



Omar Cordero P. · Abdul Lugo J.

Triada de la Aproximación en el Cálculo Diferencial e Integral

El comportamiento numérico como
elemento fundamental para la
comprensión y estudio del Cálculo
Diferencial e Integral

Omar Cordero P.
Abdul Lugo J.

Triada de la Aproximación en el Cálculo Diferencial e Integral

**Omar Cordero P.
Abdul Lugo J.**

Triada de la Aproximación en el Cálculo Diferencial e Integral

**El comportamiento numérico como elemento
fundamental para la comprensión y estudio del
Cálculo Diferencial e Integral**

Editorial Académica Española

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

Editorial Académica Española

is a trademark of

International Book Market Service Ltd., member of OmniScriptum Publishing Group

17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

ISBN: 978-613-9-40879-5

Copyright © Omar Cordero P., Abdul Lugo J.

Copyright © 2019 International Book Market Service Ltd., member of OmniScriptum Publishing Group

TRIADA

DE LA

APROXIMACIÓN

EN EL

CÁLCULO DIFERENCIAL

E

INTEGRAL

Omar Cordero P.

Abdul Lugo J.

Agradecimiento

Encabezo este agradecimiento a Dios, que me ha concedido la salud física y mental así como la suficiente inspiración para la edición. Igualmente a Francy Garrido y a mis hijos, quienes me fortalecieron y fueron importantes pilares que coadyuvaron el desarrollo de esta obra. Los esposos Escontrela-Dieguez quienes donaron la licencia de Wolfram Mathematica, programa central de este libro. La observaciones, correcciones y colaboración hecha por el coautor Dr. Abdul Lugo Jiménez como destacado docente en cálculo numérico quien fue un importante apoyo en el segundo capítulo de su autoría. Mi más sincera gratitud a todos.

Índice general

Introducción	6
Capítulo 1. Interpolación	11
1.1. Interpolación de Lagrange	12
1.2. Diferencias Divididas de Newton	15
1.3. Spline Cúbico	16
EJERCICIOS	21
Capítulo 2. Integración	25
2.1. Las Reglas de Newton-Cotes	26
2.1.1. Obtención de la Regla de Newton-Cotes	27
2.1.2. Implementación	29
2.2. Error Newton-Cotes	31
2.3. Regla Compuesta	33
2.3.1. Obtención de la Regla Compuesta	33
2.3.2. Error	35
2.4. Cuadratura Adaptativa	37
2.4.1. Un Proceso Adaptativo de Newton-Cotes	37
EJERCICIOS	41
Capítulo 3. Ecuaciones Diferenciales	43
3.1. Ecuaciones Diferenciales de Primer Orden	43
3.1.1. EDO de variable separable	44
3.1.2. Homogéneas	46
3.1.3. Exactas	47
3.1.4. Teorema de la existencia y unicidad	48
3.1.5. Métodos numéricos	51
3.1.6. Aplicaciones	55
3.2. Ecuaciones Diferenciales de Segundo Orden	60
3.2.1. Ecuaciones diferenciales homogéneas con coeficientes constantes	65
3.2.2. Movimiento Vibratorio Simple	68
3.2.3. Movimiento Vibratorio Amortiguado	70
3.2.4. Ecuaciones Diferenciales con Coeficientes Variables	72
EJERCICIOS	76
Bibliografía	81

Introducción

Son muchos los casos que abriga el quehacer de la aproximación matemática desde la antigüedad, uno es el hecho que los círculos conservan una estrecha relación entre su perímetro y el radio pero ¿se conoce con exactitud esa relación o se debe dar una aproximación?, la Biblia le asigna el valor 3, en Babilonia $3\frac{1}{8}$, los egipcios $4\left(\frac{8}{9}\right)^2$, Siddhantas 3,1416; Brahmagupta 3,162277 y en China 3,1724; desde el siglo XVII esta relación se convirtió en un número y fue identificado con el nombre π (Pi) que proviene del griego *periphereia* que se refiere a perímetro, largo fue el camino hasta aceptar a π y fue en Grecia donde la exacta relación entre el diámetro y el perímetro de una circunferencia comenzó a consolidarse como un clásico problema de aproximación.

Euclides de Alejandría (325AC – 265AC) precisa en su obra *Elementos*, los pasos que involucran la aplicación del límite y desarrolla el método de *exhaustión* consistente en doblar el número de lados de los polígonos regulares inscritos y circunscritos hasta la convergencia del procedimiento. Arquímedes reúne y desarrolla estos resultados y muestra que los perímetros de los polígonos dentro del círculo deben ser más cortos que el círculo, mientras que los de fuera del círculo deben ser más largos que el círculo. Para hacer el cálculo Arquímedes construyó sus polígonos bisecando repetidamente los lados de un hexágono regular para obtener polígonos regulares con 12 lados, 24, 48 y así sucesivamente, los resultado fueron entre $\frac{223}{71} = 3,14084$ y $\frac{22}{7} = 3,14285$.

Georg Von Purbach (1423 – 1461) asume que $\pi = \frac{377}{120} = 3.1466\dots$ y fue con el desarrollo de la trigonometría a partir del siglo XV con las investigaciones de Adrien Romain (1561-1615) y Ludolph de Colonia (1539-1610) que se obtienen 15 y 32 cifras.

Con el cómputo de π hecho por el matemático inglés William Shanks (1812-1882) en 1853 se obtuvo 707 cifras pero cometió un error en el 528^{o} decimal. En 1949 John Von Neumann (1903-1957) con ayuda del ordenador presenta unas 2.037 cifras y con David H. Bailey (1948) se extrajo 29.360.000 cifras.

Cuántas cifras son necesarias y suficientes para una apropiada precisión, cuánto basta para medir el diámetro de una fibra óptica, el espesor aproximado de un vaso capilar cerebral o la longitud de una circunferencia que tenga por radio la distancia media de la tierra al sol o más allá, del sol a la nebulosa más lejana; entonces ¿qué necesidad obliga buscar tantas cifras? quizá la necesidad práctica y nunca la resignación a la limitación numérica.

$$\begin{aligned} \pi \approx & 3.141592653589793238462643383279502884197169399375105820 \\ & 97494459230781640628620899862803482534211706798214808651 \\ & 32823066470938460955058223172535940812848111745028410270 \\ & 19385211055596446229489549303819644288109756659334461284 \\ & 75648233786783165271201909\dots \end{aligned}$$

Número π con 250 cifras

Anticipándose a Isaac Newton (1643-1727) y Gottfried Withelm Leibniz (1646-1716) Arquímedes de Siracusa (ca 287-212 AC) halla el área de una región debajo de una parábola mediante la suma de los rectángulos producto de la subdivisión en bandas paralelas de igual

anchura, tantas como sean posibles a fin de reducir al máximo la exclusión del extremo curvado el cual fue el proceso base para la denominada *Integral*. Newton hace frente a esta nueva visión de las matemáticas construyendo métodos para calcular áreas y trazar tangentes a curvas que son esencialmente los problemas claves en el *Cálculo Diferencial* y su estrecha relación con el *Cálculo Integral*.

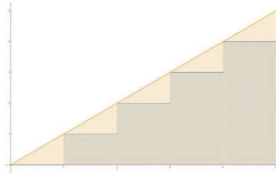


FIGURA 0.0.1. Área debajo de una curva.

El *Cálculo Diferencial* considera un tópico de suma importancia para las matemáticas, las *ecuaciones diferenciales*, originarias de la necesidad de modelar sistemas naturales estas irrumpen en modernos campos de la mecánica, astronomía, óptica, medicina y economía entre otros. Las ecuaciones diferenciales involucran derivadas de una función desconocida de una o más variables, si la función desconocida depende sólo de una variable (de tal modo que las derivadas son derivadas ordinarias) la ecuación se llama *ecuación diferencial ordinaria* o su abreviatura **EDO**. Sin embargo, si la función desconocida depende de más de una variable (de tal modo que las derivadas son derivadas parciales) la ecuación se llama *ecuación diferencial parcial*.

Frecuentemente se escribe $y = f(x)$ y se llama a x variable independiente, y y , la cual depende de x , la variable dependiente. Por conveniencia, se puede denotar el valor de y en x por $y(x)$, y sus derivadas sucesivas por $y'(x)$ y $y''(x)$, o simplemente y' , y'' ; en otros casos se puede hallar las siguientes notaciones:

Newton	\dot{y}
Lagrange	y'
Leibniz	$\frac{dy}{dx}$
Arbogast	Dy

Nos permitimos marcar el inicio del estudio de las ecuaciones diferenciales con Newton y Leibniz resaltando la incursión de éstos en las leyes naturales, por una parte la consideración de una *matemática en movimiento*, velocidad y aceleración son algunos de sus componentes fundamentales y por otra parte reflexiones más estáticas que involucran conceptos como área, volumen, distancias y temperatura.

Newton incorpora en *De analysi per aequationes numero terminorum infinitas* publicado en el 1711 la exposición de métodos para la diferenciación, pero no es hasta el 1671 que en *Methodus fluxionum et serierum infinitarum* publicado en el 1742 aparecen claramente

ecuaciones diferenciales, se le reconocen diferentes aportes a la Astronomía, Mecánica, óptica y métodos de series para resolver ecuaciones diferenciales.

Leibniz presta atención a las obras de Blaise Pascal (1623-1662) sobre la cicloide, y el año 1673 determina que la tangente a una curva depende del cociente de las diferencias entre las ordenadas y las abscisas cuando estas diferencias se hacen infinitamente pequeñas y utilizó la notación que hoy día se emplea de $\frac{dy}{dx}$, fue el primero en introducir el término *ecuación diferencial* y el término *derivar*. Hacia el 1691 Leibniz resuelve por primera vez ecuaciones diferenciales de primer orden de variables separables y homogéneas.

Los más famosos matemáticos de la familia Bernoulli, Jakob (1654-1705), Johann (1667-1748) y Daniel (1700-1792) aportaron destacados elementos en la física, astronomía, el uso del factor integrante, el problema de la cuerda vibrante y la baquistócrona (James) y la propuesta de la Ecuación de Bernoulli en el 1695, la cual la resuelven John y Leibniz.

John Bernoulli expuso en la revista *Acta Eroditorum* hallar la solución de una catenaria, lo cual consiste en la curva formada por un hilo pesado, flexible, inextensible y de densidad constante en toda su longitud, fijo o suspendido en sus dos extremos. Este problema fue resuelto por él mismo conjuntamente con su hermano John Bernoulli (1667- 1748) y Charles Huyghens (1629-1695) y publicado en esa revista el año 1691.

En año de 1734 Euler introdujo el concepto de *Ecuaciones Diferenciales Parciales*, y hacia el 1746 Jean le Ronde D'Alembert (1771-1783) se dirige a las EDP desempolvando el problema de la *cuerda vibrante* que en 1727 Johann Bernoulli adoptó el *elemento finito* de masas puntuales equidistantes en una cuerda sin masa, y es cuando D'Alembert lo expone como una cuerda continua de densidad uniforme haciendo los cálculos de Bernoulli a n masas que tienden al infinito concluyendo que la cuerda continua era de infinitos segmentos infenitesimales interconectados.

Se debe a Augustin Louis Cauchy (1789-1857) el inicio de los métodos para hallar la existencia de las soluciones en los problemas de valor inicial y para el 1824 las aproximaciones sucesivas, Carles Émile Picard (1856-1941) publica en el 1890 el *Teorema de la existencia y unicidad* quien lo presenta en una versión más moderna.

Nombre	Ecuación	Año	Autor
Separable	$\frac{dy}{dx} = f(x)g(y)$	1691	Leibniz
Homogénea	$\frac{dy}{dx} = f\left(\frac{x}{y(x)}\right)$	1691	Leibniz
Lineal de primer orden	$\frac{dy}{dx} + P(x)y(x) = Q(x)$	1694	Leibniz
Bernoulli	$\frac{dy}{dx} + yP(x) = y^n Q(x)$	1695	J. Bernoulli
Riccati	$y'(x) = f(x) + g(x)y(x) + h(x)y(x)^2$	1724	Riccati
Exactas	$M(x, y)dx + N(x, y) = 0$ con $M(x, y) = \frac{\partial f}{\partial x}$ y $N(x, y) = \frac{\partial f}{\partial y}$	1734	Euler
Lineales con coeficientes constantes	$a_n y^n(x) + a_{n-1} y^{n-1}(x) + \dots + a_0 y(x) = P(x)$ con $\{a_n, a_{n-1}, \dots, a_0\}$ constantes	1743	Euler
Legengre	$(1 - x^2)y''(x) - 2xy'(x) - n(n-1)y(x) = 0$	1785	Legengre
Bessel	$x^2 y''(x) + xy'(x) + (x^2 - n^2)y(x) = 0$	1824	Bessel
Abel	$y'(x) = f(x) + g(x)y(x) + h(x)y(x)^2 + k(x)y(x)^3$	1834	Abel
Chini	$y'(x) = f(x) + g(x)y(x) + h(x)y(x)^n$	1924	Chini

Wolfram Research Inc., 2018.

De la mano con la complejidad y formalidad que ha representado el estudio de los métodos numéricos y el compromiso académico que ello rige, se abre paso a la intención de un aprendizaje significativo en situaciones donde se desea estudiar e interpretar los efectos que originen cambios en las variables o constantes en funciones de variable real con el uso de la herramienta gráfica y de cálculo Mathematica, en esta oportunidad se usará como asistente para graficar funciones y familias de funciones que manualmente sería una ardua tarea, facilitar el cálculo con alta precisión, hacer cambios en los parámetros generales para estudiar el comportamiento de los resultados y aportar soluciones numéricas y gráficas.

La intención es reconocer lo que el aporte académico y bibliográfico especializado nos ha consolidado, y manipular esas contribuciones en un ambiente de laboratorio de matemáticas donde profesor y estudiante experimenten de manera interactiva, cooperativa y agradable, ideas, conceptos y fundamentos matemáticos.

En general se pretende exponer el comportamiento gráfico y numérico como elementos fundamentales para la comprensión y estudio de la interpolación, la integral y las ecuaciones diferenciales ordinarias; se exhibe en las siguientes partes:

Capítulo 1: Una introducción y desarrollo que sirve como base ilustrativa en la utilidad de los métodos para la interpolación así como el uso de Mathematica en la aproximación.

Capítulo 2: Abarca métodos numéricos de integración como la regla del trapecio, la de Simpson, las fórmulas de Newton-Cotes, el método de integración de Gauss y el método para integrales impropias.

Capítulo 3: Recoge las ecuaciones diferenciales ordinarias de primer y segundo orden.

Interpolación



Isaac Newton
(1643-1727)

El método de flujiones es la clave general en cuya virtud la matemática moderna revela el secreto de la geometría y en consecuencia, de la naturaleza.

Obispo de Berkeley

En este libro se mostrará con frecuencia aproximaciones numéricas, por eso se considera de suma utilidad indicar que el valor numérico está ceñido al concepto de *cifra significativa*, se refiere a las que pueden ser usadas en forma confiable, 25.5 tiene tres cifras significativas y 1020.02 tiene seis cifras significativas, pero aunque resulte sencillo en algunos casos, puede haber confusión en el caso donde los ceros no siempre son cifras significativas 00032, 000032 y 00032 tienen dos cifras significativas, sin embargo en los casos donde se usa notación científica como 2.36×10^{12} se puede afirmar que tiene tres cifras significativas y 2.360×10^{12} tiene cuatro.

Se inicia este capítulo reconociendo que la interpolación es esencial para la construcción de diferentes modelos numéricos, su característica esencial es que a partir de un muestreo que proporcione una tabla de datos sigue la construcción de una función que los aproxima, y como consecuencia se obtiene un ajuste de una función entre dos puntos de su dominio.

Los algoritmos que se expondrán en este libro comienzan con un punto cercano $f(x_i)$ que se ve afectado por reiteradas correcciones a medida que la información de valores $f(x_i)$ más distantes es incorporada. Si la función tiene un comportamiento óptimo, la última corrección será la más cercana y puede ser utilizada para estimar el error. La manipulación del proceso sugiere usar un número finito de vecinos próximos los cuales generan valores interpolados que en general no tienen continuidad, esto se debe a que el valor interpolado x cruza uno de los puntos disponibles x_i , el procedimiento de interpolación cambia el grupo de vecinos próximos y cual puede generar una discontinuidad en la función interpolada en ese punto, lo cual no es óptimo. Aumentar el número de x_{i-1} usado en el esquema de interpolación

no necesariamente aumenta la precisión especialmente en la interpolación polinomial. Al aumentar el número de puntos vecinos al punto de interés x , el polinomio de mayor orden tiende a hacer que la función de interpolación fluctúe excesivamente entre los puntos x_i . Esa oscilación puede no tener nada que ver con la función verdadera a menos que se tenga evidencia que la función interpolante sea similar a f , por esto se recomienda monitorear el error en la interpolación con un orden alto.

El comportamiento numérico de determinadas características de un fenómeno o evento pueden predecirse mediante los *polinomios interpoladores*, pero no existen reglas generales para la toma de decisión de cuál es la interpolación más adecuada, pues el aumento del grado del polinomio no es garante de precisión, en este capítulo se consideran tres importantes métodos.

1.1. Interpolación de Lagrange

El ajuste lineal de una función entre dos puntos de su dominio será:

$$g(x) = \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b)$$

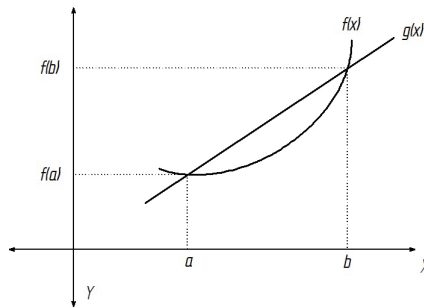


FIGURA 1.1.1. Interpolación Lineal

pensemos ahora en un polinomio que ajuste los f_0, f_1, \dots, f_n de $f(x)$, dicho polinomio es único y de orden n , además tendrá $n + 1$ términos así que la serie de potencias será:

$$g(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

dicho de otra manera, para cada $n + 1$ términos serán:

$$f_0 = a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n$$

$$f_1 = a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n$$

$$\vdots$$

$$f_n = a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n$$

las potencias de x se pueden escribir como:

$$P_0(x) = (x - x_1)(x - x_2) \dots (x - x_n)$$

dividiendo $P_0(x)$ por $P_0(x_0)$ se tiene:

$$\frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)}$$

finalmente la ecuación de Lagrange se escribe como:

$$g(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} f_0 + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} f_1 + \dots + \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})} f_n$$

el error cometido será:

$$O(x) = \frac{1}{2} (x-a)(x-b) f''(\xi), \quad a < \xi < b$$

Mathematica usa el comando *InterpolatingPolynomial* para dicha operación:

EJEMPLO 1.1. La tabla temperaturas de un determinado procesador con relación al tiempo viene dada por:

temperatura ($^{\circ}C$)	40	58	75
tiempo (<i>segundos</i>)	10	20	30

CUADRO 1.1.1. Temperaturas procesador.

el tiempo estimado para que dicho procesador eleve su temperatura a unos $45^{\circ}C$ según la interpolación de Lagrange será:

datos = {{10, 40}, {20, 58}, {30, 75}}

$$\begin{pmatrix} 10 & 40 \\ 20 & 58 \\ 30 & 75 \end{pmatrix}$$

y = LinearModelFit[datos, t, t]

FittedModel **1.75 t + 22.66666667**

Normal[y]

$$1.75t + 22.6667$$

Solve[%=45]

$$\{\{t \rightarrow 12.7619\}\}$$

finalmente el requerimiento será a partir de los 12,76 segundos.

