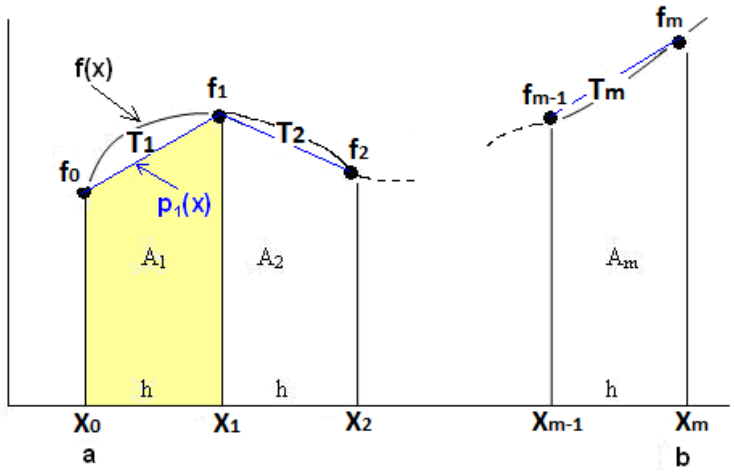
$$m \to \infty \Rightarrow T \to 0 \Rightarrow \sum_{i=1}^{m} A_i \to A$$

Por simplicidad se usarán puntos regularmente espaciados a una distancia h

$$A = \int_a^b f(x)dx \approx \int_{x_0}^{x_1} p_1(s)dx + \int_{x_1}^{x_2} p_1(s)dx + \ldots. + \int_{x_{m-1}}^{x_m} p_1(s)dx$$

$$A \approx A_1 + A_2 + \ldots + A_m$$

m: cantidad de franjas espaciadas regularmente en **h,** siendo $h = \dfrac{b-a}{m}$



Aproximación del integral con el área de trapecios

Para obtener la fórmula es suficiente encontrar el valor del área de un trapecio y luego extender este resultado a las demás, como se indica a continuación.

Área del primer trapecio:

$$A_1 = \int_{x_0}^{x_1} p_1(s)dx = \int_{x_0}^{x_1} (f_0 + \Delta f_0 s)dx$$

Mediante las sustituciones:

$s = (x - x_0)/h$

$x = x_0 \Rightarrow S = 0$

$x = x_1 \Rightarrow S = 1$

$dx = h\, ds$

$$A_1 = \int_0^1 (f_0 + \Delta f_0 s)h\,ds = h\left[\, f_0 s + \frac{1}{2}\Delta f_0 s^2 \,\right]_0^1 = h\left[f_0 + \frac{1}{2}(f_1 - f_0)\right]$$

$A_1 = \dfrac{h}{2}[f_0 + f_1]$, es la conocida fórmula de la geometría para el área de un trapecio

El resultado anterior se extiende directamente a los restantes intervalos:

$$A \approx A_1 + A_2 + \ldots + A_m = \frac{h}{2}[f_0 + f_1] + \frac{h}{2}[f_1 + f_2] + \ldots + \frac{h}{2}[f_{m-1} + f_m]$$

$$A \approx \frac{h}{2}[f_0 + 2f_1 + 2f_2 + \ldots + 2f_{m-1} + f_m]$$

Definición: Fórmula de los trapecios

$$A \approx \frac{h}{2}[f_0 + 2f_1 + 2f_2 + \ldots + 2f_{m-1} + f_m] = \frac{h}{2}[f_0 + 2\sum_{i=1}^{m-1} f_i + f_m],$$

m es la cantidad de trapecios.

Ejemplo. *La siguiente función no es integrable analíticamente.* $f(x) = \sqrt{x}\ \text{sen}(x)$, $0 \le x \le 2$

Use la fórmula de los trapecios con ***m = 4****, para obtener una respuesta aproximada del integral:*

$$A = \int_0^2 f(x)dx$$

$$A \approx \frac{h}{2}[f_0 + 2f_1 + 2f_2 + 2f_3 + f_4], \quad h = \frac{b-a}{m} = \frac{2-0}{4} = 0.5$$

$$= \frac{0.5}{2}[f(0) + 2f(0.5) + 2f(1) + 2f(1.5) + f(2)] = 1.5225$$

Sin embargo, es necesario poder contestar una pregunta fundamental: ¿Cuál es la precisión del resultado calculado?

7.1.2 Error de truncamiento en la fórmula de los trapecios

Es necesario estimar el error en el resultado obtenido con los métodos numéricos

Error al aproximar **f(x)** mediante $p_1(x)$ en el primer sub-intervalo:

$$E_1(s) = \binom{s}{2} h^2 f^{(2)}(z_1),\ x_0 < z_1 < x_1$$

Cálculo del área correspondiente al error en el uso del polinomio para aproximar a **f**

$$T_1 = \int_{x_0}^{x_1} E_1(s)dx = \int_{x_0}^{x_1} \binom{s}{2} h^2 f^{(2)}(z_1)dx = \int_{x_0}^{x_1} \frac{1}{2} s(s-1) h^2 f^{(2)}(z_1)dx$$

Usando las sustituciones anteriores:

$s = (x - x_0)/h$

$x = x_0 \Rightarrow S = 0$

$x = x_1 \Rightarrow S = 1$

$dx = h\,ds$

$$T_1 = \frac{h^3}{2}\int_0^1 s(s-1)f''(z_1)ds\ ,$$

Con el teorema del valor medio para integrales, puesto que **s(s-1),** no cambia de signo en el intervalo **[0,1]**, se puede sacar del integral la función **f''** evaluada en algún punto z_1 desconocido, en el mismo intervalo.

$$T_1 = \frac{h^3}{2} f''(z_1)\int_0^1 s(s-1)ds\ ,\ x_0 < z_1 < x_1$$

Luego de integrar, se obtiene

$$T_1 = -\frac{h^3}{12} f''(z_1),\ x_0 < z_1 < x_1$$

Este resultado se extiende a los **m** sub-intervalos en la integración

$T = T_1 + T_2 + \ldots + T_m$

$$T = -\frac{h^3}{12} f''(z_1) - \frac{h^3}{12} f''(z_2) - \ldots - \frac{h^3}{12} f''(z_m)$$

$$T = -\frac{h^3}{12} [f''(z_1) + f''(z_2) + \ldots + f''(z_m)]$$

$$T = -\frac{h^3}{12} m f''(z),$$ siendo **z** algún valor en el intervalo **(a, b)**

Mediante la sustitución $h = \frac{b-a}{m}$, la fórmula se puede expresar de la siguiente forma

Definición. Fórmula del error de truncamiento en la fórmula de los trapecios

$$T = -\frac{h^2}{12} (b - a) f''(z), \quad a \le z \le b$$

Esta fórmula se utiliza para acotar el error de truncamiento.

Siendo **z** desconocido, para acotar el error se puede usar un criterio conservador tomando el mayor valor de $|f''(z)|, a \le z \le b$. Este criterio no proporciona una medida muy precisa para el error y su aplicación puede ser un problema más complicado que la misma integración, por lo cual se puede intentar usar como criterio para estimar el error, la definición de convergencia indicada al inicio de esta sección, siempre que **f** sea una función integrable:

$$m \to \infty \Rightarrow \sum_{i=1}^{m} A_i \to A$$

Sean $A_m = \sum_{i=1}^{m} A_i$, $A_{m'} = \sum_{i=1}^{m'} A_i$ dos aproximaciones sucesivas con **m** y m' trapecios, $m' > m$

Entonces, se puede estimar el error de truncamiento absoluto del resultado con:

$$T \approx |A_m - A_{m'}|$$

Mientras que el error de truncamiento relativo se puede estimar con:

$$t \approx \frac{|A_m - A_{m'}|}{|A_{m'}|}$$

Hay que tener la precaución de no usar valores muy grandes para **m** por el efecto del error de redondeo acumulado en las operaciones aritméticas y que pudiera reducir la precisión en el resultado.

En caso de conocer únicamente puntos de **f**, al no disponer de más información para estimar el error de truncamiento, un criterio simple puede ser tomar el mayor valor de las segundas diferencias finitas como una aproximación para la segunda derivada en la fórmula del error, siempre que no cambien significativamente:

$$T = -\frac{h^2}{12} (b - a) f''(z), \quad a \le z \le b$$

$$f''(z) \approx \frac{\Delta^2 f_i}{h^2}$$

$$T \le \left|-\frac{(b-a)}{12}\right| \max(|\Delta^2 f_i|)$$

Esta fórmula también pudiera usarse para estimar el error de truncamiento en el caso de que **f(x)** se conozca explícitamente y **m** haya sido especificado. Habría que tabular las diferencias finitas para los puntos usados en la integración numérica y estimar el error con la fórmula anterior.

Ejemplo. *Estime cuantos trapecios deben usarse para integrar* ***f(x) = sen(x)*** *en el intervalo* ***[0,2]*** *de tal manera que la respuesta tenga el error absoluto menor a 0.0001*

Se requiere que error de truncamiento cumpla la condición:

$$|T| < 0.0001$$

$$\left| -\frac{h^2}{12}(b-a)\, f''(z) \right| < 0.0001$$

Siendo el valor de ***z*** *desconocido se debe usar el máximo valor de* ***f''(z) = -sen(z), 0<z<2***

$$\max |f''(z)| = 1$$

$$\left| -\frac{h^2}{12}(2-0)(1) \right| < 0.0001$$

De donde

$$h^2 < 0.0006$$

$$h < 0.0245$$

$$\frac{b-a}{m} < 0.0245$$

Entonces

$$m > (2-0)/0.0245$$

$$m > 81.63 \Rightarrow m = 82 \text{ trapecios}$$

Ejemplo. *Estime cuantos trapecios deben usarse para integrar* $f(x) = \sqrt{x}\ \text{sen}(x)$ *en el intervalo* ***[0,2]*** *de tal manera que la respuesta tenga el error absoluto menor a 0.0001*

Se requiere que error de truncamiento cumpla la condición:

$$|T| < 0.0001$$

$$\left| -\frac{h^2}{12}(b-a)\, f''(z) \right| < 0.0001$$

Siendo el valor de ***z*** *desconocido se debe usar el máximo valor de*

$$f''(z) = \frac{\cos(z)}{\sqrt{z}} - \sqrt{z}\,\text{sen}(z) - \frac{\text{sen}(z)}{4\sqrt{z^3}}, \quad 0 < z < 2$$

Problema demasiado complicado para estimar el mayor valor de la derivada y acotar el error.

En esta situación, se puede estimar el error comparando resultados con valores sucesivos de ***m*** *hasta que la diferencia sea suficientemente pequeña. Para los cálculos conviene instrumentar una función en MATLAB.*

En este ejemplo no se pueden tabular las diferencias finitas para estimar ***f''(x)*** *con* $\Delta^2 f(x)$ *pues* ***m*** *no está especificado*

Ejemplo. *Estime el error de truncamiento en la integración de* $f(x) = \sqrt{x}\ \text{sen}(x)$, ***0≤x≤2*** *con la fórmula de los trapecios con* ***m = 4***

En este caso, se tabulan los cinco puntos de f(x) para estimar el error, aproximando la derivada con la diferencia finita respectiva:

x	f	Δf	Δ^2f
0.0	0.0000	0.3390	0.1635
0.5	0.3390	0.5025	-0.1223
1.0	0.8415	0.3802	-0.3159
1.5	1.2217	0.0643	
2.0	1.2859		

$$T \le \left| -\frac{(b-a)}{12} \right| \max(|\Delta^2 f_i|) = \frac{(2-0)}{12}(0.3159) = 0.0527$$

Ejemplo. *Dados los puntos de una función* ***f: (0.1, 1.8), (0.2, 2.6), (0.3, 3.0), (0.4, 2.8), (0.5, 1.9)*** *Calcule el área* ***A*** *debajo de* ***f*** *aproximando mediante cuatro trapecios y estime el error en el resultado obtenido.*

$$A = \frac{0.1}{2}[1.8 + 2(2.6) + 2(3.0) + 2(2.8) + 1.9] = 1.0250$$

Al no disponer de más información, se usarán las diferencias finitas para estimar el error

x	f	Δf	$\Delta^2 f$
0.1	1.8	0.8	-0.4
0.2	2.6	0.4	-0.6
0.3	3.0	-0.2	-0.7
0.4	2.8	-0.9	
0.5	1.9		

$$T \leq \left|-\frac{(b-a)}{12}\right| \max(|\Delta^2 f_i|) = \frac{(0.5-0.1)}{12}(0.7) = 0.0233$$

Esto indicaría que solamente podemos tener confianza en el primer decimal

7.1.3 Instrumentación computacional de la fórmula de los trapecios

Si se quiere integrar debajo de una función dada en forma explícita, conviene definir una función de MATLAB para evaluar el integral dejando como dato el número de trapecios m. Los resultados calculados pueden usarse como criterio para estimar el error de truncamiento.

En la siguiente instrumentación debe suministrarse la función **f (**definida en forma simbólica y en formato **inline**), el intervalo de integración **a, b** y la cantidad de franjas o trapecios **m**

```
function r = trapecios(f, a, b, m)
h=(b-a)/m;
s=0;
for i=1: m - 1
   s=s + f(a + i*h);
end
r = h/2*(f(a) + 2*s + f(b));
```

Ejemplo. *Probar la función esta función para integral* ***f(x)=sen(x), 0≤x≤2***

```
>> syms x;
>> f = sin(x);
>> t = trapecios(inline(f), 0, 2, 5)
t =
  1.397214
```

Se puede probar con más trapecios para mejorar la aproximación

```
>> t = trapecios(inline(f), 0, 2, 50)
t =
  1.415958
>> t = trapecios(inline(f), 0, 2, 500)
t =
  1.416144
```

Compare con el valor exacto que proporciona MATLAB

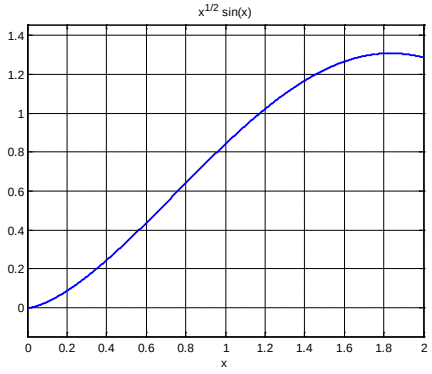
```
>> t = eval(int(f,0,2))
t =
  1.416146…
```

Los resultados tienden hacia el valor exacto a medida que se incrementa el número de trapecios. Estos resultados pueden usarse como criterio para determinar la precisión de la aproximación.

Ejemplo. *La siguiente función no es integrable analíticamente:* $S = \int_0^2 \sqrt{x}\ sen(x)dx$

Use la fórmula de los trapecios para obtener la respuesta aproximada y estimar el error.

```
>> syms x
>> f=sqrt(x)*sin(x);
>> ezplot(f,[0,2]),grid on
>> r=trapecios(inline(f),0,2,10)
r =
   1.5313
>> r=trapecios(inline(f),0,2,20)
r =
   1.5323
>> r=trapecios(inline(f),0,2,40)
r =
   1.5326
>> r=trapecios(inline(f),0,2,50)
r =
   1.5326
```



El último resultado tiene cuatro decimales que no cambian y se pueden considerar correctos.

7.1.4 Fórmula de Simpson

Esta fórmula usa como aproximación para **f** un polinomio de segundo grado, o parábola:

$$f(x) \cong p_2(s) = f_0 + \Delta f_0\, s + \frac{1}{2}\Delta^2 f_0\, s(s-1)$$

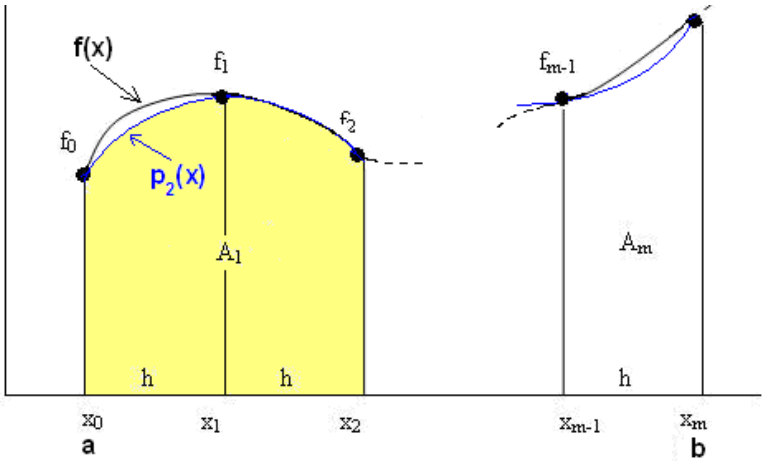
La integración se realiza dividiendo el intervalo de integración **[a, b]**, en subintervalos, incluyendo tres puntos en cada uno para colocar una parábola.

Por simplicidad se usarán puntos regularmente espaciados a una distancia **h**

$$A = \int_a^b f(x)dx \cong \int_{x_0}^{x_2} p_2(s)dx + \int_{x_2}^{x_4} p_2(s)dx + \ldots + \int_{x_{m-2}}^{x_m} p_2(s)dx$$

$$A \cong A_1 + A_2 + \ldots + A_{m/2}$$

El área de dos intervalos consecutivos es aproximada mediante el área debajo de parábolas. Los puntos son numerados $x_0, x_1, \ldots, x_m$, Siendo **m** debe ser un número par. Así, la cantidad de parábolas es **m/2**.



h=(b-a)/m

Para obtener la fórmula se debe encontrar el valor del área para una parábola:

$$A_1 = \int_{x_0}^{x_2} p_2(s)dx = \int_{x_0}^{x_2} [f_0 + \Delta f_s s + \frac{1}{2}\Delta^2 f_0 s(s-1)]\, dx$$

Mediante las sustituciones:

$s = (x - x_0)/h$

$x = x_0 \Rightarrow s = 0$

$x = x_2 \Rightarrow s = 2$

$dx = h\, ds$

$$A_1 = \int_0^2 (f_0 + \Delta f_0 s + \frac{1}{2}\Delta^2 f_0 s(s-1)h ds$$

Luego de integrar, sustituir las diferencias finitas y simplificar se tiene

$A_1 = \frac{h}{3}[f_0 + 4f_1 + f_2]$ es el área debajo de la parábola en la primera franja

Por lo tanto, habiendo **m/2** franjas, el área total es la suma:

$A = A_1 + A_2 + \ldots + A_{m/2}$

Después de sustituir y simplificar se obtiene la fórmula de integración

Definición. Fórmula de Simpson (fórmula de las parábolas)

$$A = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots\ldots + 2f_{m-2} + 4f_{m-1} + f_m]$$

m es un parámetro para la fórmula (debe ser un número par)

7.1.5 Error de truncamiento en la fórmula de Simpson

Del análisis del error se obtiene

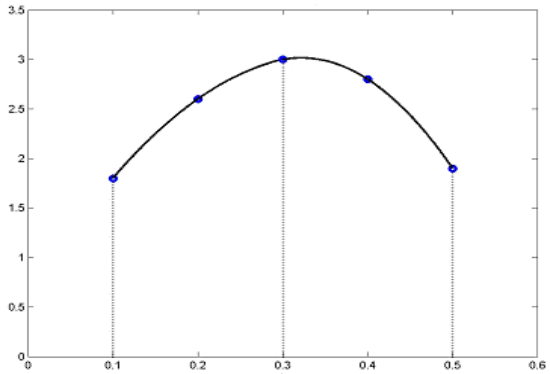
$$T = -\frac{h^4}{180}(b-a)\, f^{(4)}(z), \quad a<z<b$$

Esta fórmula se puede usar para acotar el error de truncamiento. Bajo ciertas consideraciones y si se conoce m, se puede estimar la derivada con la diferencia finita correspondiente:

$$f^{(4)}(z) \cong \frac{\Delta^4 f_i}{h^4}, \qquad T \cong -\frac{(b-a)}{180} \max|\Delta^4 f_i|$$

Ejemplo

*Dados los puntos de una función **f: (0.1, 1.8), (0.2, 2.6), (0.3, 3.0), (0.4, 2.8), (0.5, 1.9)***
*Calcule el área debajo de **f** mediante una aproximación con parábolas.*



Aproximación del área mediante parábolas

La suma del área debajo de las dos parábolas, con la fórmula anterior:

$$A = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + f_4]$$

$$A = \frac{0.1}{3}(1.8 + 4(2.6) + 2(3.0) + 4(2.8) + 1.9) \quad = \; 1.0433$$

Este resultado es aproximadamente igual al área debajo de ***f.***

Estimar el error en el resultado obtenido

Al no disponer de más información, se usarán las diferencias finitas para estimar el error

x	f	Δf	Δ^2f	Δ^3f	Δ^4f
0.1	1.8	0.8	-0.4	-0.2	0.1
0.2	2.6	0.4	-0.6	-0.1	
0.3	3.0	-0.2	-0.7		
0.4	2.8	-0.9			
0.5	1.9				

$$T = -\frac{h^4}{180}(b-a)\, f^{(4)}(z), \quad a<z<b$$

$$T \le \left|-\frac{(b-a)}{180}\right| max|\Delta^4 f_i| = \frac{(0.5-0.1)}{180}(0.1) = 0.00022$$

Esto indicaría que podemos tener confianza en la respuesta hasta el tercer decimal. Resultado mejor que el obtenido con la Regla de los Trapecios

Ejemplo. *Si* ***f*** *es una función diferenciable en el intervalo* ***[a,b]****, la longitud del arco de la curva* ***f(x)*** *en ese intervalo se puede calcular con el integral* $S = \int_a^b \sqrt{1+[f'(x)]^2}\,dx$

Calcular la longitud del arco de la curva ***f(x)=sen(x), x∈[0, 2]*** *usando 2 parábolas* ***(m = 4)***

Longitud del arco: $S = \int_a^b \sqrt{1+[f'(x)]^2}\,dx$

$s = \int_0^2 g(x)dx = \int_0^2 \sqrt{1+\cos^2(x)}\,dx$ *(no se puede evaluar analíticamente)*

$$h = \frac{b-a}{m} = \frac{2-0}{4} = 0.5$$

$$S = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + f_4]$$

$$S = \frac{0.5}{3}[g(0) + 4g(0.5) + 2g(1) + 4g(1.5) + g(2)]$$

$$S = 2.3504$$

Estimar el error en el resultado anterior

Dado que ***m*** *está especificado, se usa una aproximación de diferencias finitas para la derivada en la fórmula del error de truncamiento*

$$T = -\frac{h^4}{180}(b-a)\, f^{(4)}(z), \quad a<z<b$$

x	*f*	*Δf*	*Δ^2f*	*Δ^3f*	*Δ^4f*
0.0	*1.4142*	*-0.0837*	*-0.1101*	*0.1698*	*-0.0148*
0.5	*1.3305*	*-0.1938*	*0.0597*	*0.1551*	
1.0	*1.1366*	*-0.1341*	*0.2148*		
1.5	*1.0025*	*0.0806*			
2.0	*1.0831*				

$$T \le \left|-\frac{(b-a)}{180}\right| max(\Delta^4 f_i) = \frac{(2-0)}{180}(0.0148) = 0.00016$$

Es una cota aproximada para el error de truncamiento

7.1.6 Instrumentación computacional de la fórmula de Simpson

La función recibirá a la función **f** definida en forma simbólica, el intervalo de integración **a, b** y la cantidad de franjas **m**

```
function r = simpson(f, a, b, m)
h=(b-a)/m;
s=0;
for i=1:m-1
  s=s+2*(mod(i,2)+1)*f(a+i*h);        %Sumar los términos con coeficientes 4, 2, 4, 2,...,4
end
r=h/3*(f(a) + s + f(b));
```

Ejemplo. *Integrar* $f(x)=\sqrt{1+\cos^2(x)}$ $0\le x\le 2$, *con la fórmula de Simpson iterativamente hasta que el error de truncamiento sea menor que* ***0.0001***

```
>> syms x
>> f = sqrt(1+(cos(x))^2);
>> r=simpson(inline(f),0,2,4)
r =
   2.3504
>> r=simpson(inline(f),0,2,8)
r =
   2.3516
>> r=simpson(inline(f),0,2,12)
r =
   2.3517
>> r=simpson(inline(f),0,2,16)
r =
   2.3517
```

7.1.7 Error de truncamiento vs. Error de redondeo

En las fórmulas de integración numérica se observa que el error de truncamiento depende de **h**

Fórmula de los Trapecios: $T = -\frac{h^2}{12}(b-a)\,f''(z) = O(h^2)$

Fórmula de Simpson: $T = -\frac{h^4}{180}(b-a)\,f^{(4)}(z) = O(h^4)$

Además $h = \frac{b-a}{m}$

Entonces, para ambas fórmulas, cuando $h \to 0 \Rightarrow T \to 0$

Está claro que la fórmula de Simpson converge más rápido, supuesto que **h<1**

Por otra parte, al evaluar cada operación aritmética se puede introducir un error de redondeo R_i si no se conservan todos los dígitos decimales en los cálculos numéricos. La suma de estos errores es el error de redondeo acumulado **R**. Mientras más sumas se realicen, mayor es la cantidad de términos que acumulan error de redondeo.

$$R = R_1 + R_2 + R_3 + \ldots + R_m$$

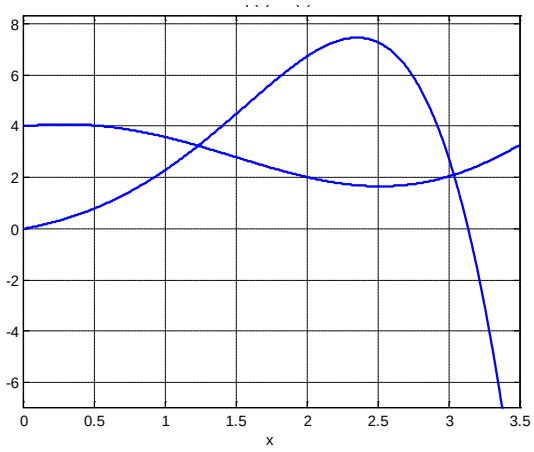
Estos errores de redondeo pueden tener signos diferentes y anularse, pero también puede ocurrir que tengan igual signo, por lo tanto el valor puede crecer

Siendo $m = \frac{b-a}{h}$, cuando $h \to 0 \Rightarrow m \to \infty \Rightarrow$ **R** puede crecer

En conclusión, para prevenir el crecimiento de **R**, es preferible usar fórmulas que no requieran que **m** sea muy grande para obtener el resultado con una precisión requerida. Este es el motivo para preferir la fórmula de Simpson sobre la fórmula de los Trapecios.

Ejemplo. *Encontrar el área entre* ***f(x) = 4 + cos(x+1)****, y* ***g(x)=e^x sen(x)****, que incluya el área entre las intersecciones de* ***f*** *y* ***g*** *en el primer cuadrante. Use la Regla de Simpson,* ***m=10.***

```
>> syms x
>> f=4+x*cos(x+1);
>> g=exp(x)*sin(x);
>> ezplot(f,[0,3.5]),grid on
>> hold on
>> ezplot(g,[0,3.5])
```



Se utiliza un método numérico para calcular las intersecciones:

```
>> h=f - g;
>> a=biseccion(inline(h),1,1.5,0.0001)
a =
   1.2337
>> b=biseccion(inline(h),3,3.2,0.0001)
b =
   3.0407
```

Finalmente se integra:

```
>> s=simpson(inline(h),a,b,10)
s =
    6.5391
```

Comparación con el valor que proporciona la function ***int*** *de MATLAB*

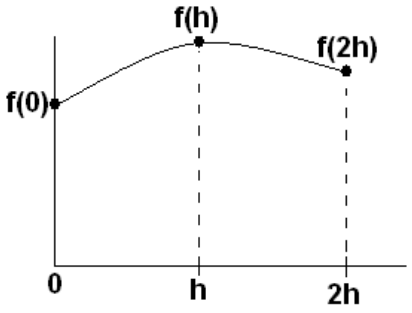
```
>> r=eval(int(h,a,b))
r =
    6.5393
```

7.2 Obtención de fórmulas de integración numérica con el método de coeficientes indeterminados

En esta sección se describe una técnica denominada de los Coeficientes Indeterminados para obtener fórmulas de integración numérica.

El procedimiento consiste en proponer una fórmula conteniendo algunas incógnitas. Esta fórmula es aplicada a casos conocidos con el propósito de obtener ecuaciones, de las cuales se determinan finalmente los valores para las incógnitas.

Como ejemplo se usa este método para obtener una fórmula de tres puntos espaciados en h:



Fórmula propuesta

$$A = \int_0^{2h} f(x)dx = c_0 f(0) + c_1 f(h) + c_2 f(2h)$$

Deben determinarse los coeficientes c_0, c_1, c_2. Para obtenerlos, se usarán tres casos con polinomios de grado 0, 1 y 2 con los cuales queremos que se cumpla la fórmula. Es suficiente considerar la forma más simple de cada caso:

1) f(x) = 1,

$$A = \int_0^{2h} (1)dx = 2h = c_0 f(0) + c_1 f(h) + c_2 f(2h) = c_0(1) + c_1(1) + c_2(1) \Rightarrow c_0 + c_1 + c_2 = 2h$$

2) f(x) = x,

$$A = \int_0^{2h} x dx = 2h^2 = c_0 f(0) + c_1 f(h) + c_2 f(2h) = c_0(0) + c_1(h) + c_2(2h) \Rightarrow c_1 + 2c_2 = 2h$$

3) f(x) = x^2,

$$A = \int_0^{2h} x^2 dx = \frac{8}{3}h^3 = c_0 f(0) + c_1 f(h) + c_2 f(2h) = c_0(0) + c_1(h^2) + c_2(4h^2) \Rightarrow c_1 + 4c_2 = \frac{8}{3}h$$

Resolviendo las tres ecuaciones resultantes se obtienen: $c_0 = \frac{h}{3}$, $c_1 = \frac{4h}{3}$, $c_2 = \frac{h}{3}$

Reemplazando en la fórmula propuesta se llega a la conocida fórmula de Simpson

$$A = \frac{h}{3}(f(0) + 4f(h) + f(2h))$$

La obtención de la fórmula implica que es exacta si **f** es un polinomio de grado menor o igual a dos. Para otra **f**, será una aproximación equivalente a sustituir **f** por un polinomio de grado dos.

7.3 Cuadratura de Gauss

Las fórmulas de Newton-Cotes estudiadas utilizan polinomios de interpolación construidos con puntos fijos equidistantes. Estas fórmulas son exactas si la función es un polinomio de grado menor o igual al polinomio de interpolación respectivo.

Si se elimina la restricción de que los puntos sean fijos y equidistantes, entonces las fórmulas de integración contendrán incógnitas adicionales.

