

Introducción a los métodos numéricos + Octave.

Salvador Díaz Maldonado
2022.

Contenido

1	Introducción.	1
2	Series de Taylor.	2
2.1	Ejemplos.	5
2.3.	Ejercicios propuestos.	19
3	Raíces de ecuaciones.	20
3.1	Método de bisección.	21
3.2	Método de Newton-Raphson.	22
3.3.	Ejemplos.	23
3.4	Ejercicios propuestos.	28
4	Sistemas de ecuaciones.	29
4.1	Repasso de algunos conceptos de matrices.	29
4.2	Sistema de ecuaciones lineales.	32
4.2.1	Eliminación de Gauss.	35
4.2.2	Método de Gauss-Jordan.	36
4.2.3	Método de la matriz inversa.	37
4.2.4	Ejemplos.	39
4.2.5	Ejercicios propuestos.	45
4.3	Sistema de ecuaciones no lineales.	46
4.3.1	Método de Newton-Raphson.	46
4.3.2	Ejemplos.	49
4.3.3	Ejercicios propuestos.	52
5	Interpolación polinomial	54
5.1	Interpolación de Lagrange.	55
5.2	Ejemplos.	57
5.3	Ejercicios propuestos.	61
6	Aproximación funcional	62
6.1	Método de mínimos cuadrados.	62
6.1.1	Regresión polinomial.	63
6.1.2	Regresión lineal múltiple.	63
6.2	Ejemplos.	65
6.3	Ejercicios propuestos.	72

7 Diferenciación numérica	74
7.1 Introducción	74
7.2 Diferencias hacia adelante y hacia atrás.	74
7.3 Diferencias centradas.	79
7.4. Diferenciación numérica con datos desigualmente espaciados.	80
7.5. Ejemplos.	82
7.6. Ejercicios propuestos.	89
8 Integración numérica	90
8.1 Introducción.	90
8.2 Regla trapezoidal.	90
8.3 Regla de Simpson.	93
8.3.1 Simpson 3/8.	95
8.4 Ejemplos.	98
8.5. Ejercicios propuestos.	105
9 Ecuaciones diferenciales ordinarias de primer orden.	107
9.1 Introducción.	107
9.2. Método de Euler.	107
9.3. Método de Runge-Kutta de 4to orden.	108
9.4. Sistemas de ecuaciones diferenciales ordinarias de primer orden.	109
9.5. Ejemplos.	109
9.6. Ejercicios propuestos.	128
Apéndice A. Introducción a la dinámica de sistemas.	130
Apéndice B. Ejemplos prácticos usando Octave.	193
Raíces de ecuaciones.	194
Sistemas de ecuaciones lineales.	224
Sistemas de ecuaciones no lineales.	246
Interpolación polinomial y Aproximación funcional.	253
Diferenciación numérica.	278
Integración numérica.	291
Ecuaciones diferenciales.	324
Apéndice C. Derivación de la regla de Simpson con corrección final.	349

1 Introducción.

El propósito del presente libro es presentar, de una manera clara y simple (no simplista), una introducción en algunas técnicas elementales de los métodos numéricos. Se pretende apoyar al profesor de la materia para contar con una referencia básica que le permita homogeneizar el curso con algunos otros posibles profesores de la misma; enriqueciendo así la estrategia de aprendizaje de los estudiantes.

En este trabajo de introducción a los métodos numéricos se procura también interesar al alumno a través de conceptos básicos, evitando algunos otros tales como convergencia y estabilidad, que pudieran ser estudiados posteriormente. Para esto, se recurre al apoyo del paquete computacional de software libre Octave (además de la aplicación Xcos de Scilab para una introducción a dinámica de sistemas), resolviendo ejemplos de las diferentes técnicas numéricas a través de sus comandos, así como de la elaboración de códigos.

Debido a que frecuentemente cuando se decide aplicar algún método numérico se debe a que el problema no tiene solución analítica o porque, debido a su complejidad, es mucho más fácil resolverlo con el apoyo de una computadora, uno de los retos importantes a enfrentar es la verificación que la solución encontrada sea la apropiada. Para esto se proponen varias formas de calcular el error, de tal manera que sea determinado con la tolerancia fijada por el ingeniero, según la naturaleza del problema a resolver.

Vale la pena mencionar que realmente los métodos numéricos son técnicas que preceden a la computadora, sin embargo, por ser frecuentemente procedimientos lógicos y aritméticos muy laboriosos, su aplicación era muy limitada, no fue sino hasta mediados del siglo pasado que tuvieron su ascenso definitivo con la combinación del uso de la computadora, sustituyendo en muchos casos a los procedimientos analíticos de las matemáticas.

Con el apoyo de la computadora, el ingeniero puede seleccionar y aplicar un método numérico que satisfaga las demandas del trabajo requerido, o solicitar a un analista especializado el producto preciso. Es decir, el conocimiento básico de los métodos numéricos es vital para una ejecución apropiada del trabajo deseado.

Salvador Díaz Maldonado.

2 Series de Taylor.

Las series de Taylor son fundamentalmente en los métodos numéricos. Muchas de las técnicas numéricas se derivan directamente de ellas, así como también es usada para la estimación de los errores involucrados en el empleo de algunas de dichas técnicas.

Si el valor de una función $f(x)$ puede expresarse en una región x cercana a $x = x_0$ por la serie de potencia infinita,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \frac{(x - x_0)^3}{3!}f'''(x_0) + \dots + \frac{(x - x_0)^n}{n!}f^n(x_0) \quad (2.1)$$

entonces $f(x)$ es analítica en la región cercana a $x = x_0$ y la serie (2.1) es única y denominada Serie de Expansión de Taylor de $f(x)$ en la vecindad de $x = x_0$. Si la serie de Taylor existe, entonces conociendo la $f(x)$ y sus derivadas para $x = x_0$, se puede encontrar el valor de $f(x)$ para algunas x diferentes de x_0 . Notar que n es la máxima derivada considerada en la serie de Taylor, es decir, el número de términos menos uno.

El comportamiento de la serie de Taylor puede apreciarse gráficamente por medio de la figura 2.1 considerando que se desea obtener el valor de la función en $x_1, f(x_1)$, expandiendo la serie en la vecindad de x_0 .

$$f(x_1) = f(x_0) + (x_1 - x_0) f'(x_0) + \frac{(x_1 - x_0)^2}{2!} f''(x_0) + \frac{(x_1 - x_0)^3}{3!} f'''(x_0) + \dots + \frac{(x_1 - x_0)^n}{n!} f^n(x_0)$$

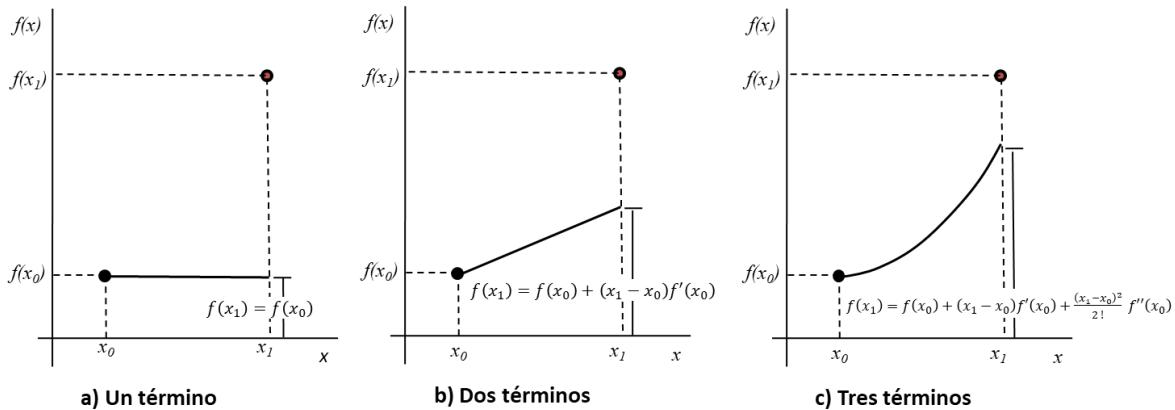


Figura 2.1. Interpretación gráfica de la serie de Taylor (adaptada¹).

¹ Hornbeck, R.W. Numerical Methods. Quantum Publishers, INC. USA. 1975.

Como se aprecia, entre más términos se consideran en la serie de Taylor, más exacto es el resultado aproximado a $f(x_1)$. Si solamente se considera el primer término, entonces la función resulta constante, $f(x_1)=f(x_0)$, como se muestra en la figura 2.1a. Si se usan los dos primeros términos, entonces se tiene una pendiente de $f'(x_0)$, figura 2.1b. Usando tres términos se incluye una curvatura al considerar la segunda derivada de la $f(x_0)$, figura 2.1c.

Error de truncamiento, R_n . En relación con el error de truncamiento de la serie de Taylor, se puede decir lo siguiente: si se expande la serie de Taylor en la vecindad de $x=x_0$, se obtiene:

$$f(x) = f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \frac{(x - x_0)^3}{3!} f'''(x_0) + \dots + \frac{(x - x_0)^n}{n!} f^n(x_0) + R_n \quad (2.2)$$

donde R_n es precisamente el error por truncamiento.

Para procesar el error de truncamiento es necesario aplicar el teorema del Valor Medio, el cual establece que si una función $f(x)$ y su primera derivada son continuas en un intervalo $[x_i, x_{i+1}]$, entonces existe al menos un punto sobre la función que tiene una pendiente dada por $f'(\xi)$, la cual es paralela a la línea que une $f'(x_i)$ con $f'(x_{i+1})$. El parámetro ξ marca el valor de x en donde se presenta la pendiente mencionada. Esto se puede apreciar en la figura 2.2.

Entonces, si se trunca la serie de la ecuación (2.2) en el primer término,

$$f(x) \approx f(x_0)$$

y

$$R_0 = h f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f'''(x_0) + \dots$$

donde $h = x - x_0$.

Tomando solo el primer término de esta serie se obtiene

$$R_0 = h f'(x_0) \approx h f'(\xi)$$

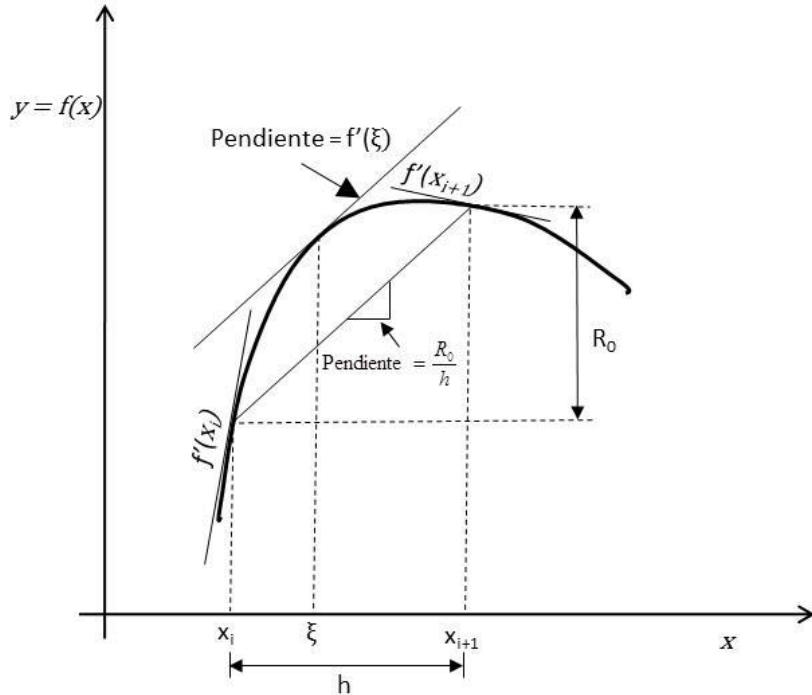


Figura 2.2. Representación gráfica del teorema del Valor Medio.

Y por el teorema del valor medio,

$$f'(\xi) = \frac{R_0}{h}$$

También,

$$R_0 = h f'(\xi)$$

Por inducción, si se toma hasta el segundo término de la serie de Taylor, el residuo con respecto a la segunda derivada queda:

$$R_1 = \frac{h^2}{2!} f''(\xi)$$

De tal manera que con lo anterior se puede apreciar que:

$$R_n = \frac{h^{n+1}}{(n+1)!} f^{n+1}(\xi)$$

Y finalmente, el error de la serie de Taylor cuando se trunca después del término contenido en $(x-a)^n$ no es mayor que

$$R_n = \frac{|h|^{n+1}}{(n+1)!} \left| \frac{d^{n+1}f}{dx^{n+1}} \right|_{\max} \quad (2.3)$$

donde el subíndice “máx” denota la magnitud máxima de la derivada en el intervalo desde x hasta x_0 . O también se puede decir que el error es del orden de $(x-x_0)^{n+1}$ o $O(x-x_0)^{n+1}$ o $O(h)^{n+1}$.

Entonces, el error en la serie de Taylor para $f(x)$, cuando la serie es truncada después del término contenido en $(x-a)^n$, no es mayor que R_n obtenido con la ecuación (2.3).

Nota:

Debido a que ξ varía entre x y x_0 , es recomendable aplicar la ecuación 2.3 para estos valores, obteniendo así un intervalo en donde se encuentra el error real.

Otra forma de manejar el error es asignando una tolerancia inicial, tomando en cuenta algún término de la serie de Taylor; es decir, se puede determinar la condición de que el error tolerado esté en función del valor del término de la serie que cumpla con dicha tolerancia. Sin embargo, se debe tener cuidado con este criterio ya que existen series que tienen mezclados términos nulos, no cumpliéndose lo anterior si no se toman las precauciones debidas.

2.1 Ejemplos.

Ejemplo 2.1.1 Expandir la serie de Taylor del siguiente polinomio, en $x_0=0$ con $h=1$. Es decir, terminar la función en $x_I=1^2$:

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

Inicialmente se grafica la función en el intervalo h para visualizar su comportamiento, figura 2.3. Además, de la función a resolver se obtiene que para $x=1$,

$$f(x) = -0.1(1)^4 - 0.15(1)^3 - 0.5(1)^2 - 0.25(1) + 1.2 = 0.20$$

² Chapra, S.C y Canale, R.P. Numerical Methods for Engineers. Seventh Edition. McGraw-Hill. 2015.