EJEMPLO 1.2. En un laboratorio se ha determinado que la vida media (minutos) de una bacteria varía con la exposición de temperatura ($^{\circ}C$) según la siguiente tabla:

data = {{6, 3.2}, {8, 3.4}, {12, 3.7}, {15, 3.8}, {16, 4.1}, {30, 2.1}}; Prepend[data, {" $^{\circ}C$ ", "vida media"}]

$$\begin{pmatrix} \text{°C} & \text{vida media} \\ 6 & 3.2 \\ 8 & 3.4 \\ 12 & 3.7 \\ 15 & 3.8 \\ 16 & 4.1 \\ 30 & 2.1 \end{pmatrix}$$

el polinomio de interpolación vendrá dado por:

$y = \text{InterpolatingPolynomial}[\text{data}, \text{°C}]$

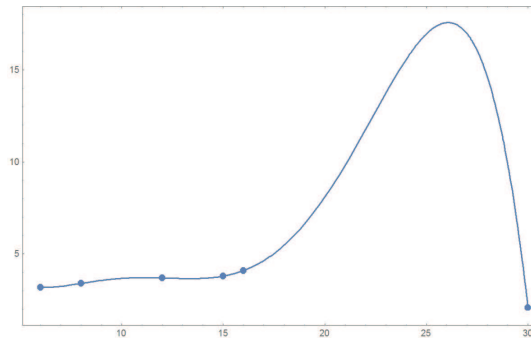
luego:

$$\begin{aligned} & ((((-0.0000659622(\text{°C}-8)-0.0000618537)(\text{°C}-12)-0.000631614)(\text{°C}-16)-0.00970238)(\text{°C}- \\ & 6) - 0.0458333)(\text{°C} - 30) + 2.1 \end{aligned}$$

¿Qué vida media tendrá la bacteria estudiada para si la temperatura es de 26°C ?
 $y/.^{\circ}\text{C} \rightarrow 26//N$

17.555555556 minutos

Show[ListPlot[data], Plot[y, {°C, 6, 30}], Frame \rightarrow True, PlotRange \rightarrow All]



1.2. Diferencias Divididas de Newton

Este polinomio de interpolación tiene la presentación:

$$P(x) = A_0 + A_1(x - x_0) + A_2(x - x_0)(x - x_1) + \dots + A_k(x - x_0)(x - x_1) \dots (x - x_{k-1})$$

donde a cada coeficiente $A_k = f(x_0, x_1, \dots, x_k)$ se les denomina *diferencias divididas* de la función f y se genera mediante:

$$f(x_0, x_1, \dots, x_k) = \frac{f(x_1, x_2, \dots, x_k) - f(x_0, x_1, \dots, x_{k-1})}{(x_k - x_0)}, k = 1, 2, \dots, n$$

finalmente dicho polinomio de interpolación se puede escribir como:

$$P(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \dots + f(x_0, x_1, \dots, x_k)(x - x_0)(x - x_1) \dots (x - x_{k-1}).$$

Ilustramos este proceso usando Mathematica:

```
puntos = {{1, 7}, {3, 6}, {5, 4}, {7, 1}, {9, 5}};
n = Length[puntos] - 1; For[i = 0, i <= n, i++, x_i = puntos[[i + 1, 1]]; f_i = puntos[[i + 1, 2]]]
Clear[d]; dif = Array[d, {2 * n + 2, n + 2}, -1];
For[i = -1, i <= 2 * n, i++, For[j = -1, j <= n, j++, d[i, j] = " "]]
d[-1, 1] = "x_k"; d[-1, 0] = "f_k"; For[i = 0, i <= n + 1, i++, d[2 * i, -1] = x_i;
d[2 * i, 0] = f_i]; For[j = 1, j <= n, j++,
For[i = j, i <= 2 * n - j, i = i + 2, d[i, j] = (d[i + 1, j - 1] - d[i - 1, j - 1]) / (d[i + j, -1] - d[i - j, -1])]
dif // MatrixForm
```

$$\begin{pmatrix} f_k & x_k & & & & \\ 1 & 7 & & & & \\ & & -\frac{1}{2} & & & \\ 3 & 6 & & -\frac{1}{8} & & \\ & & -1 & & 0 & \\ 5 & 4 & & -\frac{1}{8} & & \frac{1}{48} \\ & & -\frac{3}{2} & & \frac{1}{6} & \\ 7 & 1 & & \frac{7}{8} & & \\ & & 2 & & & \\ 9 & 5 & & & & \end{pmatrix}$$

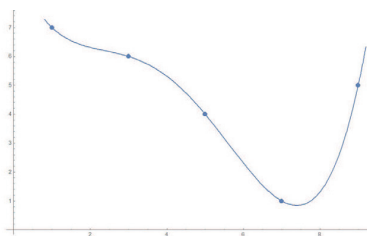
$$\begin{aligned}
 &(x-1) \\
 &(x-1)(x-3) \\
 &(x-1)(x-3)(x-5)(x-7)
 \end{aligned}
 \left(\begin{array}{cc|ccc}
 & f_k & x_k & & \\
 & & & \text{coeficientes} & \\
 \hline
 1 & 7 & & & \\
 3 & 6 & -\frac{1}{2} & & \\
 5 & 4 & -1 & 0 & \\
 7 & 1 & -\frac{1}{8} & \frac{1}{48} & \\
 & & -\frac{3}{2} & \frac{1}{6} & \\
 & & \frac{7}{8} & & \\
 & & 2 & & \\
 9 & 5 & & &
 \end{array} \right)$$

`sol = Sum [d[k, k] * Product [x - xj, {j, 0, k - 1}], {k, 0, n}]`

$$\frac{1}{48}(x-7)(x-5)(x-3)(x-1) - \frac{1}{8}(x-3)(x-1) - \frac{x-1}{2} + 7$$

`xmin = Min [Table [xi, {i, 0, n}]] ; xmax = Max [Table [xi, {i, 0, n}]] ;`

`g1 = ListPlot[puntos]; g2 = Plot[sol, {x, xmin - .2, xmax + .2}]; Show[g1, g2, PlotRange -> All]`



1.3. Spline Cúbico

La interpolación conocida como *Spline Cúbico* es una de las más empleadas pues con ello se garantiza un excelente ajuste. El polinomio de interpolación S es diferente para cada intervalo $[t_0, t_1], [t_1, t_2], \dots, [t_{n-1}, t_n]$ entonces, si S_i es el polinomio cúbico en el intervalo $[t_{n-1}, t_n]$ se tiene:

$$S(x) = \begin{cases} S_0(x), & x \in [t_0, t_1] \\ S_1(x), & x \in [t_1, t_2] \\ \vdots \\ S_{n-1}(x), & x \in [t_{n-1}, t_n] \end{cases}$$

como los polinomios S_{i-1} y S_i interpolan el mismo valor en t_i se cumple:

$$S_{i-1}(t_i) = y_i = S_i(t_i)$$

garantiza que S' y S'' son continuas.

Mediante estas condiciones y considerando S' y S'' se halla:

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \left(\frac{y_{i+1}}{h_i} + \frac{z_{i+1}h_i}{6}\right)(x - t_i) + \left(\frac{y_i}{h_i} - \frac{z_i h_i}{6}\right)(t_{i+1} - x)$$
donde $h_i = t_{i+1} - t_i$ y $\{z_0, z_1, \dots, z_n\}$ son variables que originan un sistema de ecuaciones lineales de $n + 1$ incógnitas.

En lenguaje Wolfram Mathematica se interpreta la interpolación *Spline Cúbico* con el comando *BSpline*.

EJEMPLO 1.3. Dados:

puntos = {{1, 7}, {3, 6}, {5, 4}, {7, 1}, {9, 5}}

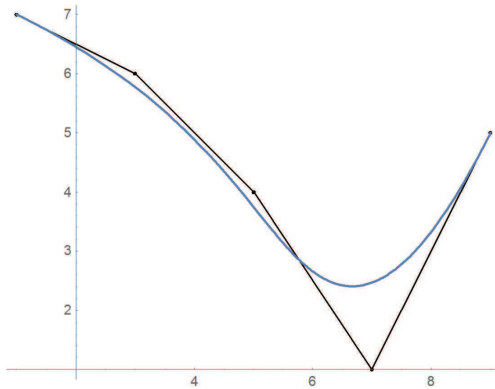
$$\begin{pmatrix} 1 & 7 \\ 3 & 6 \\ 5 & 4 \\ 7 & 1 \\ 9 & 5 \end{pmatrix}$$

f = BSplineFunction[puntos]

BSplineFunction [  Argument count: 1
Output dimension: 2]

Show[Graphics[{Point[puntos], Line[puntos]}, Axes → True],

ParametricPlot[f[t], {t, 0, 9}]



planteamos una rutina con Mathematica:

```
CSplineCoef[S_,db_,de_,a_,b_,c_,d_,n_,t_] := Module[{f,alpha,
mu,z,L,h},
n=Length[S];
f=Table[S[[i,2]],{i,1,n}];
t=Table[S[[i,1]],{i,1,n}];
a=Table[0,{i,1,n}];
```

```

b=Table[0,{i,1,n}];
c=Table[0,{i,1,n}];
d=Table[0,{i,1,n}];
alpha=Table[0,{i,1,n}];
mu=Table[0,{i,1,n}];
z=Table[0,{i,1,n}];
L=Table[0,{i,1,n}];
h=Table[0,{i,1,n}];
h[[1]]=t[[2]]-t[[1]];
alpha[[1]]=3(f[[2]]-f[[1]])/h[[1]]-3db;
L[[1]]=2h[[1]];
mu[[1]]=1/2;
z[[1]]=alpha[[1]]/L[[1]];
b[[1]]=db;
For[i=2,i<n,i++,
h[[i]]=t[[i+1]]-t[[i]];
alpha[[i]]=(3/(h[[i]]*h[[i-1]]))*(f[[i+1]]*h[[i-1]]-
f[[i]]*(h[[i]]+h[[i-1]])+f[[i-1]]*h[[i]]);
L[[i]]=2*(h[[i]]+h[[i-1]])-h[[i-1]]*mu[[i-1]];
mu[[i]]=h[[i]]/L[[i]];
z[[i]]=(alpha[[i]]-h[[i-1]]*z[[i-1]])/L[[i]]
];
alpha[[n]]=3de-3(f[[n]]-f[[n-1]])/h[[n-1]];
L[[n]]=h[[n-1]]*(2-mu[[n-1]]);
z[[n]]=(alpha[[n]]-h[[n-1]]z[[n-1]])/L[[n]];
c[[n]]=z[[n]];
For[j=n-1,j>0,j--,
c[[j]]=z[[j]]-mu[[j]]*c[[j+1]];
b[[j]]=(f[[j+1]]-f[[j]])/h[[j]]-h[[j]]*
(c[[j+1]]+2*c[[j]])/3;
d[[j]]=(c[[j+1]]-c[[j]])/(3*h[[j]]);
a[[j]]=f[[j]]
];
Print[" Listado de coeficientes:"]

```

CSplineCoef[puntos, 0, -100, a, b, c, d, n, t]

a
b
c
d

Listado de coeficientes:

```
{7, 6, 4, 1, 0}
{0,  $\frac{8}{7}$ ,  $-\frac{127}{14}$ ,  $\frac{387}{14}$ , 0}
{ $-\frac{37}{28}$ ,  $\frac{53}{28}$ , -7,  $\frac{355}{14}$ ,  $-\frac{2497}{28}$ }
{ $\frac{15}{28}$ ,  $-\frac{83}{56}$ ,  $\frac{151}{28}$ ,  $-\frac{1069}{56}$ , 0}
```

```
Clear[Sp]
```

```
For[i=1,i<n,i++,
```

```
Sp[i,x_]=a[[i]]+b[[i]]*(x-t[[i]])+
c[[i]]*(x-t[[i]])^2+d[[i]]*(x-t[[i]])^3
```

```
?Sp
```

```
Sp[1, x_] =  $7 - \frac{37}{28}(-1 + x)^2 + \frac{15}{28}(-1 + x)^3$ 
```

```
Sp[2, x_] =  $6 + \frac{8}{7}(-3 + x) + \frac{53}{28}(-3 + x)^2 - \frac{83}{56}(-3 + x)^3$ 
```

```
Sp[3, x_] =  $4 - \frac{127}{14}(-5 + x) - 7(-5 + x)^2 + \frac{151}{28}(-5 + x)^3$ 
```

```
Sp[4, x_] =  $1 + \frac{387}{14}(-7 + x) + \frac{355}{14}(-7 + x)^2 - \frac{1069}{56}(-7 + x)^3$ 
```

para determinar el valor de la aproximación $x = 4$ se observa que $3 \leq x \leq 5$ por lo tanto se escoge $Sp[2, x]$ con $x = 4$ será:

```
f[x_] = Sp[2, x]
```

$$-\frac{83}{56}(x-3)^3 + \frac{53}{28}(x-3)^2 + \frac{8(x-3)}{7} + 6$$

```
N[Sp[2, 4]]
```

```
7.55357
```

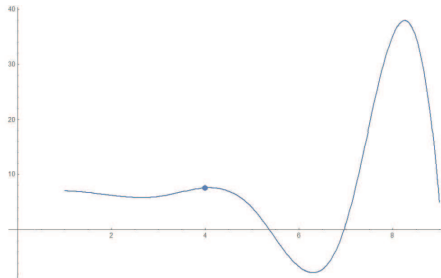
ahora se muestra un nuevo conjunto de puntos:

```
puntos2 = {{1, 7}, {3, 6}, {4, %}, {5, 4}, {7, 1}, {9, 5}, {12, 8}, {18, 25}}
```

$$\begin{pmatrix} 1 & 7 \\ 3 & 6 \\ 4 & 7.55357 \\ 5 & 4 \\ 7 & 1 \\ 9 & 5 \\ 12 & 8 \\ 18 & 25 \end{pmatrix}$$

```
For[i = 1, i < n, i++, P[i] = Plot[Sp[i, x], {x, t[[i]], t[[i + 1]]}]
```

`SP = Show[Table[P[i], {i, 1, n - 1}]]Show[ListPlot[{{4, Sp[2, 4]}}], SP, PlotMarkers -> ●, PlotRange->All]`



EJERCICIOS

1. En un registro que involucró el tiempo en años que relaciona el crecimiento de la población de una especie animal se obtuvo lo siguiente:

tiempo	1	1.5	1.7	2	4
población	1.58	1.95	2.02	2.05	0.6

use Mathematica para estimar el tiempo que origina el máximo crecimiento.

SOLUCIÓN.

`datos = {{1, 1.58}, {1.5, 1.95}, {1.7, 2.02}, {2, 2.05}, {4, 0.6}}`

$$\begin{pmatrix} 1 & 1.58 \\ 1.5 & 1.95 \\ 1.7 & 2.02 \\ 2 & 2.05 \\ 4 & 0.6 \end{pmatrix}$$

`f[x]:=Expand[InterpolatingPolynomial[datos,x]]//N`

determinación de los puntos críticos:

`Solve[D[f[x], x] == 0, x]`

`{{x -> 1.95477}, {x -> 7.52982}, {x -> 202.165}}`

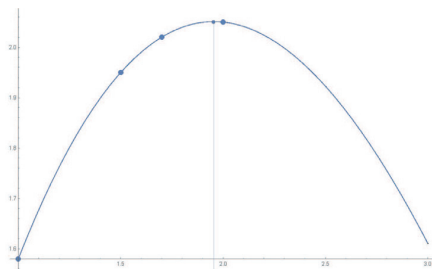
como $1 \leq x \leq 4$ se escoge $x = 1.95477$, luego:

`f[1.95477]`

2.05094

`Show[Plot[f[x], {x, 1, 3}], ListPlot[datos],`

`ListPlot[{{f[1.95477]}}, PlotMarkers -> ●, Filling->Axis]`



2. Dados los puntos:

$$\begin{pmatrix} 0.01 & 0.224 \\ 0.02 & 0.576 \\ 0.05 & 0.847 \\ 0.08 & 0.912 \\ 0.9 & 0.561 \\ 1.3 & 0.309 \end{pmatrix}$$

use la rutina de Newton para determinar el polinomio de interpolación.

SOLUCIÓN.

Matriz de coeficientes:

$$\begin{pmatrix} f_k & x_k & & & & & \\ 0.01 & 0.224 & & & & & \\ & & 35.2 & & & & \\ 0.02 & 0.576 & & -654.167 & & & \\ & & 9.03333 & & 7710.32 & & \\ 0.05 & 0.847 & & -114.444 & & -8521.05 & \\ & & 2.16667 & & 126.582 & & 6530.2 \\ 0.08 & 0.912 & & -3.05261 & & -97.0875 & \\ & & -0.428049 & & 2.30966 & & \\ 0.9 & 0.561 & & -0.165534 & & & \\ & & -0.63 & & & & \\ 1.3 & 0.309 & & & & & \end{pmatrix}$$

polinomio de interpolación:

$$\begin{aligned} & 6530.2(x - 0.9)(x - 0.08)(x - 0.05)(x - 0.02)(x - 0.01) \\ & - 8521.05(x - 0.08)(x - 0.05)(x - 0.02)(x - 0.01) + 7710.32(x - 0.05)(x - 0.02)(x - 0.01) \\ & - 654.167(x - 0.02)(x - 0.01) + 35.2(x - 0.01) + 0.224 \end{aligned}$$

3. Experimentos han demostrado que la tensión normal para cada tipo de metal se relaciona directamente con la resistencia de corte. Para una determinada muestra se obtuvo:

Tensión Normal (x)	Resistencia al corte (y)
25.8	25.4
24.5	23.7
29.8	22.4
26.3	21.7
26.7	26.3
29.3	26.3
27.4	29.5
21.8	24.1
25.6	28.5

estime la resistencia al corte para una tensión de 27.1.

SOLUCIÓN.

```
datos = SortBy[{{24.5, 23.7}, {29.8, 22.4}, {26.3, 21.7}, {26.7, 26.3},
{29.3, 26.3}, {27.4, 29.5}, {21.8, 24.1}, {25.6, 28.5}}, First]
```

$$\begin{pmatrix} 21.8 & 24.1 \\ 24.5 & 23.7 \\ 25.6 & 28.5 \\ 26.3 & 21.7 \\ 26.7 & 26.3 \\ 27.4 & 29.5 \\ 29.3 & 26.3 \\ 29.8 & 22.4 \end{pmatrix}$$

```
ifun = Interpolation[datos]
```

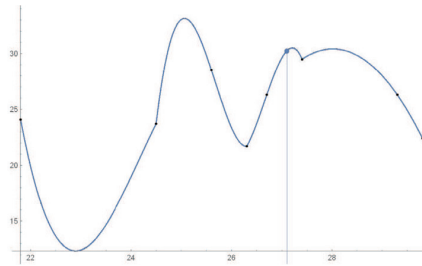
InterpolatingFunction[  Domain: (21.8 29.8)
Output: scalar]

```
ifun[27.1]
```

30.2489

```
Show[Plot[ifun[x], {x, 21.8, 29.8}, Epilog -> Point[datos]],
```

```
ListPlot[{{27.1, ifun[27.1]}}, PlotMarkers -> ●, Filling -> Axis]]
```



4. Un reporte sobre el comportamiento y la proyección económica de un país dado en millones de USD se registró en la siguiente tabla:

año	ingresos
1990	23.29
1995	31.27
2000	36.61
2005	43.7
2010	59.47
2015	26.3
2020	73.5
2025	81.23

- Determine el polinomio de interpolación que ajuste el comportamiento de la economía citada.
- Estime una proyección para el año 2030 y grafique.

Integración



Gottfried Wilhelm Leibniz
(1646-1716)

He tenido muchas ideas que quizá puedan ser útiles con el tiempo si otros con más penetración que yo calan profundamente en ellas algún día, y unen la belleza de sus mentes con el trabajo de la mía.

