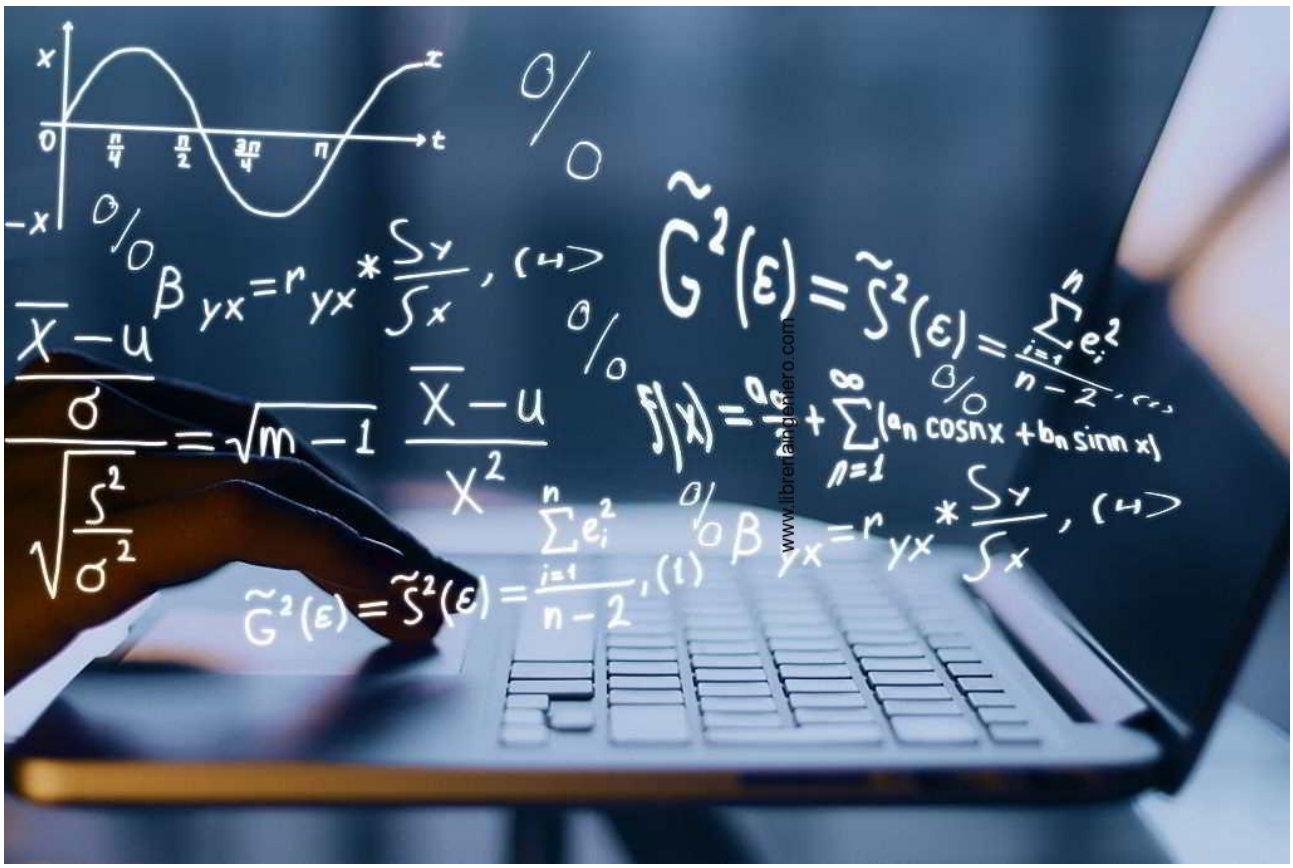
Apuntes Análisis Numérico I (75.12)



IMPORTANTE:

Este apunte fue confeccionado como medio de estudio de la asignatura durante el tiempo que tuve que cursarla. Se tomaron como fuentes las clases de la asignatura en cuestión (presenciales y/o virtuales) y la bibliografía detallada en el campus de la materia. Además, en algunas partes donde lo consideré necesario se complementó con información adicional obtenida de Internet, libros o alguna otra fuente.

De ninguna manera este apunte pretende reemplazar la bibliografía de la materia, sino que el objetivo es poder (en lo posible) tener un documento con la mayor cantidad de información posible para poder estudiar la materia desde un solo lugar, pero puede fallar. Puede ser que algunos temas se encuentren incompletos, simplemente no me dio el tiempo de la cursada para poder completar todo. Y también puede ser que haya pequeños errores, pero se resuelven conversando los temas con los compañeros.

Espero que te sirva el apunte para estudiar y APROBAR la materia (vamos que se puede). Si al finalizar el cuatrimestre pudiste aprobar la asignatura, primero que nada, **¡te felicito!.** Y, en segundo lugar, este apunte se hizo con mucho esfuerzo, falta de horas de sueño y emociones encontradas así que **se agradece si querés colaborar para que estos apuntes sigan existiendo y mejorando ☺, para eso te dejo mi alias de MP**: ***juampi.quijano.mp***.

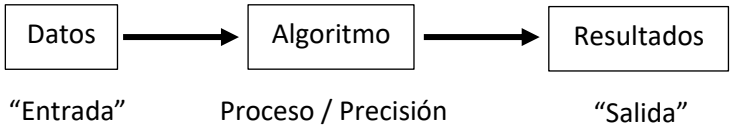
Además, **si encontras algún error o aporte que consideres importante podés enviarlo a: jquijano@fi.uba.ar**

¡ÉXITOS EN LA CURSADA!

Índice

Errores (Guía 1)

Síntesis del modelado numérico



En el mundo ideal, léase el de los cálculos matemáticos, los datos son exactos al igual que los resultados obtenidos, y tanto el proceso como la precisión se consideran infinitos.

Datos exactos ∞ / ∞ Resultado exacto

Ahora bien, cuando caemos al mundo real esto no es posible ya que los datos poseen un error de entrada, el cual se propagará a lo largo de los cálculos, debido a que tanto el proceso como la precisión de estos es finita, y dará como resultado un valor aproximado junto con su cota de error.

Datos con error finito / finita Resultado con error

Representación de los valores

En la realidad muchos de los cálculos que se llevan a cabo en la ingeniería son aproximados. Al aproximar resultados el costo a pagar es la generación de errores respecto al resultado exacto. Un valor real se representa como:

$$a = \hat{a} + \partial a$$

Donde a es el **valor exacto**, $\hat{a}$ representa el **valor estimado o aproximado** y ∂a es el **error absoluto exacto** (tiene signo). De manera similar se define el **error relativo exacto** (r_a) como:

$$r_a = \frac{\partial a}{\hat{a}}$$

No obstante, estas fórmulas descriptas carecen de utilidad en el mundo real pero, por otro lado, sus cotas son esenciales para dar una idea acerca de la efectividad del cálculo numérico realizado. Por ello, se definen:

- **Cota del error absoluto**: $|\boldsymbol{\partial a}| \leq \Delta \boldsymbol{a}$
- **Cota del error relativo**: $|r_a| = \left|\frac{\partial a}{\hat{a}}\right| \leq \frac{\Delta a}{a} \approx \frac{\Delta a}{\hat{a}} \equiv R_a \rightarrow \boldsymbol{R_a} = \frac{\boldsymbol{\Delta a}}{\hat{\boldsymbol{a}}}$
- **Valor aproximado**: $\boldsymbol{a} = \hat{\boldsymbol{a}} \pm \Delta \boldsymbol{a}$

La **cota del error absoluto** indica el rango hacia arriba y hacia abajo respecto al valor estimado donde se puede encontrar el valor exacto. Mientras que, la **cota del error relativo** expresa la relación entre la cota del error absoluto y el valor aproximado, en otras palabras, es una forma de evaluar la precisión del resultado obtenido. Entre más pequeño sea, más precisa será la respuesta aportada. No obstante, tampoco es recomendable intentar alcanzar cotas de errores relativos muy pequeños si no son necesarios.

Fuentes de error

Existen 3 fuentes de error principales:

- **Errores de Entrada:** son de origen diverso y surgen, por ejemplo, al hacer mediciones, cálculos previos, etc.
- **Errores de Truncamiento:** se deben a que el proceso infinito del mundo ideal se hace finito para el mundo real. Por ejemplo, al obtener el resultado de una serie infinita realizando solo una N cantidad de pasos. Se estima el error de truncamiento ΔY como la diferencia entre el cálculo a orden N y N+1, es decir:

$$\Delta Y = |Y_{N+1} - Y_N|$$

- **Errores de Redondeo:** surgen al momento de recurrir a una máquina para realizar cálculos, pues corresponden al error propio de la computadora al llevar a cabo las operaciones y utilizar cierta cantidad de dígitos finitos (precisión finita) para informar o almacenar el resultado.
  Se toma como cota para todos los errores de redondeo u_i, cometidos en cada una de las operaciones, la **unidad de máquina**; denotada con la letra $\boldsymbol{u}$. Entonces:

$$|u_i| \leq u$$

- Otras fuentes de error:
  - **Errores en el modelo:** se debe a la elección de un modelo erróneo para describir el comportamiento de un cierto fenómeno o sistema en estudio.
  - **Errores humanos**

Escritura de resultados y dígitos significativos

Los dígitos se pueden clasificar en tres tipos:

- **Significativos:** aquellos que se presentan en el resultado.
- **Medianamente significativos:** aquel dígito significativo cuyo valor puede variar dependiendo del valor de la cota del error absoluto.
- **No significativos:** todos aquellos que no son significativos, por ende, no tienen significancia para la información del resultado.

Ahora bien, existen dos tipos de redondeo:

- **Redondeo simétrico:** el último dígito significativo o medianamente significativo no se modifica si el primer dígito no significativo es menor a 5. Caso contrario, se lo incrementa en 1.
- **Redondeo truncado:** se escribe hasta el último dígito significativo o medianamente significativo.

Por convención las cotas de los errores se escriben con 1 dígito significativo; o a lo sumo dos dígitos significativos para el caso de la cota del error relativo. Además, estos dígitos siempre se obtienen redondeando hacia arriba, es decir, siempre se sobreestima las cotas de los errores.

Notación de punto flotante

La memoria de la computadora guarda los valores en lo que se conoce como **notación de punto flotante** donde los números se representan como números enteros y la posición de la coma se expresa a partir de un exponente en notación científica, léase:

$$m \cdot 10^q \tag{1}$$

Donde $\boldsymbol{q}$ es el **exponente** que ubica la posición de la coma y $\boldsymbol{m}$ es la **mantisa**, que corresponde al número entero. Así por ejemplo el número 321,5789 se representa en punto flotante como:

$$3215789 \cdot 10^4$$

Propagación de errores de entrada

Para calcular la propagación de errores absolutos de entrada a lo largo del algoritmo se utiliza la **fórmula general de propagación de errores de entrada**:

$$\Delta y = \sum_{i=1}^{n} \left| \left[\frac{\partial F}{\partial x_i} \right]_P \right| \Delta x_i \tag{2}$$

En pocas palabras, lo que se hace es derivar parcialmente la expresión $F(x_i)$ (que depende de las variables x_i) respecto de la variable x_i. Luego se reemplazan los datos de entrada en la derivada obtenida, se toma su módulo y se la multiplica por la cota del error absoluto de la variable x_i correspondiente. Esto se realiza con cada parámetro y se suman todos los resultados, lo que arroja la cota del error absoluto del resultado final.

Para las operaciones comunes: suma, resta, división y multiplicación; se tienen tabuladas las maneras en las que se propagan los errores absolutos:

Op.	Exacto		Cota	
	Absoluto	Relativo	Absoluto	Relativo
+	$\delta x_1 + \delta x_2$	$\frac{x_1}{y} r_1 + \frac{x_2}{y} r_2$	$\Delta x_1 + \Delta x_2$	$\left\|\frac{x_1}{y}\right\| R_1 + \left\|\frac{x_2}{y}\right\| R_2$
-	$\delta x_1 - \delta x_2$	$\frac{x_1}{y} r_1 - \frac{x_2}{y} r_2$	$\Delta x_1 + \Delta x_2$	$\left\|\frac{x_1}{y}\right\| R_1 + \left\|\frac{x_2}{y}\right\| R_2$
*	$x_2 \delta x_1 + x_1 \delta x_2$	$r_1 + r_2$	$\|x_2\| \Delta x_1 + \|x_1\| \Delta x_2$	$R_1 + R_2$
/	$\frac{1}{x_2} \delta x_1 - \frac{x_1}{{x_2}^2} \delta x_2$	$r_1 - r_2$	$\left\|\frac{1}{x_2}\right\| \Delta x_1 + \left\|\frac{x_1}{{x_2}^2}\right\| \Delta x_2$	$R_1 + R_2$

Donde la forma de cada operación es:

$$x_1 + x_2 = y \;;\; x_1 - x_2 = y \;;\; x_1 * x_2 = y \;;\; \frac{x_1}{x_2} = y$$

Asimismo, a partir de la tabla presentada se puede confeccionar una nueva únicamente con los factores de amplificación sujetos a cada operación básica. Se define el **factor de amplificación** $\boldsymbol{f_j}$ como aquellos coeficientes que multiplican a cada error relativo r_j.

Op.	Factores de amplificación	
	f_1	f_2
+	$\frac{x_1}{y}$	$\frac{x_2}{y}$
-	$\frac{x_1}{y}$	$-\frac{x_2}{y}$
*	1	1
/	1	-1

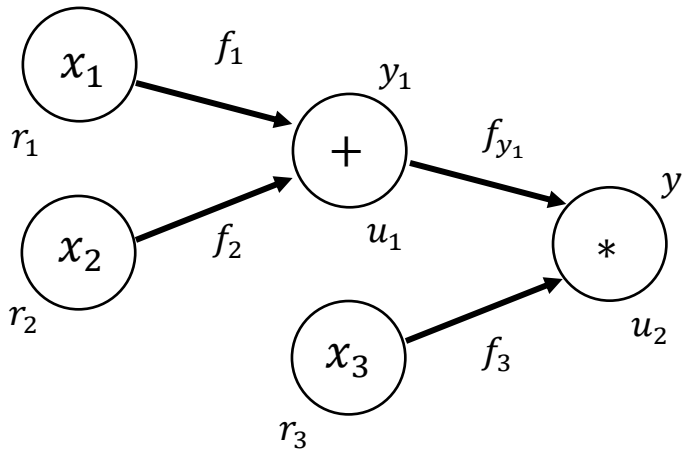
Generalizando, para conseguir el factor de amplificación de cualquier operación se hace uso de la fórmula general de propagación de errores de entrada:

$$\delta y = \frac{\partial F(x)}{\partial x} * \delta x \quad \Longrightarrow \quad r_y = \frac{\delta y}{y}; r_x = \frac{\delta x}{x} \quad \Longrightarrow \quad r_y = \frac{\frac{\partial F(x)}{\partial x} \cdot x}{y} \cdot r_x = \frac{x}{y}\frac{\partial F(x)}{\partial x} \cdot r_x$$

$$\boldsymbol{f_x = \frac{x}{y} \cdot \frac{\partial F(x)}{\partial x}} \tag{3}$$

Gráficas de procesos

Las gráficas de procesos son diagramas útiles para poder determinar cómo se propaga el error a lo largo de los cálculos de un algoritmo dado. Utiliza para ello los **errores relativos** en cada paso. Su esquema general es el siguiente:



Donde los x_i son datos de entrada y r_i son sus respectivos errores relativos. Los f_j son los factores de amplificación, los y_k son los resultados obtenidos en cada operación de cálculo y los u_k son los errores de redondeo cometidos en cada una de dichas operaciones. Entonces, analizando por pasos tenemos:

- Primera operación: suma

$$r_{y_1} = f_1 * r_1 + f_2 * r_2 + u_1$$

- Segunda operación: producto

$$r_y = f_{y_1} * r_{y_1} + f_3 * r_3 + u_2 = f_{y_1} * (f_1 * r_1 + f_2 * r_2 + u_1) + f_3 * r_3 + u_2$$

Reagrupando y planteando el módulo a ambos lados:

$$|r_y| = |f_{y_1} * f_1 * r_1 + f_{y_1} * f_2 * r_2 + f_3 * r_3 + f_{y_1} * u_1 + u_2|$$

$$|r_y| = |f_{y_1} * f_1 * r_1| + |f_{y_1} * f_2 * r_2| + |f_3 * r_3| + |f_{y_1} * u_1| + |u_2|$$

Para pasar al mundo real, tomamos cota de los errores relativos y de redondeo:

$$R_y = |r_y|;\ |r_j| \leq R_j;\ |u_i| \leq u\ (unidad\ de\ máquina)$$

$$R_y = |f_{y_1} * f_1| * R_1 + |f_{y_1} * f_2| * R_2 + |f_3| * R_3 + (|f_{y_1}| + 1) * u$$

Por lo que se puede generalizar la expresión como:

$$R_y = \sum_{j=1}^{n} |f_j| * R_j + \sum_{i=1}^{m} |g_i| * u \qquad (4)$$

Donde:

- n representa la cantidad de datos de entrada y f_j son los respectivos factores de amplificación de los errores de entrada. Ambos parámetros dependen del problema.
- m representa la cantidad de operaciones del algoritmo de cálculo y g_i son los factores de amplificación de los errores de redondeo. Ambos parámetros dependen del problema y del algoritmo empleado.

Estabilidad del Problema y del Algoritmo

Con la fórmula (4) es posible analizar la estabilidad del problema y del algoritmo empleado. Para eso, la podemos reescribir como:

$$R_y = C_P * R + F_u * u \qquad (5)$$

Donde R es una cota general para todos los errores relativos de cada variable y $\boldsymbol{F_u}$ se denomina **factor global de amplificación de errores de redondeo**. Esto permite realizar algunas consideraciones.

Estabilidad del Problema: da una idea de la sensibilidad del resultado frente a variaciones en los datos de entrada. Surge claramente de las ecuaciones expuesta que:

$$C_P = \sum_{j=1}^{n} |f_j| \qquad (6)$$

Donde $\boldsymbol{C_P}$ corresponde al **número de condición del problema** y es la sumatoria de todos los factores de amplificación de los errores relativos en módulo. **Depende solo del problema**. Por lo que:

- Si $C_P < 1$→ el problema se considera estable (bien condicionado)
- Si $C_P \gg 1$→ el problema se considera inestable (mal condicionado)

Estabilidad del Algoritmo: da una idea de la sensibilidad del resultado frente a errores de redondeo en las operaciones realizadas por la máquina. Nuevamente de las ecuaciones se observa que:

$$F_u = \sum_{i=1}^{m} |g_i|$$

Y se define:

$$C_A \equiv \frac{F_u}{C_P} \tag{7}$$

Donde $\boldsymbol{C_A}$ corresponde al **número de condición del algoritmo** y es un parámetro relativo al problema matemático planteado que permite analizar la estabilidad del algoritmo. **Depende del problema y del algoritmo empleado.** Por lo que:

- Si $C_A < 1$→ el algoritmo se considera estable
- Si $C_A \gg 1$→ el algoritmo se considera inestable
- Si $C_{A1} > C_{A2}$→ indica que el Algoritmo 1 está mejor condicionado que el Algoritmo 2. Donde $C_{A1}\ y\ C_{A2}$ representan a los números de condición de los hipotéticos algoritmos 1 y 2, respectivamente.

Además, el condicionamiento del problema numérico es aceptable si:

$$C_A \ll \frac{R}{u} \tag{8}$$

El cumplimiento de esta condición garantiza que los errores de redondeo NO están influyendo significativamente en la propagación de errores totales en el resultado final. Por ende, el costo que se paga por resolver el problema en la computadora es totalmente admisible.

Perturbaciones Experimentales

Se trata de un método utilizado para calcular de manera experimental el error relativo propagado hasta el resultado final del problema. Se utiliza el algoritmo mismo como laboratorio numérico para analizar la propagación de errores. Es especialmente práctico para casos donde el algoritmo es extenso y se realizan varias operaciones, por lo que resultaría inviable hacer una gráfica de procesos. Para ello:

1) Se realizan pequeñas variaciones (perturbaciones) en los valores de los parámetros iniciales, modificando una sola variable a la vez y manteniendo las demás constantes.
2) Se corre el programa con dicha modificación y se observa la diferencia generada en el resultado final.
3) En base a dicha diferencia se consigue, por medio de ciertas fórmulas, determinar de forma aproximada la cota del error relativo propio del resultado obtenido.

Se define la perturbación relativa de la variable x_j (ρ_j) como:

$$\rho_j \equiv \left|\frac{\varepsilon_j}{x_j}\right|$$

Donde ε_j representa la pequeña perturbación aportada al valor original de la variable x_j. El valor de la perturbación agregada suele ser similar al error absoluto de la variable perturbada. Además, se debe definir:

$$\omega_j \equiv \left|\frac{Y_j - Y}{Y}\right|$$

Donde ω_j representa la variación relativa del resultado por la perturbación de x_j, Y_j es el resultado obtenido al modificar x_j, e Y denota el valor original obtenido sin perturbación alguna.

Ahora bien, suponiendo despreciable el error de redondeo, se tiene que:

$$|f_j| \approx \frac{\omega_j}{\rho_j}$$

Siendo f_j el factor de amplificación del error de entrada x_j, ergo se tendrá tantos f_j como variables posea el problema analizado. Para verificar esta hipótesis se debería observar que al disminuir ε_j, el $|f_j|$ tienda a una constante. Si no se cumple, indica inestabilidad.

En cuanto a cómo afectan los errores de redondeo, lo que se hace es:

1) Se realiza el cálculo con la precisión elegida. Digamos t dígitos.
2) Se vuelve a realizar el cálculo con los mismos datos, pero con doble precisión, es decir, utilizando $2t$ dígitos.

Entonces se plantea:

$$\omega_{2t} \equiv \left|\frac{Y_{2t} - Y}{Y_{2t}}\right|$$

Donde ω_{2t} es la variación relativa del resultado por el cambio de precisión, Y es el resultado del paso 1) e Y_{2t} es el resultado del paso 2). Entonces, se tiene que:

$$F_u \approx \frac{\omega_{2t}}{u}$$

Siendo u la unidad de máquina y F_u el factor global de amplificación de errores de redondeo. De esta manera es posible, con todas las fórmulas expuestas, analizar la estabilidad del problema numérico planteado. En consecuencia, la expresión (4) puede reescribirse para este caso como:

$$R_y \approx \sum_{j=1}^{n} \left|\frac{\omega_j}{\rho_j}\right| * R + \frac{\omega_{2t}}{u} * u \qquad (9)$$

Ecuaciones NO lineales (Guía 3.a)

El problema tipo a resolver es:

$$Hallar\ x\ /\ F(x) = 0$$

Donde $F(x)$ es una función NO lineal cualquiera y x es el cero o raíz de dicha función.

Para la resolución de ecuaciones no lineales se hace uso de lo que se denomina métodos iterativos.

Un **método iterativo** es una solución analítica que se plantea para resolver un problema matemático mediante aproximaciones sucesivas a una solución, empezando a partir de ciertas condiciones iniciales impuestas (semilla, intervalo, etc.). Las soluciones obtenidas mediante estos métodos no son exactas, sino que cargan consigo cierto error.

Convergencia y divergencia

Se entiende por **convergencia** a la tendencia que presenta un algoritmo de acercarse a un número a través de las sucesivas iteraciones. Por el otro lado, se habla de **divergencia** cuando el algoritmo tiende a infinito, es decir, cuando no converge.

Orden de convergencia

El orden de convergencia brinda una idea de que tan rápido se reducen los errores de una iteración a la siguiente, es decir, habla sobre la velocidad de convergencia que posee un método. Se define como:

$$Si\ \exists\ P \neq 0\ y\ C \neq 0\ /\ \lim_{k\to\infty} \frac{e_{k+1}}{{e_k}^P} = C$$

Donde e_k representa al error absoluto en la iteración k, definido como $e_k = x_k - x_{k-1}$, mientras que $\boldsymbol{P}$ representa al **orden de convergencia** y $\boldsymbol{C}$ es la **constante asintótica del error**.

En el mundo real, la constante C se puede aproximar como el cociente entre el error absoluto en el paso $\mathrm{k}+1$ y el del paso k, de la siguiente manera:

$$\frac{e_{k+1}}{{e_k}^P} = \mathrm{C} = \frac{e_k}{{e_{k-1}}^P} \tag{10}$$

Aplicando logaritmo a ambos lados:

$$\ln(e_{k+1}) - P * \ln(e_k) = \ln(e_k) - P * \ln(e_{k-1})$$

$$\ln\left(\frac{e_{k+1}}{e_k}\right) = \ln(e_{k+1}) - \ln(e_k) = P * \ln\left(\frac{e_k}{e_{k-1}}\right)$$

Entonces, se puede aproximar P mediante la siguiente fórmula:

$$P = \frac{\ln\left[\frac{e_{k+1}}{e_k}\right]}{\ln\left[\frac{e_k}{e_{k-1}}\right]} \tag{11}$$

El orden de convergencia puede tomar 3 valores:

- $P = 1$→ orden 1 o lineal.
- $P = 2$→ orden 2 o cuadrático.
- $1 \leq P \leq 2$→ orden superlineal, el valor de P varía entre 1 y 2.

Donde a mayor orden, mayor será la velocidad de convergencia del método empleado. Es decir, menor será la cantidad de iteraciones k necesarias para alcanzar una precisión deseada ya que los dígitos significativos que se ganan de una iteración a otra es mayor cuanto mayor sea el orden de convergencia.

Métodos de arranque

En primer lugar, se cuenta con los **métodos de arranque** denominados así por su sencillez en el algoritmo y porque generalmente son empleados para encontrar una semilla y luego utilizarla en un método iterativo más complejo y eficiente.

Estos métodos parten del **Teorema de Bolzano** que establece que:

$$Dada\ una\ función\ F(x)\ continua\ en\ un\ intervalo\ [a,b].$$

$$Si\ F(a)\cdot F(b)<0 \Longrightarrow \exists\ \alpha \in [a,b]\ /\ F(\alpha)=0$$

En otras palabras, si la imagen por $F(x)$ evaluada en los extremos "a" y "b" posee signos opuestos, por ejemplo: a positivo y b negativo, está garantizada la existencia de un número impar de raíces α dentro del intervalo $[a,b]$.

Método de Bisección

Dado un intervalo inicial $[a_0,b_0]$ donde $F(x)$ es continua, sabiendo que existe una raíz de la función y suponiendo que esta última está dada por:

$$\alpha = m_{k+1} \pm \Delta m_{k+1}$$

Donde α representa a la raíz exacta y m_{k+1} es el punto medio del intervalo:

$$m_{k+1}=\frac{b_{k+1}+a_{k+1}}{2}$$

Ergo, $\boldsymbol{m_{k+1}}$ **es el estimador de la raíz**. Se plantean los pasos del algoritmo para la iteración k:

1) Se evalúa la función en los extremos a, b y en m_k.
2) Si $F(m_k)=0$, se halló la raíz. Sino se continua con el paso 3).
3) Se evalúa el cumplimiento del Teorema de Bolzano entre los extremos y m_k:

$$Si\ F(m_k)*F(a_k)<0 \rightarrow se\ redefine\ b_{k+1}=m_k$$

$$Si\ F(m_k)*F(b_k)<0 \rightarrow se\ redefine\ a_{k+1}=m_k$$

4) Se repiten los pasos con los nuevos extremos del intervalo para $k+1$.

Error relativo y absoluto

La **cota para el error absoluto** generada por el error de truncamiento en el paso iterativo $k+1$ esta dada por:

$$\Delta m_{k+1}=\frac{b_{k+1}-a_{k+1}}{2}$$

Donde a_{k+1} y b_{k+1} son los extremos del intervalo en el paso $k+1$. Además, para este algoritmo es equivalente plantear:

$$\Delta m_{k+1}=\frac{b_0-a_0}{2^{k+1}}$$

Donde a_0 y b_0 son los extremos del intervalo inicial en el paso $k = 0$ y el exponente del denominador indica la cantidad de pasos iterativos realizados hasta alcanzar el valor deseado de m_{k+1}.

Esta última expresión, permite calcular la cantidad de pasos iterativos necesarios para alcanzar cierta precisión deseada. Pues dado un valor para la cota del error absoluto y el intervalo inicial, es posible determinar el valor de k.

Asimismo, la **cota para el error relativo** en el paso iterativo $k + 1$ esta dada por:

$$r_{k+1} = \frac{\Delta m_{k+1}}{m_{k+1}}$$

Ventajas

Este método tiene la particularidad de que es incondicionalmente convergente, ya que, la raíz siempre se encuentra dentro del intervalo, por lo que viéndolo desde el mundo ideal si $k \to \infty$ entonces $m_{k+1} \to \alpha$.

Desventajas

Requiere muchas iteraciones para aumentar su precisión considerablemente, ya que su orden de convergencia es 1. En consecuencia, el método es lento.