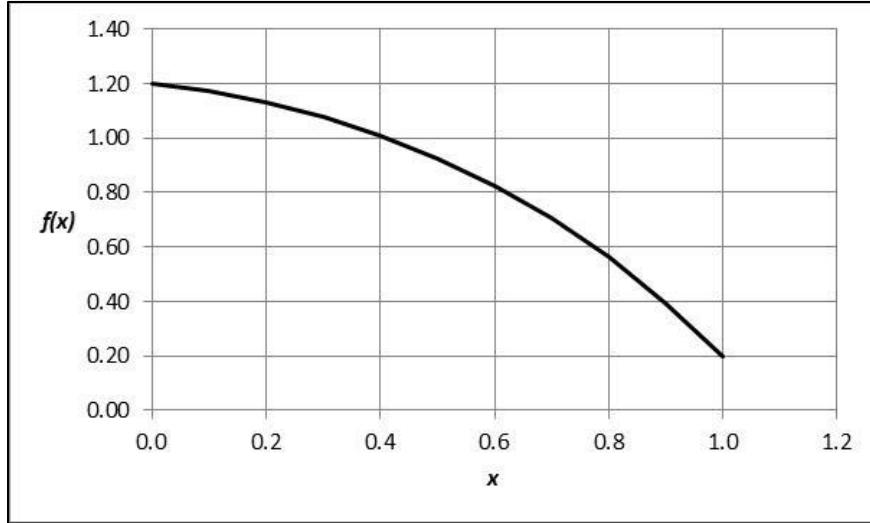


Figura 2.3. Gráfica del ejemplo 2.1.

De la serie de Taylor,

$$f(x) = f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \frac{(x - x_0)^3}{3!} f'''(x_0) + \dots + \frac{(x - x_0)^n}{n!} f^n(x_0)$$

$$f(x) = f(0) + (x - 0) f'(0) + \frac{(x - 0)^2}{2!} f''(0) + \frac{(x - 0)^3}{3!} f'''(0) + \dots + \frac{(x - 0)^n}{n!} f^n(0)$$

Considerando que:

$$f(x_0) = -0.1 x_0^4 - 0.15 x_0^3 - 0.5 x_0^2 - 0.25 x_0 + 1.2 , \quad x_0 = 0, \quad f(x_0) = 1.2$$

$$f'(x_0) = -0.4 x_0^3 - 0.45 x_0^2 - x_0 - 0.25 , \quad x_0 = 0, \quad f'(x_0) = -0.25$$

$$f''(x_0) = -1.2 x_0^2 - 0.90 x_0 - 1 , \quad x_0 = 0, \quad f''(x_0) = -1.00$$

$$f'''(x_0) = -2.4 x_0 - 0.90 , \quad x_0 = 0, \quad f'''(x_0) = -0.90$$

$$f^{iv}(x_0) = -2.4$$

Por lo tanto, la función del polinomio se expande hasta máximo la cuarta derivada y se puede expresar como:

$$f(x) = f(0) + (x-0)f'(0) + \frac{(x-0)^2}{2!}f''(0) + \frac{(x-0)^3}{3!}f'''(0) + \frac{(x-0)^4}{4!}f^{iv}(0) + R_4$$

$$f(x) = 1.2 - 0.25x - \frac{x^2}{2!} - 0.9\frac{x^3}{3!} - 2.4\frac{x^4}{4!} + R_4 \quad (2.4)$$

De la ecuación (2.4) se obtiene, sustituyendo el valor de $x=1$, un resultado de 0.20. El cual es igual al valor de la función original, por lo que el error real o verdadero es de 0.00.

Procesando la ecuación (2.4), de acuerdo al número de sus términos considerados, se obtienen los siguientes resultados mostrados en la tabla 2.1, variando el valor de x .

x	Número de términos de la serie de Taylor				
	1	2	3	4	5
0.0	1.2000	1.2000	1.2000	1.2000	1.2000
0.1	1.2000	1.1750	1.1250	1.1249	1.1248
0.2	1.2000	1.1500	1.0500	1.0488	1.0486
0.3	1.2000	1.1250	0.9750	0.9710	0.9701
0.4	1.2000	1.1000	0.9000	0.8904	0.8878
0.5	1.2000	1.0750	0.8250	0.8063	0.8000
0.6	1.2000	1.0500	0.7500	0.7176	0.7046
0.7	1.2000	1.0250	0.6750	0.6236	0.5995
0.8	1.2000	1.0000	0.6000	0.5232	0.4822
0.9	1.2000	0.9750	0.5250	0.4157	0.3500
1.0	1.2000	0.9500	0.4500	0.3000	0.2000

Tabla 2.1 Resultados del ejemplo 1 con $x_0=0$ y $h=1$.

En la figura 2.4 se puede apreciar que entre más términos de la serie de Taylor sean considerados, mayor exactitud se logra, aproximándose así al valor verdadero de la raíz del polinomio propuesto.

Por otro lado, con el fin de verificar el comportamiento del resultado expandiendo la serie de Taylor en otro valor de x_0 , más cercano a $x=1$, se supondrá ahora $x_0=0.2$, sustituyendo al valor anterior de 0. Entonces, se repite el procedimiento con el nuevo valor de $x_0 = 0.2$, es decir, se disminuye la h de 1.0 a 0.8, con lo cual se pretende tener mayor exactitud.

$$f(x) = f(0.2) + (x-0.2)f'(0.2) + \frac{(x-0.2)^2}{2!}f''(0.2) + \frac{(x-0.2)^3}{3!}f'''(0.2) + \dots + \frac{(x-0.2)^n}{n!}f^n(0.2)$$

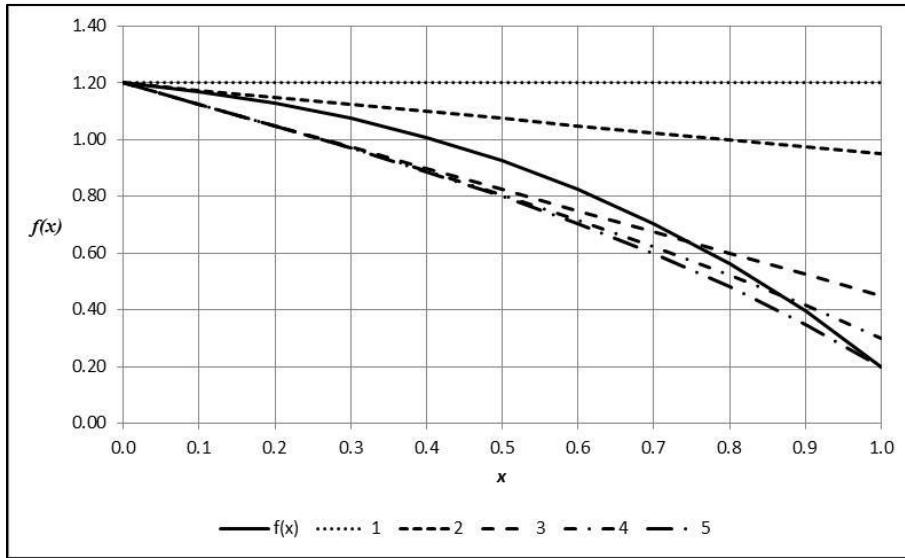


Figura 2.4. Resultados del ejemplo 1 con $x_0 = 0.0$ y $h = 1.0$.

Considerando que:

$$f(x_0) = -0.1x_0^4 - 0.15x_0^3 - 0.5x_0^2 - 0.25x_0 + 1.2, \quad x_0 = 0.2, \quad f(x_0) = 1.128640$$

$$f'(x_0) = -0.4x_0^3 - 0.45x_0^2 - x_0 - 0.25, \quad x_0 = 0.2, \quad f'(x_0) = -0.471200$$

$$f''(x_0) = -1.2x_0^2 - 0.90x_0 - 1, \quad x_0 = 0.2, \quad f''(x_0) = -1.228000$$

$$f'''(x_0) = -2.4x_0 - 0.90, \quad x_0 = 0.2, \quad f'''(x_0) = -1.380000$$

$$f^{iv}(x_0) = -2.4$$

Por lo tanto, la función del polinomio se expresa como:

$$\begin{aligned} f(x) &= 1.128640 - (x-0.2)(0.471200) - \frac{(x-0.2)^2}{2!}(1.228000) - \frac{(x-0.2)^3}{3!}(1.380000) \\ &\quad - \frac{(x-0.2)^4}{4!}(2.4) \end{aligned} \tag{2.5}$$

En la tabla 2.2 y figura 2.5 se muestran los resultados al evaluar la ecuación (2.5) para varios valores de x . De la figura 2.5 se observa que con cinco términos de la serie, prácticamente se sobreponen la curva de aproximación sobre la curva de la función. Este incremento en la exactitud se debe a que la serie de Taylor se expandió en un punto más cercano a 1.

x	Número de términos de la serie de Taylor				
	1	2	3	4	5
0.2	1.128640	1.128640	1.128640	1.128640	1.128640
0.3	1.128640	1.081520	1.075380	1.075150	1.075140
0.4	1.128640	1.034400	1.009840	1.008000	1.007840
0.5	1.128640	0.987280	0.932020	0.925810	0.925000
0.6	1.128640	0.940160	0.841920	0.827200	0.824640
0.7	1.128640	0.893040	0.739540	0.710790	0.704540
0.8	1.128640	0.845920	0.624880	0.575200	0.562240
0.9	1.128640	0.798800	0.497940	0.419050	0.395040
1.0	1.128640	0.751680	0.358720	0.240960	0.200000

Tabla 2.2 Resultados del ejemplo 1 con $x_0 = 0.2$ y $h=0.8$.

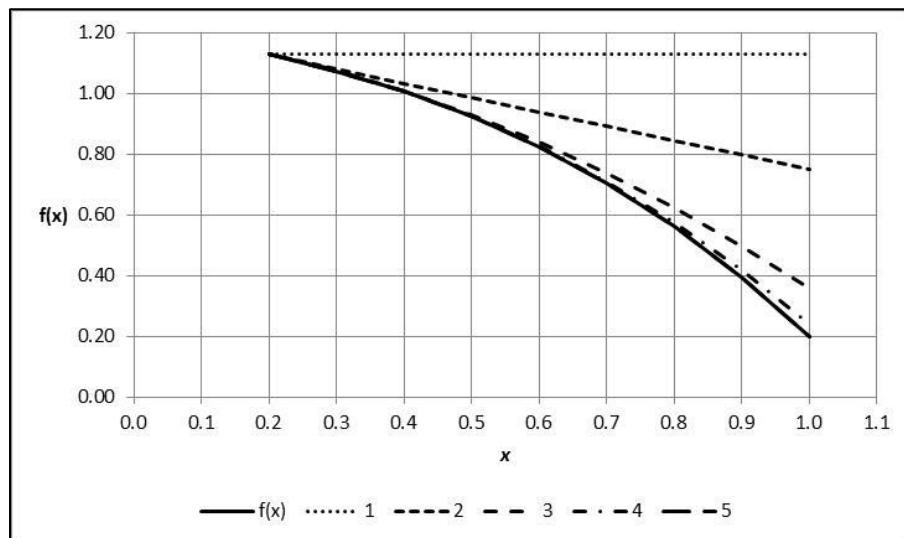


Figura 2.5. Resultados del ejemplo 1 con $x_0 = 0.2$ y $h = 0.8$.

Finalmente, en ambos casos de resolución del ejemplo, el error residual o de truncamiento es igual a cero, ya que no existe la quinta derivada de la función. Entonces,

$$R_4 = \frac{|h|^5}{5!} \left| \frac{d^5 f}{dx^5} \right| = \frac{|1|^5}{5!} |0| = 0$$

Al presentarse estos casos, se puede observar que el resultado del error de truncamiento es igual si se toman 4 o 5 términos de la serie de Taylor, por lo tanto, para que el error de truncamiento tome un valor mayor que cero, se puede calcular con $n = n-1$ (para este caso $n=3$). Por lo tanto:

$$R_3 = \frac{|h|^4}{4!} \left| \frac{d^4 f}{dx^4} \right| = \frac{|1|^4}{4!} |-2.4| = 0.10$$

Es decir, resulta que el error verdadero debería ser menor que 0.10, lo cual se cumple.

➤ Comandos Octave.

```
>> pkg load symbolic
>> syms x
>> fx=-0.1*x^4-0.15*x^3-0.5*x^2-0.25*x+1.2;
>> t5=taylor(fx,x,'order',5)
t5 = (sym)

        4      3      2
x      3*x      x      x      6
- - - - - - - - - - - + -
10      20      2      4      5

>> t5=function_handle(t5)
t5 =

@(x) -x .^ 4 / 10 - 3 * x .^ 3 / 20 - x .^ 2 / 2 - x / 4 + 6 / 5

>> valor_t5=t5(1);
>> fprintf('%10.6f \n',valor_t5)
0.200000
>>
```

➤ En la figura 2.6 se muestran los resultados (con seis al aplicar series de Taylor con el código Octave, *Taylor_Octave.m*.

Figura 2.6. Resultados del código en Octave para el Ejemplo 2.1.1.

```
** Este programa determina la serie de Taylor, simbólica y
   numéricamente, de una función f(x)suministrada por pantalla **

Introduce una función con respecto a x    -0.1*x^4-0.15*x^3-0.5*x^2-0.25*x+1.2
Expandir alrededor de x0 = ? 0

*** Serie de Maclaurin ***

Determinar en x          = ? 1
Número de términos de la serie de Taylor = ? 5

Términos, Serie y su valor

1 , 6/5 , 1.200000
2 , 6/5 - x/4 , 0.950000
3 , -x**2/2 - x/4 + 6/5 , 0.450000
```

```

4 , -3*x**3/20 - x**2/2 - x/4 + 6/5 , 0.300000
5 , -x**4/10 - 3*x**3/20 - x**2/2 - x/4 + 6/5 , 0.200000

serie = (sym)


$$-\frac{x^4 - 3x^3 - x^2 + x}{10 \cdot 20 \cdot 2 \cdot 4 \cdot 5}$$


* Serie de Taylor solicitada:
* Valor: 0.200000

* Valor verdadero de la función: 0.200000

* Error verdadero: 0.00000000

* Error de truncamiento x : 0.10000000
* Error de truncamiento x0: 0.10000000

Debido a que la derivada de mayor orden de la serie de Taylor
es igual a 0, el error de truncamiento se calculó en n=n-1
>>

```

Ejemplo 2.1.2 Encontrar la serie de expansión de Taylor para la función: $f(x) = \sin(x)$, en la vecindad de $x_0 = 0$ y calcular la función en $x = \pi/4$ radianes. Calcular con un error del orden de $O(h)^7$.