G. Leibniz

Una regla de cuadratura Q de m puntos para la integral definida:

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

es una aproximación de la forma:

$$Q = (b - a) \sum_{i=1}^m \omega_i f(x_i)$$

donde x_i son las abscisas y ω_i son los pesos. Las abscisas y los pesos definen la regla y se eligen de modo que Q sea aproximadamente igual a I . La eficiencia depende esencialmente del número de evaluaciones de la función. Esto se debe a que el tiempo de evaluación necesario de f en x_i suele ser mucho mayor que el tiempo necesario para formar la combinación lineal requerida de valores de función. Por lo tanto, una regla de seis puntos es el doble de costosa que una regla de tres puntos.

Iniciaremos con las reglas de cuadraturas de *Newton-Cotes*, estas reglas se obtienen mediante la integración de un interpolador polinómico de la función a integrar $f(x)$, luego discutiremos reglas compuestas basadas en una partición del intervalo $[a, b]$ en subintervalos. En una regla compuesta, se aplica una regla simple a cada subintegral y posteriormente se suma el resultado. Se enfatiza la determinación adaptativa de la partición con control del error y conduce a una clase importante de procedimientos de cuadratura que se adaptan al

comportamiento del integrando. El problema de cuadratura es ideal para introducir algunas de las ideas clave asociadas con la computación paralela, la cual no abordaremos en este capítulo.

2.1. Las Reglas de Newton-Cotes

Una forma de obtener una regla de cuadratura Q es integrar una aproximación polinomial $p(x)$ del integrando $f(x)$, en esencia se puede resumir como sigue:

$$p(x) \approx f(x) \implies \int_a^b f(x)dx \approx \int_a^b p(x)dx$$

La integración de las interpolaciones mediante un espaciado uniforme es la clave para la obtención de estas reglas de cuadratura.

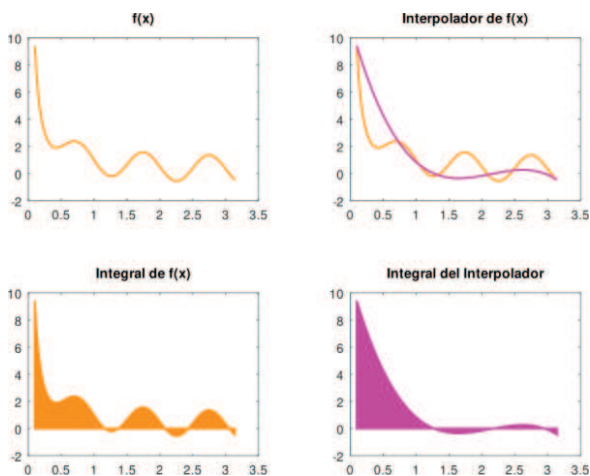


FIGURA 2.1.1. Gráficas que la función y su interpolante, así como de sus respectivas integrales.

La regla de Newton-Cotes de m puntos, para valores de m mayores o iguales a 2, está definida por:

$$Q_{NC(m)} = \int_a^b p_{m-1}(x)dx$$

donde $p_{m-1}(x)$ interpola a $f(x)$ con:

$$x = a + \frac{i-1}{m-1}(b-a), \quad i = 1$$

Si $m = 2$, entonces obtenemos la regla del trapecioide:

$$\begin{aligned} Q_{NC(2)} &= \int_a^b \left(f(a) + \frac{f(b) - f(a)}{b - a} (x - a) \right) dx \\ &= (b - a) \left(\frac{1}{2} f(a) + \frac{1}{2} f(b) \right) \end{aligned}$$

Si $m = 3$ y $c = \frac{(a+b)}{2}$ entonces obtenemos la regla de Simpson:

$$\begin{aligned} Q_{NC(3)} &= \int_a^b \left(f(a) + \frac{f(c) - f(a)}{c - a} (x - a) + \frac{\frac{f(b) - f(c)}{b - c} - \frac{f(c) - f(a)}{c - a}}{b - a} (x - a)(x - c) \right) dx \\ &= \frac{(b - a)}{6} \left(f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right) \end{aligned}$$

A partir de esta ilustración, indica que se obtiene una combinación lineal de evaluaciones f al expandir:

$$\int_a^b p_{m-1}(x) dx$$

2.1.1. Obtención de la Regla de Newton-Cotes. Para generalizar los m puntos, se procede sustituyendo la representación de Newton:

$$p_{m-1}(x) = \sum_{k=1}^m \left(c_k \prod_{i=1}^{k-1} (x - x_i) \right)$$

y se obtiene:

$$Q_{NC(m)} = \int_a^b p_{m-1}(x) dx = \sum_{k=1}^m c_k \int_a^b \left(\prod_{i=1}^{k-1} (x - x_i) \right) dx$$

Si se establece $x = a + sh$, donde $h = \frac{(b - a)}{(m - 1)}$, entonces resulta la siguiente ecuación:

$$Q_{NC(m)} = \int_a^b p_{m-1}(x) dx = h \int_0^{m-1} p_{m-1}(a + sh) ds = \sum_{k=1}^m c_k h^k S_{mk}$$

donde:

$$S_{mk} = \int_0^{m-1} \left(\prod_{i=1}^{k-1} (s - i + 1) \right) ds$$

Los términos c_k son diferencias divididas y por estar igualmente espaciados, tienen una forma particularmente simple en términos de f_i como se ha explicado en el capítulo anterior.

EJEMPLO 2.1.

$$\left\{ \begin{array}{l} c_1 = f_1 \\ c_2 = \frac{f_2 - f_1}{h} \\ c_3 = \frac{f_3 - 2f_2 + f_1}{2h^2} \\ c_4 = \frac{f_4 - 3f_3 + 3f_2 - f_1}{3!h^3} \end{array} \right.$$

Las fórmulas para $S_{m,k}$ también pueden ser obtenidas:

$$\left\{ \begin{array}{l} S_{m1} = \int_0^1 1 \cdot ds = (m-1) \\ S_{m2} = \int_0^1 s \, ds = \frac{(m-1)^2}{2} \\ S_{m3} = \int_0^1 s(s-1) \, ds = \frac{(m-\frac{5}{2})(m-1)^2}{3} \\ S_{m4} = \int_0^1 s(s-1)(s-2) \, ds = \frac{(m-3)^2(m-1)^2}{4} \end{array} \right.$$

Usando estas tabulaciones, se puede derivar con relativa facilidad los pesos para cualquier regla particular con m puntos.

Si $m = 4$, entonces:

$$S_{41} = 3, \quad S_{42} = \frac{9}{2}, \quad S_{43} = \frac{9}{2}, \quad S_{44} = \frac{9}{4}$$

Así:

$$\begin{aligned} Q_{NC(4)} &= S_{41}c_1h + S_{42}c_2h^2 + S_{43}c_3h^3 + S_{44}c_4h^4 \\ &= 3f_1h + \frac{9}{2}c_2\frac{f_2-f_1}{h}h^2 + \frac{9}{2}\frac{f_3-2f_2+f_1}{2h^2}h^3 + \frac{9}{4}\frac{f_4-3f_3+3f_2-f_1}{6h^3}h^4 \\ &= \frac{3h}{8}(f_1 + 3f_2 + 3f_3 + f_4) \\ &= \frac{(b-a)}{8}(f_1 + 3f_2 + 3f_3 + f_4) \end{aligned}$$

mostrando que:

$$\frac{[1 \ 3 \ 3 \ 1]}{8}$$

es el vector de pesos para $Q_{NC}(4)$.

2.1.2. Implementación. Por conveniencia para cálculos posteriores, podemos hacer un script para una función con los vectores de pesos de la regla de *Newton-Cotes* para casos puntuales:

```
function w=PesosNC(m)
%
% Entrada:
%     m: un entero que satisface 2<=m<=11
%
% Salida:
%     w: un vector columna de longitud m que contiene los pesos de la regla
%     de m puntos de Newton-Cotes
if m==2
    w=[1 1]'/2;
elseif m==3
    w=[1 4 1]'/6;
elseif m==4
    w=[1 3 3 1]'/8;
elseif m==5
    w=[7 32 12 32 7]'/90;
elseif m==6
    w=[19 75 50 50 75 19]'/288;
elseif m==7
    w=[41 216 27 272 27 216 41]'/840;
elseif m==8
    w=[751 3577 1323 2989 2989 1323 3577 751]'/17280;
elseif m==9
    w=[989 5888 -928 10496 -4540 10496 -928 5888 989]'/28350;
elseif m==10
    w=[2857 15741 1080 19344 5778 5778 19344 1080 15741 2857]'/89600;
elseif m==11
    w=[16067 106300 -48525 272400 -260550 427368 -260550 272400 -48525 106300 ...
        16067]'/598752;
end
```

Observe que los vectores de pesos son simétricos con respecto a su centro, volviendo ahora a la evaluación de $Q_{NC}(m)$ vemos que:

$$Q_{NC}(m) = (b-a) \sum_{i=1}^m \omega_i f_i = (b-a) \begin{bmatrix} \omega_1 & \cdots & \omega_m \end{bmatrix} \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}$$

es un producto interno escalar del vector de peso y el vector de valores de la función, por lo tanto obtenemos:

```
function numI = QNC(fname,a,b,m)
%
% Entrada:
%   fname: cadena que nombra una función disponible de la forma f(x) que se
%           definida en [a,b]. f devuelve un vector de columna si x es un
%           vector de columna.
%   a,b: numeros reales
%   m: un entero que satisface 2<=m<=11.
%
% Salida:
%   numI: la aproximación m puntos de Newton-Cotes de la integral de f(x)
%         desde a hasta b.
w = wNC(m);
x = linspace(a,b,m)';
f = feval(fname,x);
numI = (b-a)*(w'*f);
```

El siguiente script sirve como ilustración:

```
% Script Aplicación: ExperimentoQNC
%
% Examina las reglas de Newton-Cotes.
while input('¿Otro ejemplo? (1=Si, 0=No).')
    fname = input('Ingrese entre comillas el nombre de la función a integrar:');
    a = input('Ingrese el punto inicial del intervalo: ');
    b = input('Ingrese el punto final del intervalo: ');
    s = ['QNC(' fname sprintf('%d, %d ',m),a,b)];
    clc
    disp([' m ' s])
    disp(' ')
    for m=2:11
        numI = QNC(fname,a,b,m);
        disp(sprintf('%d %e', m, numI))
    end
end
```

Para la integral de la función $\sin(x)$ desde 0 hasta $\frac{\pi}{2}$ del script anterior será:

m	$QNC(' \sin', 0, \frac{\pi}{2}, m)$
2	0.7853981633974483
3	1.0022798774922104
4	1.0010049233142790
5	0.9999915654729927
6	0.9999952613861667
7	1.000000258372355
8	1.000000158229039
9	0.999999999408976
10	0.999999999621675
11	1.000000000001021

2.2. Error Newton-Cotes

¿Qué tan eficientes son las reglas de *Newton-Cotes*?, como se basan en la integración de un polinomio de interpolación, la respuesta depende claramente de la calidad del interpolante. Aquí mostraremos un resultado para la regla de Simpson:

Si $f(x)$ y sus primeras cuatro derivadas son continuas en $[a, b]$, entonces:

$$\left| \int_a^b f(x) dx - Q_{NC}(3) \right| \leq \frac{(b-a)^5}{2880} M_4$$

donde M_4 es una cota superior de $|f^{(4)}(x)|$ en $[a, b]$.

Notemos que si $f(x)$ es un polinomio cúbico, entonces $f^{(4)} = 0$ y entonces la regla de Simpson es exacta. Esto es algo sorprendente porque la regla se basa en la integración de un interpolador cuadrático. En general, se puede demostrar que

$$\int_a^b f(x) dx = Q_{NC}(m) + c_m f^{(d+1)}(\eta) \left(\frac{b-a}{m-1} \right)^{d+2}$$

donde c_m es una constante pequeña, η esta en el intervalo $[a, b]$, y

$$d = \begin{cases} m-1, & \text{si } m \text{ es par} \\ m, & \text{si } m \text{ es impar} \end{cases}$$

Observe que si m es impar, como en la regla de Simpson, entonces se obtiene un grado extra de precisión al pretender conocer $f^{(d+1)}$, por ejemplo, si:

$$\left| f^{(d+1)}(x) \right| \leq M_{d+1}$$

en $[a, b]$, entonces:

$$\left| Q_{NC}(m) - \int_a^b f(x) dx \right| \leq |c_m| M_{d+1} \left(\frac{b-a}{m-1} \right)^{d+2}$$

La siguiente función es útil para devolver esta cota superior conociendo el intervalo $[a, b]$, m y la apropiada cota de la derivada:

```

function error = NCErr(a,b,m,DerCota)
%
% Entrada:
%   a,b: escalares reales que satisfacen a<=b
%   m: un entero que satisface 2<=m<=11.
% DerCota: una cota superior para la d-esima derivada de la function f(x)
%           definida en [a,b] donde d=m si m es impar y d=m-1 si m es par.
%
% Salida:
%   error: una cota superior para el error absoluto de la regla de m puntos
%           de Newton-Cotes cuando se aplica a la integral de f(x) de a hasta b.
if m==2, d = 1; c = -1/12;
    elseif m==3, d = 3; c = -1/90;
    elseif m==4, d = 3; c = -3/80;
    elseif m==5, d = 5; c = -8/945;
    elseif m==6, d = 5; c = -275/12096;
    elseif m==7, d = 7; c = -9/1400;
    elseif m==8, d = 7; c = -8183/518400;
    elseif m==9, d = 9; c = -2368/467775;
    elseif m==10, d = 9; c = -173/14620;
    else
        d=11; c=-1346350/326918592;
end;
error = abs( c*DerCota*((b-a)/(m-1))^(d+2) );

```

Si se está considerando una regla m puntos con m par, entonces la regla de $(m-1)$ puntos es probablemente igual de aceptable y requiere una evaluación menos de la función; el siguiente script ilustra este punto:

```

% Archivo Script ExperimentoQNCError
%
% Describe la calidad de la cota del error de Newton-Cotes.
clc
disp('Caso Facíl: Integral desde 0 a pi/2 de sin(x)')
disp('      ')
disp('Tomar DerCota = 1.')
```

m	QNC(m)	Error	Cota del Error'
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

```

disp('      ')
for m=2:11
    numI = QNC('sin',0,pi/2,m);
    error = abs(numI-1);
    errorCota = NCErr(0,pi/2,m,1);
    s = sprintf('%e %e %e', numI,error,errorCota);

```

```
disp(sprintf(' %d ',m) s])
end
```

Los resultados los podemos resumir en la siguiente tabla:

m	$Q_{NC}(m)$	Error	Cota del Error
2	0.7853981633974483	2.146×10^{-1}	3.230×10^{-1}
3	1.0022798774922104	2.280×10^{-3}	3.321×10^{-3}
4	1.0010049233142790	1.005×10^{-3}	1.476×10^{-3}
5	0.9999915654729927	8.435×10^{-6}	1.219×10^{-5}
6	0.9999952613861667	4.739×10^{-6}	6.867×10^{-6}
7	1.0000000258372355	2.584×10^{-8}	3.714×10^{-8}
8	1.0000000158229039	1.582×10^{-8}	2.277×10^{-8}
9	0.9999999999408976	5.910×10^{-11}	8.466×10^{-11}
10	0.9999999999621675	3.783×10^{-11}	5.417×10^{-11}
12	1.0000000000001021	1.021×10^{-13}	1.460×10^{-13}

2.3. Regla Compuesta

Se puede resaltar que la cota del error descrito en la sección anterior no es óptima o no debería usarse salvo el que intervalo o dominio donde queremos acotar sea bastante pequeño, es decir, que $h = b - a$ sea lo suficientemente pequeño. Afortunadamente, hay una manera más sencilla o fácil de organizar el cálculo de la integral, de modo que las cuadraturas de intervalos pequeños prevezcan. A esto se le llama o conoce como *Reglas Compuestas*.

2.3.1. Obtención de la Regla Compuesta. Si tenemos una partición:

$$a = z_1 < z_2 < \dots < z_n < z_{n+1} = b$$

entonces

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{z_i}^{z_{i+1}} f(x) dx$$

Ahora, si se aplica $Q_{NC}(m)$ a cada una de las subintegrales, resulta una regla de cuadratura compuesta basada en $Q_{NC}(m)$. Por ejemplo, si:

$$\Delta_i = z_{i+1} - z_i \quad \text{y} \quad z_{i+\frac{1}{2}} = \frac{z_i + z_{i+1}}{2}, \quad i = 1 : n$$

entonces:

$$Q = \sum_{i=1}^n \frac{\Delta_i}{6} \left(f(z_i) + 4f\left(z_{i+\frac{1}{2}}\right) + f(z_{i+1}) \right)$$

es una regla compuesta de Simpson. En general, si z esta contenida en una partición de $[a, b]$ y **fname** es una cadena de caracteres de alguna función, entonces podemos tener el siguiente script de instrucciones:

```
numI = 0
for i=1:length(z)-1
    numI = numI + QNC(fname,z(i),z(i+1),m);
end
```

asignamos a `numI` a la estimación compuesta de m puntos de Newton-Cotes de la integral basada en la partición donde se encuentra z ; luego ilustraremos como automatizar la elección de una apropiada partición, por ahora nos centraremos en describir reglas compuestas que están basadas en particiones uniformes. En estas reglas:

$$z_i = a + (i-1)\Delta, \Delta = \frac{b-a}{n}, n > 1$$

```
numI = 0;
Delta = (b-a)/n;
for i=1:n
    numI = numI + QNC('f',a+(i-1)*Delta,a+i*Delta,m);
end
```

designemos la estimación producida por la regla de cuadratura mediante $Q_{NC}^{(n)}(m)$. El cálculo es un poco ineficiente ya que se involucra $n-1$ evaluaciones de funciones y adicionalmente un bucle. La evaluación de más a la derecha en la i -ésima llamada en Q_{NC} es la misma que la evaluación de f más a la izquierda de la $(i+1)$ -ésima llamada. La figura siguiente muestra y representa la situación en el caso de cuatro subintervalos, para una regla de cinco puntos.

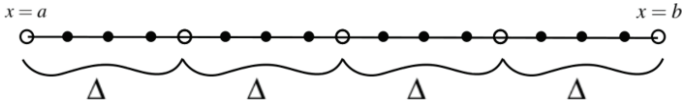


FIGURA 2.3.1. Representación de la evaluación de la función para $Q_{NC}^{(4)}(5)$.

Para evitar redundancia de las evaluaciones de f y un bucle con llamadas repetidas es mejor no aplicar Q_{NC} a cada una de las subintegrales. En su lugar calculamos previamente todas las evaluaciones de la función que sean necesarias y las almacenamos en un vector de columna simple, es decir `fval(1:n*(m-1)+1)`, así podemos calcular la combinación lineal correcta que define la regla compuesta.