La cuadratura de Gauss propone una fórmula general en la que los puntos incluidos no son fijos como en las fórmulas de Newton-Cotes:

$$A = \int_a^b f(x)dx = c_0 f(t_0) + c_1 f(t_1) + \ldots + c_m f(t_m)$$

Los puntos $t_0, t_1, \ldots, t_m$, son desconocidos. Adicionalmente también deben determinarse los coeficientes $c_0, c_1, \ldots, c_m$

El caso simple es la fórmula de dos puntos. Se usa el método de los coeficientes indeterminados para determinar las cuatro incógnitas

7.3.1 Fórmula de la cuadratura de Gauss con dos puntos

Fórmula propuesta

$$A = \int_a^b f(x)dx = c_0 f(t_0) + c_1 f(t_1)$$

Por simplicidad se usará el intervalo **[-1, 1]** para integrar. Mediante una sustitución será extendido al caso general:

$$A = \int_{-1}^{1} f(t)dt = c_0 f(t_0) + c_1 f(t_1)$$

Habiendo cuatro incógnitas se tomarán cuatro casos en los que la fórmula sea exacta. Se usarán polinomios de grado 0, 1, 2, 3. Es suficiente considerarlos en su forma más simple:

1) f(t)=1, $A = \int_{-1}^{1} (1)dt = 2 = c_0 f(t_0) + c_1 f(t_1) = c_0(1) + c_1(1) \Rightarrow 2 = c_0 + c_1$

2) f(t)=t, $A = \int_{-1}^{1} t dt = 0 = c_0 f(t_0) + c_1 f(t_1) = c_0 t_0 + c_1 t_1 \Rightarrow 0 = c_0 t_0 + c_1 t_1$

3) f(t)=t^2, $A = \int_{-1}^{1} t^2 dt = \frac{2}{3} = c_0 f(t_0) + c_1 f(t_1) = c_0 t_0^2 + c_1 t_1^2 \Rightarrow \frac{2}{3} = c_0 t_0^2 + c_1 t_1^2$

4) f(t)=t^3, $A = \int_{-1}^{1} t^3 dt = 0 = c_0 f(t_0) + c_1 f(t_1) = c_0 t_0^3 + c_1 t_1^3 \Rightarrow 0 = c_0 t_0^3 + c_1 t_1^3$

Se genera un sistema de cuatro ecuaciones no-lineales. Una solución para este sistema se obtiene con facilidad mediante simple sustitución:

Los valores $c_0 = c_1 = 1$ satisface a la ecuación **1)**.

De la ecuación **2)** se tiene $t_0 = -t_1$. Esto satisface también a la ecuación **4)**.

Finalmente, sustituyendo en la ecuación **3):** $2/3 = (1)(-t_1)^2 + (1)(t_1)^2$ se obtiene:

$t_1 = \frac{1}{\sqrt{3}}$, entonces, $t_0 = -\frac{1}{\sqrt{3}}$ y se reemplazan en la fórmula propuesta:

Definición: **Fórmula de cuadratura de Gauss con dos puntos**

$$A=\int_{-1}^{1} f(t)dt = c_0 f(t_0)+c_1 f(t_1) = f(-\frac{1}{\sqrt{3}})+f(\frac{1}{\sqrt{3}})$$

Esta simple fórmula es exacta si **f** es un polinomio de grado menor o igual a tres. Para otra **f** es una aproximación equivalente a sustituir **f** con un polinomio de grado tres.

Ejemplo. *Calcule* $A=\int_{-1}^{1}(2t^3+t^2-1)dt$

$$A=\int_{-1}^{1} f(t)dt = f(-\frac{1}{\sqrt{3}})+f(\frac{1}{\sqrt{3}}) = [2(-\frac{1}{\sqrt{3}})^3+(-\frac{1}{\sqrt{3}})^2-1]+[2(\frac{1}{\sqrt{3}})^3+(\frac{1}{\sqrt{3}})^2-1] = -4/3$$

La respuesta es exacta pues ***f*** *es un polinomio de grado 3*

Mediante un cambio de variable se extiende la fórmula al caso general:

$$A=\int_{a}^{b} f(x)dx = c_0 f(t_0)+c_1 f(t_1)$$

Sea $x=\frac{b-a}{2}t+\frac{b+a}{2}$

Se tiene que $t = 1 \Rightarrow x = b$, $t = -1 \Rightarrow x = a$, $dx=\frac{b-a}{2}dt$

Sustituyendo se tiene

Definición: **Fórmula general de Cuadratura de Gauss para dos puntos**

$$A=\int_{a}^{b} f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f(\frac{b-a}{2}t+\frac{b+a}{2})dt = \frac{b-a}{2}\left[f(-\frac{1}{\sqrt{3}})+f(\frac{1}{\sqrt{3}})\right]$$

Ejemplo. *Calcule* $A=\int_{1}^{2} xe^x dx$ *con la fórmula de la Cuadratura de Gauss con dos puntos*

$$x=\frac{b-a}{2}t+\frac{b+a}{2}=\frac{2-1}{2}t+\frac{2+1}{2}=\frac{1}{2}t+\frac{3}{2}$$

$$A=\int_{a}^{b} f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f(\frac{b-a}{2}t+\frac{b+a}{2})dt$$

$$=\frac{1}{2}\int_{-1}^{1} f(\frac{1}{2}t+\frac{3}{2})dt = \frac{1}{2}\int_{-1}^{1}(\frac{1}{2}t+\frac{3}{2})e^{\frac{1}{2}t+\frac{3}{2}}dt$$

$$=\frac{1}{2}[f(-\frac{1}{\sqrt{3}})+f(\frac{1}{\sqrt{3}})]$$

$$=\frac{1}{2}\left[(-\frac{1}{2\sqrt{3}}+\frac{3}{2})e^{-\frac{1}{2\sqrt{3}}+\frac{3}{2}})+(\frac{1}{2\sqrt{3}}+\frac{3}{2})e^{\frac{1}{2\sqrt{3}}+\frac{3}{2}})\right] = 7.3832$$

La respuesta exacta con seis decimales es ***7.389056.***
Se observa que usando únicamente dos puntos se tiene una precisión mejor que usando la fórmula de Simpson con tres puntos.

7.3.2 Instrumentación computacional de la cuadratura de Gauss

```
function s = cgauss(f, a, b)
t1=-(b-a)/2*1/sqrt(3)+(b+a)/2;
t2= (b-a)/2*1/sqrt(3)+(b+a)/2;
s = (b-a)/2*(f(t1) + f(t2));
```

Ejemplo. *Use la función cgauss para calcular* $A = \int_1^2 xe^x dx$

```
>> syms x
>> f=x*exp(x);
>> s=cgauss(inline(f),1,2)
     s = 7.3832
```

Para mejorar la precisión de ésta fórmula se la puede aplicar más de una vez dividiendo el intervalo de integración en sub-intervalos.

Ejemplo. *Aplique dos veces la cuadratura de Gauss en el ejemplo anterior*

$$A = \int_1^2 xe^x dx = A_1 + A_2 = \int_1^{1.5} xe^x dx + \int_{1.5}^2 xe^x dx$$

En cada subintervalo se aplica la fórmula de la Cuadratura de Gauss:

```
>> syms x
>> f=x*exp(x);
>> s=cgauss(inline(f),1,1.5) + cgauss(inline(f),1.5,2)
     s  =  7.3886
```

Se puede dividir el intervalo en más sub-intervalos para obtener mayor precisión. Conviene definir una función en MATLAB para determinar la precisión del resultado, comparando valores consecutivos, en base a la convergencia del integral.

7.3.3 Instrumentación extendida de la cuadratura de Gauss

```
function t=cgaussm(f, a, b, m)
h=(b-a)/m;
t=0;
x=a;
for i=1:m
   a=x+(i-1)*h;
   b=x+i*h;
   s=cgauss(f,a,b);
   t=t+s;
end
```

m es la cantidad de sub-intervalos

Ejemplo. *Aplicar sucesivamente la Cuadratura de Gauss incrementando el número de sub-intervalos, hasta que la respuesta tenga cuatro decimales exactos*

```
>> syms x
>> f=x*exp(x);
>> s=cgaussm(inline(f),1,2,1)
s =
    7.3833
>> s=cgaussm(inline(f),1,2,2)
s =
    7.3887
>> s=cgaussm(inline(f),1,2,3)
s =
    7.3890
>> s=cgaussm(inline(f),1,2,4)
s =
    7.3890
```

En el último cálculo se han usado 4 sub-intervalos. El valor obtenido tiene cuatro decimales fijos

Para obtener fórmulas de cuadratura de Gauss con más puntos no es práctico usar el método de coeficientes indeterminados. Se puede usar un procedimiento general basado en la teoría de polinomios ortogonales. En la bibliografía se pueden encontrar estas fórmulas así como expresiones para estimar el error de truncamiento pero imprácticas para su uso.

7.4 Integrales con límites infinitos

Estos integrales se denominan integrales impropios del primer tipo

Ocasionalmente puede ser de interés calcular integrales cuyos límites no se pueden evaluar en las fórmulas de integración. Mediante alguna sustitución deben reducirse a una forma simple eliminando estos límites impropios.

Ejemplo. *Calcule* $A = \int_0^{\infty} \frac{dx}{(1+x^2)^3}$ *con la Cuadratura de Gauss,* ***m = 1, 2, 4***

Antes de la sustitución conviene separar el integral en dos sub-intervalos

$$A = \int_0^{\infty} \frac{dx}{(1+x^2)^3} = \int_0^{1} \frac{dx}{(1+x^2)^3} + \int_1^{\infty} \frac{dx}{(1+x^2)^3} = A_1 + A_2$$

A_1 *se puede calcular inmediatamente con la Cuadratura de Gauss*

Para ***A_2*** *se hace la sustitución*

$x = 1/t$

$x \to \infty \Rightarrow t \to 0,\ \ x = 1 \Rightarrow t = 1,\ \ dx = -1/t^2\, dt$

$$A_2 = \int_1^{\infty} \frac{dx}{(1+x^2)^3} = \int_1^{0} \frac{1}{(1+1/t^2)^3}\left(-\frac{dt}{t^2}\right) = \int_0^{1} \frac{t^4}{(1+t^2)^3}\, dt$$

Ahora se puede aplicar también la Cuadratura de Gauss

Resultados calculados:

m=1: $A = A_1 + A_2 = 0.6019$
m=2: $A = A_1 + A_2 = 0.5891$
m=4: $A = A_1 + A_2 = 0.5890$

El último resultado tiene un error en el orden de 0.0001

7.5 Integrales con singularidades

Estos integrales se denominan integrales impropios del segundo tipo

Mediante alguna sustitución deben reducirse a una forma eliminando los puntos singulares.

Ejemplo. *Calcule* $A=\int_0^1 \frac{2^x}{(x-1)^{2/5}}dx$ *con la fórmula de Simpson,* ***m=4*** *y estimar el error*

Mediante una sustitución adecuada se puede eliminar el punto singular

x-1 = u^5: $x = 1 \Rightarrow u = 0$

$x = 0 \Rightarrow u = -1$

$dx = 5u^4du$

$$A=\int_0^1 \frac{2^x}{(x-1)^{2/5}}dx=\int_{-1}^0 \frac{2^{u^5+1}}{u^2}(5u^4du)=\int_{-1}^0 5(2^{u^5+1}u^2)du=\int_{-1}^0 f(u)du$$

integral bien definido en ***[-1,0]***

Regla de Simpson, m=4

$$A=\frac{h}{3}[f_0+4f_1+2f_2+4f_3+f_4]$$

$m = 4 \Rightarrow h = 0.25$

$A = 0.25/3(f(-1)+4f(-0.75)+2f(-0.5)+4f(-0.25)+f(0)) = 2.6232$

Estimación del error

Al no disponer de más información, se usarán las diferencias finitas para estimar el error

x	*f*	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
-1	1.2500	-0.0570	-0.5243	0.6502	-0.4768
-0.75	1.1930	-0.5814	0.1259	0.1734	
-0.5	0.6116	-0.4555	0.2993		
-0.25	0.1561	-0.1561			
0	0				

$$T \le \left|-\frac{(b-a)}{180}\right| max(|\Delta^4 f_i|) = \frac{(0-(-1))}{180}(0.4768) = 0.0026$$

Nota.- *Este integral no puede evaluarse analíticamente. Si se usa la fórmula de Simpson incrementando el número de franjas, el resultado tiende a* ***2.6246019…*** *consistente con el error calculado con la fórmula anterior.*

Ejemplo. *Calcule* $A=\int_0^1 \frac{sen(x)}{x}dx$ *con la Cuadratura de Gauss con* ***m=1, 2***

La fórmula de la Cuadratura de Gauss no requiere evaluar ***f*** *en los extremos, por lo tanto se puede aplicar directamente:*

$$A=\int_a^b f(x)dx=\frac{b-a}{2}\left[f(-\frac{b-a}{2}\frac{1}{\sqrt{3}}+\frac{b+a}{2})+f(\frac{b-a}{2}\frac{1}{\sqrt{3}}+\frac{b+a}{2})\right]$$

Aplicando la fórmula en el intervalo [0, 1]:

A = 0.94604

Aplicando la fórmula una vez en cada uno de los intervalos [0, 0.5] y [0.5, 1] y sumando:

A = 0.94608

Comparando ambos resultados se puede estimar el error en el quinto decimal.

NOTA: *Para calcular el integral no se puede aplicar la fórmula de Simpson pues ésta requiere evaluar* ***f(x)*** *en los extremos. En este ejemplo, se tendría un resultado indeterminado al evaluar en* ***x = 0***

7.6 Integrales múltiples

Para evaluar integrales múltiples se pueden usar las reglas de integración numérica anteriores integrando separadamente con cada variable:

Suponer que se desea integrar $V = \int_c^d \int_a^b f(x,y)dxdy = \int_c^d \left(\int_a^b f(x,y)dx \right) dy$ con la regla de Simpson con **m** subintervalos en ambas direcciones.

Se puede aplicar la regla integrando en la dirección **X** fijando **Y** en cada punto**.** Los símbolos Δx y Δy denotan la distancia entre los puntos en las direcciones **X, Y,** respectivamente.

$$V \cong \frac{\Delta y}{3}\left[\int_a^b f(x,y_0)dx + 4\int_a^b f(x,y_1)dx + 2\int_a^b f(x,y_2)dx + 4\int_a^b f(x,y_3)dx + \ldots + \int_a^b f(x,y_m)dx \right]$$

$$\int_a^b f(x,y_0)dx \cong \frac{\Delta x}{3}\left(f(x_0,y_0) + 4f(x_1,y_0) + 2f(x_2,y_0) + 4f(x_3,y_0) + \ldots + f(x_m,y_0)\right)$$

$$\int_a^b f(x,y_1)dx \cong \frac{\Delta x}{3}\left(f(x_0,y_1) + 4f(x_1,y_1) + 2f(x_2,y_1) + 4f(x_3,y_1) + \ldots + f(x_m,y_1)\right)$$

$$\int_a^b f(x,y_2)dx \cong \frac{\Delta x}{3}\left(f(x_0,y_2) + 4f(x_1,y_2) + 2f(x_2,y_2) + 4f(x_3,y_2) + \ldots + f(x_m,y_2)\right)$$

$$\int_a^b f(x,y_3)dx \cong \frac{\Delta x}{3}\left(f(x_0,y_3) + 4f(x_1,y_3) + 2f(x_2,y_3) + 4f(x_3,y_3) + \ldots + f(x_m,y_3)\right)$$

. . .

$$\int_a^b f(x,y_m)dx \cong \frac{\Delta x}{3}\left(f(x_0,y_m) + 4f(x_1,y_m) + 2f(x_2,y_m) + 4f(x_3,y_m) + \ldots + f(x_m,y_m)\right)$$

Ejemplo. *Calcule el integral de* ***f(x,y)=sen(x+y), 0≤x≤1, 2≤y≤3****, con* ***m = 2*** *en cada dirección.*

$$V = \int_2^3 \int_0^1 \operatorname{sen}(x+y)dxdy = \int_2^3 \left(\int_0^1 \operatorname{sen}(x+y)dx \right) dy, \qquad \Delta x = \Delta y = 0.5$$

$$V \cong \frac{0.5}{3}\left[\int_0^1 \operatorname{sen}(x+2)dx + 4\int_0^1 \operatorname{sen}(x+2.5)dx + \int_0^1 \operatorname{sen}(x+3)dx \right]$$

$$\int_0^1 \operatorname{sen}(x+2)dx \cong \frac{0.5}{3}\left(\operatorname{sen}(0+2) + 4\operatorname{sen}(0.5+2) + \operatorname{sen}(1+2)\right) = 0.5741$$

$$\int_0^1 \operatorname{sen}(x+2.5)dx \cong \frac{0.5}{3}\left(\operatorname{sen}(0+2.5) + 4\operatorname{sen}(0.5+2.5) + \operatorname{sen}(1+2.5)\right) = 0.1354$$

$$\int_0^1 \operatorname{sen}(x+3)dx \cong \frac{0.5}{3}\left(\operatorname{sen}(0+3) + 4\operatorname{sen}(0.5+3) + \operatorname{sen}(1+3)\right) = -0.3365$$

$$V \cong \frac{0.5}{3}\left[0.5741 + 4(0.1354) + (-0.3365)\right] = 0.1299$$

Ejemplo. *Un lago tiene forma aproximadamente rectangular de 200m x 400m. Se ha trazado un cuadriculado y se ha medido la profundidad en metros en cada cuadrícula de la malla como se indica en la tabla siguiente:*

X / Y	0	100	200	300	400
0	**0**	**0**	**4**	**6**	**0**
50	**0**	**3**	**5**	**7**	**3**
100	**1**	**5**	**6**	**9**	**5**
150	**0**	**2**	**3**	**5**	**1**
200	**0**	**0**	**1**	**2**	**0**

Con todos los datos de la tabla estime el volumen aproximado de agua que contiene el lago. Utilice la ***fórmula de Simpson*** *en ambas direcciones.*

$$V = \int_c^d \int_a^b f(x,y)\,dx\,dy$$

$$V = \frac{\Delta x}{3}\left\{\frac{\Delta y}{3}\left[f(x_0,y_0) + 4f(x_0,y_1) + 2f(x_0,y_2) + 4f(x_0,y_3) + f(x_0,y_4)\right]\right.$$
$$+ 4\frac{\Delta y}{3}\left[f(x_1,y_0) + 4f(x_1,y_1) + 2f(x_1,y_2) + 4f(x_1,y_3) + f(x_1,y_3)\right]$$
$$+ 2\frac{\Delta y}{3}\left[f(x_2,y_0) + 4f(x_2,y_1) + 2f(x_2,y_2) + 4f(x_2,y_3) + f(x_2,y_4)\right]$$
$$+ 4\frac{\Delta y}{3}\left[f(x_3,y_0) + 4f(x_3,y_1) + 2f(x_3,y_2) + 4f(x_3,y_3) + f(x_3,y_4)\right]$$
$$\left. + \frac{\Delta y}{3}\left[f(x_4,y_0) + 4f(x_4,y_1) + 2f(x_4,y_2) + 4f(x_4,y_3) + f(x_4,y_4)\right]\right\}$$

$$V = \frac{100}{3}\left\{\frac{50}{3}\left[0 + 4(0) + 2(1) + 4(0) + 0\right]\right.$$
$$+ 4\frac{50}{3}\left[0 + 4(3) + 2(5) + 4(2) + 0\right]$$
$$+ 2\frac{50}{3}\left[4 + 4(5) + 2(6) + 4(3) + 1\right]$$
$$+ 4\frac{50}{3}\left[6 + 4(7) + 2(9) + 4(5) + 2\right]$$
$$\left. + \frac{50}{3}\left[0 + 4(3) + 2(5) + 4(1) + 0\right]\right\}$$

$$V = 287777.78 \quad \text{metros cúbicos de agua}$$

Ejemplo. *Calcule el integral de* $f(x,y)=(x^2+y^3)$, $0\le x\le 1$, $x\le y\le 2x$. *Use la fórmula de Simpson con* $m = 2$ *en cada dirección.*

$$V=\int_0^1\int_x^{2x}(x^2+y^3)dydx$$

Puntos para la integración numérica:

	y				
x		x	1.5x	2x	Δy
	0	0	0	0	0
	0.5	0.5	0.75	1	0.25
	1	1	1.5	2	0.5

Los límites y la distancia de los sub-intervalos para integrar en **Y** cambian dependiendo de **X**:

$$V=\frac{\Delta x}{3}\left\{\frac{\Delta y}{3}\left[f(x_0,y_0)+4f(x_0,y_1)+f(x_0,y_2)\right]\right.$$
$$+4\frac{\Delta y}{3}\left[f(x_1,y_0)+4f(x_1,y_1)+f(x_1,y_2)\right]$$
$$\left.+\frac{\Delta y}{3}\left[f(x_2,y_0)+4f(x_2,y_1)+f(x_2,y_2)\right]\right\}$$

$$V=\frac{0.5}{3}\left\{\frac{0}{3}\left[f(0,0)+4f(0,0)+f(0,0)\right]\right.$$
$$+4\frac{0.25}{3}\left[f(0.5,y_0)+4f(0.5,0.75)+f(0.5,0.75)\right]$$
$$\left.+\frac{0.5}{3}\left[f(1,1)+4f(1,1.5)+f(1,2)\right]\right.=0.7917$$

7.6.1 Instrumentación computacional de la fórmula de Simpson en dos direcciones

Se instrumenta el método de Simpson para una función de dos variables **f(x,y)**. Primero se integra en la dirección **X** y después, con los resultados obtenidos, se aplica nuevamente la fórmula de Simpson en la dirección **Y**.

f: función de dos variables
ax, bx, ay, by: límites de integración en las direcciones **x, y** respectivamente
mx, my: cantidad de franjas en cada dirección

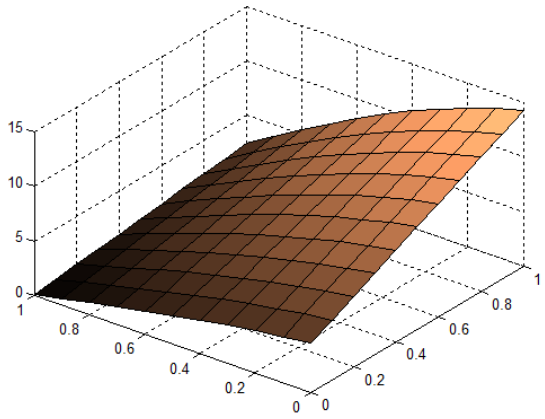
```
function s=simpson2(f, ax, bx, ay, by, mx, my)
dy=(by-ay)/my;
y=ay;
for i=1:my+1
   g = subs(f,'y',y);                    %Sustituye en f el símbolo y por cada valor
   r(i) = simpson(inline(g), ax, bx, mx);
   y = y+dy;
end
s=0;
for i=2:my
   s=s+2*(2-mod(i,2))*r(i);
end
s=dy/3*(r(1)+s+r(my+1));
```

Ejemplo.

Calcule $V = \int_0^1 \int_0^1 \cos(x^2 + y)(x + y)dxdy$ *con la regla de Simpson instrumentada en la función anterior. Use* ***m=4,8,12,…*** *en ambas variables para determinar la convergencia.*

Gráfico de la superficie usando comandos de MATLAB

```
>> x=0:0.1:1;                 Definir coordenadas para el plano XY
>> y=0:0.1:1;
>> [u,v]=meshgrid(x,y);       Definir la malla de puntos para evaluar
>> f=cos(u.^2+v)*(u+v);       Obtener puntos de la superficie
>> surf(x,y,f)                Gráfico de la superficie
>> colormap copper            Definir el color
```



Se observa que el integral entre el plano X-Y debajo de la superficie existe y está bien definido.

Solución con la función ***simpson2***

```
>> format long
>> syms x y;
>> f=cos(x^2+y)*(x+y);
>> v=simpson2(f,0,1,0,1,4,4)
v =
   0.500415316612236
>> v=simpson2(f,0,1,0,1,8,8)
v =
   0.500269266271819
>> v=simpson2(f,0,1,0,1,12,12)
v =
   0.500258596816339
>> v=simpson2(f,0,1,0,1,16,16)
v =
   0.500256714328757
>> v=simpson2(f,0,1,0,1,20,20)
v =
   0.500256191304419
```

Si se incrementa el número de franjas, el resultado tiende a un valor fijo que es el integral.

Nota. *Este integral no se puede resolver por métodos analíticos o con MATLAB:*

```
>> syms x y;
>> f=cos(x^2+y)*(x+y);
>> v=eval(int(int(f,0,1),0,1))
     Error using …
```

7.7 Ejercicios y problemas de integración numérica

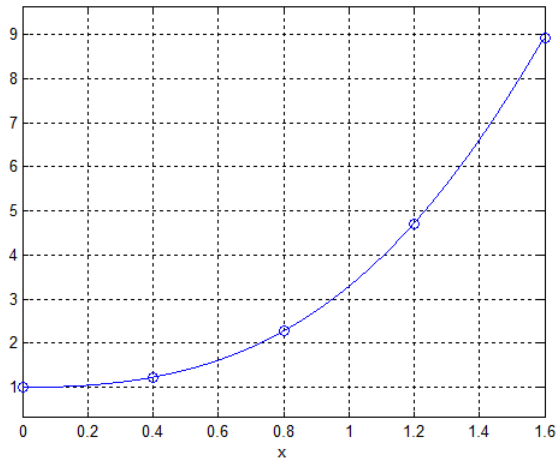
1. Se desea calcular numéricamente $\mathbf{A = \int_1^2 ln(x)dx}$ con la Regla de los Trapecios. Use la fórmula del error de truncamiento respectiva y sin realizar la integración, estime la cantidad de trapecios que tendría que usar para que la respuesta tenga cinco decimales exactos.

2. Se desea integrar $\mathbf{f(x) = exp(x) + 5x^3}$, $\mathbf{x \in [0, 2]}$

a) Use la fórmula del error para determinar la cantidad de trapecios que se deberían usar para obtener la respuesta con 2 decimales exactos.

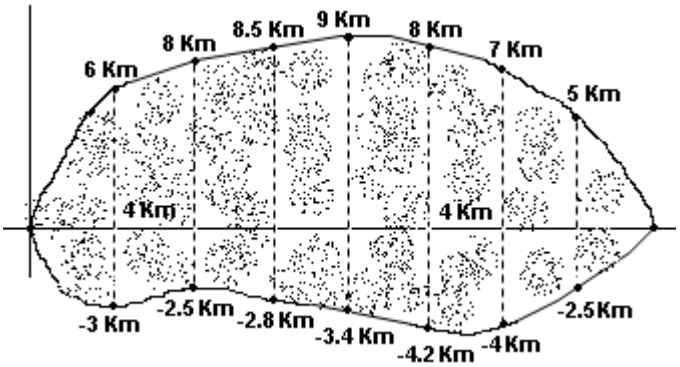
b) Calcule la respuesta usando la cantidad de trapecios especificada

3. a) Encuentre en forma aproximada el área debajo de **f(x), 0≤x≤1.6** Use la fórmula de Simpson incluyendo los **cinco puntos** tomados mediante aproximación visual del gráfico.



b) Estime el error en el resultado obtenido con una aproximación de diferencias finitas.

4. En el siguiente gráfico se muestra delineada la zona de un derrame de petróleo ocurrido en cierta región. Las mediciones han sido obtenidas a distancias de 4 Km.



Con la fórmula de Simpson, encuentre en forma aproximada el área total cubierta por el derrame de petróleo.

5. La siguiente definición se denomina función error: $\mathbf{erf(x) = \frac{1}{\sqrt{\pi}} \int_0^x e^{-t^2} dt}$

Esta función se usa para definir la función de densidad de probabilidad de la distribución normal estándar. Calcule **erf(1)** con la fórmula de Simpson con **m=2,4,6**. Con estos resultados estime la precisión en la respuesta.

6. Una empresa está vendiendo una licencia para manejo de un nuevo punto de venta. La experiencia indica que dentro de **t** años, la licencia generará utilidades según **f(t) = 14000+490t** dólares por año. Si la tasa de interés **r** permanece fija durante los próximos **n** años, entonces el valor presente de la licencia se puede calcular con

$$V = \int_0^n f(t)e^{-rt}dt$$

Calcule **V** si **n=5** años y **r=0.07**. Use la fórmula de Simpson con **m=4,6,8.** Con estos resultados estime el error de truncamiento

7. Un comerciante usa el siguiente modelo para describir la distribución de la proporción x del total de su mercadería que se vende semanalmente:

$f(x)=20x^3(1-x),\ 0\le x\le 1$

El área debajo de f(x) representa entonces la probabilidad que venda una cantidad x en cualquier semana.
Se desea conocer la probabilidad que en una semana venda más de la mitad de su mercadería (**debe integrar f entre 0.5 y 1**). Use la Fórmula de Simpson con m=4

8. Según la teoría de Kepler, el recorrido de los planetas es una elipse con el Sol en uno de sus focos. La longitud del recorrido total de la órbita esta dada por

$$s = 4\int_0^{\pi/2} a\sqrt{1-k^2\,\mathrm{sen}^2 t}\ dt,$$

Siendo **k:** excentricidad de la órbita y **a:** longitud del semieje mayor
Calcule el recorrido del planeta Mercurio sabiendo que **k=0.206**, **a=0.387** UA
(1 UA = 150 millones de km). Use la fórmula de Simpson con **m=4** (cantidad de franjas)

9. Una placa rectangular metálica de 0.45 m por 0.60 m pesa 5 Kg. Se necesita recortar este material para obtener una placa de forma elíptica, con eje mayor igual a 50 cm, y eje menor igual a 40 cm. Calcule el área de la elipse y determine el peso que tendrá esta placa. Para calcular el área de la elipse use la fórmula de Simpson con m = 4. Finalmente, estime cual es el el error de truncamiento en el valor del área calculada.

10. Una curva **C** puede darse en forma paramétrica con las ecuaciones:

$x = f(t)$

$y = g(t),\ \ t \in [a, b]$

Si no hay intersecciones entre **f** y **g**, entonces, la longitud del arco de **C** se puede calcular con la integral:

$$S = \int_a^b \sqrt{(x'(t))^2 + (y'(t))^2}\,dt$$

Use la fórmula de Simpson, **m=4**, para calcular la longitud del arco de la curva dada con las siguientes ecuaciones paramétricas

$$x = 2\cos(t),\ \ y = \sqrt{3}\,\mathrm{sen}(t),\ \ t \in [0, \pi/2]$$

11. Calcule el valor del integral $A = \int_2^\infty \frac{1}{3x^2+2}dx$. Use la Cuadratura de Gauss con dos puntos.

12. Calcule $A = \int_{1/2}^{1} \frac{dx}{(2x-1)^{1/3}}$.

a) Use directamente la Cuadratura de Gauss de dos puntos
b) Use la Regla de Simpson con m=4. Previamente debe usar una transformación: $t^3 = 2x - 1$

13. Calcule $A = \int_0^\infty \frac{dx}{1+x^4}$ Use la regla de Simpson, m=1,2,4 y estime el error

14. Calcule $\mathbf{A = \int_0^{\infty} x^2 e^{-x^2}}$ Use la Regla de Simpson con **m=4** y estime el error

15. Calcule $\mathbf{A = \int_2^{\infty} \frac{dx}{x^2 + x - 2}}$. Aplique dos veces la cuadratura de Gauss de dos puntos

16. Calcule $\mathbf{A = \int_1^{1.8} \int_{0.5}^{2.5} (x + y)\,sin(x)dxdy}$. Use la regla de Simpson en ambas direcciones, m=4

17. Calcular la siguiente integral, con el algoritmo de la integral doble de Simpson:

$$\mathbf{S = \iint_R x^2 \sqrt{9 - y^2}\, dA}$$

Donde **R** es la región acotada por: $\mathbf{x^2 + y^2 = 9}$. Usar m=4 en ambas direcciones

18. La función $\mathbf{\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx}$ se denomina Función Gamma y tiene aplicaciones en algunos modelos de probabilidad.