De la ecuación (2.1) se obtiene:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \frac{(x - x_0)^3}{3!}f'''(x_0) + \frac{(x - x_0)^4}{4!}f^{iv}(x_0) + \frac{(x - x_0)^5}{5!}f^v(x_0) + \frac{(x - x_0)^6}{6!}f^{vi}(x_0) + \frac{(x - x_0)^7}{7!}f^{vii}(x_0) + R_7$$

$$\begin{aligned} \sin(x) &= \sin(0) + (x - 0)\cos(0) - \frac{(x - 0)^2}{2!}\sin(0) - \frac{(x - 0)^3}{3!}\cos(0) + \frac{(x - 0)^4}{4!}\sin(0) \\ &\quad + \frac{(x - 0)^5}{5!}\cos(0) - \frac{(x - 0)^6}{6!}\sin(0) - \frac{(x - 0)^7}{7!}\cos(0) + R_7 \end{aligned}$$

$$\sin(x) = 0 + x - 0 - \frac{x^3}{3!} + 0 + \frac{x^5}{5!} - 0 - \frac{x^7}{7!} + R_7$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + R_7$$

Se puede observar que para este caso, el error de aproximación $O(h)^7$ es igual a $O(h)^6$, ya que el penúltimo término es igual a cero.

Entonces, $\sin(\pi/4)$ se puede aproximar de la siguiente forma:

$$\sin(\pi/4) = (\pi/4) - \frac{(\pi/4)^3}{3!} + \frac{(\pi/4)^5}{5!} - \frac{(\pi/4)^7}{7!} + R_7$$

$$\sin(\pi/4) = (0.785398) - \frac{(0.785398)^3}{3!} + \frac{(0.785398)^5}{5!} - \frac{(0.785398)^7}{7!} + R_7$$

$$\sin(\pi/4) = 0.707106$$

Y el error verdadero es:

$$\text{Valor verdadero} - \text{aproximación} = 0.707107 - 0.707106 = 3.116000 \times 10^{-7}$$

Es decir, considerando un error del orden de $O(h)^7$ el error verdadero es 3.116000×10^{-7} .

Con respecto al error de truncamiento dado por la ecuación (2.3),

$$R_n = \frac{|h|^{n+1}}{(n+1)!} |f^{(n+1)}(x)|_{\max}$$

$$R_7 = \frac{|h|^8}{8!} |f^{(viii)}(x)|$$

Se tiene que $h = \pi/4 - 0 = \pi/4$. Entonces,

$$R_7 = \frac{|\pi/4|^8}{8!} |\sin(\pi/4)| = 0.000003 = 2.539122 \times 10^{-6}$$

Así se concluye que, si no se conociera el error verdadero, se puede afirmar que sería menor que el error de truncamiento de 2.539122×10^{-6} , considerando un error del orden de $O(h)^7$ en la serie de Taylor. Lo cual es comprobable en este caso.

➤ Usando comandos de Octave se obtiene lo siguiente:

```
>> pkg load symbolic
>> syms x
>> fx=sin(x);
```

```
>> t8=taylor(fx,x,'order',8)
t8 = (sym)

    7      5      3
    x      x      x
- ---- + --- - -- + x
 5040    120    6

>> t8=function_handle(t8)
t8 =

@(x) -x .^ 7 / 5040 + x .^ 5 / 120 - x .^ 3 / 6 + x

>> valor_t8=t8(pi/4);
>> fprintf('%10.6f \n',valor_t8)
 0.707106
>>
```

- En la figura 2.7 se muestra los resultados obtenidos con código de Octave, *Taylor_Octave.m*.

Figura 2.7. Resultados del código en Octave para el Ejemplo 2.1.2.

```
** Este programa determina la serie de Taylor, simbólica y
   numéricamente, de una función f(x)suministrada por pantalla **

Introduce una función con respecto a x    sin(x)
Expandir alrededor de x0 = ? 0

*** Serie de Maclaurin ***

Determinar en x          = ? pi/4
Número de términos de la serie de Taylor = ? 8

Términos, Serie y su valor

1 , 0 , 0.000000
2 , x , 0.785398
3 , x , 0.785398
4 , -x**3/6 + x , 0.704653
5 , -x**3/6 + x , 0.704653
6 , x**5/120 - x**3/6 + x , 0.707143
7 , x**5/120 - x**3/6 + x , 0.707143
8 , -x**7/5040 + x**5/120 - x**3/6 + x , 0.707106
```

```

serie = (sym)

      7      5      3
      x      x      x
- ----- + ---- - -- + x
 5040    120     6

* Serie de Taylor solicitada:
* Valor:   0.707106

* Valor verdadero de la función:   0.707107

* Error verdadero:   0.00000031

* Error de truncamiento x :   0.00000254
* Error de truncamiento x0:   0.00000000
>>

```

El error real será menos que 0.00000254, lo cual se cumple.

Ejemplo 2.1.3 Usando series de expansión de Taylor para e^x alrededor de $x_0=0$, determinar e^x en $x=0.5$; considerando un error de $O(h)^4$. Encontrar el error de truncamiento y comparar con el resultado verdadero.

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \frac{(x - x_0)^3}{3!}f'''(x_0) + \frac{(x - x_0)^4}{4!}f^{IV}(x_0) + R_4$$

$$e^x = e^0 + (x - 0)e^0 + \frac{(x - 0)^2}{2!}e^0 + \frac{(x - 0)^3}{3!}e^0 + \frac{(x - 0)^4}{4!}e^0 + R_4$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + R_4$$

Entonces, calculando en $x = 0.5$,

$$e^{0.5} = 1 + 0.5 + \frac{0.5^2}{2!} + \frac{0.5^3}{3!} + \frac{0.5^4}{4!} + R_4$$

$$e^{0.5} = 1.648438$$

Y el error verdadero es: Valor verdadero – Aproximación = $e^{0.5} - 1.648438 = 0.000284$.

En cuanto al error de truncamiento, de la ecuación (2.3), y considerando que en la aproximación con cuatro términos de la serie se llegó hasta la cuarta derivada, $n=4$.

$$R_4 = \frac{|h|^{n+1}}{(n+1)!} |f^{n+1}(x)|_{\max} = \frac{|0.5|^5}{5!} |f^{IV}(x)| = \frac{|0.5|^5}{4!} |e^{0.5}| = 0.000429$$

Con esto se afirma que el error de truncamiento (0.000429) obtenido con un error de orden $O(h)^4$, es mayor que el error verdadero (0.000284).

En caso de que se requiera un menor error de truncamiento, se puede incrementar el orden del error de la función determinada.

- Usando comandos de Octave se obtienen los siguientes resultados:

```
>> pkg load symbolic
>> syms x
>> fx=exp(x);
>> warning('off','all');
>> t5=taylor(fx,x,'order',5)
t5 = (sym)

      4      3      2
x      x      x
-- + -- + -- + x + 1
24      6      2

>> t5=function_handle(t5)
t5 =

@(x) x .^ 4 / 24 + x .^ 3 / 6 + x .^ 2 / 2 + x + 1

>> valor_t5=t5(0.5);
>> printf('%10.6f\n', valor_t5)
    1.648438
>>
```

- En la figura 2.8 se muestran los resultados obtenidos con código en Octave.

Figura 2.8. Resultados del código en Octave, *Taylor_Octave.m*, para el Ejemplo 2.1.3.

```
** Este programa determina la serie de Taylor, simbólica y
   numéricamente, de una función f(x) suministrada por pantalla **
```

```
Introduce una función con respecto a x      exp(x)
Expandir alrededor de x0 = ? 0
```

```
*** Serie de Maclaurin ***
```

```
Determinar en x      = ? 0.5
Número de términos de la serie de Taylor = ? 5
```

Términos, Serie y su valor

```

1 , 1 ,    1.000000
2 , x + 1 ,   1.500000
3 , x**2/2 + x + 1 ,   1.625000
4 , x**3/6 + x**2/2 + x + 1 ,   1.645833
5 , x**4/24 + x**3/6 + x**2/2 + x + 1 ,   1.648438

serie = (sym)


$$\frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$


* Serie de Taylor solicitada:
* Valor: 1.648438

* Valor verdadero de la función: 1.648721

* Error verdadero: 0.00028377

* Error de truncamiento x : 0.00042935
* Error de truncamiento x0: 0.00026042
>>

```

El error real será menor que 0.00042935, lo cual se cumple.

Ejemplo 2.1.4. Usando series de expansión de Taylor alrededor de $x_0=0$, determinar $e^{\sin(x)}$ con

$O(h)^4$. Calcular $e^{\sin(\pi/4)}$.

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \frac{(x - x_0)^3}{3!}f'''(x_0) + \frac{(x - x_0)^4}{4!}f^{iv}(x_0) + R_4$$

Expandiendo alrededor de $x_0=0$ y calculando previamente las derivadas de la función,

$$f(x) = e^{\sin(x)}$$

$$f'(x) = e^{\sin(x)} \cos(x)$$

$$f''(x) = e^{\sin(x)} [\cos(x)^2 - \sin(x)]$$

$$f'''(x) = e^{\sin(x)} [\cos(x)^3 - \cos(x) - 3\cos(x)\sin(x)]$$

$$f^{iv}(x) = e^{\sin(x)} [\sin(x) - 4\cos(x)^2 + \cos(x)^4 + 3\sin(x)^2 - 6\cos(x)^2\sin(x)]$$

$$f^v(x) = e^{\sin(x)} \cos(x) [5\sin(x) + 23\sin(x)^2 + 10\sin(x)^3 + \sin(x)^4 - 8]$$

Entonces, expandiendo en x_0 :

$$e^{\sin(x)} = 1 + x + \frac{x^2}{2!} + 0 - \frac{3x^4}{4!} + R_4 = 1 + x + \frac{x^2}{2!} - \frac{3x^4}{4!} + R_4$$

Valorando en $x = \pi/4$,

$$\begin{aligned} e^{\sin(\pi/4)} &= 1 + (\pi/4) + \frac{(\pi/4)^2}{2!} - \frac{3(\pi/4)^4}{4!} + R_4 \\ e^{\sin(\pi/4)} &= 2.046260 \end{aligned}$$

El error verdadero = $e^{\sin(\pi/4)} - 2.046260 = -0.018145$

En cuando al error de truncamiento,

$$R_4 = \frac{|h|^{4+1}}{(4+1)!} |f^{4+1}(x)|_{\max} = \frac{(\pi/4)^5}{5!} |f^v(\pi/4)| = 0.038647$$

Luego se puede decir que el error verdadero es menor que el error de truncamiento.

➤ Usando comandos de Octave se obtienen los siguientes resultados:

```
>> pkg load symbolic
>> syms x
>> fx=exp(sin(x));
>> t5=taylor(fx,x,'order',5)
t5 = (sym)
```

$$\begin{aligned} &4 &2 \\ &x &x \\ &- \frac{--}{--} + \frac{--}{--} + x + 1 \end{aligned}$$

8 2

```
>> t5=function_handle(t5)
t5 =
@(x) -x.^4 / 8 + x.^2 / 2 + x + 1

>> valor_t5=t5(pi/4);
>> printf('%10.6f\n', valor_t5)
2.046260
>>
```

- En la figura 2.9 se muestra el resultado arrojado por el código de *Octave*, *Taylor_Octave.m*.

Figura 2.9. Resultado del Ejemplo 2.1.4 con código de Octave.

```
** Este programa determina la serie de Taylor, simbólica y
numéricamente, de una función f(x) suministrada por pantalla **
```

```
Introduce una función con respecto a x    exp(sin(x))
Expandir alrededor de x0 = ? 0
```

```
*** Serie de Maclaurin ***
```

```
Determinar en x          = ? pi/4
Número de términos de la serie de Taylor = ? 5
```

Términos, Serie y su valor

1 , 1 , 1.000000

2 , x + 1 , 1.785398

3 , x**2/2 + x + 1 , 2.093823

4 , x**2/2 + x + 1 , 2.093823

5 , -x**4/8 + x**2/2 + x + 1 , 2.046260

serie = (sym)

```
4      2  
x      x  
- -- + -- + x + 1  
8      2
```

```
* Serie de Taylor solicitada:  
* Valor:    2.046260  
  
* Valor verdadero de la función:   2.028115  
  
* Error verdadero:    0.01814529  
  
* Error de truncamiento x :    0.03864701  
* Error de truncamiento x0:    0.01992316  
>>
```

2.3. Ejercicios propuestos.

- a) Encontrar la serie de expansión de Taylor para $\operatorname{senh}(x)$ alrededor de $x = 0$.
- b) Encontrar $\operatorname{sinh}(0.9)$ con error del orden de $O(h)^5$ usando la serie de expansión de Taylor del ejercicio 2.3.1. Determinar el error verdadero y el error de truncamiento.
- c) Encontrar una expresión para $\operatorname{sen}(x) \cos(x)$ con un error del orden de $O(h)^2$, usando series de Taylor alrededor de $x = 0$. Determinar la función encontrada en $x_I = \pi/4$, así como los errores verdadero y de truncamiento.

3 Raíces de ecuaciones.

Generalmente, el evaluar raíces consiste en encontrar los valores de x que satisfacen ciertas relaciones tales como:

$$ax^3 + bx^2 - cx = d \quad (3.1)$$

o

$$e^{\operatorname{sen}(x)} - \frac{1}{\cos(x)} = 0 \quad (3.2)$$

Dependiendo del problema, estos valores de x pueden ser reales o complejos y pueden ser finitos o infinitos en número.

El procedimiento para encontrar raíces, siempre consistirá en reunir los términos en un solo miembro. Entonces, las ecuaciones (3.1) y (3.2) quedan:

$$ax^3 + bx^2 - cx - d = 0 \quad (3.3)$$

$$e^{\operatorname{sen}(x)} - \frac{1}{\cos(x)} = 0 \quad (3.4)$$

Para otros valores que no sean raíces, estas igualdades no serán satisfechas. En general,

$$f(x) = ax^3 + bx^2 - cx - d = 0 \quad (3.5)$$

$$g(x) = e^{\operatorname{sen}(x)} - \frac{1}{\cos(x)} = 0 \quad (3.6)$$

Determinar las raíces de estas ecuaciones es equivalente a encontrar los valores de x para los cuales las funciones $f(x)$, $g(x)$, son iguales a cero.

Una recomendación muy importante al tratar de encontrar las raíces de una ecuación, es graficarla previamente para tener un panorama de su comportamiento, de tal manera que se pueda determinar inicialmente, de manera aproximada, la situación de la(s) raíz(es).

3.1 Método de bisección.

Para aplicar este método se considera que una función, conocida, tiene una raíz en el intervalo $a < x < b$, es decir, corta el eje x en dicho intervalo. En la figura 3.1 se aprecia que para comprobar que la raíz se encuentra entre a y b , el producto de sus funciones, $f(a)*f(b)$, debe ser negativo.