En el ejemplo anterior de la regla compuesta de cinco puntos $Q_{NC}^{(4)}(5)$, las 17 evaluaciones de funciones requeridas se almacenan en `fval(1:17)`; si w es el vector de peso para $Q_{NC}(5)$, entonces:

$$Q_{NC}^{(4)}(5) = \Delta (w^T \text{fval}(1:5) + w^T \text{fval}(5:9) + w^T \text{fval}(9:13) + w^T \text{fval}(13:17))$$

A partir de esto concluimos que $Q_{NC}^{(n)}(m)$ es una suma de n productos internos, cada uno de los cuales implica el vector de peso w de la regla subyacente y una parte del vector `fval`, el siguiente siguiente listado lo muestra:

```
function numI = CompQNC(fname,a,b,m,n)
%
% Entradas:
%   fname: una cadena de caracteres del nombre de una funcion disponible de la
%           forma f(x) que esta definida en [a,b]. f devuelve un vector columna
%           si x es un vector columna.
```

```

%   a, b: son valores reales.
%   m: es un entero que satisface 2<=m<=11
%   n: un entero positivo
%
%   Salida:
%   numI: la aproximación compuesta de m puntos de Newton-Cotes de la integral
%         de f(x) desde a hasta b. La regla se aplica a cada uno de los n
%         subintervalos iguales de [a, b].
Delta = (b-a)/n;
w = WNC(m);
x = linspace(a,b,n*(m-1)+1)';
f = feval(fname,x);
numI = 0;
first = 1;
last = m;
for i=1:n
    numI = numI + w'*f(first:last);
    first = last;
    last = last+m-1;
end
numI = Delta*numI;

```

2.3.2. Error. Supongamos que Q_i^2 es la estimación de m puntos de Newton-Cotes de la i -ésima subintegral, si regla es exacta para polinomios de grado d , luego:

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{z_i}^{z_{i+1}} f(x) dx = \sum_{i=1}^n \left(Q_i + c_m f^{(d+1)}(\eta_i) \left(\frac{z_{i+1} - z_i}{m-1} \right)^{d+2} \right)$$

por definición:

$$Q_{NC}^{(n)}(m) = \sum_{i=1}^n Q_i$$

y

$$z_{i+1} - z_i = \Delta = \frac{b-a}{n}$$

sin embargo, se puede demostrar que:

$$\frac{1}{n} \sum_{i=1}^n f^{(d+1)}(\eta_i) = f^{(d+1)}(\eta)$$

para algún valor de η en $[a, b]$ y así:

$$\int_a^b f(x) dx = Q_{NC}^{(n)}(m) + c_m \left(\frac{b-a}{n(m-1)} \right)^{d+2} n f^{(d+1)}(\eta)$$

si $|f^{(d+1)}(z)|$ es menor o igual a M^{d+1} para todo x en el intervalo $[a, b]$, entonces:

$$\left| Q_{NC}^{(n)}(m) - \int_a^b f(x) dx \right| \leq \left[|c_m| M_{d+1} \left(\frac{b-a}{m-1} \right)^{d+2} \right] \frac{1}{n^{d+1}}$$

Si comparamos con el error obtenido usando la regla de Newton-Cotes, se comprueba que el error en la regla compuesta es el error simple dividido por n^{d+1} . Así que con un valor de m fijo es posible ejercer el control del error eligiendo un valor de n suficientemente grande. Por ejemplo, supongamos que queremos aproximar la integral con una regla de Simpson compuesta uniformemente espaciada para que el error sea menor que una tolerancia prescrita, sabemos que la cuarta derivada de f está acotada por M_4 , entonces elegimos n tal que:

$$\frac{1}{90}M_4 \left(\frac{b-a}{2}\right)^5 \frac{1}{n^4} \leq \text{tol}$$

para mantener el número de evaluaciones de la función lo más pequeño posible, n debe ser el número entero positivo más pequeño que satisfaga la siguiente relación:

$$n \geq (b-a) \sqrt[4]{\frac{M_4(b-a)}{2880 \cdot \text{tol}}}$$

El siguiente script, muestra dos ejemplos donde se aplican lo visto para dos funciones simples:

```
% Archivo de Script MostrarRCQNC
%
% Es script muestra la regla compuesta de Newton-Cotes
%
close all
% Ejemplo 01
figure(1)
a = 0;
b = pi/2;
for m = [3 5 7]
    err = [];
    for n = [2 4 8 16 32 64]
        err = [err abs(CompQNC('sin',0,pi/2,m,n)-1)+eps];
    end
    semilogy([2 4 8 16 32 64],err)
    axis([0 66 10^(-20) 10^(-3)])
    text(33,err(5),sprintf('m = %d',m))
    hold on
end
title('QNC(m,n) error para la integral de la funcion sin desde 0 hasta pi/2')
xlabel('n = numero de subintervalos.')
ylabel('Error QNC(m,n)')
% Ejemplo 02
figure(2)
a = 0;
b = 1;
for m = [3 5 7]
```



```

err = [];
for n = [2 4 8 16 32 64]
    err = [err abs(CompQNC('sqrt',0,1,m,n)-(2/3))+eps];
end
semilogy([2 4 8 16 32 64],err)
axis([0 66 10^(-6) 10^(-1)])
text(33,err(6),sprintf('m = %d',m))
hold on
end
title('QNC(m,n) error para la integral de la funcion f(x)=sqrt(x) de 0 a 1')
xlabel('n = numero de subintervalos.')
ylabel('Error QNC(m,n)')

```

Observe el comportamiento obtenido:

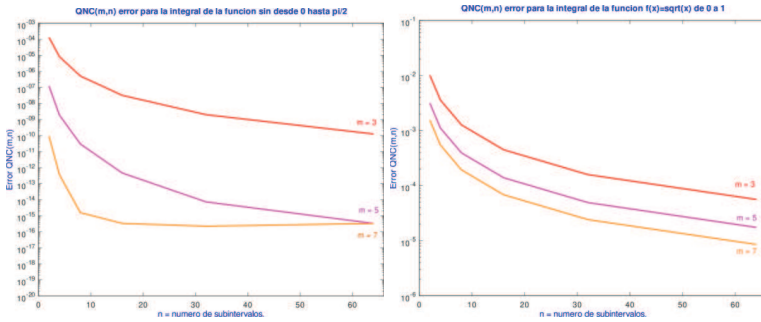


FIGURA 2.3.2. Aproximación del error.

2.4. Cuadratura Adaptativa

Las reglas compuestas uniformemente espaciadas que son exactas para polinomios de grado d son eficientes si $f^{(d+1)}$ se comporta uniformemente en $[a, b]$, sin embargo, si la magnitud de esta derivada varía ampliamente a lo largo del intervalo de integración entonces el proceso de control de errores puede resultar en una cantidad innecesaria de evaluaciones de la función. Esto se debe a que el valor de n está determinado por una cota M_{d+1} de la derivada a lo largo del intervalo. En regiones donde $f^{(d+1)}$ sea pequeña en comparación con este valor, los subintervalos son mucho más pequeños de lo necesario. Los métodos de cuadratura adaptativa abordan este problema descubriendo dónde el integrando presenta un comportamiento errático para disminuir el tamaño de los subintervalos.

2.4.1. Un Proceso Adaptativo de Newton-Cotes. Para un cómputo que ofrezca la mejor partición en $[a, b]$ debemos poder estimar el error, de esta manera la partición puede ser refinada si el error no es lo suficientemente pequeño, para ello se puede usar dos reglas de

cuadratura diferentes pues la diferencia entre los dos valores predichos de la integral puede tomarse como una medida de su inexactitud.

La forma de realizarla es similar al que desarrollamos para la aproximación lineal por tramos adaptativa. Para ver los detalles debemos desarrollar el método para estimar el error.

$$\begin{cases} A_1 = Q_{NC}^{(1)}(m) \\ A_2 = Q_{NC}^{(2)}(m) \end{cases}$$

Por lo tanto, A_1 es la estimación de regla de m puntos simple y A_2 es la estimación de regla de m puntos en 2 intervalos. Si estas reglas son exactas para polinomios de grado d , entonces se puede demostrar que:

$$I = A_1 + \left[c_m f^{(d+1)}(\eta_1) \left(\frac{b-a}{m-1} \right)^{d+2} \right]$$

$$I = A_2 + \left[c_m f^{(d+1)}(\eta_2) \left(\frac{b-a}{m-1} \right)^{d+2} \right] \frac{1}{2^{d+1}}$$

Supongamos que:

$$f^{(d+1)}(\eta_1) = f^{(d+1)}(\eta_2)$$

Lo cual, es bastante razonable si $f^{(d+1)}$ no cambia mucho en $[a, b]$. Por lo tanto, $I = A_1 + C$ y $I = A_2 + \frac{C}{2^{d+1}}$ donde:

$$C = \left[c_m f^{(d+1)}(\eta_1) \left(\frac{b-a}{m-1} \right)^{d+2} \right]$$

Al restar estas dos ecuaciones para I y resolvemos C , obtenemos

$$C = \frac{A_2 - A_1}{1 - \frac{1}{2^{d+1}}}$$

y así:

$$|I - A_2| \approx \frac{|A_1 - A_2|}{2^{d+1} - 1}$$

Por lo tanto, la discrepancia entre las dos estimaciones divididas por $2^{d+1} - 1$ proporciona una estimación razonable del error en A_2 . Si nuestro objetivo es producir una estimación de I que tenga un error absoluto tol o menor, entonces tenemos la siguiente función:

```
function numI = AdapQNC(fname,a,b,m,tol)
%
% Entrada:
%  fname: cadena de caracteres del nombre de una función de la forma f(x) que
%         esta definida en [a, b]. f devuelve un vector columna si x es un
%         vector columna.
%  a,b: numeros reales
%  m: es un entero que satisface 2<=m<=11.
%  tol: es un numero real positivo.
```

```

%
% Salida:
% numI: es la aproximación compuesta de m puntos de Newton-Cotes de la
% integral de f(x) de a hasta b, con la abscisa seleccionada
% adaptativamente.
%
A1 = CompQNC(fname,a,b,m,1);
A2 = CompQNC(fname,a,b,m,2);
d = 2*floor((m-1)/2)+1;
error = (A2-A1)/(2^(d+1)-1);
if abs(error) <= tol
    numI = A2 + error;
else
    medio = (a+b)/2;
    numI = AdapQNC(fname,a,medio,m,tol/2) + AdapQNC(fname,medio,b,m,tol/2);
end

```

Si la estimación heurística del error es mayor que `tol` se inician dos llamadas recursivas para obtener estimaciones:

$$\begin{cases} Q_L = \int_a^{\text{medio}} f(x) dx = I_L \\ Q_R = \int_{\text{medio}}^b f(x) dx = I_R \end{cases}$$

que satisfacen:

$$\begin{cases} |I_L - Q_L| \leq \frac{\text{tol}}{2} \\ |I_R - Q_R| \leq \frac{\text{tol}}{2} \end{cases}$$

Configurando, $Q = Q_L + Q_R$, vemos que:

$$|I - Q| = |(I_L - Q_L) + (I_R - Q_R)| \leq |I_L - Q_L| + |I_R - Q_R| \leq \frac{\text{tol}}{2} + \frac{\text{tol}}{2} = \text{tol}$$

El siguiente script muestra como usar la función `AdapQNC` para varios valores de `tol` y `m`:

```

% Archivo de Script: AplicacionAdapQNC
%
% Muestra el uso de la funcion AdapQNC para un rango de valores de tol y un
% rango de reglas de Newton-Cotes. Usaremos la función humps para el integrando
%
global FEvals VFEvals
close all
x = linspace(0,1,100);

```

```

y = humps(x);
plot(x,y)
u=[];
for tol = [.01 .001 .0001 .00001]
    for m=3:2:9
        figure
        plot(x,y)
        hold on
        title(sprintf('m = %d   tol = %e',m,tol))
        FEvals = 0;
        VFEvals = 0;
        num0 = AdapQNC('SpecHumps',0,1,m,tol);
        xlabel(sprintf('Scalar Evals = %d   Vector Evals = %d',FEvals,VFEvals))
        hold off
        u = [u; FEvals VFEvals];
    end
end
end

```

Obteniendo los siguientes resultados, resumidos en la siguiente tabla:

	$m = 3$	$m = 5$	$m = 7$	$m = 9$
tol= 0.01	104,26	98,14	60,6	26,2
tol= 0.001	216,54	154,22	60,6	26,2
tol= 0.0001	376,94	210,30	140,14	130,10
tol= 0.00001	696,174	322,46	260,26	182,14

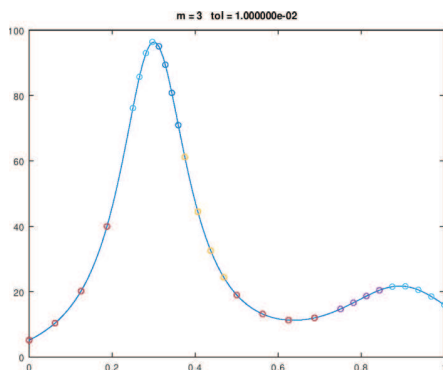


FIGURA 2.4.1. Gráfica del proceso adaptativo con $m = 3$.

EJERCICIOS

1. Obtenga una regla de Newton-Cotes abierta con 4 puntos.

Sugerencia

Se hace necesario calcular los pesos de la regla Newton-Cotes cerrada a través de la resolución de una sistema lineal adecuado. Considere que en la regla de m puntos se debe calcular la integral:

$$\int_0^1 x^{i-1} dx = \frac{1}{i}$$

exactamente para $i = 1 : m$. donde las abscisas vienen dadas por:

$$x_j = \frac{j-1}{m-1}, \quad j = 1 : m$$

de este modo los pesos w_1, w_2, \dots, w_m satisfacen la ecuación:

$$w_1 x_1^{i-1} + w_2 x_2^{i-1} + \dots + w_m x_m^{i-1} = \frac{1}{i}$$

para $i = 1 : m$

esto define un sistema lineal cuya solución es el vector de peso para la regla de m puntos.

2. Construya una función que calcule los pesos configurando el sistema lineal anterior y resuelva el sistema para $Xw = b$, de la forma típica y compare los resultados.

3. Construya una función que retorne un número entero n tal que el error vinculado a $Q_{NC}^{(n)}(m)$ es menor que `tol`.

4. Sea $C(x)$ el interpolante cúbico de Hermite de $f(x)$ en $[a, b]$. Desarrolle una regla compuesta uniformemente espaciada basada en esta regla de cuadratura.

Sugerencia

La regla de punto medio Q_1 en un intervalos para la integral:

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

esta definida por:

$$Q_1 = (b-a)f\left(\frac{a+b}{2}\right)$$

La regla de punto medio de dos intervalos para Q_2 viene dada por:

$$Q_2 = \frac{(b-a)}{2} \left(f\left(\frac{3a+b}{4}\right) + f\left(\frac{a+3b}{4}\right) \right)$$

5. Usando la heurística $|I - Q_2| < |Q_2 - Q_1|$, escribir una rutina para la cuadratura adaptativa que retorne una estimación de f que es más precisa que la tolerancia dada por `tol`.

Ecuaciones Diferenciales



Pierre Simon Laplace
(1749-1827)

Vemos... que la teoría de la probabilidad es en realidad únicamente el sentido común reducido a cálculo; nos hace apreciar con exactitud lo que las mentes razonadoras sienten por una especie de instinto, sin ser muchas veces capaces de explicarlo... Es notable que esta ciencia, que nació al estudiar los juegos de azar, haya venido a constituir el objeto más importante del conocimiento humano.

P. S. Laplace

3.1. Ecuaciones Diferenciales de Primer Orden

DEFINICIÓN 3.1. Una ecuación diferencial ordinaria tiene como característica esencial que la función desconocida depende sólo de una variable (de tal modo que las derivadas son derivadas ordinarias).

EJEMPLO 3.2. Con $\frac{dy}{dx} = 3x - 2y$ o bien $y' = 3x - 2y$ se asume que y es de variable independiente x , pero para la correcta edición en Mathematica es importante dejar claro quién opera como variable independiente, es decir:

$$y'[x] == 3 * x - 2 * y[x]$$

EJEMPLO 3.3. La función $y(x) - \cos(x) = \cos(x) + \sin(x)y(x)$, o bien $y(x) = \frac{\cos(x)}{1 - \sin(x)}$ es una solución a la EDO:

$$y'(x) + \sin(x) = \sin(x)y'(x) + y(x)\cos(x) \quad (1)$$

en efecto:

$$\frac{d\left(\frac{\cos(x)}{1 - \sin(x)}\right)}{dx} + \sin(x) = \sin(x) \frac{d\left(\frac{\cos(x)}{1 - \sin(x)}\right)}{dx} + \left(\frac{\cos(x)}{1 - \sin(x)}\right) \cos(x)$$

use Mathematica para comprobar, se sugiere sustituir en 1 :

$$\text{Solve}[\text{Dt}[y[x] - \text{Cos}[x] == y[x] * \text{Sin}[x], x], y'[x]]$$

= se usa para
identidades

$$\left\{ \left\{ y'(x) \rightarrow \frac{\sin(x) - y(x) \cos(x)}{\sin(x) - 1} \right\} \right\}$$

de esta manera:

$$\text{Simplify} \left[D \left[\frac{\text{Sin}[x] - y[x] * \text{Cos}[x]}{\text{Sin}[x] - 1}, x \right] + \text{Sin}[x] == \text{Sin}[x] * D \left[\frac{\text{Sin}[x] - y[x] * \text{Cos}[x]}{\text{Sin}[x] - 1}, x \right] + \text{Cos}[x] / (\text{Sin}[x] - 1) \right]$$

3.1.1. EDO de variable separable.

DEFINICIÓN 3.4. Toda EDO que se pueda escribir de la forma:

$$f(x) dx + g(y) dy = 0$$

se llamará *ecuación diferencial ordinaria de variable separable*, pues:

$$\int f(x) dx + \int g(y) dy = C$$

EJEMPLO 3.5. Dada $(x^3 - x) \frac{dy}{dx} - 2y = 0$, note que se puede expresar como:

$$\frac{dx}{x^3 - x} - \frac{dy}{2y} = 0$$

luego, usando el comando DSolve apropiadamente:

ECUACIÓN DIFERENCIAL

$$\text{DSolve} \left[\left(x^3 - x \right) \cdot y' [x] - 2 \cdot y [x] == 0, y [x], x \right]$$

IMPORTANTE
variable dependiente variable independiente

se tiene:

$$\text{DSolve}[(x^3 - x) * y'[x] - 2 * y[x] == 0, y[x], x]$$

$$\left\{ \left\{ y(x) \rightarrow c_1 e^{-2(\log(x) - \frac{1}{2} \log(1 - x^2))} \right\} \right\}, \text{ simplificando:}$$

$$\text{Simplify}[\text{DSolve}[(x^3 - x) * y'[x] - 2 * y[x] == 0, y[x], x]]$$

$$\left\{ \left\{ y(x) \rightarrow c_1 \left(\frac{1}{x^2} - 1 \right) \right\} \right\}$$

El comando *DSolve* devuelve la función explícita, y para utilizar la solución para cálculos posteriores se utilizará la regla de sustitución local /.sol[[1, 2, ..., n]].

EJEMPLO 3.6. Solucionar $x \frac{dy}{dx} - y = 2x^2 y$.