Método de Regula-Falsi

Se formula con el objetivo de acelerar el proceso de convergencia del Método de Bisección. Los pasos realizados para una iteración k son los mismos que en el caso anterior. La diferencia radica en que ahora el estimador de la raíz m_{k+1} surge de la intersección entre la recta que pasa por los puntos generados por los extremos del intervalo $[a_k, b_k]$ y sus imágenes, con el eje de abscisas. Es decir, partiendo de la ecuación de una recta:

$$(y_2 - y_1) = m * (x_2 - x_1)$$

Evaluando $F(x)$ en los extremos del intervalo $[a, b]$ para conseguir la pendiente m:

$$\frac{F(b) - F(a)}{b - a} = m$$

Reemplazando en la ecuación de la recta, donde un punto corresponderá a la intersección con el eje de abscisas $(x, 0)$ y el otro será uno de los extremos del intervalo, por ejemplo, $(a, F(a))$:

$$(F(a) - 0) = \frac{F(b) - F(a)}{b - a}(a - x) \to F(a) * \frac{b - a}{F(b) - F(a)} = a - x$$

$$x = a - (b - a)\frac{F(a)}{F(b) - F(a)}$$

Se obtiene así, la **fórmula para el Método de Regula-Falsi**:

$$m_{k+1} = a_k - (b_k - a_k)\frac{F(a_k)}{F(b_k) - F(a_k)} \qquad (12)$$

El algoritmo empleado es el mismo que en el Método de Bisección.

Error absoluto y relativo

La **cota para el error absoluto** generada por el error de truncamiento en el paso iterativo $k + 1$ esta dada por:

$$\Delta m_{k+1} = \frac{b_{k+1} - a_{k+1}}{2}$$

Donde a_{k+1} y b_{k+1} son los extremos del intervalo en el paso $k+1$. Además, para este algoritmo es equivalente plantear:

$$\Delta m_{k+1} = |m_{k+1} - m_k|$$

La expresión del método de bisección no aplica ya que los intervalos no se van reduciendo proporcionalmente y por lo tanto no hay garantía de que m_{k+1} sea el punto medio del intervalo.

Por otro lado, la **cota para el error relativo** en el paso iterativo $k+1$ esta dada por:

$$r_{k+1} = \frac{\Delta m_{k+1}}{m_{k+1}} = \frac{|m_{k+1} - m_k|}{m_{k+1}}$$

Ventajas

La raíz siempre se encuentra contenida en el intervalo $[a_k, b_k]$, por lo que el método resulta incondicionalmente convergente.

Desventajas

El método posee un orden de convergencia lineal, es decir, aún no es lo suficientemente rápido.

Métodos de Punto Fijo

Estos métodos surgen con el fin de acelerar el proceso de convergencia y promueven la resolución del problema original por medio del análisis de un problema matemáticamente equivalente.

Este método plantea la solución de una ecuación de partida $F(x)$, basada en la existencia de un punto fijo, léase, un valor β perteneciente al eje de abscisas cuya imagen a través de una función $g(x)$ es igual a β en el eje de ordenadas:

$$\beta \in [a, b] \,/\, g(\beta) = \beta$$

Explicado coloquialmente, se está buscando el punto de intersección entre la mencionada función $g(x)$ y la recta $Y = X$. Siendo una de las formas de $g(x)$ la obtenida a partir de:

$$g(x) = x - q \cdot F(x) \tag{13}$$

Donde q es una constante no nula.

Principio de equivalencia

A continuación, se demuestra la equivalencia del planteo mencionado:

$$Si\ \beta\ es\ punto\ fijo\ de\ g \rightarrow g(\beta) = \beta \rightarrow g(\beta) = \beta - q * F(\beta) \rightarrow g(\beta) - \beta = -q * F(\beta)$$

$$pero\ g(\beta) - \beta = 0\ \rightarrow\ 0 = -q * F(\beta) \leftrightarrow F(\beta) = 0 \rightarrow \therefore\ \boldsymbol{\beta\ es\ raíz\ de\ F}$$

Por el otro lado:

$$Si\ \alpha\ es\ raíz\ de\ F \rightarrow F(\alpha) = 0 \rightarrow g(\alpha) = \alpha - q * F(\alpha)$$

$$pero - q * F(\alpha) = 0\ \rightarrow\ g(\alpha) = \alpha \rightarrow \therefore\ \boldsymbol{\alpha\ es\ punto\ fijo\ de\ g}$$

Es decir, $\alpha = \beta$ entonces:

$$\beta\ es\ punto\ fijo\ de\ g(x) \Leftrightarrow \beta\ es\ raíz\ de\ F(x)$$

Formas de obtener las funciones $g(x)$

La expresión más común es la antes mencionada, tomando $q = 1$:

$$g(x_k) = x_k - F(x_k) \tag{14}$$

También se puede obtener despejando alguna de las variables x de la función $F(x)$. Más adelante se verán nuevas maneras de determinar funciones $g(x)$.

Teorema de Punto Fijo (Condiciones de convergencia)

Existen dos condiciones suficientes pero no necesarias que garantizan la convergencia del método de punto fijo en un intervalo $[a, b]$ para cualquier semilla dentro del intervalo.

1) **Condición de existencia**: garantiza que existe al menos una raíz dentro del intervalo.

$$Debe\ existir\ una\ función\ g(x)\ continua\ en\ [a,b]\ /\ g_{(x)} \epsilon [a,b]\ \forall\ x \epsilon [a,b]$$

2) **Condición de unicidad**: garantiza que dicha raíz es única.

$$Debe\ \exists g'_{(x)}\ \in (a,b)\ y\ \exists\ 0 < K < 1\ /\ |g'_{(x)}| \leq K < 1\ \ \forall\ x \epsilon [a,b]$$

Corolario (error de truncamiento)

En las condiciones anteriores, una cota para el error cometido en el enésimo paso al aproximar β por x_n es:

$$|x_n - \beta| \leq K^n \cdot \max(x_0 - a, b - x_0)$$

Donde si $K \to 1$ el método es lento, mientras que si $K \to 0$ el método será más rápido.

Otra forma de expresar el error de truncamiento es:

$$|x_{k+1} - \beta| \leq \frac{K}{1-K} |x_{k+1} - x_k|$$

La cual surge del siguiente desarrollo:

Si β es punto fijo:

$$x_{k+1} - \beta = g_{(x_k)} - \beta = g'_{(\gamma_k)}(x_k - \beta) = g'_{(\gamma_k)}(x_k - \beta + x_{k+1} - x_{k+1})$$

$$x_{k+1} - \beta = g'_{(\gamma_k)}(x_k - x_{k+1}) + g'_{(\gamma_k)}(x_{k+1} - \beta)$$

Si el método es convergente se cumple que $|g'_{(x)}| \leq K < 1$ entonces puedo reescribir:

$$|x_{k+1} - \beta| \leq |g'_{(\gamma_k)}||x_k - x_{k+1}| + |g'_{(\gamma_k)}||x_{k+1} - \beta|$$

$$|x_{k+1} - \beta| \leq K|x_k - x_{k+1}| + K|x_{k+1} - \beta|$$

$$|x_{k+1} - \beta|(1 - K) = |x_{k+1} - \beta| - K|x_{k+1} - \beta| \leq K|x_k - x_{k+1}|$$

$$|x_{k+1} - \beta| \leq \frac{K}{1-K} |x_{k+1} - x_k|$$

Algoritmo

Conociendo el intervalo $[a_0, b_0]$ donde $F(x)$ es continua y habiendo comprobado que se cumplen las condiciones de convergencia para la función $g(x_k)$ siendo:

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

Se plantean los pasos del algoritmo:

1) Para $k = 0$, se elige una semilla (valor inicial) $x_0 \in [a, b]$.
2) Se evalúa x_k en la función $g(x_k)$:

$$g(x_0) = x_0 - F(x_0) = x_1 = x_{k+1}$$

3) Se repite el paso 2) con el nuevo valor x_{k+1} hasta cumplir con la condición de corte impuesta.

Error absoluto y relativo

La **cota para el error absoluto** generada por el error de truncamiento en el paso iterativo $k + 1$ esta dada por:

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

En otras palabras, es el módulo de la diferencia entre el valor evaluado en el paso x_{k+1} y el x_k obtenido en el paso anterior. Asimismo, la **cota para el error relativo** en el paso iterativo $k + 1$ esta dada por:

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

Ventajas

Este método aumenta la velocidad de convergencia respecto de los métodos de arranque.

Desventajas

El método no es incondicionalmente convergente, sino que debe cumplir con ciertas condiciones para garantizar su convergencia. Requiere la existencia de la derivada de $g_{(x)}$.

Su orden de convergencia sigue siendo 1.

Método de Newton-Raphson

Es una variación del método de punto fijo donde se propone la forma de $g_{(x)} \in [a, b]$ tal que:

$$g_{NR}(x_k) = x_k - \frac{F(x_k)}{F'(x_k)} \qquad (15)$$

Condiciones de convergencia

Además de cumplir con las condiciones comunes para cualquier método de punto fijo, debe cumplir que:

1) $\exists\, F'(x) \neq 0\ \forall\, x \in (a, b)\ tal\ que\ \exists\, g_{NR}(x)$
2) $\exists\, F''(x)\ \forall\, x \in (a, b)\ tal\ que\ \exists\, {g_{NR}}'(x)$. Pues:

$${g_{NR}}'(x) = 1 - \frac{{F'}_{(x)} * {F'}_{(x)} - F_{(x)} * {F''}_{(x)}}{{{F'}_{(x)}}^2} = \frac{{{F'}_{(x)}}^2 - {{F'}_{(x)}}^2 + F_{(x)} * {F''}_{(x)}}{{{F'}_{(x)}}^2}$$

$$g_{NR}{}'(x) = \frac{F_{(x)} * F''_{(x)}}{F'_{(x)}{}^2}$$

3) $|g_{NR}{}'(x)| = \left|\frac{F_{(x)} * F''_{(x)}}{F'_{(x)}{}^2}\right| < 1 \; \forall \; x \in (a, b).$

Ventajas

Aumenta considerablemente la velocidad de convergencia, siendo un método de orden 2.

Desventajas

Requiere la existencia de la derivada segunda de $F(x)$, ya que para verificar la segunda condición de convergencia se precisa conocer $g'(x)$ la cual viene dada por la expresión antes desarrollada.

Método de la secante

Surge como equivalencia al Método de Newton Raphson. Evita el cálculo de la derivada de $F(x)$ proponiendo aproximar su valor según:

$$F'(x) \approx \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}} \tag{16}$$

Por lo que reemplazando en la expresión (15):

$$g(x_k) = x_k - \frac{F(x_k)}{\frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}} = x_k - (x_k - x_{k-1})\frac{F(x_k)}{F(x_k) - F(x_{k-1})}$$

Resulta entonces:

$$g_{Sec}(x_k) = x_{k+1} = x_k - (x_k - x_{k-1})\frac{F(x_k)}{F(x_k) - F(x_{k-1})} \tag{17}$$

Ventajas

Consigue una buena velocidad de convergencia sin necesidad de conocer la derivada de $F(x)$.

Desventajas

Pierde la garantía del orden de convergencia cuadrático pasando a tener un orden superlineal, es decir, varía entre orden 1 y 2.

Debido a su expresión no es conveniente evaluar las condiciones del teorema de punto fijo.

Requiere conocer dos semillas x_0 para poder iniciar con las iteraciones, por lo que es necesario utilizar algún método de arranque.

No es incondicionalmente convergente.

Comparación entre métodos iterativos

Método	Ventajas	Desventajas
Tablas y gráficos	Fácil y rápido	No tienen precisión
Bisección	Fácil implementación Incondicionalmente convergente	Convergencia lenta P=1
Regula-Falsi	Fácil implementación Incondicionalmente convergente Pocos requisitos	Convergencia lenta P=1
Punto fijo (común)	Fácil aplicación	Convergencia condicionada P=1
Newton-Raphson	P=2 Aplicables en muchos casos prácticos de ingeniería	Muchos requisitos de convergencia Requiere derivada de F(x)
Secante	1<P<2 No requiere derivad de F(x)	Convergencia condicionada Pierde P=2 Requiere dos semillas para arrancar

Aproximaciones (Guías 4 y 5)

El problema tipo a resolver es:

$$Aproximar\ f\ utilizando\ funciones\ "convenientes": f \approx f^*$$

Donde $\boldsymbol{f^*}$ se denomina **función aproximante** y es la función que se busca determinar explícitamente.

Teoría lineal de la Aproximación

La función aproximante f^* es una combinación lineal de otras funciones que generalmente son funciones NO lineales:

$$f^*(x) = C_0 * \varphi_{0(x)} + C_1 * \varphi_{1(x)} + \cdots + C_n * \varphi_{n(x)} = \sum_{j=0}^{n} C_j \varphi_{j(x)} \tag{18}$$

Donde $\varphi_{j(x)}$ son $n+1$ funciones NO lineales y C_j son $n+1$ parámetros **a determinar.**

Objetivo de la aproximación discreta

Se busca determinar los $n+1$ coeficientes C_j tal que se cumpla de forma exacta o lo más próxima posible que:

$$f^*_{(x_i)} = f_{(x_i)} \ \ \forall x_i\ con\ i = 0, \cdots, m$$

Donde x_i representa a los $m+1$ datos disponibles. De esta forma, se puede plantear para cada i una función aproximante tal que:

$$\begin{cases} i = 0 \rightarrow C_0 \cdot \varphi_{0(x_0)} + C_1 \cdot \varphi_{1(x_0)} + \cdots + C_n \cdot \varphi_{n(x_0)} = f_{(x_0)} \\ i = 1 \rightarrow C_0 \cdot \varphi_{0(x_1)} + C_1 \cdot \varphi_{1(x_1)} + \cdots + C_n \cdot \varphi_{n(x_1)} = f_{(x_1)} \\ \vdots \\ i = m \rightarrow C_0 \cdot \varphi_{0(x_m)} + C_1 \cdot \varphi_{1(x_m)} + \cdots + C_n \cdot \varphi_{n(x_m)} = f_{(x_m)} \end{cases}$$

O en forma matricial:

$$A \cdot c = b$$

$$\begin{bmatrix} \varphi_{0(x_0)} & \varphi_{1(x_0)} & \cdots & \varphi_{n(x_0)} \\ \varphi_{0(x_1)} & \varphi_{1(x_1)} & \cdots & \varphi_{n(x_1)} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{0(x_m)} & \varphi_{1(x_m)} & \cdots & \varphi_{n(x_m)} \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} f_{(x_0)} \\ f_{(x_1)} \\ \vdots \\ f_{(x_m)} \end{bmatrix} \tag{19}$$

Donde A es una matriz de números, ya que las funciones φ_i están evaluadas en los x_i datos, de dimensión $(m+1)x(n+1)$; c es un vector incógnita de dimensión $(n+1)x1$ y b es un vector de números, ya que se encuentra compuesto de las imágenes de los datos x_i, y es de dimensión $(m+1)x1$.

A raíz de este planteo se presentan tres posibilidades:

- $m < n$ → sistema compatible indeterminado: existen infinitas soluciones.
- $m = n$ → sistema compatible determinado: existe solución única. La matriz es cuadrada pues hay igual cantidad de datos que incógnitas → se aplica **Interpolación.**
- $m > n$ → problema sobre- determinado: existen más datos m que incógnitas n, por lo que la solución es aproximada → se aplica **Ajuste.**

Ajuste discreto por cuadrados mínimos (Guía 4)

Se aplica para salvar la sobre determinación en el caso $m > n$, es decir, cuando existen más datos $m+1$ que parámetros C_i a determinar, con $i = 0, \dots, n$. Se busca minimizar la diferencia entre la función aproximante y la nube de puntos (datos disponibles).

Gráficamente el problema es el de la Figura 1, donde los puntos representan los $m+1$ datos y la línea continua corresponde a la función aproximante f^*. Para un dato x_i cualquiera, su **error absoluto** se puede calcular por medio de la expresión:

$$e_i = f^*_{(x_i)} - f_{(x_i)}$$

En consecuencia, realizar un ajuste implica encontrar la función aproximante f^* tal que el error global absoluto sea el mínimo posible. Se proponen entonces, las siguientes tres expresiones para el cálculo del **error global absoluto** (e)**:**

$$e^I = \sum_{i=0}^{m} e_i \quad ; \quad e^{II} = \sum_{i=0}^{m} |e_i| \quad ; \quad e^{III} = \sum_{i=0}^{m} {e_i}^2$$

La **expresión e^I** es la primera que se nos viene a la cabeza, es decir sumar todos los errores absolutos de cada punto x_i, no obstante los mismos tienen signo podría darse el caso de que la sumatoria de todos sea nula, por lo tanto la expresión e^I no sirve. Se propone entonces la **fórmula e^{II}** donde se resuelve este inconveniente sumando los módulos de cada error, sin embargo, como el problema de minimización implica luego la derivada del error global absoluto, y la derivada del módulo puede resultar bastante complicada, se prescinde de esta fórmula. Finalmente se propone la **expresión e^{III}**, la cual tiene resuelto el problema de los signos (pues todos los sumandos serán positivos por el cuadrado) y además es fácilmente derivable. En consecuencia, se adopta esta expresión $e^{III} \equiv e$ para el cálculo del error global absoluto y de aquí se debe el nombre de **ajuste por cuadrados mínimos,** pues el problema consistirá en minimizar la siguiente expresión:

$$e = \sum_{i=0}^{m} {e_i}^2 \qquad (20)$$

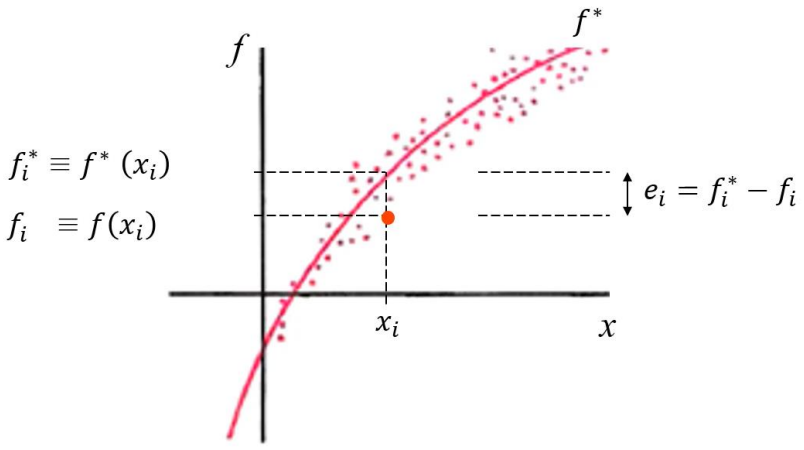


Figura 1 - Representación de una nube de datos y su función de ajuste por cuadrados mínimos.

Se desarrolla la expresión (20) para el **error global absoluto** de la siguiente forma:

$$e = \sum_{i=0}^{m} {e_i}^2 = \sum_{i=0}^{m} \left[{f^*}_{(x_i)} - f_{(x_i)} \right]^2 = \sum_{i=0}^{m} \left[f_{(x_i)} - {f^*}_{(x_i)} \right]^2$$

Reemplazando la expresión (18) dada para la función aproximante $f^*_{(x)}$:

$$e = \sum_{i=0}^{m} \left[f_{(x_i)} - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right]^2 \tag{21}$$

Analizamos la expresión. Se observa que la primer sumatoria depende de la cantidad de datos que dispongamos, por lo tanto m es un valor fijo. Luego contamos con el término $f_{(x_i)}$ el cual también depende de los datos disponibles y es información fija, al igual que las funciones $\varphi_{j(x_i)}$. Por ende, los únicos parámetros a determinar son los coeficientes C_j, por ende se tiene que el error global es función de dichos parámetros, es decir $e = e_{(C_j)}$. En consecuencia, para minimizar el error debemos derivar la fórmula respecto de C_j e igualarla a cero:

$$\frac{\partial e}{\partial C_k} = 0 \quad con \quad k = 0, \dots, n$$

Obtención de las Ecuaciones Normales

Procedemos a derivar la fórmula (21) según lo antes expuesto:

$$\frac{\partial e}{\partial C_k} = \frac{\partial}{\partial C_k} \left\{ \sum_{i=0}^{m} \left[f_{(x_i)} - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right]^2 \right\} = \frac{\partial}{\partial C_k} \left\{ \sum_{i=0}^{m} [R_i]^2 \right\} = \sum_{i=0}^{m} \frac{\partial [R_i]^2}{\partial C_k} = 0$$

$$\frac{\partial e}{\partial C_k} = 2R_i \frac{\partial R_i}{\partial C_k} = \left\{ \sum_{i=0}^{m} 2 \left[f_{(x_i)} - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right] \right\} \cdot \frac{\partial}{\partial C_k} \left\{ - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right\} = 0$$

Haciendo uso de la **delta de Kronecker:**

$$\frac{\partial C_j}{\partial C_k} = \delta_{jk} = \begin{cases} 0 \; si \; j \neq k \\ 1 \; si \; j = k \end{cases} \tag{22}$$

Es una función de dos variables que vale 1 si $j = k$ y 0 si $j \neq k$. Entonces:

$$\frac{\partial e}{\partial C_k} = 2 \sum_{i=0}^{m} \left[f_{(x_i)} - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right] \cdot \left[- \sum_{j=0}^{n} \frac{\partial C_j}{\partial C_k} \varphi_{j(x_i)} \right] = 0$$

$$\frac{\partial e}{\partial C_k} = 2 \sum_{i=0}^{m} \left[f_{(x_i)} - \sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right] \cdot \left[-\varphi_{k(x_i)} \right] = 0$$

$$\frac{\partial e}{\partial C_k} = 2 \sum_{i=0}^{m} \left[\varphi_{k(x_i)} \left(\sum_{j=0}^{n} C_j \varphi_{j(x_i)} \right) - \varphi_{k(x_i)} f_{(x_i)} \right] = 0$$

$$\frac{\partial e}{\partial C_k} = 2 \sum_{i=0}^{m} \left[\sum_{j=0}^{n} C_j \varphi_{j(x_i)} \varphi_{k(x_i)} \right] - 2 \sum_{i=0}^{m} \left[f_{(x_i)} \varphi_{k(x_i)} \right] = 0$$

Reescribimos la sumatoria del segundo término como si fuese el producto interno entre dos vectores, pues:

$$f_{(x_0)}\varphi_{k(x_0)} + f_{(x_1)}\varphi_{k(x_1)} + \cdots + f_{(x_m)}\varphi_{k(x_m)} = [f_{(x_0)} \quad f_{(x_1)} \quad \dots \quad f_{(x_m)}] \begin{bmatrix} \varphi_{k(x_0)} \\ \varphi_{k(x_1)} \\ \dots \\ \varphi_{k(x_m)} \end{bmatrix} = \langle f_{(x_i)}, \varphi_{k(x_i)} \rangle$$

$$\varphi_{j(x_0)}\varphi_{k(x_0)} + \cdots + \varphi_{j(x_m)}\varphi_{k(x_m)} = [\varphi_{j(x_0)} \quad \dots \quad \varphi_{j(x_m)}] \begin{bmatrix} \varphi_{k(x_0)} \\ \dots \\ \varphi_{k(x_m)} \end{bmatrix} = \langle \varphi_{j(x_i)}, \varphi_{k(x_i)} \rangle = \langle \varphi_{k(x_i)}, \varphi_{j(x_i)} \rangle$$

Donde los tres vectores $f_{(x_i)}, \varphi_{k(x_i)}, \varphi_{j(x_i)}$ son de dimensiones $(m \text{ x } 1)$. Entonces:

$$\frac{\partial e}{\partial C_k} = 2 \sum_{j=0}^{n} \left[C_j \sum_{i=0}^{m} \varphi_{j(x_i)} \varphi_{k(x_i)} \right] - 2 \langle f_{(x_i)}, \varphi_{k(x_i)} \rangle = 0$$

En el primer término se conmutaron las sumatorias. Lo mismo cabe para el primer término, por lo que:

$$\frac{\partial e}{\partial C_k} = 2 \sum_{j=0}^{n} \left[C_j \langle \varphi_{j(x_i)}, \varphi_{k(x_i)} \rangle \right] - 2 \langle f_{(x_i)}, \varphi_{k(x_i)} \rangle = 0 \Longrightarrow \sum_{j=0}^{n} C_j \langle \varphi_{j(x_i)}, \varphi_{k(x_i)} \rangle = \langle f_{(x_i)}, \varphi_{k(x_i)} \rangle$$

$$\sum_{j=0}^{n} C_j \langle \varphi_{k(x_i)}, \varphi_{j(x_i)} \rangle = \langle \varphi_{k(x_i)}, f_{(x_i)} \rangle \quad con \quad \begin{cases} k = 0, \cdots, n \\ i = 0, \cdots, m \end{cases} \tag{23}$$

Siendo esta última la expresión de las **Ecuaciones Normales.** Donde $\langle , \rangle$ indica el producto interno entre los elementos. En consecuencia, se cuenta con $n + 1$ ecuaciones normales y $n + 1$ incógnitas; y para cada valor de k se obtiene una fila de la expresión matricial. Pues la expresión (23) conforma un sistema de $(n + 1)x(n + 1)$ y se salva la sobre-determinación.

Propiedades de las Ecuaciones Normales

- Garantía de existencia y unicidad de la $f^*_{(x)}$ solución:

$$Si\ los\ \varphi_j\ son\ Linealmente\ Independientes \rightarrow f^*_{(x)}\ es\ \text{única}$$

- Ortogonalidad de la matriz A:

$$\begin{cases} < \varphi_j, \varphi_k > = \|\varphi_k\| * \delta_{jk} \\ C_k = \dfrac{< f, \varphi_k >}{\|\varphi_k\|^2} \end{cases}$$

- Ortonormalidad de la matriz A:

$$\begin{cases} < \varphi_j, \varphi_k > = \delta_{jk} \\ C_k = < f, \varphi_k > \end{cases}$$

Construcción matricial

$$A_{jk} * c = b_k$$

$$\begin{bmatrix} < \varphi_0, \varphi_0 > & < \varphi_0, \varphi_1 > & \cdots & < \varphi_0, \varphi_n > \\ < \varphi_1, \varphi_0 > & < \varphi_1, \varphi_1 > & \cdots & < \varphi_1, \varphi_n > \\ \vdots & \vdots & \ddots & \vdots \\ < \varphi_n, \varphi_0 > & < \varphi_n, \varphi_1 > & \cdots & < \varphi_n, \varphi_n > \end{bmatrix} * \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} < f, \varphi_0 > \\ < f, \varphi_1 > \\ \vdots \\ < f, \varphi_n > \end{bmatrix}$$

Donde A_{jk} es una **matriz cuadrada simétrica** de la forma $\langle \varphi_k, \varphi_j \rangle$ de dimensión $(n + 1)x(n + 1)$, c es el vector de coeficientes incógnita de dimensión $(n + 1)x1$ y b_k es un vector de términos independientes de la forma $< f, \varphi_k >$ y es de dimensión $(n + 1)x1$.

Errores globales

El **error absoluto para el punto x_i** esta dado por:

$$e_i = f^*_{(x_i)} - f_{(x_i)}$$

En consecuencia, el **error cuadrático global absoluto** de la función aproximante se determina como:

$$\|e\| = \sqrt{\sum_{i=0}^{m} {e_i}^2} = \sqrt{\sum_{i=0}^{m} \left[f^*_{(x_i)} - f_{(x_i)}\right]^2} \tag{24}$$

Es decir, se toma la norma de la sumatoria de los errores absolutos propios de cada punto x_i.

Del mismo modo, el **error relativo para el punto x_i** esta dado por:

$$r_i = \frac{e_i}{x_i} = \frac{f^*_{(x_i)} - f_{(x_i)}}{x_i}$$

Por otro lado, el **error cuadrático global relativo** de la función aproximante resulta:

$$\|r\| = \frac{\|e\|}{\|f\|} \tag{25}$$

Donde la expresión del denominador es:

$$\|f\| = \sqrt{\sum_{i=0}^{m} {f_{(x_i)}}^2} \tag{26}$$

Es decir, se toma la norma de la sumatoria de las imágenes de los puntos x_i.

Caso funciones potenciales

Para este tipo de funciones no se pueden aplicar las ecuaciones normales de manera directa, por lo que una alternativa es utilizar una escala logarítmica. Es decir, si la función aproximante es de la forma:

$$f^*_{(x)} = a * x^P \rightarrow f_{(x_i)} = a * {x_i}^P$$

Aplicando logaritmo:

$$\ln\left(f_{(x_i)}\right) = \ln(a) + P * \ln\left(x_i\right)$$

Es decir, queda de la forma:

$$Y_i = C_0 + C_1 * Z_1$$

Y es posible aplicar las ecuaciones normales.

Ajuste continuo por cuadrados mínimos

Suponiendo que se conoce la función $f_{(x)} \in [a, b]$ se busca el **polinomio $P_{n(x)}$** que minimice el error global. El objetivo del ajuste continuo es reducir el costo computacional del cálculo de una función complicada, por ello es que se ajusta con polinomios, que son las funciones más simples.