En este método aprovecha esta situación de cambio de signo del producto de esas funciones, para aproximarse a la raíz. Se subdivide este intervalo las veces que sea necesario para llegar a la tolerancia especificada. Entonces, x_m será el valor de la raíz buscada.

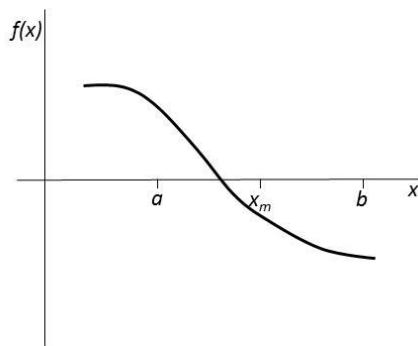


Figura 3.1. Método de bisección.

En la figura 3.2 se muestra un algoritmo propuesto para aplicar este método.

Figura 3.2. Algoritmo del método de bisección.

```

Lee a, b, tol, niter
iter=0
f(xm)=2*tol
mientras (iter<niter) y (|f(xm)|>tol)
    inicia
        iter=iter+1
        xm=(a+b)/2
        si f(b)*f(xm)>0 entonces
            b=xm
        otro
            a=xm
        termina si
    termina
imprime iter, xm
fin
```

3.2 Método de Newton-Raphson.

Para describir este método se recurre a las series de Taylor. Considerar un punto x_0 , el cual no es la raíz de la función, pero está razonablemente cerca de ella.

$$f(x) = f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \frac{(x - x_0)^3}{3!} f'''(x_0) + \dots \quad (3.7)$$

Si $f(x)$ es igual a cero, entonces x es una raíz de la función. Desafortunadamente la ecuación (3.7) es un polinomio de grado infinito. Sin embargo, se puede aproximar el valor de la raíz haciendo la función igual a cero y tomando solamente los dos primeros términos de la ecuación (3.7).

Entonces,

$$f(x) = f(x_0) + (x - x_0) f'(x_0) = 0 \quad (3.8)$$

Es decir, se tiene un error de truncamiento R_2 . Despejando x de la ecuación 3.8,

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.9)$$

Entonces, la expresión general para este método queda de la siguiente forma:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (3.10)$$

Cuando se avanza en el proceso iterativo, la diferencia entre x_{i+1} y x_i se hace cada vez menor si el procedimiento converge a la solución, de tal manera que el procedimiento de puede parar cuando esa diferencia es menor que una tolerancia fijada previamente, tol . A esa diferencia se le denomina δ , de tal manera que:

$$x_{i+1} - x_i = \delta = - \frac{f(x_i)}{f'(x_i)} \quad (3.11)$$

Entonces, cuando $|\delta| < tol$, se detiene el procedimiento porque se obtiene la raíz con la tolerancia prefijada.

Este método iterativo converge para muchas funciones, además de hacerlo rápidamente.

En la figura 3.3 se muestra el algoritmo propuesto,

Figura 3.3. Algoritmo para el método de Newton-Raphson.

```
Lee  $x_0, tol, niter$ 
iter=0
 $x=x_0$ 
delta=2*tol
mientras ( $|delta|>tol$ ) y ( $iter<=niter$ )
    comienza
        delta=-f(x)/f'(x)
        x=x+delta
        iter=iter+1
    termina
imprime iter, x
fin
```

3.3. Ejemplos.

Ejemplo 3.3.1 Determinar la raíz de la ecuación $y = x^2 - 0.9x - 8.5$ usando el método de Bisección, con una tolerancia de 0.0001.

Primero se recomienda graficar la función para visualizar en qué intervalo de x se encuentra la raíz, figura 3.4.

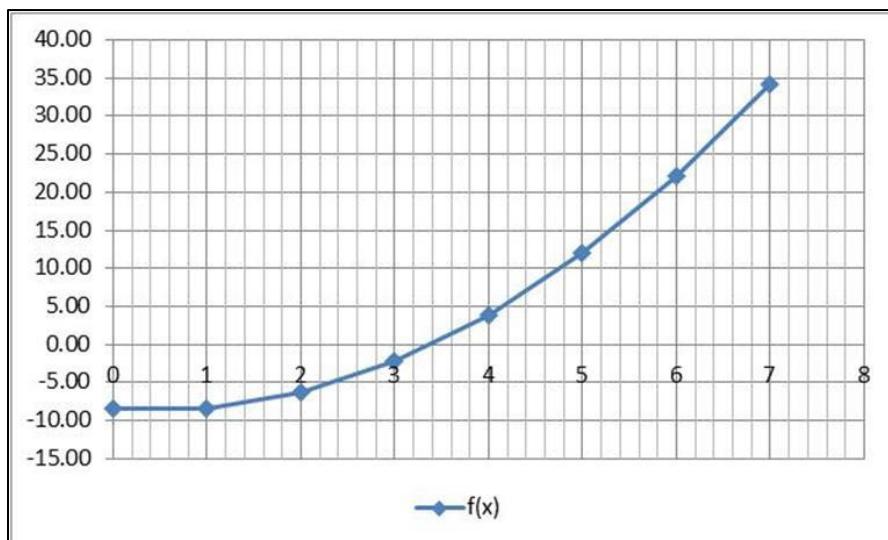


Figura 3.4. Gráfica de la función del ejemplo 3.3.1. Bisección.

Entonces, se observa que una raíz de la ecuación está dentro del intervalo $3 < \text{raíz} < 4$.

Con lo anterior, se puede seleccionar $a = 3$ y $b = 4$, para iniciar el proceso iterativo. Entonces, considerando el algoritmo de la figura 3.1 se supone lo siguiente:

$$a = 3, b = 4, tol = 0.0001 \text{ y } niter = 30$$

Si se desea comprobar si la raíz se encuentra entre los valores iniciales a y b de x , se puede obtener la función en ambos puntos y comprobar que contienen a la raíz si al multiplicarlas el resultado es menor que 0 (negativo): $f(a) = -2.2$, $f(b) = 3.9$; entonces $f(a)*f(b) < 0$, con lo que se garantiza que la raíz se encuentra entre a y b .

Calculando se obtiene lo siguiente según el algoritmo del método:

$$iter = 0, a = 3, b = 4,$$

$$\text{el valor medio entre } a \text{ y } b: x_m = (3+4)/2 = 3.5$$

$$f(b) = 3.9, f(x_m) = 0.6,$$

$f(b)*f(x_m) = 2.34 > 0$ entonces b y x_m están a la derecha de la raíz, por lo que para que la raíz se conserve entre los valores de a y b , b toma el valor de x_m ($b=x_m$). Así, para $iter=1$ ahora $a=3$ y $b=3.5$, valores entre los que se encuentra la raíz.

Para $iter=1$, $a=3$, $b=3.5$.

$x_m = (3+3.5)/2 = 3.25$. $f(b) = 0.6, f(x_m) = -0.8625$, por lo tanto, $f(b)*f(x_m)=-0.5175 < 0$. Esto significa que x_m está a la izquierda de la raíz. Entonces, ahora $a=x_m=3.25$.

Es decir, como se muestra en el algoritmo, cuando $f(b)*f(x_m) < 0$, $a=x_m$, en otro caso, $a=x_m$.

El procedimiento se repite hasta que se llega al número de iteraciones máximo prefijado ($iter > niter$) o cuando se cumple con la tolerancia ($|f(x_m)| \leq tol$).

En la tabla 3.1 (elaborada en Excel) se muestra el resultado del procedimiento. $x=3.399994$ con un error menor que la tolerancia fijada de 0.0001 (0.000036).

Tabla 3.1. Resultado final del ejemplo 3.3.1. Bisección.

<i>i</i>	<i>a</i>	<i>b</i>	<i>xm</i>	<i>f(b)</i>	<i>f(xm)</i>	<i>f(b)*f(xm)</i>	<i> f(xm) </i>
0	3.000000	4.000000	3.500000	3.900000	0.600000	2.340000	0.600000
1	3.000000	3.500000	3.250000	0.600000	-0.862500	-0.517500	0.862500
2	3.250000	3.500000	3.375000	0.600000	-0.146875	-0.088125	0.146875
3	3.375000	3.500000	3.437500	0.600000	0.222656	0.133594	0.222656
4	3.375000	3.437500	3.406250	0.222656	0.036914	0.008219	0.036914
5	3.375000	3.406250	3.390625	0.036914	-0.055225	-0.002039	0.055225
6	3.390625	3.406250	3.398438	0.036914	-0.009216	-0.000340	0.009216
7	3.398438	3.406250	3.402344	0.036914	0.013834	0.000511	0.013834
8	3.398438	3.402344	3.400391	0.013834	0.002305	0.000032	0.002305
9	3.398438	3.400391	3.399414	0.002305	-0.003457	-0.000008	0.003457
10	3.399414	3.400391	3.399902	0.002305	-0.000576	-0.000001	0.000576
11	3.399902	3.400391	3.400146	0.002305	0.000864	0.000002	0.000864
12	3.399902	3.400146	3.400024	0.000864	0.000144	0.000000	0.000144
13	3.399902	3.400024	3.399963	0.000144	-0.000216	0.000000	0.000216
14	3.399963	3.400024	3.399994	0.000144	-0.000036	0.000000	0.000036

- Usando comandos de Octave se obtienen los siguientes resultados:

```
>> p = [1 -0.9 -8.5]
p =
1.00000  -0.90000  -8.50000
```

```
>> p = [1 -0.9 -8.5];
>> roots(p)
ans =
3.4000
-2.5000
```

>>

- En la figura 3.5 se muestran los resultados obtenidos aplicando el código en octave.

Figura 3.5. Resultados obtenidos para el ejemplo 3.3.1 aplicando código Octave,

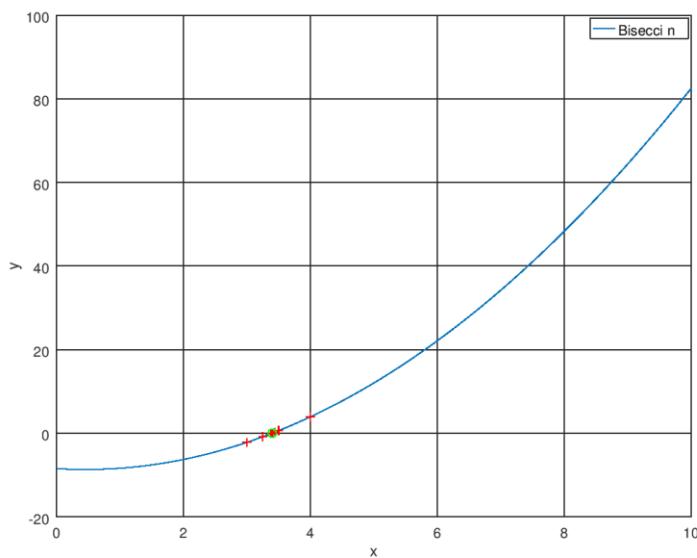
Biseccion_Octave.m.

Método de Bisección.

Introduce una función con respecto a x x^2-0.9*x-8.5

Valores para graficar la f(x)

```
x inicial? 0
x final? 10
```



Método de Bisección.

Datos:

a = 3

b = 4

Iteraciones máximas = 30

Tolerancia = 0.0001

i	a	b	xm	f (b) * f (xm)	f (xm)
0	3.000000	4.000000	3.500000	2.340000	0.600000
1	3.000000	3.500000	3.250000	-0.517500	-0.862500
2	3.250000	3.500000	3.375000	-0.088125	-0.146875
3	3.375000	3.500000	3.437500	0.133594	0.222656
4	3.375000	3.437500	3.406250	0.008219	0.036914
5	3.375000	3.406250	3.390625	-0.002039	-0.055225
6	3.390625	3.406250	3.398438	-0.000340	-0.009216
7	3.398438	3.406250	3.402344	0.000511	0.013834
8	3.398438	3.402344	3.400391	0.000032	0.002305
9	3.398438	3.400391	3.399414	-0.000008	-0.003457
10	3.399414	3.400391	3.399902	-0.000001	-0.000576
11	3.399902	3.400391	3.400146	0.000002	0.000864
12	3.399902	3.400146	3.400024	0.000000	0.000144
13	3.399902	3.400024	3.399963	-0.000000	-0.000216
14	3.399963	3.400024	3.399994	-0.000000	-0.000036

Raíz: 3.399994

>>

Ejemplo 3.3.2 Determinar la raíz de la ecuación $y = x^2 - 0.9x - 8.5$ con una tolerancia de 0.0001. Usar el método de Newton-Raphson.

Utilizando la ecuación (3.10), se genera el siguiente procedimiento (iniciando con una $x_0=3$):

$$f'(x) = 2x - 0.9$$

$$i=0, x_i = x_0 = 3,$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 3 - \frac{3^2 - 0.9 \cdot 3 - 8.5}{2(3) - 0.9} = 3 - \frac{-2.000000}{5.100000} = 3.431333$$

Como $\delta(0.431373) > tol$ (0.0001), el procedimiento continua con la siguiente iteración $i = 1$. Y así se repite hasta que $|\delta| < tol$. En la tabla 3.2 se muestra el resultado, en el cual se llega al valor de la raíz en la cuarta iteración, con la tolerancia requerida.

Tabla 3.2. Resultados del ejemplo 3.3.2. Newton-Raphson.

<i>i</i>	<i>x</i>	<i>f(x)</i>	<i>f'(x)</i>	δ
0	3.000000	-2.200000	5.100000	0.431373
1	3.431373	0.186082	5.962745	-0.031207
2	3.400165	0.000974	5.900330	-0.000165
3	3.400000	0.000000	5.900000	-4.62E-09
4	3.400000			

En la figura 3.6 se muestran los resultados del ejemplo 3.3.2 aplicando código de Octave.

Figura 3.6. Resultados obtenidos para el ejemplo 3.3.2 aplicando código Octave,

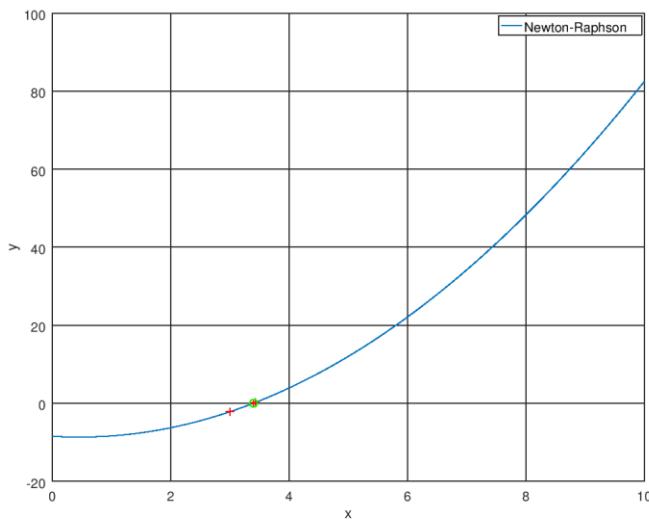
N_R_Octave.m.