Aquí:

$$\text{DSolve}[x * y'[x] - y[x] == 2 * x^2 * y[x], y[x], x]$$

$$\left\{ \left\{ y(x) \rightarrow c_1 e^{x^2} x \right\} \right\}$$

Note que una ecuación diferencial de primer orden $y'(x) = f(x, y(x))$, genera una curva de nivel $f(x, y) = c$ donde c es una constante y define lo que se denomina *curva isoclina*.

```
sol = DSolve[(x^3 - x) * y'[x] - 2 * y[x] == 0, y[x], x];
```

Las constantes constantes arbitrarias se denotan por $C[1], C[2], \dots, C[n]$:

```
Plot[Table[(y[x]/.sol[[1]])/.C[1] -> c, {c, -3, 3, .5}]]/Evaluate, {x, -2, 2}, PlotStyle -> Black]
```

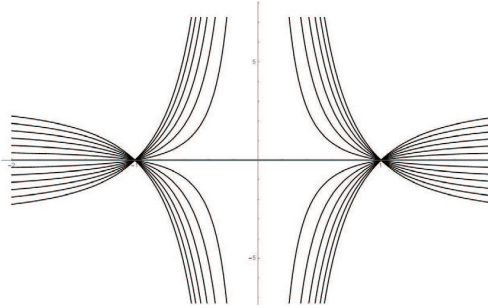


FIGURA 3.1.1. Isoclina 1

EJEMPLO 3.7. Para la EDO $y'(x) = -2x$ se plantea:

```
sol = DSolve[y'[x] == -2 * x, y[x], x];
```

```
Plot[Table[(y[x]/.sol[[1]])/.C[1] -> c, {c, -5, 5, 1}]]/Evaluate, {x, -5, 5}, PlotStyle -> Black]
```

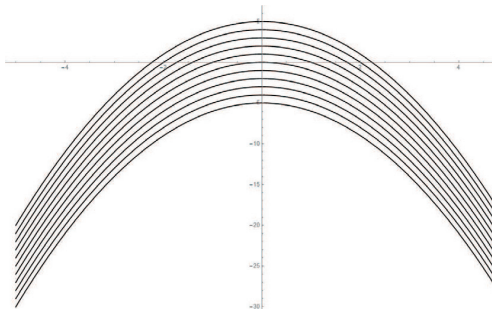


FIGURA 3.1.2. Isoclina 2

DEFINICIÓN 3.8. Problema de Valor Inicial:

Un problema de valor inicial es aquel que busca determinar una solución a una EDO sujeta a condiciones sobre la función desconocida y sus derivadas especificadas en un valor de la variable independiente. Tales condiciones se llaman condiciones iniciales.

$$\frac{dy}{dx} + a(x) = y \text{ con } y(x_0) = y_0$$

EJERCICIO 3.9. Resolver $\frac{dy}{dt} + \text{sen}(t)y = 0$ para $y(0) = \frac{3}{2}$

SOLUCIÓN. Se usará:

DSolve[[*y'*[*t*] + *Sen*[*t*] * *y*[*t*] == 0, *y*[0] == 3/2], *y*[*t*], *t*];

$$\left\{ \left\{ y(t) \rightarrow \frac{3}{2} e^{Cos(t)-1} \right\} \right\}$$

3.1.2. Homogéneas.

DEFINICIÓN 3.10. Una EDO de la forma $M(x, y) dx + N(x, y) = 0$ es *homogénea* si los coeficientes de $M(x, y)$ y $N(x, y)$ son funciones homogéneas del mismo grado, es decir:

Se dice que es de grado n . $M(tx, ty) = t^n M(x, y)$ y $N(tx, ty) = t^n N(x, y)$

EJEMPLO 3.11. La EDO $(x^2 - y^2) dx - (x^2 - xy) dy = 0$ es homogénea de grado 2, en efecto:

a. $M(tx, ty) = ((tx)^2 - (ty)^2) = t^2(x^2 - y^2)$.

b. $N(tx, ty) = ((tx)^2 - (tx)(ty)) = t^2(x^2 - xy)$.

Una apropiada sustitución transforma la EDO propuesta en una de variable separable, se probará con $x = vy$ y diferenciando de esta manera:

$$dx = vdy + ydv$$

así que:

$$\begin{aligned} (v^2y^2 - y^2)(vdy + ydv) - (v^2y^2 - vy^2)dy &= 0 \\ (v^3y^2 - v^2y^2)dy + (v^2y^3 - y^3)dv & \\ v^2y^2(v-1)dy + y^3(v^2-1)dv & \end{aligned}$$

y este proceso de sustitución condujo a la EDO de variable separable:

$$\frac{dy}{y} + \frac{(v+1)}{v^2}dv = 0$$

EJERCICIO 3.12. Resuelva $y' = \frac{y}{x} + \frac{y^2}{x^2}$ con $y(1) = 1$


SOLUCIÓN. Usando el comando indicado:

DSolve[[*y'*[*x*] == *y*[*x*]/*x* - *y*[*x*]^2/*x*^2, *y*[1] == 1], *y*[*x*], *x*]

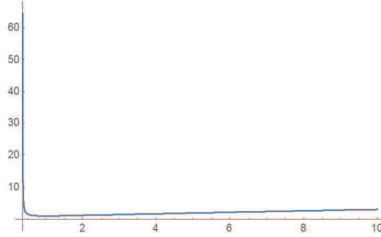
$$\left\{ \left\{ y(x) \rightarrow \frac{x}{\log(x)+1} \right\} \right\}$$

El comando *NDSolve* interpola el tamaño de la holgura hasta alcanzar una solución optima, el problema surge cuando la solución tiene una singularidad el cual lo indicará para tomar la acción pertinente.

sol = NDSolve[[*y'*[*x*] == *y*[*x*]/*x* - *y*[*x*]^2/*x*^2, *y*[1] == 1], *y*[*x*], {*x*, .37, 10}]

InterpolatingFunction[ Domain: (0.37 10.) Output: scalar](x)

Plot[Evaluate[y[x]/.sol], {x, .37, 10}, PlotRange -> All]



3.1.3. Exactas. La EDO de la forma $M(x, y) dx + N(x, y) = 0$ con $M(x, y)$ y $N(x, y)$ continuas y con derivadas parciales en un dominio $D \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}^2$ será *exacta* si existe una función f continua en toda x de D tal que:

$$M(x, y) = \frac{\partial f}{\partial x} \text{ y } N(x, y) = \frac{\partial f}{\partial y}$$

en consecuencia:

$$M(x, y) = \frac{\partial(\frac{\partial f}{\partial x})}{\partial y} = \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial(\frac{\partial f}{\partial y})}{\partial x} = N(x, y)$$

Así que:

$$f(x, y) = \int M(x, y) dx + g(y)$$

luego:

$$\frac{\partial f}{\partial y} = \frac{\partial(\int M(x, y) dx)}{\partial y} + g'(y) = N(x, y)$$

donde:

$$g'(y) = N(x, y) - \frac{\partial(\int M(x, y) dx)}{\partial y}$$

EJEMPLO 3.13. Usando lo expuesto al inicio de esta sección y Mathematica para resolver:

$$2xy dx + (x^2 - 1) dy = 0$$

se han de declarar $M(x, y)$ y $N(x, y)$ como elementos una matriz $F_{2 \times 1}$:

$$M[x, y] := 2 * x * y$$

$$P[x, y] := x^2 - 1$$

construyendo la matriz F :

$$F = \{ \{M[x, y]\}, \{P[x, y]\} \}$$

$$\begin{pmatrix} 2xy \\ x^2 - 1 \end{pmatrix}$$

a cada elemento de F se determina la respectiva integral:

$$\varphi_1 = \text{Integrate}[\{F[[1]]\} /. \{x \rightarrow u, y \rightarrow 0\}, \{u, 0, x\}]$$

$\varphi_2 = \text{Integrate}\{F[[2]]\} /. \{y \rightarrow u\}, \{u, 0, y\}$

$\varphi[x_-, y_-] := \varphi_1 + \varphi_2 == c$

$\text{Expand}[\varphi[x, y]]$

$$\left(x^2 y - y \right) = c$$

Para usar el comando *DSolve* es recomendado:

$\text{DSolve}[2 * x * y[x] + (x^2 - 1) * y'[x] == 0, y[x], x]$

$$\left\{ \left\{ y(x) \rightarrow \frac{c_1}{x^2 - 1} \right\} \right\}$$

En el caso que no se den las condiciones expresadas al inicio de esta sección es útil hallar un *factor integrante*.

La ecuación lineal en su forma standard:

$$\frac{dy}{dx} + yP(x) = Q(x)$$

se puede resolver multiplicando cada un miembro por $e^{\int P(x)dx}$ el cual se denominará *factor integrante*, en efecto:

$$\frac{d(ye^{\int P(x)dx})}{dx} = \frac{dy}{dx} e^{\int P(x)dx} + yP(x) e^{\int P(x)dx} = e^{\int P(x)dx} \left(\frac{dy}{dx} + yP(x) \right) = e^{\int P(x)dx} Q(x)$$

así que:

$$y(x) = \frac{\int e^{\int P(x)dx} Q(x) dx}{e^{\int P(x)dx}}$$

Para la que es denominada **ecuación de Bernoulli** $\frac{dy}{dx} + yP(x) = y^n Q(x)$ se procede de manera análoga pero habrá que modificar la ecuación por:

$$y^{-n} \frac{dy}{dx} + y^{1-n} P(x) = Q(x) \quad (1)$$

luego, si: $v = y^{1-n}$ y $dv = (1-n) y^{-n} dy$ o $dy = \frac{dv}{(1-n)y^{-n}}$

se obtiene:

$$\frac{y^{-n}}{(1-n)y^{-n}} \frac{dv}{dx} + vP(x) = Q(x)$$

equivalente a:

$$\frac{dv}{dx} + (1-n)vP(x) = (1-n)Q(x)$$

donde $e^{(1-n) \int P(x)dx}$ es un factor integrante, por consiguiente:

$$\begin{aligned} \frac{d(v e^{(1-n) \int P(x)dx})}{dx} &= v(1-n)P(x) e^{(1-n) \int P(x)dx} + \frac{dv}{dx} e^{(1-n) \int P(x)dx} = \\ e^{(1-n) \int P(x)dx} \left(\frac{dv}{dx} + v(1-n)P(x) \right) &= e^{(1-n) \int P(x)dx} (1-n)Q(x) \end{aligned}$$

finalmente:

$$y^{1-n} e^{(1-n) \int P(x)dx} = (1-n) \int e^{(1-n) \int P(x)dx} Q(x)$$

3.1.4. Teorema de la existencia y unicidad. Dada $f(x, y)$ para la que existe $\frac{\partial f}{\partial y}$, ambas continuas en un rectángulo:

$$R : t_0 \leq t \leq t_0 + \alpha, |y - y_0| \leq b$$

entonces el problema de valor inicial $y' = f(t, y); y(t_0) = y_0$ tiene solución única en:

$$t_0 \leq t \leq t_0 + \alpha.$$

DEMOSTRACIÓN. Suponga que $z(t)$ es una segunda solución de y' , entonces: □

$$y(t) = y_0 + \int_{t_0}^t f(s, y(s)) ds \text{ y } z(t) = y_0 + \int_{t_0}^t f(s, z(s)) ds$$

luego:

$$\begin{aligned} |y(t) - z(t)| &= \left| \int_{t_0}^t [f(s, y(s)) - f(s, z(s))] ds \right| \leq \int_{t_0}^t |f(s, y(s)) - f(s, z(s))| ds \\ &\leq L \int_{t_0}^t |f(s, y(s)) - f(s, z(s))| ds \end{aligned}$$

donde L indica el máximo de los valores de $\left| \frac{\partial f}{\partial y} \right|$ para (t, y) en R y por ser $z(t) \leq L \int_{t_0}^t z(s) ds$ entonces $z(t)$ es nula, por lo tanto el problema tiene solución única.

Con relación al teorema que comienza esta sección se pueden discutir los siguientes aspectos al resolver una ecuación diferencial de valor inicial:

EXISTENCIA	UNICIDAD
$y' = f(x, y)$ ¿tiene soluciones? $(x_0, f(x_0))$ ¿es solución?	¿Existe una sola curva que tenga como solución a $(x_0, f(x_0))$

EJEMPLO 3.14. Para:

$$\frac{dy}{dx} = 3\sqrt[3]{y^2}, y(2) = 0$$

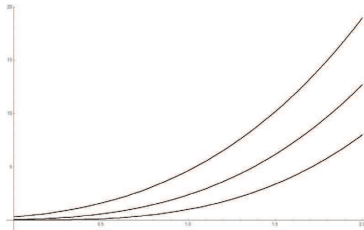
aquí $F(x, y) = 3\sqrt[3]{y^2}$ y $\frac{\partial F}{\partial y} = \frac{2}{y^{1/3}}$ aunque no puede garantizar que $y(2) = 0$ sea única.

Usando Mathematica:

sol1 = Factor[DSolve[y'[x] == 3 * y[x]^(2/3), y[x], x]]

$\left\{ \left\{ y(x) \rightarrow \frac{1}{27} (c_1 + 3x)^3 \right\} \right\}$ que es la solución general.

Plot[Evaluate[Table[y[x]/.sol1[[1]]/.C[1] → c, {c, 0, 2, 1}], {x, 0, 2}, PlotStyle → Black]



observe ahora usando el valor inicial:

sol2 = DSolve[{y'[x] == 3 * y[x]^(2/3), y[2] == 0}, y[x], x]

$\left\{ \left\{ y(x) \rightarrow x^3 - 6x^2 + 12x - 8 \right\} \right\}$

MatrixForm[Table[{x, y[x]/.sol2}, {x, 0, 2, .5}]]

$$\begin{pmatrix} 0. & -8. \\ 0.5 & -3.375 \\ 1. & -1. \\ 1.5 & -0.125 \\ 2. & 0. \end{pmatrix}$$

Considere el caso $\frac{dy}{dx} = 3\sqrt[3]{y^2}$, $y(0) = 0$

3.1.5. Métodos numéricos. Método de Euler.

El método de Euler o método de las tangentes es una técnica simple basada en el comportamiento de los incrementos diferenciales, suponga que se quiere aproximar:

$$y' = f(x, y) \text{ con } y(x_0) = y_0$$

es decir:

$$y'_0 = \frac{y_1 - y_0}{(x_0 + h) - x_0} \text{ o } y_1 = y_0 + h y'_0$$

Se usó: $h = x_1 - x_0$.

así:

$$y_{n+1} = y_n + h y'_n$$

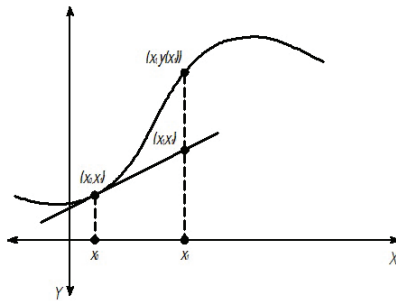
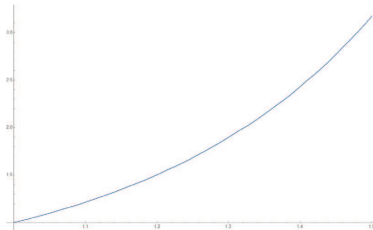


FIGURA 3.1.3. Tangente.

EJEMPLO 3.15. Usando el método de Euler para hallar el valor aproximado de y cuando $y(1) = 1$ y $\frac{dy}{dx} = 2xy$ y $h = 0.05$, se tiene:

```
s = NDSolve[{y'(x) = 2xy(x), y(1) = 1}, y(x), {x, 1, 1.5},
"ExplicitEuler", "StartingStepSize" -> .05]
```



```
MatrixForm[Table[{x, y[x]}/.s[[1]], {x, 1, 1.5, .05}]]//N
```

1.	1.
1.05	1.1
1.1	1.2155
1.15	1.34921
1.2	1.50436
1.25	1.68489
1.3	1.8955
1.35	2.14191
1.4	2.43107
1.45	2.77142
1.5	3.17328

Las Aproximaciones Sucesivas de Picard.

El problema de valor inicial:

$$y' = f(x, y) \text{ con } y(x_0) = y_0$$

y $f(x, y)$ continua en una región que contiene el punto (x_0, y_0) , se tiene:

$$y(x) = c + \int_{x_0}^x f(x, y) dx \text{ y es obvio que } y(x_0) = c \text{ o } y_0 = c$$

así que:

$$\begin{aligned} y(x) &= y_0 + \int_{x_0}^x f(x, y) dx \\ y_1(x) &= y_0 + \int_{x_0}^x f(x, y_0) dx \\ y_2(x) &= y_0 + \int_{x_0}^x f(x, y_1) dx \\ &\vdots \\ y_n(x) &= y_0 + \int_{x_0}^x f(x, y_{n-1}) dx, \quad n = 1, 2, 3, \dots \end{aligned}$$

EJEMPLO 3.16. Para $y' = x - y$ con $y(0) = 1$, se tiene:

$$\begin{aligned} y_1(x) &= 1 + \int_0^x (x-1) dt = \frac{1}{2}x^2 - x + 1 \\ y_2(x) &= 1 + \int_0^x \left(-\frac{1}{2}x^2 + 2x - 1\right) dt = -\frac{1}{6}x^3 + x^2 - x + 1 \\ y_3(x) &= 1 + \int_0^x \left(\frac{1}{6}x^3 - x^2 + 2x - 1\right) dt = \frac{1}{24}x^4 - \frac{1}{3}x^3 + x^2 - x + 1 \\ y_4(x) &= 1 + \int_0^x \left(-\frac{1}{24}x^4 + \frac{1}{3}x^3 - x^2 + 2x - 1\right) dt = -\frac{1}{120}x^5 + \frac{1}{12}x^4 - \frac{1}{3}x^3 + x^2 - x + 1 \end{aligned}$$

La usando Mathematica se sugiere:

```
PicardIteration[x0_, y0_, max_] := Module[{f}, Y0[x_] = y0; k = 0;
Print [Y0, "(x)=", Y0[x]]; While [k < max, Module[{f}, Y1[x_] = y0 + Integrate [f [t, Y0[t]], {t, x0, x}];
Y1[x_] = Normal [Series [Y1[x], {x, x0, 100}]]; k = k + 1; Print [Yk, "(x)=", Y1[x]];
Y0[x_] = Y1[x]]]
```


se inserta $f(x, y)$ al final de las líneas y se evalúa el cuaderno:

$$f[x.,y.]:=x - y$$

PicardIteration[0, 1, 4]

$$Y_0(x)=1$$

$$Y_1(x)=\frac{x^2}{2} - x + 1$$

$$Y_2(x)=-\frac{x^3}{6} + x^2 - x + 1$$

$$Y_3(x)=\frac{x^4}{24} - \frac{x^3}{3} + x^2 - x + 1$$

$$Y_4(x)=-\frac{x^5}{120} + \frac{x^4}{12} - \frac{x^3}{3} + x^2 - x + 1$$

y una aproximación para $x = .001$ para las 4 iteraciones dadas será:

$$0.9901$$

Método de Runge - Kutta.

Este método es considerado uno de los más exactos, los alemanes Carl Runge (1856-1927) y Wilhelm Kutta (1867-1944) plantearon el método a partir de $y(x) = y_0 + \int_{x_0}^x f(x, y) dx$ y el desarrollo de Taylor de $y = g(x)$ para (x_0, y_0) :

$$y = g(x_0) + (x - x_0)g'(x_0) + \frac{1}{2}(x - x_0)^2g''(x_0) + \frac{1}{6}(x - x_0)^3g'''(x_0) + \dots \quad (1)$$

luego:

$$\begin{aligned} y' &= g'(x) = f(x, y) \\ y'' &= g''(x) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \\ y''' &= g'''(x) = \frac{d(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f)}{dx} = \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} \right) \left(\frac{\partial f}{\partial x} + f \frac{\partial f}{\partial y} \right) = \\ &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} + 2f \frac{\partial^2 f}{\partial x \partial y} + f \left(\frac{\partial f}{\partial y} \right)^2 + f^2 \frac{\partial^2 f}{\partial y^2} \end{aligned} \quad (2)$$

⋮

de (1), (2) y haciendo:

$$p = \frac{\partial f}{\partial x}, q = \frac{\partial f}{\partial y}, r = \frac{\partial^2 f}{\partial x^2}, s = \frac{\partial^2 f}{\partial x \partial y}, t = \frac{\partial^2 f}{\partial y^2} \text{ se obtiene:}$$

$$y = y_0 + hf_0 + \frac{1}{2}h^2(p_0 + f_0q_0) + \frac{1}{6}h^3(r_0 + p_0q_0 + 2f_0s_0 + f_0q_0^2 + f_0^2t_0) + \dots$$

ahora, de 1 y $y(x) = y_0 + \int_{x_0}^{x_0+h} f(x, y) dx$ se observa que:

$$\begin{aligned} k &= y - y_0 = y_0 + \int_{x_0}^{x_0+h} f(x, y) dx = \\ &= hf_0 + \frac{1}{2}h^2(p_0 + f_0q_0) + \frac{1}{6}h^3(r_0 + p_0q_0 + 2f_0s_0 + f_0q_0^2 + f_0^2t_0) + \dots \end{aligned}$$

y como sólo se conoce y_0 entonces el método se traduce en aproximaciones de la siguiente manera:

$$\begin{aligned} k_1 &= hf_0 \\ k_2 &= hf(x_0 + h, y_0 + k_1) \\ k_3 &= hf(x_0 + h, y_0 + k_2) \\ k_4 &= hf(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1) \\ &\vdots \end{aligned}$$

$$k \approx \frac{1}{6} (k_1 + 4k_4 + k_3)$$

Planteando una rutina sería:

```
Clear[x0,y0,xn,m0];
Runge[x0_,y0_,xn_,m0_]:=Module[{a=x0,b=xn,j=m,m0},h=(b-a)/m;
Y=T=Table[0,{m+1}];T[[1]]=a;Y[[1]]=y0;
For[j=1,j<=m,j++,k1=h*f[T[[j]],Y[[j]]];
k2=h*f[T[[j]]+h/2,Y[[j]]+k1/2];k3=h*f[T[[j]]+h/2,Y[[j]]+k2/2];
k4=h*f[T[[j]]+h,Y[[j]]+k3];Y[[j+1]]=Y[[j]]+(1/6)*(k1+2*k2+2*k3+k4);
T[[j+1]]=a+h*j];sol=Transpose[{T,Y}];Return[sol];
```

Ahora, si $y' = (x - y)^2$ con $y(0) = 0.5$ se puede hallar $y(0.5)$, observe:

```
f[x_,y_-:=(x-y)^2;
```

```
x0=0
```

```
y0=.5
```

```
xn=.5
```

```
m=5
```

```
Prepend [Table[Runge[x0,y0,xn,m]],{"x_n","y_n"}]//N
```

$$\begin{pmatrix} x_n & y_n \\ 0. & 0.5 \\ 0.1 & 0.521329 \\ 0.2 & 0.53576 \\ 0.3 & 0.544267 \\ 0.4 & 0.548207 \\ 0.5 & 0.549267 \end{pmatrix}$$

3.1.6. Aplicaciones.

Ley de Crecimiento Poblacional de Malthus.