Proponiendo la misma forma para el error global absoluto que antes (20) se tiene que:

$$E = \int_a^b e^2 = \int_a^b \left[f_{(x)} - P_{n(x)}\right]^2 dx \tag{27}$$

Siendo $P_{n(x)} = \sum_{k=0}^{n} a_k\, x^k$ resulta:

$$E = \int_a^b \left[f_{(x)} - \sum_{k=0}^{n} a_k \cdot x^k\right]^2 dx$$

Desarrollando el binomio al cuadrado:

$$E = \int_a^b \left[f_{(x)}\right]^2 dx - 2\int_a^b \left[f_{(x)} \sum_{k=0}^{n} a_k \cdot x^k\right] dx + \int_a^b \left[\sum_{k=0}^{n} a_k \cdot x^k\right]^2 dx$$

Como E es función de los a_j con $j = 0, \cdots, n$; para minimizar el error global planteamos que $\frac{\partial E}{\partial a_j} = 0$ con $j = 0, \cdots, n$. Entonces haciendo uso de la **delta de Kronecker** (22) derivo:

$$\frac{\partial E}{\partial a_j} = 0 - 2\int_a^b \left[f_{(x)} x^j\right] dx + \int_a^b 2\left[\sum_{k=0}^{n} a_k \cdot x^k\right] x^j dx = 0$$

Reordenando se obtienen las $n+1$ **Ecuaciones Normales** cuya expresión es:

$$\sum_{k=0}^{n} a_k \int_a^b \left[x^{k+j}\right] dx = \int_a^b \left[f_{(x)} \cdot x^j\right] dx \quad con \quad j = 0, \dots, n \tag{28}$$

Lo que en forma matricial supone:

$$\bar{\bar{H}} \cdot \bar{a} = \bar{r}$$

$$\begin{bmatrix} \int_a^b [x^{0+0}]\, dx & \int_a^b [x^{0+1}]\, dx & \cdots & \int_a^b [x^{0+n}]\, dx \\ \int_a^b [x^{1+0}]\, dx & \int_a^b [x^{1+1}]\, dx & \cdots & \int_a^b [x^{1+n}]\, dx \\ \vdots & \vdots & \ddots & \vdots \\ \int_a^b [x^{n+0}]\, dx & \int_a^b [x^{n+1}]\, dx & \cdots & \int_a^b [x^{n+n}]\, dx \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \int_a^b \left[f_{(x)}\right] dx \\ \int_a^b \left[f_{(x)} x\right] dx \\ \vdots \\ \int_a^b \left[f_{(x)} x^n\right] dx \end{bmatrix}$$

Donde $\bar{\bar{H}}$ representa a la **Matriz de Hilbert** (que es simétrica), $\bar{a}$ el vector de coeficientes $a_k\ con\ k = 0, \cdots, n$ a determinar y $\bar{r}$ es el vector de términos independientes r_{j+1}. Los elementos de dicha matriz se identifican como $H_{j+1,k+1}$ y los elementos del vector $\bar{r}$ dados por r_{j+1} corresponden a:

$$\int_a^b \left[x^{k+j}\right] dx = H_{j+1,k+1} \quad ; \quad \int_a^b \left[f_{(x)} * x^j\right] dx = r_{j+1}$$

Además, una forma sencilla de determinar los elementos de la Matriz de Hilbert es realizando la integral:

$$H_{j+1,k+1} = \int_a^b \left[x^{k+j}\right] dx = \left[\frac{x^{k+j+1}}{k+j+1}\right]_a^b = \frac{b^{k+j+1} - a^{k+j+1}}{k+j+1}$$

$$H_{j+1,k+1} = \frac{b^{j+k+1} - a^{j+k+1}}{j+k+1} \tag{29}$$

Donde $j+1$ indica la fila y $k+1$ representa la columna, con $j, k \in [0,1,2 \dots, n]$. Por lo tanto, la expresión matricial desarrollada se puede reescribir como:

$$\begin{bmatrix} \frac{b-a}{1} & \frac{b^2-a^2}{2} & \cdots & \frac{b^{n+1}-a^{n+1}}{n+1} \\ \frac{b^2-a^2}{2} & \frac{b^3-a^3}{3} & \cdots & \frac{b^{n+2}-a^{n+2}}{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{b^{n+1}-a^{n+1}}{n+1} & \frac{b^{n+2}-a^{n+2}}{n+2} & \cdots & \frac{b^{2n+1}-a^{2n+1}}{2n+1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \int_a^b [f_{(x)}]dx \\ \int_a^b [f_{(x)}x]dx \\ \vdots \\ \int_a^b [f_{(x)}x^n]dx \end{bmatrix}$$

Interpolación discreta (Guía 5)

Se aplica para el caso $m = n$, es decir, hay igual cantidad de datos que de funciones aproximantes.

Existencia de la solución

Se justifica la existencia de la solución por medio del **Teorema de Weierstrass** el cual establece:

$$Si\ F(x)\ es\ continua\ en\ el\ intervalo\ [a,b]\ entonces$$

$$la\ función\ alcanza\ sus\ máximos\ y\ mínimos\ absolutos\ en\ el\ intervalo\ mencionado$$

Entonces:

$$Si\ f \in C[a,b]\ y\ p(x) \in P_n[x]$$

$$Sea\ el\ error\ E_{n(f)} = \|f-p\|_\infty = \max_{x\in[a,b]} |f(x)-p(x)|$$

$$Entonces\ \lim_{n\to\infty} E_{n(f)} = 0$$

Propiedades:

- La función f puede ser aproximada con un error tan pequeño como se desee si se aumenta suficientemente el grado del polinomio. Esto es en el mundo ideal.
- Si f es suave → $E_{n(f)}$ decrece rápido (n pequeño).
- Cuando $E_{n(f)}$ decrece lento (n grande) el método resulta impráctico.

Representaciones alternativas del polinomio $p(x)$ de grado n

A) Expresión desarrollada:

$$p(x) = a_0 + a_1 * x + a_2 * x^2 + \cdots + a_n * x^n = \sum_{j=0}^{n} a_j * x^j\ ; \quad con\ a_n \neq 0$$

B) Esta expresión permite minimizar errores de redondeo si $c = \frac{a+b}{2}$:

$$p(x) = \sum_{j=0}^{n} b_j * (x-c)^j\ ; \quad con\ b_n \neq 0$$

C) Familia triangular:

$$\begin{cases} \varphi_0 = a_{00} \\ \varphi_1 = a_{10} + a_{11} * x \\ \varphi_2 = a_{20} + a_{21} * x + a_{22} * x^2 \\ \vdots \\ \varphi_n = a_{n0} + a_{n1} * x + a_{n2} * x^2 + \cdots + a_{nn} * x^n \end{cases}$$

En forma comprimida:

$$p(x) = \sum_{j=0}^{n} C_j * \varphi_j \; ; \quad con\ C_n \neq 0$$

Problema tipo

El problema tipo es determinar un polinomio de grado n que concuerde con los valores $f_{(x_i)}$ de la función f sobre una red de $n+1$ puntos x_i.

Teorema unicidad de la solución

El problema tiene solución única y es el **polinomio interpolador**, que resulta ser:

$$p(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \cdots + C_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

Teorema para el error de truncamiento

Sea $f \in C^{(n+1)}[a,b]$. Si $p(x) \in P_n[x]\ tal\ que\ p(x_i) = f(x_i)\ \ con\ i = 0, \cdots, n$ entonces el **error de truncamiento** viene dado por:

$$f(x) - p(x) = \frac{f_{(\gamma)}^{(n+1)}}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_{n-1})(x - x_n) \qquad (30)$$

Con $\gamma \in [a,b]$ desconocido. Es trivial que $f(x_i) - p(x_i) = 0\ \forall x_i\ con\ i = 0, \cdots, n$ ya que se fuerza que el polinomio interpolador pase por los puntos x_i dato.

A continuación, se desarrollan algunos métodos para la obtención de polinomios $p(x)$:

Método de Lagrange

Sea f una función definida en $x_0, \ldots, x_m$; con m+1 puntos distintos de la red, entonces $\exists\ p(x) \in P_n[x]$ con $n \leq m\ tal\ que\ p(x_i) = f(x_i)\ con\ i = 0, \ldots, m$. Siendo la fórmula para construir el **Polinomio de Lagrange** la siguiente:

$$p_L(x) = \sum_{k=0}^{n} L_{nk}(x) * f(x_k) \qquad (31)$$

Donde $L_{nk}(x)$ representa:

$$L_{nk}(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i} \rightarrow \ L_{nk}(x) = \begin{cases} 0\ si\ i \neq k \\ 1\ si\ i = k \end{cases}$$

Es decir, $L_{nk}(x)$ funciona como una delta de Kronecker.

Método de Newton

Partiendo de la expresión de la familia triangular, se plantea lo que se conoce como **diferencias divididas**, léase:

$$f_{[x_0,x_1,\cdots,x_{k-1},x_k]} = \frac{f_{[x_1,\cdots,x_{k-1},x_k]} - f_{[x_0,x_1,\cdots,x_{k-1}]}}{x_k - x_0} = C_k$$

Donde $k \leq m$ siendo $m+1$ la cantidad de puntos dato y siendo C_k los coeficientes a determinar de la expresión del **Polinomio de Newton** dada por:

$$p_N(x) = f_0 + \sum_{j=1}^{m} f_{[x_0,x_1,\cdots,x_j]}(x-x_0)(x-x_1)\cdots(x-x_{j-1})$$

$$p_N(x) = C_0 + \sum_{j=1}^{m} C_j(x-x_0)(x-x_1)\cdots(x-x_{j-1}) \tag{32}$$

Esquema de cálculo

$$\begin{array}{cc|ccc|cc|c} x_i & f_{(x_i)} & & & & & & \\ x_0 & \color{red}{f_{(x_0)}} & \color{red}{f_{[x_0,x_1]}} & & & & & \\ x_1 & f_{(x_1)} & f_{[x_1,x_2]} & \color{red}{f_{[x_0,x_1,x_2]}} & & & & \\ x_2 & f_{(x_2)} & f_{[x_2,x_3]} & f_{[x_1,x_2,x_3]} & \color{red}{f_{[x_0,x_1,x_2,x_3]}} & & & \\ x_3 & f_{(x_3)} & f_{[x_3,x_4]} & f_{[x_2,x_3,x_4]} & f_{[x_1,x_2,x_3,x_4]} & \color{red}{f_{[x_0,x_1,x_2,x_3,x_4]}} & & \\ x_4 & f_{(x_4)} & & & & & & \end{array}$$

Donde los términos marcados en rojo indican los coeficientes C_j, es decir:

$$C_0 = f_{(x_0)}\ ;\ C_1 = f_{[x_0,x_1]} = \frac{f_{(x_1)} - f_{(x_0)}}{x_1 - x_0}\ ;\ C_2 = f_{[x_0,x_1,x_2]} = \frac{f_{[x_1,x_2]} - f_{[x_0,x_1]}}{x_2 - x_0};$$

$$C_3 = f_{[x_0,x_1,x_2,x_3]} = \frac{f_{[x_1,x_2,x_3]} - f_{[x_0,x_1,x_2]}}{x_3 - x_0}\ ;\ C_4 = f_{[x_0,x_1,x_2,x_3,x_4]} = \frac{f_{[x_1,x_2,x_3,x_4]} - f_{[x_0,x_1,x_2,x_3]}}{x_4 - x_0}$$

Cabe aclarar que el polinomio interpolador obtenido por el método de Newton es SIEMPRE igual al obtenido con el método de Lagrange.

Método de Hermite

Mejora la aproximación de f cuando se conocen una o más de sus derivadas en los puntos de interpolación. Existen dos métodos para obtener el polinomio:

A) Método Directo (obtenido por medio de Lagrange)

Sea $f \in C^1[a,b]$ y $x_0,\cdots,x_m$ distintos pertenecientes al intervalo $[a,b]$. Entonces el polinomio de mínimo grado que coincide con f y f' en $x_0,\cdots,x_m$ tiene grado a lo sumo $2m+1$ y esta dado por el polinomio de Hermite, de fórmula:

$$H_{2m+1}(x) = \sum_{j=0}^{m} f(x_j) * H_{mj}(x) + \sum_{j=0}^{m} f'(x_j) * \widehat{H}_{mj}(x) \tag{33}$$

Donde:

- $H_{mj}(x) \equiv \left[1 - 2(x-x_j){L'}_{mj}(x_j)\right]{L^2}_{mj}(x)$

- $\widehat{H}_{mj}(x) \equiv (x - x_j)L^2{}_{mj}(x)$
- $L_{mj}(x)$: $Polinomio\ de\ Lagrange$
- $L'_{mj}(x)$: $Derivada\ del\ Polinomio\ de\ Lagrange$

B) Método generalizando las diferencias divididas de Newton

Resulta de agregar en el esquema de cálculo de las diferencias divididas de Newton a las derivadas de f evaluadas en los puntos $x_0, \cdots, x_m$ distintos pertenecientes al intervalo $[a, b]$. Por lo que, el esquema de cálculo actualizado quedaría:

$$
\begin{array}{cc|cccccc}
x_i & f_{(x_i)} & & & & & & \\
x_0 & f_{(x_0)} & f_{[x_0,x_0]} = f'_{(x_0)} & & & & & \\
x_0 & f_{(x_0)} & f_{[x_0,x_1]} & & & & & \\
x_1 & f_{(x_1)} & f_{[x_1,x_1]} = f'_{(x_1)} & & & & & \\
x_1 & f_{(x_1)} & f_{[x_1,x_2]} & & & & & \\
x_2 & f_{(x_2)} & f_{[x_2,x_2]} = f'_{(x_2)} & & & & & \\
x_2 & f_{(x_2)} & f_{[x_2,x_3]} & & & & & \\
x_3 & f_{(x_3)} & f_{[x_3,x_3]} = f'_{(x_3)} & & & & & \\
x_3 & f_{(x_3)} & f_{[x_3,x_4]} & & & & & \\
x_4 & f_{(x_4)} & f_{[x_4,x_4]} = f'_{(x_4)} & & & & & \\
x_4 & f_{(x_4)} & & & & & &
\end{array}
$$

$$
\begin{array}{cccccc}
f_{[x_0,x_0,x_1]} & f_{[x_0,x_0,x_1,x_1]} & f_{[x_0,x_0,x_1,x_1,x_2]} & f_{[x_0,x_0,x_1,x_1,x_2,x_2]} & & \\
f_{[x_0,x_1,x_1]} & f_{[x_0,x_1,x_1,x_2]} & f_{[x_0,x_1,x_1,x_2,x_2]} & f_{[x_0,x_1,x_1,x_2,x_2,x_3]} & & \\
f_{[x_1,x_1,x_2]} & f_{[x_1,x_1,x_2,x_2]} & f_{[x_1,x_1,x_2,x_2,x_3]} & f_{[x_1,x_1,x_2,x_2,x_3,x_3]} & & \\
f_{[x_1,x_2,x_2]} & f_{[x_1,x_2,x_2,x_3]} & f_{[x_1,x_2,x_2,x_3,x_3]} & f_{[x_1,x_2,x_2,x_3,x_3,x_4]} & & \\
f_{[x_2,x_2,x_3]} & f_{[x_2,x_2,x_3,x_3]} & f_{[x_2,x_2,x_3,x_3,x_4]} & f_{[x_2,x_2,x_3,x_3,x_4,x_4]} & & \\
f_{[x_2,x_3,x_3]} & f_{[x_2,x_3,x_3,x_4]} & f_{[x_2,x_3,x_3,x_4,x_4]} & & & \\
f_{[x_3,x_3,x_4]} & f_{[x_3,x_3,x_4,x_4]} & & & & \\
f_{[x_3,x_4,x_4]} & & & & &
\end{array}
$$

$$
\begin{array}{cccc}
f_{[x_0,x_0,x_1,x_1,x_2,x_2,x_3]} & f_{[x_0,x_0,x_1,x_1,x_2,x_2,x_3,x_3]} & f_{[x_0,x_0,x_1,x_1,x_2,x_2,x_3,x_3,x_4]} & f_{[x_0,x_0,x_1,x_1,x_2,x_2,x_3,x_3,x_4,x_4]} \\
f_{[x_0,x_1,x_1,x_2,x_2,x_3,x_3]} & f_{[x_0,x_1,x_1,x_2,x_2,x_3,x_3,x_4]} & f_{[x_0,x_1,x_1,x_2,x_2,x_3,x_3,x_4,x_4]} & \\
f_{[x_1,x_1,x_2,x_2,x_3,x_3,x_4]} & f_{[x_1,x_1,x_2,x_2,x_3,x_3,x_4,x_4]} & & \\
f_{[x_1,x_2,x_2,x_3,x_3,x_4,x_4]} & & &
\end{array}
$$

Donde los términos marcados en rojo indican los coeficientes C_j de la fórmula:

$$p_N(x) = C_0 + \sum_{j=1}^{m} C_j(x - x_0)(x - x_1)\cdots(x - x_{j-1})$$

Excepto $f_{(x_0)}$ el cual corresponde a C_0.

Método de Tchebychef

Este método surge para dar solución al **Fenómeno de Runge**. Dicho fenómeno aparece al realizar una aproximación polinomial sobre una red de puntos equidistantes. Es típico de funciones pares en intervalos de convergencia simétricos respecto del cero. Además, entre mayor sea el grado del polinomio interpolador peor resulta la convergencia en los extremos. En la Figura 2 se representa este fenómeno.

La solución de Tchebychef surge a partir de la expresión para el error de truncamiento (30):

$$f(x) - p(x) = \frac{f_{(\gamma)}{}^{(n+1)}}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_{n-1})(x - x_n)$$

Minimiza la función de dicho error seleccionando nuevos puntos $\tilde{x}_0, \cdots, \tilde{x}_m$ a partir de una transformación lineal y una fórmula que da origen a las abscisas de Tchebychef. Los nuevos puntos pertenecen al intervalo $[-1,1]$ y son simétricos respecto al cero. Además, la transformación redistribuye los puntos acumulando mayor cantidad en los extremos para así disminuir el error en esas zonas.

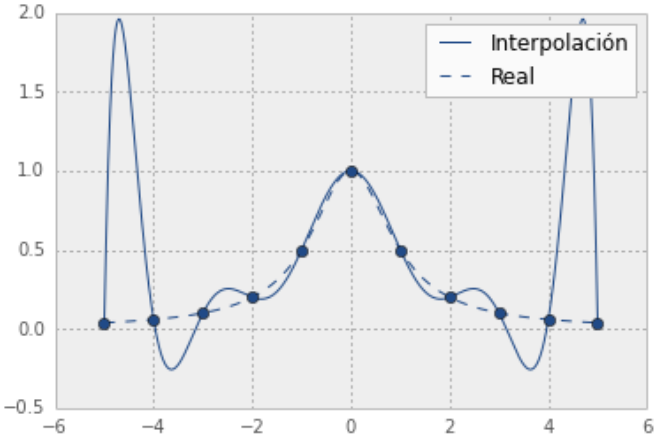


Figura 2 - Fenómeno de Runge (mala interpolación entre los datos).

Procedimiento

1) Por medio de la siguiente transformación lineal se convierten los valores $x_i \in [a,b]\ con\ i = 1, \dots, m$ a valores $t_i \in [-1,1]$. El $i = 0$ no se utiliza ya que $t_0 = t_1$ por lo que, de usarse, se estaría duplicando un dato.

$$t_i = \cos\left(\frac{2i-1}{n} \cdot \frac{\pi}{2}\right) \tag{34}$$

   Siendo n la cantidad de puntos dato, es decir: $n = m + 1$

2) Luego, a través de la expresión a continuación, se obtienen las abscisas de Tchebychef que serán los nuevos puntos $\tilde{x}_i$ dato, pertenecientes al intervalo $[a,b]$ para poder calcular el polinomio interpolador.

$$\tilde{x}_i = a + \frac{b-a}{2}(t_i + 1) \tag{35}$$

3) Se evalúan los nuevos puntos $\tilde{x}_i$ en la función $f_{(x)}$ y se obtienen las imágenes $f_{(\tilde{x}_i)}$.

4) Finalmente se emplea alguno de los métodos conocidos (Lagrange, Newton o Hermite) para generar un nuevo polinomio interpolante a partir de los $\tilde{x}_i$, el cual habrá aumentado su precisión respecto al enseñado en el gráfico anterior.

Cabe aclarar que el polinomio será de grado $m - 1$, ya que al contar con m datos $\tilde{x}_i$ el orden de este se reduce en uno, dado que un nodo se emplea para el término independiente.

Ajuste vs Interpolación

Comparando los procedimientos requeridos a la hora de realizar una aproximación por ajuste o realizarlo mediante un polinomio interpolador, existen algunas reglas prácticas recomendadas para tomar una decisión eficiente y efectiva:

- El problema se encontrará mejor condicionado si se aproximan los $m + 1$ puntos dato mediante ajuste por cuadrados mínimos con un polinomio $P_n[x]$ de grado $n < 2\sqrt{m}$.
- Si hay muchos datos NO es conveniente plantear interpolación, ya que el polinomio interpolador requeriría muchos cálculos y sería engorroso.
- Para pocos datos es conveniente aplicar una interpolación.

Sistemas de Ecuaciones Lineales (Guía 2)

El **problema tipo** a resolver es: Hallar la solución del Sistema de Ecuaciones Lineales (SEL) definido por las siguientes n ecuaciones lineales:

$$\begin{cases} E_1: a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ E_2: a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ E_n: a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Es decir, dados los a_{ij} y los b_i con $i, j = 1, \cdots, n$ se desea encontrar los x_i con $i = 1, \cdots, n$.

Métodos Directos

Método: Eliminación de Gauss

Se trata de la resolución de toda la vida para matrices. Las operaciones básicas con las que se cuentan son:

- $\gamma E_i \rightarrow E_i \; con \; \gamma \neq 0$ → Multiplicar una fila por una constante
- $E_i + \gamma E_j \rightarrow E_i$ → Sumar a una fila otra multiplicada por una constante
- $E_i \leftrightarrow E_j$ → Intercambiar dos filas de la matriz.

En forma matricial, se busca el vector incógnita $\bar{x} \in R^{nx1}$ tal que $\bar{\bar{A}}\,\bar{x} = \bar{b}$, siendo:

$$\bar{\bar{A}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in R^{nxn} \quad ; \quad \bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{nx1} \quad ; \quad \bar{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in R^{nx1}$$

Se utiliza la notación de la **matriz aumentada:**

$$[\bar{\bar{A}}, \bar{b}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{bmatrix} \in R^{nx(n+1)} \quad con \quad \begin{bmatrix} a_{1,n+1} \\ a_{2,n+1} \\ \vdots \\ a_{n,n+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad (36)$$

Donde los elementos $a_{i,n+1} \; con \; i = 1, \cdots, n$ corresponden a los elementos del vector de términos independientes b_i.

A continuación, se procede a la triangulación de la matriz por medio de los métodos de eliminación de Gauss antes expuestos. Para ello se aplica el siguiente algoritmo:

- Siempre que $a_{ii} \neq 0$, se aplica la transformación lineal:

$$E_j - \frac{a_{ji}}{a_{ii}} E_i \rightarrow E_j \quad con \quad \begin{cases} j = k, k+1, \cdots, n \\ k = i+1 \end{cases}$$

Es decir, se eliminan los x_i de las nuevas ecuaciones $E_j^{(k)}$. De esta manera se logra obtener ceros debajo de la diagonal en las primeras $k-1$ columnas. Luego de k pasos, el SEL queda definido como:

$$a_{ij}{}^{(k)} \begin{cases} a_{ij}{}^{(k-1)} & con \quad \begin{cases} i = 1, \cdots, k-1 \\ j = 1, \cdots, n+1 \end{cases} \\ 0 & con \quad \begin{cases} i = k, \cdots, n \\ j = 1, \cdots, k-1 \end{cases} \\ a_{ij}{}^{(k-1)} - \dfrac{a_{i,k-1}{}^{(k-1)}}{a_{k-1,k-1}{}^{(k-1)}} a_{k-1,j}{}^{(k-1)} & con \quad \begin{cases} i = k, \cdots, n \\ j = k, \cdots, n+1 \end{cases} \end{cases}$$

Donde se acaba de eliminar la variable x_{k-1} de las nuevas ecuaciones $E_k{}^{(k)}, \cdots, E_n{}^{(k)}$.

Observación: si el **pivote** $a_{kk}{}^{(k)} = 0$ el procedimiento falla. Para salvarlo se aplica un intercambio de filas $E_k \leftrightarrow E_j$ para algún $j > k \; tal \; que \; a_{jk}{}^{(k)} = 0$.

Recordatorio: el cambio de filas NO modifica el orden de las incógnitas x_i.

Se obtiene entonces una matriz de la forma:

$$[\bar{\bar{A}}, \bar{b}]^{(n)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22}{}^{(2)} & \cdots & a_{2n}{}^{(2)} & a_{2,n+1}{}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn}{}^{(n)} & a_{n,n+1}{}^{(n)} \end{bmatrix}$$

Donde los valores que aparecen entre paréntesis indican el número de iteración en el cual se consiguió la ecuación $E^{(k)}$. Luego, para obtener las incógnitas del problema se realiza una **sustitución hacia atrás.** Donde:

- Paso 1: $x_n = \frac{a_{n,n+1}{}^{(n)}}{a_{nn}{}^{(n)}}$
- Paso 2: $x_{n-1} = \frac{a_{n-1,n+1}{}^{(n-1)} - a_{n-1,n}{}^{(n-1)} * x_n}{a_{n-1,n-1}{}^{(n-1)}}$
- Paso i:

$$x_i = \frac{a_{i,n+1}{}^{(i)}}{a_{ii}{}^{(i)}} - \frac{1}{a_{ii}{}^{(i)}} \sum_{j=i+1}^{n} a_{ij}{}^{(i)} * x_j \; con \; i = n-1, n-2, \cdots, 1$$

Norma de una matriz

Existen distintos tipos de normas matriciales, sin embargo, las que utilizaremos nosotros serán la **norma infinito** y la **norma uno** cuyas expresiones son:

$$\left\|\bar{\bar{A}}\right\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| ; \; \left\|\bar{\bar{A}}\right\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}| \quad siendo \; \bar{\bar{A}} \in R^{nxn}$$

Es decir, la **norma infinito** está dada por aquella fila cuya sumatoria del valor absoluto de sus elementos es máxima. Mientras que, la **norma uno** está dada por aquella columna cuya sumatoria del valor absoluto de sus elementos es máxima.

Para la resolución de ejercicios, se puede optar por cualquiera de las dos normas matriciales, pero es necesario mantener la consistencia.

Propagación de errores

Para la **propagación de los errores de entrada** se cuenta con la siguiente expresión:

$$\frac{\left\|\overline{\delta x}\right\|}{\|\bar{x}\|} \le \left\|\bar{\bar{A}}\right\| \cdot \left\|\bar{\bar{A}}^{-1}\right\| \cdot \left(\frac{\left\|\overline{\delta b}\right\|}{\|\bar{b}\|} + \frac{\left\|\overline{\overline{\delta A}}\right\|}{\left\|\bar{\bar{A}}\right\|} \right) \qquad (37)$$

Donde $\overline{\delta b}$ y $\overline{\overline{\delta A}}$ son los errores de entrada dados en forma de vector y matriz, respectivamente, es decir cada elemento posee su propio error de entrada. Además, $K_{(A)} \equiv \left\|\bar{\bar{A}}\right\| \cdot \left\|\bar{\bar{A}}^{-1}\right\|$ se denomina **número de condición de A.**

- Si $K_{(A)}$ es pequeño, léase aproximado a orden 1 → $\bar{\bar{A}}$ se encuentra bien condicionada.
- Si $K_{(A)}$ es grande, léase mucho mayor a orden 1 → $\bar{\bar{A}}$ se encuentra mal condicionada.

Observaciones:

- Si $\overline{\overline{\delta A}} = \bar{\bar{0}}$ → $C_P \leq K_{(A)}$
- Si $\overline{\delta b}$ y $\overline{\overline{\delta A}}$ son producto solo de errores de redondeo → $\left\|\overline{\delta x}\right\| = 2 \cdot u \cdot K_{(A)} \cdot \|\bar{x}\|$

Respecto a la **propagación de los errores de redondeo,** al resolver se obtiene una solución $\tilde{\boldsymbol{x}}$ que NO satisface el sistema original (debido al redondeo durante el proceso), en consecuencia, nos aparece un **residuo** $\bar{\boldsymbol{R}}$ que se puede calcular como:

$$\bar{R} = \bar{\bar{A}}\bar{x} - \bar{\bar{A}}\tilde{x} = \bar{b} - \tilde{b} \qquad (38)$$
$$\bar{x} - \tilde{x} = \bar{\bar{A}}^{-1}\bar{R}$$

Entonces, acotando y usando que $\left\|\bar{b}\right\| \leq \left\|\bar{\bar{A}}\right\| \cdot \|\bar{x}\|$:

$$\frac{\|\bar{x} - \tilde{x}\|}{\|\bar{x}\|} \leq K_{(A)} \frac{\|\bar{R}\|}{\left\|\bar{b}\right\|} \qquad (39)$$

Observaciones:

- Para obtener $\tilde{b}$ se utiliza doble precisión, para minimizar el error de redondeo.
- Además, se realizan muchas menos operaciones para calcular $\bar{\bar{A}}\tilde{x}$ que para calcular $\bar{\bar{A}}$.

Por estas dos cuestiones se considera que el cálculo del residuo es acotado y admisible.

Costo computacional

- Para multiplicaciones y divisiones → $\frac{1}{3}n^3 + n^2 - \frac{1}{3}n$
- Para sumas y restas → $\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$

Eligiendo un n se puede obtener la cantidad de operaciones necesarias para resolver el problema. Claramente no es un método para aplicar a mano incluso para órdenes bajos.