Método de Newton-Raphson.

```
Introduce una función con respecto a x      x^2-0.9*x-8.5
```

```
Valores para graficar la f(x)
```

```
desde x = 0
hasta x = 10
```



```
x inicial? 3
tolerancia? 0.0001
Máximas iteraciones? 20
```

i	x	f(x)	f'(x)	delta
0	3.000000	-2.200000	5.100000	0.43137255
1	3.431373	0.186082	5.962745	-0.03120748
2	3.400165	0.000974	5.900330	-0.00016506
3	3.400000	0.000000	5.900000	-0.00000000
4	3.400000			

>>

3.4 Ejercicios propuestos.

Determinar la raíz de cada uno de los siguientes ejercicios, usando Bisección y Newton-Raphson.

a) $f(x) = e^{-x/4} (2-x) - 1$

b) $f(x) = x^2 - 0.5$

c) $f(x) = x - 0.8 - 0.2 \operatorname{sen}(x)$

d) $f(x) = x^4 - 8.6x^3 - 35.51x^2 + 464.4x - 998.46$

4 Sistemas de ecuaciones.

4.1 Repaso de algunos conceptos de matrices.

Una matriz es un arreglo de números. Por ejemplo:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 & 0 \\ 4 & 5 & 2 & 3 \\ 8 & 7 & 1 & 2 \end{bmatrix}$$

\mathbf{A} es una matriz de 3×4 , es decir, tres filas y cuatro columnas. Cualquier elemento de la matriz \mathbf{A} se puede denotar por a_{ij} , donde i es la fila y j la columna. Entonces, $a_{24} = 3$.

Existen matrices con dimensión de una columna o de un renglón. Un ejemplo de las primeras es:

$$\mathbf{B} = \begin{bmatrix} 2 \\ 3 \\ 5 \\ 8 \end{bmatrix}$$

y se denominan vectores columna, mientras que las matrices con un renglón,

$$\mathbf{C} = [0 \ 4 \ 8 \ 2 \ 10]$$

son llamadas vectores renglón.

Como en este capítulo se resolverán sistemas de ecuaciones donde se tienen el mismo número de ecuaciones que el de incógnitas, es de interés conocer las matrices cuadradas, es decir, con igual número de renglones y columnas. Por ejemplo:

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}$$

La matriz \mathbf{D} es una matriz cuadrada de 4×4 , es decir, de cuatro filas y cuatro columnas. La diagonal principal de la matriz está formada por los términos d_{11}, d_{22}, d_{33} y d_{44} .

Una matriz se denomina simétrica cuando $d_{ij} = d_{ji}$.

Una matriz diagonal superior es aquella cuyos elementos bajo la diagonal principal son igual a cero,

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{bmatrix}$$

Por otro lado, una matriz cuyos elementos sobre la diagonal principal son nulos, se denomina matriz diagonal inferior.

Cuando la diagonal principal es igual a 1 y el resto de los elementos son igual a cero, se tiene una matriz identidad o unitaria,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A continuación se muestran algunas de las propiedades fundamentales de las operaciones básicas entre matrices.

Suma. La suma está representada como:

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

y se define como:

$$c_{ij} = a_{ij} + b_{ij}$$

A y **B** deben ser del mismo orden, así como también el resultado **C**.

Resta. La resta es similar a la suma,

$$\mathbf{C} = \mathbf{A} - \mathbf{B}$$

y se define como:

$$c_{ij} = a_{ij} - b_{ij}$$

Las dos operaciones anteriores son conmutativas,

$$\mathbf{A} + \mathbf{B} = \mathbf{B} - \mathbf{A}$$

$$\mathbf{A} - \mathbf{B} = -\mathbf{B} + \mathbf{A}$$

Multiplicación. La multiplicación se representa como:

$$\mathbf{C} = \mathbf{A} \mathbf{B}$$

y se define como.

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

donde n es la dimensión de la columna \mathbf{A} y del renglón de \mathbf{B} . Entonces,

$$C = \begin{bmatrix} 3 & 2 \\ 4 & 0 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 5 & 4 & 3 \\ 8 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 31 & 18 & 11 \\ 21 & 16 & 12 \\ 50 & 18 & 11 \end{bmatrix}$$

Por ejemplo:

$$c_{23} = a_{21} b_{13} + a_{22} b_{23} = 4(3) + 0(1) 12$$

La multiplicación no es comutativa,

$$\mathbf{AB} \neq \mathbf{BA}$$

La multiplicación de la matriz identidad por una matriz cuadrada tiene las siguientes propiedades:

$$\mathbf{AI} = \mathbf{A}$$

o,

$$\mathbf{IA} = \mathbf{A}$$

La división entre matrices no está definida.

La matriz inversa, \mathbf{C}^{-1} , de la matriz cuadrada \mathbf{C} , está definida por:

$$\mathbf{CC}^{-1} = \mathbf{I}$$

También,

$$(\mathbf{C}^{-1})^{-1} = \mathbf{C}$$

La transpuesta de una matriz \mathbf{A} puede obtenerse sustituyendo a_{ij} por a_{ji} . Esto es:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}, \quad \mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

Orden: 2x3 3x2

4.2 Sistema de ecuaciones lineales.

En este subcapítulo se resolverán sistemas de n ecuaciones con n incógnitas, que tienen la siguiente forma:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\dots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{4.1}$$

En caso de que todas las ecuaciones que componen el sistema (4.1) sean lineales, se puede hacer la simplificación mostrada en la ecuación (4.2):

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + \dots + a_{1n} x_n &= b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + \dots + a_{2n} x_n &= b_2 \\ &\dots \\ a_{n1} x_1 + a_{n2} x_2 + a_{n3} x_3 + \dots + a_{nn} x_n &= b_n \end{aligned} \tag{4.2}$$

Sin embargo, si el sistema es no lineal, el procedimiento se complica porque las incógnitas de cada ecuación pueden tener más de una solución.

Analizando un sistema simple de dos ecuaciones con dos incógnitas, se pueden establecer tres alternativas:

- a) El sistema tiene una solución única. Por ejemplo:

$$2x + y = 4$$

$$x + y = -1$$

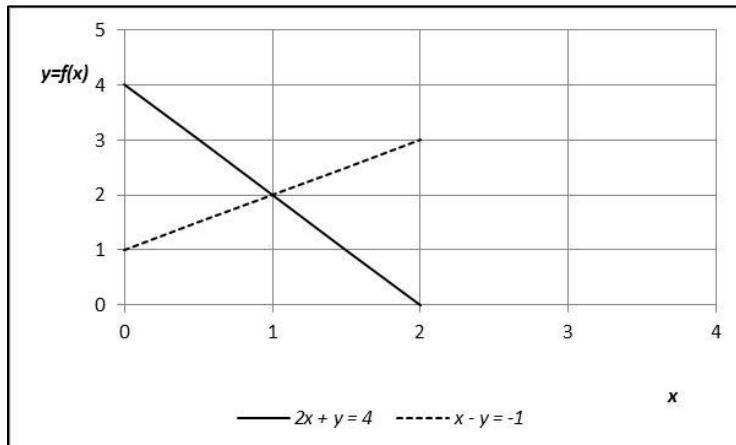


Figura 4.1. Ejemplo de un sistema de ecuaciones con solución única.

Este tipo de sistema es el que se estudiará en este subcapítulo.

- b) El sistema no tiene solución. Por ejemplo:

$$4x + 6y = 10$$

$$2x + 3y = 6$$

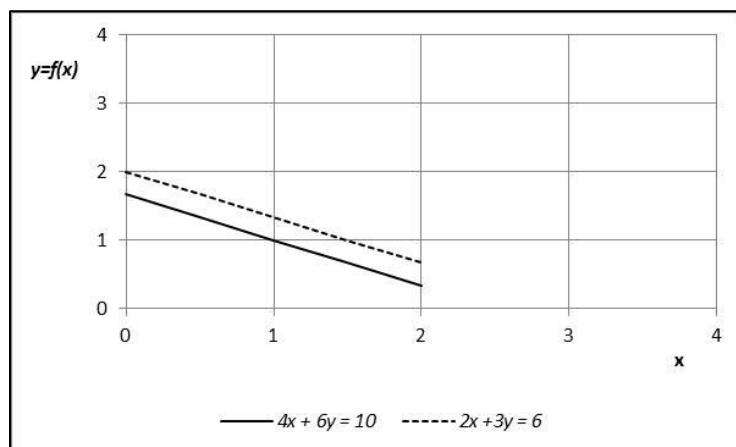


Figura 4.2. Ejemplo de un sistema de ecuaciones sin solución.

- c) El sistema tiene un número infinito de soluciones. Por ejemplo.

$$4x + 6y = 12$$

$$2x + 3y = 6$$

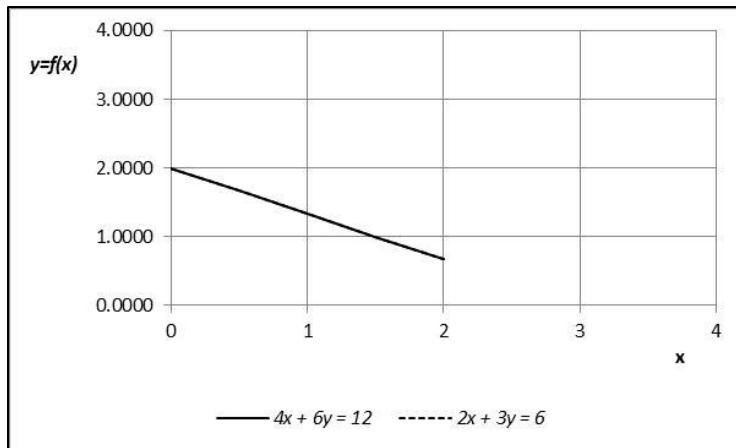


Figura 4.3. Ejemplo de un sistema de soluciones con un número infinito de soluciones.

Por otro lado, se denomina *singular* a un sistema del tipo b) o c). Una prueba directa de singularidad consiste en evaluar el determinante de la matriz formada por los elementos del primer miembro del sistema de ecuaciones; si es igual a cero, indicará singularidad.

Para explicar los métodos de solución de sistemas de ecuaciones lineales, primero se particularizará para conceptualizar. Considérese el siguiente sistema de tres ecuaciones lineales:

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + a_{13} x_3 &= b_1 \\ a_{21} x_1 + a_{22} x_2 + a_{23} x_3 &= b_2 \\ a_{31} x_1 + a_{32} x_2 + a_{33} x_3 &= b_3 \end{aligned} \tag{4.3}$$

El cual puede ser representado matricialmente de la siguiente forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.4}$$

Para encontrar el vector solución \mathbf{x} se usan dos reglas básicas del equilibrio de las ecuaciones, por medio de las cuales se puede modificar el sistema sin alterarlo. Por ejemplo:

- a) **R1: Multiplicar, o dividir cualquier ecuación, del sistema por una constante.**
- b) **R2: Reemplazar cualquier ecuación pro la suma, o resta, de otras dos ecuaciones del sistema.**

Es decir, al no alterar el comportamiento de una o más ecuaciones, el sistema tampoco se alterará.

4.2.1 Eliminación de Gauss.

Para aplicar este método se usan las dos propiedades mencionadas anteriormente para modificar ecuaciones del sistema, sin alterar el sistema. Considerando el sistema (4.4) y sabiendo que lo que se modifica al procesar las matrices son la matriz \mathbf{A} y el vector \mathbf{b} , y el vector \mathbf{x} se mantiene intacto. Entonces, el sistema (4.4) se puede expresar como una matriz aumentada para procesarla más eficientemente:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \quad (4.5)$$

Se divide el renglón pivote 1 entre el elemento pivote a_{11} , (R1).

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \quad (4.6)$$

Se elimina el término de x_1 del segundo renglón dividiendo el segundo renglón entre a_{21} , (R1) y restando los renglones 1 y 2, (R2). Se elimina el término de x_1 del tercer renglón dividiendo el tercer renglón entre a_{31} , (R1) y restando los renglones 1 y 3, (R2).

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & a'_{32} & a'_{33} & b'_3 \end{bmatrix} \quad (4.7)$$

Renglón pivote 2 entre el elemento pivote a_{22} , (R1).

$$\begin{bmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & a'_{32} & a''_{33} & b''_3 \end{bmatrix} \quad (4.8)$$

Se elimina el término de x_2 del tercer renglón dividiendo el tercer renglón entre a'_{32} , (R1), y restando los renglones 2 y 3, (R2).

$$\left[\begin{array}{cccc} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & a'''_{33} & b'''_3 \end{array} \right] \quad (4.9)$$

La matriz aumentada (4.9) puede expresarse algebraicamente de la siguiente manera:

$$\begin{aligned} x_1 + a'_{12} x_2 + a'_{13} x_3 &= b'_1 \\ x_2 + a'''_{23} x_3 &= b'''_2 \\ x_3 &= b^V_3 \end{aligned} \quad (4.10)$$

Obteniendo primero x_3 , luego x_2 y finalmente x_1 .

$$\begin{aligned} x_3 &= b^V_3 \\ x_2 &= b'''_2 - a'''_{23} x_3 \\ x_1 &= b'_1 - a'_{12} x_2 - a'_{13} x_3 \end{aligned} \quad (4.11)$$

Hasta aquí se presenta el método de eliminación de Gauss.

Un inconveniente de este método es que el elemento pivote siempre se encuentra en la diagonal principal, por lo que si es muy pequeño o igual a cero, comparado con el resto de los elementos del renglón pivote, puede provocarse inexactitud en la solución del sistema, o una indeterminación. Este inconveniente puede ser resuelto intercambiando filas y colocando los elementos mayores en la diagonal principal o definitivamente usando otro método de solución, como el del pivote máximo o el de la matriz inversa.