Encuentre una valor aproximado de $\mathbf{\Gamma(2)}$ aplicando una vez la Cuadratura de Gauss.

*Sugerencia: Separe el integral en dos partes **[0,1], [1, ∞]**. Con un cambio de variable elimine el límite superior ∞.*

19. El siguiente cuadro contiene valores de una función **f(x,y).** Use la fórmula de Simpson para calcular el volumen entre el plano **X-Y** y la superficie **f(x,y),** $\mathbf{0.1 \le x \le 0.5,\ \ 0.2 \le y \le 1.0}$

X \ Y	0.2	0.4	0.6	0.8	1.0
0.1	0.04	0.08	0.12	0.16	0.20
0.2	0.41	0.47	0.53	0.58	0.62
0.3	0.81	0.83	0.84	0.83	0.82
0.4	0.76	0.70	0.62	0.53	0.42
0.5	0.06	0.02	0.01	0.01	0.02

20. Cuando un cuerpo de masa m se desplaza verticalmente hacia arriba desde la superficie de la tierra, si prescindimos de toda fuerza de resistencia (excepto la fuerza de la gravedad), la velocidad de escape ***v*** está dada por: $\mathbf{v^2 = 2gR \int_1^{\infty} z^{-2} dz;} \quad \mathbf{z = \frac{x}{R}}$

Donde **R = 6371** km. es el radio promedio de la tierra, **g = 9.81** m/s^2 es la constante gravitacional. Utilice cuadratura Gaussiana para hallar **v**.

21. El siguiente integral no puede resolverse analíticamente. Aplique la fórmula de Simpson con 2, 4, 6, 8 subintervalos. Con estos resultados estime la precisión del resultado de la integración.

$$\mathbf{A = \int_0^2 e^{sen(x)} dx}$$

22. El siguiente integral no puede resolverse analíticamente y tampoco pueden aplicarse las fórmulas comunes de integración numérica.:

$$\mathbf{A = \int_0^1 \frac{sen(x)}{\sqrt{x}} dx}$$

Aplique la Cuadratura de Gauss con uno, dos y tres subintervalos. Con estos resultados estime la precisión del resultado de la integración.

23. En el techado de las casas se utilizan planchas corrugadas con perfil ondulado:

Cada onda tiene la forma **f(x) = sen(x),** con un periodo de **2π** pulgadas
El perfil de la plancha tiene **8** ondas y la longitud $\mathbf{L}$ de cada onda se la puede calcular con la siguiente integral: $\mathbf{L = \int_0^{2\pi} \sqrt{1 + (f'(x))^2}\, dx}$
Este integral no puede ser calculado por métodos analíticos.

a) Use la fórmula de Simpson con **m = 4, 6, 8, 10** para calcular $\mathbf{L}$ y estime el error en el último resultado
b) Con el último resultado encuentre la longitud del perfil de la plancha.

8 DIFERENCIACIÓN NUMÉRICA

En este capítulo se describe un procedimiento para obtener fórmulas para evaluar derivadas. Estas fórmulas son de especial interés en algunos métodos para resolver ecuaciones diferenciales. Estos métodos se estudiarán posteriormente y consisten en transformar las ecuaciones diferenciales en ecuaciones con el dominio discretizado.

8.1 Obtención de fórmulas de diferenciación numérica

Dados los puntos (x_i, f_i), **i=0, 1, 2, ..., n,** suponer que es de interés evaluar alguna derivada $f^{(k)}(x_i)$ siendo **f** desconocida pero supuestamente diferenciable.

Un enfoque para obtención de fórmulas es usar el polinomio de interpolación como una aproximación para **f** y derivar el polinomio:

$$f(x) \cong p_n(x) \Rightarrow f^{(k)}(x_i) \cong \frac{d^k}{dx^k}[p_n(x)]_{x=x_i}$$

Adicionalmente, con la fórmula del error en la interpolación se puede estimar el error para las fórmulas de las derivadas.

Otro enfoque más simple consiste en usar la serie de Taylor, bajo la suposición de que la función **f** se puede expresar mediante este desarrollo. Entonces mediante artificios algebraicos se pueden obtener fórmulas para aproximar algunas derivadas.

Si se desarrolla alrededor del punto x_i, usando la notación $f(x_i) \equiv f_i$, se obtienen:

(1) $$f_{i+1} = f_i + hf'_i + \frac{h^2}{2!}f''_i + \ldots + \frac{h^n}{n!}f^{(n)}_i + \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(z), \quad x_i \le z \le x_{i+1}$$

(2) $$f_{i-1} = f_i - hf'_i + \frac{h^2}{2!}f''_i - \ldots + \frac{h^n}{n!}f^{(n)}_i - \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(z), \quad x_{i-1} \le z \le x_i$$

8.2 Una fórmula para la primera derivada

Tomando tres términos de (1) y despejando **f'** se obtiene una aproximación para la primera derivada y para el error de truncamiento en la aproximación::

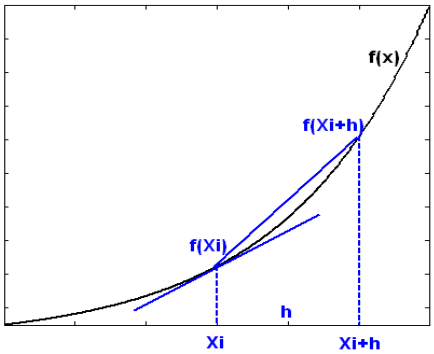
$$f_{i+1} = f_i + hf'_i + \frac{h^2}{2!}f''(z) \Rightarrow f'_i = \frac{f_{i+1} - f_i}{h} - \frac{h}{2}f''(z), \quad x_i \le z \le x_{i+1}$$

Definición: Una fórmula para la primera derivada con el error de truncamiento

$$f'_i \cong \frac{f_{i+1} - f_i}{h} = \frac{\Delta f_i}{h}$$

$$E = -\frac{h}{2}f''(z) = O(h), \quad x_i \le z \le x_{i+1}$$

Esta aproximación, gráficamente significa que la pendiente de la tangente a **f** en el punto **i** es aproximada mediante la pendiente de la recta que incluye a los puntos **i** e **i+1**, como se muestra en la figura:



Para que la aproximación sea aceptable, **h** debe ser suficientemente pequeño puesto que:

$$h \to 0 \Rightarrow E = -\frac{h}{2}f''(z) \to 0$$

Por lo tanto: $$h \to 0 \Rightarrow \frac{f_{i+1} - f_i}{h} \to f'_i$$

Sin embargo, si **h** es demasiado pequeño, puede introducirse el error de redondeo como se describe a continuación. Este error se debe a la imprecisión en los cálculos aritméticos o a la limitada capacidad de representación de los dispositivos de memoria para almacenamiento de los números reales.

Suponer que se debe evaluar **f** en un punto x_i. Por lo mencionado anteriormente no se obtendrá exactamente f_i sino un valor aproximado $\bar{f}_i$. Sea R_i el error de redondeo: $f_i = \bar{f}_i + R_i$

Si se sustituye en la fórmula para la derivada se tiene:

$$f'_i = \frac{f_{i+1} - f_i}{h} - \frac{h}{2} f''(z) = \frac{\bar{f}_{i+1} + R_{i+1} - (\bar{f}_i + R_i)}{h} - \frac{h}{2} f''(z) = \frac{\bar{f}_{i+1} - \bar{f}_i}{h} + \frac{R_{i+1} - R_i}{h} - \frac{h}{2} f''(z)$$

Debido a que el error de redondeo solamente depende del dispositivo de almacenamiento y no de **h** entonces, mientras que el error de truncamiento $E = -\frac{h}{2} f''(z)$ se reduce cuando $h \to 0$, puede ocurrir que la suma del error de redondeo R_{i+1} y R_i crezca ilimitadamente anulando la precisión que se obtiene al reducir el error de truncamiento **E**.

El siguiente programa escrito en MATLAB incluye la fórmula para estimar la primera derivada de $f(x)=x\,e^x$ para **x=1**. El valor exacto es $f'(1) = 5.436563656918091\ldots$

```
% demostración del comportamiento del error en diferenciación numérica
% f(x)=x*exp(x)
h=0.1;
v=5.436563656918091;
disp('          h              v. aproximado      error');
f=inline('x*exp(x)');
for i=1:15
  d=(f(1+h)-f(1))/h;
  e=v-d;
  fprintf('%20.15f    %12.10f   %12.10f\n',h,d,e);
  h=h/10;
end
```

h	v. aproximado	error
0.100000000000000	5.8630079788	-0.4264443219
0.010000000000000	5.4775196708	-0.0409560139
0.001000000000000	5.4406428924	-0.0040792355
0.000100000000000	5.4369714173	-0.0004077604
0.000010000000000	5.4366044314	-0.0000407744
0.000001000000000	5.4365677333	-0.0000040764
0.000000100000000	5.4365640656	-0.0000004087
0.000000010000000	5.4365635993	0.0000000576
0.000000001000000	5.4365636437	0.0000000132
0.000000000100000	5.4365623114	0.0000013455
0.000000000010000	5.4365401070	0.0000235499
0.000000000001000	5.4369841962	-0.0004205393
0.000000000000100	5.4312110365	0.0053526205
0.000000000000010	5.3734794392	0.0630842177
0.000000000000001	5.7731597281	-0.3365960711

Se observa que la precisión mejora cuando se reduce **h**, pero a partir de cierto valor de **h**, el resultado pierde precisión.

La aproximación propuesta para f'_i es una fórmula de primer orden pues el error de truncamiento es **E=O(h)**. Por lo tanto, si se desea una aproximación con mayor precisión, se debe elegir para **h** un valor muy pequeño, pero ya hemos mencionado que esto puede hacer que aparezca el error de redondeo.

Adicionalmente, si únicamente se tienen puntos de **f**, no se puede elegir **h**. Entonces es preferible usar fórmulas con mayor precisión pero que usen únicamente los puntos dados.

8.3 Una fórmula de segundo orden para la primera derivada

Una fórmula más precisa para la primera derivada se obtiene restando (1) y (2) con cuatro términos:

(1) $f_{i+1} = f_i + hf'_i + \frac{h^2}{2!}f''_i + \frac{h^3}{3!}f'''(z_1)\,, \quad x_i \le z_1 \le x_{i+1}$

(2) $f_{i-1} = f_i - hf'_i + \frac{h^2}{2!}f''_i - \frac{h^3}{3!}f'''(z_2)\,, \quad x_{i-1} \le z_2 \le x_i$

(1) – (2): $f_{i+1} - f_{i-1} = 2hf'_i + \frac{h^3}{3!}f'''(z_1) + \frac{h^3}{3!}f'''(z_2)$

$$\Rightarrow f'_i = \frac{f_{i+1} - f_{i-1}}{2h} - \frac{h^2}{12}(f'''(z_1) + f'''(z_2)) = \frac{f_{i+1} - f_{i-1}}{2h} - \frac{h^2}{12}(2f'''(z)),\ x_{i-1} \le z \le x_{i+1}$$

Definición: Una fórmula de segundo orden para la primera derivada

$$f'_i \cong \frac{f_{i+1} - f_{i-1}}{2h}$$

$$E = -\frac{h^2}{6}f'''(z) = O(h^2),\ x_{i-1} \le z \le x_{i+1}$$

Ejemplo. *Usar las fórmulas para evaluar* ***f'(1.1)*** *dados los siguientes datos:*
(x, f(x)): (1.0, 2.7183), (1.1, 3.3046), (1.2, 3.9841), (1.3, 4.7701)

$$f'_1 \cong \frac{f_2 - f_1}{h} = \frac{3.9841 - 3.3046}{0.1} = 6.7950, \qquad E=O(h)=O(0.1)$$

$$f'_1 \cong \frac{f_2 - f_0}{2h} = \frac{3.9841 - 2.7183}{2(0.1)} = 6.3293, \qquad E=O(h^2)=O(0.01)$$

Para comparar, estos datos son tomados de la función $\mathbf{f(x) = x\,e^x}$. El valor exacto es $\mathbf{f'(1.1) = 6.3087}$. El error en la primera fórmula es **-0.4863** y para la segunda es **-0.0206**

En la realidad, únicamente se conocen puntos de f(x) con los cuales se debe intentar estimar el error en los resultados de las fórmulas mediante las aproximaciones de diferencias finitas.

Ejemplo. *Estime el error en los resultados del ejemplo anterior, con los dados dados:*
(x, f(x)): (1.0, 2.7183), (1.1, 3.3046), (1.2, 3.9841), (1.3, 4.7701)

Tabulación de las diferencias finitas:

i	x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
0	1.0	2.7183	0.5863	0.0932	0.0133
1	1.1	3.3046	0.6795	0.1065	
2	1.2	3.9841	0.7860		
3	1.3	4.7701			

$$f'_1 \cong \frac{f_2 - f_1}{h} = \frac{3.9841 - 3.3046}{0.1} = 6.7950,$$

$$E = -\frac{h}{2}f''(z) \cong -\frac{h}{2}\frac{\Delta^2 f_i}{h^2} = -\frac{\Delta^2 f_i}{2h} = -\frac{0.1066}{2(0.1)} = -0.5325$$

$$f'_1 \cong \frac{f_2 - f_0}{2h} = \frac{3.9841 - 2.7183}{2(0.1)} = 6.3293,$$

$$E = -\frac{h^2}{6}f'''(z) \cong -\frac{h^2}{6}\frac{\Delta^3 f_i}{h^3} = -\frac{\Delta^3 f_i}{6h} = -\frac{0.0133}{6(0.1)} = -0.0222$$

En el primer resultado, el error está en el primer decimal. En el segundo resultado, el error está en el segundo decimal. Esto coincide los valores calculados.

8.4 Una fórmula para la segunda derivada

Una fórmula para la segunda derivada se obtiene sumando (1) y (2) con 5 términos:

(1) $f_{i+1} = f_i + hf'_i + \frac{h^2}{2!}f''_i + \frac{h^3}{3!}f'''_i + \frac{h^4}{4!}f^{iv}(z_1), \quad x_i \le z_1 \le x_{i+1}$

(2) $f_{i-1} = f_i - hf'_i + \frac{h^2}{2!}f''_i - \frac{h^3}{3!}f'''_i + \frac{h^4}{4!}f^{iv}(z_2), \quad x_{i-1} \le z_2 \le x_i$

(1) + (2): $f_{i+1} + f_{i-1} = 2f_i + 2(\frac{h^2}{2!}f''_i) + \frac{h^4}{4!}(f^{iv}(z_1) + f^{iv}(z_2))$

$$\Rightarrow f''_i = \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2} - \frac{h^2}{4!}(2f^{iv}(z)), \quad x_{i-1} \le z \le x_{i+1}$$

Definición: Una fórmula para la segunda derivada con el error de truncamiento

$$f''_i \cong \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}$$

$$E = -\frac{h^2}{12}f^{iv}(z) = O(h^2), \quad x_{i-1} \le z \le x_{i+1}$$

Ejemplo. *Use la fórmula para calcular* ***f''(1.1)*** *dados los siguientes datos:*
(x, f(x)): (1.0, 2.7183), (1.1, 3.3046), (1.2, 3.9841), (1.3, 4.7701)

$$f''_1 \cong \frac{f_0 - 2f_1 + f_2}{h^2} = \frac{2.7183 - 2(3.3046) + 3.9841}{0.1^2} = 9.32, \qquad E=O(h^2)=O(0.01)$$

Estos datos son tomados de la función $\mathbf{f(x) = x\,e^x}$. El valor exacto es **f''(1.1) = 9.3129**. El error de truncamiento es **-0.0071**. Si se tuviese un punto adicional, se pudiera estimar el error con la fórmula, usando los datos y una aproximación de diferencias finitas para la cuarta derivada.

8.5 Obtención de fórmulas de diferenciación numérica con el método de coeficientes indeterminados

Este método permite obtener fórmulas para derivadas usando como base cualquier grupo de puntos.

Dados los puntos (x_i, f_i), **i= 0, 1, 2, ..., n**, encontrar una fórmula para la k-ésima derivada de f(x) con la siguiente forma propuesta:

$$f^{(k)}(x_j) = c_0 f_i + c_1 f_{i+1} + c_2 f_{i+2} + \ldots, + c_m f_{i+m} + E, \qquad j \in \{i, i+1, i+2, \ldots, i+m\}$$

La derivada debe evaluarse en el punto **j** que debe ser alguno de los puntos que intervienen en la fórmula.

Para determinar los coeficientes y el error de truncamiento se sigue el procedimiento:

1) Desarrollar cada término alrededor del punto j con la serie de Taylor
2) Comparar los términos en ambos lados de la ecuación
3) Resolver el sistema resultante y obtener los coeficientes y la fórmula para **E**

***Ejemplo**. Con el Método de Coeficientes Indeterminados obtenga una fórmula para **f'** en el punto **i** usando los puntos **i** e **i+1**:*

$$f'_i = c_o f_i + c_1 f_{i+1} + E$$

1) *Desarrollar cada término alrededor del punto **i***

$$f'_i = c_o f_i + c_1 \left(f_i + h f'_i + \frac{h^2}{2!} f''(z)\right) + E$$

$$f'_i = (c_0 + c_1) f_i + c_1 h f'_i + c_1 \frac{h^2}{2!} f''(z) + E$$

2) *Comparar términos*

$$0 = c_0 + c_1$$
$$1 = c_1 h$$
$$0 = c_1 \frac{h^2}{2!} f''(z) + E$$

3) *Con las dos primeras ecuaciones se obtiene:*

$$c_1 = 1/h, \quad c_0 = -1/h$$

Con la tercera ecuación se obtiene:

$$E = -(1/h) \frac{h^2}{2!} f''(z))$$

Sustituyendo en la fórmula propuesta:

$$f'_i = (-1/h) f_i + (1/h) f_{i+1} + E = \frac{f_{i+1} - f_i}{h} - \frac{h}{2} f''(z), \quad x_i \le z \le x_{i+1}$$

Igual a la que se obtuvo directamente con la serie de Taylor.

8.6 Algunas otras fórmulas de interés para evaluar derivadas

Fórmulas para la primera derivada

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_1) - f(x_2)}{2h} + O(h^2)$$

$$f'(x_n) = \frac{3f(x_{n-2}) - 4f(x_{n-1}) + f(x_n)}{2h} + O(h^2)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h} + O(h^4)$$

....

8.7 Extrapolación para diferenciación numérica

Esta técnica se puede aplicar a la diferenciación numérica para mejorar la exactitud del valor de una derivada usando resultados previos de menor precisión.

Examinamos el caso de la primera derivada, comenzando con una fórmula conocida:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} + O(h^2)$$ El error de truncamiento es de segundo orden

Si se define el operador **D**:

$$D(h) = \frac{f(x_i + h) - f(x_i - h)}{2h}$$

$$f'(x_i) = D(h) + O(h^2)$$

Reduciendo **h,** la aproximación mejora pero el error sigue de segundo orden:

$$f'(x_i) = D(h/2) + O(h^2)$$

La técnica de extrapolación permite combinar estas aproximaciones para obtener un resultado más preciso, es decir con un error de truncamiento de mayor orden

Para aplicar la extrapolación se utiliza la serie de Taylor con más términos:

$$f(x_i + h) = f_i + hf_i^{(1)} + \frac{h^2}{2}f_i^{(2)} + \frac{h^3}{6}f_i^{(3)} + \frac{h^4}{6}f_i^{(4)} + O(h^5)$$

$$f(x_i - h) = f_i - hf_i^{(1)} + \frac{h^2}{2}f_i^{(2)} - \frac{h^3}{6}f_i^{(3)} + \frac{h^4}{6}f_i^{(4)} + O(h^5)$$

Restando y dividiendo para 2h

$$D(h) = f_i^{(1)} + \frac{h^2}{6}f_i^{(3)} + O(h^4)$$

Definiendo

$$A = \frac{f_i^{(3)}}{6} \Rightarrow D(h) = f_i^{(1)} + Ah^2 + O(h^4)$$

Si se puede evaluar en h/2:

$$D(h/2) = f_i^{(1)} + A\frac{h^2}{4} + O(h^4)$$

Para eliminar A se combinan estas dos expresiones:

$$D(h) - 4D(h/2) = f_i^{(1)} + Ah^2 + O(h^4) - 4f_i^{(1)} - Ah^2 + O(h^2) = -3f_i^{(1)} + O(h^2)$$

De donde se obtiene

$$f_i^{(1)} = f'(x_i) = \frac{4D(h/2) - D(h)}{3} + O(h^4)$$

Es una fórmula mayor precisión que la fórmula inicial, para evaluar la primera derivada.

Este procedimiento puede continuar para encontrar fórmulas de mayor orden.

Ejm. *Dados los puntos* ***(0.1, 0.1105), (0.2, 0.2442), (0.3, 0.4049), (0.4, 0.5967), (0.5, 0.8243)*** *de una función* ***f(x)****, calcule* ***f'(0.3)*** *usando extrapolación en la diferenciación*

$$h = 0.2,\ x_i = 0.3$$

$$f'(x_i) \cong D(h) = 1.7845;$$

$$f'(x_i) \cong D(h/2) = 1.7625;$$

$$f'(x_i) \cong \frac{4D(h/2) - D(h)}{3} = 1.7551$$

Para comparación, estos datos fueron tomados de la función $f(x)=x\,e^x$
Valor exacto ***f'(0.3) = 1.7548....*** *Se puede verificar que usando resultados con precisión limitada, se obtuvo un resultado con mayor precisión.*

8.8 Ejercicios de diferenciación numérica

1. Se tomaron los siguientes datos en Km. para las coordenadas del recorrido de un cohete: **(50, 3.5), (80, 4.2), (110, 5.7), (140, 3.8), (170, 1.2).**
Mediante aproximaciones de **segundo orden** determine

a) Velocidad en el centro de la trayectoria
b) Aceleración en el centro de la trayectoria

2. La fórmula de segundo orden $f'_i \cong \frac{f_{i+1} - f_{i-1}}{2h}$ para aproximar la primera derivada no puede aplicarse en los puntos extremos del conjunto de datos pues se requiere un punto a cada lado. Use el método de coeficientes indeterminados para encontrar fórmulas de segundo orden para la primera derivada con los siguientes puntos:

a) $f'_0 = C_0 f_0 + C_1 f_1 + C_2 f_2 + E_T$
b) $f'_n = C_n f_n + C_{n-1} f_{n-1} + C_{n-2} f_{n-2} + E_T$

3. Con el método de los coeficientes indeterminados demuestre la siguiente fórmula que relaciona la segunda derivada con la segunda diferencia finita:

$$f''(z) = \frac{\Delta^2 f_i}{h^2}, \text{ para algún } z \in (x_i, x_{i+2})$$

9 MÉTODOS NUMÉRICOS PARA RESOLVER ECUACIONES DIFERENCIALES ORDINARIAS

El análisis matemático de muchos problemas en ciencias e ingeniería conduce a la obtención de ecuaciones diferenciales ordinarias (EDO). El estudio clásico de las EDO ha enfatizado el estudio de técnicas de resolución pero que solamente son aplicables a un número reducido de EDO. Programas como MATLAB ya incorporan instrumentos para obtener y graficar estas soluciones analíticas. Los métodos numéricos son una opción importante para resolver estas ecuaciones especialmente cuando la solución analítica es muy complicada o imposible obtener. Estos métodos instrumentados computacionalmente proporcionan soluciones aproximadas para analizar el comportamiento de la solución con respecto al problema propuesto. Adicionalmente se puede experimentar numéricamente con la convergencia y la estabilidad.

Un problema importante es determinar las condiciones para que la solución exista y sea única, y conocer el dominio en el que la solución tiene validez. Otros temas relacionados son la sensibilidad de la solución a los cambios en la ecuación o en la condición inicial y la estabilidad de la solución calculada, es decir el estudio de la propagación de los errores en el cálculo numérico. Sin embargo, el objetivo principal es resolver la ecuación.

Una ecuación diferencial ordinaria de orden n es una ecuación del tipo:

$$F(x, y, y', y'', ..., y^{(n-1)}, y^{(n)}) = 0$$

En donde **y** es una función de la variable independiente **x.** El orden de la ecuación diferencial es el de su derivada más alta.

Si es que es posible expresar la ecuación diferencial en la forma:

$$y^{(n)} + a_1 y^{(n-1)} + \ldots + a_{n-1} y' + a_n y = b$$

en donde los coeficientes $a_1, a_2, \ldots a_n, b$ son constantes o solamente dependen de **x**, entonces es una ecuación diferencial lineal explícita de orden **n**

Una EDO es una ecuación en la que la incógnita es una función **y(x)** que satisface a la ecuación en cierto dominio y a las condiciones que normalmente se suministran para particularizar la ecuación. Los métodos numéricos proporcionan puntos de la función como una aproximación a la solución analítica, y con una estimación de la precisión.

Ejemplo. *Un cuerpo de masa* ***m*** *sujeto a un extremo de un resorte con constante de amortiguación* ***k****, con el otro extremo fijo, se desliza sobre una mesa con un coeficiente de fricción* ***c****. A partir de un estado inicial, las oscilaciones decrecen hasta que se detiene. La ecuación del movimiento es*

$$ma = \sum F_x = -cv - kx$$

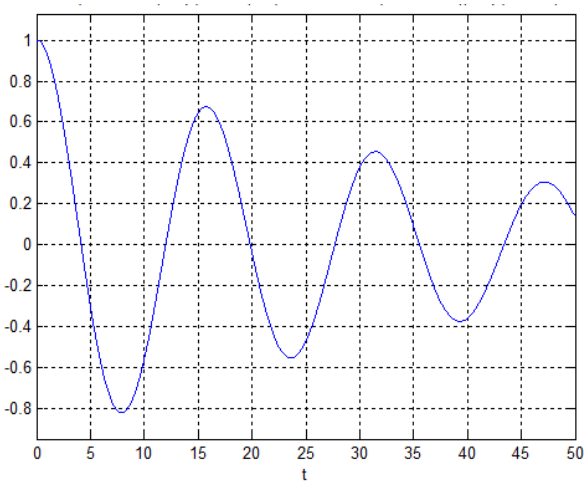
En donde ***a*** *es la aceleración,* ***v*** *es la velocidad,* ***x*** *es desplazamiento horizontal,* ***t*** *tiempo:*

$$\frac{d^2x}{dt^2} + \frac{c}{m}\frac{dx}{dt} + \frac{k}{m}x = 0$$

Es una ecuación diferencial ordinaria lineal de segundo orden. Su solución describe el desplazamiento ***x*** *en función del tiempo* ***t*** *partiendo de alguna condición inicial.*

Obtener la solución con la función ***dsolve*** *de* ***MATLAB*** *con los siguientes datos:* ***m=5, c=0.25, k=0.8, x(0)=1, x'(0)=0.*** *Graficar la solución en el intervalo,* ***0≤t≤50***

```
>> y=dsolve('D2x+0.25/5*Dx+0.8/5*x=0','x(0)=1,Dx(0)=0','t')
   y =
      cos((255^(1/2)*t)/40)/exp(t/40) + (255^(1/2)*sin((255^(1/2)*t)/40))/(255*exp(t/40))
>> digits(6)
>> y=vpa(y)                                   Solución en formato decimal con 6 dígitos
   y =
      cos(0.399218*t)/exp(0.025*t) + (0.0626224*sin(0.399218*t))/exp(0.025*t)
>> ezplot(y,[0,50]),grid on
```



En el gráfico se observan las oscilaciones del desplazamiento amortiguado.

9.1 Ecuaciones diferenciales ordinarias lineales de primer orden con la condición en el inicio

Estas ecuaciones tienen la forma general siguiente:

$\mathbf{F(x, y, y') = 0,}$ con la condición inicial $\mathbf{y(x_0) = y_0}$

Si es una ecuación diferencial explícita, se puede escribir en la siguiente manera:

$\mathbf{y'(x) = f(x, y),}\quad \mathbf{y(x_0) = y_0}$

En la notación común:

$$\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0$$

Su solución es una función **y(x)** definida en algún intervalo, que satisface a la ecuación e incluye a la condición inicial. La solución de esta ecuación se puede obtener integrando:

$$\int_{x_0}^{x} dy = \int_{x_0}^{x} f(x,y)dx \quad \Rightarrow \quad y(x) = y(x_0) + \int_{x_0}^{x} f(x,y)dx$$

Los métodos numéricos permiten obtener una solución aproximada cuando no es posible o es muy complicado obtenerla en forma explícita. Es conveniente comparar la solución numérica con la solución analítica de ecuaciones simples, para adquirir confianza al resolver ecuaciones más complicadas para las cuales ya no se pueda obtener la solución analítica.

9.1.1 Existencia de la solución

Condición de Lipschitz

Sea la función $\mathbf{f: A \to R,\ A \subseteq R}$. Si existe una constante $\mathbf{k \in R^+}$ tal que

$\mathbf{|f(a) - f(b)| \leq k|a - b|}$, para $\mathbf{\forall a,b \in A,}$ entonces **f** satisface la condición de Lipschitz en **A**

La condición de Lipschitz se puede interpretar geométricamente re-escribiéndola:

$$\left|\frac{f(a) - f(b)}{a - b}\right| \leq k, \text{ para } \forall a,b \in A, \quad a \neq b$$

Entonces, una función **f** satisface la condición de Lipschitz si y solo si las pendientes de todos los segmentos de recta que unen dos puntos de la gráfica de **y=f(x)** en **A**, están acotadas por algún número positivo **k**

Teorema.- Si $\mathbf{f: A \to R,\ A \subseteq R}$ es una función de Lipschitz, entonces el dominio y el rango de **f** son conjuntos acotados.

Sea una ecuación diferencial de primer orden con una condición inicial:

$$y'(x)=f(x,y),\ y(x_0)=y_0,\ x_0 \le x \le x_n$$

Teorema.- Si **f** es continua en $x_0 \le x \le x_n$, $-\infty < y < \infty$. Si f satisface la condición de Lipschitz en la variable $-\infty < y < \infty$, entonces la ecuación diferencial $y'(x)=f(x,y),\ y(x_0)=y_0$ tiene una solución única **y(x)** en $x_0 \le x \le x_n$

Es decir que **f** debe ser continua en el rectángulo $x_0 \le x \le x_n$, $-\infty < y < \infty$, y el cambio de **y'(x)=f(x,y)** para diferentes valores de **y** debe estar acotado.

Adicionalmente, es importante verificar si la solución calculada es muy sensible a los errores en la formulación de la ecuación diferencial o en la condición inicial. Se puede detectar esta situación calculando numéricamente el problema original y el problema modificado con alguna perturbación.

9.1.2 Método de la serie de Taylor

Se puede usar este desarrollo para obtener puntos de la solución a partir de la condición inicial conocida y para estimar el error de truncamiento. Debe elegirse la distancia **h** entre los puntos:

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \ldots + \frac{h^n}{n!}y^{(n)}_i + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(z),\ x_i \le z \le x_{i+1}$$

La ventaja de este enfoque es que se puede mejorar la precisión incluyendo más términos del desarrollo. Sin embargo, al usar las derivadas de **y(x)** se obtienen fórmulas aplicables únicamente para la ecuación especificada. Esto contrasta con el objetivo de los métodos numéricos que es proporcionar fórmulas generales.