Estrategias de pivoteo para reducir la propagación de errores de redondeo

Se denomina **pivote** a los elementos a_{kk} de la matriz $\bar{\bar{A}}$. Si el pivote de la columna que se está triangulando es pequeño comparado con los demás elementos de la columna, es decir $\left|{a_{kk}}^{(k)}\right| \ll \left|{a_{ik}}^{(k)}\right|$, entonces el **multiplicador asociado** $\boldsymbol{m_{ik}}$ puede incrementar los errores.

$$m_{ik} = \frac{{a_{ik}}^{(k)}}{{a_{kk}}^{(k)}} \xrightarrow[si\ |{a_{kk}}^{(k)}| \ll |{a_{ik}}^{(k)}|]{} |m_{ik}| \gg 1 \qquad (40)$$

Para reducir este efecto se aplican técnicas de pivoteo para encontrar el mayor pivote disponible y generar así los menores multiplicadores para cada fila.

- **Pivoteo parcial:** se selecciona el menor entero r de la columna k y se permuta la fila de r con la fila del m_{ik}: $E_r \leftrightarrow E_j$.

$$\left|{a_{rk}}^{(k)}\right| = \max\left|{a_{jk}}^{(k)}\right| \; con\; k \leq j \leq n$$

  Deben realizarse conjuntamente las permutaciones de filas sobre el vector de términos independientes $\bar{b}$: $b_r \leftrightarrow b_j$.
- **Pivoteo total o completo:** además de aplicar pivoteo parcial se agregan permutaciones de columnas. Se selecciona el "menor par" de enteros r y s para los cuales:

$$\left|{a_{rs}}^{(k)}\right| = \max\left|{a_{ji}}^{(k)}\right| \; con\; k \leq j \; ; \; i \leq n$$

  Luego, se intercambian las filas r y j y las columnas s e i:

$$E_r \leftrightarrow E_j \; ; \; C_s \leftrightarrow C_i$$

  Deben realizarse conjuntamente las permutaciones de filas sobre el vector de términos independientes $\bar{b}$: $b_r \leftrightarrow b_j$, y además, las permutaciones de columnas requieren el intercambio de filas en el vector de incógnitas $\bar{x}$: $x_s \leftrightarrow x_i$.

Estas técnicas traen consigo un costo computacional extra, sin embargo, es justificable para casos donde la exactitud es indispensable.

Método: Descomposición LU

Este método propone descomponer la matriz $\bar{\bar{A}}$ como el producto entre dos matrices de la siguiente forma:

$$\bar{\bar{A}} = \bar{\bar{L}} \cdot \bar{\bar{U}} \tag{41}$$

Donde $\bar{\bar{L}}$ es triangular inferior (Lower) y $\bar{\bar{U}}$ es triangular superior (Upper). Resulta útil para casos donde tengo distintos posibles escenarios, es decir, varios posibles $\bar{b}$. Además, dichas matrices quedan definidas al realizar el proceso de eliminación de Gauss de la siguiente manera:

$$\bar{\bar{L}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21}^{(1)} & 1 & 0 & 0 \\ m_{31}^{(1)} & m_{32}^{(2)} & 1 & 0 \\ m_{41}^{(1)} & m_{42}^{(2)} & m_{43}^{(3)} & 1 \end{bmatrix} \; ; \; \bar{\bar{U}} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} \\ 0 & {a_{22}}^{(2)} & {a_{23}}^{(2)} & {a_{24}}^{(2)} \\ 0 & 0 & {a_{33}}^{(3)} & {a_{34}}^{(3)} \\ 0 & 0 & 0 & {a_{44}}^{(4)} \end{bmatrix} \tag{42}$$

Donde $m_{ik} = \frac{{a_{ik}}^{(k)}}{{a_{kk}}^{(k)}}$. De esta manera el problema original queda expresado como:

$$\bar{\bar{A}}\,\bar{x} = \bar{b} \rightarrow \bar{\bar{L}}\,\bar{\bar{U}}\,\bar{x} = \bar{b}$$

Realizando el cambio de variables:

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

Resulta:

$$\bar{\bar{L}}\,\bar{y} = \bar{b}$$

Donde $\bar{y}$ es el vector incógnita transitorio. En consecuencia, el problema se soluciona mediante la resolución de dos SEL. Por la naturaleza de $\bar{\bar{L}}$ se obtiene $\bar{y}$ por sustitución directa y luego $\bar{\bar{U}}$ por sustitución inversa.

Costo computacional

- Para sustitución directa más inversa → n^2
- Para triangulación de Gauss → $\frac{2}{3}n^3$

Obtención de L y U

Dada $\bar{\bar{A}} \in R^{nxn}$ no singular (determinante $\neq 0$), existen:

$$\bar{\bar{P}}\,\bar{\bar{A}} = \bar{\bar{L}}\,\bar{\bar{U}}$$

- Matriz $\bar{\bar{P}}$ de permutaciones, indica los intercambios de filas (pivoteos) que se le hacen a $\bar{\bar{A}}$ para encontrar la descomposición LU.
- Matriz $\bar{\bar{U}}$ triangular superior, producto de la eliminación de Gauss.
- Matriz $\bar{\bar{L}}$ triangular inferior con unos en la diagonal, construida con los multiplicadores m_{ik} obtenidos durante el proceso de eliminación. Donde:

$$l_{ik} = \begin{cases} m_{ik} = \dfrac{{a_{ik}}^{(k)}}{{a_{kk}}^{(k)}} & si\ i > k \\ 1 & si\ i = k \\ 0 & si\ i < k \end{cases}$$

Existen tres formas de armar las matrices para la descomposición LU:

1) Método de Doolittle → $l_{kk} = 1$
2) Método de Croot → $u_{kk} = 1$
3) Método de Chlesky → $l_{kk} = u_{kk}$

Nosotros utilizamos el Método de Doolittle porque guardamos los multiplicadores durante el proceso.

Matrices especiales

Matriz inversa

Si se conoce $\bar{\bar{A}}^{-1}$ el sistema $\bar{\bar{A}}\bar{x} = \bar{b}$ se resuelve fácil:

$$\bar{x} = \bar{\bar{A}}^{-1}\bar{b}$$

Como $\bar{\bar{A}}^{-1}\,\bar{\bar{A}} = \bar{\bar{I}}$, cada columna j de $\bar{\bar{A}}^{-1}$ se obtiene de resolver un SEL de la forma:

$$\begin{cases} {\bar{\bar{A}}_x}^{(j)} = \bar{\bar{I}}^{(j)}, & j = 1, \cdots, n \\ {x_i}^{(j)} = {a_{ij}}^{-1}, & i = 1, \cdots, n \\ {I_i}^{(j)} = I_{ij}, & i = 1, \cdots, n \end{cases}$$

Matriz diagonal dominante

Es aquella matriz $\bar{\bar{A}}$ tal que los elementos de la diagonal en cada fila son mayores en módulo a la sumatoria de los valores absolutos de los demás elementos que componen a la fila:

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

Esta clase de matriz NO requiere pivoteo alguno, siendo **estable respecto a la propagación de errores de redondeo.**

Matriz simétrica

Si $\bar{\bar{A}}$ es simétrica $a_{ij} = a_{ji}$, la eliminación de Gauss también produce una matriz simétrica entonces **se reduce la cantidad de operaciones** prácticamente a la mitad.

Matriz definida positiva

Es aquella tal que: $\bar{x}^T \, \bar{\bar{A}} \, \bar{x} > 0$. Esta clase de matriz NO requiere pivoteo alguno, siendo **estable respecto a la propagación de errores de redondeo.**

Matriz banda

Son matrices ralas o dispersas (matriz de gran tamaño en la que la mayor parte de sus elementos son nulos) donde los elementos no nulos se localizan en una banda centrada a lo largo de la diagonal principal y algunas diagonales adyacentes.

$$a_{ij} = 0, \qquad si\ j > i + p, \qquad o, \qquad i > j + q$$

El ancho de banda esta dado por: $w = p + q + 1$

Si no se aplica pivoteo, L y U serán matrices banda con:

$$\begin{cases} l_{ij} = m_{ij} = 0, & si\ j > i\ o\ i > j + q \\ u_{ij} = 0, & si\ j < i\ o\ j > i + p \end{cases}$$

Cuando p y q son mucho menores que n el costo computacional y cantidad de operaciones se reducen sensiblemente.

Además, si $\bar{\bar{A}}$ es **tridiagonal** los elementos NO nulos se encuentran únicamente en la diagonal principal y en las diagonales adyacentes por encima y por debajo de esta, es decir, $p = q = 1$ y solo se requieren $3(n-1)$ sumas y multiplicaciones, y $2n - 1$ divisiones. Un ejemplo de la matriz es:

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

Refinamiento iterativo - Minimización del residuo

Considerando el residuo definido en (38), si $\tilde{x}^{(1)}$ es la solución aproximada, se tiene que:

$$\bar{R}^{(1)} = \bar{b} - \tilde{b} = \bar{b} - \bar{\bar{A}}\tilde{x}^{(1)}$$

Recordando que $\tilde{x}^{(1)}$ no es exactamente igual a la solución teórica $\bar{b}$ sino solo una aproximación, debido a la propagación de errores de redondeo durante los cálculos, la expresión anterior arrojará que el residuo es no nulo. A raíz de ello, el **método de refinamiento iterativo** proponer resolver un SEL definido por la misma matriz $\bar{\bar{A}}$ pero ahora el término independiente es reemplazado por el residuo $\bar{R}^{(1)}$, es decir el siguiente sistema:

$$\bar{\bar{A}}\,\overline{\delta x}^{(1)} = \bar{R}^{(1)} \tag{43}$$

Donde $\overline{\delta x}^{(1)}$ es el vector que se denomina **corrección,** pues permite mejorar el resultado $\bar{x}$ obtenido por el método de eliminación de Gauss. Desarrollando (43):

$$\bar{R}^{(1)} = \bar{\bar{A}}\left(\bar{x} - \tilde{x}^{(1)}\right) = \bar{\bar{A}} * \overline{\delta x}^{(1)}$$

$$\bar{b} - \bar{\bar{A}}\tilde{x}^{(1)} = \bar{\bar{A}} * \overline{\delta x}^{(1)}$$

$$\bar{\bar{A}}\left(\tilde{x}^{(1)} + \overline{\delta x}^{(1)}\right) = \bar{b} \tag{44}$$

De aquí se observa que resolver el planteo de (43) nos permite mejorar el resultado de la resolución común por medio de la adición de $\overline{\delta x}^{(1)}$ a la aproximación $\tilde{x}^{(1)}$, según indica (44).

Para obtener $\bar{R}^{(1)}$ se utiliza doble precisión, se define:

$$\tilde{x}^{(2)} = \tilde{x}^{(1)} + \overline{\delta x}^{(1)} \rightarrow \bar{\bar{A}}\,\tilde{x}^{(2)} = \bar{b}$$

El proceso se repite hasta alcanzar la precisión deseada y converge bajo ciertas hipótesis que no vemos. El proceso iterativo se esquematiza en las siguientes ecuaciones:

$$\bar{R}^{(k)} = \bar{b} - \bar{\bar{A}}\tilde{x}^{(k)} \tag{45}$$

$$\bar{\bar{A}}\,\overline{\delta x}^{(k)} = \bar{R}^{(k)} \tag{46}$$

$$\tilde{x}^{(k+1)} = \tilde{x}^{(k)} + \overline{\delta x}^{(k)} \tag{47}$$

Heurística:

En cada paso del refinamiento iterativo, trabajando con t dígitos para la mantisa, se mejoran q dígitos de precisión donde:

$$q = t - p \tag{48}$$

$$p = \log\left(K_{(A)}\right) \tag{49}$$

$$K_{(A)} \approx \frac{\left\|\overline{\delta x}^{(1)}\right\|}{\left\|\bar{x}^{(1)}\right\|} * 10^{t} \tag{50}$$

El valor de p se define al inicio y queda fijo para el resto del problema. La expresión (48) nos da una idea de cuantos dígitos vamos a ganar por cada paso iterativo realizado, por lo tanto, observando el resultado de esta fórmula podremos saber si tiene sentido o no utilizar el procedimiento de refinamiento iterativo y cuantas veces lo deberemos aplicar para una deseada precisión en el resultado final. La expresión (50) es un

atajo para calcular $K_{(A)}$ y está planteada con norma infinito y solo se debe plantear para la primera aproximación del refinamiento.

Métodos iterativos

Para resolver el problema $\bar{\bar{A}}\bar{x} = \bar{b}$ vamos a proponer una forma iterativa. Recordando el método de punto fijo $x^{(k)} = g\left(x^{(k-1)}\right)$, planteamos:

$$\bar{x}^{(k)} = \bar{\bar{T}} \cdot \bar{x}^{(k-1)} + \bar{c} \tag{51}$$

Donde $\bar{\bar{T}}\ y\ \bar{c}$ son fijos durante el proceso. Se utiliza el criterio de convergencia: $\lim_{k\to\infty} \bar{x}^{(k)} = \bar{x}$. Se propondrán tres alternativas diferentes para la matriz $\bar{\bar{T}}$ y el vector $\bar{c}$, de manera tal que obtendremos tres métodos de resolución distintos. Como estos métodos se programan tienen implícito el error de truncamiento, problema que no existía en los métodos directos.

Método de Jacobi

Despejando la incógnita x_i de la ecuación E_i del sistema, siempre que $a_{ii} \neq 0$, se tiene que:

$$x_i = -\frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j + \frac{b_i}{a_{ii}}, con\ i = 1, \cdots, n \tag{52}$$

En caso de que algún elemento $a_{ii} = 0$ se aplica un pivoteo. De la ecuación (52) se identifican: $(\delta_{ij}: delta\ de\ Kronecker)$

$$T_{ij} = -\frac{a_{ij}}{a_{ii}}\left(1 - \delta_{ij}\right) \quad c_i = \frac{b_i}{a_{ii}} \quad con\ i, j = 1, \cdots, n \tag{53}$$

Entonces, el **método de Jacobi** resulta:

$$x_i{}^{(k)} = -\frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j{}^{(k-1)} + \frac{b_i}{a_{ii}} \quad con \quad \begin{cases} i = 1, \cdots, n \\ k \geq 1 \end{cases} \tag{54}$$

Se observa que el método requiere de una semilla para $k = 1$. Ahora bien, para poder programarlo se necesitan la matriz $\bar{\bar{T}}$ y el vector $\bar{c}$ según (51), de coeficientes dados por (53).

Obtención de $\bar{\bar{T}}$ y $\bar{c}$ en forma matricial

Sin embargo, parecería ser útil poder obtener $\bar{\bar{T}}\ y\ \bar{c}$ partiendo de $\bar{\bar{A}}$, por ello se propone expresar $\bar{\bar{A}}$ como:

$$\bar{\bar{A}} = \bar{\bar{D}} - \bar{\bar{L}} - \bar{\bar{U}} \tag{55}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & 0 & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Donde $\bar{\bar{D}}$ es una matriz diagonal, $\bar{\bar{L}}$ es triangular inferior con 0 en la diagonal y $\bar{\bar{U}}$ es triangular superior con 0 en la diagonal (no tienen nada que ver con la descomposición LU). Despejando:

$$\bar{\bar{A}}\bar{x} = \bar{b} \rightarrow \left[\bar{\bar{D}} - \bar{\bar{L}} - \bar{\bar{U}}\right]\bar{x} = \bar{b} \rightarrow \bar{\bar{D}}\,\bar{x} - \left[\bar{\bar{L}} + \bar{\bar{U}}\right]\bar{x} = \bar{b} \rightarrow \bar{\bar{D}}\,\bar{x} = \left[\bar{\bar{L}} + \bar{\bar{U}}\right]\bar{x} + \bar{b}$$

$$\bar{x}^{(k)} = \bar{\bar{D}}^{-1}\left[\bar{\bar{L}} + \bar{\bar{U}}\right] \bar{x}^{(k-1)} + \bar{\bar{D}}^{-1}\bar{b}$$

Por ende, se deduce que:

$$\bar{\bar{T}}_J = \bar{\bar{D}}^{-1}\left[\bar{\bar{L}} + \bar{\bar{U}}\right] \quad ; \quad \bar{c}_J = \bar{\bar{D}}^{-1}\bar{b} \tag{56}$$

El subíndice J indica que la matriz y el vector corresponden al método de Jacobi.

Criterio de corte

$$\frac{\left\|\bar{x}^{(k)} - \bar{x}^{(k-1)}\right\|}{\left\|\bar{x}^{(k)}\right\|} < Tolerancia \tag{57}$$

Condiciones de convergencia

Para lograr la convergencia para toda semilla $\bar{x}^{(0)}$ es **condición necesaria y suficiente** que:

$$\rho\left(\bar{\bar{T}}_J\right) = \max_{1 \le i \le n} |\mu_i| < 1 \tag{58}$$

Donde $\rho\left(\bar{\bar{T}}_J\right)$ representa al **radio espectral** de la matriz $\bar{\bar{T}}_J$ y μ_i representa a los autovalores de dicha matriz. Es decir que los autovalores de la matriz de Jacobi deben ser menor que 1 en módulo para que el método sea convergente. Además, el cumplimiento de la condición anterior también se puede expresar como:

$$Si \left\|\bar{\bar{T}}_J\right\| < 1 \; el\ método\ converge\ \forall \bar{x}^{(0)} \tag{59}$$

Como consecuencia de ello se tiene la siguiente **propiedad:** si $\bar{\bar{A}}$ es estrictamente diagonal dominante, el método converge $\forall \bar{x}^{(0)}$.

Error de truncamiento

Si se cumple (59), la **cota para el error de truncamiento** viene dada por cualquiera de las siguientes expresiones:

$$\left\|\bar{x} - \bar{x}^{(k)}\right\| \begin{cases} \le \dfrac{\left\|\bar{\bar{T}}_J\right\|^k}{1 - \left\|\bar{\bar{T}}_J\right\|} \left\|\bar{x}^{(1)} - \bar{x}^{(0)}\right\| \\ \le \dfrac{\left\|\bar{\bar{T}}_J\right\|}{1 - \left\|\bar{\bar{T}}_J\right\|} \left\|\bar{x}^{(k)} - \bar{x}^{(k-1)}\right\| \end{cases} \tag{60}$$

Si $\left\|\bar{\bar{T}}_J\right\| < \frac{1}{2}$, la cota para dicho error puede tomarse directamente como $\left\|\bar{x}^{(k)} - \bar{x}^{(k-1)}\right\|$. Se utiliza la norma infinito.

Error de redondeo

Siempre que el error de redondeo este razonablemente acotado en cada paso iterativo, se puede asumir que el hecho de pasar de una iteración a la siguiente absorbe el error cometido y se puede suponer que esta presente en la diferencia entre la nueva aproximación y el valor teórico de la solución buscada, es decir, está contenida en el error de truncamiento:

$$\left\|\bar{x} - \bar{x}^{(k)}\right\|$$

Por lo que NO tiene impacto significativo para valores de t (precisión) razonables.

Propagación de errores de entrada

Para la forma del método general $\bar{x}^{(k)} = \bar{\bar{T}}\,\bar{x}^{(k-1)} + \bar{c}$ se tiene que los errores se van a propagar manteniendo la cota de la expresión (61):

$$\left\|\hat{x}^{(k)} - \bar{x}\right\| \leq \frac{\left\|\overline{\overline{\delta T}}\right\|}{1 - \left\|\bar{\bar{T}}\right\|}\left\|\bar{x}^{(k-1)}\right\| + \frac{\left\|\overline{\delta c}\right\|}{1 - \left\|\bar{\bar{T}}\right\|} \qquad (61)$$

Donde $\hat{x}^{(k)}$ es la mejor aproximación obtenida, es decir, la originada al finalizar el proceso en la iteración k. $\overline{\overline{\delta T}}$ y $\overline{\delta c}$ representan a los errores absolutos de entrada transformados.

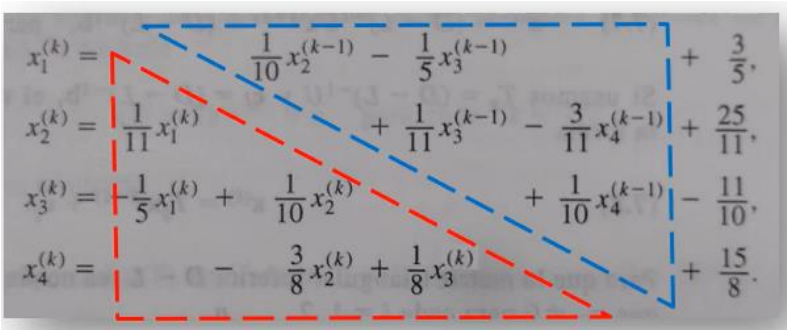
Método de Gauss-Seidel

Permite acelerar la convergencia respecto a Jacobi y lo logra a partir de la expresión:

$$x_i{}^{(k)} = -\frac{1}{a_{ii}}\sum_{j=1}^{i-1} a_{ij}x_j{}^{(k)} - \frac{1}{a_{ii}}\sum_{j=i+1}^{n} a_{ij}x_j{}^{(k-1)} + \frac{b_i}{a_{ii}} \quad con \quad \begin{cases} i = 1, \cdots, n \\ k \geq 1 \end{cases} \qquad (62)$$

Donde la sumatoria del primer término representa a las $i-1$ primeras incógnitas ya actualizadas en el paso iterativo k, mientras que la sumatoria del segundo término son las $n-i-1$ incógnitas sin actualizar del paso iterativo $k-1$. Es decir, **el planteo es idéntico al de Jacobi con la diferencia de que se van utilizando en el mismo paso iterativo los valores de las incógnitas x_i que se fueron obteniendo.**

Si se utiliza la descomposición propuesta en (55), la matriz $\bar{\bar{L}}$ se actualiza en el paso k mientras que para la matriz $\bar{\bar{U}}$ se utiliza la que quedó en el paso $k-1$, ver superíndices en la figura.



Condiciones de convergencia

- Son las mismas que en Jacobi.
- Generalmente, Gauss-Seidel acelera la convergencia.
- La convergencia de uno de los métodos NO asegura la del otro.

Error de truncamiento

Idem Jacobi pero con $\overline{\overline{T_{GS}}}$, son válidas las expresiones (60).

Error de redondeo

Idem Jacobi.

Propagación de errores de entrada

Como el problema matemático es el mismo, sigue valiendo (61).

Método de SOR (Succesive Over-Relaxation)

Logra acelerar aún más la convergencia por medio de sobre relajaciones sucesivas aplicadas a través de un coeficiente ω. Partiendo de la expresión para Gauss-Seidel (62) se tiene que:

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

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

Sumando y restando $a_{ii}x_i^{(k-1)}$ en el miembro derecho:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left\{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^{n} a_{ij}x_j^{(k-1)} + a_{ii}x_i^{(k-1)}\right\} \tag{63}$$

Recordando la expresión del **residuo $\bar{R}^{(k)}$** dada por (45):

$$\bar{R}^{(k)} = \bar{b} - \bar{\bar{A}}\tilde{x}^{(k-1)}$$

Se puede identificar de la expresión (63), que los primeros tres términos de las llaves dan lugar a un residuo:

$$b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^{n} a_{ij}x_j^{(k-1)} = \bar{b} - \bar{\bar{A}}\tilde{x}^{(k-1)} = r^{(k)}$$

El residuo $r_i^{(k)}$ cambia según el método y corresponde a:

$$r_i^{(k)} = \begin{cases} b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i}^{n} a_{ij}x_j^{(k-1)} & (Gauss-Seidel) \\ b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k-1)} & (Jacobi) \end{cases} \tag{64}$$

Por lo que, la ecuación (63) se puede reescribir como:

$$x_i^{(k)} = \frac{1}{a_{ii}}\left\{r_i^{(k)} + a_{ii}x_i^{(k-1)}\right\} = x_i^{(k-1)} + \frac{r_i^{(k)}}{a_{ii}} \tag{65}$$

Donde se observa que lo que se está haciendo es sumarle al resultado de la iteración anterior un término de corrección que está asociado al residuo.

El **método SOR** proponer sumarle al resultado de la iteración anterior la corrección amplificada por un parámetro $\boldsymbol{\omega}$ denominado **factor de peso o parámetro de relajación,** es decir**:**

$$x_i^{(k)} = x_i^{(k-1)} + \boldsymbol{\omega}\frac{r_i^{(k)}}{a_{ii}} \tag{66}$$

Trabajamos la expresión para llevarla a una forma más operativa. Sumando y restando $\omega x_i^{(k-1)}$ y expresando $r_i^{(k)}$ para Gauss-Seidel:

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

$$x_i{}^{(k)} = (1-\omega) x_i{}^{(k-1)} + \frac{\omega}{a_{ii}}\left\{ b_i - \sum_{j=1}^{i-1} a_{ij} x_j{}^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j{}^{(k-1)} \right\} \tag{67}$$

La expresión (67) corresponde al método SOR para $i = 1, \dots, n$ y con $k \geq 1$.

La programación del método resulta directa a partir del Método de Gauss-Seidel, lo cual se puede expresar como:

$$x_i{}^{(k)} = (1-\omega) x_i{}^{(k-1)} + \omega \left[x_i{}^{(k)} \right]_{GS} \tag{68}$$

Para llevarlo a una forma matricial se multiplica la expresión (67) por el elemento de la diagonal y se agrupan los términos a ambos lados según el paso de iteración:

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

Lo que en forma matricial resulta:

$$\bar{\bar{D}} \bar{x}^{(k)} - \omega \bar{\bar{L}} \bar{x}^{(k)} = (1-\omega) \bar{\bar{D}} \bar{x}^{(k-1)} + \omega \bar{\bar{U}} \bar{x}^{(k-1)} + \omega \bar{b}$$

Los signos para las matrices $\bar{\bar{L}}$ y $\bar{\bar{U}}$ se invierten debido a su forma, según lo expuesto en (55):

$$\left[\bar{\bar{D}} - \omega \bar{\bar{L}} \right] \bar{x}^{(k)} = \left[(1-\omega) \bar{\bar{D}} + \omega \bar{\bar{U}} \right] \bar{x}^{(k-1)} + \omega \bar{b}$$

$$\bar{x}^{(k)} = \left[\bar{\bar{D}} - \omega \bar{\bar{L}} \right]^{-1} \left[(1-\omega) \bar{\bar{D}} + \omega \bar{\bar{U}} \right] \bar{x}^{(k-1)} + \omega \left[\bar{\bar{D}} - \omega \bar{\bar{L}} \right]^{-1} \bar{b}$$

En consecuencia, se tiene que:

$$\begin{cases} \bar{\bar{T}}_\omega = \left[\bar{\bar{D}} - \omega \bar{\bar{L}} \right]^{-1} \left[(1-\omega) \bar{\bar{D}} + \omega \bar{\bar{U}} \right] \\ \bar{c}_\omega = \omega \left[\bar{\bar{D}} - \omega \bar{\bar{L}} \right]^{-1} \end{cases} \tag{69}$$

Condiciones de convergencia

Se puede probar que:

$$Si\ a_{ii} \neq 0\ \forall i \rightarrow \rho\left(\bar{\bar{T}}_\omega \right) \geq |\omega - 1|$$

Ahora bien, para que el método sea convergente, según lo expuesto para el método general, el radio espectral debe ser $\rho\left(\bar{\bar{T}}_J \right) = \max_{1 \leq i \leq n} |\mu_i| < 1$. Por lo tanto, teniendo en cuenta ambas condiciones se llega a la conclusión de que:

$$|\omega - 1| \leq \rho\left(\bar{\bar{T}}_\omega \right) < 1 \quad \Longrightarrow \quad \mathbf{0 < \omega < 2} \tag{70}$$

Donde:

- Si $0 < \omega < 1$ se trata de **sub-relajaciones.**
- Si $1 \leq \omega < 2$ se trata de **sobre relajaciones.**

- Si $\omega = 1$ se trata del **método de Gauss-Seidel.**

Nosotros siempre trabajamos con sobre relajaciones por lo que utilizaremos $\omega \in (1,2)$.

Observaciones:

- Si $\bar{\bar{A}}$ es definida positiva y se cumple $\omega \in (0,2)$ → SOR converge $\forall\ \bar{x}^{(0)}$.
- Si $\bar{\bar{A}}$ es simétrica y definida positiva → SOR converge $\forall\ \bar{x}^{(0)}$.

Aplicaciones en Ingeniería

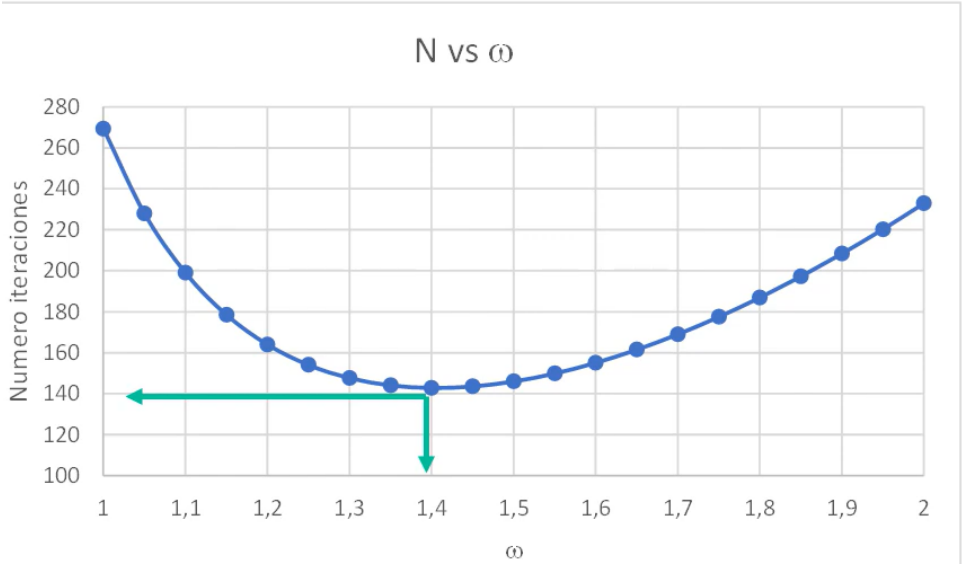
1) En términos prácticos siempre se utiliza $1 < \omega < 2$
2) Si $\bar{\bar{A}}$ es definida positiva y tridiagonal, la máxima velocidad se obtiene con:

$$\omega_{optimo} = \frac{2}{1 + \sqrt{1 - \rho\left(\bar{\bar{T}}_J\right)^2}}$$

   Siendo $\bar{\bar{T}}_j = \bar{\bar{D}}^{-1}(\bar{\bar{L}} + \bar{\bar{U}})$ la matriz de Jacobi, y en este caso $\rho\left(\bar{\bar{T}}_\omega\right) = \omega - 1$.