4.2.2 Método de Gauss-Jordan.

Este método es una continuación del método de eliminación de Gauss, es decir, una vez que se llega al sistema solución de eliminación de Gauss, se procede de manera similar pero inversa (de abajo para arriba). Entonces, iniciando en donde se termina en el punto anterior,

$$\left[\begin{array}{cccc} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & a'''_{33} & b'''_3 \end{array} \right] \quad (4.9)$$

Se divide el renglón pivote 3 entre el elemento pivote a'''_{33} , (R1).
(Sustitución inversa).

$$\left[\begin{array}{cccc} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & 1 & b^{IV} \end{array} \right] \quad (4.10)$$

Se elimina el término de x_3 del segundo renglón dividiendo el segundo renglón por a''_{23} , (R1) y restando los renglones 2 y 3, (R2). Se elimina el término de x_3 del primer renglón dividiendo el primer renglón entre a'_{13} , (R1) y restando los renglones 1 y 3, (R2).

$$\left[\begin{array}{cccc} 1 & a''_{12} & 0 & b''_1 \\ 0 & 1 & 0 & b'''_2 \\ 0 & 0 & 1 & b_3^{IV} \end{array} \right] \quad (4.11)$$

Se elimina el término de x_2 del primer renglón dividiendo el primer renglón entre a''_{12} , (R1), y restando los renglones 2 y 3, (R2).

$$\left[\begin{array}{cccc} 1 & 0 & 0 & b'''_1 \\ 0 & 1 & 0 & b'''_2 \\ 0 & 0 & 1 & b_3^{IV} \end{array} \right] \quad (4.12)$$

Así se llega a la solución final, la ecuación (4.13) obtenida anteriormente,

$$\begin{aligned} x_1 &= b_1^{VI} \\ x_2 &= b_2^{VI} \\ x_3 &= b_3^V \end{aligned} \quad (4.13)$$

Al igual que el método de eliminación de Gauss, se tiene que procurar que en la diagonal principal no se encuentre un término igual a cero.

4.2.3 Método de la matriz inversa.

Por propiedades de las matrices, visto en el primer inciso del presente capítulo, se puede comprobar lo siguiente:

$$\mathbf{A}\mathbf{I} = \mathbf{I}\mathbf{A} = \mathbf{A}$$

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Entonces, algebraicamente se pueden aprovechar estas propiedades,

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

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

$$\mathbf{I}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

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

Básicamente existen dos métodos para obtener la matriz inversa:

4.2.3.1 Método de Gauss-Jordan

Teniendo una matriz cuadrada n , el método consiste principalmente en³:

- (a) Generar una matriz de orden $n \times 2n$ de tal manera que las primeras n columnas sean de la matriz \mathbf{A} y las siguientes n columnas de la matriz identidad de la matriz n .
- (b) Aplicando Gauss-Jordan en ambas matrices simultáneamente, convertir la primera en una matriz identidad y la segunda se convertirá precisamente en la matriz inversa.

Es decir, se colocan juntas la matriz a invertir y la matriz identidad, por decir, \mathbf{AI} ; y mediante el procedimiento de Gauss-Seidel, transformarlas en \mathbf{IA} . Es decir, todos los pasos que se apliquen en la matriz \mathbf{A} se deben aplicar simultáneamente en la matriz identidad, de tal manera que automáticamente se invierte el orden del producto inicial de ambas matrices.

4.2.3.2 Adjuntos y determinantes.

Para determinar la matriz inversa de \mathbf{A} , estableciendo que su determinante es diferente de cero, se aplica la siguiente fórmula:

$$\mathbf{A}^{-1} = \frac{(\mathbf{A}^d)^T}{|\mathbf{A}|}$$

³ Tomado de https://www.uoc.edu/in3/emath/docs/Matriz_Inversa.pdf

donde A^d es la matriz de adjuntos de A y $(A^d)^T$, su transpuesta.

4.2.4 Ejemplos.

Ejemplo 4.2.4.1 Eliminación de Gauss. Determinar las raíces del siguiente sistema de ecuaciones lineales:

$$\begin{aligned} 3x_1 + x_2 - x_3 &= 2 \\ x_1 + 4x_2 + x_3 &= 12 \\ 2x_1 + x_2 + 2x_3 &= 10 \end{aligned}$$

Expresando el sistema matricialmente,

$$\begin{bmatrix} 3 & 1 & -1 \\ 1 & 4 & 1 \\ 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 12 \\ 10 \end{bmatrix}$$

Y para un procesamiento más eficiente se representa como matriz aumentada, la cual será procesada con las reglas $R1$ y $R2$ para que el sistema no se altere,

$$\left[\begin{array}{ccc|c} 3 & 1 & -1 & 2 \\ 1 & 4 & 1 & 12 \\ 2 & 1 & 2 & 10 \end{array} \right]$$

Tomando el primer renglón y su primer elemento como pivotes, se harán cero los primeros elementos del 2do y 3er renglones. Entonces, dividiendo el primer renglón entre su primer elemento ($R1$), 3,

$$\left[\begin{array}{ccc|c} 1 & 1/3 & -1/3 & 2/3 \\ 1 & 4 & 1 & 12 \\ 2 & 1 & 2 & 10 \end{array} \right]$$

Dividiendo el segundo renglón entre 1 ($R1$) y restándolo del primero se obtiene el segundo renglón ($R2$). Similarmente, dividiendo el tercer renglón entre 2 ($R1$) y restándolo del primero se obtiene el tercer renglón ($R2$),

$$\left[\begin{array}{ccc|c} 1 & 1/3 & -1/3 & 2/3 \\ 0 & -1/3 & -4/3 & -34/3 \\ 0 & -1/6 & -4/3 & -13/3 \end{array} \right]$$

$$\left[\begin{array}{ccc|c} 1 & 1/3 & -1/3 & 2/3 \\ 0 & -11/3 & -4/3 & -34/3 \\ 0 & -1/6 & -4/3 & -13/3 \end{array} \right]$$

Ahora tomando el segundo renglón y su segundo elemento como pivote, se elimina el segundo elemento del 3er renglón. Se divide el segundo renglón entre su elemento pivote -11/3, (*R1*)

$$\begin{bmatrix} 1 & 1/3 & -1/3 & 2/3 \\ 0 & 1 & 4/11 & 34/11 \\ 0 & -1/6 & -4/3 & -13/3 \end{bmatrix}$$

Dividiendo el segundo renglón entre -1/6 (*R1*) y restándolo del segundo se obtiene el tercer renglón (*R2*).

$$\begin{bmatrix} 1 & 1/3 & -1/3 & 2/3 \\ 0 & 1 & 4/11 & 34/11 \\ 0 & 0 & -84/11 & -252/11 \end{bmatrix}$$

Expresando la matriz anterior como un sistema de ecuaciones,

$$\begin{aligned} x_1 + (1/3)x_2 - (1/3)x_3 &= 2/3 \\ x_2 + (4/11)x_3 &= 34/11 \\ \left(-\frac{84}{11}\right)x_3 &= -252/11 \end{aligned}$$

Este sistema tiene el mismo comportamiento que el sistema original, solo que ahora si se aprecia su solución fácilmente. Entonces,

$$x_3 = \frac{-252/11}{-84/11} = 3$$

$$x_2 = \frac{34}{11} - \frac{4}{11}(3) = 2$$

$$x_1 = \frac{2}{3} - \frac{1}{3}(2) + \frac{1}{3}(3) = 1$$

➤ Usando comandos de Octave se obtienen lo siguiente:

```
>> A=[3 1 -1; 1 4 1; 2 1 2];
>> b=[2; 12; 10];
>> x=A\b
x =
1.00000
2.00000
3.00000
```

>>

- Aplicando el código en Octave, *Gauss_Jordan_Octave.m*. En la figura 4.4 se muestran los resultados del ejemplo 4.2.4.1.
 - Observar que para ejecutar el código se requiere que previamente se genere una hoja en Excel que contenga los términos del sistema de ecuaciones lineales. Para esto, en caso de requerirse, modificar las líneas del código en Octave en donde se leen las matrices **A** y **b**.

```
A=xlsread('Libro1.xlsx','Hoja1','A1:C3');
b.xlsread('Libro1.xlsx','Hoja1','D1:D3');
```

Figura 4.4. Resultados del ejemplo de eliminación de Gauss-Jordan.

	A	B	C	D	
1	3	1	-1	2	
2	1	4	1	12	
3	2	1	2	10	
4					
5					
-					

Detected XLS interfaces:

Método de Gauss-Jordan.

```
3      1     -1      2
1      4      1     12
2      1      2     10

1.00000   0.33333   -0.33333    0.66667
1.00000   4.00000   1.00000   12.00000
2.00000   1.00000   2.00000   10.00000

1.00000   0.33333   -0.33333    0.66667
0.00000  -3.66667   -1.33333  -11.33333
0.00000  -0.16667   -1.33333   -4.33333

1.00000   0.33333   -0.33333    0.66667
-0.00000   1.00000   0.36364   3.09091
0.00000  -0.16667   -1.33333   -4.33333
```

```

1.00000  0.33333 -0.33333  0.66667
-0.00000  1.00000  0.36364  3.09091
 0.00000  0.00000 -7.63636 -22.90909

```

x =

```
1.00000  2.00000  3.00000
```

>>

Ejemplo 4.2.4.2 Eliminación de Gauss-Jordan. Resolver de nuevo el ejemplo 4.2.4.1 aplicando ahora Gauss-Jordan.

Debido a que Gauss-Jordan es una continuación del método de Gauss, se partirá de la última matriz determinada con este.

$$\begin{bmatrix} 1 & 1/3 & -1/3 & 2/3 \\ 0 & 1 & 4/11 & 34/11 \\ 0 & 0 & -84/11 & -252/1 \end{bmatrix}$$

Ahora el 3er renglón y su 3er elementos son los pivotes con los que se harán cero los terceros elementos del 1er y 2do renglones. Se divide el tercer renglón entre su elemento pivote, -84/11,
(R1)

$$\begin{bmatrix} 1 & 1/3 & -1/3 & 2/3 \\ 0 & 1 & 4/11 & 34/11 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Dividiendo el segundo renglón entre 4/11 (R1) y restándolo del tercero se obtiene el segundo renglón (R2). Similarmente, dividiendo el primer renglón entre -1/3 (R1) y restándolo del primero se obtiene el primer renglón (R2),

$$\begin{bmatrix} 3 & 1 & 0 & 5 \\ 0 & -1/4 & 0 & -11/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Ahora el 2do renglón y su 2do elemento son los pivotes. Se divide el segundo renglón entre su elemento pivote -1/4, (R1)

$$\begin{bmatrix} 3 & 1 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Dividiendo el primer renglón entre 1 (*R1*) y restándolo del segundo se obtiene el primer renglón (*R2*).

$$\begin{bmatrix} -3 & 0 & 0 & -3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Finalmente, dividiendo el 1er renglón entre su primer elemento -3, (*R1*),

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Obteniéndose,

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \\ x_3 &= 3 \end{aligned}$$

➤ Usando comandos de Octave, se obtienen lo siguiente:

```
>> A=[3 1 -1; 1 4 1; 2 1 2];
>> b=[2; 12; 10];
>> x=A\b
x =
1.00000
2.00000
3.00000
```

>>

➤ En la figura 4.4 se muestran los resultados del ejemplo 4.2.4.1 aplicando el código en Octave, *Gauss_Jordan_Octave.m*.

- Observar que en procedimiento establecido en este código se incluye eliminación de Gauss, por ser parte de Gauss-Gordan. También, que para ejecutar el código se requiere que previamente se genere una hoja en Excel que contenga los términos del sistema de ecuaciones lineales. Para esto, en caso de requerirse, modificar las líneas del código en Octave en donde se leen las matrices **A** y **b**.

```
A=xlsread('Libro1.xlsx','Hoja1','A1:C3');
b.xlsread('Libro1.xlsx','Hoja1','D1:D3');
```

Figura 4.4. Resultados del ejemplo de eliminación de Gauss-Jordan.

	A	B	C	D	
1	3	1	-1	2	
2	1	4	1	12	
3	2	1	2	10	
4					
5					
-					

Detected XLS interfaces:

Método de Gauss-Jordan.

3	1	-1	2
1	4	1	12
2	1	2	10

Sustitución hacia adelante:

1.00000	0.33333	-0.33333	0.66667
1.00000	4.00000	1.00000	12.00000
1.00000	0.50000	1.00000	5.00000
1.00000	0.33333	-0.33333	0.66667
0.00000	-3.66667	-1.33333	-11.33333
0.00000	-0.16667	-1.33333	-4.33333
1.00000	0.33333	-0.33333	0.66667
-0.00000	1.00000	0.36364	3.09091
-0.00000	1.00000	8.00000	26.00000
1.00000	0.33333	-0.33333	0.66667
-0.00000	1.00000	0.36364	3.09091
0.00000	0.00000	-7.63636	-22.90909

Sustitución inversa:

-3.00000	-1.00000	1.00000	-2.00000
-0.00000	2.75000	1.00000	8.50000
-0.00000	-0.00000	1.00000	3.00000
3.00000	1.00000	0.00000	5.00000
0.00000	-2.75000	0.00000	-5.50000

```

-0.00000 -0.00000 1.00000 3.00000

3.00000 1.00000 0.00000 5.00000
-0.00000 1.00000 -0.00000 2.00000
-0.00000 -0.00000 1.00000 3.00000

-3.00000 0.00000 -0.00000 -3.00000
-0.00000 1.00000 -0.00000 2.00000
-0.00000 -0.00000 1.00000 3.00000

```

Matriz aumentada final:

```

1.00000 0.00000 -0.00000 1.00000
-0.00000 1.00000 -0.00000 2.00000
-0.00000 -0.00000 1.00000 3.00000

x =
1.00000
2.00000
3.00000

```

>>

4.2.5 Ejercicios propuestos.

- a) Resolver por eliminación de Gauss, por Gauss-Jordan y con Excel los dos ejercicios siguientes.

$$\begin{aligned}
 3x_1 + 2x_2 + 7x_3 &= 15 \\
 -2x_1 + 4x_2 - 3x_3 &= 12 \\
 -x_1 + 9x_2 + 4x_3 &= 27
 \end{aligned}$$

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

4.3 Sistema de ecuaciones no lineales.

En este caso, las variables que intervienen en el sistema de ecuaciones no lineales pueden tener más de una raíz, por lo que el procedimiento para su resolución es matemáticamente más complejo que cuando se tiene un sistema de ecuaciones lineales.

4.3.1 Método de Newton-Raphson.

Suponer un sistema de dos ecuaciones no lineales,

$$\begin{aligned} f_1(x_1, x_2) &= 0 \\ f_2(x_1, x_2) &= 0 \end{aligned} \tag{4.14}$$

Para iniciar el proceso iterativo, se supone un valor inicial para cada variable independiente, $x_{1,0}$ y $x_{2,0}$. También se supone que existen incrementos de cada variable independiente, $h_{1,0}$ y $h_{2,0}$, los cuales deben ser determinados y procesados de tal manera que sean inapreciables como incrementos de cada variable,

$$\begin{aligned} x_{1,1} &= x_{1,0} + h_{1,0} \\ x_{2,1} &= x_{2,0} + h_{2,0} \end{aligned} \tag{4.15}$$

donde el primer subíndice es el número de la variable y el segundo de la iteración.