Ejemplo. *Obtenga dos puntos de la solución de la siguiente ecuación diferencial utilizando los tres primeros términos de la serie de Taylor. Use* ***h = 0.1***

$$y' - y - x + x^2 - 1 = 0,\quad y(0) = 1$$

Solución

$$y' = f(x, y) = y - x^2 + x + 1,\quad x_0 = 0,\ y_0 = 1,\ h = 0.1$$

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i,\qquad E = \frac{h^3}{3!}y'''(z) = O(h^3) = O(0.001)$$ (Error de truncamiento)

$$x_{i+1} = x_i + h,\ i = 0, 1, 2, \ldots$$

$$y_{i+1} = y_i + h(y_i - x^2_i + x_i + 1) + \frac{h^2}{2!}(y'_i - 2x_i + 1)$$ (Derivar y sustituir **y'(x)**)

$$y_{i+1} = y_i + h(y_i - x^2_i + x_i + 1) + \frac{h^2}{2}(y_i - x^2_i + x_i + 1 - 2x_i + 1)$$

Fórmula para obtener puntos de la solución para el ejemplo anterior

$$y_{i+1} = y_i + h(y_i - x^2_i + x_i + 1) + \frac{h^2}{2}(y_i - x^2_i - x_i + 2)$$

$$x_{i+1} = x_i + h,\ i = 0, 1, 2, \ldots$$

Puntos de la solución:

i=0: $$y_1 = y_0 + h(y_0 - x^2_0 + x_0 + 1) + \frac{h^2}{2}(y_0 - x^2_0 - x_0 + 2)$$

$$= 1 + 0.1(1 - 0^2 + 0 + 1) + \frac{0.1^2}{2}(1 - 0^2 - 0 + 2) = 1.2150$$

$$x_1 = x_0 + h = 0 + 0.1 = 0.1$$

i=1: $$y_2 = y_1 + h(y_1 - x^2_1 + x_1 + 1) + \frac{h^2}{2}(y_1 - x^2_1 - x_1 + 2)$$

$$= 1.2150 + 0.1(1.2150 - 0.1^2 + 0.1 + 1) + \frac{0.1^2}{2}(1.2150 - 0.1^2 - 0.1 + 2) = 1.4610$$

$$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$$

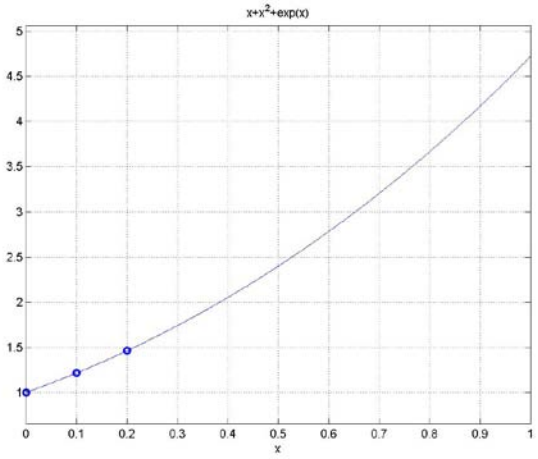
Para comprobar la exactitud comparamos con la solución exacta**:** $\mathbf{y(x) = e^x + x + x^2}$

y(0.1) = 1.2152

y(0.2) = 1.4614

El error está en el orden de los diezmilésimos y concuerda con el orden del error de truncamiento para esta fórmula

En el siguiente gráfico se muestran los dos puntos obtenidos junto con lel gráfico de la solución analítica exacta. La concordancia es muy buena.



Instrumentación computacional

Una función en MATLAB para usar la fórmula deducida para el ejemplo anterior

$$\mathbf{y_{i+1} = y_i + h(y_i - x^2_i + x_i + 1) + \frac{h^2}{2}(y_i - x^2_i - x_i + 2)}$$

$$\mathbf{x_{i+1} = x_i + h,\ \ i = 0, 1, 2, ...}$$

```
function [x,y] = taylor2(x,y,h)
y=y+h*(y-x^2+x+1) + h^2/2*(y-x^2-x+2);
x=x+h;
```

Obtención de puntos de la solución desde la ventana de comandos

```
>> x=0;y=1;h=0.1;
>> [x, y]= taylor2(x,y,h)
x =
    0.1000
y =
    1.2150
>> [x, y]=taylor2(x,y,h)
x =
    0.2000
y =
    1.4610
```

Este comando se reutiliza para obtener cada punto

El uso del método desde la ventana de comandos no es práctico para recibir los resultados. Conviene guardar los puntos obtenidos. Para esto se puede escribir un programa.

Un programa en MATLAB para calcular **m=20** puntos espaciados en una distancia **h=0.1** para el ejercicio anterior con la fórmula de Taylor:

```
x=0;
y=1;
m=20;
h=0.1;
for i=1:m
  [x,y]=taylor2(x,y,h);
  u(i)=x;
  v(i)=y;
end
```

Se ha almacenado el programa con el nombre **ed1**. Los siguientes comandos permiten usar los vectores u, v que contienen puntos de la solución para visualizarla y compararla con la solución analítica exacta obtenida con la función **dsolve** de MATLAB.

```
>> ed1
>> plot(u, v, 'o');                                  u, v contienen los puntos calculados
>> g=dsolve('Dy-y-x+x^2-1=0','y(0)=1','x')          Obtención de la solución analítica.
     g =
        x+x^2+exp(x)                                 Solución analítica
>> hold on;
>> grid on;
>> ezplot(g,0,2);
```

En esta instrumentación se usa la serie de Taylor para deducir una fórmula aplicable únicamente al problema especificado. En las siguientes secciones se desarrollarán métodos numéricos generales deduciendo fórmulas que se pueden aplicar a diferentes ecuaciones diferenciales, con lo cual los algoritmos ya no dependen de un problema particular.

9.1.3 Fórmula de Euler

El objetivo de los métodos numéricos es proporcionar fórmulas generales y algoritmos que no dependan de los datos de un problema particular. Las siguientes fórmulas y algoritmos se pueden especificar independientemente de la forma de una EDO y de su condición inicial, las cuales se pueden definir desde fuera del algoritmo.

Sea una ecuación diferencial ordinaria explícita de primer orden con una condición en el inicio:

$$y'(x) = f(x, y), \quad y(x_0) = y_0$$

La **fórmula de Euler** usa los dos primeros términos de la serie de Taylor:

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''(z) = y_i + hf(x_i, y_i) + \frac{h^2}{2!}y''(z), \quad x_i \le z \le x_{i+1}$$

Definición: Fórmula de Euler

$$y_{i+1} = y_i + h\, f(x_i, y_i)$$
$$x_{i+1} = x_i + h, \qquad i = 0, 1, 2, \ldots$$
$$E = \frac{h^2}{2!}y''(z) = O(h^2), \quad x_i \le z \le x_{i+1} \qquad \text{(Error de truncamiento en cada paso)}$$

Algoritmo para calcular puntos de la solución de una EDO de primer orden con la fórmula de Euler

1) Defina f(x,y) y la condición incial (x_0, y_0)
2) Defina h y la cantidad de puntos a calcular m
3) Para i =1, 2, ..., m
4) $y_{i+1} = y_i + h\, f(x_i, y_i)$.
5) $x_{i+1} = x_i + h$
6) fin

Ejemplo. *Obtenga dos puntos de la solución de la siguiente ecuación diferencial con la fórmula de Euler. Use h = 0.1*

$$y' + y + x - 1 = 0, \quad y(0) = 1$$

Ecuación diferencial

$$y' = f(x, y) = -y - x + 1, \quad x_0 = 0, y_0 = 1, \quad h = 0.1$$

Cálculo de los puntos

i=0: $y_1 = y_0 + h\, f(x_0, y_0) = 1 + 0.1\, f(0, 1) = 1 + 0.1\,[-1 - 0 + 1] = 1.2000;$
$x_1 = x_0 + h = 0 + 0.1 = 0.1$

i=1: $y_2 = y_1 + h\, f(x_1, y_1) = 1.2 + 0.1\, f(0.1, 1.2) = 1.2 + 0.1\,[1.2 - 0.1^2 + 0.1 + 1] = 1.4290$
$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$

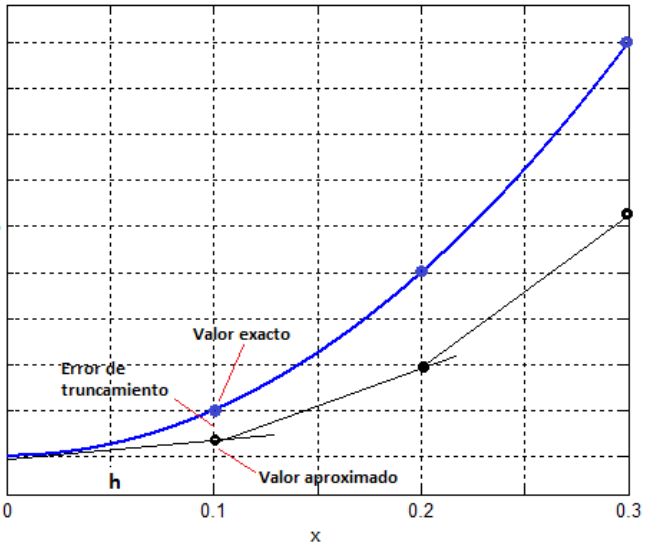
Para comprobar comparamos con la solución exacta: $y(x) = x + x^2 + e^x$

$y(0.1) = 1.2152$
$y(0.2) = 1.4614$

El error es muy significativo. Para reducirlo se pudiera reducir ***h****. Esto haría que el error de truncamiento se reduzca pero si la cantidad de cálculos es muy grande, pudiera acumular error de redondeo. Una mejor estrategia es usar métodos más precisos que no requieran que* ***h*** *sea muy pequeño.*

9.1.4 Error de truncamiento y error de redondeo

La fórmula de Euler utiliza la pendiente de la recta en cada punto para predecir y estimar la solución en el siguiente punto, a una distancia elegida **h**. La diferencia del punto calculado, con respecto al valor exacto es el error de truncamiento, el cual puede crecer al proseguir el cálculo.



En cada paso el error de truncamiento es:

$$E = \frac{h^2}{2} y''(z) = O(h^2),$$

Para reducir **E** se debe reducir **h** puesto que $h \to 0 \Rightarrow E \to 0$. Sin embargo, este hecho matemáticamente cierto, al ser aplicado tiene una consecuencia importante que es interesante analizar:

Suponer que se desea calcular la solución **y(x)** en un intervalo fijo $x_0 \le x \le x_f$ mediante **m** puntos $x_i = x_0, x_1, x_2, \ldots, x_m$ espaciados regularmente en una distancia **h**:

$$h = \frac{x_f - x_o}{m}$$

Sea E_i el error de truncamiento en el paso **i**, entonces

$$y_1 = y_0 + h\, f(x_0, y_0) + E_1$$
$$y_2 = y_1 + h\, f(x_1, y_1) + E_2 = y_0 + hf(x_0, y_0) + E_1 + hf(x_1, y_1) + E_2$$
$$= y_0 + hf(x_0, y_0) + hf(x_1, y_1) + E_1 + E_2$$
$$y_3 = y_2 + h\, f(x_2, y_2) + E_3 = y_0 + hf(x_0, y_0) + hf(x_1, y_1) + hf(x_2, y_2) + E_1 + E_2 + E_3$$
$$\ldots$$
$$y_m = y_0 + hf(x_0, y_0) + hf(x_1, y_1) + hf(x_2, y_2) + \ldots + hf(x_{m-1}, y_{m-1}) + E_1 + E_2 + E_3 + \ldots + E_m$$

Siendo $E_i = \frac{h^2}{2} y''(z_i)$

El error de truncamiento acumulado es:

$$E = E_1 + E_2 + E_3 + \ldots + E_m = mh^2(\bar{D}) = \frac{x_f - x_0}{h} h^2(\bar{D}) = h[(x_f - x_0)\bar{D}]$$

En donde suponemos que existe un valor promedio $\bar{D}$ de los valores de $\frac{y''(z_i)}{2}$, independiente de h.

Se muestra que el error de truncamiento acumulado es solamente de orden **O(h)**, por lo tanto **h** debe ser un valor mas pequeño que el previsto para asegurar que la solución calculada sea suficientemente precisa hasta el final del intervalo.

Por otra parte, cada vez que se evalúa $\mathbf{f(x_i, y_i)}$ se puede introducir el error de redondeo $\mathbf{R_i}$ debido a los errores en la aritmética computacional y al dispositivo de almacenamiento. Entonces, el error de redondeo se pudiera acumular en cada paso y al final del intervalo se tendrá:

$$\mathbf{R = R_1 + R_2 + R_3 + \ldots + R_m = m\,(\bar{R}) = \frac{x_f - x_0}{h}(\bar{R}) = \frac{1}{h}[(x_f - x_0)\bar{R}]}$$

$\bar{\mathbf{R}}$ es algún valor promedio del error de redondeo en cada paso, independiente de **h**.

El error total acumulado al realizar los cálculos con la fórmula de Euler hasta el final del intervalo es:

$$\mathbf{E_A = E + R = h[(x_f - x_0)\bar{D}] + \frac{1}{h}[(x_f - x_0)\bar{R}]}$$

Si **m** es muy grande, **h** será muy pequeño. Al reducir **h**, el error de redondeo puede llegar a ser mayor al error de truncamiento, y el resultado perderá precisión en vez de mejorarla como ocurriría si solamente se considera el error de truncamiento.

Como conclusión de lo anterior, es preferible usar fórmulas cuyo error de truncamiento **E** sea de mayor orden para que el valor de **h** no requiera ser muy pequeño si se buscan resultados con alta precisión. Esto retardará también el efecto del error de redondeo acumulado **R**.

9.1.5 Instrumentación computacional de la fórmula de Euler

Se define una función que recibe un punto de la solución y entrega el siguiente:

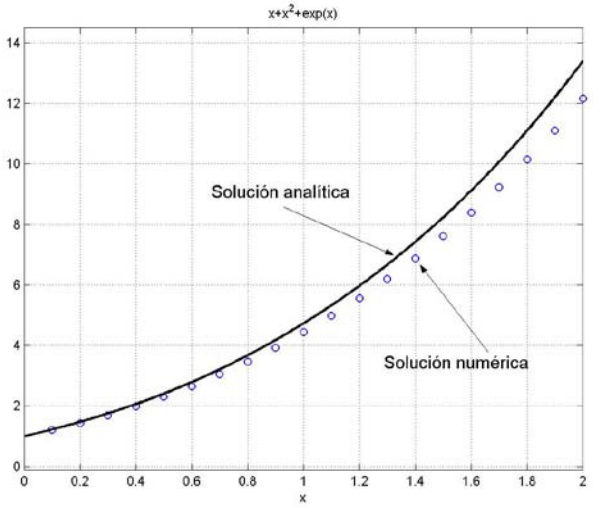
```
function [x,y] = euler(f, x, y, h)
y=y + h*f(x,y);
x=x+h;
```

Ejemplo. *Escribir un programa en MATLAB para calcular m=20 puntos espaciados en una distanciah= 0.1 del ejercicio anterior con la fórmula de Euler*

```
f=inline('y - x^2 + x + 1');
x=0;
y=1;
m=20;
h=0.1;
for i=1:m
  [x,y]=euler(f,x,y,h);
  u(i)=x;
  v(i)=y;
end
```

Si el programa se almacenó con el nombre ***ed2****. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta*

```
>> ed2
>> plot(u, v, 'o'), grid on, hold on          u, v contienen los puntos calculados
>> g=dsolve('Dy-y-x+x^2-1=0','y(0)=1','x')   Obtención de la solución analítica.
     g =
        x+x^2+exp(x)                          Solución analítica
>> hold on;
>> ezplot(g,0,2);
```



Solución analítica y solución numérica para el ejemplo anterior
Se observa la acumulación del error de truncamiento

9.1.6 Fórmula mejorada de Euler o fórmula de Heun

Sea la EDO de primer orden con una condición en el inicio: $\mathbf{y'(x) = f(x, y),\quad y(x_0) = y_0}$

La fórmula de Heun o fórmula mejorada de Euler usa los tres primeros términos de la serie de Taylor y un artificio para sustituir la primera derivada de **f(x, y)**

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \frac{h^3}{3!}y'''(z) = y_i + hf(x_i, y_i) + \frac{h^2}{2!}f'(x_i, y_i) + \frac{h^3}{3!}y'''(z),\quad x_i \le z \le x_{i+1}$$

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2}f'(x_i, y_i) + O(h^3)$$

Para evaluar $\mathbf{f'(x_i, y_i)}$ usamos una aproximación simple: $\mathbf{f'_i = \frac{f_{i+1} - f_i}{h} + O(h)}$

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\left[\frac{f_{i+1} - f_i}{h} + O(h)\right] + O(h^3) = y_i + hf_i + \frac{h}{2}f_{i+1} - \frac{h}{2}f_i + O(h^3)$$

$$y_{i+1} = y_i + \frac{h}{2}(f_i + f_{i+1}) + O(h^3)$$

Para evaluar $\mathbf{f_{i+1} = f(x_{i+1}, y_{i+1})}$ se usa $\mathbf{y_{i+1}}$ calculado con la fórmula de Euler como aproximación inicial:

$y_{i+1} = y_i + hf(x_i, y_i)$	Valor usado como una aproximación
$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y_{i+1}))$	Valor mejorado con la fórmula de Heun
$x_{i+1} = x_i + h,\quad i = 0, 1, 2, \ldots$	

Esta fórmula se puede re-escribir como se muestra en la definición:

Definición: Fórmula de Heun

$$K_1 = hf(x_i, y_i)$$
$$K_2 = hf(x_i + h, y_i + K_1)$$
$$y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2)$$
$$x_{i+1} = x_i + h,\quad i = 0, 1, 2, \ldots$$
$$E = \frac{h^3}{3!}y'''(z) = O(h^3),\quad x_i \le z \le x_{i+1}$$ (Error de truncamiento en cada paso)

Gráficamente, se puede interpretar que esta fórmula calcula cada nuevo punto usando un promedio de las pendientes en los puntos inicial y final en cada intervalo de longitud **h**.

El error de truncamiento en cada paso es de tercer orden $O(h^3)$, y el error de truncamiento acumulado es de segundo orden $O(h^2)$, mejor que la fórmula de Euler.

Algoritmo para resolver una EDO de primer orden con la fórmula de Heun

1) **Defina f(x,y) y la condición incial (x_0, y_0)**
2) **Defina h y la cantidad de puntos a calcular m**
3) **Para i =1, 2, ..., m**
4) $K_1 = hf(x_i, y_i)$
5) $K_2 = hf(x_i + h, y_i + K_1))$
6) $y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2)$.
7) $x_{i+1} = x_i + h$
8) **fin**

Ejemplo. *Obtener dos puntos de la solución de la siguiente ecuación diferencial con la fórmula de Heun. Use h = 0.1*

$$y' - y - x + x^2 - 1 = 0, \quad y(0) = 1$$

$$y' = f(x, y) = x - x^2 + y + 1, \quad x_0 = 0, y_0 = 1, \quad h = 0.1$$

Cálculos

i=0: $K_1 = hf(x_0, y_0) = 0.1\, f(0, 1) = 0.1\,(0 - 0^2 + 1 + 1) = 0.2000;$

$K_2 = hf(x_0 + h, y_0 + K_1) = 0.1\, f(0.1, 1.2) = 0.1\,[0.1 - 0.1^2 + 1.2 + 1] = 0.2290$

$y_1 = y_0 + \frac{1}{2}(K_1 + K_2) = 1 + 0.5(0.2000 + 0.2290) = 1.2145$

$x_1 = x_0 + h = 0 + 0.1 = 0.1$

i=1: $K_1 = hf(x_1, y_1) = 0.1\, f(0.1, 1.2145) = 0.1\,(0.1 - 0.1^2 + 1.2145 + 1) = 0.2305;$

$K_2 = hf(x_1 + h, y_1 + K_1) = 0.1\, f(0.2, 1.4450) = 0.1\,[0.2 - 0.2^2 + 1.4450 + 1] = 0.2605$

$y_2 = y_1 + \frac{1}{2}(K_1 + K_2) = 1.2145 + 0.5(0.2305 + 0.2605) = 1.4600$

$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$

Para comprobar comparamos con la solución exacta: $y(x) = x + x^2 + e^x$

$y(0.1) = 1.2152$

$y(0.2) = 1.4614$

El error de truncamiento en cada paso está en el orden de los milésimos, coincidiendo aproximadamente con $E=O(h^3)$

9.1.7 Instrumentación computacional de la fórmula de Heun

Se define una función que recibe un punto de la solución y entrega el siguiente:

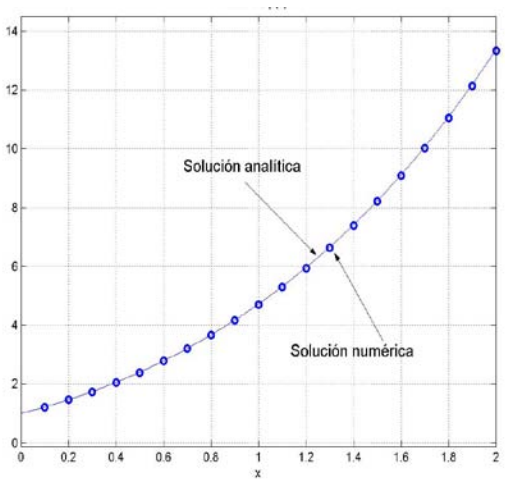
```
function [x,y] = heun(f, x, y, h)
k1=h*f(x,y);
k2=h*f(x+h, y+k1);
y=y+0.5*(k1+k2);
x=x+h;
```

***Ejemplo**. Escriba un programa en MATLAB para calcular m=20 puntos espaciados en una distancia h=0.1 del ejercicio anterior usando la fórmula de Heun*

```
f=inline('y - x^2 + x + 1');
x=0;
y=1;
m=20;
h=0.1;
for i=1:20
  [x,y]=heun(f,x,y,h);
  u(i)=x;                    %  La solución es almacenada
  v(i)=y;                    %  en los vectores u, v
end
```

*Si el programa se almacenó con el nombre **ed3**. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta*

```
>> ed3
>> plot(u, v, 'o'), grid on, hold on          % u, v contienen los puntos calculados
>> g=dsolve('Dy-y-x+x^2-1=0','y(0)=1','x')   % Obtención de la solución analítica.
     g =
        x+x^2+exp(x)                          % Solución analítica de MATLAB
>> ezplot(g,0,2);
```



Soluciones analítica y numérica para el ejemplo anterior
Se observa una reducción del error de truncamiento

9.1.8 Fórmulas de Runge-Kutta

Estas fórmulas utilizan artificios matemáticos para incorporar más términos de la serie de Taylor. Describimos la más popular, denominada fórmula de Runge-Kutta de cuarto orden, la cual incluye los cinco primeros términos de la Serie de Taylor.

Sea la ED de primer orden con una condición en el inicio: $y'(x) = f(x, y), \quad y(x_0) = y_0$

Definición: Fórmula de Runge-Kutta de cuarto orden

$$K_1 = hf(x_i, y_i)$$
$$K_2 = hf(x_i + h/2, y_i + K_1/2)$$
$$K_3 = hf(x_i + h/2, y_i + K_2/2)$$
$$K_4 = hf(x_i + h, y_i + K_3)$$
$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$
$$x_{i+1} = x_i + h, \qquad i = 0, 1, 2, \ldots$$
$$E = \frac{h^5}{5!}y^{(5)}(z) = O(h^5), \quad x_i \le z \le x_{i+1} \qquad \text{(Error de truncamiento en cada paso)}$$

Gráficamente, se puede interpretar que esta fórmula calcula cada nuevo punto usando un promedio ponderado de las pendientes en los puntos inicial, medio y final en cada intervalo de longitud **h**.

El error de truncamiento en cada paso es de quinto orden $O(h^5)$, y el error de truncamiento acumulado es de cuarto orden $O(h^4)$, suficientemente exacto para problemas comunes.

Algoritmo para resolver una EDO de primer orden con la fórmula de Runge-Kutta

1) **Defina f(x,y) y la condición inicial (x_0, y_0)**
2) **Defina h y la cantidad de puntos a calcular m**
3) **Para i =1, 2, ..., m**
4) $K_1 = hf(x_i, y_i)$
5) $K_2 = hf(x_i + h/2, y_i + K_1/2)$
6) $K_3 = hf(x_i + h/2, y_i + K_2/2)$
7) $K_4 = hf(x_i + h, y_i + K_3)$
8) $y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$.
9) $x_{i+1} = x_i + h$
10) **fin**

Ejemplo. *Obtenga un punto de la solución de la siguiente ecuación diferencial con la fórmula de Runge-Kutta de cuarto orden. Use h = 0.1*

$$y' - y - x + x^2 - 1 = 0, \quad y(0) = 1$$

Ecuación diferencial

$$y' = f(x, y) = x - x^2 + y + 1, \quad x_0 = 0, y_0 = 1, \quad h = 0.1$$

Cálculo de los puntos

i=0: $K_1 = hf(x_0, y_0) = 0.1\, f(0, 1) = 0.1\,(0 - 0^2 + 1 + 1) = 0.2000;$

$K_2 = hf(x_0 + h/2, y_0 + K_1/2) = 0.1\, f(0.05, 1.1) = 0.1\,(0.05 - 0.05^2 + 1.1 + 1) = 0.2148$

$K_3 = hf(x_0 + h/2, y_0 + K_2/2) = 0.1\, f(0.05, 1.1074) = 0.1\,(0.05 - 0.05^2 + 1.1074 + 1) = 0.2155$

$K_4 = hf(x_0 + h, y_0 + K_3) = 0.1\, f(0.1, 1.2155) = 0.1\,(0.1 - 0.1^2 + 1.2155 + 1) = 0.2305$

$$y_1 = y_0 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 1 + \frac{1}{6}[0.2 + 2(0.2148) + 2(0.2155) + 0.2305] = 1.2152$$

$$x_1 = x_0 + h = 0 + 0.1 = 0.1$$

Para comprobar comparamos con la solución exacta: $y(x) = x + x^2 + e^x$

$$y(0.1) = 1.2152$$

El error de truncamiento en cada paso está en el orden de los cienmilésimos, coincidiendo aproximadamente con $E=O(h^5)$. Los resultados tienen una precisión aceptable para la solución de problemas prácticos, por lo cual esta fórmula es muy utilizada

9.1.9 Instrumentación computacional de la fórmula de Runge-Kutta

Se define una función que recibe un punto de la solución y entrega el siguiente:

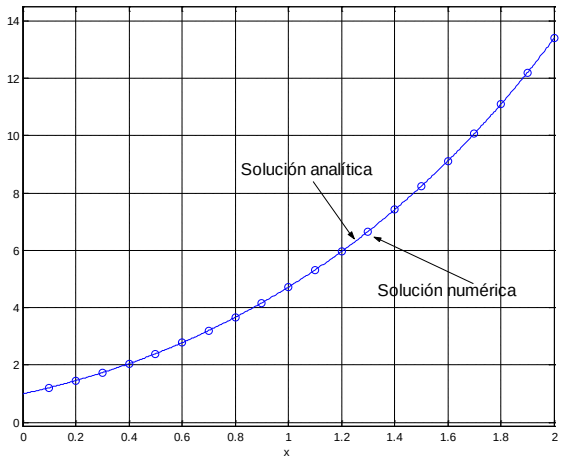
```
function [x,y]=rk4(f, x, y, h)
k1=h*f(x,y);
k2=h*f(x+h/2, y+k1/2);
k3=h*f(x+h/2, y+k2/2);
k4=h*f(x+h, y+k3);
y=y+1/6*(k1+2*k2+2*k3+k4);
x=x+h;
```

Ejemplo. *Un programa en MATLAB para calcular m=20 puntos espaciados en una distancia h=0.1 del ejercicio anterior usando la fórmula de Runge-Kutta de cuarto orden*

```
f=inline('y - x^2 + x + 1');
x=0;
y=1;
m=20;
h=0.1;
for i=1:m
  [x,y]=rk4(f,x,y,h);
  u(i)=x;
  v(i)=y;
end
```

Si el programa se almacenó con el nombre ***ed4****. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta*

```
>> ed4
>> plot(u, v, 'o'), grid on, hold on          u, v contienen los puntos calculados
>> g=dsolve('Dy-y-x+x^2-1=0','y(0)=1','x')   Obtención de la solución analítica.
     g =
        x+x^2+exp(x)                          Solución analítica
>> ezplot(g,0,2);
```



Solución analítica y solución numérica para el ejemplo anterior

Encontrar la diferencia entre la solución numérica y analítica cuando ***x=1***

```
>> yn=v(10)                    Solución numérica en el vector v
yn =
   4.718276340387802
>> x=1;
>> ya=eval(g)                  Solución analítica
ya =
   4.718281828459046
>> e=yn-ya
e =
   -5.488071243675563e-006
```

9.2 Sistemas de ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio

Los métodos numéricos desarrollados para una ecuación diferencial ordinaria de primer orden pueden extenderse directamente a sistemas de ecuaciones diferenciales de primer orden.