3) Si $\bar{\bar{A}}$ no satisface la condición anterior y se tienen que resolver múltiples SEL definidos por la misma $\bar{\bar{A}}$ y distintos $\bar{b}$ es conveniente hallar ω en términos prácticos. Para ello se realiza un gráfico de:
   - ***k vs ω***, donde para un valor constante del error absoluto se observa como varía el número de iteraciones k que necesitas para alcanzar la precisión deseada respecto a distintos valores de ω.
   - ***error vs ω***, donde para un número constante de iteraciones k se observa como cambia el valor del error absoluto a medida que varía ω.

   En ambos casos, el ω_{optimo} estará dado por el **mínimo de la curva graficada**, por lo que, se guardará ese valor para utilizarlo en todos los cálculos y reducir el costo computacional.



Sistemas de Ecuaciones NO Lineales (Guía 3b)

Generalizando el problema de una dimensión $F_{(x)} = 0$, se busca la solución del Sistema de Ecuaciones NO Lineales (SENL) definido por las siguientes **n** ecuaciones $\boldsymbol{E_i}$ no lineales:

$$\begin{cases} E_1: & F_1(x_1, x_2, \cdots, x_n) = 0 \\ E_2: & F_2(x_1, x_2, \cdots, x_n) = 0 \\ & \vdots \\ E_n: & F_n(x_1, x_2, \cdots, x_n) = 0 \end{cases}$$

Donde las $\boldsymbol{F_i}$ son funciones no lineales y x_i las variables independientes. En forma vectorial queda expresado como:

$$\bar{F}_{(\bar{x})} = \bar{0} \tag{71}$$

Donde $\bar{F}_{(\bar{x})}$ es un vector tal que $\bar{F} = (F_1, \cdots, F_n)^T$ y $\bar{x} = (x_1, \cdots, x_n)^T$.

Método de Punto Fijo

El planteo de la expresión (71) se muestra equivalente al problema de punto fijo dado por:

$$\bar{G}_{(\bar{\beta})} = \bar{\beta}$$

Donde $\bar{\beta}$ es el punto fijo de $\bar{G}$ si se define:

$$\bar{G}_{(\bar{x})} = \bar{x} - \bar{F}_{(\bar{x})} \tag{72}$$

Siendo $\bar{G} = (g_1, g_2, \cdots, g_n)^T$ un vector compuesto por funciones no lineales.

Teorema del punto fijo para SENL (Condiciones de convergencia)

Existen dos **condiciones suficientes, pero no necesarias** que garantizan la convergencia del método de punto fijo en un intervalo $[a, b]$.

1) Condición de **existencia**: garantiza que existe al menos una raíz dentro del intervalo.

$$Sea\ D = \{x\ tal\ que\ a_i \leq x_i \leq b_i\ \forall\ i = 1, \cdots, n\}$$

$$Si\ \bar{G}\ es\ una\ función\ continua\ en\ D \in R^n \to R^n\ tal\ que\ \bar{G}_{(\bar{x})} \in D\ \forall\ \bar{x} \in D \Longrightarrow$$

$$\Longrightarrow \bar{G}\ tiene\ un\ punto\ fijo\ en\ D$$

2) Condición de **unicidad**: garantiza que dicha raíz es única.

$$Si\ además\ \exists\ K < 1\ tal\ que\ \left|\frac{\partial g_i}{\partial x_j}\right| \leq \frac{K}{n}\ \forall\ \bar{x} \in D\ y\ \forall\ i, j = 1, \cdots, n.$$

$$Entonces\ la\ sucesión\ \bar{X}_{k+1} = \bar{G}_{(\bar{x}_k)}\ con\ k \geq 0\ converge\ al\ \text{único}$$

$$punto\ fijo\ \bar{\beta} \in D\ \forall x_0 \in D.$$

Corolario para el error de Truncamiento (cota)

Recordando el caso para una sola dimensión:

$$|x_k - \beta| \leq K^k \cdot \max(x_0 - a, b - x_0)$$

Surge inmediatamente una expresión equivalente para el caso de un sistema de ecuaciones:

$$\left\|\bar{X}_k - \bar{\beta}\right\|_\infty \leq \frac{K^k}{1-K} \|\bar{X}_1 - \bar{X}_0\|_\infty \tag{73}$$

Y de las misma forma que antes, si K es grande (tiende a 1) el método se considera muy lento, mientras que si K es chico (tiende a 0) el método es rápido.

Ejemplo

Contando con un sistema de 2x2 de la forma: $\begin{cases} x_1 - \frac{1}{3}\cos(x_2) = \frac{1}{3} \\ \frac{1}{20}e^{-x_1} + x_2 = -\frac{1}{2} \end{cases}$

Primero, defino mi $\bar{F}$:

$$\bar{F} = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1 - \frac{1}{3}\cos(x_2) - \frac{1}{3} \\ \frac{1}{20}e^{-x_1} + x_2 + \frac{1}{2} \end{bmatrix} = \bar{0}$$

Segundo, defino mi $\bar{G}$:

$$\bar{G}_{(\bar{x})} = \bar{x} - \bar{F}_{(\bar{x})} = \begin{bmatrix} g_1(x_1, x_2) \\ g_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1 - x_1 + \frac{1}{3}\cos(x_2) + \frac{1}{3} \\ x_2 - \frac{1}{20}e^{-x_1} - x_2 - \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\cos(x_2) + \frac{1}{3} \\ -\frac{1}{20}e^{-x_1} - \frac{1}{2} \end{bmatrix}$$

Finalmente, mi sistema resulta:

$$\begin{cases} {x_1}^{(k+1)} = \frac{1}{3}\cos\left({x_2}^{(k)}\right) + \frac{1}{3} \\ {x_2}^{(k+1)} = -\frac{1}{20}e^{-{x_1}^{(k)}} - \frac{1}{2} \end{cases}$$

Método de Gauss-Seidel

Se utiliza esta idea para intentar acelerar el proceso de convergencia. Recordamos que Gauss-Seidel plantea ir utilizando las variables calculadas dentro del mismo paso iterativo para obtener las demás variables restantes.

En consecuencia, el sistema antes presentado simplemente se modifica como:

$$\begin{cases} {x_1}^{(k+1)} = \frac{1}{3}\cos\left({x_2}^{(k)}\right) + \frac{1}{3} \\ {x_2}^{(k+1)} = -\frac{1}{20}e^{-{x_1}^{(k+1)}} - \frac{1}{2} \end{cases}$$

Observación: este método no siempre consigue acelerar la convergencia.

Método de Newton

Recordando el caso para una dimensión del método Newton-Raphson (15):

$$g_{NR} = x - \frac{F(x)}{F'(x)} = x - \varphi(x)F(x), \qquad con\ \varphi(x) = \frac{1}{F'(x)}$$

Se puede generalizar el planteo para n dimensiones como:

$$\bar{G}_{(\bar{x})} = \bar{x} - \bar{\bar{J}}_{(\bar{x})}{}^{-1}\bar{F}_{(\bar{x})} \tag{74}$$

Con $\varphi(x) = J_{(\bar{x})}{}^{-1}$ siendo $J_{(\bar{x})}$ la matriz jacobiana de la forma:

$$J_{ij}(x) = \frac{\partial F_i(x)}{\partial x_j} \tag{75}$$

La forma iterativa para el método resulta:

$$\bar{x}^{k+1} = \bar{G}(\bar{x}^k) = \bar{x}^k - \bar{\bar{J}}_{(\bar{x})}{}^{-1}\bar{F}_{(\bar{x}^k)} \tag{76}$$

Observaciones:

- Se obtiene una **convergencia cuadrática.**
- El costo computacional es mayor ya que se requieren calcular las derivadas e invertir el Jacobiano, en cada paso de iteración.

Formas prácticas del método de Newton

Una forma más sencilla del método se puede obtener a partir de plantear el siguiente cambio de variables:

$$\bar{y}^k = -\bar{\bar{J}}_{(\bar{x}^k)}{}^{-1}\bar{F}_{(\bar{x}^k)} \tag{77}$$

$$\Longrightarrow J_{(x^k)}y^k = -J_{(x^k)}J_{(x^k)}{}^{-1}F_{(x^k)} = -\bar{\bar{I}}\,F_{(x^k)} \Longrightarrow$$

$$J_{(x^k)}y^k = -F_{(x^k)} \tag{78}$$

Siendo la expresión (78) un **sistema lineal** ya que la matriz $\bar{\bar{J}}$ y el vector $\bar{F}$ están evaluados en x^k, es decir, son matrices y vectores con números. De manera que, del sistema (78) se puede despejar el vector y^k y reescribir la ecuación (76) como:

$$x^{k+1} = x^k - J_{(x^k)}{}^{-1}F_{(x^k)} = x^k + y^k \tag{79}$$

Es decir, el método de Newton consiste básicamente en resolver un Sistema de Ecuaciones Lineales. Para el ejemplo anterior correspondería:

$$\bar{F} = \begin{bmatrix} F_1(x_1, x_2) \\ F_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1 - \frac{1}{3}\cos(x_2) - \frac{1}{3} \\ \frac{1}{20}e^{-x_1} + x_2 + \frac{1}{2} \end{bmatrix} \rightarrow \begin{cases} {x_1}^{k+1} = \frac{1}{3}\cos({x_2}^k) + \frac{1}{3} \\ {x_2}^{k+1} = -\frac{1}{20}e^{-{x_1}^k} - \frac{1}{2} \end{cases}$$

$$J_{(x^k)} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{3} sen(x_2) \\ -\frac{1}{20} e^{-x_1} & 1 \end{bmatrix} \rightarrow Además, hay\ que\ invertirlo$$

Método Cuasi-Newton

Para evitar el cálculo del jacobiano del caso anterior, en casos reales de matrices de gran dimensión, se aplica la **idea de la secante** donde se aproxima la derivada como:

$$\frac{\partial F_{i_{(x^k)}}}{\partial x_j} \approx \frac{F_{i_{(x^k + h e_j)}} - F_{i_{(x^k)}}}{h} \tag{80}$$

Con $e_j = (0,0,\cdots,1,\cdots,0,0)$, léase un incremento pequeño en la variable x_j.

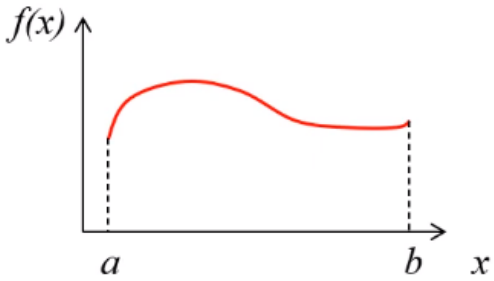
Observaciones:

Generalmente la matriz jacobiana se calcula cada cierta cantidad de pasos con el fin de reducir el costo computacional un poco. Es decir, se utiliza una $\bar{\bar{J}}$ para cierta cantidad de iteraciones y luego se la vuelve a calcular.

Integración (Guía 6)

El problema tipo a resolver es:

$$Hallar\ una\ aproximación\ para\ la\ integral\ definida: I = \int_a^b F(x)dx$$



La utilidad del próximo planteo se observa en los casos donde:

- I no está en tablas.
- I es difícil de calcular.
- I se debe calcular muchas veces.

La solución que se propone se denomina **cuadratura numérica** y queda expresada como:

$$I \approx \sum_i \omega_i F_{(x_i)} \tag{81}$$

Donde ω_i representa al **factor de peso** y x_i son los puntos donde se evalúa la función F.

Aplicación en Mecánica computacional

Este planteo es importante para resolver ecuaciones diferenciales que representan sistemas de ingeniería, a través de su forma integral. En general, el sistema posee la siguiente forma:

$$\Delta(\bar{r},t)\bar{u} = \rho(\bar{r},t)$$

$$\Delta(\bar{r},t)\bar{u} - \rho(\bar{r},t) = 0$$

$$I = \int_V W[\Delta(\bar{r},t)\bar{u} - \rho(\bar{r},t)]dV\,, \qquad con\ R(\bar{u}) = \Delta(\bar{r},t)\bar{u} - \rho(\bar{r},t)$$

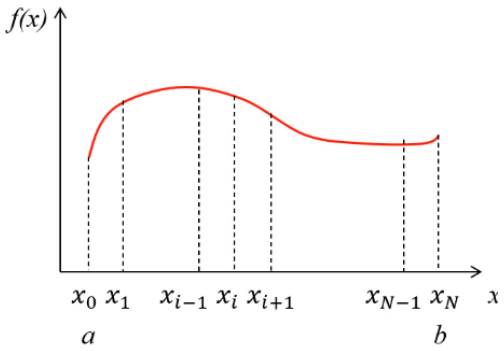
$$I = \sum_{\Delta V} \int_{\Delta V} W * R(\bar{u})dV \rightarrow \boldsymbol{Método\ de\ elementos\ finitos}$$

Donde Δ es una operación diferencial cualquiera, $\bar{r}$ representa a las variables del espacio, t representa a la variable temporal, u es una función vectorial incógnita y $\rho(\bar{r},t)$ representa a una fuente de energía o fuerzas, las cuales son conocidas para el problema.

Escalas de integración

Intervalos global y local

Descomponemos el **intervalo global** $[a, b]$ en **sub-intervalos locales** $[x_{i-1}, x_i]$. La integral original se puede expresar como:



$$I = \int_a^b F(x)dx = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} F(x)dx = \sum_{i=1}^{N} I_i$$

Donde las $I_i \equiv \int_{x_{i-1}}^{x_i} F(x)dx$ son las **sub-integrales** evaluadas en cada sub-intervalo. En el caso particular de que los sub-intervalos sean regulares:

$$x_i = a + ih, \qquad con\ i = 0, \dots, N\ y\ h = \frac{b-a}{N}$$

Siendo h la longitud del sub-intervalo.

Sub-integrales

Si se considera que los sub-intervalos son suficientemente pequeños, se desarrolla $F(x)$ alrededor del punto medio del sub-intervalo ${x_i}^{(m)} = \frac{1}{2}(x_{i-1} + x_i)$:

$$F(x) = \sum_{n=0}^{\infty} \frac{1}{n!} {f_{({x_i}^{(m)})}}^{(n)} \left(x - {x_i}^{(m)}\right)^n \rightarrow \boldsymbol{Serie\ de\ Taylor} \qquad (82)$$

Haciendo un cambio de variables $u \equiv x - {x_i}^{(m)}$ y expresando ${f_{({x_i}^{(m)})}}^{(n)} = {f_m}^{(n)}$, la sub-integral se puede reescribir como:

$$I_i \equiv \int_{-\frac{h}{2}}^{\frac{h}{2}} \left[\sum_{n=0}^{\infty} \frac{1}{n!} {f_m}^{(n)} u^n\right] du = \sum_{n=0}^{\infty} \frac{1}{n!} {f_m}^{(n)} \left[\int_{-\frac{h}{2}}^{\frac{h}{2}} u^n du\right]$$

Como en el último corchete se están integrando monomios en un intervalo simétrico respecto del cero, aquellos términos donde **n sea impar** tendrán integral nula y solo sobrevivirán los términos con **n par**. Es decir:

$$I_i = \sum_{n(par)} \frac{1}{n!} {f_m}^{(n)} \left[\frac{1}{n+1} u^{n+1}\right]_{-\frac{h}{2}}^{\frac{h}{2}} = \sum_{n(par)} \frac{1}{n!} {f_m}^{(n)} \frac{2}{n+1} \left(\frac{h}{2}\right)^{n+1}$$

Si desarrollamos la última expresión podemos sacar h como factor común, entonces:

$$I_i = h\left[f_m + \frac{1}{24} f_m'' h^2 + \frac{1}{1920} {f_m}^{iv} h^4 + \cdots\right] \qquad (83)$$

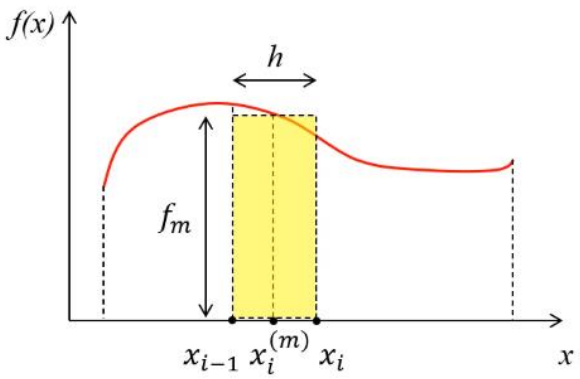
Hasta este punto no se ha truncado el desarrollo, por lo que no hay error de truncamiento. Es a la hora de truncar la expresión (83) cuando surgen los distintos métodos de aproximación.

Método del Rectángulo

Corresponde a aproximar las sub-integrales locales definidas en (83) al orden más bajo posible, o sea $n = 0$:

$$R_i = hf_m = hf_{\left(\frac{x_{i-1}+x_i}{2}\right)} \qquad (84)$$

Donde la altura viene dada al evaluar $f_{(x_i{}^{(m)})}$ y la base del rectángulo está dada por la longitud del intervalo h (paso). El **error de truncamiento** viene dado por la diferencia entre la expresión (84) y la (83) y, despreciando términos de orden superior, resulta de orden 2:



$$R_i - I_i = h\left\{-\frac{1}{24}f_m''h^2 - \frac{1}{1920}f_m{}^{iv}h^4 + O(h^6)\right\} \cong -\frac{1}{24}f_m''h^2 \qquad (85)$$

Generalizando para todos los sub-intervalos entre $[a, b]$:

$$I \approx \sum_{i=1}^{N} R_i = \sum_{i=1}^{N} hf_m \to R = h\sum_{i=1}^{N} f_{\left(\frac{x_{i-1}+x_i}{2}\right)} \qquad (86)$$

Siendo esta la regla del rectángulo para el intervalo global. De igual forma que antes, el error de truncamiento surge de la expresión:

$$R - I = \sum_{i=1}^{N} R_i - \sum_{i=1}^{N} I_i = \sum_{i=1}^{N}(R_i - I_i) = -\frac{1}{24}h^2\left\{\sum_{i=1}^{N} f_m''h\right\} + O(h^4) \qquad (87)$$

Y resulta ser de **orden 2** como se expuso antes**.** Además, si $h \to 0$, lo que está contenido entre llaves es un valor constante.

Método del Trapecio

Dado que a veces no es posible evaluar la función f en el punto medio del intervalo (por ejemplo, cuando se hace un muestreo), se dispone del **método del trapecio** para resolver dichos casos. Entonces, se hará uso de los puntos disponible como los puntos x_i.

El método surge del siguiente proceso:

1) Desarrollamos la función f en el extremo izquierdo del sub-intervalo y alrededor del punto medio: $x_{i-1} = x_i{}^{(m)} - \frac{h}{2}$

$$f_{(x_{i-1})} = \sum_n \frac{1}{n!}f_m{}^{(n)}\left(\frac{-h}{2}\right)^n = \sum_n \frac{(-1)^n}{2^n n!}f_m{}^{(n)}h^n = f_m - \frac{1}{2}f_m'h + \frac{1}{8}f_m''h^2 - \frac{1}{48}f_m'''h^3 + \cdots$$

2) Desarrollando ahora en el extremo derecho:

$$f_{(x_i)} = \sum_n \frac{1}{n!}f_m{}^{(n)}\left(\frac{h}{2}\right)^n = \sum_n \frac{1^n}{2^n n!}f_m{}^{(n)}h^n = f_m + \frac{1}{2}f_m'h + \frac{1}{8}f_m''h^2 + \frac{1}{48}f_m'''h^3 + \cdots$$

3) Se observa que, si se suman las expresiones, los términos en rojo se pueden cancelar ya que son iguales en módulo, pero de signo opuesto. Esto sucede para todos los términos impares, por lo que resulta:

$$f_{(x_{i-1})} + f_{(x_i)} = 2f_m + 2\frac{1}{8}f_m''h^2 + 2\frac{1}{384}f_m^{iv}h^4 \dots$$

4) Despejo el valor de la función f evaluada en el punto medio, es decir, f_m.

$$f_m = \frac{f_{(x_{i-1})} + f_{(x_i)}}{2} - \frac{1}{8}f_m''h^2 - \frac{1}{384}f_m^{iv}h^4 \dots$$

5) Reemplazando en la expresión (83) (donde está el igual en rojo se reemplaza):

$$I_i = hf_m + h\left[\frac{1}{24}f_m''h^2 + \frac{1}{1920}{f_m}^{iv}h^4 + \cdots\right] =$$

$$= h\left(\frac{f_{(x_{i-1})} + f_{(x_i)}}{2} - \frac{1}{8}f_m''h^2 - \frac{1}{384}f_m^{iv}h^4 \dots\right) + h\left[\frac{1}{24}f_m''h^2 + \frac{1}{1920}{f_m}^{iv}h^4 + \cdots\right]$$

Reagrupando:

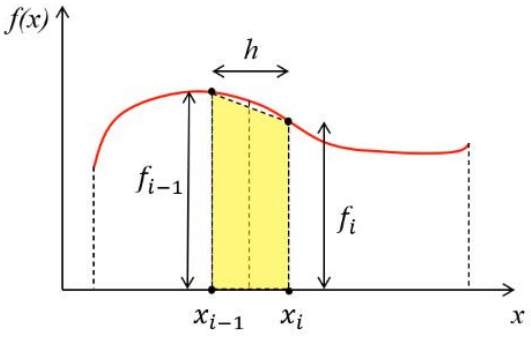
$$= h\frac{f_{(x_{i-1})} + f_{(x_i)}}{2} + h\left[\frac{1}{24}f_m''h^2 - \frac{1}{8}f_m''h^2 + \frac{1}{1920}{f_m}^{iv}h^4 - \frac{1}{384}f_m^{iv}h^4 + \cdots\right] =$$

$$= h\frac{f_{(x_{i-1})} + f_{(x_i)}}{2} + h\left[-\frac{1}{12}f_m''h^2 - \frac{1}{480}{f_m}^{iv}h^4 - \cdots\right]$$

$$= h\frac{f_{(x_{i-1})} + f_{(x_i)}}{2} - h\left[\frac{1}{12}f_m''h^2 + \frac{1}{480}{f_m}^{iv}h^4 + \cdots\right]$$

Finalmente, tomando el orden más bajo (marcado en rojo) obtenemos la **regla del Trapecio**:

$$T_i = \frac{h}{2}\left(f_{(x_{i-1})} + f_{(x_i)}\right) \quad (88)$$



Y el **error de truncamiento** está dado por diferencia entre la expresión (88) y la (83):

$$T_i - I_i = h\left(\frac{1}{12}f_m''h^2 + \frac{1}{480}{f_m}^{iv}h^4 + \cdots\right) \quad (89)$$

Resultando entonces un método de **orden 2**. Generalizando para todos los sub-intervalos:

$$I \approx \sum_{i=1}^{N} T_i = \sum_{i=1}^{N} h\left(\frac{f_{(x_{i-1})} + f_{(x_i)}}{2}\right) \rightarrow T = h\sum_{i=1}^{N}\left(\frac{f_{(x_{i-1})} + f_{(x_i)}}{2}\right)$$

Para minimizar el error se puede reducir el número de operaciones:

$$T = h\frac{f_0}{2} + h\sum_{i=1}^{N-1} f_i + h\frac{f_N}{2} = \frac{h}{2}\left[f_0 + 2\sum_{i=1}^{N-1} f_i + f_N\right] \quad (90)$$

De igual forma que antes, el error de truncamiento surge de la expresión:

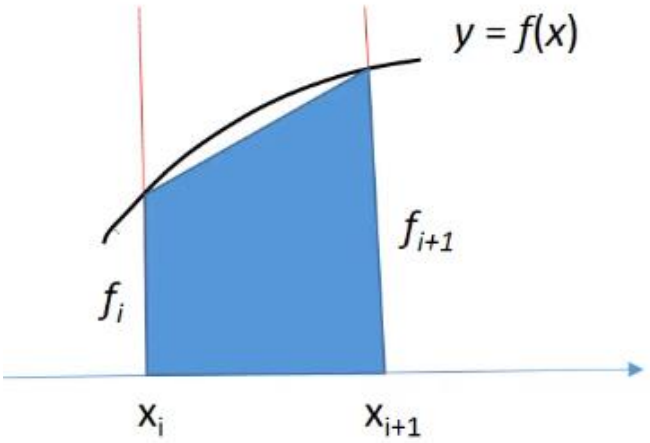
$$T - I = \sum_{i=1}^{N} T_i - \sum_{i=1}^{N} I_i = \sum_{i=1}^{N} (T_i - I_i) = -\frac{h^2}{12}\left\{\sum_{i=1}^{N} f_m'' h\right\} + O(h^4) \qquad (91)$$

Y sigue siendo de **orden 2**.

Otra forma de obtención del método (más sencilla para mí)

(https://www.youtube.com/watch?v=sJKervAz_EM&t=1323s)

Tenemos la siguiente función $f_{(x)}$ (línea negra):



Vamos a aproximar el área bajo la curva por la de un trapecio con base $(x_{i+1} - x_i)$ y alturas $f_{(x_{i+1})}$ y $f_{(x_i)}$ mayor y menor, respectivamente. Recordamos la fórmula para el área de un trapecio:

$$A = base * \frac{Altura\ mayor + Altura\ menor}{2}$$

Reemplazamos para nuestro caso:

$$I_i = \int_{x_i}^{x_{i+1}} f_{(x)} dx = (x_{i+1} - x_i) * \frac{f_{(x_{i+1})} + f_{(x_i)}}{2} + et_i^{(T)}$$

Donde $et_i^{(T)}$ representa el error de truncamiento cometido al aproximar por la presenta regla. Además, tenemos que la distancia entre los puntos en el eje de abscisas es el paso:

$$h = x_{i+1} - x_i$$

$$I_i = h * \frac{f_{(x_{i+1})} + f_{(x_i)}}{2} + et_i^{(T)} = \frac{h}{2}\left[f_{(x_{i+1})} + f_{(x_i)}\right] + et_i^{(T)}$$

Esta es la **regla del trapecio para un subintervalo**. Respecto al error de truncamiento $et_i^{(T)}$, para hallarlo tomamos la fórmula del subintervalo y proponemos una $f_{(x)}$ cuyo orden sea 1 grado mayor a la curva empleada para integrar. En este caso, como utilizamos un trapecio para aproximar quiere decir que queremos aproximar la forma de la curva con una recta que corresponde a la diagonal superior del trapecio. Por lo tanto, deberemos plantear una función tal que sea de un orden superior, es decir, una forma cuadrática:

$$f_{(x)} = (x - x_i)^2$$

Ahora, reemplazamos en la fórmula:

$$I_i = \frac{h}{2}\left[f_{(x_{i+1})} + f_{(x_i)}\right] + et_i^{(T)}$$

$$\begin{cases} I_i = \int_{x_i}^{x_{i+1}} (x - x_i)^2 dx = \frac{(x - x_i)^3}{3}\Big|_{x_i}^{x_{i+1}} = \frac{(x_{i+1} - x_i)^3}{3} - \frac{(x_i - x_i)^3}{3} = \frac{h^3}{3} \\ \frac{h}{2}\left[f_{(x_{i+1})} + f_{(x_i)}\right] = \frac{h}{2}\left[(x_{i+1} - x_i)^2 + (x_i - x_i)^2\right] = \frac{h}{2}[h^2 + 0] = \frac{h^3}{2} \end{cases}$$

$$\frac{h^3}{3} = \frac{h^3}{2} + et_i^{(T)} \rightarrow et_i^{(T)} = \frac{h^3}{3} - \frac{h^3}{2} = -\frac{h^3}{6}$$

Ahora bien, para una $f_{(x)}$ arbitraria como empleamos rectas para aproximar quiere decir que únicamente podemos aproximar de manera exacta funciones lineales, por lo tanto, el error de truncamiento aparece para ordenes mayores. Si desarrollamos por Taylor veríamos que el primer término que no se cancela corresponde a la derivada segunda (por lo mencionado antes). Entonces, para generalizar lo que hacemos es multiplicar la expresión que hallamos antes por dicho término del desarrollo de Taylor de la siguiente forma:

$$I_i - T_{i_{(h)}} = et_i^{(T)} = -\frac{h^3}{6}\frac{f''_{(x_i)}}{2!} \rightarrow et_i^{(T)} = \frac{h^3}{12} f''_{(x_i)}$$

Para pasar al **intervalo global** basta con sumar los subintervalos, considerando que el paso es uniforme entre los extremos del intervalo de integración, es decir:

$$h = \frac{b - a}{n}$$

$$I = \sum_{i=0}^{n} I_i = \sum_{i=0}^{n}\left[h * \frac{f_{(x_{i+1})} + f_{(x_i)}}{2} + et_i^{(T)}\right] = \frac{h}{2}\sum_{i=0}^{n}\left[f_{(x_{i+1})} + f_{(x_i)}\right] + \sum_{i=0}^{n} et_i^{(T)}$$

Ahora, si tenemos en cuenta que los extremos se emplean como nodos una sola vez (solo al inicio y al final) mientras que los demás puntos se emplean dos veces cada uno como nodos, podemos reescribir la fórmula del Trapecio como:

$$T_{(h)} = \frac{h}{2}\left[f_a + 2\sum_{i=0}^{n-1} f_{(x_i)} + f_N\right] \quad I = T_{(h)} + et^{(T)}$$

Respecto al error de truncamiento global $et^{(T)}$, si observamos la fórmula anterior se puede ver que éste es la sumatoria de los errores de truncamiento de cada sub-intervalo $et_i^{(T)}$. Por lo tanto, como realizamos n integrales tendremos n errores de truncamiento iguales, es decir:

$$et^{(T)} = \sum_{i=0}^{n} et_i^{(T)} = net_i^{(T)} = n\frac{h^3}{12} f''_{(x_i)}$$

Ahora, tomamos como cota de la derivada segunda aquel valor desconocido ξ del intervalo cuya evaluación daría como resultado aquella derivada segunda de mayor módulo. Además, terminamos de expresar el error de truncamiento en función del paso:

$$et^{(T)} = \frac{(b - a)}{h}\frac{h^3}{12} f''_{(\xi)} \rightarrow et^{(T)} = \frac{(b - a)h^2}{12} f''_{(\xi)}$$

Concepto de Precisión

La precisión de una fórmula de cuadratura va a ser el entero positivo más grande **n** tal que la fórmula de cuadratura es exacta para:

$$x^k, \qquad con\ k = 0, \dots, n$$

Otra forma de verlo es que la precisión de una fórmula de cuadratura está asociada con el polinomio de mayor grado que la fórmula puede integrar de manera exacta.