Como se pretende que durante el procedimiento los incrementos sean menores que cierta tolerancia prefijada, insertando el sistema (4.15) dentro del (4.14). Entonces, para la 1ra iteración,

$$\begin{aligned} f_1(x_{1,1}, x_{2,1}) &= f_1(x_{1,0} + h_{1,0}, x_{2,0} + h_{2,0}) = 0 \\ f_2(x_{1,1}, x_{2,1}) &= f_2(x_{1,0} + h_{1,0}, x_{2,0} + h_{2,0}) = 0 \end{aligned} \tag{4.16}$$

Usando series de Taylor, expandiendo alrededor de $(x_{1,0} + h_{1,0}, x_{2,0} + h_{2,0})$ se obtiene:

$$\begin{aligned}
 f_1(x_{1,0} + h_{1,0}, x_{2,0} + h_{2,0}) &= f_1(x_{1,0}, x_{2,0}) + h_{1,0} \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_1} \\
 &\quad + h_{2,0} \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_2} \approx 0 \\
 f_2(x_{1,0} + h_{1,0}, x_{2,0} + h_{2,0}) &= f_2(x_{1,0}, x_{2,0}) + h_{1,0} \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_1} \\
 &\quad + h_{2,0} \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_2} \approx 0
 \end{aligned} \tag{4.17}$$

Obteniéndose,

$$\begin{aligned}
 f_1(x_{1,0}, x_{2,0}) + h_{1,0} \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_1} + h_{2,0} \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_2} &= 0 \\
 f_2(x_{1,0}, x_{2,0}) + h_{1,0} \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_1} + h_{2,0} \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_2} &= 0
 \end{aligned} \tag{4.18}$$

De manera matricial,

$$\begin{bmatrix} \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_1} & \frac{\partial f_1(x_{1,0}, x_{2,0})}{\partial x_2} \\ \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_1} & \frac{\partial f_2(x_{1,0}, x_{2,0})}{\partial x_2} \end{bmatrix} \begin{bmatrix} h_{1,0} \\ h_{2,0} \end{bmatrix} = \begin{bmatrix} -f_1(x_{1,0}, x_{2,0}) \\ -f_2(x_{1,0}, x_{2,0}) \end{bmatrix} \tag{4.19}$$

Para la iteración k ,

$$\begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1} \right)_k & \left(\frac{\partial f_1}{\partial x_2} \right)_k \\ \left(\frac{\partial f_2}{\partial x_1} \right)_k & \left(\frac{\partial f_2}{\partial x_2} \right)_k \end{bmatrix} \begin{bmatrix} h_{1,k} \\ h_{2,k} \end{bmatrix} = \begin{bmatrix} -f_{1,k} \\ -f_{2,k} \end{bmatrix} \tag{4.20}$$

Y

$$\begin{aligned}
 x_{1,k+1} &= x_{1,k} + h_{1,k} \\
 x_{2,k+1} &= x_{2,k} + h_{2,k}
 \end{aligned} \tag{4.21}$$

Llegando a la solución cuando

$$\begin{aligned} h_{1,0} &< tol \\ h_{2,0} &< tol \end{aligned} \quad (4.22)$$

Generalizando, sean los vectores,

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^t \quad (4.23)$$

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})]^t \quad (4.24)$$

$$\mathbf{J}(\mathbf{x}) = f_{i,j}(\mathbf{x}) = \frac{\partial f_i(\mathbf{x})}{\partial x_j}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq n \quad (4.25)$$

Al $\det(\mathbf{J}(\mathbf{x}))$ se le denomina Jacobiano del sistema (4.25) y

$$\mathbf{h} = [h_1, h_2, \dots, h_n]^t \quad (4.26)$$

Es decir,

$$\begin{bmatrix} [f_{11}(\mathbf{x})]_k & [f_{12}(\mathbf{x})]_k & \dots & [f_{1n}(\mathbf{x})]_k \\ [f_{21}(\mathbf{x})]_k & [f_{22}(\mathbf{x})]_k & \dots & [f_{2n}(\mathbf{x})]_k \\ \vdots & \vdots & \ddots & \vdots \\ [f_{n1}(\mathbf{x})]_k & [f_{n2}(\mathbf{x})]_k & \dots & [f_{nn}(\mathbf{x})]_k \end{bmatrix} \begin{bmatrix} h_{ik} \\ h_{2,k} \\ \vdots \\ h_{n,k} \end{bmatrix} = \begin{bmatrix} -f_{1,k}(\mathbf{x}) \\ -f_{2,k}(\mathbf{x}) \\ \vdots \\ -f_{n,k}(\mathbf{x}) \end{bmatrix} \quad (4.27)$$

El proceso iterativo se implementa aplicando:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k \quad (4.28)$$

Resulta obvio que cuando el vector $h = 0$, se cumplirá la ecuación (4.28), obteniéndose una solución del sistema.

El método de Newton-Raphson converge siempre que se le asignen valores de \mathbf{x}_0 en la vecindad de las raíces y que $\det(\mathbf{J}(\mathbf{x})) <> 0$.

[CONTENIDO](#)

4.3.2 Ejemplos.

Ejemplo 4.3.2.1 Resolver el siguiente sistema de ecuaciones no lineales por Newton-Raphson.

Iniciar iniciando con $\mathbf{x}_0 = [0.5, 0.5]$

$$f_1(x_1, x_2) = x_1^2 - 10x_1 + x_2^2 + 8 = 0$$

$$f_2(x_1, x_2) = x_1x_2^2 + x_1 - 10x_2 + 8 = 0$$

$$\frac{\partial f_1}{\partial x_1} = 2x_1 - 10, \quad \frac{\partial f_1}{\partial x_2} = 2x_2$$

$$\frac{\partial f_2}{\partial x_1} = x_2^2 + 1, \quad \frac{\partial f_2}{\partial x_2} = 2x_1x_2 - 10$$

Entonces, el Jacobiano queda:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 2x_1 - 10 & 2x_2 \\ x_2^2 + 1 & 2x_1x_2 - 10 \end{bmatrix}$$

Sustituyendo los valores iniciales y calculando su determinante se obtiene lo siguiente:

$$J(x) = \begin{vmatrix} -9.00 & 1.00 \\ 1.25 & -9.50 \end{vmatrix} = 84.25 \neq 0 \quad \therefore \text{el método converge}$$

El sistema a resolver resulta,

$$\begin{bmatrix} 2x_{1,k} - 10 & 2x_{2,k} \\ x_{2,k}^2 + 1 & 2x_{1,k}x_{2,k} - 10 \end{bmatrix} \begin{bmatrix} h_{1,k} \\ h_{2,k} \end{bmatrix} = \begin{bmatrix} -(x_{1,k}^2 - 10x_{1,k} + x_{2,k}^2 + 8) \\ -(x_1x_2^2 + x_1 - 10x_2 + 8) \end{bmatrix}$$

Y los siguientes valores del vector \mathbf{x} se calculan con,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k$$

Con una $tol = 1 \times 10^{-6}$ se obtienen los siguientes resultados, observando que el sistema se puede resolver a través de la inversa del Jacobiano,

$$\begin{bmatrix} h_{1,k} \\ h_{2,k} \end{bmatrix} = \begin{bmatrix} 2x_{1,k} - 10 & 2x_{2,k} \\ x_{2,k}^2 + 1 & 2x_{1,k}x_{2,k} - 10 \end{bmatrix}^{-1} \begin{bmatrix} -(x_{1,k}^2 - 10x_{1,k} + x_{2,k}^2 + 8) \\ -(x_1x_2^2 + x_1 - 10x_2 + 8) \end{bmatrix}$$

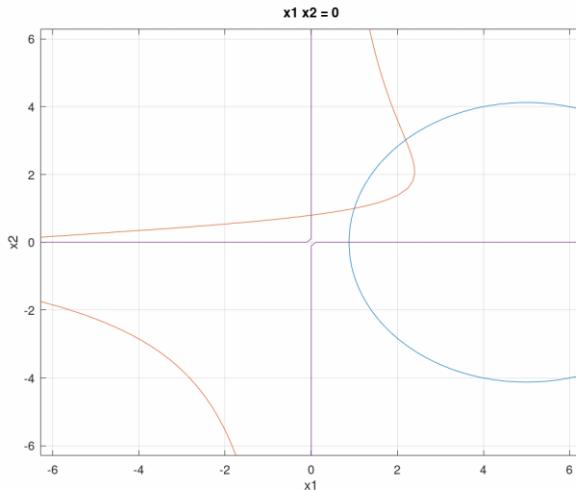
<i>k</i>	<i>x</i>	<i>J(x)</i>		<i>J⁻¹(x)</i>		<i>-f(x)</i>	<i>h</i>
0	0.500000	-9.000000	1.000000	-0.112760	-0.011869	-3.500000	0.437685
	0.500000	1.250000	-9.500000	-0.014837	-0.106825	-3.625000	0.439169
1	0.937685	-8.124629	1.878338	-0.129945	-0.029626	-0.384438	0.061009
	0.939169	1.882039	-8.238710	-0.029685	-0.128146	-0.373069	0.059219
2	0.998694	-8.002612	1.996777	-0.133252	-0.033235	-0.007229	0.001305
	0.998388	1.996779	-8.005831	-0.033235	-0.133198	-0.010289	0.001611
3	0.999999	-8.000002	1.999998	-0.133333	-0.033333	-0.000004	0.000001
	0.999999	1.999998	-8.000004	-0.033333	-0.133333	-0.000007	0.000001
4	1.000000	-8.000000	2.000000	-0.133333	-0.033333	0.000000	0.000000
	1.000000	2.000000	-8.000000	-0.033333	-0.133333	0.000000	0.000000
5	1.000000						
	1.000000						

➤ Resolviendo con comandos de Octave:

Graficando,

```
>> f1=@(x,y) x.^2-10.*x+y.^2+8;
>> f2=@(x,y) x.*y.^2+x-10.*y+8;
>> ezplot(f1)
>> hold on
>> ezplot(f2)
>> grid on
>> ezplot('x1*x2')
>>
```

En la gráfica se aprecian las ubicaciones aproximadas de las dos raíces.



Resolviendo,

```
>> pkg load symbolic  
>> syms x  
>> f=@(x) [x(1)^2-10*x(1)+x(2)^2+8; x(1)*x(2)^2+x(1)-10*x(2)+8];  
>> fsolve(f, [0.5,0.5])  
ans =  
  
1.00000 1.00000  
  
>>
```

O también,

```
>> function y = f(x)  
y(1) = x(1)^2-10*x(1)+x(2)^2+8;  
y(2)=x(1)*x(2)^2+x(1)-10*x(2)+8;  
endfunction  
>> [x,info]=fsolve("f", [0.5,0.5])  
x =  
  
1.00000 1.00000  
  
info =  
  
3.3377e-008 5.0546e-008
```

>>

➤ En figura 4.4 se muestran los resultados obtenidos con código de octave, *N_R_NL_Octave.m*.

Figura 4.4. Resultados aplicando código de Octave.

Introduce una función con respecto a x1 $x1^2-10*x1+x2^2+8$

Introduce una función con respecto a x2 $x1*x2^2+x1-10*x2+8$

Método de Newton-Raphson para ecuaciones no lineales.

```
Valores iniciales  
x1 = 0.5  
x2 = 0.5  
Tolerancia = 1e-6  
Iteraciones = 20
```

Jacobiano:

```
-9.0000 1.0000
1.2500 -9.5000
```

Determinante del Jacobiano

```
84.250
```

Inversa del Jacobiano

```
-0.112760 -0.011869
-0.014837 -0.106825
```

k	x1,k	x2,k	-f1,k	-f2,k	f11,k	f12,k	f21,k	f22,k	h1,k	h2,k
0	0.500000	0.500000	-3.500000	-3.625000	-9.000000	1.000000	1.250000	-9.500000	0.437685	0.439169
1	0.937685	0.939169	-0.384438	-0.373069	-8.124629	1.875371	1.879254	-8.238710	0.060981	0.059192
2	0.998666	0.998361	-0.007398	-0.010449	-8.002668	1.997332	1.997334	-8.005941	0.001333	0.001638
3	0.999999	0.999999	-0.000003	-0.000006	-8.000001	1.999999	1.999999	-8.000003	0.000001	0.000001
4	1.000000	1.000000								

Raíces: 1.000000 1.000000

>>

- Finalmente, suponer que se inicia con $x_{1,0} = 5$ y $x_{2,0} = 5$, aplicando comandos de Octave se obtiene lo siguiente:

```
>> pkg load symbolic
>> syms x
>> f=@(x) [x(1)^2-10*x(1)+x(2)^2+8; x(1)*x(2)^2+x(1)-10*x(2)+8];
>> fsolve(f, [5,5])
ans =
2.1934 3.0205
```

>>

Como se aprecia, se obtienen otros valores del vector **x**. Lo cual es normal por ser un sistema de ecuaciones no lineales.

4.3.3 Ejercicios propuestos.

- Resolver aplicado el método de Newton-Raphson.

$$x_1^2 + x_2^2 - x_1 =$$

$$x_1^2 - x_2^2 - x_2 = 0$$

$$\frac{1}{2} \operatorname{sen}(x_1 x_2) - \frac{x_2}{4\pi} - \frac{x_1}{2} = 0$$

$$\left(1 - \frac{1}{4}\right)(e^{2x_1} - e) + \frac{e}{\pi} x_2 - 2e x_1 = 0$$

5 Interpolación polinomial

Frecuentemente los resultados obtenidos en experimentos científicos o cálculos numéricos son puntos discretos a lo largo de cierta función, tal como se muestra en la figura 5.1. Estos valores de $f(x)$ pueden estar igual o diferentemente espaciados a lo largo del eje de las abcisas.

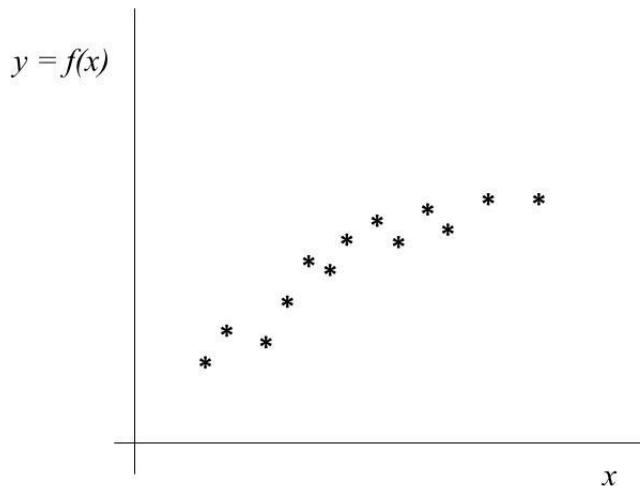


Figura 5.1. Puntos obtenidos de un experimento.