Analizamos el caso de dos ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio

$$F(x, y, z, y') = 0, \quad y(x_0) = y_0$$
$$G(x, y, z, z') = 0, \quad z(x_0) = z_0$$

Se deben escribir en la notación adecuada para usar los métodos numéricos

$$y' = f(x, y, z), \quad y(x_0) = y_0$$
$$z' = g(x, y, z), \quad z(x_0) = z_0$$

9.2.1 Fórmula de Heun extendida a dos E. D. O. de primer orden

La fórmula de Heun o fórmula mejorada de Euler para un sistema de dos EDO's de primer orden con condiciones en el inicio, es una extensión directa de la fórmula para una EDO:

Definición: Fórmula de Heun para un sistema de dos EDO de primer orden con condiciones en el inicio

$$K_{1,y} = hf(x_i, y_i, z_i)$$
$$K_{1,z} = hg(x_i, y_i, z_i)$$
$$K_{2,y} = hf(x_i + h, y_i + K_{1,y}, z_i + K_{1,z})$$
$$K_{2,z} = hg(x_i + h, y_i + K_{1,y}, z_i + K_{1,z})$$
$$y_{i+1} = y_i + \frac{1}{2}(K_{1,y} + K_{2,y})$$
$$z_{i+1} = z_i + \frac{1}{2}(K_{1,z} + K_{2,z})$$
$$x_{i+1} = x_i + h, \qquad i = 0, 1, 2, \ldots$$
$$E = O(h^3), \quad x_i \le z \le x_{i+1} \qquad \text{(Error de truncamiento en cada paso)}$$

Ejemplo. *Obtenga dos puntos de la solución del siguiente sistema de ecuaciones diferenciales con la fórmula de Heun. Use* ***h = 0.1***

$$y' - x - y - z = 0, \quad y(0) = 1$$
$$z' + x - y + z = 0, \quad z(0) = 2$$

Ecuaciones diferenciales

$$y' = f(x, y, z) = x + y + z, \quad x_0 = 0, y_0 = 1$$
$$z' = g(x, y, z) = -x + y - z, \quad x_0 = 0, z_0 = 2$$

Cálculo de dos puntos de la solución

i=0:

$$K_{1,y} = hf(x_0, y_0, z_0) = 0.1\, f(0, 1, 2) = 0.1\,(0 + 1 + 2) = 0.3$$
$$K_{1,z} = hg(x_0, y_0, z_0) = 0.1\, g(0, 1, 2) = 0.1\,(-0 + 1 - 2) = -0.1$$
$$K_{2,y} = hf(x_0+h, y_0+K_{1,y}, z_0+K_{1,z}) = 0.1\, f(0.1, 1.3, 1.9) = 0.1\,(0.1 + 1.3 + 1.9) = 0.33$$
$$K_{2,z} = hg(x_0+h, y_0+K_{1,y}, z_0+K_{1,z}) = 0.1\, f(0.1, 1.3, 1.9) = 0.1\,(-0.1 + 1.3 - 1.9) = -0.07$$
$$y_1 = y_0 + \frac{1}{2}(K_{1,y} + K_{2,y}) = 1 + 0.5(0.3 + 0.33) = 1.3150$$
$$z_1 = z_0 + \frac{1}{2}(K_{1,z} + K_{2,z}) = 2 + 0.5(-0.1 + (-0.07)) = 1.9150$$
$$x_1 = x_0 + h = 0 + 0.1 = 0.1$$

i=1: $K_{1,y} = hf(x_1, y_1, z_1) = 0.1\ f(0.1, 1.3150, 1.9150) = 0.333$
$K_{1,z} = hg(x_1, y_1, z_1) = 0.1\ g(0.1, 1.3150, 1.9150) = -0.07$
$K_{2,y} = hf(x_1+h, y_1+K_{1,y}, z_1+K_{1,z}) = 0.1\ f(0.2, 1.648, 1.845) = 0.3693$
$K_{2,z} = hg(x_1+h, y_1+K_{1,y}, z_1+K_{1,z}) = 0.1\ f(0.2, 1.648, 1.845) = -0.0397$

$$y_2 = y_1 + \frac{1}{2}(K_{1,y} + K_{2,y}) = 1.6662$$

$$z_2 = z_1 + \frac{1}{2}(K_{1,z} + K_{2,z}) = 1.8602$$

$x_2 = x_1 + h = 0.1 + 0.1 = 0.2$

Para comprobar comparamos con la solución exacta

y(0.1) = 1.3160, z(0.1) = 1.9150
y(0.2) = 1.6684, z(0.2) = 1.8604

Los resultados calculados con la fórmula de Heun tienen al menos dos decimales exactos, y coinciden aproximadamente con **$E=O(h^3)$**

9.2.2 Instrumentación computacional de la fórmula de Heun para dos E. D. O. de primer orden

Una función para calcular la solución de un sistema de dos ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio con la fórmula de Heun. En cada llamada, la función entrega el siguiente punto de la solución.

```
function [x,y,z]=heun2(f, g, x, y, z, h)
k1y=h*f(x,y,z)
k1z=h*g(x,y,z)
k2y=h*f(x+h,y+k1y,z+k1z)
k2z=h*g(x+h,y+k1y,z+k1z)
y=y+0.5*(k1y+k2y);
z=z+0.5*(k1z+k2z);
x=x+h;
```

Un programa en MATLAB para calcular 20 puntos espaciados en una distancia 0.1 del ejercicio anterior usando la fórmula de Heun

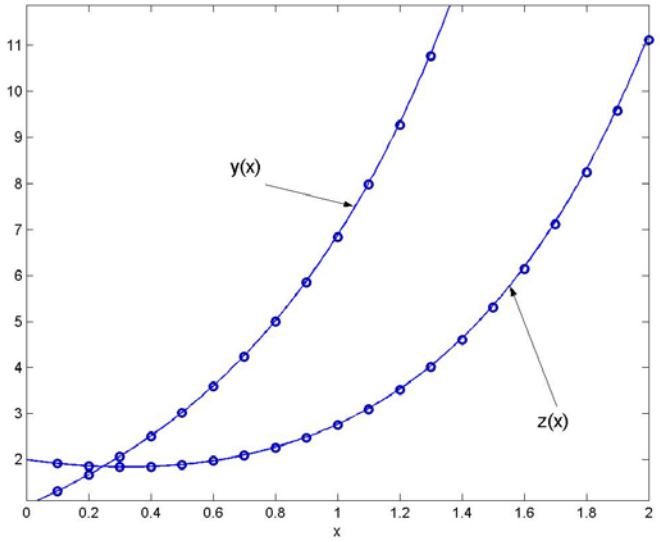
```
f=inline('x+y+z');
g=inline('-x+y-z');
x=0;
y=1;
z=2;
h=0.1;
for i=1:20
  [x,y,z]=heun2(f, g, x, y, z, h);
  u(i)=x;
  v(i)=y;
  w(i)=z;
end
```

Suponga que el programa se almacenó con el nombre **edo**. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta

```
>> edo
>> hold on;
>> plot(u, v, 'o');                                   u, v, w contienen los puntos calculados
>> plot(u, w, 'o');
>> [y, z]=dsolve('Dy-x-y-z=0,Dz+x-y+z=0','y(0)=1,z(0)=2','x')      Solución analítica
```

```
y =
    3/4*exp(2^(1/2)*x)-3/4*2^(1/2)*exp(-2^(1/2)*x)+3/4*2^(1/2)*exp(2^(1/2)*x)+3/4*exp(-
    2^(1/2)*x)-1/2
z =
    3/4*exp(2^(1/2)*x)+3/4*exp(-2^(1/2)*x)-x+1/2

>> ezplot(y, [0,2]);                                    Superponer la solución analítica
>> ezplot(z, [0,2]);
```



Solución analítica y la solución numérica para el ejemplo anterior
Se observa una coincidencia aceptable.

9.2.3 Fórmula de Runge-Kutta para dos EDO de primer orden y condiciones en el inicio

Las fórmulas de Runge-Kutta igualmente se pueden extender a sistemas de dos o más EDO's de primer orden con condiciones en el inicio.

Analizamos el caso de dos ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio, en las que **y'(x)** y **z'(x)** aparecen en forma explícita

$$F(x, y, z, y') = 0, \quad y(x_0) = y_0$$
$$G(x, y, z, z') = 0, \quad z(x_0) = z_0$$

Se pueden escribir en la notación para uso de los métodos numéricos

$$y' = f(x, y, z), \quad y(x_0) = y_0$$
$$z' = g(x, y, z), \quad z(x_0) = z_0$$

Definición: **Fórmula de Runge-Kutta de cuarto orden para un sistema de dos EDO de primer orden con condiciones en el inicio**

$$K_{1,y} = hf(x_i, y_i, z_i)$$
$$K_{1,z} = hg(x_i, y_i, z_i)$$
$$K_{2,y} = hf(x_i + h/2, y_i + K_{1,y}/2, z_i + K_{1,z}/2)$$
$$K_{2,z} = hg(x_i + h/2, y_i + K_{1,y}/2, z_i + K_{1,z}/2)$$
$$K_{3,y} = hf(x_i + h/2, y_i + K_{2,y}/2, z_i + K_{2,z}/2)$$
$$K_{3,z} = hg(x_i + h/2, y_i + K_{2,y}/2, z_i + K_{2,z}/2)$$
$$K_{4,y} = hf(x_i + h, y_i + K_{3,y}, z_i + K_{3,z})$$
$$K_{4,z} = hg(x_i + h, y_i + K_{3,y}, z_i + K_{3,z})$$

$$y_{i+1} = y_i + \frac{1}{6}(K_{1,y} + 2K_{2,y} + 2K_{3,y} + K_{4,y})$$

$$z_{i+1} = z_i + \frac{1}{6}(K_{1,z} + 2K_{2,z} + 2K_{3,z} + K_{4,z})$$

$$x_{i+1} = x_i + h, \qquad i = 0, 1, 2, \ldots$$

$$E = O(h^5), \quad x_i \le z \le x_{i+1}$$ (Error de truncamiento en cada paso)

9.2.4 Instrumentación computacional de la fórmula de Runge-Kutta para dos EDO de primer orden

Una función para calcular la solución de un sistema de dos ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio con la fórmula de Runge-Kutta de cuarto orden. En cada llamada, la función entrega el siguiente punto de la solución.

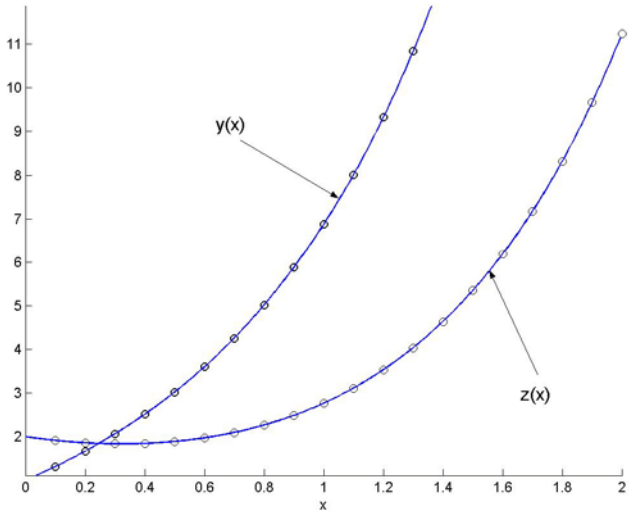
```
function [x,y,z]=rk42(f,g,x,y,z,h)
k1y=h*f(x,y,z);
k1z=h*g(x,y,z);
k2y=h*f(x+h/2,y+k1y/2,z+k1z/2);
k2z=h*g(x+h/2,y+k1y/2,z+k1z/2);
k3y=h*f(x+h/2,y+k2y/2,z+k2z/2);
k3z=h*g(x+h/2,y+k2y/2,z+k2z/2);
k4y=h*f(x+h,y+k3y,z+k3z);
k4z=h*g(x+h,y+k3y,z+k3z);
y=y+1/6*(k1y+2*k2y+2*k3y+k4y);
z=z+1/6*(k1z+2*k2z+2*k3z+k4z);
x=x+h;
```

Un programa en MATLAB para calcular 20 puntos espaciados en una distancia h = 0.1 del ejercicio anterior usando la fórmula de Runge-Kutta de cuarto orden

```
f=inline('x+y+z');
g=inline('-x+y-z');
x=0;
y=1;
z=2;
h=0.1;
for i=1:20
  [x,y,z]=rk42(f, g, x, y, z, h);
  u(i)=x;
  v(i)=y;
  w(i)=z;
end
```

Suponga que el programa se almacenó con el nombre **edo**. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta

```
>> edo
>> hold on;
>> plot(u, v, 'o');                          u, v, w contienen los puntos calculados
>> plot(u, w, 'o');
>> [y, z]=dsolve('Dy-x-y-z=0,Dz+x-y+z=0','y(0)=1,z(0)=2','x')      Solución analítica
  y =
    3/4*exp(2^(1/2)*x)-3/4*2^(1/2)*exp(-2^(1/2)*x)+3/4*2^(1/2)*exp(2^(1/2)*x)+3/4*exp(-
    2^(1/2)*x)-1/2
  z =
    3/4*exp(2^(1/2)*x)+3/4*exp(-2^(1/2)*x)-x+1/2
>> ezplot(y, [0,2]);                          Superponer la solución analítica
>> ezplot(z, [0,2]);
```



Solución analítica y la solución numérica para el ejemplo anterior
Se observa una alta coincidencia entre ambas soluciones

9.3 Ecuaciones diferenciales ordinarias de mayor orden y condiciones en el inicio

Mediante sustituciones estas ecuaciones se transforman en sistemas de ecuaciones diferenciales ordinarias de primer orden con condiciones en el inicio y se aplican los métodos numéricos como en la sección anterior.

Analizamos el caso de una ecuación diferencial ordinaria de segundo orden con condiciones en el inicio, en la que **y'(x)** y **y''(x)** aparecen en forma explícita

$$G(x, y, y', y'') = 0, \quad y(x_0) = y_0, \quad y'(x_0) = y'_0$$

Mediante la sustitución

$$z = y'$$

Se tiene

$$G(x, y, z, z') = 0$$

Se puede escribir como un sistema de dos ecuaciones diferenciales de primer orden siguiendo la notación anterior:

$$y' = f(x, y, z) = z$$
$$z' = g(x, y, z)$$ expresión que se obtiene despejando **z'** de **G**

Con las condiciones iniciales

$$y(x_0) = y_0$$
$$z(x_0) = y'_0 = z_0$$

Es un sistema de dos ecuaciones diferenciales de primer orden con condiciones en el inicio.

Ejemplo. *Calcule un punto de la solución de la siguiente ecuación diferencial de segundo orden con condiciones en el inicio, con la fórmula de Runge-Kutta de cuarto orden,* ***h = 0.1***

$$y'' - y' - x + y + 1 = 0, \quad y(0) = 1, \quad y'(0) = 2$$

Solución

Mediante la sustitución **z = y'** se obtiene

$$z' - z - x + y + 1 = 0$$

Constituyen un sistema de dos ecuaciones diferenciales de primer orden que se puede escribir

$$y' = f(x,y,z) = z, \qquad y(0) = 1$$
$$z' = g(x,y,z) = x - y + z - 1, \qquad z(0) = 2$$

Cálculo de los puntos de la solución

i=0: $x_0 = 0, y_0 = 1, z_0 = 2$

$K_{1,y} = hf(x_0, y_0, z_0) = 0.1\ f(0, 1, 2) = 0.1\ (2) = 0.2$

$K_{1,z} = hg(x_0, y_0, z_0) = 0.1\ g(0, 1, 2) = 0.1\ (0 - 1 + 2 - 1) = 0$

$K_{2,y} = hf(x_0 + h/2, y_0 + K_{1,y}/2, z_0 + K_{1,z}/2) = 0.1f(0.05, 1.1, 2) = 0.1(\ 2) = 0.2$

$K_{2,z} = hgx_0 + h/2, y_0 + K_{1,y}/2, z_0 + K_{1,z}/2) = 0.1g(0.05, 1.1, 2) = 0.1(0.05-1.1+2-1) = -0.005$

$K_{3,y} = hf(x_0 + h/2, y_0 + K_{2,y}/2, z_0 + K_{2,z}/2) = 0.1\ f(\ 0.05, 1.1, 1.9975) = 0.1998$

$K_{3,z} = hg(x_0 + h/2, y_0 + K_{2,y}/2, z_0 + K_{2,z}/2) = 0.1\ g(\ 0.05, 1.1, 1.9975) = -0.0052$

$K_{4,y} = hf(x_0 + h, y_0 + K_{3,y}, z_0 + K_{3,z}) = 0.1\ f(0.1, 1.1998, 1.9948) = 0.1995$

$K_{4,z} = hgx_0 + h, y_0 + K_{3,y}, z_0 + K_{3,z}) = 0.1\ g0.1, 1.1998, 1.9948) = -0.0105$

$$y_1 = y_0 + \frac{1}{6}(K_{1,y} + 2K_{2,y} + 2K_{3,y} + K_{4,y}) = 1 + \frac{1}{6}[0.2+2(0.2)+2(0.1998)+0.1995] = 1.1998$$

$$z_1 = z_0 + \frac{1}{6}(K_{1,z} + 2K_{2,z} + 2K_{3,z} + K_{4,z}) = 2 + \frac{1}{6}[0+2(-0.005)+2(-0.0052)-0.0105]=1.9948$$

$x_1 = x_0 + h = 0 + 0.1 = 0.1$

9.3.1 Instrumentación computacional

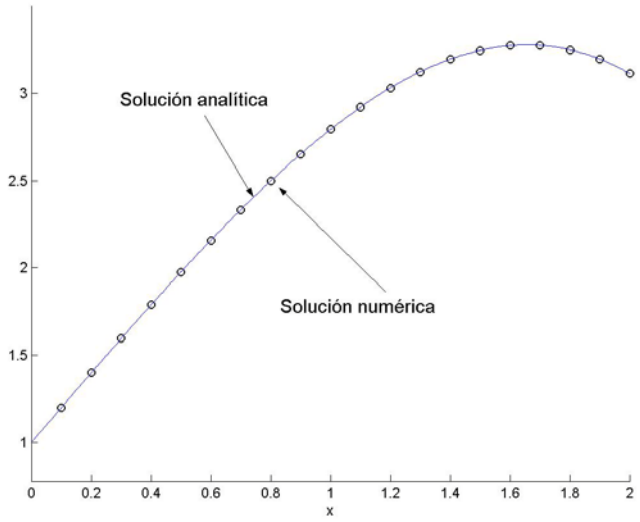
Al transformar la ecuación diferencial de segundo orden a un sistema de dos ecuaciones diferenciales de primer orden se pueden usar las mismas funciones desarrolladas anteriormente. Para el ejemplo anterior se usará función **rk42**

Un programa en MATLAB para calcular 20 puntos de la solución, espaciados en una distancia 0.1 para el ejercicio anterior usando la fórmula de Runge-Kutta de cuarto orden

```
f=inline('0*x + 0*y + z');
g=inline('x - y + z - 1');
x=0;
y=1;
z=2;
h=0.1;
for i=1:20
  [x,y,z]=rk42(f, g, x, y, z, h);
  u(i)=x;
  v(i)=y;
end
```

Suponga que el programa se almacenó con el nombre **edo**. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta

```
>> edo
>> hold on;
>> plot(u, v, 'o');                          % u, v contienen los puntos calculados
>> y = dsolve('D2y-Dy-x+y+1=0','y(0)=1,Dy(0)=2','x')    % Solución analítica
     y =
       x+1/3*3^(1/2)*exp(1/2*x)*sin(1/2*3^(1/2)*x)+exp(1/2*x)*cos(1/2*3^(1/2)*x)
>> ezplot(y, [0,2]);                         % Superponer la solución analítica
```



Solución analítica y la solución numérica para el ejemplo anterior
Se observa una alta coincidencia entre ambas soluciones

9.4 Ecuaciones diferenciales ordinarias no lineales

Los métodos numéricos pueden aplicarse igualmente para calcular la solución aproximada de ecuaciones diferenciales ordinarias no lineales, para las cuales no es posible o pudiese ser muy laborioso obtener la solución analítica

Ejemplo. *Obtenga numéricamente la solución de la ecuación*
$y'' + yy' - x + y - 3 = 0,\ y(0) = 1,\ y'(0) = 2,\ 0 \le x \le 2$

Solución

Mediante la sustitución **z = y',** se obtiene
z' + yz - x + y - 3 = 0
Se obtiene un sistema de dos ecuaciones diferenciales de primer orden que se puede escribir
y' = f(x,y,z) = z, **y(0) = 1**
z' = g(x,y,z) = x – y - yz + 3, **z(0) = 2**

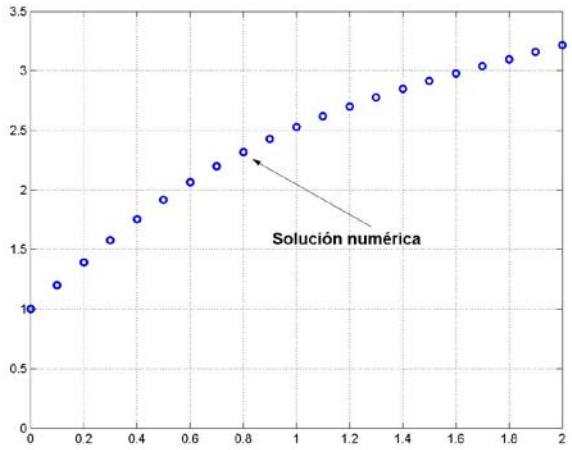
Usamos la misma función **RK42** para resolver el ejemplo. Previamente es transformada a un sistema de dos ecuaciones diferenciales de primer orden.

Un programa en MATLAB para calcular 20 puntos espaciados en una distancia 0.1 del ejercicio anterior usando la fórmula de Runge-Kutta de cuarto orden

```
f=inline('0*x + 0*y + z');
g=inline('x - y - y*z + 3');
x=0;
y=1;
z=2;
h=0.1;
m=20;
for i=1:m
  [x,y,z]=rk42(f, g, x, y, z, h);
  u(i)=x;
  v(i)=y;
end
```

Suponga que el programa se almacenó con el nombre **edo**. Los siguientes comandos permiten visualizar la solución y compararla con la solución analítica exacta

```
>> edo
>> hold on;
>> plot(u, v, 'o');                          % u, v contienen los puntos calculados
>> y = dsolve('D2y+y*Dy-x+y-3','y(0)=1','Dy(0)=2','x')
   Warning: Explicit solution could not be found        MATLAB no pudo encontrar
                                                        la solución analítica
```



Solución numérica calculada para el ejemplo anterior

9.5 Convergencia y estabilidad numérica

Los métodos numéricos que se utilizan para resolver una ecuación diferencial, como se muestra en los ejemplos anteriores, proporcionan una solución discreta aproximada. Para algunas ecuaciones diferenciales ordinarias, únicamente se tiene esta solución discreta por lo que es de interés verificar su convergencia.

La convergencia numérica puede hacerse variando el parámetro **h** del método numérico seleccionado y cuantificando la tendencia de algunos puntos de control de la solución calculada

Adicionalmente, es importante verificar si la solución obtenida es muy sensible a los errores en la formulación de la ecuación diferencial o en la condición inicial. Se puede detectar esta situación calculando numéricamente el problema original y el problema con alguna perturbación. Si la solución cambia significativamente puede interpretarse que el problema no está bien planteado.

9.6 Ecuaciones diferenciales ordinarias con condiciones en los bordes

En esta sección revisaremos los métodos numéricos para resolver ecuaciones diferenciales ordinarias para las cuales se proporcionan condiciones iniciales en los bordes, siendo de interés conocer la solución en el interior de esta región, como en el siguiente ejemplo:

$y'' - y' + y - 2e^x - 3 = 0,\ y(0) = 1, y(1) = 5,\ 0 \le x \le 1$

9.6.1 Método de prueba y error (método del disparo)

Una opción para obtener la solución numérica consiste en realizar varios intentos suponiendo una condición adicional en el inicio para poder usar los métodos vistos anteriormente. Para el ejemplo anterior probamos:

$y'' - y' + y - 2e^x - 3 = 0,\ y(0) = 1, y'(0) = 1,\ 0 \le x \le 1$

Esta es ahora una ecuación diferencial de segundo orden con condiciones en el inicio, la cual se puede re-escribir como dos ecuaciones diferenciales de primer orden:

$y' = f(x,y,z) = z,$ $y(0) = 1$
$z' = g(x,y,z) = 2e^x - y + z + 3,$ $z(0) = 1$

Aquí se puede aplicar alguno de los métodos estudiados (Heun, Runge-Kutta, etc.). El cálculo debe continuar hasta llegar al otro extremo del intervalo de interés. Entonces se puede comparar el resultado obtenido en el extremo derecho con el dato dado para ese borde **y(1) = 5**. Esto permite corregir la condición inicial supuesta y volver a calcular todo nuevamente. Este procedimiento se puede repetir varias veces.

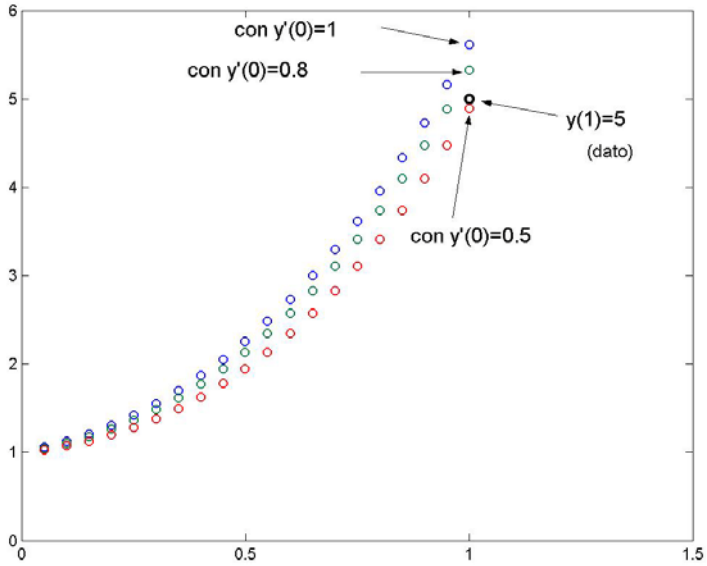
En la siguiente figura se observan tres intentos con el método de Runge-Kutta de cuarto orden probando con valores iniciales **z(0) = y'(0) = 1, 0.5, 0.8**

Usamos la conocida función **rk42** para resolver el ejemplo. El siguiente programa calcula 20 puntos de la solución espaciados en una distancia 0.05

```
f=inline('0*x+0*y+z');
g=inline('2*exp(x) - y + z+3');
x=0;
y=1;
z=1;              % este valor es modificado en cada intento
h=0.05;
m=20;
for i=1:m
  [x,y,z] = rk42(f, g, x, y, z, h);
  u(i)=x;
  v(i)=y;
end
```

Suponga que el programa se almacenó con el nombre **ed5**. Los siguientes comandos permiten visualizar la solución en los intentos realizados

```
>> ed5
>> hold on;
>> plot(u, v, 'o'), grid on, hold on        % u, v contienen los puntos calculados
```



Tres intentos con el método de Prueba y Error para el ejemplo anterior

Para sistematizar la elección del valor inicial es preferible usar una interpolación para asignar el siguiente valor de **y'(0)** en los siguientes intentos.

Sean $\mathbf{y'_a, y'_b}$ valores elegidos para **y'(0)** en dos intentos realizados
$\mathbf{y_a, y_b}$ valores obtenidos para **y** en el extremo derecho del intervalo, en los intentos
$\mathbf{y_n}$ valor suministrado como dato para el extremo derecho del intervalo
$\mathbf{y'_0}$ nuevo valor corregido para **y'(0)** para realizar un nuevo intento

Usamos una recta para predecir el valor para $\mathbf{y'_0}$**:**

$$y_b - y_n = \frac{y_b - y_a}{y'_b - y'_a}(y'_b - y'_0) \;\Rightarrow\; y'_0 = y'_b - \frac{y'_b - y'_a}{y_b - y_a}(y_b - y_n)$$

Para el ejemplo anterior, se tienen

$\mathbf{y'_a = 1}$
$\mathbf{y'_b = 0.5}$
$\mathbf{y_a = 5.6142}$ (valor obtenido en el extremo derecho, con $\mathbf{y'_a = 1}$)
$\mathbf{y_b = 4.8891}$ (valor obtenido en el extremo derecho, con $\mathbf{y'_b = 0.5}$)
$\mathbf{y_n = 5}$ (dato)

Con los que se obtiene

$$y'_0 = 0.5 - \frac{0.5 - 1}{4.8891 - 5.6142}(4.8891 - 5) = 0.5765$$

Al realizar la siguiente prueba con este valor de **y'(0)** se comprueba que la solución calculada está muy cerca de la solución analítica. Se puede verificar que el punto final calculado $\mathbf{y(x_n)}$ coincide en cinco decimales con el dato suministrado y(1) = 5

Este método también se puede usar para resolver ecuaciones diferenciales ordinarias **no lineales** con condiciones en los bordes.

9.6.2 Método de diferencias finitas

Este es un enfoque más general para resolver ecuaciones diferenciales ordinarias con condiciones en los bordes. Consiste en sustituir las derivadas por aproximaciones de diferencias finitas. La ecuación resultante se denomina ecuación de diferencias y puede resolverse por métodos algebraicos.

Es importante usar en la sustitución aproximaciones del mismo orden para las derivadas, de tal manera que la ecuación de diferencias tenga el error de truncamiento similar en cada término.

Aproximaciones de diferencias finitas de segundo orden **$O(h^2)$** conocidas:

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2)$$

$$y''_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2)$$

Ejemplo. *Sustituya las derivadas por diferencias finitas en la EDO del ejemplo anterior*

$$y'' - y' + y - 2e^x - 3 = 0, \quad y(0) = 1, y(1) = 5$$

La sustitución convierte la ecuación original en una ecuación "discretizada":

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{y_{i+1} - y_{i-1}}{2h} + y_i - 2e^{x_i} - 3 = 0, \; i = 1, 2, ..., n-1$$

Siendo **n** la cantidad de divisiones espaciadas en **h** en que se ha dividido el intervalo **$0 \le x \le 1$**

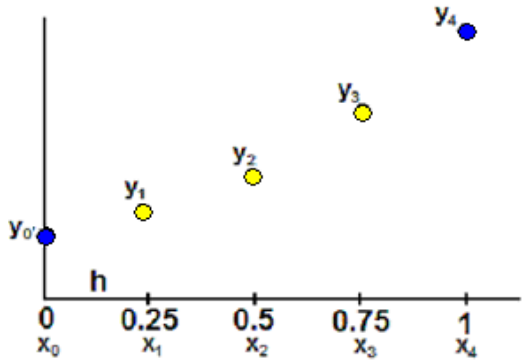
La ecuación resultante se denomina ecuación de diferencias con error de truncamiento **$O(h^2)$**.

Esta ecuación es consistente pues su límite, cuando **$h \to 0$** es la ecuación diferencial original.