Por ejemplo, para las **reglas del Rectángulo y del Trapecio**, dado que utilizan rectas para aproximar polinomios poseen una **precisión de 1**. Es decir, solo pueden integrar de manera exacta polinomios de grado 1.

Extrapolación de Richardson

Para poder aplicar el método de extrapolación de Richardson se debe conocer el término de error de cada aproximación en forma predecible. Usualmente alcanza con conocer el grado del error en términos de h.

Sirve para lograr precisión 2 o más a partir de métodos de precisión 1

La fórmula para realizar una Extrapolación de Richardson se expresa como:

$$N_j\left(\frac{h}{2}\right) = N_{j-1}\left(\frac{h}{2}\right) + \left(\frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1} - 1}\right) \tag{92}$$

Donde N representa a una regla numérica de integración (léase, trapecio, rectángulo u otra), j representa el orden de la nueva fórmula obtenida siendo de orden h^j. Además, se parte de la hipótesis que h es suficientemente pequeño (ver desarrollo en el video teórico).

Observación: dicha fórmula se obtiene de partir el intervalo h en la mitad. Si se partiese el intervalo de una forma diferente se conseguiría una fórmula distinta.

Si se observa la expresión (92), lo que propone Richardson no es otra cosa que obtener una solución más exacta a partir de dos soluciones, donde a la más exacta (dada por el paso $h/2$) se le suma una estimación del error, dado por la diferencia entre la integral calculada con un paso h y con un paso $h/2$. Este proceso se puede generalizar para cualquier método de integración de la siguiente manera:

$$I_{Rich} = I_{(h_1)} + \frac{I_{(h_1)} - I_{(h_2)}}{\left(\frac{h_2}{h_1}\right)^p - 1} + O(h^r) \tag{93}$$

Siendo h_i los pasos de cálculo tales que $h_1 < h_2$, p es el orden de convergencia del método por el cual se calcula la integral I y $r > p$ da cuenta del orden del error de truncamiento. Se puede notar que la formula aplica para nodos no equidistantes, a diferencia de (92).

Método de Simpson

Sabiendo que el método de Trapecios es de orden 2 en lo que respecta al error de truncamiento, se puede aplicar una extrapolación de Richardson (92) y generar un método cuyo error de truncamiento sea de orden 3.

$$S_3\left(\frac{h}{2}\right) = T_2\left(\frac{h}{2}\right) + \frac{T_2\left(\frac{h}{2}\right) - T_2(h)}{2^{3-1}-1} = T_2\left(\frac{h}{2}\right) + \frac{1}{3}T_2\left(\frac{h}{2}\right) - \frac{1}{3}T_2(h) = \frac{4}{3}T_2\left(\frac{h}{2}\right) - \frac{1}{3}T_2(h)$$

Desarrollando ambos términos:

$$\frac{4}{3}T_2\left(\frac{h}{2}\right) = \frac{4}{3}\left[\frac{h}{2}\frac{f_0}{2} + \frac{h}{2}\sum_{i=1}^{N-1} f_i + \frac{h}{2}\frac{f_N}{2}\right] = \frac{4}{3}\frac{h}{2}\left[\frac{f_0}{2} + f_{\frac{1}{2}} + f_1 + \cdots + f_{N-1} + f_{N-\frac{1}{2}} + \frac{f_N}{2}\right]$$

$$-\frac{1}{3}T_2(h) = -\frac{1}{3}\left[h\frac{f_0}{2} + h\sum_{i=1}^{N-1} f_i + h\frac{f_N}{2}\right] = -\frac{h}{3}\left[\frac{f_0}{2} + f_1 + f_2 + \cdots + f_{N-2} + f_{N-1} + \frac{f_N}{2}\right]$$

Recomponiendo la expresión:

$$\frac{4}{3}T_2\left(\frac{h}{2}\right) - \frac{1}{3}T_2(h) = \frac{4}{6}h\left[\frac{f_0}{2} + f_{\frac{1}{2}} + \cdots + f_{N-\frac{1}{2}} + \frac{f_N}{2}\right] - \frac{h}{6}[f_0 + 2f_1 + \cdots + 2f_{N-1} + f_N]$$

$$= \frac{1}{6}h\left[4\frac{f_0}{2} + 4f_{\frac{1}{2}} + \cdots + 4f_{N-\frac{1}{2}} + 4\frac{f_N}{2} - f_0 - 2f_1 - \cdots - 2f_{N-1} - f_N\right] =$$

$$S_3\left(\frac{h}{2}\right) = \frac{1}{3}\left(\frac{h}{2}\right)\left[f_0 + 4f_{\frac{1}{2}} + 2f_1 + \cdots + 2f_{N-1} + 4f_{N-\frac{1}{2}} + f_N\right]$$

Regla de Simpson para sub-intervalos:

$$S_i = \frac{h}{3}[f_{i-1} + 4f_i + f_{i+1}] \tag{94}$$

Con un error de truncamiento:

$$S_i - I_i = -\frac{1}{90}{f_{(\mu)}}^{iv}h^4 \tag{95}$$

Entonces, la **regla de Simpson 1/3 compuesta** para un intervalo global resulta:

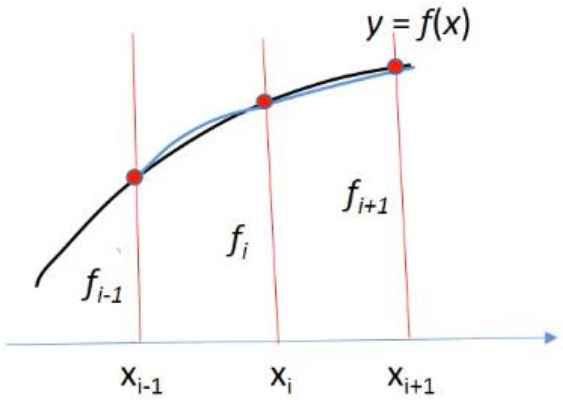
$$S_{(h)} = \frac{h}{3}\left[f_{(a)} + 2\sum_{i=1}^{N/2-1} f_{2i} + 4\sum_{i=1}^{N/2} f_{2i-1} + f_{(b)}\right]$$

Esta ecuación aplica par N par con $x_i = a + ih$ con $i = 0, \dots, N$ y el paso resulta $h = \frac{b-a}{N}$. De la misma forma, el error de truncamiento esta dado por:

$$S - I = -\frac{(b-a)}{180}{f_{(\mu)}}^{iv}h^4 \tag{96}$$

Otra forma de obtención del método (más clara para mí)

Se puede deducir la fórmula de la regla de Simpson a partir del método de los coeficientes indeterminados de Gauss. La regla elemental de Simpson aproxima integrales como el área debajo de un segmento de parábola, planteamos el caso para un subintervalo de 3 puntos uniformemente separados tal que:



La separación entre puntos la denominamos h. Entonces, la integral se calculará como:

$$I_i = \int_{x_{i-1}}^{x_{i+1}} f_{(x)} dx$$

Ahora, planteamos el método de coeficientes indeterminados de manera que sea equivalente al cálculo de la integral mostrada.

$$I_i = \int_{x_{i-1}}^{x_{i+1}} f_{(x)} dx = a f_{(x_{i-1})} + b f_{(x_i)} + c f_{(x_{i+1})} + et_i^{(S)}$$

Ahora proponemos funciones $f_{(x)}$ de orden creciente tantas veces como coeficientes necesitamos determinar. En nuestro caso tenemos que:

$$f_{(x)} = 1 \rightarrow \int_{x_{i-1}}^{x_{i+1}} 1 dx = (x_{i+1} - x_{i-1}) = 2h = a * 1 + b * 1 + c * 1$$

$$f_{(x)} = (x - x_i) \rightarrow \int_{x_{i-1}}^{x_{i+1}} (x - x_i) dx = a(x_{i-1} - x_i) + b(x_i - x_i) + c(x_{i+1} - x_i)$$

$$\frac{1}{2}[(x_{i+1} - x_i)^2 - (x_{i-1} - x_i)^2] = \frac{1}{2}[h^2 - h^2] = 0 = -ah + ch$$

$$f_{(x)} = (x - x_i)^2 \rightarrow \int_{x_{i-1}}^{x_{i+1}} (x - x_i)^2 dx = a(x_{i-1} - x_i)^2 + b(x_i - x_i)^2 + c(x_{i+1} - x_i)^2$$

$$\frac{1}{3}[(x_{i+1} - x_i)^3 - (x_{i-1} - x_i)^3] = \frac{1}{3}[h^3 + h^3] = \frac{2}{3}h^3 = ah^2 + ch^2$$

Nos queda el sistema:

$$\begin{cases} 2h = a + b + c \\ 0 = -ah + ch \\ \frac{2}{3}h^3 = ah^2 + ch^2 \end{cases} \rightarrow a = c = \frac{h}{3} \quad b = \frac{4}{3}h$$

Reescribimos nuestra expresión para la integral:

$$I_i = \frac{h}{3} f_{(x_{i-1})} + \frac{4}{3} h f_{(x_i)} + \frac{h}{3} f_{(x_{i+1})} + et_i^{(S)} = \frac{h}{3}\left[f_{(x_{i-1})} + 4f_{(x_i)} + f_{(x_{i+1})}\right] + et_i^{(S)}$$

Esta es la **regla de Simpson para un subintervalo.** Respecto al error de truncamiento $et_i^{(S)}$, para conocer su expresión planteamos la integral que desarrollamos, pero empleando una función $f_{(x)}$ de un orden mayor al que se logra integrar de manera exacta, es decir, una forma cúbica:

$$f_{(x)} = (x - x_i)^3$$

$$\int_{x_{i-1}}^{x_{i+1}} (x - x_i)^3 dx = \frac{h}{3}[(x_{i-1} - x_i)^3 + 4(x_i - x_i)^3 + (x_{i+1} - x_i)^3] + et_i^{(S)}$$

$$\frac{1}{4}[(x_{i+1} - x_i)^4 - (x_{i-1} - x_i)^4] = \frac{1}{4}[h^4 - h^4] = \frac{h}{3}[-h^3 + h^3] + et_i^{(S)}$$

$$0 = 0 + et_i^{(S)} \rightarrow et_i^{(S)} = 0$$

El resultado obtenido muestra que a pesar de que estamos aproximando con arcos de parábola, la regla de Simpson simple permite determinar de manera exacta la integral de una forma cúbica Entonces, probemos ahora con un orden mayor para la función:

$$f_{(x)} = (x - x_i)^4$$

$$\int_{x_{i-1}}^{x_{i+1}} (x - x_i)^4 dx = \frac{h}{3}[(x_{i-1} - x_i)^4 + 4(x_i - x_i)^4 + (x_{i+1} - x_i)^4] + et_i^{(S)}$$

$$\frac{1}{5}[(x_{i+1} - x_i)^5 - (x_{i-1} - x_i)^5] = \frac{1}{5}[h^5 + h^5] = \frac{h}{3}[h^4 + h^4] + et_i^{(S)}$$

$$\frac{2}{5}h^5 = \frac{2}{3}h^5 + et_i^{(S)} \rightarrow et_i^{(S)} = \frac{2}{5}h^5 - \frac{2}{3}h^5 = -\frac{4}{15}h^5$$

Ahora generalizamos el error para una función $f_{(x)}$ arbitraria multiplicando la expresión obtenida por el término correspondiente a la derivada cuarta en el desarrollo de Taylor:

$$et_i^{(S)} = -\frac{4}{15}h^5 \frac{f_{(x_i)}^{iv}}{4!} = -\frac{4}{15}\frac{f_{(x_i)}^{iv}}{24}h^5 \rightarrow et_i^{(S)} = -\frac{h^5}{90} f_{(x_i)}^{iv}$$

Para pasar al **intervalo global** basta con sumar los subintervalos, considerando que el paso es uniforme entre los extremos del intervalo de integración, es decir:

$$h = \frac{b - a}{n}$$

$$I = \sum_{i=0}^{\frac{n}{2}-1} I_i = \sum_{i=0}^{\frac{n}{2}-1} \left[\frac{h}{3}\left[f_{(x_{2i})} + 4f_{(x_{2i+1})} + f_{(x_{2i+2})}\right] + et_i^{(S)}\right]$$

$$I = \frac{h}{3}\sum_{i=0}^{\frac{n}{2}-1}\left[f_{(x_{2i})} + 4f_{(x_{2i+1})} + f_{(x_{2i+2})}\right] + \sum_{i=0}^{\frac{n}{2}-1} et_i^{(S)}$$

Ahora, si tenemos en cuenta que los extremos se emplean como nodos una sola vez (solo al inicio y al final) y además que los nodos pares se utilizan 2 veces, pues en cada sub integral hacen de extremos de la misma, podemos entonces reescribir la **fórmula de Simpson $1/3$ compuesta** como:

$$S_{(h)} = \frac{h}{3}\left[f_0 + 2\sum_{i=1}^{\frac{n}{2}-1} f_{(x_{2i})} + 4\sum_{i=0}^{\frac{n}{2}} f_{(x_{2i-1})} + f_N\right] \quad I = S_{(h)} + et^{(S)}$$

Cabe destacar que la cantidad de nodos disponibles n siempre debe ser par.

Respecto al error de truncamiento global $et^{(S)}$, si observamos la fórmula anterior se puede ver que éste es la sumatoria de los errores de truncamiento de cada sub-intervalo $et_i^{(S)}$. Por lo tanto, como los límites de la sumatoria van desde $\left[0, \frac{n}{2} - 1\right]$, si sumamos 1 a ambos extremos la cantidad de cálculos sigue siendo la misma $\left[1, \frac{n}{2}\right]$, léase se calculan $\frac{n}{2}$ errores de truncamiento iguales. Resolvemos:

$$et^{(T)} = \sum_{i=0}^{\frac{n}{2}-1} et_i^{(T)} = \frac{n}{2} et_i^{(T)} = \frac{n}{2}\left(-\frac{h^5}{90} f^{iv}_{(x_i)}\right)$$

Ahora, tomamos como cota de la derivada cuarta aquel valor desconocido ξ del intervalo cuya evaluación daría como resultado aquella derivada cuarta de mayor módulo. Además, terminamos de expresar el error de truncamiento en función del paso:

$$et^{(T)} = -\frac{(b-a)}{h}\frac{h^5}{180} f^{iv}{}_{(\xi)} \rightarrow et^{(T)} = -\frac{(b-a)h^4}{180} f^{iv}{}_{(\xi)}$$

Método de Romberg

Este método aplica el uso consecutivo de la extrapolación de Richardson (92). La construcción de las aproximaciones se consigue por los siguientes pasos:

1) Utiliza la regla compuesta del trapecio para un intervalo global como aproximación preliminar.
2) Aplica la extrapolación de Richardson hasta conseguir el orden deseado.

El método de cálculo es sencillo y se suele utilizar un diagrama en forma de matriz como se muestra a continuación:

$$\begin{matrix} h \\ h/2 \\ h/4 \\ \vdots \\ \frac{h}{2^{j-1}} \end{matrix}\begin{bmatrix} R_{11} & & & & \\ R_{21} & R_{22} & & & \\ R_{31} & R_{32} & R_{33} & & \\ \vdots & \ddots & \ddots & \ddots & \\ R_{n1} & \cdots & \cdots & R_{nn} & \end{bmatrix}$$

Donde la primera columna generalizada como R_{j1} corresponde a la regla del Trapecio $T_{(h)}$ y la segunda R_{j2} a la regla de Simpson 1/3. Para cada fila que se avanza el paso h disminuye a la mitad respecto al anterior. Es decir, tenemos que:

$$h_j = \frac{h_1}{2^{j-1}}$$

Siendo h_1 el paso empleado en la primer fila y h_j el paso a utilizar para la fila j. Este sistema de cálculo es muy útil para aumentar la precisión del cálculo con gran velocidad puesto que avanzar en el sentido de la

diagonal R_{jj} aumenta la precisión de a dos dígitos significativos. Es decir, empezamos con la regla del Trapecio, la cual es de orden 2 $O(h^2)$, en la primer columna luego pasamos a la regla de Simpson en la segunda columna la cual presenta un orden 4 $O(h^4)$, luego pasamos a un orden 6 en la tercer columna y así sucesivamente.

En pocas palabras, permite aumentar la precisión a un ritmo mayor que el que se consigue al disminuir el paso a la mitad respecto al anterior, es decir, bajando por la primer columna la convergencia será más lenta que yendo por la diagonal principal.

Ahora bien, los R_{ji} se calculan a partir de la fórmula de extrapolación de Richardson (92):

$$N_j\left(\frac{h}{2}\right) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}\left(\frac{h}{2}\right) - N_{j-1}(h)}{2^{j-1}-1}$$

Si la adaptamos a nuestro sistema tenemos que:

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

$$R_{ij} = \frac{4^{j-1}}{4^{j-1}-1} R_{i,j-1} - \frac{1}{4^{j-1}-1} R_{i-1,j-1}$$

Por otro lado, el error de truncamiento (o tolerancia) se puede calcular como:

$$\left|R_{i,j-1} - R_{i,j}\right| \leq Tolerancia$$

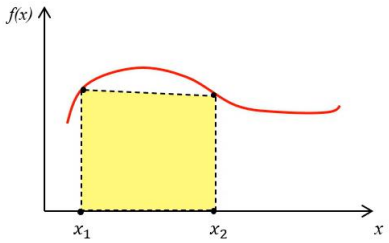
Algunos ejemplos son:

$$R_{i2} = \frac{4}{3} R_{i,1} - \frac{1}{3} R_{i-1,1}$$

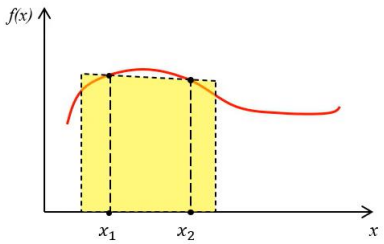
$$R_{i3} = \frac{8}{7} R_{i,2} - \frac{1}{7} R_{i-1,2}$$

Cuadratura de Gauss

Permite obtener un mejor resultado con el mismo esfuerzo de cálculo, lo logra definiendo los puntos consecutivos de forma tal que la sobreestimación se compense con la subestimación de la integral.



Selección de puntos equidistantes



Selección de puntos óptimos: cuadratura de Gauss

Para ello, se plantea una cuadratura con puntos y coeficientes de peso a definir:

$$I = \int_a^b f(x)dx \approx \sum_i c_i f_{(x_i)} = G \qquad (97)$$

A esta función G se la denomina **método de cuadratura de Gauss.** Con las siguientes condiciones:

- $x_1, x_2, \dots, x_n \in [a, b]$.
- $c_1, c_2, \dots, c_n$ deben ser tales que minimicen el error de truncamiento dado por $G - I$.

El criterio de selección de las $2n$ incógnitas $\{x_i, c_i\}$ es que integren en forma exacta el polinomio de mayor grado que se puede construir con 2n parámetros.

$$p(x) \in P_{2n-1}(x)$$

Asimismo, es importante aclarar que siempre se debe trabajar en el intervalo $[-1,1]$ por lo que si tenemos cualquier otro intervalo deberemos aplicar un cambio de variables. Una opción es:

$$x = \frac{(b-a)t + (b-a)}{2} \rightarrow dx = \frac{(b-a)}{2}dt \tag{98}$$

Donde x representa a la variable original perteneciente al intervalo distinto al unitario y t es la nueva variable en el último intervalo mencionado.

Ejemplo

Caso $n = 2$ y $[a, b] = [-1,1]$, entonces:

$$I = \int_{-1}^{1} f(x)dx \approx c_1 f_{(x_1)} + c_2 f_{(x_2)} = G_2$$

Dado que dijimos que debe ser **exacta** la aproximación, si $f(x) \in P_3(x)$ entonces:

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

Por lo que:

$$I = \int_{-1}^{1} [a_3x^3 + a_2x^2 + a_1x + a_0]dx = a_3 \int_{-1}^{1} x^3dx + a_2 \int_{-1}^{1} x^2dx + a_1 \int_{-1}^{1} xdx + a_0 \int_{-1}^{1} dx$$

Para que sea exacto, la integral debe ser exacta:

$$\begin{cases} f(x) = 1 \rightarrow \int_{-1}^{1} dx = 2 = c_1 + c_2 \\ f(x) = x \rightarrow \int_{-1}^{1} xdx = 0 = c_1x_1 + c_2x_2 \\ f(x) = x^2 \rightarrow \int_{-1}^{1} x^2dx = \frac{2}{3} = c_1{x_1}^2 + c_2{x_2}^2 \\ f(x) = x^3 \rightarrow \int_{-1}^{1} x^3dx = 0 = c_1{x_1}^3 + c_2{x_2}^3 \end{cases} \rightarrow \begin{cases} c_1 = c_2 = 1 \\ x_1 = \frac{1}{\sqrt{3}} \\ x_2 = -\frac{1}{\sqrt{3}} \end{cases}$$

Resultado:

$$G_2 = c_1 f_{(x_1)} + c_2 f_{(x_2)} = f_{\left(\frac{1}{\sqrt{3}}\right)} + f_{\left(\frac{-1}{\sqrt{3}}\right)}$$

Para un caso general

Los coeficientes se pueden obtener de la integral de la fórmula de Lagrange:

$$c_1 = \int_{-1}^{1} \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \right] dx \tag{99}$$

Los puntos de Gauss (x_i) se obtienen como las raíces de los polinomios de Legendre:

$$P_2(x) = \tfrac{1}{2}(3x^2 - 1)$$
$$P_3(x) = \tfrac{1}{2}(5x^3 - 3x)$$
$$P_4(x) = \tfrac{1}{8}(35x^4 - 30x^2 + 3)$$
$$P_5(x) = \tfrac{1}{8}(63x^5 - 70x^3 + 15x)$$
$$P_6(x) = \tfrac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5)$$
$$P_7(x) = \tfrac{1}{16}(429x^7 - 693x^5 + 315x^3 - 35x)$$

De todas formas, como los polinomios NO dependen de los puntos se encuentran tabulados en tablas $(c_1 = w_1)$:

n	w_i	x_i	n	w_i	x_i
1	2.0	0.0	8	0.1012285363	±0.9602898565
2	1.0	±0.5773502692		0.2223810345	±0.7966664774
3	0.5555555556	±0.7745966692		0.3137066459	±0.5255324099
	0.8888888889	0.0		0.3626837834	±0.1834346425
4	0.3478548451	±0.8611363116	9	0.0812743883	±0.9681602395
	0.6521451549	±0.3399810436		0.1806481607	±0.8360311073
5	0.2369268851	±0.9061798459		0.2606106964	±0.6133714327
	0.4786286705	±0.5384693101		0.3123470770	±0.3242534234
	0.5688888889	0.0		0.3302393550	0.0
6	0.1713244924	±0.9324695142	10	0.0666713443	±0.9739065285
	0.3607615730	±0.6612093865		0.1494513492	±0.8650633667
	0.4679139346	±0.2386191861		0.2190863625	±0.6794095683
7	0.1294849662	±0.9491079123		0.2692667193	±0.4333953941
	0.2797053915	±0.7415311856		0.2955242247	±0.1488743390
	0.3818300505	±0.4058451514			
	0.4179591837	0.0			

Diferenciación (Guía 7)

El problema tipo a resolver es:

$$Hallar\ una\ aproximación\ para\ la\ derivada\colon f^{(n)} = \frac{d^n f}{dx^n}$$

La solución que se propone es una derivada numérica de la forma:

$$f^{(n)} \approx \sum_i \omega_i f_{(x_i)}$$

Donde ω_i representa al factor de peso y x_i son los puntos donde se evalúa la función.

Fórmula de 2 puntos

La derivada primera está dada analíticamente por:

$$f'_{(x_0)} = \lim_{h\to 0} \frac{f_{(x_0+h)} - f_{(x_0)}}{h}$$

Supongamos que $x_0 \in [a,b]$ y $f_{(x)} \in C^2[a,b]$, con $x_1 = x_0 + h$ siendo h pequeño tal que $x_1 \in [a,b]$. Expandiendo los puntos x_0 y x_1 por el polinomio de Lagrange $P_{0,1}(x)$ obtenemos:

$$f_{(x)} = P_{0,1}(x) + \frac{(x-x_0)(x-x_1)}{2!} f''(\mu_{(x)}), \qquad con\ \mu_{(x)} \in [a,b]$$

$$f_{(x)} = f_{(x_0)} \frac{(x-x_1)}{(x_0-x_1)} + f_{(x_1)} \frac{(x-x_0)}{(x_1-x_0)} + \frac{(x-x_0)(x-x_1)}{2} f''(\mu_{(x)})$$

Reemplazando $x_1 = x_0 + h$:

$$f_{(x)} = f_{(x_0)} \frac{(x-x_0-h)}{x_0-x_0-h} + f_{(x_0+h)} \frac{(x-x_0)}{(x_0+h-x_0)} + \frac{(x-x_0)(x-x_0+h)}{2} f''(\mu_{(x)})$$

$$f_{(x)} = -\frac{f_{(x_0)}}{h}(x-x_0-h) + \frac{f_{(x_0+h)}}{h}(x-x_0) + \frac{x^2 - 2xx_0 + xh + {x_0}^2 - x_0 h}{2} f''(\mu_{(x)})$$

Derivando:

$$f'_{(x)} = -\frac{f_{(x_0)}}{h} + \frac{f_{(x_0+h)}}{h} + \left[x - x_0 + \frac{h}{2}\right] f''(\mu_{(x)}) + \frac{x^2 - 2xx_0 + xh + {x_0}^2 - x_0 h}{2} \frac{df''(\mu_{(x)})}{dx}$$

Evaluando en x_0:

$$f'_{(x)} = \frac{f_{(x_0+h)} - f_{(x_0)}}{h} + \frac{h}{2} f''(\mu_{(x_0)}) + \frac{0}{2} \frac{df''(\mu_{(x)})}{dx} = \frac{f_{(x_0+h)} - f_{(x_0)}}{h} + \frac{h}{2} f''(\mu_{(x_0)})$$

Entonces, para h suficientemente pequeño, la **fórmula de dos puntos** resulta:

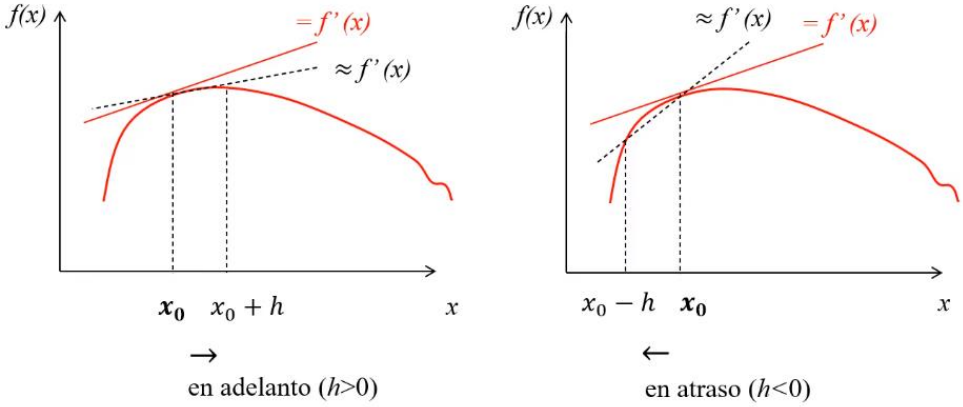
$$f'_{(x)} \approx \frac{f_{(x_0+h)} - f_{(x_0)}}{h} \tag{100}$$

Y el **error de truncamiento está acotado por** $\frac{M}{2}h$ donde M es tal que:

$$|f''(x)| \leq M\ \forall x \in [a,b]$$

Además, si $h > 0$ la derivada se calcula en adelanto y si $h < 0$ se está calculando en atraso.