Si una población está aislada, es decir si no existe emigración o inmigración, además si se representa su comportamiento en un tiempo como la diferencia entre la natalidad y la mortalidad por $p(t + \Delta t) - p(t)$, entonces sus tasas promedios se pueden definir como $k_1 = \frac{p(t+\Delta t)}{p(t)\Delta t}$ y $k_2 = \frac{p(t)}{p(t)\Delta t}$, entonces:

$\frac{p(t+\Delta t)-p(t)}{\Delta t} = \frac{p(t+\Delta t)}{\Delta t} - \frac{p(t)}{\Delta t} = (k_1 - k_2)p(t) = kp(t)$, así que cuando $\Delta t \rightarrow 0$ se usará la EDO:

$$\frac{dp}{dt} = kp(t)$$

EJEMPLO 3.17. Según el organismo responsable por la administración de la salud de un país la población contagiada por virus del zika asciende a un 20% anual durante el 2010 y 2017. Para enero de 2011 ese país tuvo 35 millones de habitantes.

Se declara P_0 por:

$$P_0 = 3.34 * 10^9$$

`Simplify[DSolve[{P'[t] == k * P[t], P[t0] == P0}, P[t], t]]`

$$\{\{P(t) \rightarrow 3.34 \times 10^9 e^{k(t-1.0)}\}\}$$

`DSolve[{P'[t] == .02 * P[t], P[1965] == 3.34 * 10^9}, P[t], t]`

$$\{\{P(t) \rightarrow 2.8574150512182982 * 10^8 * e^{0.02t}\}\}$$

$$\text{Pobl}[t].:=2.857415051218298 * 10^8 - 8 * E^{(0.02 * t)}$$

La población de infectados se duplicará en el año:

`Solve[Pobl[t] == 2 * P0, t, Reals]`

$$\{\{t \rightarrow 1999.66\}\}$$

`Plot[Pobl[t], {t, 0, 100}]`

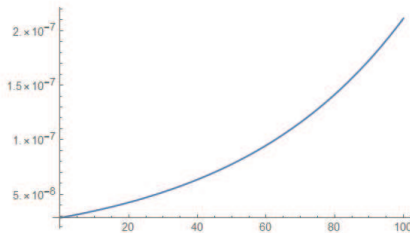


FIGURA 3.1.4. Malthus.

finalmente:

$$1999.66 - 1965 = 34.66$$

o bien aproximadamente 34 años y 8 meses.

Mezclas de Soluciones.

Se disuelven 20 Kg de cloro en 500 lt de agua y posteriormente se vierte en un tanque a razón de 10 lt por minuto, luego dicha solución se bombea fuera del tanque a razón de 10 lt por minuto, Si la concentración del líquido que entra es de 2 Kg/lt, se puede hallar cuánto cloro permanece después cierto tiempo, en efecto:

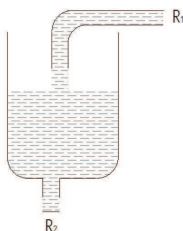


FIGURA 3.1.5. Mezclas.

sea $A(t)$ la cantidad de cloro que hay en el tanque, entonces:

$$\frac{dA}{dt} = R_1 - R_2$$

$$\frac{dA}{dt} = (10 \text{ lt/min})(2 \text{ Kg/lt}) - (10 \text{ lt/min})\left(\frac{A}{500} \text{ Kg/lt}\right) = \left(20 - \frac{A}{50}\right) \text{ Kg/min}$$

o bien:

$$\frac{dA}{dt} = 20 - \frac{A}{50}$$

así que:

$$A(t) = 1000 + ce^{-t/50}$$

Ley del Enfriamiento de Newton.

La rapidez con que se enfría un objeto es proporcional a la diferencia de su temperatura y la del medio que le rodea.

$$\frac{dT}{dt} = k(T - T_m)$$

Use $T(t)$ como la temperatura del cuerpo para un determinado momento t , T_m la temperatura circundante, $\frac{dT}{dt}$ representa la rapidez con que se enfría el cuerpo y k es la constante de proporcionalidad.

Si el objeto se enfría se ha de cumplir que $T > T_m$ entonces $k < 0$.

PROBLEMA 3.18. Después de hornear un perril a $150^\circ C$ se retira del horno y a los 10 minutos baja su temperatura a $120^\circ C$. Cuánto tiempo tardará en enfriarse a la temperatura ambiente de $30^\circ C$.

SOLUCIÓN. Se usará:

`DSolve[{T'[t] == k * (T[t] - 31.2), T[0] == 150}, T[t], t]`

`{ {T(t) -> 118.8 (0.262626 + e^{kt}) }`

`Temp = 118.8 * (0.26262626262626265 + e^{k*t}) == 120/.t -> 5//N`

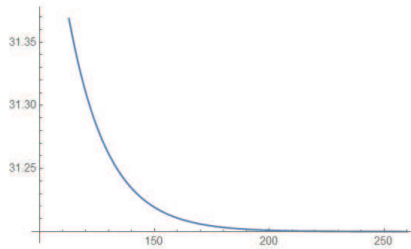
`118.8 (0.262626 + 2.71828^{5.k}) = 120.`

`Solve[%, k]`

`{ {k -> -0.058211} }`

`Tf[t.]:=118.8 * (0.26262626262626265 + e^{-0.058210951386084044*t})`

`Plot[Tf[t], {t, 100, 260}]`



`MatrixForm[Table[{t, Tf[t]}, {t, 252.2, 252.21, .001}]]`

$$\begin{pmatrix} 252.2 & 31.2001 \\ 252.201 & 31.2001 \\ 252.202 & 31.2001 \\ 252.203 & 31.2 \\ 252.204 & 31.2 \\ 252.205 & 31.2 \\ 252.206 & 31.2 \\ 252.207 & 31.2 \\ 252.208 & 31.2 \\ 252.209 & 31.2 \\ 252.21 & 31.2 \end{pmatrix}$$

Se puede afirmar que el perfil estará a temperatura ambiente en 4,2 horas aproximadamente.

El Marcapasos Cardíaco.

Un marcapasos está formado esencialmente por conmutador, una batería y un condensador. Cuando el conmutador S está en A el condensador se carga, pero cuando está en B , el órgano recibe un estímulo eléctrico y como el corazón en ese instante ofrece una resistencia eléctrica, el condensador es descarga activando el conmutador para un nuevo ciclo de carga lo que garantizará la regularidad del latido.

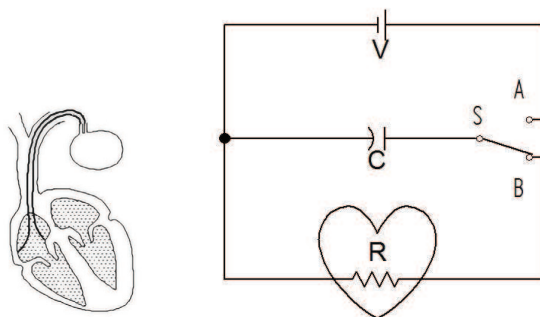


FIGURA 3.1.6. Marcapasos.

El voltaje V que se aplica al corazón está determinado por:

$$\frac{dV}{dt} = -\frac{1}{RC}V$$

editando los comandos apropiados:

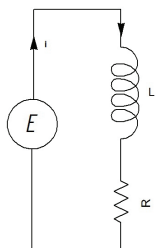
Simplify[DSolve[{V'[t] == (-1/R * C) * V[t], V[t0] == V0}, V[t], t]]

$$\left\{ \left\{ V(t) \rightarrow V_0 e^{\frac{C(t_0-t)}{R}} \right\} \right\}$$

Circuitos Eléctricos.

En la figura anexa se tiene la ecuación fundamental aplicada a los circuitos:

$$L \frac{di}{dt} + Ri = E(t) \quad (1)$$



L : Inductancia.

R : Resistencia.

i : Corriente.

E : Fuerza electromotriz (fem).

La ecuación 1 se puede escribir:

$$\frac{di}{dt} + \frac{R}{L}i = \frac{E(t)}{L}$$

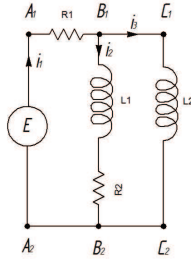
y es de la forma:

$$\frac{dy}{dx} + yP(x) = Q(x)$$

por lo tanto $e^{\frac{R}{L} \int dt} = e^{\frac{R}{L}t}$ es su factor integrante, así que:

$$i = \frac{E}{R} + Ce^{-\frac{Rt}{L}}$$

EJEMPLO 3.19. Dado el circuito eléctrico:



La corriente i_1 se ramifica en B_1 , entonces según la primera ley de *Kirchoff* se tiene:

$$i_1 = i_2 + i_3 \quad (a)$$

se debe considerar el recorrido $A_1 \rightarrow B_1 \rightarrow B_2 \rightarrow A_2$ pues la ecuación diferencial será:

$$E(t) = i_1 R_1 + L_1 \frac{di_2}{dt} + i_2 R_2 \quad (b)$$

de manera análoga el recorrido $A_1 \rightarrow B_1 \rightarrow C_1 \rightarrow C_2 \rightarrow B_2 \rightarrow A_2$ sugiere la ecuación diferencial:

$$E(t) = i_1 R_1 + L_2 \frac{di_3}{dt} \quad (c)$$

finalmente de las partes a, b y c se obtiene el sistema:

$$\begin{cases} L_1 \frac{di_2}{dt} + (i_2 + i_3) R_1 + i_2 R_2 = E(t) \\ L_2 \frac{di_3}{dt} + (i_2 + i_3) R_1 = E(t) \end{cases}$$

3.2. Ecuaciones Diferenciales de Segundo Orden

DEFINICIÓN 3.20. Una *ecuación diferencial lineal de segundo orden* es de la forma:

$$a_2(x) \frac{d^2y}{dx^2} + a_1(x) \frac{dy}{dx} + a_0(x)y = g(x)$$

y si se considera determinar una solución sujeta a condiciones sobre la función desconocida especificadas en dos o más valores de la variable independiente, tales condiciones son llamadas *condiciones de frontera*, es decir:

$$a_2(x) \frac{d^2y}{dx^2} + a_1(x) \frac{dy}{dx} + a_0(x)y = g(x) \text{ sujeta a } y(a) = y_0 \text{ y } y(b) = y_1$$

EJEMPLO 3.21. *Segunda Ley de Newton.*

La tasa de variación del momentum de un cuerpo en función del tiempo es proporcional a la fuerza neta que actúa sobre el cuerpo teniendo la misma dirección de la fuerza.

Dicho de otra manera, esta ley establece que la aceleración $\frac{d^2y}{dt^2}$ que experimenta un objeto es proporcional a la fuerza que actúa sobre él:

Momentum es el producto de la masa m por la velocidad v .

$$\frac{d^2y}{dt^2} = \frac{F}{m}$$

en nuestro caso:

$$m \frac{d^2y}{dt^2} = F\left(t, y, \frac{dy}{dt}\right)$$

donde se desprenden las condiciones:

$$y(t_0) = y_0 \text{ y } y'(t_0) = y'_0$$

Así que la aceleración instantánea será $a = \frac{dv}{dt}$ y la velocidad instantánea es entonces $v = \frac{dx}{dt}$, si ahora el cuerpo experimenta una caída libre la fuerza tendrá magnitud mg , luego:

$$\frac{dx^2}{dt^2} = g$$

EJEMPLO 3.22. *Descenso en paracaídas.*

Usando la segunda ley de Newton se puede determinar la ecuación diferencial de la velocidad de un cuerpo que cae colgado de un paracaídas.

Como el movimiento del cuerpo es descendente, la resistencia del aire es proporcional a la velocidad instantánea, y considerando este movimiento como una dirección positiva se obtiene:

$$mg - kv$$

por ser:

$$m \frac{d^2y}{dt^2} = F$$

es decir:

$$m \frac{d^2y}{dt^2} = mg - kv$$

o bien:

$$m \frac{dv}{dt} = mg - kv$$

finalmente:

$$\frac{dv}{dt} + \frac{k}{m}v = g$$

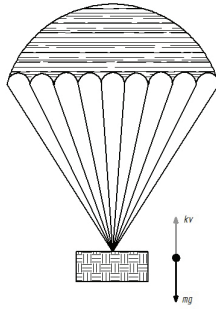


FIGURA 3.2.1. Paracaídas.

EJEMPLO 3.23. *Catenaria.*

En la forma que adopta un cable uniforme suspendido en sus extremos intervienen su peso (W), una fuerza horizontal (H) una tensión (T), por lo tanto:

$$H = T \cos(\phi), \quad W = T \sin(\phi), \quad \tan(\phi) = \frac{dy}{dx} = \frac{W}{H}$$

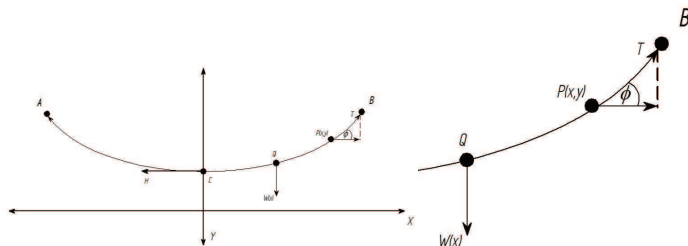


FIGURA 3.2.2. Catenaria.

así que:

$W = ws$ y s es la longitud de arco en un punto del cable.

$$\frac{d^2y}{dx^2} = \frac{1}{H} \frac{dW}{dx} = \frac{w}{H} \frac{ds}{dx} = \frac{w}{H} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}$$

si se hace $\frac{dy}{dx} = p$ queda:

$$\frac{dp}{dx} = \frac{w}{H} \sqrt{1 + p^2}$$

que es una EDO de variable separable, en efecto:

$$\frac{dp}{\sqrt{1+p^2}} = \frac{w}{H} dx$$

lo que conduce a:

$$a \operatorname{senh}(p) = \frac{w}{H} x \quad \text{o} \quad p = \operatorname{senh}\left(\frac{w}{H} x\right)$$

de 1 se deduce:

$$\frac{dy}{dx} = \operatorname{senh}\left(\frac{w}{H} x\right) \quad \text{o} \quad dy = \operatorname{senh}\left(\frac{w}{H} x\right) dx$$

finalmente:

$$y = \frac{H}{w} \left(\cosh\left(\frac{w}{H} x\right) - 1 \right)$$

EJEMPLO 3.24. *Vigas empotradas.*

Una viga horizontal de carga uniforme w y de l metros de longitud está empotrada en ambos extremos con Kwl^2 en paredes de concreto. Si se usa a O como el origen flecha, se tiene un empuje vertical hacia arriba de $\frac{1}{2}wl$ y una carga de wx en el punto medio de OP .

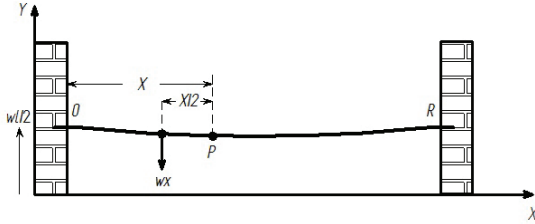


FIGURA 3.2.3. Viga empotrada.

El momento flector se puede escribir por:

$$M = EI \frac{d^2y}{dx^2} = K + \frac{1}{2}wx - \frac{1}{2}wx^2$$

donde E es el *Módulo de Young* el cual considera la elasticidad del material que está hecha la viga e I es el momento de inercia de la sección trasversal y el producto EI se denomina *rigidez*.

En O :

si $y'(0) = 0$ se tiene:

$$EI \frac{dy}{dx} = Kx + \frac{1}{4}wx^2 - \frac{1}{6}wx^3$$

En R :

si $y'(l) = 0$ se obtienen:

$$Kl + \frac{1}{4}wl^3 - \frac{1}{6}wx^3 = 0, K = -\frac{1}{12}wl^2 \text{ y } EI \frac{dy}{dx} = -\frac{1}{12}wl^2x + \frac{1}{4}wx^2 - \frac{1}{6}wx^3$$

Usando ahora $y(0) = 0$ en O :

$$EIy = -\frac{1}{24}wl^2x^2 + \frac{1}{12}wx^3 - \frac{1}{24}wx^4 \text{ o bien:}$$

$$y = \frac{wx^2}{24EI} (2lx - l^2 - x^2)$$

finalmente, la flecha máxima o máxima deformación vertical en $x = \frac{1}{2}l$ (punto medio de la viga) será:

$$-y_{max} = \frac{wl^4}{384EI}$$

Suponga ahora que en el punto medio interviene un peso W , entonces se tiene que el empuje vertical será $\frac{1}{2}(wl + W)$, por lo tanto:

$$EI \frac{d^2y}{dx^2} = K + \frac{1}{2}(wl + W) - \frac{1}{2}wx^2 = K + \frac{1}{2}wlx - \frac{1}{2}wx^2 + \frac{1}{2}Wx$$

a. Para $0 < x < \frac{1}{2}l$.