Para facilitar los cálculos es conveniente expresar la ecuación de diferencias en forma estándar agrupando términos:

$$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2 e^{x_i} + 6h^2, \; i = 1, 2, ..., n-1$$

Para describir el uso de esta ecuación, supondremos que **h=0.25.** En la realidad debería ser más pequeño para que el error de truncamiento se reduzca. Con este valor de **h** el problema se puede visualizar de la siguiente manera



$y_0 = y(0) = 1,$ (dato en el borde izquierdo)
$y_4 = y(1) = 5,$ (dato en el borde derecho)
y_1, y_2, y_3 , son los puntos que se calcularán

A continuación se aplica la ecuación de diferencias en los puntos especificados

$$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2 e^{x_i} + 6h^2, \; i = 1, 2, 3$$

i=1: $(2+0.25)y_0 + (2(0.25^2)-4)y_1 + (2-0.25)y_2 = 4(0.25^2)\ e^{0.25} + 6(0.25^2)$

$$-3.875y_1 + 1.75y_2 = -1.5540$$

i=2: $(2+0.25)y_1 + (2(0.25^2)-4)y_2 + (2-0.25)y_3 = 4(0.25^2)\ e^{0.5} + 6(0.25^2)$

$$2.25y_1 - 3.875y_2 + 1.75y_3 = 0.7872$$

i=3: $(2+0.25)y_2 + (2(0.25^2)-4)y_3 + (2-0.25)y_4 = 4(0.25^2)\ e^{0.75} + 6(0.25^2)$

$$2.25y_2 - 3.875y_3 = -7.9628$$

Estas tres ecuaciones conforman un sistema lineal

$$\begin{bmatrix} -3.875 & 1.75 & 0 \\ 2.25 & -3.875 & 1.75 \\ 0 & 2.25 & -3.875 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1.5540 \\ 0.7872 \\ -7.9628 \end{bmatrix}$$

Cuya solución es

$$y_1 = 1.3930,\ y_2 = 2.1964,\ y_3 = 3.4095$$

En el siguiente gráfico se visualizan los resultados calculados

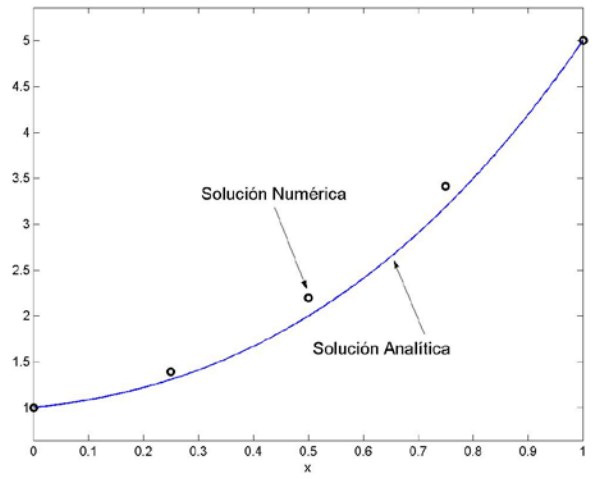
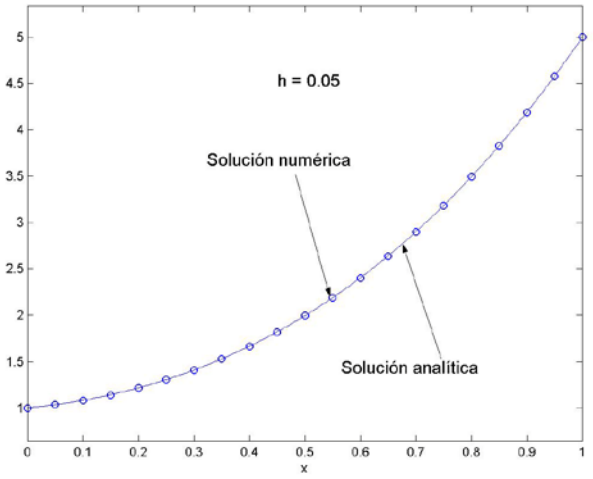


Fig. Comparación de la solución numérica y la solución analítica

La aproximación no es muy buena. Esto se debe a que la cantidad de puntos es muy pequeña. Si se desea una mejor aproximación se debe reducir **h.**

El siguiente gráfico muestra la solución calculada con **n=20.** A este valor de n le corresponde **h=0.05**.

Se obtiene un sistema de 19 ecuaciones lineales cuyas incógnitas son los puntos interiores. Se observa que los resultados tienen una aproximación aceptable



Las ecuaciones que se obtienen con estas ecuaciones de diferencias conforman un sistema tridiagonal de ecuaciones lineales. Estos sistemas pueden resolverse con un algoritmo específico muy eficiente con medida T(n) = O(n) el cual fue utilizada en el capítulo del trazador cúbico.

9.6.3 Instrumentación computacional

La siguiente instrumentación del método de diferencias finitas corresponde a la solución de una ecuación de diferencias después de ser escrita en forma estandarizada

$$(P)y_{i-1} + (Q)y_i + (R)y_{i+1} = (S),\ i = 1, 2, 3, \ldots, n-1$$

En donde **P, Q, R, S** son expresiones que pueden contener x_i y **h**, siendo **x** la variable independiente. La función entrega los puntos calculados de la solución **x, y** en los vectores **u, v**. Los puntos en los bordes: (x_0, y_0) y (x_n, y_n) son dados como datos, **n** es la cantidad de sub intervalos.

```
function [u,v] = edodif(P, Q, R, S, x0, y0, xn, yn, n)
% Método de Diferencias Finitas
% Solución de una EDO con condiciones constantes en los bordes
h=(xn-x0)/n;
clear a b c d;
for i=1:n-1
  x=x0+h*i;
  a(i)=eval(P);                 % Diagonales del sistema tridiagonal
  b(i)=eval(Q);
  c(i)=eval(R);
  d(i)=eval(S);
  u(i)=x;
end
d(1)=d(1)-a(1)*y0;
d(n-1)=d(n-1)-c(n-1)*yn;
v=tridiagonal(a,b,c,d);         % Solucion del sistema tridiagonal
```

Ejemplo. *Escribir un programa que usa la función EDODIF para resolver el ejemplo anterior con* ***n=20***

Forma estándar de la ecuación de diferencias para el ejemplo anterior

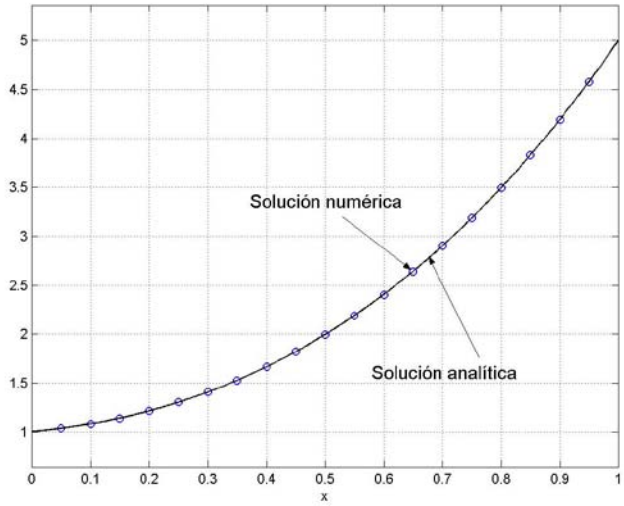
$$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2 e^{x_i} + 6h^2,\ i=1, 2,..., n-1;\ y_0=y(0)=1;\ y_n=y(1)=5$$

```
n=20;
x0=0;
y0=1;
xn=1;
yn=5;
P='2+h';
Q='2*h^2-4';
R='2-h';
S='4*h^2*exp(x)+6*h^2';
[u,v]=edodif(P,Q,R,S,x0,y0,xn,yn,n);
```

Suponer que este programa es almacenado con el nombre P6.

Escribimos las siguientes líneas para activarlo desde la ventana de comandos

```
>> p6;
>> plot(u,v,'o');
>> y=dsolve('D2y-Dy+y-2*exp(x)-3=0','y(0)=1','y(1)=5','x');    %Solución analítica
>> hold on, ezplot(y,[0,1])
```



Comparación de las soluciones analítica y numérica para el ejemplo anterior

9.6.4 Ecuaciones diferenciales ordinarias con condiciones en los bordes con derivadas

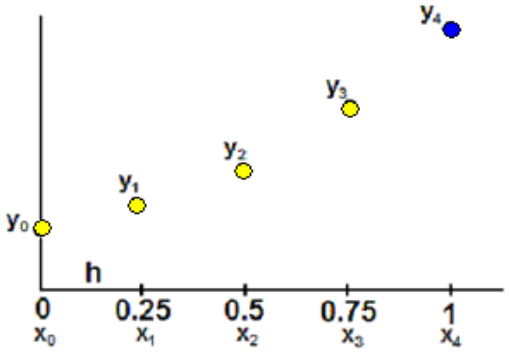
Consideremos una variación del problema anterior:

$$y'' - y' + y - 2e^x - 3 = 0,\ y'(0) = 0.5,\ y(1) = 5,\ 0 \le x \le 1$$

Luego de sustituir las derivadas y simplificar se tiene la ecuación de diferencias como antes:

$$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2 e^{x_i} + 6h^2,\ i = 1, 2, ..., n-1$$

Por simplicidad, consideremos que $h = 0.25$



El punto y_0 también es desconocido. Ahora aplicamos la ecuación de diferencias en cada uno de los puntos desconocidos, incluyendo el punto y_0

$$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2 e^{x_i} + 6h^2,\ i = 0, 1, 2, 3$$

i=0: $(2+0.25)y_{-1} + (2(0.25^2)-4)y_0 + (2-0.25)y_1 = 4(0.25^2)\ e^0 + 6(0.25^2)$

$$2.25y_{-1} - 3.875y_0 + 1.75y_1 = 0.6250$$

i=1: $(2+0.25)y_0 + (2(0.25^2)-4)y_1 + (2-0.25)y_2 = 4(0.25^2)\ e^{0.25} + 6(0.25^2)$

$$2.25y_0 - 3.875y_1 + 1.75y_2 = 0.6960$$

i=2: $(2+0.25)y_1 + (2(0.25^2)-4)y_2 + (2-0.25)y_3 = 4(0.25^2)\ e^{0.5} + 6(0.25^2)$

$$2.25y_1 - 3.875y_2 + 1.75y_3 = 0.7872$$

i=3: $(2+0.25)y_2 + (2(0.25^2)-4)y_3 + (2-0.25)y_4 = 4(0.25^2)\ e^{0.75} + 6(0.25^2)$

$$2.25y_2 - 3.875y_3 = -7.8475$$

Se obtiene un sistema de cuatro ecuaciones con cinco incógnitas: $y_{-1},\ y_0,\ y_1,\ y_2,\ y_3$.
Se ha introducido un punto ficticio y_{-1}

Usamos una aproximación central de segundo orden para la primera derivada

$$y'_0 = 0.5 = \frac{y_1 - y_{-1}}{2h}$$ de donde se obtiene que $y_{-1} = y_1 - 2h(0.5) = y_1 - 0.25$

Esto permite eliminar el punto ficticio y_{-1} en la primera ecuación anterior:

$$\textbf{i=0:} \quad 2.25(y_1 - 0.25) - 3.875y_0 + 1.75y_1 = 0.6250 \Rightarrow -3.875y_0 + 4y_1 = 1.1875$$

Finalmente, el sistema se puede escribir:

$$\begin{bmatrix} -3.875 & 4 & 0 & 0 \\ 2.25 & -3.875 & 1.75 & 0 \\ 0 & 2.25 & -3.875 & 1.75 \\ 0 & 0 & 2.25 & -3.875 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1.1875 \\ 0.6960 \\ 0.7872 \\ -7.8475 \end{bmatrix}$$

Cuya solución es: $y_0 = 1.4738,\ y_1 = 1.7246,\ y_2 = 2.3216,\ y_3 = 3.3732$

El sistema resultante tiene forma tridiagonal, y por lo tanto se puede usar un algoritmo específico muy eficiente para resolverlo. La instrumentación se desarrolla más abajo.

Otra alternativa sería la aplicación de la ecuación de diferencias en los puntos interiores. Se obtendría un sistema de tres ecuaciones con las cuatro incógnitas: y_0, y_1, y_2, y_3. La cuarta ecuación se la obtendría con una fórmula de segundo orden para la derivada en el punto izquierdo definida con los mismos puntos desconocidos:

$$y'_0 = 0.5 = \frac{-3y_0 + 4y_1 - y_2}{2h}$$

El inconveniente es que el sistema final no tendrá la forma tridiagonal.

9.6.5 Instrumentación computacional

La siguiente instrumentación del método de diferencias finitas permite resolver problemas de tipo similar al ejemplo anterior. La ecuación de diferencias debe escribirse en forma estandarizada

$$(P)y_{i-1} + (Q)y_i + (R)y_{i+1} = (S)$$

En donde **P, Q, R, S** son expresiones que pueden contener x_i y **h**, siendo **x** la variable independiente. La función entrega los **n-1** puntos calculados de la solución **x, y** en los vectores **u, v**. Los siguientes datos son proporcionados en los bordes: y'_0 y y_n

```
function [u,v]=edodifdi(P,Q,R,S,x0,dy0,xn,yn,n)
% Método de Diferencias Finitas
% Solución de una EDO con una derivada a la izquierda
% y una condición constante a la derecha
h=(xn-x0)/n;
clear a b c d;
for i=1:n          % corresponde a las ecuaciones  i = 0, 1, 2, ..., n-1
  x=x0+h*(i-1);
  a(i)=eval(P);
  b(i)=eval(Q);
  c(i)=eval(R);
  d(i)=eval(S);
  u(i)=x;
end
x=h;
c(1)=c(1)+eval(P);
d(1)=d(1)+eval(P)*2*h*dy0;
d(n)=d(n)-c(n)*yn;
v=tridiagonal(a,b,c,d);      % solucion del sistema tridiagonal
```

Escriba un programa que usa la función EDODIFDI para resolver el ejemplo anterior

Forma estándar de la ecuación de diferencias para el ejemplo anterior, incluyendo la forma especial para la primera ecuación.

$(2+h)y_{i-1} + (2h^2-4)y_i + (2-h)y_{i+1} = 4h^2\,\mathbf{e}^{x_i} + 6h^2$, $\quad i = 1, 2,..., n-1$
$(\mathbf{P})y_{i-1} + (\mathbf{Q})y_i + (\mathbf{R})y_{i+1} = (\mathbf{S})$, $\quad i = 1, 2,..., n-1;$

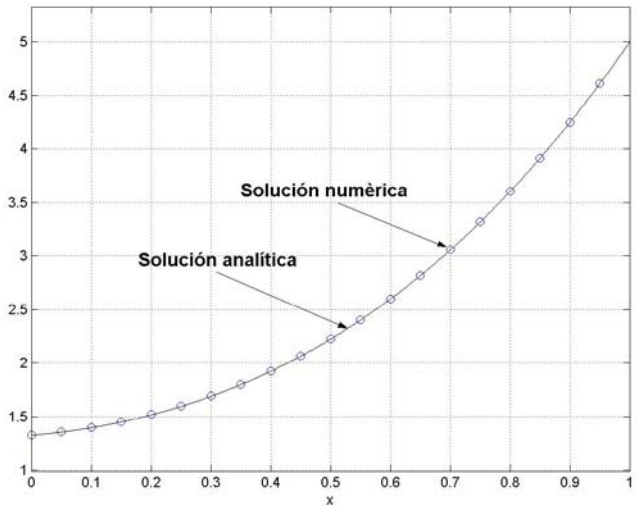
$(2+h)y_{-1} + (2h^2-4)y_0 + (2-h)y_1 = 4h^2\,\mathbf{e}^{x_0} + 6h^2$, $\quad i = 0$
$(2+h)(y_1 + 2hy'_0) + (2h^2-4)y_0 + (2-h)y_1 = 4h^2\,\mathbf{e}^{x_0} + 6h^2$, $\quad i = 0$
$(\mathbf{P})(y_1 + 2hy'_0) + (\mathbf{Q})y_0 + (\mathbf{R})y_1 = (\mathbf{S})$, $\quad i = 0$
$(\mathbf{Q})y_0 + (\mathbf{P} + \mathbf{R})y_1 = (\mathbf{S}) - (\mathbf{P})2hy'_0$, $\quad i = 0$

$y'_0=y'(0)=0.5$; $y_n=y(1)=5$

```
n=20;
x0=0;
dy0=0.5;
xn=1;
yn=5;
P='2+h';
Q='2*h^2-4';
R='2-h';
S='4*h^2*exp(x)+6*h^2';
[u,v]=edodifdi (P,Q,R,S,x0,dy0,xn,yn,n);
```

Suponer que este programa es almacenado con el nombre P7.
Escribimos las siguientes líneas para activarlo desde la ventana de comandos

```
>> p7;
>> plot(u,v,'o');
>> y=dsolve('D2y-Dy+y-2*exp(x)-3=0','Dy(0)=0.5','y(1)=5','x');
>> hold on, ezplot(y, [0,1])
```



Comparación de las soluciones analítica y numérica para el ejemplo anterior

9.6.6 Normalización del dominio de la E.D.O

Previamente a la aplicación de los métodos numéricos es conveniente normalizar la ecuación diferencial llevándola al dominio [0, 1] mediante las siguientes sustituciones. De esta manera se puede ver con claridad si el valor elegido para **h** es apropiado.

Ecuación diferencial original

$$\mathbf{F(x, y(x), y'(x), y''(x)) = 0,\ y(a) = u,\ y(b) = v,\ a \le x \le b}$$

Mediante las sustituciones:

$$\mathbf{x = (b - a)t + a:\quad t = 0 \Rightarrow x = a,\quad t = 1 \Rightarrow x = b}$$

$$\mathbf{\frac{dt}{dx} = \frac{1}{b - a}}$$

$$\mathbf{y'(x) = \frac{dy}{dx} = \frac{dy}{dt}\frac{dt}{dx} = \frac{1}{b - a}\frac{dy}{dt} = \frac{1}{b - a}y'(t)}$$

$$\mathbf{y''(x) = \frac{d^2y}{dx^2} = \frac{d}{dx}\left(\frac{dy}{dx}\right) = \frac{d}{dt}\left(\frac{dy}{dx}\right)\frac{dt}{dx} = \frac{d}{dt}\left(\frac{1}{b - a}\frac{dy}{dt}\right)\frac{dt}{dx} = \frac{1}{(b - a)^2}\frac{d^2y}{dt^2} = \frac{1}{(b - a)^2}y''(t)}$$

Se obtiene la ecuación diferencial normalizada

$$\mathbf{F(t, y(t), y'(t), y''(t)) = 0,\ y(0) = u, y(1) = v,\ 0 \le t \le 1}$$

9.7 Ejercicios con ecuaciones diferenciales ordinarias

1. Obtenga dos puntos de la solución de la siguiente ecuación diferencial utilizando tres términos de la Serie de Taylor. Use **h = 0.1**

$$y' - 2x + 2y^2 + 3 = 0,\ y(0)=1$$

2. Dada la siguiente ecuación diferencial ordinaria de primer orden

$$y' - 2y + 2x^2 - x + 3 = 0,\ y(0)=1.2$$

a) Obtenga dos puntos de la solución con la fórmula de Euler. Use **h = 0.1**
b) Obtenga dos puntos de la solución con la fórmula de Heun. Use **h = 0.1**
c) Obtenga dos puntos de la solución con la fórmula de Runge-Kutta de cuarto orden. Use **h = 0.1**
d) Compare con la solución exacta: $y(x) = x/2 + x^2 - 11/20\ e^{2x} + 7/4$

3. Al resolver una ecuación diferencial con un método numérico, el error de truncamiento tiende a acumularse y crecer. Use el método de Euler con **h=0.1** para calcular 10 puntos de la solución de:

$$y' - 2x + 5y - 1 = 0,\ y(0) = 2$$

Compare con la solución analítica $y(x) = 2x/5 + 47/25\ e^{-5x} + 3/25$. Observe que el error de truncamiento tiende a reducirse. Explique este comportamiento.

4. La solución exacta de la ecuación diferencial: $y' - 2xy = 1,\ y(0)=y_0$ es

$$y(x) = e^{x^2}\left(\frac{\sqrt{\pi}}{2}erf(x) + y_0\right),\ \text{donde}\ erf(x) = \frac{1}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$$

Encuentre **y(0.4)** de la ecuación diferencial: $y' - 2xy = 1,\ y(0)=1$

a) Con la fórmula de Runge-Kutta, **h=0.2**
b) Evaluando la solución exacta con la cuadratura de Gauss con dos puntos

5. Dada la siguiente ecuación diferencial ordinaria de segundo orden

$$y'' - y' - sen(x) + y + 1 = 0,\ y(0)=1.5,\ y'(0) = 2.5$$

a) Obtenga dos puntos de la solución con la fórmula de Heun. (**h = 0.1**)
b) Obtenga un punto de la solución con la fórmula de Runge-Kutta de cuarto orden. (**h = 0.1**)
c) Compare con la solución exacta. Obténgala con la función **dsolve** de MATLAB

6. Dada la siguiente ecuación diferencial

$$2y''(x) - 3y'(x) + 2x = 5,\ y(1) = 2,\ y(3) = 4$$

a) Normalice la ecuación diferencial en el intervalo **[0, 1]**
b) Use el método de diferencias finitas y obtenga la solución, **h = 0.2** en el intervalo normalizado.

10 ECUACIONES DIFERENCIALES PARCIALES

En este capítulo se estudiará el método de diferencias finitas aplicado a la resolución de ecuaciones diferenciales parciales.

Sea **u** una función que depende de dos variables independientes **x, y.** La siguiente ecuación es la forma general de una ecuación diferencial parcial de segundo orden:

$$F(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial x \partial y}) = 0$$

Una forma de clasificar estas ecuaciones es por su tipo: parabólicas, elípticas e hiperbólicas,

En forma similar a las ecuaciones diferenciales ordinarias, el método numérico que usaremos consiste en sustituir las derivadas por aproximaciones de diferencias finitas. El objetivo es obtener una ecuación denominada ecuación de diferencias que pueda resolverse por métodos algebraicos. Esta sustitución discretiza el dominio con un espaciamiento que debe elegirse.

10.1 Aproximaciones de diferencias finitas

Las siguientes son algunas aproximaciones de diferencias finitas de uso común para aproximar las derivadas en un punto x_i, y_j, siendo $\Delta x, \Delta y$ el espaciamiento entre los puntos de **x, y,** respectivamente. El término a la derecha en cada fórmula representa el orden del error de truncamiento.

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \qquad (8.1)$$

$$\frac{\partial u_{i,j}}{\partial y} = \frac{u_{i,j+1} - u_{i,j}}{\Delta y} + O(\Delta y) \qquad (8.2)$$

$$\frac{\partial u_{i,j}}{\partial y} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y} + O(\Delta y) \qquad (8.3)$$

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2(\Delta x)} + O(\Delta x)^2 \qquad (8.4)$$

$$\frac{\partial u_{0,j}}{\partial x} = \frac{-3u_{0,j} + 4u_{1,j} - u_{2,j}}{2(\Delta x)} + O(\Delta x)^2 \qquad (8.5)$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 \qquad (8.6)$$

$$\frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} + O(\Delta y)^2 \qquad (8.7)$$

10.2 Ecuaciones diferenciales parciales de tipo parabólico

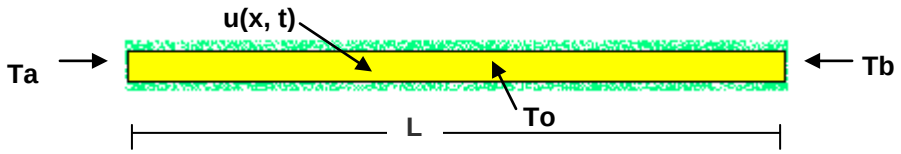
Estas ecuaciones se caracterizan porque en su dominio una de las variables no está acotada. Para aplicar el método de diferencias finitas usaremos un ejemplo básico para posteriormente interpretar los resultados obtenidos.

Ejemplo. *Resolver la ecuación de difusión en una dimensión con los datos suministrados*

$$u(x,t): \quad \frac{\partial^2 u}{\partial x^2} = k\frac{\partial u}{\partial t}$$

Suponer que ***u*** *es una función que depende de* ***x, t,*** *en donde* ***u*** *representa valores de temperatura,* ***x*** *representa posición, mientras que* ***t*** *es tiempo,*

Esta ecuación se puede asociar al flujo de calor en una barra muy delgada aislada transversalmente y sometida en los extremos a alguna fuente de calor. La constante **k** depende del material de la barra. La solución representa la distribución de temperaturas en cada punto **x** de la barra y en cada instante **t**



Ta, Tb son valores de temperatura de las fuentes de calor aplicadas en los extremo de la barra.

En este primer ejemplo suponer que son valores constantes. **To** es la temperatura inicial y **L** es la longitud de la barra.

Estas condiciones se pueden expresar de manera simbólica en un sistema de coordenadas

u(0, t) = Ta, t ≥ 0
u(L, t) = Tb, t ≥ 0
u(x, 0) = To, 0 < x < L

Para aplicar el método de diferencias finitas, debe discretizarse el dominio de **u** mediante una malla con puntos en dos dimensiones en la cual el eje horizontal representa la posición x_i mientras que el eje vertical representa el tiempo t_j.

$$u=u(x,t),\ 0 \le x \le L,\ t \ge 0 \quad \Rightarrow \quad u(x_i, t_j) = u_{i,j}\ ;\ i = 0, 1, ..., n;\ j = 0, 1, 2, ...$$

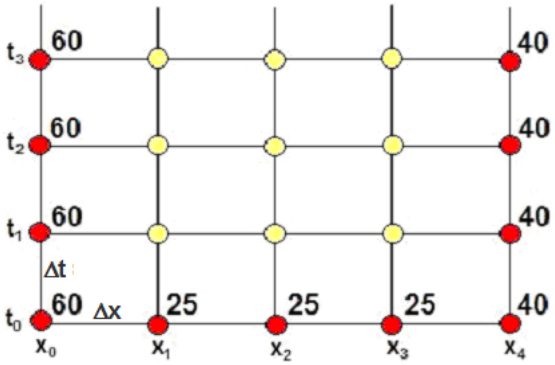
El método de diferencias finitas permitirá encontrar **u** en estos puntos.

Para el ejemplo supondremos los siguientes datos, en las unidades que correspondan

Ta = 60
Tb = 40
To = 25
k = 4
L = 1

Decidimos además, para simplificar la descripción del método que **Δx = 0.25, Δt = 0.1**

Con esta información se define el dominio de $u_{i,j}$. En la malla se representan los datos en los bordes y los puntos interiores que deberán calcularse:



10.2.1 Un esquema de diferencias finitas explícito

Para nuestro primer intento elegimos las fórmulas (**8.6**) y (**8.2**) para sustituir las derivadas de la ecuación diferencial

$$\frac{\partial^2 u}{\partial x^2} = k\frac{\partial u}{\partial t} \quad \Rightarrow \quad \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 = k\frac{u_{i,j+1} - u_{i,j}}{\Delta t} + O(\Delta t)$$

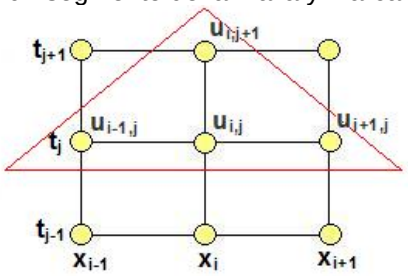
Se obtiene la ecuación de diferencias

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} = k\frac{u_{i,j+1} - u_{i,j}}{\Delta t}$$

Cuyo error de truncamiento es $\mathbf{T = O(\Delta x)^2 + O(\Delta t)}$. Si $\Delta x, \Delta t \to 0 \Rightarrow T \to 0$ y la ecuación de diferencias tiende a la ecuación diferencial parcial (contínua)

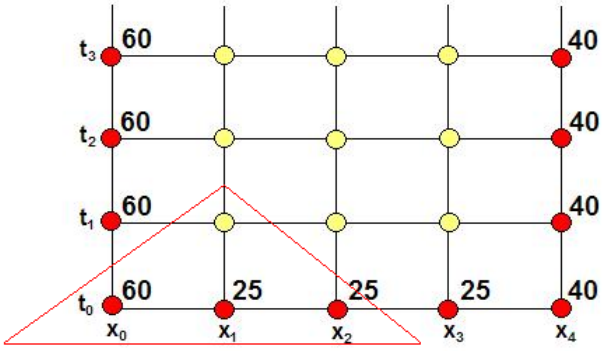
Para que esta sustitución sea consistente, es decir, que ambos términos de la ecuación tengan un error de truncamiento de orden similar, **Δt** debe ser menor que **Δx,** aproximadamente en un orden de magnitud.

Es conveniente analizar cuales puntos están incluidos en la ecuación de diferencias. Para esto consideramos un segmento de la malla y marcamos los puntos de la ecuación.



Los puntos que están incluidos en la ecuación de diferencia conforman un triángulo. Este triángulo puede colocarse en cualquier lugar de la malla asignando a **i, j** los valores apropiados.

Por ejemplo, si **i=1, j=0**, la ecuación de diferencias se ubica en el extremo inferior izquierdo de la malla. Los puntos en color rojo son los datos conocidos. Los puntos en amarillo son los puntos que deben calcularse.



Se puede observar que solo hay un punto desconocido en la ecuación. Por lo tanto, esta ecuación de diferencias proporciona un **método explícito** de cálculo**.** Esto significa que cada punto de la solución puede ser obtenerse en forma individual y directa cada vez que se aplica la ecuación.

Despejamos el punto desconocido $u_{i,j+1}$

$$u_{i,j+1} = \frac{\Delta t}{k(\Delta x)^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + u_{i,j}$$

Definiendo $\lambda = \frac{\Delta t}{k(\Delta x)^2} \quad \Rightarrow \quad u_{i,j+1} = \lambda(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + u_{i,j}$

La ecuación se puede escribir

$$u_{i,j+1} = \lambda u_{i-1,j} + (1-2\lambda)u_{i,j} + \lambda u_{i+1,j}\ ,\ \ i = 1, 2, 3;\ \ j = 0, 1, 2, \ldots$$

La forma final de la ecuación de diferencias se obtiene sustituyendo los datos:

$$\lambda = \frac{\Delta t}{k(\Delta x)^2} = \frac{0.1}{4(0.25)^2} = 0.4$$

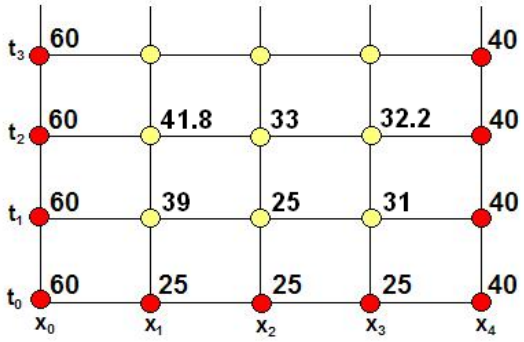
$$u_{i,j+1} = 0.4u_{i-1,j} + 0.2u_{i,j} + 0.4u_{i+1,j}\ ,\ \ i = 1, 2, 3;\ \ j = 0, 1, 2, \ldots$$

La solución se debe calcular sucesivamente para cada nivel t_j hasta donde sea de interés analizar la solución de la ecuación diferencial. Calculemos dos niveles de la solución:

j = 0, i = 1: $u_{1,1} = 0.4u_{0,0} + 0.2u_{1,0} + 0.4u_{2,0} = 0.4(60) + 0.2(25) + 0.4(25) = 39$
i = 2: $u_{2,1} = 0.4u_{1,0} + 0.2u_{2,0} + 0.4u_{3,0} = 0.4(25) + 0.2(25) + 0.4(25) = 25$
i = 3: $u_{3,1} = 0.4u_{2,0} + 0.2u_{3,0} + 0.4u_{4,0} = 0.4(25) + 0.2(25) + 0.4(40) = 31$

j = 1, i = 1: $u_{1,2} = 0.4u_{0,1} + 0.2u_{1,1} + 0.4u_{2,1} = 0.4(60) + 0.2(39) + 0.4(25) = 41.8$
i = 2: $u_{2,2} = 0.4u_{1,1} + 0.2u_{2,1} + 0.4u_{3,1} = 0.4(39) + 0.2(25) + 0.4(31) = 33$
i = 3: $u_{3,2} = 0.4u_{2,1} + 0.2u_{3,1} + 0.4u_{4,1} = 0.4(25) + 0.2(31) + 0.4(40) = 32.2$

En el siguiente gráfico se anotan los resultados para registrar el progreso del cálculo



Para que la solución tenga precisión aceptable, los incrementos **Δx** y **Δt** deberían ser mucho más pequeños, pero esto haría que la cantidad de cálculos involucrados sea muy grande para hacerlo manualmente.

10.2.2 Estabilidad del método de diferencias finitas

Existen diferentes métodos para determinar el crecimiento en el tiempo del error en el proceso de cálculo de la solución. Si el error no crece con el tiempo, el método es estable.