Representación



Formula de N+1 puntos

Para mejorar la precisión se utiliza la interpolación de Lagrange:

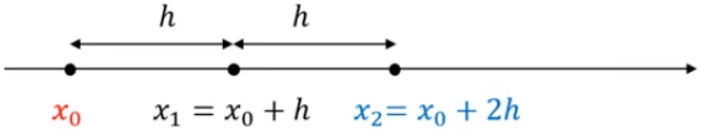
$$f_{(x)} = \sum_{k=0}^{N} f_{(x_k)} L_{Nk(x)} + \frac{(x-x_0)(x-x_1)\dots(x-x_N)}{(N+1)!} f^{(N+1)}_{(\gamma_{(x)})}$$

Con $\gamma_{(x)} \in [a, b]$. Derivando y evaluando la expresión en $x = x_j$ para algún $0 \leq j \leq N$:

$$f'_{(x_j)} = \sum_{k=0}^{N} f(x_k) L'_{Nk}(x) + \frac{f^{(N+1)}_{\left(\gamma_{(x_j)}\right)}}{(N+1)!} \prod_{\substack{k=0 \\ k \neq j}}^{N} (x_j - x_k) \qquad (101)$$

Donde el **primer término** corresponde a la **fórmula de N+1 puntos** y el segundo determina el error de truncamiento.

Para el caso de tres nodos equidistantes tenemos:



$$en\ adelanto: \quad f'_{(x_0)} = \frac{-f_{(x_2)} + 4f_{(x_1)} - 3f_{(x_0)}}{2h} + \frac{h^2}{3} f'''_{(\mu_0)}$$

$$centrada: \quad f'_{(x_1)} = \frac{f_{(x_2)} - f_{(x_0)}}{2h} - \frac{h^2}{6} f'''_{(\mu_1)}$$

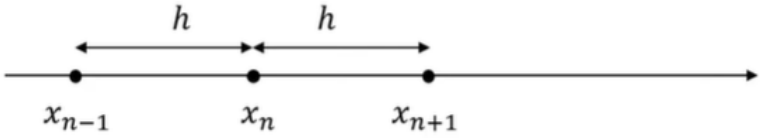
$$en\ atraso: \quad f'_{(x_2)} = \frac{3f_{(x_2)} - 4f_{(x_1)} + f_{(x_0)}}{2h} + \frac{h^2}{3} f'''_{(\mu_2)}$$

La notación para 3 nodos equidistantes (siendo x_0 el nodo central) es:

$$f'_{(x_0)} \approx \frac{f_{(x_0+h)} - f_{(x_0-h)}}{2h}$$

Método de los coeficientes indeterminados

Consiste en desarrollar $f_{(x)}$ por serie de Taylor alrededor del punto deseado y solicitar con una fórmula de N+1 puntos ciertos requisitos. Como es un método muy práctico se explica con un ejemplo.



Se busca la derivada primera con 3 nodos equidistantes de una función cualquiera. En forma genérica se quiere:

$$f'_n \approx af_{n-1} + bf_n + cf_{n+1} \tag{102}$$

Desarrollamos por Taylor alrededor del punto x_n:

$$f_{(x)} = f_{(x_n)} + \frac{f'_{(x_n)}}{1!}(x - x_n) + \frac{f''_{(x_n)}}{2!}(x - x_n)^2 + \frac{f'''_{(x_n)}}{3!}(x - x_n)^3 + O(h^4)$$

Evaluamos en $f_{(x_{n-1})} \equiv f_{n-1}$ y en $f_{(x_{n+1})} \equiv f_{n+1}$:

$$f_{(x_{n-1})} = f_{n-1} = f_n - hf'_n + \frac{h^2}{2}f''_n - \frac{h^3}{6}f'''_n + O(h^4)$$

$$f_{(x_{n+1})} = f_{n+1} = f_n + hf'_n + \frac{h^2}{2}f''_n + \frac{h^3}{6}f'''_n + O(h^4)$$

Reemplazando las expresiones en (102):

$$f'_n + R_D = a\left[f_n - hf'_n + \frac{h^2}{2}f''_n - \frac{h^3}{6}f'''_n + O(h^4)\right] + bf_n + c\left[f_n + hf'_n + \frac{h^2}{2}f''_n + \frac{h^3}{6}f'''_n + O(h^4)\right]$$

Reordenando:

$$f'_n + R_D = (a + b + c)f_n + (-a + c)hf'_n + (a + c)\frac{h^2}{2}f''_n + (-a + c)\frac{h^3}{6}f'''_n + (a + c)O(h^4)$$

Donde, para que ambos miembros sean iguales, deben cumplirse las siguientes condiciones:

$$\begin{cases} a + b + c = 0 \\ (-a + c)h = 1 \\ (a + c)\dfrac{h^2}{2} = 0 \end{cases} \rightarrow \begin{cases} -\dfrac{1}{2h} + b + \dfrac{1}{2h} = 0 \rightarrow b = 0 \\ (c + c)h = 1 \rightarrow c = \dfrac{1}{2h} \\ a = -c \end{cases} \rightarrow \begin{cases} a = -\dfrac{1}{2h} \\ b = 0 \\ c = \dfrac{1}{2h} \end{cases}$$

Finalmente:

$$\begin{cases} f'_n \approx -\dfrac{1}{2h}f_{n-1} + \dfrac{1}{2h}f_{n+1} \\ R_D = (-a + c)\dfrac{h^3}{6}f'''_n = \left(\dfrac{1}{2h} + \dfrac{1}{2h}\right)\dfrac{h^3}{6}f'''_n = \dfrac{h^2}{6}f'''_{(\mu)} \end{cases}$$

Regla Práctica

En general se cumple que:

$$OP = N - OD \tag{103}$$

Donde $\boldsymbol{N}$ indica la **cantidad de nodos** o puntos disponibles, $\boldsymbol{OD}$ es el **orden de derivación**, léase qué derivada se desea calcular, y $\boldsymbol{OP}$ indica el **orden de precisión** de la aproximación obtenida.

Excepción a la regla

Si la aproximación es centrada y la paridad (si es par o impar) de N es distinta de la paridad del OD, entonces se gana 1 orden de precisión:

$$OP = OP + 1$$

Esto tiene que ver con que cuando se hace el desarrollo de Taylor algunos términos se cancelan por ser una aproximación centrada.

Ejemplos:

$$\begin{cases} Primera\ derivada\ con\ 2\ nodos: N = 2, OD = 1 \rightarrow OP = (2-1) + 1 = 2 \\ Primera\ derivada\ con\ 3\ nodos: N = 3, OD = 1 \rightarrow OP = 3 - 1 = 2 \\ Segunda\ derivada\ con\ 3\ nodos: N = 3, OD = 2 \rightarrow OP = (3-2) + 1 = 2 \end{cases}$$

Extrapolación de Richardson

Si se cuenta con dos aproximaciones para la derivada de una función en un punto calculadas por el mismo método, es posible emplear la extrapolación de Richardson vista en el apartado de integración para obtener una mejor aproximación. Recordamos la fórmula (93):

$$I_{Rich} = I_{(h_1)} + \frac{I_{(h_1)} - I_{(h_2)}}{\left(\frac{h_2}{h_1}\right)^p - 1} + O(h^r)$$

Siendo h_i los pasos de cálculo tales que $h_1 < h_2$, p es el orden de convergencia del método por el cual se calcula la integral I y $r > p$ da cuenta del orden del error de truncamiento.

Problemas de Valores Iniciales (Guía 8)

Los problemas de valores iniciales están asociados a sistemas que evolucionan en el tiempo (para el caso de ingeniería).

Ecuación diferencial tipo

La ecuación a resolver se puede expresar genéricamente como:

$$y' = \frac{dy}{dt} = f_{(t,y)} \quad para\ t \geq 0 \tag{104}$$

Con $y_{(t=0)} = y_0$ representando al dato de la **condición inicial**. Este dato es necesario para hallar la respuesta de la ecuación (104), de aquí el nombre "problema de valores iniciales". La solución de (104) con la condición inicial es la función $y_{(t)}$ y es lo que buscamos aproximar.

Definiciones conceptuales

Se pueden identificar varios parámetros importantes de las ecuaciones:

- $y_{(t)}$ → Característica o propiedad del sistema.
- $\frac{dy}{dt} = f_{(t,y)}$ → Ley de evolución del sistema.
- $f_{(t,y)}$ → fuente.
- y_0 → estado de partida.
- t → parámetro de evolución del sistema.
- $t \geq 0$ → sentido de flujo de la información.

Condiciones de existencia y unicidad de la solución

Sea $D = \{(t,y)\ tal\ que\ t \geq 0\ , -\infty < y < \infty\}$:

- Si $f_{(t,y)}$ es continua en D → $\exists\ la\ solución\ y_{(t)}$.
- Si $\frac{\partial f_{(t,y)}}{\partial y}$ es continua en D → la solución es única.

Estabilidad del problema diferencial

Considerando el problema perturbado para $z(t)$:

$$\frac{dz}{dt} = f_{(t,z)} + \delta_{(t)} \quad para\ t \geq 0$$

$$z_{(t=0)} = y_0 + \varepsilon_0 \quad con\ |\varepsilon_0| < \varepsilon\ y\ \left|\delta_{(t)}\right| < \varepsilon$$

Entonces el problema diferencial es estable (bien planteado) si existe $k > 0$ tal que:

$$\left|z_{(t)} - y_0\right| < k\varepsilon$$

Problemas diferenciales en una dimensión del tipo PVI

Se pueden generalizar en tres tipos:

a) 1 ecuación de orden 1:

$$\frac{dy}{dt} = f_{(t,y)} \quad y_{(a)} = \alpha \quad a \leq t \leq b$$

b) n ecuaciones de orden 1:

$$\begin{cases} \frac{dy_1}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{1(a)} = \alpha_1 \\ \frac{dy_2}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{2(a)} = \alpha_2 \\ \vdots & \vdots \\ \frac{dy_n}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{n(a)} = \alpha_n \end{cases} \quad con\ a \leq t \leq b$$

c) 1 ecuaciones de orden n:

$$\frac{d^n y}{dt^n} = f_{(t,y,y',\cdots,y^{(n-1)})} \quad \begin{matrix} y_{(a)} = \alpha_1 \\ y'_{(a)} = \alpha_2 \\ \vdots \\ \vdots \\ y^{(n-1)}_{(a)} = \alpha_n \end{matrix} \quad con\ a \leq t \leq b$$

Es decir, preciso una condición inicial por cada derivada de la función hasta $n-1$. Para derivadas de orden n necesito n condiciones iniciales.

Cabe aclarar que mediante cambios de variables el último caso puede transformarse en el b).

Problema numérico

Proceso de discretización

El tiempo, al pasar al mundo real, deja de ser un continuo y pasa a estar representado por una cantidad discreta de valores t^n. Para indicar esto empleamos la siguiente notación:

$$t \to t^n \quad y \to y_{(t^n)} \approx u^n \tag{105}$$

En consecuencia, tenemos información solo en los instantes t^i pero no en los puntos intermedios entre dos instantes sucesivos t^i y t^{i+1}. Además, tampoco conocemos el valor exacto de $y_{(t^i)}$ sino que obtenemos una aproximación u^i. Por lo tanto, podemos calcular el error cometido por dicha aproximación como:

$$\left|u^n - y_{(t^n)}\right| = \delta \tag{106}$$

Se pueden ordenar los pasos para discretizar de la siguiente manera:

1) **Discretización del tiempo (variable independiente):**

$$t \to t^n \quad n = 0,1,2,\cdots \quad a \leq t \leq b: t^0 = a\ , t^N = b$$

Siendo $h = t^{(n+1)} - t^{(n)}$ el paso de tiempo. Si el paso es constante tenemos que:

$$t^n = a + nh \quad con\ h = \frac{b-a}{N}$$

2) **Discretización de la ecuación diferencial ordinaria (EDO):**

$$y \to y_{(t^n)} \approx u^n$$

En consecuencia, la derivada se puede calcular empleando, por ejemplo, la fórmula de dos puntos para la derivada primera (100):

$$\frac{dy}{dt} = f_{(t,y)} \quad \frac{dy}{dt} \to \frac{u^{n+1} - u^n}{h} \quad f_{(t,y)} \to f_{(t^n,u^n)}$$

3) Discretización de las condiciones iniciales:

En el estado inicial tenemos que $t^0 = a$, por lo tanto: $y_{(a)} = u^0 = \alpha$.

Métodos de PVI de paso simple

Euler explícito

El más sencillo es el **método de Euler explícito** cuya fórmula surge de despejar u^{n+1} de la fórmula de dos puntos para la derivada primera dada por (100). Su forma es:

$$u^{n+1} = u^n + hf_{(t^n,u^n)} \quad con\ u^0 = \alpha \tag{107}$$

Se observa que **el método se calcula en atraso** y resulta ser un **procedimiento de avance explícito** pues todos los datos necesarios para calcular u^{n+1} corresponden a los del paso anterior. De esta manera se logra avanzar obteniendo la solución numérica para cada paso de tiempo hasta $t^N = b$.

Errores

El **error global** indica que tan cerca se encuentra el valor obtenido del real. Se calcula como la diferencia entre la aproximación numérica y la solución teórica en un mismo estado temporal:

$$E^n = u^n - y_{(t^n)} \tag{108}$$

Por otro lado, el **error de truncamiento** es el error de discretización de la EDO y se puede calcular como la diferencia entre utilizar el operador discreto y evaluar la función fuente en el estado temporal n fijo, léase:

$$e^{n+1} = \{operador\ discreto\} - f_{\left(t^n,y_{(t^n)}\right)} \tag{109}$$

En otras palabras, es la diferencia entre haber utilizado y' y el operador discreto que la representa. Si en (109) reemplazáramos como operador discreto y' el error sería nulo y recuperaríamos la ecuación diferencial original (104).

A partir de esta expresión se puede determinar el orden del método, es decir de que orden es el error de truncamiento del mismo. A continuación, se desarrolla para el caso del método de Euler (107):

Primero se plantea $y_{(t^{n+1})}$ como desarrollo de Taylor hasta algún orden mayor al del método:

$$y_{(t^{n+1})} = y_{(t^n)} + hy'_{(t^n)} + \frac{h^2}{2}y''_{(t^n)} + O(h^3)$$

Como se cumple la EDO: $y'_{(t^n)} = f_{\left(t^n,y_{(t^n)}\right)}$ podemos reemplazar:

$$y_{(t^{n+1})} = y_{(t^n)} + hf_{\left(t^n,y_{(t^n)}\right)} + \frac{h^2}{2}y''_{(t^n)} + O(h^3)$$

Reordenamos:

$$\frac{y_{(t^{n+1})} - y_{(t^n)}}{h} = f_{\left(t^n,y_{(t^n)}\right)} + \frac{h}{2}y''_{(t^n)} + O(h^2)$$

Observamos que el miembro izquierdo presenta una aproximación de la derivada primera $y'_{(t^n)} = f_{\left(t^n, y_{(t^n)}\right)}$ de igual forma que el método de Euler:

$$u^{n+1} = u^n + hf_{(t^n,u^n)} \rightarrow f_{(t^n,u^n)} = \frac{u^{n+1} - u^n}{h} \approx f_{\left(t^n, y_{(t^n)}\right)}$$

Por lo tanto, podemos aproximar el miembro izquierdo por $f_{(t^n,u^n)}$ y de esta manera la diferencia entre la aproximación y el valor exacto dará lugar al error de truncamiento. Es decir:

$$e^{n+1} = \{operador\ discreto\} - f_{\left(t^n, y_{(t^n)}\right)} = \frac{u^{n+1} - u^n}{h} - f_{\left(t^n, y_{(t^n)}\right)}$$

Reemplazando en la fórmula:

$$\frac{u^{n+1} - u^n}{h} - f_{\left(t^n, y_{(t^n)}\right)} = \frac{h}{2} y''_{(t^n)} + O(h^2)$$

$$e^{n+1} = \frac{h}{2} y''_{(t^n)} + O(h^2)$$

Entonces, si existe un M tal que $|y''| \leq M$ podemos despreciar los términos de mayor orden y decir que:

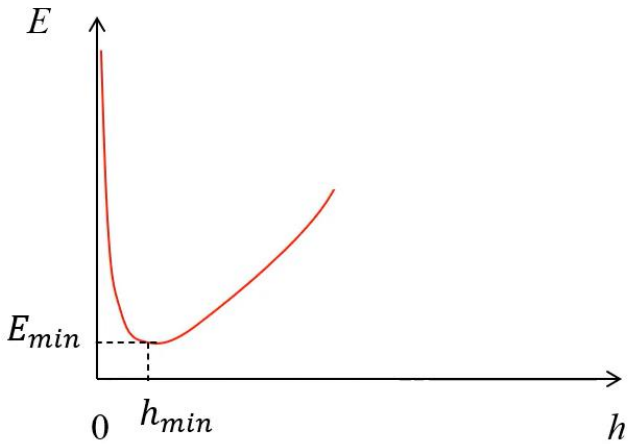
$$e^{n+1} \sim \frac{h}{2} M$$

De esta manera concluimos en que el **método de Euler es de orden 1,** pues su error de truncamiento es de primer orden, dado que el paso se encuentra elevado a la 1.

Error de redondeo

Si δ indica el error de redondeo y $|y''| \leq M$, el error global puede minimizarse para:

$$h_{min} \approx \sqrt{\frac{2\delta}{M}} \qquad (110)$$



Análisis de Convergencia de un método

Consistencia

Un método de discretización es consistente si el error de truncamiento tiende a cero para $h \to 0$.

$$\lim_{h\to 0} e^{n+1} = 0 \tag{111}$$

En otras palabras, en (109) el operador discreto debe tender a la derivada que representa cuando $h \to 0$.

Estabilidad

Para analizar la **estabilidad** empleamos el **método de Von Neumann** el cual consiste en introducir una pequeña perturbación δ en el estado temporal n y observar como se transforma en el estado temporal siguiente $n+1$. Para el método de Euler tenemos que:

$$u^{n+1} = u^n + hf_{(t^n,u^n)}$$

$$(u^{n+1} + \delta u^{n+1}) = (u^n + \delta u^n) + hf_{(t^n,u^n)}$$

Reordenamos:

$$u^{n+1} + \delta u^{n+1} = u^n + hf_{(t^n,u^n)} + \delta u^n$$

Observamos que los términos en rojo son iguales por lo que se pueden cancelar y tenemos que:

$$\delta u^{n+1} = \delta u^n \to 1 = \frac{\delta u^{n+1}}{\delta u^n} = g^n$$

Definimos el **factor de amplificación g^n** como:

$$g^n \equiv \frac{\delta u^{n+1}}{\delta u^n} \tag{112}$$

Tenemos que:

- Si $|g^n| > 1$ → el método es **inestable**, pues las perturbaciones están creciendo de un paso al siguiente.
- Si $|g^n| = 1$ → el método es **marginalmente estable**, pues las perturbaciones se mantienen constantes de un paso al siguiente.
- Si $|g^n| < 1$ → el método es **estable**, pues las perturbaciones están disminuyendo de un paso al siguiente.

De esta manera podemos concluir que **el método de Euler es marginalmente estable.**

En el caso de contar con un sistema de ecuaciones n diferenciales de orden 1, el mismo se puede escribir de forma matricial. Para analizar la estabilidad del problema se introduce la perturbación y luego de simplificar se debe obtener un sistema de la forma:

$$\begin{bmatrix} \delta u_{n+1} \\ \delta v_{n+1} \\ \delta w_{n+1} \end{bmatrix} = \overline{\overline{G^n}} \begin{bmatrix} \delta u_n \\ \delta v_n \\ \delta w_n \end{bmatrix}$$

Donde la matriz $\overline{\overline{G^n}}$ se denomina **matriz de amplificación** y para saber si el problema es estable es necesario pedir que el radio espectral de la matriz sea menor a 1, léase:

$$\rho\left(\overline{\overline{G^n}}\right) < 1$$

De aquí se pueden despejar las condiciones que debe cumplir el paso h.

En el caso de que los autovalores sean número complejos, significa que nos encontramos frente a una solución oscilatoria y se pueden presentar tres casos:

- $|\lambda| > 1$ → la solución es **amplificada**, crece el error a medida que se avanza en las iteraciones.
- $|\lambda| > 1$ → la solución es **conservativa**, el error se mantiene constante a medida que se avanza en las iteraciones.
- $|\lambda| < 1$ → la solución es **amortiguada**, disminuye el error a medida que se avanza en las iteraciones.

Convergencia

Un método es **convergente** si su error global tiende a cero para $h \to 0$:

$$\lim_{\substack{h\to 0 \\ t^n\ fijo}} u^n = y_{(t^n)}$$

Como probar esto es muy complejo, hacemos uso del **Teorema de Lax** el cual establece que si un método resulta consistente y estable será también convergente:

$$Consistente + Estable = Convergente \tag{113}$$

Respecto al **método de Euler**, como resultó ser estable y consistente podemos afirmar que también **es un método convergente**.

Más métodos de paso simple

Método de Euler Inverso o Implícito

Surge de plantear la derivada con dos puntos en adelanto:

$$u^{n+1} = u^n + hf_{(t^{n+1},u^{n+1})} \tag{114}$$

Como se observa es un método fuertemente implícito pues **evalúa la fuente en adelanto** (t^{n+1}, u^{n+1}). Por ende, pierde la propiedad explícita y resulta incondicionalmente convergente, es decir, no existe un $h_{máx}$. **Es un método de primer orden** (se obtiene a partir de una derivada de 2 puntos). La desventaja que presentan los métodos implícitos es que si no son lineales su resolución requiere emplear un método para ENL.

Método implícito ponderado

Surge de combinar Euler explícito y Euler implícito afectando a ambos por un factor de peso β:

$$u^{n+1} = u^n + h\left[\beta f_{(t^{n+1},u^{n+1})} + (1-\beta) f_{(t^n,u^n)}\right] \tag{115}$$

Siendo $0 \leq \beta \leq 1$ el denominado **parámetro de ponderación**. Si $\beta = 0$ se obtiene el método de Euler explícito, mientras que si $\beta = 1$ se consigue el método de Euler implícito. **Es un método de orden 1,** pero resulta incondicionalmente convergente si $\beta \geq \frac{1}{2}$.

Método de Crank-Nicholson

Es el caso particular del método implícito ponderado (115) con $\beta = \frac{1}{2}$:

$$u^{n+1} = u^n + \frac{h}{2}\left[f_{(t^{n+1},u^{n+1})} + f_{(t^n,u^n)}\right] \tag{116}$$

Es un método de orden 2, resulta **incondicionalmente estable** por ser implícito y, por este último motivo implica la resolución de un problema algebraico NO lineal (ENL) para cada paso de cálculo. En consecuencia, **implica un mayor costo computacional**. No obstante, dependiendo de la expresión del término fuente, puede darse el caso de que sea posible despejar u^{n+1} y así evitar emplear un método de resolución de ENL.

Métodos de Runge-Kutta (RK)

Existen varios métodos de Runge-Kutta, los cuales **son métodos explícitos**. El orden de estos se indica con un número seguido del acrónimo, por ejemplo, RK-2 indica un método Runge-Kutta de orden 2. Algunos son:

- **Método Predictor-Corrector explícito o del Punto medio**

$$\begin{cases} u^{n+\frac{1}{2}} = u^n + \frac{h}{2} f_{(t^n,u^n)} \\ u^{n+1} = u^n + h f_{\left(t^{n+\frac{1}{2}},u^{n+\frac{1}{2}}\right)} \end{cases} \tag{117}$$

Presenta dos etapas, la primera se denomina **Predictor** y calcula un valor intermedio de la incógnita u^{n+1} empleando el método Euler explícito con un paso $\frac{h}{2}$. Luego, se utiliza este dato en la segunda etapa denominada **Corrector** para hallar el valor de la incógnita en t^{n+1}. El valor calculado en el paso predictor NO forma parte de la solución. **Es un método de orden 2**, por lo tanto, se nota como **RK-2.**

- **Método Euler modificado**

Es un método de Runge-Kutta de orden 2.

$$\begin{cases} u_*^{n+1} = u^n + h f_{(t^n,u^n)} \\ u^{n+1} = u^n + \frac{h}{2}\left[f_{(t_*^{n+1},u_*^{n+1})} + f_{(t^n,u^n)}\right] \end{cases} \tag{118}$$

- **Método RK-4**

Presenta dos etapas y consta de 4 cálculos totales (3 predictores y 1 corrector) para hallar la incógnita en t^{n+1}:

$$Predictor \rightarrow \begin{cases} u_*^{n+\frac{1}{2}} = u^n + \frac{h}{2} f_{(t^n,u^n)} \\ u_{**}^{n+\frac{1}{2}} = u^n + \frac{h}{2} f_{\left(t^{n+\frac{1}{2}},u_*^{n+\frac{1}{2}}\right)} \\ u_*^{n+1} = u^n + h f_{\left(t^{n+\frac{1}{2}},u_{**}^{n+\frac{1}{2}}\right)} \end{cases} \tag{119}$$

$$Corrector \rightarrow u^{n+1} = u^n + \frac{h}{6}\left[f_{(t^n,u^n)} + 2u_*^{n+\frac{1}{2}} + 2u_{**}^{n+\frac{1}{2}} + f_{(t^n,u_*^{n+1})}\right]$$

Otra forma de expresarlo es:

$$Predictor \rightarrow \begin{cases} q_1 = hf_{(t^n, u^n)} \\ q_2 = hf_{\left(t^{n+\frac{1}{2}},\ u_n + \frac{1}{2}q_1\right)} \\ q_3 = hf_{\left(t^{n+\frac{1}{2}},\ u_n + \frac{1}{2}q_2\right)} \\ q_4 = hf_{(t^{n+1},\ u_n + q_3)} \end{cases}$$

$$Corrector \rightarrow u^{n+1} = u^n + \frac{1}{6}[q_1 + 2q_2 + 2q_3 + q_4]$$

Es un método de orden 4 y es uno de los más usados debido a su relación costo/beneficio.

Desempeño de los métodos

El esfuerzo de cálculo para los métodos de Euler y RK es de la forma:

$$\begin{cases} 1\ evaluación\ de\ f\ en\ Euler \rightarrow O(1) \\ 2\ evaluaciones\ de\ f\ en\ RK-2 \rightarrow O(2) \\ 4\ evaluaciones\ de\ f\ en\ RK-4 \rightarrow O(4) \end{cases}$$

Si igualamos el esfuerzo de cálculo de cada método para avanzar de un estado n hasta $n+1$, es decir ajustamos el paso para cada uno tal que se realicen la misma cantidad de cálculos en cada método, tenemos que:

$$Precisión: \quad RK-4(h) > RK-2\left(\frac{h}{2}\right) > Euler\left(\frac{h}{4}\right)$$

En las tres situaciones el esfuerzo computacional es el mismo. No obstante, el error de truncamiento es muy superior en el caso de RK-4 con un paso h:

$$Orden\ de\ truncamiento: \quad O(h^4) \ll O\left(\frac{h^2}{4}\right) \ll O\left(\frac{h}{4}\right)$$

Entonces podemos concluir que **el método óptimo es el RK-4.**

Métodos de PVI de paso múltiple

Hasta ahora se vieron aproximaciones donde el estado $n+1$ se determina a partir de información de ese mismo estado y/o del estado previo n. Como el error global (108) crece con n, es posible utilizar información de estados previos para ganar precisión.

Fórmulas generales

Podemos expresar los métodos lineales en general según la fórmula:

$$F(t^n, u^{n+r}, \cdots, u^n, h, f) = \sum_{i=0}^{r} b_i f_{(t^{n+i}, u^{n+i})} \qquad (120)$$

Considerando $r \geq 1$ estados consecutivos $(n, n+1, n+2, \cdots, n+r)$.

Para los métodos de paso múltiple tenemos la siguiente regla general:

$$u^{n+r} + \sum_{i=0}^{r-1} a_i u^{n+i} = h \sum_{i=0}^{r} b_i f_{(t^{n+i}, u^{n+i})} \tag{121}$$

Donde si $b_r = 0$ el método resulta explícito (predictor) y si $b_r \neq 0$ el método resulta implícito (corrector). Cada método queda definido por el conjunto $a_0, a_1, \cdots, a_{r-1}$ y $b_0, b_1, \cdots, b_{r-1}$.