Si $y'(0) = 0$ en O se tiene:

$$EI \frac{dy}{dx} = Kx + \frac{1}{4}wlx^2 - \frac{1}{6}wx^3 + \frac{1}{4}Wx^2 \quad (1)$$

usando ahora $y(0) = 0$:

$$EIy = \frac{1}{2}Kx^2 + \frac{1}{12}wlx^3 - \frac{1}{24}wx^4 + \frac{1}{12}Wx^3 \quad (2)$$

b. Para $\frac{1}{2}l < x < l$.

$$EI\frac{d^2y}{dx^2} = K + \frac{1}{2}wlx - \frac{1}{2}wx^2 + \frac{1}{2}Wx - W\left(x - \frac{1}{2}l\right), \text{ luego:}$$

$$EIy = \frac{1}{2}Kx^2 + \frac{1}{12}wlx^3 - \frac{1}{24}wx^4 + \frac{1}{12}Wx^3 - \frac{1}{6}W\left(x - \frac{1}{2}l\right)^3 + C_1x + C_2$$

es decir:

$$EIy = \frac{1}{2}Kx^2 + \frac{1}{12}wlx^3 - \frac{1}{24}wx^4 + \frac{1}{12}Wx^3 - \frac{1}{6}W\left(x - \frac{1}{2}l\right)^3 \quad (3)$$

con $C_1 = C_2 = 0$

Considerando $y'(\frac{1}{2}l) = 0$ en 1:

$$EI\frac{dy}{dx} = \frac{1}{2}lK + \frac{1}{16}wl^3 - \frac{1}{48}wl^3 + \frac{1}{16}Wl^2 = 0, K = -\frac{1}{12}wl^2 - \frac{1}{8}Wl$$

sustituyendo en 2 y 3:

Para $0 < x < \frac{1}{2}l$.

$$EIy = -\frac{1}{24}wl^2x^2 + \frac{1}{12}wlx^3 - \frac{1}{24}wx^4 + \frac{1}{12}Wx^3 - \frac{1}{16}Wlx^2$$

$$y = \frac{w}{24EI}(2lx^3 - l^2x^2 - x^4) + \frac{W}{48EI}(4x^3 - 3lx^2)$$

Para $\frac{1}{2}l < x < l$.

$$EIy = -\frac{1}{24}wl^2x^2 + \frac{1}{12}wlx^3 - \frac{1}{24}wx^4 + \frac{1}{12}Wx^3 - \frac{1}{6}W\left(x - \frac{1}{2}l\right)^3 - \frac{1}{16}Wlx^2$$

$$y = \frac{w}{24EI}(2lx^3 - l^2x^2 - x^4) + \frac{W}{48EI}(l^3 - 6l^2x + 9lx^2 - 4x^3)$$

finalmente la flecha máxima en el punto medio es:

$$-y_{max} = \frac{wl^4 + 2Wl^3}{384EI}$$

3.2.1. Ecuaciones diferenciales homogéneas con coeficientes constantes.

DEFINICIÓN. Una ecuación diferencial homogénea con coeficientes constantes es de la forma:

$$a_0 \frac{d^n y}{dx^n} + a_1 \frac{d^{n-1} y}{dx^{n-1}} + a_2 \frac{d^{n-2} y}{dx^{n-2}} + \dots + a_{n-1} \frac{dy}{dx} + a_n y = 0$$

donde $\{a_0, a_1, \dots, a_n\}$ son constantes y $a_0 \neq 0$.

En este tipo de ED se puede usar la notación $\frac{d^n y}{dx^n} = D^n y$, es decir:

$$2 \frac{d^2 y}{dx^2} - 5 \frac{dy}{dx} - 3y = 0$$

es equivalente a:

$$(2D^2 - 5D - 3)y = 0$$

en efecto:

$$\begin{aligned} (2D^2 - 5D - 3)y &= (D - 3)(2D + 1)y = 0 \\ (D - 3)\left(2 \frac{d}{dx} + 1\right)y &= (D - 3)\left(2 \frac{dy}{dx} + y\right) = \frac{d\left(2 \frac{dy}{dx} + y\right)}{dx} - 3\left(2 \frac{dy}{dx} + y\right) = \\ 2 \frac{d^2 y}{dx^2} + \frac{dy}{dx} - 6 \frac{dy}{dx} - 3y &= 2 \frac{d^2 y}{dx^2} - 5 \frac{dy}{dx} - 3y = 0 \end{aligned}$$

por otra parte la ecuación:

$F(D) = (D - m_1)(D - m_2) \dots (D - m_n) = 0$ se denomina ecuación característica y a $\{m_1, m_2, \dots, m_n\}$ se les denomina valores característicos; ahora suponga que:

a. $m_1 \neq m_2 \neq \dots \neq m_n$, en este caso la solución general es:

$$y(x) = C_1 e^{m_1 x} + C_2 e^{m_2 x} + \dots + C_n e^{m_n x}$$

b. $m = m_1 = m_2 = \dots = m_n$ la solución general será:

$$y(x) = C_1 e^{mx} + C_2 x e^{mx} + C_3 x^2 e^{mx} + \dots + C_n x^{n-1} e^{mx}$$

c. $m = a \pm bi$ la solución general es:

$$y(x) = e^{ax} (C_1 \cos(bx) + C_2 \operatorname{sen}(bx))$$

EJEMPLO 3.25. Para la solución general de $\frac{d^2 x}{dt^2} + 16x = 0$ se sugiere:

$$m^2 + 16 = 0 \text{ o bien } m = \pm 4i$$

entonces:

$$x(t) = C_1 \cos(4t) + C_2 \operatorname{sen}(4t)$$

Usando Mathematica:

`DSolve[x''[t] + 16 * x[t]==0, x[t], t]`

$$\{\{x(t) \rightarrow c_2 \sin(4t) + c_1 \cos(4t)\}\}$$

aplicando las condiciones iniciales:

$$x(0) = 10 \text{ y } x'(0) = 0$$

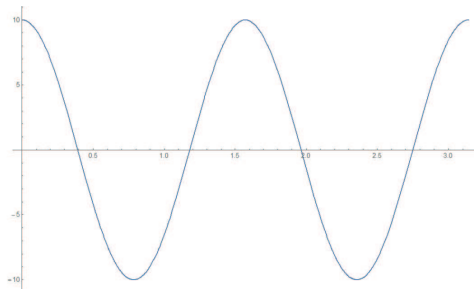
`DSolveValue[\{x''[t] + 16 * x[t]==0, \{x'[0]==0, x[0]==10\}\}, x[t], t]`

$$10 \cos(4t)$$

Sea:

```
s1 = NDSolve[{x''[t] + 16 * x[t]==0, {x'[0] == 0, x[0] == 10}}, x[t], {t, Pi}]
```

```
Plot[Evaluate[x[t]/s1], {t, 0, Pi}]
```



EJERCICIO 3.26. Determine la solución general para $\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + 10x = 0$ y además:

- Ofrezca una solución particular para $x'(0) = 0$ y $x(0) = -2$.
- Tabule los resultados con $h = 0.5$.
- Compare gráficamente los resultados obtenidos y el método de Euler con $h = 0.1$.

SOLUCIÓN. Aquí:

```
DSolve[x''[t] + 2 * x'[t] + 10 * x[t]==0, x[t], t]
```

```
{ {x(t) -> c1 e^{-t} sin(3t) + c2 e^{-t} cos(3t)} }
```

```
a. DSolveValue[{x''[t] + 2 * x'[t] + 10 * x[t]==0, {x'[0] == 0, x[0] == -2}}, x[t], t]
```

```
-\frac{2}{3} e^{-t} (\sin(3t) + 3 \cos(3t))
```

```
b. s2 = NDSolve[{x''[t] + 2 * x'[t] + 10 * x[t]==0, {x'[0] == 0, x[0] == -2}}, x[t], {t, 2 * Pi}]
```

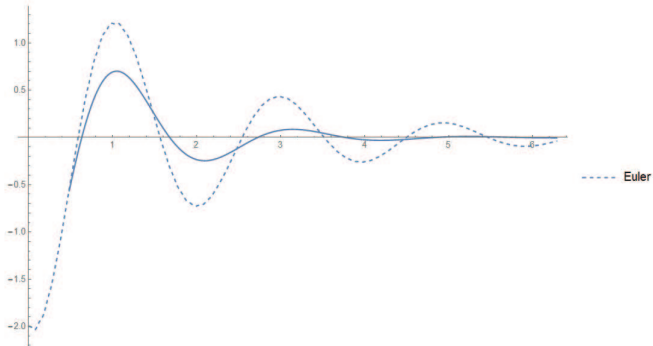
```
Prepend [Table[{t, x[t]}/s2[[1]], {t, 0, 2 * Pi, .5}], {"t_n", "x_n"}]//N
```

t_n	x_n
0.	-2.
0.5	-0.489149
1.	0.693786
1.5	0.239481
2.	-0.23468
2.5	-0.108238
3.	0.0770462
3.5	0.0464296
4.	-0.0243597
4.5	-0.0191708
5.	0.00731641
5.5	0.00768034
6.	-0.00203252

```

c. eu = NDSolve[{x''[t] + 2 * x'[t] + 10 * x[t] == 0, {x'[0] == 0, x[0] == -2}}, x[t], {t, 0, 2 * Pi},
Method -> "ExplicitEuler", "StartingStepSize" -> .1]
Show[{Plot[Evaluate[x[t]/.s2], {t, 0, 2 * Pi}],
Plot[Evaluate[x[t]/.eu], {t, 0, 2 * Pi}, PlotStyle -> Dashed, PlotLegends -> {Euler}], PlotRange -> All]

```



3.2.2. Movimiento Vibratorio Simple. Ley de Hooke.

Según esta ley, una masa m que está sujeta a un resorte ejerce una fuerza opuesta y proporcional al alargamiento, en condición de equilibrio $mg = ks$ donde k es la constante de proporcionalidad y s es el alargamiento del resorte. En un sistema no amortiguado se tiene:

Por ser $mg = ks$ o
 $mg - ks = 0$

$$m \frac{d^2 x}{dt^2} = -k(s + x) + mg = -ks - kx + mg = -kx$$

finalmente:

$$\frac{d^2 x}{dt^2} + \frac{k}{m} x = 0$$

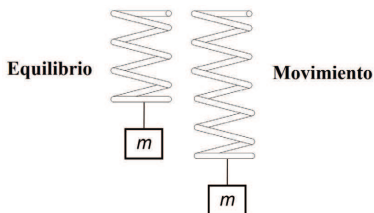


FIGURA 3.2.4. Movimiento vibratorio simple.

Suponga que se dispone de un sistema con una masa de $10kg$ lo cual alarga un resorte $2cm$, entonces:

$$10kg = 2kcm \text{ o } k = 5kg/cm$$

luego, si manualmente se estira hacia abajo bajo la acción del resorte hasta que dicha masa esté a $6cm$ de su posición de equilibrio se puede plantear la EDO:

$$\frac{d^2 x}{dt^2} + \frac{k}{m} x = 0 \text{ con } k = 5kg/cm, m = 10kg$$

con las condiciones iniciales:

$$x(0) = 6, x'(0) = 0$$

en nuestro caso:

`DSSolve[x''[t] + 1/2 * x[t]==0, x[t], t]`

la ecuación general del movimiento es:

$$\left\{ \left\{ x(t) \rightarrow c_2 \sin\left(\frac{t}{\sqrt{2}}\right) + c_1 \cos\left(\frac{t}{\sqrt{2}}\right) \right\} \right\}$$

`DSSolveValue[{x''[t] + 1/2 * x[t]==0, {x'[0] == 0, x[0] == 6}}, x[t], t]`

para las condiciones iniciales dadas la ecuación del movimiento será:

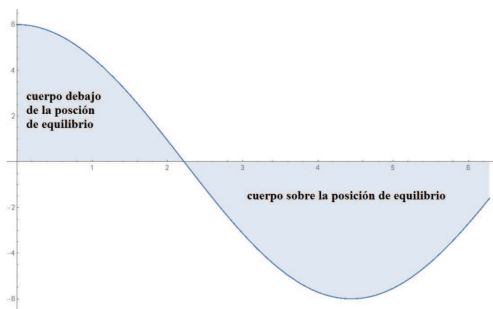
$$6 \cos\left(\frac{t}{\sqrt{2}}\right)$$

`s = NDSolve[{x''[t] + 1/2 * x[t]==0, {x'[0] == 0, x[0] == 6}}, x[t], {t, 2 * Pi}]`

`Prepend[Table[{t, x[t]}/.s[[1]], {t, 0, 2 * Pi, .5}], {"t_n", "x_n"}]`

t_n	x_n
0.	6.
0.5	5.62889
1.	4.56147
1.5	2.92978
2.	0.935662
2.5	-1.1742
3.	-3.1388
3.5	-4.71513
4.	-5.70818
4.5	-5.99511
5.	-5.54042
5.5	-4.40037
6.	-2.71597

Plot[Evaluate[x[t]/.s], {t, 0, 2 * Pi}]



PROBLEMA 3.27. Un cuerpo con masa de $6kg$ estira un resorte $2cm$, si dicho cuerpo se suelta en $t = 0$ desde un punto que está a $8cm$ bajo su posición de equilibrio adquiere una velocidad de $\frac{3}{2}cm/seg$. Determine el tiempo que tardaría en pasar por primera vez en la posición de equilibrio.

SOLUCIÓN. Por la ley de Hooke se tiene $6 = 2k$ o $k = 3$

así que:

$$\frac{d^2 x}{dt^2} + \frac{1}{2}x = 0 \text{ con:}$$

$$x(0) = 8, x'(0) = -\frac{3}{2}$$

los comandos Mathematica serán:

DSolve[$x''[t] + 1/2 * x[t] == 0, x[t], t$]

la ecuación general del movimiento es:

$$\left\{ \left\{ x(t) \rightarrow c_2 \sin\left(\frac{t}{\sqrt{2}}\right) + c_1 \cos\left(\frac{t}{\sqrt{2}}\right) \right\} \right\}$$

DSolveValue[$\{x''[t] + 1/2 * x[t] == 0, \{x'[0] == -3/2, x[0] == 8\}\}, x[t], t$]

para las condiciones iniciales dadas la ecuación del movimiento será:

$$\frac{1}{2} \left(16 \cos\left(\frac{t}{\sqrt{2}}\right) - 3\sqrt{2} \sin\left(\frac{t}{\sqrt{2}}\right) \right)$$

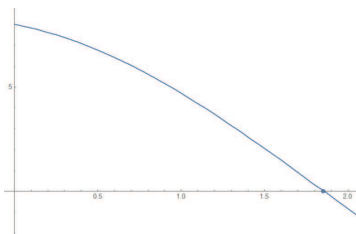
s = NDSolve[$\{x''[t] + 1/2 * x[t] == 0, \{x'[0] == -3/2, x[0] == 8\}\}, x[t], \{t, 2 * \text{Pi}\}$]

Solve[**Evaluate**[$x[t]/.s == 0, t$]

$$\{\{t \rightarrow 1.85488\}\}$$

el cuerpo pasará por primera vez en la posición de equilibrio a los 1.8548774260806893 segundos.

Show[**Plot**[**Evaluate**[$x[t]/.s$], $\{t, 0, 2 * \text{Pi}\}$], **ListPlot**[$\{\{1.8548774260806893, 0\}\}$], **PlotMarkers** \rightarrow **•**]



3.2.3. Movimiento Vibratorio Amortiguado. La fuerza que origina un proceso de amortiguación es proporcional a una potencia o múltiplo constante de la velocidad instantánea, por la segunda ley de Newton se tiene:

$$m \frac{d^2 x}{dt^2} + \lambda \frac{dx}{dt} = -kx$$

o bien:

$$\frac{d^2 x}{dt^2} + \frac{\lambda}{m} \frac{dx}{dt} + \frac{k}{m} x = 0$$

EJEMPLO 3.28. Un cuerpo con una masa de $6kg$ estira un resorte unos $2cm$, si la fuerza de amortiguación es el doble de la velocidad instantánea y dicho cuerpo se suelta en $t = 0$ desde su punto de equilibrio éste adquiere una velocidad hacia arriba de $\frac{3}{2} cm/seg$. Se hallará la ecuación del movimiento:

Según la ley de Hooke se tiene:

$$6 = 2k \text{ o } k = 3$$

y por la segunda ley de Newton:

$$\frac{d^2x}{dt^2} + \frac{1}{3} \frac{dx}{dt} + \frac{1}{2}x = 0 \text{ con } x(0) = 0, x'(0) = -\frac{3}{2}$$

DSolve[$x''[t] + 1/3 * x'[t] + 1/2 * x[t] == 0, x[t], t$]

la ecuación general del movimiento es:

$$\left\{ \left\{ x(t) \rightarrow c_1 e^{-t/6} \sin\left(\frac{\sqrt{17}t}{6}\right) + c_2 e^{-t/6} \cos\left(\frac{\sqrt{17}t}{6}\right) \right\} \right\} \quad \left(\sqrt{c_1^2 + c_2^2} \right) e^{-t/6} \text{ se denomina amplitud amortiguada.}$$

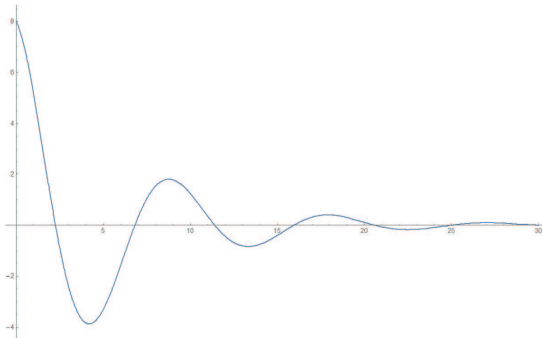
DSolveValue[{ $x''[t] + 1/3 * x'[t] + 1/2 * x[t] == 0, \{x'[0] == -3/2, x[0] == 0\}$ }, $x[t], t$]

para las condiciones iniciales dadas la ecuación del movimiento será:

$$-\frac{9e^{-t/6} \sin\left(\frac{\sqrt{17}t}{6}\right)}{\sqrt{17}}$$

s = NDSolve[{ $x''[t] + 1/3 * x'[t] + 1/2 * x[t] == 0, \{x'[0] == -3/2, x[0] == 8\}$ }, $x[t], \{t, 30\}$]

Plot[**Evaluate**[$x[t]/.s$], { $t, 0, 30$ }, **PlotRange** → **All**]



3.2.4. Ecuaciones Diferenciales con Coeficientes Variables.

Ecuación de Cauchy Euler o Ecuación Equidimensional.