Criterio de Von Newman

Consiste en suponer que el error propagado $E_{i,j}$ en el tiempo en cada punto (x_i, t_j) tiene forma exponencial compleja:

$$E_{i,j} = e^{\sqrt{-1}\beta x_i + \alpha t_j},$$

Si se define el coeficiente de amplificación del error en iteraciones consecutivas:

$$M = \frac{E_{i,j+1}}{E_{i,j}} = \frac{e^{\sqrt{-1}\beta x_i + \alpha(t_j + \Delta t)}}{e^{\sqrt{-1}\beta x_i + \alpha t_j}} = e^{\alpha \Delta t}$$

Entonces, si $|e^{\alpha\Delta t}| \le 1$ el error no crecerá en el tiempo y el método es estable.

Ecuación de diferencias del método explícito:

$$u_{i,j+1} = \lambda(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + u_{i,j}, \quad \lambda = \frac{\Delta t}{k(\Delta x)^2}$$

Ecuación de los errores introducidos

$$e^{\sqrt{-1}\beta x_i + \alpha(t_j + \Delta t)} = \lambda(e^{\sqrt{-1}\beta(x_i + \Delta x) + \alpha t_j} - 2e^{\sqrt{-1}\beta x_i + \alpha t_j} + e^{\sqrt{-1}\beta(x_i - \Delta x) + \alpha t_j}) + e^{\sqrt{-1}\beta x_i + \alpha t_j}$$

Dividiendo por $e^{k\beta x_i + \alpha t_j}$

$$e^{\alpha \Delta t} = \lambda(e^{\sqrt{-1}\beta \Delta x} - 2 + e^{\sqrt{-1}\beta_i(-\Delta x)}) + 1$$

Sustituyendo las equivalencias y simplificando

$$e^{\pm\sqrt{-1}\beta \Delta x} = \cos(\beta \Delta x) \pm \sqrt{-1}\,\mathrm{sen}(\beta \Delta x)$$

Se obtiene

$$e^{\alpha \Delta t} = \lambda(2\cos(\beta \Delta x) - 2) + 1$$

La estabilidad está condicionada a: $|e^{\alpha \Delta t}| \leq 1$

$$|\lambda(2\cos(\beta \Delta x) - 2) + 1| \leq 1$$

Sustituyendo los valores extremos de $\cos(\beta \Delta x) = 1, \quad \cos(\beta \Delta x) = -1$

Se obtiene la restricción para que este método sea estable:

$$\lambda \leq \frac{1}{2} \quad \Rightarrow \quad \frac{\Delta t}{k(\Delta x)^2} \leq \frac{1}{2}$$

Si se cumple esta condición, el error propagado no crecerá.

Si se fija **k** y **Δx,** debería elegirse **Δt** para que se mantenga la condición anterior. Desde el punto de vista del error de truncamiento **Δt** debería ser aproximadamente un orden de magnitud menor que **Δx**.

Se puede verificar que si no se cumple esta condición, el método se hace inestable rápidamente y los resultados obtenidos son incoherentes.

10.2.3 Instrumentación computacional

El siguiente programa es una instrumentación en MATLAB para resolver el ejemplo anterior y es una referencia para aplicarlo a problemas similares. Los resultados obtenidos se muestran gráficamente.

Para la generalización es conveniente expresar la ecuación de diferencias en forma estándar

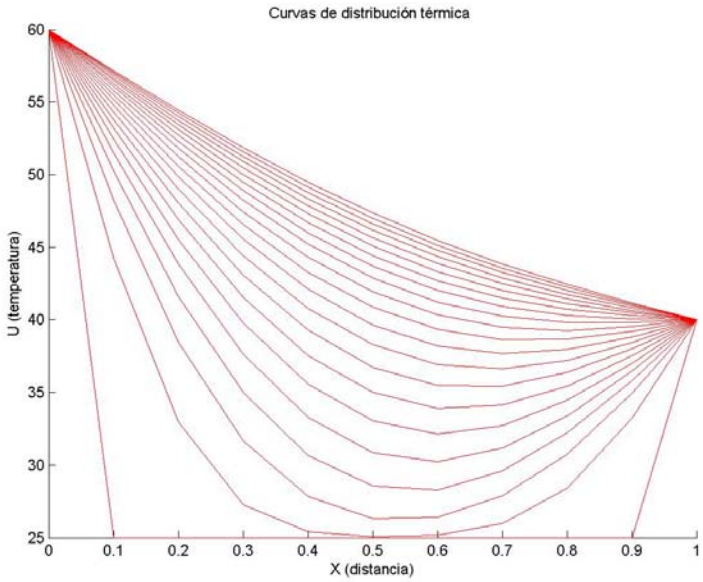
$$u_{i,\,j+1} = (P)\,u_{i-1,\,j} + (Q)u_{i,\,j} + (R)u_{i+1,j},\quad i = 1, 2, 3, \ldots, n-1;\quad j = 1, 2, 3, \ldots$$

En donde **P**, **Q**, **R** dependen de los datos de la ecuación que se desea resolver.
Para el ejemplo propuesto se tiene: $u_{i,j+1} = \lambda u_{i-1,j} + (1-2\lambda)u_{i,j} + \lambda u_{i+1,j}$,

```
%   Ecuación de diferencias estandarizada
%   U(i,j+1)=(P)U(i-1,j) + (Q)U(i,j) + (R)U(i-1,j)
%   P,Q,R son constantes evaluadas con los datos de la EDP
clf;
m=11;                % Número de puntos en x
n=100;               % Número de niveles en t
Ta=60; Tb=40;        % Condiciones en los bordes
To=25;               % Condición en el inicio
dx=0.1; dt=0.01;     % incrementos
L=1;                 % longitud
k=4;                 % dato especificado
clear x U;
U(1)=Ta;             % Asignación inicial
U(m)=Tb;
for i=2:m-1
  U(i)=To;
end
lambda = dt/(k*dx^2);
P=lambda;
Q=1-2*lambda;
R=lambda;
hold on;
title('Curvas de distribución térmica');
xlabel('X (distancia)');
ylabel('U (temperatura)');
x=0:dx:L;                % Coordenadas para el gráfico
plot(x,U,'r'); grid on;  % Distribución inicial
for j=1:n
   U=EDPDIF(P,Q,R,U,m);
    if mod(j,5)==0
      plot(x,U,'r');   % Para graficar curvas cada 5 niveles de t
      pause
   end
end

function u=EDPDIF(P,Q,R,U,m)
% Solución U(x,t) de una EDP con condiciones constantes en los bordes
% Método explícito de diferencias finitas
u(1)=U(1);
for i=2:m-1
  u(i)=P*U(i-1)+Q*U(i)+R*U(i+1);
end
u(m)=U(m);
```

Almacenar la función y el programa y ejecutarlo para obtener el gráfico siguiente



Curvas de distribución térmica para el ejemplo. La distribución tiende a una forma estable.

La ejecución controlada con el comando **pause** permite visualizar el progreso de los cálculos

Con esta instrumentación se puede verificar que al incrementar Δt a un valor mayor que 0.02, ya no se cumple la condición de convergencia y el método se hace inestable. Los resultados son incoherentes

10.2.4 Un esquema de diferencias finitas implícito

En un segundo intento elegimos las aproximaciones (8.6) y (8.3) para sustituir las derivadas de la ecuación diferencial

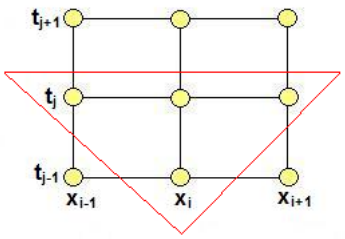
$$\frac{\partial^2 u}{\partial x^2} = k\frac{\partial u}{\partial t} \quad \Rightarrow \quad \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 = k\frac{u_{i,j} - u_{i,j-1}}{\Delta t} + O(\Delta t)$$

Se obtiene la ecuación de diferencias

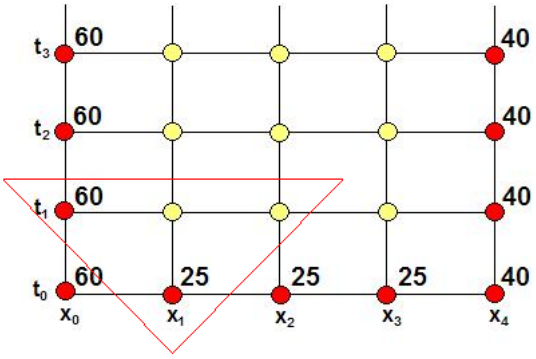
$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} = k\frac{u_{i,j} - u_{i,j-1}}{\Delta t}$$

Su error de truncamiento es igualmente $T = O(\Delta x)^2 + O(\Delta t)$, debiendo cumplirse que $\Delta t < \Delta x$

Marcamos los puntos incluidos en esta ecuación de diferencias.



Los puntos marcados conforman un triángulo de puntos invertido. Este triángulo correspondiente a los puntos incluidos en la ecuación de diferencias y puede colocarse en cualquier lugar de la malla asignando a **i, j** los valores apropiados. Por ejemplo, si **i=1, j=1**, la ecuación de diferencias se aplicada en el extremo inferior izquierdo de la malla:



La ecuación de diferencias ahora contiene dos puntos desconocidos. Si la ecuación se aplica sucesivamente a los puntos **i=2, j=1** e **i=3, j=1**, se obtendrá un sistema de tres ecuaciones lineales y su solución proporcionará el valor de los tres puntos desconocidos. Esta ecuación de diferencias genera un **método implícito** para obtener la solución.

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2} = k\frac{u_{i,j}-u_{i,j-1}}{\Delta t}, \quad T = O(\Delta x)^2 + O(\Delta t), \quad \Delta t < \Delta x$$

$$\frac{\Delta t}{k(\Delta x)^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j}) = u_{i,j}-u_{i,j-1}$$

Definiendo $\lambda = \dfrac{\Delta t}{k(\Delta x)^2}$, la ecuación se puede escribir

$$\lambda u_{i-1,j} + (-1-2\lambda)u_{i,j} + \lambda u_{i+1,j} = -u_{i,j-1}, \quad i = 1, 2, 3; \ j = 1, 2, 3, \ldots$$

Finalmente con los datos $\lambda = \dfrac{\Delta t}{k(\Delta x)^2} = \dfrac{0.1}{4(0.25)^2} = 0.4$, se obtiene la forma final

$$0.4u_{i-1,j} - 1.8u_{i,j} + 0.4u_{i+1,j} = -u_{i,j-1}, \quad i = 1, 2, 3; \ j = 1, 2, 3, \ldots$$

La cual genera un sistema de ecuaciones lineales para obtener la solución en cada nivel t_j

Calcular un nivel de la solución con este método:

j = 1, i = 1: $0.4u_{0,1} - 1.8u_{1,1} + 0.4u_{2,1} = -u_{1,0}$

$0.4(60) - 1.8u_{1,1} + 0.4u_{2,1} = -25 \quad \Rightarrow \quad -1.8\,u_{1,1} + 0.4u_{2,1} = -49$

i = 2: $0.4u_{1,1} - 1.8u_{2,1} + 0.4u_{3,1} = -u_{2,0}$

$0.4u_{1,1} - 1.8u_{2,1} + 0.4u_{3,1} = -25 \quad \Rightarrow \quad 0.4u_{1,1} - 1.8\,u_{2,1} + 0.4u_{3,1} = -25$

i = 3: $0.4u_{2,1} - 1.8u_{3,1} + 0.4u_{4,1} = -u_{3,0}$

$0.4u_{2,1} - 1.8u_{3,1} + 0.4(40) = -25 \quad \Rightarrow \quad 0.4u_{2,1} - 1.8\,u_{3,1} = -41$

Se tiene el sistema lineal

$$\begin{bmatrix} -1.8 & 0.4 & 0 \\ 0.4 & -1.8 & 0.4 \\ 0 & 0.4 & -1.8 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} = \begin{bmatrix} -49 \\ -25 \\ -41 \end{bmatrix}$$

Cuya solución es

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} = \begin{bmatrix} 33.0287 \\ 26.1290 \\ 28.5842 \end{bmatrix}$$

Un análisis de estabilidad demuestra que el método implícito no está condicionado a la magnitud de $\lambda = \frac{\Delta t}{k(\Delta x)^2}$ como en el método explícito.

10.2.5 Instrumentación computacional

La siguiente instrumentación del método de diferencias finitas implícito permite resolver problemas de tipo similar al ejemplo anterior. La ecuación de diferencias debe escribirse en forma estandarizada

$$(P)u_{i-1,j} + (Q)u_{i,j} + (R)u_{i+1,j} = -u_{i,j-1}, \quad i = 1, 2, 3, \ldots, m-1; \; j = 1, 2, 3, \ldots$$

En donde **P, Q, R** dependen de los datos de la ecuación que se desea resolver.

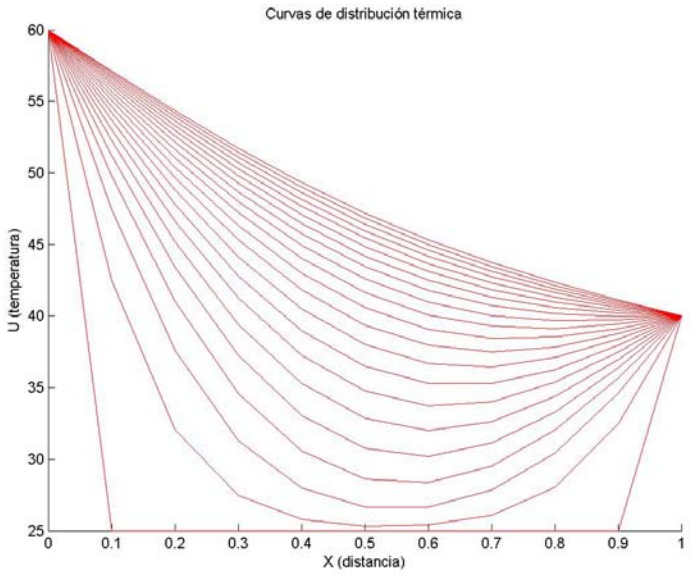
Para el ejemplo propuesto se tiene: $\lambda u_{i-1,j} + (-1-2\lambda)u_{i,j} + \lambda u_{i+1,j} = -u_{i,j-1}$

Se define una función **EDPDIFPI** que genera y resuelve el sistema de ecuaciones lineales en cada nivel. El sistema de ecuaciones lineales resultante tiene forma tridiagonal, y por lo tanto se puede usar un algoritmo específico muy eficiente con el nombre **TRIDIAGONAL** cuya instrumentación fue realizada en capítulos anteriores.

```
%  Método de Diferencias Finitas implícito para una EDP
%  Ecuación de diferencias estandarizada
%  (P)u(i-1,j) + (Q)u(i,j) + (R)u(i+1,j) = -u(i,j-1)
%  P,Q,R son constantes evaluadas con los datos de la EDP
clf;
m=11;                % Número de puntos en x
n=100;               % Número de niveles en t
Ta=60; Tb=40;        % Condiciones en los bordes
To=25;               % Condición en el inicio
dx=0.1; dt=0.01;     % incrementos
L=1;                 % longitud
k=4;             % dato especificado
clear x u U;
U(1)=Ta;             % Asignación inicial
U(m)=Tb;
for i=2:m-1
  U(i)=To;
end
lambda=dt/(k*dx^2);
P=lambda;
Q=-1-2*lambda;
R=lambda;
hold on;
title('Curvas de distribución térmica ');
xlabel('X (distancia)');
ylabel('U (temperatura)');
x=0:dx:L;                % Coordenadas para el grafico
plot(x,U,'r');           % Distribución inicial
for j=1:n
   U =EDPDIF(P,Q,R,U,m);
   if mod(j,5)==0
     plot(x,U,'r');   % Para graficar curvas cada 5 niveles de t
     pause
  end
end
```

```
function U = EDPDIFPI(P, Q, R, U, m)
% Solución de una EDP con condiciones constantes en los bordes
% Método de Diferencias Finitas Implícito
% Generación del sistema tridiagonal
clear a b c d;
for i=1:m-2
  a(i)=P;
  b(i)=Q;
  c(i)=R;
  d(i)=-1*U(i+1);
end
d(1)=d(1)-a(1)*U(1);
d(m-2)=d(m-2)-c(m-2)*U(m);
u=tridiagonal(a,b,c,d);
U=[U(1) u U(m)];   % Incluir datos en los extremos
```

Almacenar la función, el programa y ejecutarlo para obtener el gráfico siguiente



Curvas de distribución térmica para el ejemplo. La distribución tiende a una forma estable.

10.2.6 Práctica computacional

Probar los métodos explícito e implícito instrumentados computacionalmente para resolver el ejemplo anterior cambiando Δx y Δt para analizar la convergencia de los esquemas de diferencias finitas.

Realizar pruebas para verificar la condición de estabilidad condicionada para el método explícito e incondicional para el método implícito.

10.2.7 Condiciones variables en los bordes

Analizamos la ecuación anterior cambiando las condiciones inicial y en los bordes.

Suponer que inicialmente la barra se encuentra a una temperatura que depende de la posición mientras que en el borde derecho se aplica una fuente de calor que depende del tiempo y en el extremo izquierdo hay una pérdida de calor, según las siguientes especificaciones:

$$u(x,t): \quad \frac{\partial^2 u}{\partial x^2} = k\frac{\partial u}{\partial t}, \quad 0 \le x \le 1, \quad t \ge 0$$

$$\frac{\partial u(0,t)}{\partial x} = -5, \qquad t \ge 0$$

$$u(1, t) = 20 + 10\,\text{sen}(t), \quad t \ge 0$$

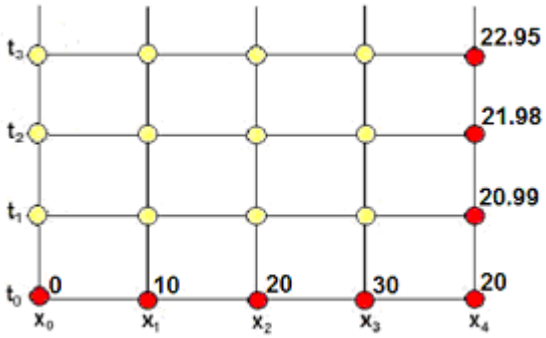
$$u(x, 0) = 40\,x, \qquad 0 < x < 1$$

$$\Delta x = 0.25, \quad \Delta t = 0.1, \quad k = 4$$

Para aplicar el método de diferencias finitas, debe discretizarse el dominio de **u**

$$u(x_i, t_j) = u_{i,j} \; ; \; i = 0, 1, ..., n; \quad j = 0, 1, 2, ...$$

La red con los datos incluidos en los bordes inferior y derecho:



Los puntos en color amarillo, en el borde izquierdo y en el centro, son desconocidos

Usamos el esquema de diferencias finitas implícito visto anteriormente:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} = k\frac{u_{i,j} - u_{i,j-1}}{\Delta t}, \quad E = O(\Delta x)^2 + O(\Delta t), \quad \Delta t < \Delta x$$

$$\frac{\Delta t}{k(\Delta x)^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = u_{i,j} - u_{i,j-1}$$

Definiendo $\lambda = \dfrac{\Delta t}{k(\Delta x)^2}$, la ecuación se puede escribir

$$\lambda u_{i-1,j} + (-1 - 2\lambda)u_{i,j} + \lambda u_{i+1,j} = -u_{i,j-1}, \quad i = 1, 2, 3; \quad j = 1, 2, 3, \ldots$$

Finalmente se tiene la ecuación de diferencias con los datos $\lambda = \dfrac{\Delta t}{k(\Delta x)^2} = \dfrac{0.1}{4(0.25)^2} = 0.4$,

Ahora la ecuación es aplicada en cada punto desconocido, incluyendo el borde izquierdo:

$$0.4u_{i-1,j} - 1.8u_{i,j} + 0.4u_{i+1,j} = -u_{i,j-1}, \quad i = 0, 1, 2, 3; \quad j = 1, 2, 3, \ldots$$

La cual genera un sistema de ecuaciones lineales para obtener la solución en cada nivel t_j

A continuación calculamos un nivel de la solución con este método:

$j = 1$, $i = 0$: $0.4u_{-1,1} - 1.8u_{0,1} + 0.4u_{1,1} = -u_{1,0} \Rightarrow 0.4u_{-1,1} - 1.8u_{0,1} + 0.4u_{1,1} = 0$ (1)

$i = 1$: $0.4u_{0,1} - 1.8u_{1,1} + 0.4u_{2,1} = -u_{1,0} \Rightarrow 0.4u_{0,1} - 1.8u_{1,1} + 0.4u_{2,1} = -10$ (2)

$i = 2$: $0.4u_{1,1} - 1.8u_{2,1} + 0.4u_{3,1} = -u_{2,0} \Rightarrow 0.4u_{1,1} - 1.8u_{2,1} + 0.4u_{3,1} = -20$ (3)

$i = 3$: $0.4u_{2,1} - 1.8u_{3,1} + 0.4u_{4,1} = -u_{3,0}$

$0.4u_{2,1} - 1.8u_{3,1} + 0.4(20.99) = -30 \Rightarrow 0.4u_{2,1} - 1.8\,u_{3,1} = -38.4$ (4)

Se tiene un sistema de cuatro ecuaciones y cinco puntos desconocidos: $u_{-1,1}$, $u_{0,1}$, $u_{1,1}$, $u_{2,1}$, $u_{3,1}$ incluyendo el punto ficticio $u_{-1,1}$

El dato adicional conocido: $\frac{\partial u(0,t)}{\partial x} = -5$ es aproximado ahora mediante una fórmula de diferencias finitas central:

$$\frac{\partial u_{0,j}}{\partial x} = \frac{u_{1,j} - u_{-1,j}}{2(\Delta x)} = -5, \quad j = 1, 2, 3, \ldots, \; E = O(\Delta x)^2$$

De donde se obtiene $u_{-1,j} = u_{1,j} + 5(2\Delta x)$ para sustituir en la ecuación **(1)** anterior:

$$j = 1, \quad i = 0: \quad 0.4u_{-1,1} - 1.8u_{0,1} + 0.4u_{1,1} = 0$$
$$0.4[u_{1,j} + 5(2\Delta x)] - 1.8u_{0,1} + 0.4u_{1,1} = 0 \Rightarrow -1.8u_{0,1} + 0.8u_{1,1} = -1 \qquad (1)$$

En notación matricial:

$$\begin{bmatrix} -1.8 & 0.8 & 0 & 0 \\ 0.4 & -1.8 & 0.4 & 0 \\ 0 & 0.4 & -1.8 & 0.4 \\ 0 & 0 & 0.4 & -1.8 \end{bmatrix} \begin{bmatrix} u_{0,1} \\ u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} = \begin{bmatrix} -1 \\ -10 \\ -20 \\ -38.4 \end{bmatrix}$$

Cuya solución es

$$u_{0,1} = 5.4667, \; u_{1,1} = 11.0500, \; u_{2,1} = 19.2584, \; u_{3,1} = 25.6130$$

El sistema de ecuaciones lineales resultante tiene forma tridiagonal, y por lo tanto se puede usar un algoritmo específico muy eficiente para resolverlo.

10.2.8 Instrumentación computacional

La siguiente instrumentación del método de diferencias finitas implícito permite resolver problemas de tipo similar al ejemplo anterior. La ecuación de diferencias debe escribirse en forma estandarizada. Se suponen conocidos los datos en los bordes: $\delta_0 = \frac{\partial u(0,t)}{\partial x}$, $u_{m,j}$

Para el ejemplo propuesto se tiene la ecuación de diferencias estandarizada, incluyendo la forma especial de la ecuación para el borde izquierdo

$$\lambda u_{i-1,j} + (-1 - 2\lambda)u_{i,j} + \lambda u_{i+1,j} = -u_{i,j-1}$$

$$(P)u_{i-1,j} + (Q)u_{i,j} + (R)u_{i+1,j} = -u_{i,j-1}, \qquad i = 1, 2, 3, \ldots, m-1$$

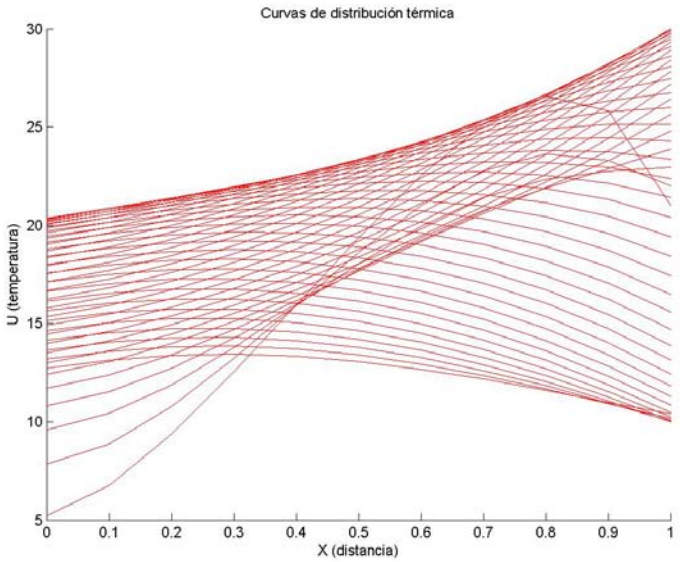
$$\lambda u_{-1,j} + (-1 - 2\lambda)u_{0,j} + \lambda u_{1,j} = -u_{0,j-1}, \qquad i = 0$$
$$\lambda [u_{1,j} + \delta_0 (2\Delta x)] + (-1 - 2\lambda)u_{0,j} + \lambda u_{1,j} = -u_{0,j-1}$$
$$(-1 - 2\lambda)u_{0,j} + 2\lambda u_{1,j} = -u_{0,j-1} - \lambda \delta_0 (2\Delta x)$$
$$(Q)u_{0,j} + 2(P)u_{1,j} = -u_{0,j-1} - (P)\delta_0 (2\Delta x)$$

```
%  Método de Diferencias Finitas implícito para una EDP
%  condiciones en los bordes:
%  A la izquierda una derivada
%  Condición inicial y en el borde derecho variables
%  Ecuación de diferencias estandarizada
%  (P)ui-1,j + (Q)ui,j + (R)ui+1,j = -ui,j-1
%  P,Q,R son constantes evaluadas con los datos de la EDP
clf;
m=11;                % Número de puntos en x
n=50;                % Número de niveles en t
der0=-5;             % Derivada en el borde izquierdo
dx=0.1;
dt=0.1;    % incrementos
L=1;                 % longitud
k=4;           % dato especificado
clear x U;
x=0;
```

```
for i=1:m                  % Condición variable en el inicio
  U(i)=40*x;
  x = x + dx;
end
lambda=dt/(k*dx^2);
P=lambda;
Q=-1-2*lambda;
R=lambda;
hold on;
title('Curvas de distribución térmica');
xlabel('X (distancia)');
ylabel('U (temperatura)');
x=0:dx:L;                  % Coordenadas para el grafico
t=0;
for j=1:n
    t = t + dt;
    U(m) = 20 + 10*sin(t);      % Condición variable en el borde derecho
    U = EDPDIFPID(P, Q, R, U, der0, dx, m);
    plot(x,U,'r');
    pause;
end

function U = EDPDIFPID(P, Q, R, U, der0, dx, m)
% Método de Diferencias Finitas Implicito
% EDP con una derivada a la izquierda y condiciones variables
% Generación del sistema tridiagonal
clear a b c d;
for i=1:m-1
  a(i)=P;
  b(i)=Q;
  c(i)=R;
  d(i)=-1*U(i);
end
c(1)=2*P;
d(1)=d(1)-P*der0*2*dx;
d(m-1)=d(m-1)-c(m-1)*U(m);
u=tridiagonal(a,b,c,d);
U=[ u U(m)];   % Incluir dato en el extremos
```

Almacenar la función, el programa y ejecutarlo para obtener el gráfico siguiente



Curvas de distribución térmica para el ejemplo.

10.2.9 Método de diferencias finitas para EDP no lineales

Si la ecuación tiene términos no lineales, se puede adaptar un método de diferencias finitas explícito como una primera aproximación.

Ejemplo. Formule un esquema de diferencias finitas para resolver la siguiente ecuación diferencial parcial no lineal del campo de la acústica

$$\mathbf{u(x,t):}\quad \frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - k\frac{\partial^2 u}{\partial x^2}$$

Se usarán las siguientes aproximaciones:

$$\frac{\partial u_{i,j}}{\partial t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} + O(\Delta t)$$

$$\frac{\partial u_{i,j}}{\partial x} = \frac{u_{i+1,j} - u_{i-1,j}}{2(\Delta x)} + O(\Delta x)^2$$

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2$$

Sustituyendo en la EDP

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = -u_{i,j}\frac{u_{i+1,j} - u_{i-1,j}}{2(\Delta x)} - k\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2}$$

Se obtiene un esquema explícito de diferencias finitas que permitirá calcular cada punto en la malla que represente al dominio de la ecuación diferencial siempre que estén previamente definidas las condiciones en los bordes así como los parámetros $\mathbf{k, \Delta t, \Delta x}$. También debería analizarse la estabilidad del método.

$$u_{i,j+1} = \Delta t\left(-u_{i,j}\frac{u_{i+1,j} - u_{i-1,j}}{2(\Delta x)} - c\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2}\right) + u_{i,j}$$

Con error de truncamiento

$$T = O(\Delta t) + O(\Delta x)^2$$

10.3 Ecuaciones diferenciales parciales de tipo elíptico

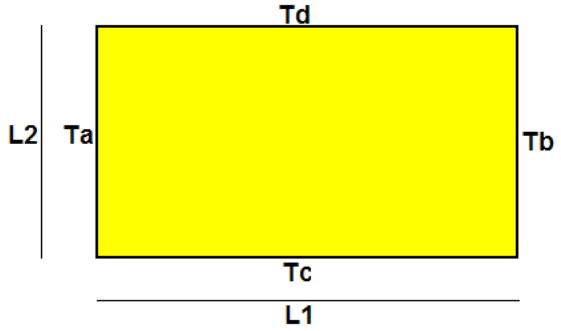
Este tipo de ecuaciones se caracteriza porque su dominio es una región cerrada. Para aplicar el método de diferencias finitas usaremos un ejemplo particular para posteriormente interpretar los resultados obtenidos.

Ejemplo. *Resolver la ecuación de difusión en dos dimensiones:*

$$u(x,y): \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

*Suponer que **u** es una función que depende de **x, y,** en donde **u** representa valores de temperatura, **x, y** representan posición.*

Esta ecuación se puede asociar al flujo de calor en una placa muy delgada aislada térmicamente en sus caras superior e inferior y sometida en los bordes a alguna condición. La solución representa la distribución final de temperaturas en la placa en cada punto **(x, y)**



Ta, Tb, Tc, Td son valores de temperatura, suponer constantes, de alguna fuente de calor aplicada en cada borde de la placa. **L1, L2** son las dimensiones de la placa.