Si $r = 1$ se obtienen los métodos de paso simple, por ejemplo:

$$u^{n+1} + a_0 u^n = h\left[b_0 f_{(t^n, u^n)} + b_1 f_{(t^{n+1}, u^{n+1})}\right]$$

Para Euler explícito:

$$a_0 = -1 \quad b_0 = 1 \quad b_1 = 0\} \rightarrow u^{n+1} - u^n = h\left[f_{(t^n, u^n)} + 0 f_{(t^{n+1}, u^{n+1})}\right]$$

$$u^{n+1} = u^n + h f_{(t^n, u^n)}$$

Para Euler implícito:

$$a_0 = -1 \quad b_0 = 0 \quad b_1 = 1\} \rightarrow u^{n+1} - u^n = h\left[0 f_{(t^n, u^n)} + f_{(t^{n+1}, u^{n+1})}\right]$$

$$u^{n+1} = u^n + h f_{(t^{n+1}, u^{n+1})}$$

Para Crank-Nicholson:

$$a_0 = -1 \quad b_0 = \frac{1}{2} \quad b_1 = \frac{1}{2}\Big\} \rightarrow u^{n+1} - u^n = h\left[\frac{1}{2} f_{(t^n, u^n)} + \frac{1}{2} f_{(t^{n+1}, u^{n+1})}\right]$$

$$u^{n+1} = u^n + \frac{h}{2}\left[f_{(t^n, u^n)} + f_{(t^{n+1}, u^{n+1})}\right]$$

Obtención de los métodos de paso múltiple

Desarrollamos el problema tipo PVI:

$$\frac{dy}{dt} = f(t, y) \rightarrow dy = f(t, y)dt \quad (1)$$

$$\int_{y(t^n)}^{y(t^{n+1})} dy = \int_{t^n}^{t^{n+1}} f(t, y)dt \rightarrow y_{(t^{n+1})} - y_{(t^n)} = \int_{t^n}^{t^{n+1}} f(t, y)dt$$

Entonces:

$$y_{(t^{n+1})} = y_{(t^n)} + \int_{t^n}^{t^{n+1}} f(t, y)dt$$

Sin embargo, esta ecuación no se puede resolver sin conocer $y_{(t)}$, que es la solución de (1). En su lugar se aplica el polinomio interpolante $P(t)$ que usa los valores previos u^{n-1}, u^{n-2}, etc. Quedando entonces:

$$u^{n+1} \approx u^n + \int_{t^n}^{t^{n+1}} P(t)dt \tag{122}$$

Métodos explícitos de m pasos: Adams-Bashforth

Empleando la fórmula de **interpolación de Newton** a orden $m-1$:

$$f(t,y) = P_{m-1}(t) + \frac{f^m(\mu, y_{(\mu)})}{m!}(t-t^n)(t-t^{n-1})\cdots(t-t^{n+1-m}) \tag{123}$$

Donde $P_{m-1}(t)$ interpola los puntos $(t^n, f^n), (t^{n-1}, f^{n-1}), \cdots (t^{n+1-m}, f^{n+1-m})$ con la notación usual $f^n = f(t^n, u^n)$. El polinomio $P_{m-1}(t)$ depende de t, lo que permite resolver la integral (122). Con ello es posible obtener la **fórmula explícita** de cada método una vez definido el valor de m. Además, el orden del método surge de la aproximación polinomial, por lo que de la expresión presentada se ve que **el método es de orden $\boldsymbol{m}$**, también descripto como $O(h^m)$.

Definimos entonces algunos **métodos de Adams-Bashforth (AB):**

- **AB de 2 pasos:** $O(h^2) = \frac{5}{12}h^2y'''$

$$u^{n+1} = u^n + \frac{h}{2}[3f^n - f^{n-1}]$$

- **AB de 3 pasos:** $O(h^3) = \frac{3}{18}h^3y^{iv}$

$$u^{n+1} = u^n + \frac{h}{12}[23f^n - 16f^{n-1} + 5f^{n-2}]$$

- **AB de 4 pasos:** $O(h^4) = \frac{251}{720}h^4y^{v}$

$$u^{n+1} = u^n + \frac{h}{24}[55f^n - 59f^{n-1} + 37f^{n-2} - 9f^{n-3}]$$

- **AB de 5 pasos:** $O(h^5) = \frac{95}{288}h^5y^{vi}$

$$u^{n+1} = u^n + \frac{h}{720}[1901f^n - 2774f^{n-1} + 2616f^{n-2} - 1274f^{n-3} + 251f^{n-4}]$$

Métodos implícitos de m pasos: Adams-Moulton

Utilizando un punto adicional en el estado temporal "actual" $n+1$ en la fórmula de interpolación de Newton se gana 1 orden y la fórmula resulta **implícita**:

$$f(t,y) = P_{m-1}(t) + \frac{f^{m+1}(\mu, y_{(\mu)})}{m!}(t-t^n)(t-t^{n-1})\cdots(t-t^{n+2-m}) \tag{124}$$

Siendo (t^{n+2-m}, f^{n+2-m}) el punto agregado. En este caso, **el método es de orden $\boldsymbol{m+1}$** ($O(h^{m+1})$).

Definimos entonces algunos **métodos de Adams-Moulton (AM):**

- **AM de 2 pasos:** $O(h^3) = \frac{-1}{24}h^3y^{iv}$

$$u^{n+1} = u^n + \frac{h}{12}[5f^{n+1} + 8f^n - 5f^{n-1}]$$

- **AM de 3 pasos:** $O(h^4) = \frac{-19}{172}h^4y^{v}$

$$u^{n+1} = u^n + \frac{h}{24}[9f^{n+1} + 19f^n - 5f^{n-1} + f^{n-2}]$$

- **AM de 4 pasos:** $O(h^5) = \frac{-3}{160} h^5 y^{vi}$

$$u^{n+1} = u^n + \frac{h}{720}[251f^{n+1} + 646f^n - 264f^{n-1} + 106f^{n-2} - 19f^{n-3}]$$

Propiedades de los métodos multipaso

Un método de m pasos requiere m valores de arranque, de lo cuales solo 1 está dado por la condición inicial del problema diferencial. Los restantes $m-1$ se obtienen con métodos de paso simple de igual orden que el multipaso, esto es para no perder la precisión que aporta el multipaso.

Para evaluar la estabilidad del método cada nivel temporal se asocia a una variable perturbada distinta y se analiza como un sistema de ecuaciones.

Sistemas de ecuaciones

Recordamos el caso b) de PVI donde tenemos n ecuaciones de orden 1:

$$\begin{cases} \frac{dy_1}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{1(a)} = \alpha_1 \\ \frac{dy_2}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{2(a)} = \alpha_2 \\ \quad\vdots & \quad\vdots \\ \frac{dy_n}{dt} = f_{(t,y_1,\cdots,y_n)} & y_{n(a)} = \alpha_n \end{cases} \quad con\ a \le t \le b$$

Si el problema es lineal, léase si las incógnitas $y_1 \quad \cdots \quad y_n$ aparecen en forma lineal en $f_1 \quad \cdots \quad f_n$, podemos escribir el sistema de forma más compacta como:

$$\frac{dy}{dt} = \bar{\bar{A}}\,\bar{y} \quad con\ \bar{y}^T = (y_1 \quad \cdots \quad y_n)$$

La expresión resulta **estable** si:

$$Re(\rho_j) < 0 \tag{125}$$

Donde ρ_j representa los autovalores de la matriz $\bar{\bar{A}}$.

Método de discretización

Para cada incógnita del sistema se utiliza una variable discreta y se aplican los métodos previamente estudiados para el caso de 1 ecuación de orden 1.

$$y_i(t^n) \approx u_i^n$$

Estabilidad

El problema numérico es estable si, realizando el análisis de perturbaciones, la matriz de amplificación cumple con la siguiente condición para sus autovalores γ_j:

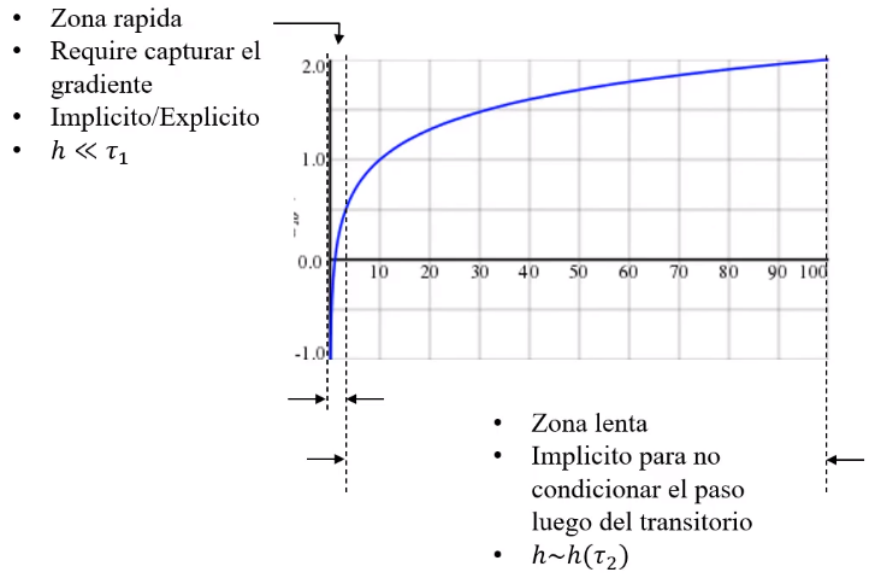
$$|\gamma_j| < 1\ \forall j \tag{126}$$

Es decir, el radio espectral debe ser menor a 1.

Para el caso de los métodos multipaso, como el análisis de estabilidad requiere tomar valores en t^{n+1} y t^n, es necesario realizar un cambio de variables de estado temporal para aquellas evaluaciones en t^{n-i} con $i = 1,2,\ldots..$

Ecuaciones Rígidas

Las **ecuaciones rígidas** aparecen en aquellos problemas donde coexisten escalas de tiempo (τ) de evolución muy diferentes: $\tau_1 \ll \tau_2$



Ejemplos de estos sistemas son los circuitos, la cinética química, resortes y amortiguadores. Los **sistemas rígidos de ecuaciones diferenciales** poseen soluciones del tipo:

$$e^{\alpha t} \quad con\ \alpha \in C\ tal\ que\ Re\{\alpha\} < 0$$

Por lo que presenta problemas al resolverlas numéricamente. El caso típico se ilustra como:

$$c_1 y'' + c_2 y' + c_3 y = f(t, y, y') \quad con\ |c_2| \gg |c_1|$$

Estos problemas de rigidez tienen lugar cuando el coeficiente que acompaña a la derivada primera es mucho mayor en módulo que el que acompaña a la derivada segunda.

PVI Conservativos (Guía 10)

Para estos problemas particulares se buscan soluciones numéricas donde se quiere preservar la característica de conservación de una propiedad importante del sistema. Por ejemplo:

- Conservación de la energía.
- Conservación de la masa.
- Conservación de la cantidad de movimiento.

El método RK-2 no es apto para describir este tipo de sistemas por lo que se cuenta con métodos que brindan mayor precisión.

Método de Nystrom

Surge de **discretizar la EDO a orden 2** con una aproximación centrada para la derivada segunda según:

$$u'' \approx \frac{u^{n+1} - 2u^n + u^{n-1}}{h^2}$$

Es un método explícito de paso múltiple. Por ejemplo, para el caso de un oscilador armónico tendríamos:

$$Ec.de\ movimiento \rightarrow u'' + \omega^2 u = 0$$

Discretizamos:

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{h^2} + \omega^2 u^n = 0$$

Definimos $\rho^2 = h^2\omega^2$ y reescribimos:

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{h^2} + \frac{\rho^2}{h^2}u^n = \frac{u^{n+1} - 2u^n + u^{n-1} + \rho^2 u^n}{h^2} = 0$$

$$u^{n+1} = (2 - \rho^2)u^n - u^{n-1}$$

Para analizar la solución numérica proponemos $u^n \sim \gamma^n$ con $\gamma \in C$. Reemplazamos en la ecuación:

$$\gamma^{n+1} = (2 - \rho^2)\gamma^n - \gamma^{n-1}$$

Como $\gamma \neq 0$ no es solución, podemos sacar factor común γ^{n-1}:

$$\gamma^{n-1}\gamma^2 - (2 - \rho^2)\gamma^{n-1}\gamma + \gamma^{n-1} = 0$$

$$\gamma^2 - (2 - \rho^2)\gamma + 1 = 0$$

Planteando la resolvente tenemos que:

$$\gamma_{1,2} = \left(1 - \frac{\rho^2}{2}\right) \pm \rho\sqrt{\frac{\rho^2}{4} - 1}$$

Para que la solución sea oscilatoria debe asegurarse que γ pertenece a los complejos, por ende:

$$\frac{\rho^2}{4} - 1 < 0 \rightarrow \rho^2 < 4 \rightarrow \rho < 2 \rightarrow h\omega < 2 \rightarrow h < \frac{2}{\omega}$$

Además, para que sea conservativo debe comprobarse que $|\gamma| = 1$.

Entonces:

- $\gamma_{1,2} = \left(1 - \frac{\rho^2}{2}\right) \pm i\rho\sqrt{1 - \frac{\rho^2}{4}} \in C$ → Oscila ✓
- $|\gamma| = 1$ → γ^n solo rota, por lo que es conservativo ✓

$$|\gamma| = \gamma_1\gamma_2 = \sqrt{\left(1 - \frac{\rho^2}{2}\right)^2 + \rho^2\sqrt{1 - \frac{\rho^2}{4}}^2} = \sqrt{\frac{\rho^4}{4} - \rho^2 + 1 + \rho^2 - \frac{\rho^4}{4}} = \sqrt{1} = 1$$

Método de Newmark

Es un método implícito de paso simple. Su fórmula es:

$$u_{n+1} = u_n + hu'_n + \frac{h^2}{4}(u''_n + u''_{n+1})$$

$$u'_{n+1} = u'_n + \frac{h}{2}(u''_n + u''_{n+1})$$

Problemas de Valores de Contorno (Guía 9)

A diferencia de los PVI que avanzan en una sola dirección (temporal), para los problemas de valores de contorno o PVC nos interesa conocer cómo se comporta el sistema entre dos valores extremos establecidos. Además, generalmente nos es indiferente la variable temporal para este tipo de problemas.

Problema diferencial tipo

El problema está dado por la ecuación:

$$y'' = \frac{d^2y}{dx} = f_{(x,y,y')} \quad para\ a \leq x \leq b \tag{127}$$

Junto con las condiciones de contorno:

$$y_{(x=a)} = \alpha \quad y_{(x=b)} = \beta$$

De manera que la solución de la ecuación empleando las condiciones de contorno es $y_{(x)}$.

Definiciones conceptuales

- $y_{(x)}$ → Característica o propiedad del sistema que queremos encontrar.
- $\frac{d^2y}{dx} = f_{(x,y,y')}$ → Ley de distribución.
- $f_{(x,y,y')}$ → fuente.
- α, β → condiciones de contorno.
- x → parámetro sobre cual el sistema cambia (en general es el espacio).
- $a \leq x \leq b$ → intervalo de flujo de la información.

Condiciones de existencia y unicidad de la solución

Sea $f_{(x,y,y')}$ continua en $D = \{(x, y, y')\ tal\ que\ a \leq x \leq b\ , -\infty < y < \infty, -\infty < y' < \infty\}$ y $\frac{\partial f}{\partial y}, \frac{\partial f}{\partial y'}$ continuas en D.

Si además:

- $\frac{\partial f}{\partial y}(x, y, y') > 0\ \forall (x, y, y') \in D$
- $\exists M = cte\ tal\ que \left|\frac{\partial f}{\partial y}\right| \leq M\ \forall (x, y, y') \in D$

Entonces, el problema tiene solución única.

Proceso de discretización

Se pueden ordenar los pasos para discretizar de la siguiente manera:

1) **Discretización del espacio:**

$$x \to x_i \quad con\ i = 0,1, \cdots, N$$

$$a \leq x_i \leq b \quad con\ x_0 = a\ , x_N = b$$

Siendo $\Delta x = x_{i+1} - x_i$ el **paso espacial**. Si el paso es constante tenemos que:

$$x_i = a + i\Delta x \quad con\ \Delta x = h = k = \frac{b-a}{N}$$

En general $\Delta x \neq cte$.

2) Discretización de la ecuación diferencial ordinaria (EDO):

$$u_i \approx y_{(x_i)}$$

La derivada segunda se calcula empleando la fórmula de tres puntos para la derivada segunda:

$$y'' \rightarrow \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \quad (centrada\ de\ orden\ 2) \tag{128}$$

Para el término fuente (derivada primera) tenemos tres opciones:

$$y' \rightarrow \frac{u_i - u_{i-1}}{\Delta x} \quad (en\ atraso\ O(1))$$

$$y' \rightarrow \frac{u_{i+1} - u_i}{\Delta x} \quad (en\ adelanto\ O(1))$$

$$y' \rightarrow \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad (centrada\ O(2))$$

Es importante mantener el orden de la derivada elegido durante todo el cálculo.

3) Discretización de las condiciones de contorno (CC):

La forma más general de éstas es:

$$c_1 y'_{(a)} + c_2 y_{(a)} = \alpha \quad d_1 y'_{(b)} + d_2 y_{(b)} = \beta$$

De ellas se desprenden los siguientes casos particulares:

- Si $c_1 = d_1 = 0$ → CC tipo **Dirichlet**.
- Si $c_2 = 0\ o\ d_2 = 0$ → CC tipo **Neuman**.
- Si las constantes son distintas de cero (caso genérico) → CC tipo **Riemann**.

Ejemplo: Ecuación del calor

A partir de aquí aplicaremos la metodología de discretización antes presentada denominada **método de diferencia finitas.**

Condiciones de Contorno tipo Dirichlet:

$$\frac{d^2T}{dx^2} = \cos^2(x) \quad 0 \leq x \leq \pi \quad T_{(0)} = T_a\ ; T_{(\pi)} = T_b$$

Discretización:

1) Espacio:

$$\Delta x = cte \quad x_i = i\Delta x \quad con\ i = 0,1,\cdots,N \quad \Delta x = \frac{\pi}{N}$$

$$a \leq x_i \leq b \quad con\ x_0 = a\ , x_N = b$$

2) EDO:

$$u_i \approx T_{(x_i)} \quad \frac{d^2T}{dx^2} \rightarrow \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \quad ; \quad \cos^2(x) \rightarrow \cos^2(x_i)$$

3) Condiciones de contorno (CC):

$$i = 0 \rightarrow u_0 = T_a \quad i = N \rightarrow u_N = T_b$$

Luego, se tiene que:

$$\begin{cases} u_0 = T_a & i = 0 \\ u_{i+1} - 2u_i + u_{i-1} = \Delta x^2 \cos^2(x_i) & i = 1, \cdots, N-1 \\ u_N = T_b & i = N \end{cases}$$

O en forma matricial:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} T_a \\ \Delta x^2 \cos^2(x_1) \\ \Delta x^2 \cos^2(x_2) \\ \vdots \\ \Delta x^2 \cos^2(x_{N-2}) \\ \Delta x^2 \cos^2(x_{N-1}) \\ T_b \end{bmatrix}$$

Se observa que la matriz es tridiagonal. Para resolver el sistema se aplica un método de resolución de SEL adecuado. Se resuelven conjuntamente todos los puntos.

Condiciones de Contorno tipo Neuman:

$$\frac{d^2T}{dx^2} = \cos^2(x) \quad 0 \leq x \leq \pi \quad T_{(0)} = T_a \,; T'_{(\pi)} = 0$$

Discretización:

1) **Espacio:** Idem anterior.
2) **EDO:** Idem anterior.
3) **Condiciones de contorno (CC):**

$$i = 0 \rightarrow u_0 = T_a$$

Para mantener el orden 2 en $i = N$ haremos uso de la derivada primera centrada:

$$i = N \rightarrow T'_{(\pi)} = \frac{u_{N+1} - u_{N-1}}{2\Delta x} = 0$$

Sin embargo, contamos con un **nodo fantasma u_{N+1}** (pues se sale del entorno que estamos analizando y no tenemos información de él) por lo tanto debemos eliminarlo. Para ello, evaluamos la ley de distribución en el extremo $\cos^2(x_N) = \cos^2(\pi) = 1$:

$$\frac{u_{N+1} - u_{N-1}}{2\Delta x} = 0 \rightarrow u_{N+1} = u_{N-1}$$

$$\frac{u_{N+1} - 2u_N + u_{N-1}}{\Delta x^2} = 1 \rightarrow u_{N+1} - 2u_N + u_{N-1} = \Delta x^2$$

$$u_{N-1} - 2u_N + u_{N-1} = \Delta x^2 \rightarrow -2u_N + 2u_{N-1} = \Delta x^2$$

$$u_{N-1} - u_N = \frac{\Delta x^2}{2}$$

De esta forma nos deshacemos del nodo fantasma. La matriz resulta:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} T_a \\ \Delta x^2 \cos^2(x_1) \\ \Delta x^2 \cos^2(x_2) \\ \vdots \\ \Delta x^2 \cos^2(x_{N-2}) \\ \Delta x^2 \cos^2(x_{N-1}) \\ \frac{\Delta x^2}{2} \end{bmatrix}$$

Se muestra en rojo el cambio en la matriz respecto al caso anterior. No obstante, aunque solo se modifique la última fila se debe resolver el SEL completo para obtener la nueva solución en todos los puntos.

Otra opción para eliminar el nodo fantasma es expresar la derivada en el punto final en función de puntos interiores del espacio. Por ejemplo, por el uso de la derivada primera discretizada en atraso con un método de 3 puntos:

$$\frac{1}{\Delta x}\left(\frac{1}{2}u_{N-2} - 2u_{N-1} + \frac{3}{2}u_N\right) = 0 \rightarrow \frac{1}{2}u_{N-2} - 2u_{N-1} + \frac{3}{2}u_N = 0$$

En este caso NO hay nodo fantasma y la matriz resulta:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & \frac{1}{2} & -2 & \frac{3}{2} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} T_a \\ \Delta x^2 \cos^2(x_1) \\ \Delta x^2 \cos^2(x_2) \\ \vdots \\ \Delta x^2 \cos^2(x_{N-2}) \\ \Delta x^2 \cos^2(x_{N-1}) \\ 0 \end{bmatrix}$$

Otro ejemplo: Problema de capa límite

Tenemos que:

$$-\mu\frac{d^2u}{dx^2} + b\frac{du}{dx} = 0 \quad 0 \le x \le 1 \quad \begin{matrix} u_{(0)} = 0 \\ u_{(1)} = 1 \end{matrix}$$

Discretización:

1) **Espacio:**

$$\Delta x = cte \quad x_i = i\Delta x \quad con\ i = 0,1,\cdots,N \quad \Delta x = \frac{1}{N}$$

2) **EDO:**

$$u_i \approx u_{(x_i)} \quad -\mu\left[\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}\right] + b\left[\frac{u_{i+1} - u_{i-1}}{2\Delta x}\right] = 0 \quad para\ i = 1,\cdots,N-1$$

Definimos el número de Peclet (Pe) como:

$$Pe = \frac{b\Delta x}{2\mu}$$

Entonces reescribimos:

$$-\frac{\mu}{\Delta x^2}u_{i+1} + \frac{\mu}{\Delta x^2}2u_i - \frac{\mu}{\Delta x^2}u_{i-1} + \frac{b}{2\Delta x}u_{i+1} - \frac{b}{2\Delta x}u_{i-1} = 0$$

$$\left(\frac{b\Delta x}{2}-\mu\right)u_{i+1}+2\mu u_i-\left(\frac{b\Delta x}{2}+\mu\right)u_{i-1}=0$$

$$\left(\frac{b\Delta x}{2\mu}-1\right)u_{i+1}+2u_i-\left(\frac{b\Delta x}{2\mu}+1\right)u_{i-1}=0 \quad Pe=\frac{b\Delta x}{2\mu}$$

$$-(Pe+1)u_{i-1}+2u_i+(Pe-1)u_{i+1}=0$$

3) Condiciones de contorno (CC):

$$i=0\rightarrow u_0=0 \quad i=N\rightarrow u_N=1$$

Analizaremos numéricamente el comportamiento de la solución numérica obtenida. Para ello se propone que herede la solución analítica genérica, es decir:

$$u_{(x)}\sim e^{\alpha x}\rightarrow u_i=e^{\alpha x_i}=e^{\alpha i\Delta x}=e^{(\alpha\Delta x)i}=\gamma^i$$

Reemplazamos $u_i=c\gamma^i$ con $c=cte\neq 0$ en la formula obtenida:

$$-(Pe+1)c\gamma^{i-1}+2c\gamma^i+(Pe-1)c\gamma^{i+1}=0$$

Cancelamos los c y como sabemos que $\gamma^i=0$ no es solución de la ecuación, tenemos que:

$$-(Pe+1)\gamma^{i-1}+2\gamma^{i-1}\gamma^1+(Pe-1)\gamma^{i-1}\gamma^2=0$$

$$-(Pe+1)+2\gamma^1+(Pe-1)\gamma^2=0\rightarrow Ecuación\ característica$$

Resolvemos:

$$\gamma_{1,2}=\frac{-2\pm\sqrt{4+4(Pe-1)(Pe+1)}}{2(Pe-1)}=\frac{-2\pm\sqrt{4+4(Pe^2-1)}}{2(Pe-1)}=$$

$$=\frac{-2\pm\sqrt{4+4Pe^2-4}}{2(Pe-1)}=\frac{-2\pm\sqrt{4Pe^2}}{2(Pe-1)}=\frac{-2\pm 2Pe}{2(Pe-1)}=\frac{-1\pm Pe}{(Pe-1)}$$

$$\begin{cases}\gamma_1=\dfrac{-1+Pe}{(Pe-1)}=1\\ \gamma_2=\dfrac{-1-Pe}{(Pe-1)}=\dfrac{1+Pe}{1-Pe}\end{cases}$$

Por lo que la solución numérica general es:

$$u_i=A_1\gamma_1^i+A_2\gamma_2^i=A_1+A_2\left(\frac{1+Pe}{1-Pe}\right)^i$$

Para determinar los coeficientes A_j se utilizan las CC:

$$u_0=0=A_1+A_2\left(\frac{1+Pe}{1-Pe}\right)^0=A_1+A_2=0\rightarrow A_1=-A_2$$

$$u_N=1=A_1+A_2\left(\frac{1+Pe}{1-Pe}\right)^N=A_1+A_2\left(\frac{1+Pe}{1-Pe}\right)^N=A_1-A_1\left(\frac{1+Pe}{1-Pe}\right)^N=$$

$$=A_1\left(1-\left(\frac{1+Pe}{1-Pe}\right)^N\right)=1\rightarrow A_1=\frac{1}{1-\left(\frac{1+Pe}{1-Pe}\right)^N}=-A_2$$

Entonces:

$$u_i = \frac{1}{1-\left(\frac{1+Pe}{1-Pe}\right)^N} - \frac{1}{1-\left(\frac{1+Pe}{1-Pe}\right)^N}\left(\frac{1+Pe}{1-Pe}\right)^i = \frac{1-\left(\frac{1+Pe}{1-Pe}\right)^i}{1-\left(\frac{1+Pe}{1-Pe}\right)^N}$$

Se observa que si $Pe > 1$ el resultado oscila (es inestable), pues con cada potencia (par o impar) el resultado tomará un valor positivo y otro negativo. Como queremos evitar esto surge la condición:

$$Pe < 1 \rightarrow \frac{b\Delta x}{2\mu} < 1 \rightarrow \Delta x < \frac{2\mu}{b}$$

De esta forma obtenemos un límite máximo para el paso de discretización. No obstante, para valores prácticos tenemos que $\mu \sim 10^{-5}$ y $b \sim [10^0 - 10^1]$ es decir que:

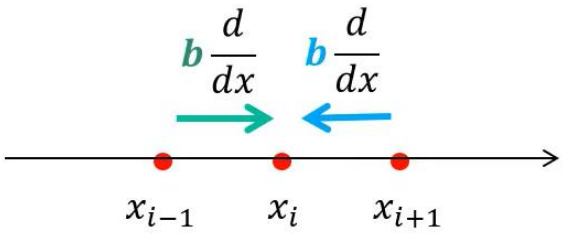
$$\Delta x \sim 10^{-5} m \ (m: metros)$$

Por lo tanto, la condición mostrada es impracticable por su elevado costo computacional.

Técnica de Upwinding

En consecuencia, como alternativa para minimizar el comportamiento oscilatorio, dado que $Pe < 1$ no se puede corregir, se aplica la **técnica de upwinding** discretizando a orden 1 la derivada primera:

$$\begin{cases} \frac{du}{dx} \rightarrow \frac{u_1 - u_{i-1}}{\Delta x} & si\ b > 0\ (en\ atraso) \\ \frac{du}{dx} \rightarrow \frac{u_{1+1} - u_i}{\Delta x} & si\ b < 0\ (en\ adelanto) \end{cases}$$



Si $b > 0$ la información fluye de izquierda a derecha (el nodo $i-1$ transmite información al nodo i) y debe emplearse la fórmula en atraso, caso contrario fluye de derecha a izquierda (el nodo $i+1$ transmite información al nodo i), debe usarse la fórmula en adelanto. De aquí la elección de discretización en atraso o en adelanto según el valor de b.

Sin embargo, hay que tener en cuenta que esto conlleva perder el orden 2 de la discretización puesto que estamos empleando derivadas de orden 1. Además, no se elimina el comportamiento oscilatorio, sino que se minimiza su efecto. Por ende, se requiere un paso pequeño.