La ecuación diferencial de la forma:

$$a_n x^n \frac{d^n y}{dx^n} + a_{n-1} x^{n-1} \frac{d^{n-1} y}{dx^{n-1}} + a_{n-2} x^{n-2} \frac{d^{n-2} y}{dx^{n-2}} + \dots + a_1 x \frac{dy}{dx} + a_0 y = g(x)$$

tiene como característica obvia que el grado de los coeficientes monomiales x^k coinciden uno a uno con el orden de diferenciación $\frac{d^k y}{dx^k}$, para ilustrar métodos de soluciones se escoge la ecuación no homogénea de segundo orden:

$$ax^2 \frac{d^2 y}{dx^2} + bx \frac{dy}{dx} + cy = g(x) \quad (1)$$

y para dar comienzo se puede escribir $y = x^m$, luego:

$$\frac{dy}{dx} = mx^{m-1} \text{ y } \frac{d^2 y}{dx^2} = m(m-1)x^{m-2} \quad (2)$$

sustituyendo en 1:

$$ax^2 \frac{d^2 y}{dx^2} + bx \frac{dy}{dx} + cy = x^m [am(m-1) + bm + c] = x^m [am^2 + (b-a)m + c] = g(x)$$

y usando el factor integrante $e^{-\int \frac{b}{ax} dx}$ se obtienen:

a. $m_1 \neq m_2 \neq \dots \neq m_n$, en este caso la solución general es:

$$y(x) = C_1 x^{m_1} + C_2 x^{m_2} + \dots + C_n x^{m_n}$$

b. $m = m_1 = m_2 = \dots = m_n$ la solución general será:

$$y(x) = C_1 x^m + C_2 x^m \ln(x) + C_3 x^m \ln^2(x) + \dots + C_n x^m \ln^{n-1}(x)$$

c. $m = a \pm bi$ la solución general es:

$$y(x) = x^a (C_1 \cos(b \ln(x)) + C_2 \operatorname{sen}(b \ln(x)))$$

EJEMPLO 3.29. Para resolver $x^2 \frac{d^2 y}{dx^2} - 2x \frac{dy}{dx} - 4y = 0$ se tiene:

$$x^2 \frac{d^2 y}{dx^2} - 2x \frac{dy}{dx} - 4y = x^m (m^2 - 3m - 4) = x^m (m+1)(m-4) = 0$$

finalmente:

$$y = \frac{C_1}{x} + C_2 x^4$$

EJEMPLO 3.30. $4x^2 \frac{d^2 y}{dx^2} + 8x \frac{dy}{dx} + y = 0$ es otro caso, observe que al sustituir $y = x^m$ se obtiene:

$$4x^2 \frac{d^2 y}{dx^2} + 8x \frac{dy}{dx} + y = x^m (2m+1)^2 = 0 \text{ donde } m_1 = m_2 = -\frac{1}{2}$$

finalmente:

$$y = \frac{C_1}{\sqrt{x}} + \frac{C_2}{\sqrt{x}} \ln(x)$$

DEFINICIÓN 3.31. Sea $\{y_1, y_2, \dots, y_n\}$ un conjunto de funciones con $n-1$ derivadas en cualquier intervalo I , de denomina Wronskiano al determinante:

$$W = \begin{vmatrix} y_1 & y_2 & \dots & y_n \\ y_1' & y_2' & \dots & y_n' \\ \vdots & \vdots & \dots & \vdots \\ y_1^{n-1} & y_2^{n-1} & \dots & y_n^{n-1} \end{vmatrix}$$

Reduciendo 1 a $y'' + P(x)y' + Q(x)y = f(x)$, las formulas:

$$u_1' = -\frac{y_2 f(x)}{W(y_1, y_2)}$$

$$u'_2 = \frac{y_1 f(x)}{W(y_1, y_2)}$$

proporcionan la solución particular de la ecuación diferencial dada, en efecto, suponga que y_1 y y_2 son el conjunto de soluciones de:

$$y_1'' + P(x)y_1' + Q(x)y_1 = 0 \text{ y } y_2'' + P(x)y_2' + Q(x)y_2 = 0$$

y que $y_p = u_1(x)y_1(x) + u_2(x)y_2(x)$ es una solución particular, luego:

$$y_p' = u_1 y_1' + u_1' y_1 + u_2 y_2' + u_2' y_2 \quad (a)$$

si:

$$u_1' y_1 + u_2' y_2 = 0 \quad (b)$$

entonces:

$$y_p'' = u_1 y_1'' + u_1' y_1' + u_2 y_2'' + u_2' y_2' \quad (c)$$

además:

$$\begin{aligned} u_1 y_1'' + u_1' y_1' + u_2 y_2'' + u_2' y_2' + P(u_1 y_1' + u_1' y_1 + u_2 y_2' + u_2' y_2) + Q(u_1 y_1 + u_2 y_2) = \\ u_1 (y_1'' + P y_1' + Q y_1) + u_2 (y_2'' + P y_2' + Q y_2) + u_1' y_1' + u_2' y_2' = \\ u_1' y_1' + u_2' y_2' = f \end{aligned}$$

de este resultado y b se plantea el sistema de ecuaciones:

$$\begin{cases} u_1' y_1 + u_2' y_2 = 0 \\ u_1' y_1' + u_2' y_2' = f \end{cases}$$

finalmente:

$$\begin{aligned} u_1' &= -\frac{y_2 f(x)}{W(y_1, y_2)} \\ u_2' &= \frac{y_1 f(x)}{W(y_1, y_2)} \end{aligned}$$

EJERCICIO 3.32. Resolver $x^2 y'' - 3xy' + 3y = 2x^2 e^x$

SOLUCIÓN. La sustitución $y = x^m$ conlleva a:

$$\begin{aligned} x^2 m(m-1)x^{m-2} - 3xm x^{m-1} + 3x^m = x^m(m^2 - m - 3m + 3) = x^m(m^2 - 4m + 3) = \\ x^m(m-3)(m-1) = 0 \end{aligned}$$

se construye:

$$y_c = C_1 x + C_2 x^3 \quad (1)$$

en esta oportunidad se usará Mathematica, declare $f(x)$:

f[x.]:=2 * x^2 * E^x

los componentes y_1 y y_2 son:

y1 = x

x

y2 = x^3

x^3

```

W=Wronskian[{y1,y2},x];u1'=-y2*f[x]/W;u2'=y1*f[x]/W;
Simplify[Integrate[u1',x]*y1+Integrate[u2',x]*y2]Print["y(x)=",
"C1"*y1+"C2"*y2+Simplify[Integrate[u1',x]*y1+Integrate[u2',x]*y2]];

```

para finalizar:

$$y(x) = C_2 x^3 + C_1 x + 2e^x(x-1)x$$

OBSERVACIÓN.

Resulta importante apuntar que el comando *NDSolve* tiene la tarea de tomar una secuencia de pasos en la variable independiente en un proceso adaptativo reduciendo dicha holgura hasta presentar una mejor solución, un aliado en la manipulación de los resultados es *AccuracyGoal*, este determina el valor absoluto permitido en la solución mientras que otro importante comando *PrecisionGoal* controlará el error relativo, de esta manera si desea una solución de mejor calidad se puede configurar con *AccuracyGoal* \rightarrow *Infinity*; otro importante comando es *WorkingPrecision* y es quien manipulará la precisión, si se configuran altos valores para *AccuracyGoal* y/o *PrecisionGoal* se deben acompañar con igual longitud los valores correspondientes de *WorkingPrecision*.

EJERCICIO 3.33. Resolver $\frac{d^2x}{dt^2} + \frac{1}{3}\frac{dx}{dt} + \frac{1}{2}x = 0$

SOLUCIÓN.

```

s = NDSolve[{x''[t] + 1/2 * x'[t] == 0, x'[0] == -3/2, x[0] == 8}, x[t], {t, 2}, AccuracyGoal -> 20,
PrecisionGoal -> 20, WorkingPrecision -> 25]

```

```

Prepend [Table[{t, x[t]}/.s[[1]], {t, 1.85487, 1.85488, 10^-6}], {"t_n", "x_n"}]

```

t_n	x_n
1.85487	0.0000433197
1.85487	0.0000374673
1.85487	0.000031615
1.85487	0.0000257626
1.85487	0.0000199103
1.85488	0.0000140579
1.85488	$8.205574679773231 \cdot 10^{-6}$
1.85488	$2.353224723590508 \cdot 10^{-6}$
1.85488	$-3.499125231280764 \cdot 10^{-6}$
1.85488	$-9.351475186158975 \cdot 10^{-6}$
1.85488	-0.0000152038

```

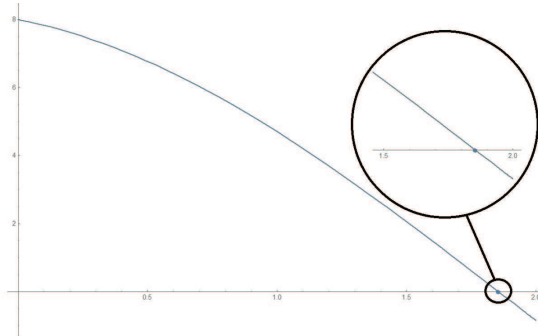
Solve[Evaluate[x[t]/.s] == 0, t]

```

$$\{t \rightarrow 1.854877402099112766596727\}$$

Show[Plot[Evaluate[x[t]/.s], {t, 0, 2}],

ListPlot[{{1.85487740209911276659672699243386360563, 0}}, PlotMarkers \rightarrow ●]



EJERCICIOS

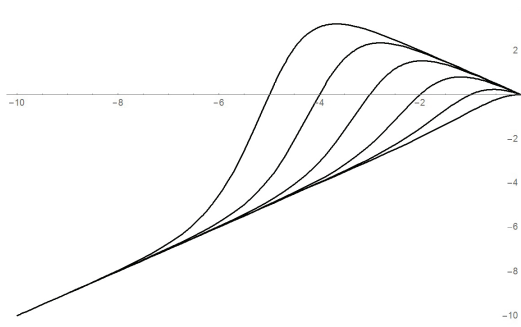
1. Demuestre que $\ln(x^2) + \ln\left(\frac{y^2}{x^2}\right) = A + x$ se puede escribir como $y^2 = Be^x$.
2. Demuestre que $\ln(1+y) + \ln(1+x) = A$ se puede escribir como $xy + x + y = C$.
3. Resuelva : $xy' = y^2 + y - x^2$ y grafique para $c = \{0, 1, \dots, 5\}$ y $y : [-10, 0] \rightarrow \mathbb{R}$.

SOLUCIÓN.

`sol1 = DSolve[x * y'[x] == y[x]^2 + y[x] - x^2, y[x], x]`

$$\left\{ \left\{ y(x) \rightarrow -\frac{x(e^{2c_1+2x}-1)}{e^{2c_1+2x}+1} \right\} \right\}$$

`Plot[Evaluate[Table[y[x]/.sol1[[1]]/.C[1] -> c, {c, 0, 5, 1}], {x, -10, 0}, PlotStyle -> Black]`



4. Use el método de Picard con 4 iteraciones para obtener un valor aproximado de y cuando $x = .2$ sabiendo que $y(1) = 0$ y $\frac{dy}{dx} = 3x + y^2$.

SOLUCIÓN.

$$Y_0(x) = 1$$

$$Y_1(x) = \frac{3x^2}{2} + x + 1$$

$$Y_2(x) = \frac{9x^5}{20} + \frac{3x^4}{4} + \frac{4x^3}{3} + \frac{5x^2}{2} + x + 1$$

$$Y_3(x) = \frac{81x^{11}}{4400} + \frac{27x^{10}}{400} + \frac{47x^9}{240} + \frac{17x^8}{32} + \frac{1157x^7}{1260} + \frac{68x^6}{45} + \frac{25x^5}{12} + \frac{23x^4}{12} + 2x^3 + \frac{5x^2}{2} + x + 1$$

$$Y_4(x) = \frac{6561x^{23}}{445280000} + \frac{2187x^{22}}{19360000} + \frac{6903x^{21}}{12320000} + \frac{16191x^{20}}{7040000} + \frac{797659x^{19}}{105336000} + \frac{2865683x^{18}}{133056000} + \frac{245509763x^{17}}{4523904000} + \frac{6384409x^{16}}{53222400} + \frac{31409197x^{15}}{130977000} + \frac{60887581x^{14}}{139708800} + \frac{92842733x^{13}}{129729600} + \frac{3629173x^{12}}{3326400}$$

$$3x^6 + \frac{169x^5}{60} + \frac{9x^4}{4} + 2x^3 + \frac{5x^2}{2} + x + 1$$

usando $x = .2$:

$$1 + x + \frac{(5 * x^2)}{2} + 2 * x^3 + \frac{(9 * x^4)}{4} +$$

$$\frac{(169 * x^5)}{60} + 3 * x^6 + \frac{(3739 * x^7)}{1260} + \frac{(28907 * x^8)}{10080} +$$

$$\begin{aligned}
 & (56603 * x^9)/22680 + (101183 * x^{10})/50400 + (2568707 * x^{11})/1663200 + \\
 & (3629173 * x^{12})/3326400 + (92842733 * x^{13})/129729600 + \\
 & (60887581 * x^{14})/139708800 + (31409197 * x^{15})/130977000 + \\
 & (6384409 * x^{16})/53222400 + (245509763 * x^{17})/4523904000 + \\
 & (2865683 * x^{18})/133056000 + (797659 * x^{19})/105336000 + \\
 & (16191 * x^{20})/7040000 + (6903 * x^{21})/12320000 + \\
 & (2187 * x^{22})/19360000 + (6561 * x^{23})/445280000/.x->0.1
 \end{aligned}$$

1.12726

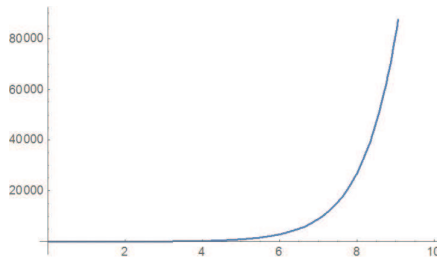
5. Resolver: $y' - y = e^x$ con $y(0) = 1$

SOLUCIÓN.

```
sol2 = Simplify[DSolve[{y'[x] - y[x] == E^x, y[0] == 1}, y[x], x]]
```

$$\{\{y(x) \rightarrow e^x(x + 1)\}\}$$

```
Plot[y[x]/.sol2[[1]], {x, 0, 10}]
```



6. Resolver $ty^2y' = y^3 - t^3$ con $y(t = 1) = 2$ y presente soluciones numéricas en una matriz de datos.

SOLUCIÓN. La EDO se puede expresar como $ty \frac{dy}{dt} = y^3 - t^3$

```
sol = DSolve[{t * y[t]^2 * y'[t] == y[t]^3 - t^3, y[1] == 2}, y[t], t]
```

$$\{\{y(t) \rightarrow t \sqrt[3]{8 - 3 \log(t)}\}\}$$

```
Prepend[Table[{t, y[t]}/.sol[[1]], {t, 1, 10, 1}], {x, "y(x)"}]//N
```

x	y(x)
1.	2.
2.	3.61813
3.	5.02669
4.	6.2644
5.	7.34627
6.	8.27648
7.	9.05179
8.	9.66196
9.	10.0881
10.	10.2985

7. Resuelva $\frac{dy}{dt} + y = y^2 (\cos(t) - \operatorname{sen}(t))$

SOLUCIÓN. $y(x) = -\frac{1}{\operatorname{sen}(t) - c_1 e^t}$

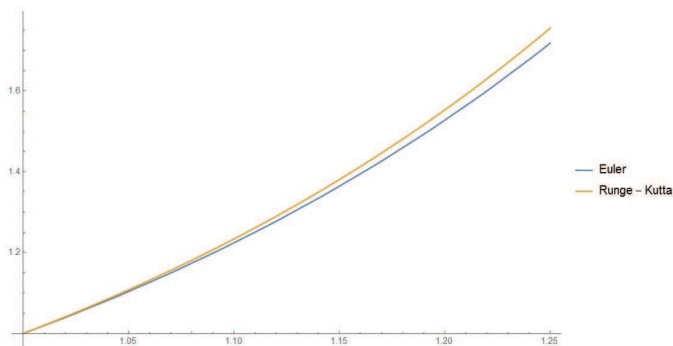
8. Use los métodos de Euler y Runge-Kutta para hallar el valor aproximado de y cuando $y(1) = 1$ y $\frac{dy}{dx} = 2xy$ y $h = 0.2$, compare los resultados y plantee conclusiones.

SOLUCIÓN.

EULER = NDSolve[{ $y'(x) = 2xy(x)$, $y(1) = 1$ }, $y(x)$, { x , 1, 1.25}],
Method → "ExplicitEuler", "StartingStepSize" → 0.2]

RK = NDSolve[{ $y'(x) = 2xy(x)$, $y(1) = 1$ }, $y(x)$, { x , 1, 1.25}],
Method → "ExplicitRungeKutta", "StartingStepSize" → 0.2]

Plot[{ $y[x]/.$ EULER, $y[x]/.$ RK}, { x , 1, 1.25}, PlotLegends → {Euler, Runge - Kutta}]



Prepend[Table[{ x , $y[x]/.$ EULER, $y[x]/.$ RK, Abs[($y[x]/.$ RK) - ($y[x]/.$ EULER)]}, { x , 1, 1.25, .02}],

{x, "Euler", "R-K", "Error"}//N

x	Euler	R-K	Error
1.	{1.}	{1.}	{0.}
1.02	{1.03951}	{1.04123}	{0.00171515}
1.04	{1.08169}	{1.08502}	{0.00332995}
1.06	{1.12656}	{1.13156}	{0.00500198}
1.08	{1.1741}	{1.18105}	{0.00695015}
1.1	{1.22421}	{1.23368}	{0.00947066}
1.12	{1.27739}	{1.28969}	{0.0122957}
1.14	{1.33429}	{1.34932}	{0.0150246}
1.16	{1.39496}	{1.41284}	{0.0178792}
1.18	{1.45936}	{1.48053}	{0.0211706}
1.2	{1.52738}	{1.55271}	{0.0253241}
1.22	{1.59973}	{1.62971}	{0.0299719}
1.24	{1.67734}	{1.71189}	{0.0345518}

9. Use la rutina de Runge-Kutta expuesta para hallar una aproximación de

$$y' = -\frac{1}{1+y^2} \text{ con } y(0) = 1 \text{ y } h = 0.25$$

10. Resuelva:

$$x'' + 32x + 16x = 0 \text{ con } x(0) = 0 \text{ y } \left. \frac{dx}{dt} \right|_{t=0} = -1.5$$

11. Una masa de 500 gr se sujeta a un resorte cuya constante es de 8 N/m, si el sistema se sumerge en un líquido que aplica una fuerza de amortiguamiento de 5 veces la velocidad instantánea, determine las ecuaciones del movimiento si:

a. la masa se suelta a partir del reposo a partir de 50 cm por debajo de la posición de equilibrio.

b. la masa se suelta a partir del reposo a partir de 50 cm por debajo de la posición de equilibrio con una velocidad dirigida hacia arriba de 6 m/s.

12. Determine una solución particular de:

$$\frac{d^2x}{dt^2} + 2\lambda \frac{dx}{dt} + \omega^2 x = k$$

SOLUCIÓN. Se sugiere usar:

$$s = \text{DSolve}[x''[t] + 2 * \lambda * x'[t] + \omega^2 * x[t] == A, x[t], t]$$

$$\left\{ \left\{ x(t) \rightarrow \frac{A}{2\lambda + \omega^2} + c_1 e^{t\sqrt{-2\lambda - \omega^2}} + c_2 e^{-t\sqrt{-2\lambda - \omega^2}} \right\} \right\}$$

s1 = Solve[x[t] == 0]/.s/.t → 0

$$\left(\left\{ \frac{A}{\omega^2 + 2\lambda} + c_1 + c_2 \rightarrow 0 \right\} \right)$$

Solve[D[x[t]/.s, t] == 0, C[1]]/.s/.t → 0/.s1

$$\left(\left(\{c_1 \rightarrow c_2\} \right) \right)$$

Bibliografía

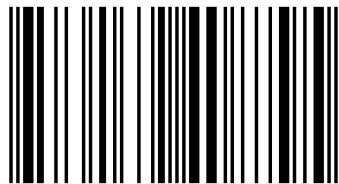
- [1] Bainov D. y Hristova S., *Differential Equations with Maxima*, Chapman & Hall/CRC, 2011.
- [2] Björck Å., *Numerical methods in matrix computations*. Vol. 59. Cham: Springer, 2015.
- [3] Burden R. y Faires J., *Análisis Numérico*, Thomson, 2011.
- [4] Cordero O., *Elementos para el Cálculo Matricial: Soluciones analíticas y numéricas asistidas con Mathematica*, Editorial Académica Española, 2018.
- [5] Davis P. and Rabinowitz P., *Methods of Numerical Integration*, 2nd Edition, Academic Press, New York, 1984.
- [6] Demidovich B. and Maron I., *Computational Mathematics*, Editorial Mir, Moscow, 1976.
- [7] Fundación Polar, *Matemática Maravillosa*, <http://www.fpolar.org.ve/matematica3>, 2006.
- [8] Golub G. and Ortega J., *Scientific Computing: An Introduction with Parallel Computing*, Academic Press, Boston, 1993.
- [9] Meriam J., *Dinámica*. Reverté, Barcelona, 2007.
- [10] Nakamura S., *Métodos Numéricos Aplicados con Software*, Prentice-Hall, 1992.
- [11] Stewart I., *Historia de las Matemáticas*. Crítica, 2007.
- [12] Stroud A., *Approximate Calculation of Multiple Integrals*, Prentice Hall, Englewood Cliffs, NJ, 1972.
- [13] Timberlake T. and Mixon J., *Classical Mechanics with Maxima*, Springer, 2016.
- [14] Wolfram, S., *The Mathematica Book*, Wolfram Media, Cambridge University Press, 1998.

Triada de la Aproximación en el Cálculo Diferencial e Integral

La intención en esta obra es reconocer lo que el aporte académico y bibliográfico especializado nos ha consolidado, y manipular esas contribuciones en un ambiente de laboratorio de matemáticas donde profesor y estudiante experimenten de manera interactiva, cooperativa y agradable, ideas, conceptos y fundamentos matemáticos. En general se pretende exponer el comportamiento gráfico y numérico como elementos fundamentales para la comprensión y estudio de la interpolación, la integral y las ecuaciones diferenciales ordinarias.



Docente Universitario perteneciente al Área de Matemáticas de la Universidad Nacional Abierta (Venezuela). Maestría en Educación Abierta y a Distancia - Venezuela.



978-613-9-40879-5

editorial académica **española**