Estas condiciones se pueden expresar simbólicamente en un sistema de coordenadas **X-Y:**

u(0, y) = Ta, 0 < y < L2
u(L1, y) = Tb, 0 < y < L2
u(x, 0) = Tc, 0 < x < L1
u(x, L2) = Td, 0 < x < L1

Para aplicar el método de diferencias finitas, debe discretizarse el dominio de **u** mediante una malla con puntos **$u(x_i, y_j)$,** en la cual **x_i** , **y_j** representan coordenadas

$$u=u(x,y),\ 0 \le x \le L1,\quad 0 \le y \le L2 \Rightarrow \quad u(x_i, y_j) = u_{i,j}\ ;\ i = 0, 1, ..., n;\ \ j = 0, 1, 2, ..., m$$

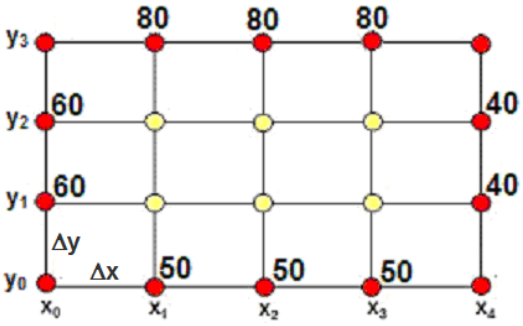
El método de diferencias finitas permitirá encontrar **u** en estos puntos.

Para el ejemplo supondremos los siguientes datos, en las unidades que correspondan

Ta = 60
Tb = 40
Tc = 50
Td = 80
L1 = 2
L2 = 1.5

Supondremos además que **$\Delta x = 0.5,\ \Delta y = 0.5$**

Con esta información describimos el dominio de **u** mediante una malla con puntos en dos dimensiones en la cual el eje horizontal representa la posición **x_i** mientras que el eje vertical representa **y_j**. En esta malla se representan los datos en los bordes y los puntos interiores que deben ser calculados



10.3.1 Un esquema de diferencias finitas implícito

Elegimos las siguientes aproximaciones de diferencias finitas para sustituir las derivadas de la ecuación diferencial

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2$$

$$\frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} + O(\Delta y)^2$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2 + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} + O(\Delta y)^2 = 0$$

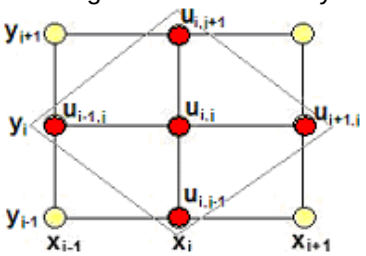
Se obtiene la ecuación de diferencias

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0$$

Cuyo error de truncamiento es $\mathbf{E = O(\Delta x)^2 + O(\Delta y)^2}$

Para que esta sustitución sea consistente, **Δx** debe ser muy cercano a **Δy** en magnitud.

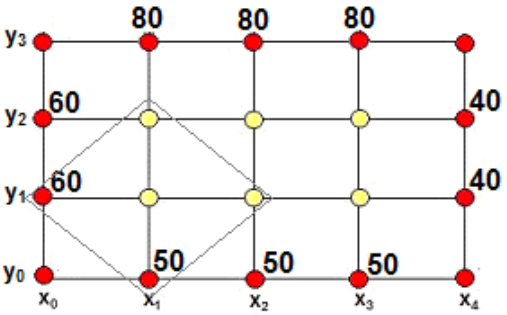
Es conveniente analizar los puntos incluidos en la ecuación de diferencias. Para esto consideramos un segmento de la malla y marcamos los puntos de la ecuación.



Los puntos marcados conforman un rombo. Este rombo describe los puntos incluidos en la ecuación de diferencias y puede colocarse en cualquier lugar de la malla asignando a **i, j** los valores apropiados.

Por ejemplo, si **i=1, j=1**, la ecuación de diferencias se aplica al extremo inferior izquierdo de la malla.

Se puede observar que la ecuación de diferencias contiene tres puntos desconocidos



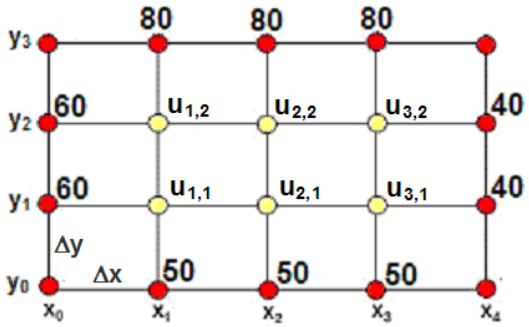
Si se aplica a todos los puntos interiores: **i = 1, 2, 3** con **j = 1, 2** se obtendrá un sistema de seis ecuaciones lineales con los seis puntos desconocidos cuyos valores se pueden determinar resolviendo el sistema. Por lo tanto, la ecuación de diferencias proporciona un **método implícito** para obtener la solución. Una simplificación adicional se obtiene haciendo **Δx = Δy**

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0$$

Forma final de la ecuación de diferencias:

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0\,, \quad i = 1, 2, 3;\ \ j = 1, 2$$

Esta ecuación se aplica para en la malla para generar el sistema de ecuaciones lineales



$$\begin{aligned}
&j = 1, i = 1: \ 60 + 50 + u_{2,1} + u_{1,2} - 4u_{1,1} = 0 \quad \Rightarrow \quad -4u_{1,1} + u_{2,1} + u_{1,2} = -110\\
&\qquad i = 2: \ 50 + u_{1,1} + u_{3,1} + u_{2,2} - 4u_{2,1} = 0 \quad \Rightarrow \quad u_{1,1} - 4u_{2,1} + u_{3,1} + u_{2,2} = -50\\
&\qquad i = 3: \ 50 + 40 + u_{3,2} + u_{2,1} - 4u_{3,1} = 0 \quad \Rightarrow \quad u_{2,1} - 4u_{3,1} + u_{3,2} = -90
\end{aligned}$$

$$\begin{aligned}
&j = 2, i = 1: \ 60 + 80 + u_{1,1} + u_{2,2} - 4u_{1,2} = 0 \quad \Rightarrow \quad u_{1,1} - 4u_{1,2} + u_{2,2} = -140\\
&\qquad i = 2: \ 80 + u_{1,2} + u_{2,1} + u_{3,2} - 4u_{2,2} = 0 \quad \Rightarrow \quad u_{2,1} + u_{1,2} - 4u_{2,2} + u_{3,2} = -80\\
&\qquad i = 3: \ 80 + 40 + u_{2,2} + u_{3,1} - 4u_{3,2} = 0 \quad \Rightarrow \quad u_{3,1} + u_{2,2} - 4u_{3,2} = -120
\end{aligned}$$

Se tiene el sistema lineal

$$\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 \\ 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{3,2} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{bmatrix} = \begin{bmatrix} -110 \\ -50 \\ -90 \\ -140 \\ -80 \\ -120 \end{bmatrix}$$

Cuya solución es

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{bmatrix} = \begin{bmatrix} 57.9917 \\ 56.1491 \\ 51.3251 \\ 65.8178 \\ 65.2795 \\ 59.1511 \end{bmatrix}$$

Se ha usado un método directo para resolver el sistema cuya forma es diagonal dominante, por lo que también se podrían usar métodos iterativos cuya convergencia es segura.

10.3.2 Instrumentación computacional

La siguiente instrumentación en MATLAB está diseñada para resolver una ecuación diferencial parcial elíptica con condiciones constantes en los bordes. Se supondrá además que $\Delta x = \Delta y$

Al aplicar la ecuación de diferencias en los puntos de la malla, cada ecuación tiene no más de cuatro componentes y tiene una forma adecuada para instrumentar un método iterativo para obtener la solución.

Ecuación de diferencias

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} = 0 \;,\quad i = 1, 2, 3, \ldots \quad ;\; j = 1, 2, 3, \ldots$$

Fórmula iterativa

$$u_{i,j}^{(k+1)} = \frac{1}{4}\left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right), \quad k=0, 1, 2, \ldots$$

En la siguiente instrumentación se ha usado el método de Gauss-Seidel para calcular la solución partiendo de una matriz con los valores iniciales iguales al promedio de los valores de los bordes.

```
% Programa para resolver una EDP Elíptica
% con condiciones constantes en los bordes
Ta=60;Tb=60;Tc=50;Td=70;          % Bordes izquierdo, derecho, abajo, arriba
n=10;                             % Puntos interiores en dirección horizontal (X)
m=10;                             % Puntos interiores en dirección vertical (Y)
miter=100;                        % Máximo de iteraciones
e=0.001;                          % Error de truncamiento relativo requerido  0.1%
clear u;
for i=1:n+2
   u(i,1)=Tc;
   u(i,m+2)=Td;
end
for j=1:m+2
   u(1,j)=Ta;
   u(n+2,j)=Tb;
end
p=0.25*(Ta+Tb+Tc+Td);             %valor inicial interior es el promedio de los bordes
for i=2:n-1
   for j=2:m-1
      u(i,j)=p;
   end
end
k=0;                              %conteo de iteraciones
conv=0;                           %señal de convergencia
while k<miter & conv==0
   k=k+1;
   t=u;
   for i=2:n+1
      for j=2:m+1
         u(i,j)=0.25*(u(i-1,j)+u(i+1,j)+u(i,j+1)+u(i,j-1));
      end
   end
   if norm((u-t),inf)/norm(u,inf)<e
      conv=1;
   end
end
if conv==1
   disp(u);                       % Muestra la solución final en la malla
   disp(k);                       % Cantidad de iteraciones realizadas
   [x,y]=meshgrid(1:m+2, 1:n+2);  % Malla para el grafico en tres dimensiones
   surf(x,y,u)                    % Grafico en tres dimensiones
   colormap copper                % Color
```

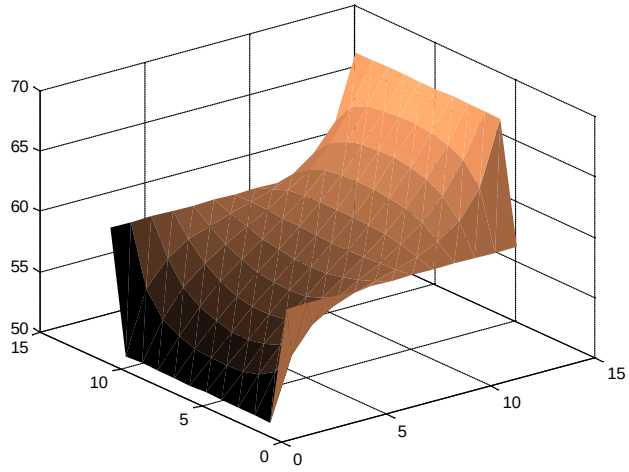
```
    shading flat                    % Suavizado
else
    disp('No converge');
end
```

Solución numérica calculada en los puntos de la malla

60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000
50.0000	55.0555	57.1094	58.1502	58.8179	59.3552	59.8800	60.4789	61.2649	62.4714	64.7253	70.0000
50.0000	53.1504	55.2931	56.7522	57.8553	58.8149	59.7729	60.8462	62.1676	63.9325	66.4446	70.0000
50.0000	52.3140	54.2564	55.8324	57.1733	58.4166	59.6826	61.0788	62.7133	64.7007	67.1404	70.0000
50.0000	51.9277	53.7079	55.3016	56.7603	58.1708	59.6256	61.2131	63.0120	65.0824	67.4393	70.0000
50.0000	51.7775	53.4827	55.0770	56.5863	58.0739	59.6166	61.2895	63.1550	65.2488	67.5585	70.0000
50.0000	51.7911	53.5081	55.1116	56.6271	58.1173	59.6589	61.3274	63.1856	65.2700	67.5691	70.0000
50.0000	51.9651	53.7780	55.3973	56.8731	58.2907	59.7427	61.3179	63.0965	65.1409	67.4687	70.0000
50.0000	52.3661	54.3539	55.9657	57.3301	58.5835	59.8454	61.2245	62.8308	64.7821	67.1813	70.0000
50.0000	53.2029	55.3914	56.8866	58.0135	58.9831	59.9370	60.9930	62.2860	64.0145	66.4858	70.0000
50.0000	55.0911	57.1760	58.2412	58.9249	59.4689	59.9908	60.5780	61.3448	62.5267	64.7531	70.0000
60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000	60.0000

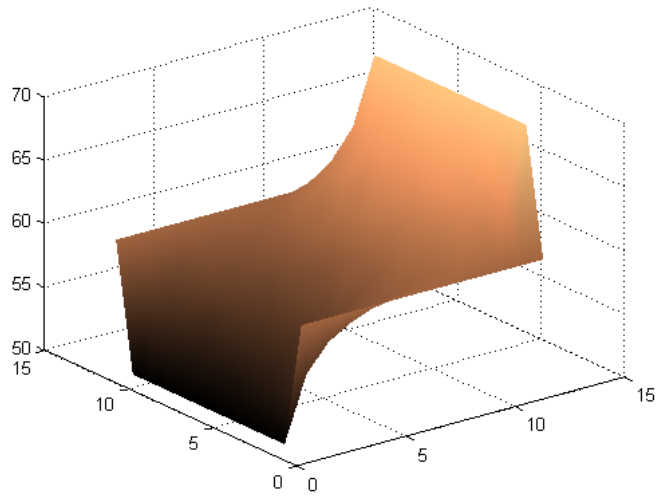
Error relativo: **0.1%**
Cantidad de iteraciones: **41**

Representación gráfica en tres dimensiones de la solución calculada



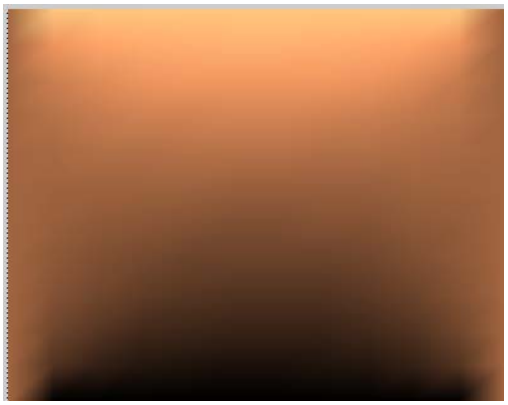
Se puede suavizar el gráfico escribiendo el comando

>> shading Interp



Los valores mas claros representan mayor temperatura

Con el editor de gráficos se puede además rotarlo y observarlo desde diferentes perspectivas



Vista superior. Los valores mas claros representan mayor temperatura

10.4 Ecuaciones diferenciales parciales de tipo hiperbólico

En este tipo de ecuaciones el dominio está abierto en uno de los bordes Para aplicar el método de diferencias finitas usaremos un ejemplo particular para posteriormente interpretar los resultados obtenidos.

Ejemplo. *Ecuación de la onda en una dimensión:* ***u(x, t):*** $\dfrac{\partial^2 u}{\partial t^2} = c^2 \dfrac{\partial^2 u}{\partial x^2}$

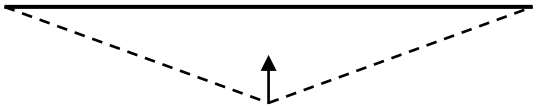
Suponer que ***u*** *es una función que depende de* ***x, t*** *en donde* ***u*** *representa el desplazamiento vertical de la cuerda en función de la posición horizontal* ***x,*** *y el tiempo* ***t,*** *mientras que* ***c*** *es una constante.* ***L*** *es la longitud de la cuerda*

Suponer que los extremos de la cuerda están sujetos y que la longitud es **L = 1**

$$u = u(x, t),\ 0 \le x \le 1, \quad t \ge 0$$
$$u(0, t) = 0, \quad t \ge 0$$
$$u(1, t) = 0, \quad t \ge 0$$

Inicialmente la cuerda es estirada desplazando su punto central una distancia de **0.25**

$$u(x, 0) = \begin{cases} -0.5x, & 0 < x \le 0.5 \\ 0.5(x-1), & 0.5 < x < 1 \end{cases}$$



Al soltar la cuerda desde la posición indicada, sin velocidad inicial:

$$\frac{\partial u(x,0)}{\partial t} = 0$$

10.4.1 Un esquema de diferencias finitas explícito

Para aplicar el método de diferencias finitas, debe discretizarse el dominio de **u** mediante una malla con puntos en dos dimensiones en la cual el eje horizontal representa la posición x_i mientras que el eje vertical representa el tiempo t_j.

$$u=u(x,t),\ 0 \le x \le L,\ t \ge 0 \quad \Rightarrow \quad u(x_i, t_j) = u_{i,j}\ ;\ i = 0, 1, ..., n;\ j = 0, 1, 2, ...$$

Elegimos las siguientes aproximaciones de diferencias finitas para las derivadas:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta x)^2$$

$$\frac{\partial^2 u_{i,j}}{\partial t^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta t)^2} + O(\Delta t)^2$$

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta t)^2} + O(\Delta x)^2 = c^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + O(\Delta t)^2$$

Se obtiene la ecuación de diferencias:

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta t)^2} = c^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}$$

Cuyo error de truncamiento es $E = O(\Delta x)^2 + O(\Delta t)^2$

Esta ecuación se puede expresar de la siguiente forma:

$$u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = \frac{c^2 (\Delta t)^2}{(\Delta x)^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$$

Mediante un análisis se demuestra que si $\frac{c^2(\Delta t)^2}{(\Delta x)^2} \leq 1$, entonces la solución calculada con el método de diferencias finitas es estable.

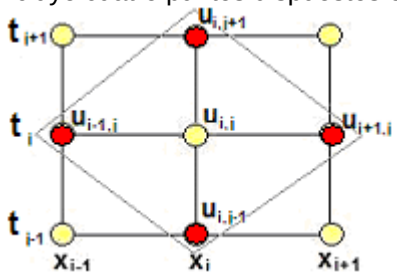
Para el ejemplo, suponer que **c = 2, Δx = 0.2,** entonces de la condición anterior se tiene:

$$\frac{2^2(\Delta t)^2}{(0.2)^2} = 1 \Rightarrow \Delta t = 0.1$$

Esta sustitución permite además simplificar:

$$u_{i,j+1} + u_{i,j-1} = u_{i+1,j} + u_{i-1,j}$$

La ecuación incluye cuatro puntos dispuestos en un rombo.

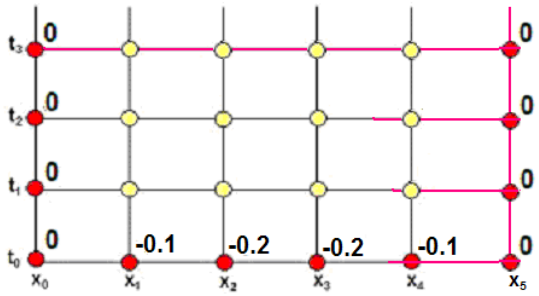


La solución progresa en la dirección **t** por lo que despejamos el punto en el vértice superior.

$$u_{i,j+1} = u_{i+1,j} + u_{i-1,j} - u_{i,j-1}, \quad i = 1, 2, 3, 4; \quad j = 1, 2, 3, \ldots \quad (1)$$

Se puede notar observando el gráfico que para obtener cada nuevo punto de la solución se requieren conocer dos niveles previos de la solución

Describimos el dominio de **u** mediante una malla con puntos en dos dimensiones en la cual el eje horizontal representa la posición x_i mientras que el eje vertical representa tiempo t_j. En esta malla se representan los datos en los bordes y los puntos interiores que deben ser calculados



Siguiendo una estrategia usada anteriormente, expresamos el dato adicional $\frac{\partial u(x,0)}{\partial t} = 0$ mediante una fórmula de diferencias central:

$$\frac{\partial u_{i,0}}{\partial t} = \frac{u_{i,1} - u_{i,-1}}{2(\Delta t)} = 0 \Rightarrow u_{i,-1} = u_{i,1} \quad (2)$$

Esta ecuación incluye el punto ficticio $u_{i,-1}$ Conviene entonces evaluar la ecuación **(1)** cuando **t = 0:**

$$j = 0: \quad u_{i,1} = u_{i+1,0} + u_{i-1,0} - u_{i,-1}, \quad i = 1, 2, 3, 4$$

Sustituyendo en esta ecuación el resultado **(2)** se elimina el punto ficticio $u_{i,-1}$

$$u_{i,1} = u_{i+1,0} + u_{i-1,0} - u_{i,1}$$

$$u_{i,1} = \frac{1}{2}(u_{i+1,0} + u_{i-1,0}) \text{ , } i = 1, 2, 3, 4 \qquad (3)$$

La ecuación **(3)** debe aplicarse únicamente cuando t **= 0,** para encontrar $u_{i,j}$ en el nivel j=1

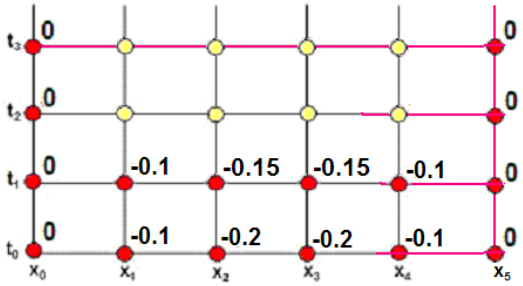
j = 0, **i = 1:** $u_{1,1} = \frac{1}{2}(u_{2,0} + u_{0,0}) = \frac{1}{2}(-0.2 + 0) = -0.1$

i = 2: $u_{2,1} = \frac{1}{2}(u_{3,0} + u_{1,0}) = \frac{1}{2}(-0.2 + (-0.1)) = -0.15$

i = 3: $u_{3,1} = \frac{1}{2}(u_{4,0} + u_{2,0}) = \frac{1}{2}(-0.1 + (-0.2)) = -0.15$

i = 4: $u_{4,1} = \frac{1}{2}(u_{5,0} + u_{3,0}) = \frac{1}{2}(0 + (-0.2)) = -0.1$

Los valores calculados son colocados en la malla:



Ahora se tienen dos niveles con puntos conocidos. A partir de aquí se debe usar únicamente la ecuación **(1)** como un esquema explícito para calcular directamente cada punto en los siguientes niveles **j,** cada uno a una distancia **Δt = 0.1**

$$u_{i,j+1} = u_{i+1,j} + u_{i-1,j} - u_{i,j-1}, \quad i = 1, 2, 3, 4; \qquad j = 1, 2, 3, \ldots$$

j = 1, **i = 1:** $u_{1,2} = u_{2,1} + u_{0,1} - u_{1,0} = -0.15 + 0 - (-0.1) = -0.05$

i = 2: $u_{2,2} = u_{3,1} + u_{1,1} - u_{2,0} = -0.15 + (-0.1) - (-0.2) = -0.05$

i = 3: $u_{3,2} = u_{4,1} + u_{2,1} - u_{3,0} = -0.1 + (-0.15) - (-0.2) = -0.05$

i = 4: $u_{4,2} = u_{5,1} + u_{3,1} - u_{4,0} = 0 + (-0.15) - (-0.1) = -0.05$

j = 2, **i = 1:** $u_{1,3} = u_{2,2} + u_{0,2} - u_{1,1} = -0.05 + 0 - (-0.1) = 0.05$

i = 2: $u_{2,3} = u_{3,2} + u_{1,2} - u_{2,1} = -0.05 + (-0.05) - (-0.15) = 0.05$

i = 3: $u_{3,3} = u_{4,2} + u_{2,2} - u_{3,1} = -0.05 + (-0.05) - (-0.15) = 0.05$

i = 4: $u_{4,3} = u_{5,2} + u_{3,2} - u_{4,1} = 0 + (-0.05) - (-0.1) = 0.05$

j = 3, **i = 1:** $u_{1,4} = u_{2,3} + u_{0,3} - u_{1,2} = 0.05 + 0 - (-0.05) = 0.1$

i = 2: $u_{2,4} = u_{3,3} + u_{1,3} - u_{2,2} = 0.05 + 0.05 - (-0.05) = 0.15$

i = 3: $u_{3,4} = u_{4,3} + u_{2,3} - u_{3,2} = 0.05 + 0.05 - (-0.05) = 0.15$

i = 4: $u_{4,4} = u_{5,3} + u_{3,3} - u_{4,2} = 0 + 0.05 - (-0.05) = 0.1$

etc.

10.4.2 Instrumentación computacional

La siguiente instrumentación del método de diferencias finitas permite resolver problemas de tipo similar al ejemplo anterior: extremos fijos, estiramiento central, velocidad inicial nula y parámetro $\frac{c^2(\Delta t)^2}{(\Delta x)^2} = 1$ cuya ecuación de diferencias es:

$$u_{i,1} = \frac{1}{2}(u_{i+1,0} + u_{i-1,0}), \qquad i = 1, 2, 3, \ldots, m\text{-}1 \quad \text{(Para el primer nivel } j = 1\text{)}$$

$$u_{i,j+1} = u_{i+1,j} + u_{i-1,j} - u_{i,j-1} \quad i = 1, 2, 3, \ldots, m\text{-}1 \quad \text{(Para los siguientes niveles } j = 2, 3, 4, \ldots\text{)}$$

El esquema de cálculo utilizado es explícito. Cada punto es calculado directamente mediante funciones que entregan la solución calculada y un programa que contiene los datos y muestra gráficamente la solución. Se puede visualizar el movimiento de la cuerda incluyendo en el programa los comandos **pause** y **clf**. En la ejecución presione alguna tecla para visualizar el movimiento.

```
%  Método de Diferencias Finitas explícito para una EDP Hiperbólica
%         con parametro c^2 dt^2/dx^2 = 1
%  Extremos fijos, estiramiento central y velocidad inicial cero
clf;
m=11;                          % Número de puntos en x
n=50;                          % Número de niveles en t
c=2;                           % dato especificado
dx=0.1;
dt=sqrt(dx^2/2^2);             % incrementos
L=1;                           % longitud
clear x U0 U1 Uj;
U0(1)=0;                       % Extremos fijos
U0(m)=0;
x=0;
for i=2:m-1                    % Posición inicial de la cuerda
   x=x+dx;
   if x<L/2;
      U0(i)=-0.5*x;
   else
      U0(i)=0.5*(x-1);
   end
end
x=0:dx:L;                      % Coordenadas para el grafico
plot(x,U0,'r');                % Distribución inicial
axis([0,1,-0.5,0.5]);
pause;
clf;
U1=EDPDIFH1(U0,m);             % Calculo del primer nivel
plot(x,U1,'r');
axis([0,1,-0.5,0.5]);
pause;
clf;
for j=1:n
    Uj=EDPDIFHJ(U0,U1,m);      % Siguientes niveles
    plot(x,Uj,'r');
    axis([0,1,-0.5,0.5]);
    pause;
    clf;
    U0=U1;
    U1=Uj;
 end
```

```
function U1=EDPDIFH1(U0,m)
% Solución U(x,t) de una EDP hiperbolica con parametro igual a 1
% Cálculo del primer nivel de la solucion
U1(1)=U0(1);
for i=2:m-1
   U1(i)=0.5*(U0(i-1)+U0(i+1));
end
U1(m)=U0(m);
```

```
function Uj=EDPDIFHJ(U0,U1,m)
% Solución U(x,t) de una EDP hiperbolica con parametro igual a 1
% Cálculo de los siguientes niveles de la solucion
Uj(1)=U1(1);
for i=2:m-1
   Uj(i)=U1(i+1)+U1(i-1)-U0(i);
end
Uj(m)=U1(m);
```

Almacenar las funciones y el programa y ejecutarlo para visualizar la solución:

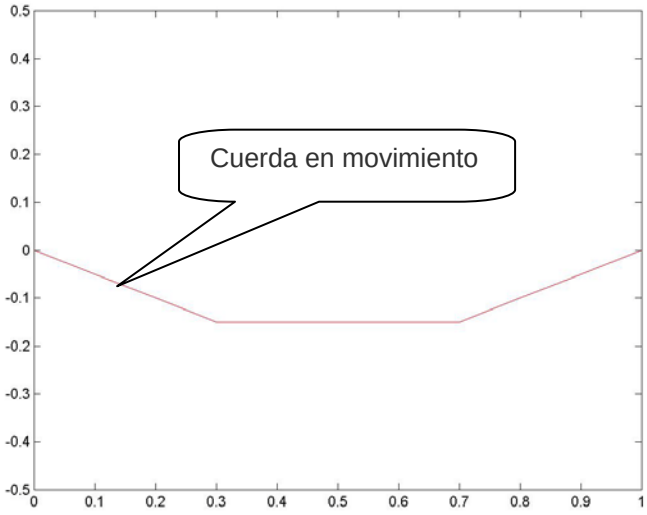


Gráfico de la cuerda en un instante de su desplazamiento

10.5 Ejercicios con ecuaciones diferenciales parciales

1. Dada la siguiente ecuación diferencial parcial de tipo parabólico

$$\frac{\partial^2 u}{\partial x^2} = C\frac{\partial u}{\partial t}, u = u(x,t), 0 < x < 2, t > 0,$$ con las condiciones

a) u(x,0)=25 sen(x), 0<x<2

b) u(0,t)=10 t, t≥0

c) $$\frac{\partial u(2,t)}{\partial x} = 5, t \geq 0$$

a) Calcule manualmente dos niveles de la solución con el método explícito de diferencias finitas

Utilice **Δx=0.5, C=1.6**

Para que el método sea estable use la condición $$\frac{\Delta t}{C(\Delta x)^2} = \frac{1}{2}$$

Use una aproximación de diferencias finitas de segundo orden para la condición c)

b) Calcule dos niveles de la solución con el método implícito de diferencias finitas. Use las mismas especificaciones dadas en **1.**

2. Con el criterio de Von Newman, analice la estabilidad del método implícito de diferencias finitas para resolver la EDP de tipo parabólico. Demuestre que es incondicionalmente estable.

$$\frac{\partial^2 u}{\partial x^2} = C\frac{\partial u}{\partial t}$$

$$cu_{i-1,j} + (-1-2c)u_{i,j} + cu_{i+1,j} = -u_{i,j-1}, \quad c = \frac{\Delta t}{C(\Delta x)^2}$$

3. Resuelva la siguiente ecuación diferencial parcial de tipo elíptico con el método de diferencias finitas.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

u(0, y) = 10, **0≤y≤3**
u(2, y) = 20 sen (πy), **0≤y≤3**
u_y (x, 0) = 20, **0≤x≤2**
u(x, 3) = 25x, **0≤x≤2**

Determine **u** en los puntos interiores. Use **Δx = Δy = 0.5**

4. La siguiente ecuación diferencial parcial de tipo hiperbólico describe la posición **u** de cierta cuerda en cada punto **x**, en cada instante **t**

$$\frac{\partial^2 u}{\partial t^2} = (1+2x)\frac{\partial^2 u}{\partial x^2}, \quad 0<x<1, t>0$$

Use las siguientes condiciones iniciales y de borde:

u(x, 0) = 0; 0≤x≤1

$$\frac{\partial u(x,0)}{\partial t} = x(1-x)$$

Use **Δx = Δt = 0.25**, y encuentre la solución cuando **t = 1**

BIBLIOGRAFÍA

1. Burden R., Faires J. *Análisis Numérico*, Thomson Learning, 2002, Mexico

2. Gerald, C., *Aplied Numerical Análysis*, Addison-Wesley Publishing Company

3. Conte S., Boor C., *Análisis Numérico Elemental*, McGraw-Hill

4. Carnahan B., Luther H. Wilkes J., *Applied Numerical Methods*, John Wiley & Sons, Inc

5. Nakamura S., *Análisis Numérico y Visualización Gráfica con MATLAB*, Prentice-Hall

6. The Mathworks, Inc. *Using MATLAB Computation, Visualization, Programming*

7. Pérez López C. *MATLAB y sus Aplicaciones a las Ciencias y la Ingeniería*, Pearson Ed.