La fórmula general para un polinomio de n -ésimo grado es:

$$y = a_0 + a_1 x + a_2 x^2 + \dots + a^n x^n \quad (5.1)$$

Para $n+1$ puntos, hay uno y solo un polinomio de grado n que pasa a través de todos los puntos.
Por ejemplo, figura 5.2:

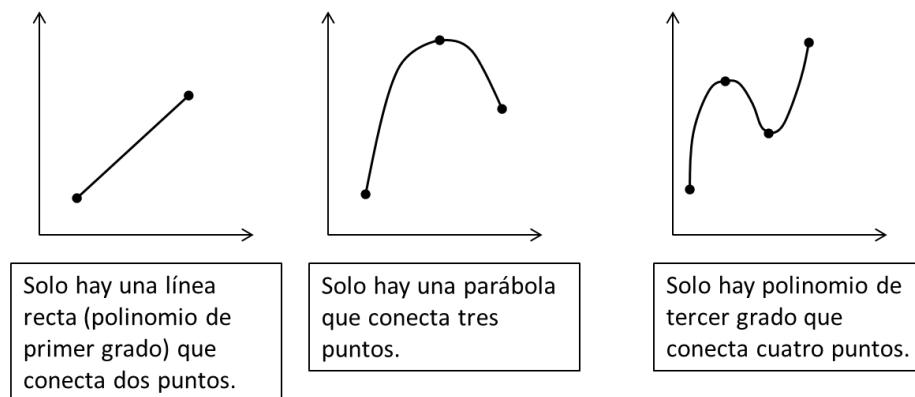


Figura 5.2. Ejemplos de puntos ajustados por polinomios.

5.1 Interpolación de Lagrange.

En la figura 5.3 se muestra una gráfica de tres puntos, a la cual se le ajustará un polinomio de 2do grado,

$$y = f(x) = a_0 + a_1 x + a_2 x^2 \quad (5.2)$$

el cual pasa por los puntos conocidos (x_1, y_1) , (x_2, y_2) y (x_3, y_3) .

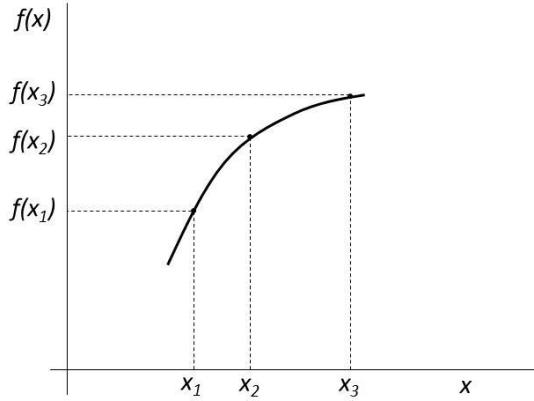


Figura 5.3. Ejemplo de ajuste de un polinomio de 2do grado.

La ecuación (5.2) se puede representar como:

$$y = A_1(x - x_2)(x - x_3) + A_2(x - x_1)(x - x_3) + A_3(x - x_1)(x - x_2) \quad (5.3)$$

ya que

$$\begin{aligned} y &= (A_1 x - A_1 x_2)(x - x_3) + (A_2 x - A_2 x_1)(x - x_3) + (A_3 x - A_3 x_1)(x - x_2) \\ y &= A_1 x^2 A_1 x x_2 - A_1 x x_3 + A_1 x_2 x_3 + A_2 x^2 - A_2 x x_3 - A_2 x x_1 + A_2 x_1 x_3 + A_3 x^2 - A_3 x x_2 - A_3 x x_1 + A_3 x_1 x_2 \\ y &= (A_1 x_2 x_3 + A_2 x_1 x_3 + A_3 x_1 x_2) + [-A_1(x^2 + x_3) - A_2(x_3 + x_1) - A_3(x_2 + x_1)]x + (A_1 + A_2 + A_3)x^2 \\ a_0 &= A_1 x_2 x_3 + A_2 x_1 x_3 + A_3 x_1 x_2 \\ a_1 &= -A_1(x_2 + x^3) - A_2(x_3 + x_1) - A_3(x_2 + x_1) \\ a_2 &= A_1 + A_2 + A_3 \end{aligned} \quad (5.4)$$

Los coeficientes A_1 , A_2 y A_3 , para este caso, se calculan de tal manera que la curva del polinomio pase por los tres puntos conocidos.

Procesando la ecuación (5.3) e iniciando por hacer $x = x_1$ se obtiene,

$$y_1 = A_1 (x_1 - x_2)(x_1 - x_3)$$

despejando A_1 ,

$$A_1 = \frac{y_1}{(x_1 - x_2)(x_1 - x_3)} \quad (5.5)$$

Si $x = x_2$,

$$y_2 = A_2 (x_2 - x_1)(x_2 - x_3)$$

Despejando A_2 ,

$$A_2 = \frac{y_2}{(x_2 - x_1)(x_2 - x_3)} \quad (5.6)$$

Finalmente, si $x = x_3$,

$$y_3 = A_3 (x_3 - x_1)(x_3 - x_2)$$

Despejando A_3 ,

$$A_3 = \frac{y_3}{(x_3 - x_1)(x_3 - x_2)} \quad (5.7)$$

Entonces, una vez conocidos los valores de A_1, A_2 y A_3 se puede aplicar la ecuación (5.3) para interpolar cualquier punto entre $(x_1, y_1), (x_2, y_2)$ y (x_3, y_3) .

En forma general, sea un polinomio de grado m y n puntos conocidos,

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_{n-1} x^{n-1} + a_n x^n \quad (5.8)$$

el cual también se puede representar como:

$$\begin{aligned} f(x) &= A_1 (x - x_1)(x - x_2)(x - x_3)(x - x_4) \dots (x - x_n) + \\ &\quad A_2 (x - x_1)(x - x_2)(x - x_3)(x - x_4) \dots (x - x_n) + \\ &\quad A_3 (x - x_1)(x - x_2)(x - x_3)(x - x_4) \dots (x - x_n) + \\ &\quad \dots \\ &\quad A_n (x - x_1)(x - x_2)(x - x_3)(x - x_4) \dots (x - x_{n-1}) \end{aligned} \quad (5.9)$$

Los coeficientes $A_1, A_2, A_3, \dots, A_n$ se calculan de tal manera que la curva del polinomio pase por todos y cada uno de los puntos conocidos $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)$.

Si $x = x_1$,

$$A_1 = \frac{y_1}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4) \dots (x_1 - x_n)}$$

Si $x = x_2$,

$$A_2 = \frac{y_2}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4) \dots (x_2 - x_n)}$$

Si $x = x_3$,

$$A_3 = \frac{y_3}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4) \dots (x_3 - x_n)}$$

Y así sucesivamente, hasta $x = x_n$

$$A_n = \frac{y_n}{(x_n - x_1)(x_n - x_2)(x_n - x_3) \dots (x_n - x_{n-1})} \quad (5.10)$$

5.2 Ejemplos.

Ejemplo 5.2.1 Se tiene la siguiente tabulación:

i	1	2	3	4
x	0	2	4	5
y	2	2	10	17

Encontrar $f(1), f(1.5)$ y $f(3)$.

- Solución.

En la figura 5.4 se muestra una gráfica con los puntos originales.

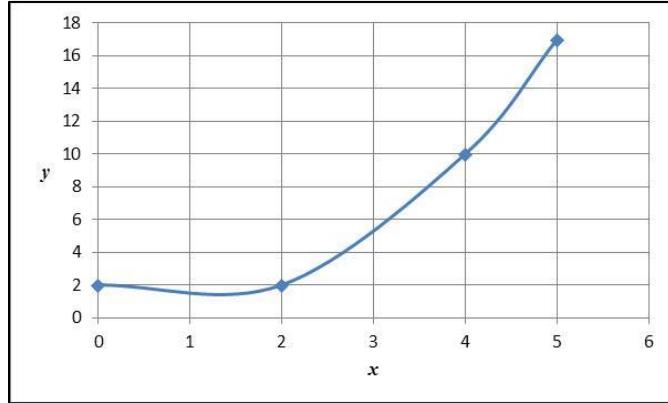


Figura 5.4. Gráfica con los puntos originales del ejemplo 5.1.

Interpolando por medio de las ecuaciones (5.9) y (5.19) se obtiene lo siguiente:

$$f(x) = A_1 \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} + A_2 \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} + A_3 \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} + A_4 \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}$$

$$A_1 = \frac{y_1}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} = \frac{2}{(0-2)(0-4)(0-5)} = -\frac{1}{20}$$

$$A_2 = \frac{y_2}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} = \frac{2}{(2-0)(2-4)(2-5)} = \frac{1}{6}$$

$$A_3 = \frac{y_3}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} = \frac{10}{(4-0)(4-2)(4-5)} = -\frac{5}{4}$$

$$A_4 = \frac{y_4}{(x_4-x_1)(x_4-x_2)(x_4-x_3)} = \frac{14}{(5-0)(5-2)(5-4)} = \frac{17}{15}$$

Entonces,

$$\begin{aligned} f(x) &= -\frac{1}{20}(x-2)(x-4)(x-5) + \frac{1}{6}(x-0)(x-4)(x-5) \\ &\quad - \frac{5}{4}(x-0)(x-2)(x-5) + \frac{17}{15}(x-0)(x-2)(x-4) \end{aligned}$$

Con esta última ecuación ya se puede interpolar para determinar la $f(x)$ para cualquier valor de x entre $0 \leq x \leq 5$.

O también se puede reducir aún más, hasta obtener su representación polinómica:

$$f(x) = x^2 - 2x + 2$$

Entonces, resolviendo se obtienen los siguientes resultados:

$$f(1) = 1.00, f(1.5)=1.25, f(3) = 5.00$$

En la figura 5.5 se aprecian una gráfica con los valores originales, incluyendo los calculados en la interpolación.

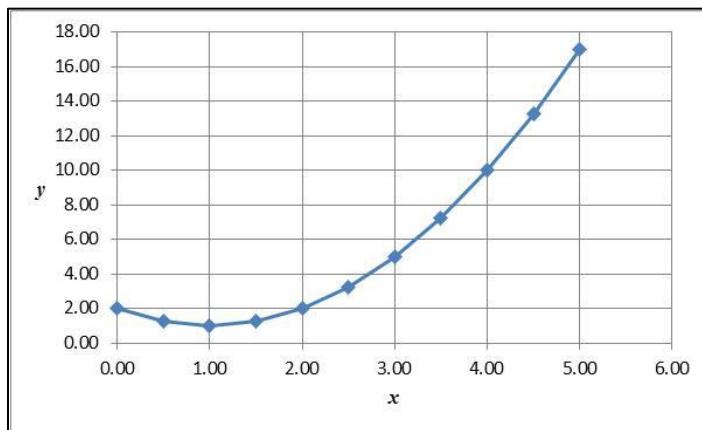


Figura 5.5. Gráfica incluyendo puntos originales e interpolados en el ejemplo 5.1.

➤ Usando comandos de Octave se obtiene lo siguiente:

```
>> x=[0 2 4 5];
>> y=[2 2 10 17];
>> P3=polyfit(x,y,3)
P3 =
1.0103e-016 1.0000e+000 -2.0000e+000 2.0000e+000

>> polyval(P3,1)
ans = 1.0000
>> polyval(P3,1.5)
ans = 1.2500
>> polyval(P3,3)
ans = 5.0000
>>
```

➤ En la figura 5.5 se muestran los resultados usando código de Octave, *Lagrange_Octave.m*.

- Inicialmente se genera una hoja en Excel, en este caso *Libro1.xlsx* con los valores de x y y . Posteriormente se adaptan las líneas correspondientes en dicho código:

```
X=xlsread('Libro1.xlsx','Hoja1','A1:C1');  
Y=xlsread('Libro1.xlsx','Hoja1','A2:C2');
```

	A	B	C	D	E
1	0	2	4	5	
2	2	2	10	17	
3					
4					

Luego se ejecuta,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

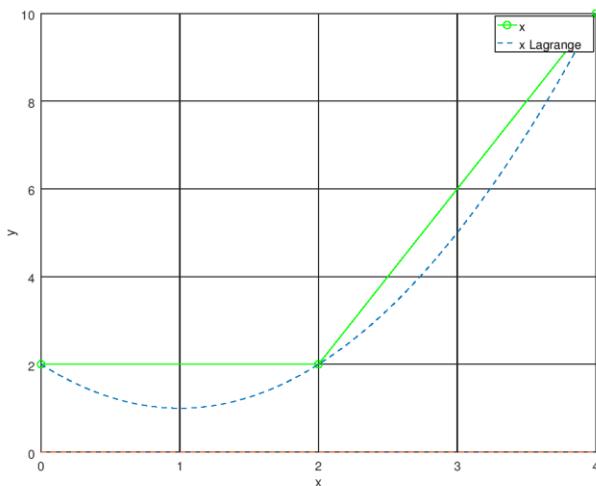
Detected XLS interfaces:

Valores de las
3 Aes

```
0.250000  
-0.500000  
1.250000
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 1  
1.000000  
  
x1 = 1.5  
1.250000  
  
x1 = 3  
5.000000  
  
x1 =  
>>
```



5.3 Ejercicios propuestos.

- a) Dada la siguiente función tabulada

i	1	2	3	4	5	6	7
x_i	0	0.3	0.6	0.9	1.2	1.5	1.8
$f(xi)$	-3.000	-0.742	2.143	6.452	14.579	31.480	65.628

Encontrar (1) $f(1.09)$, (2) $f(0.93)$, (3) $f(1.42)$, (4) $f(0.21)$.

- b) Usando interpolación de Lagrange, encontrar $f(4.3)$ para la siguiente función:

i	1	2	3	4	5
x_i	0.0	1.0	2.0	3.8	5.0
$f(xi)$	0.000	0.569	0.791	0.224	-0.185

6 Aproximación funcional

En el capítulo anterior, a partir de puntos discretos conocidos, se obtuvo una función en función de x . En este capítulo, se ajustarán curvas a esas series de puntos discretos, en particular, serán polinómicas y se usará el método de mínimos cuadrados.

6.1 Método de mínimos cuadrados.

Suponer la función,

$$y_i = c_0 + c_1 x_i + c_2 x_i^2 \quad (6.1)$$

y que su gráfica es:

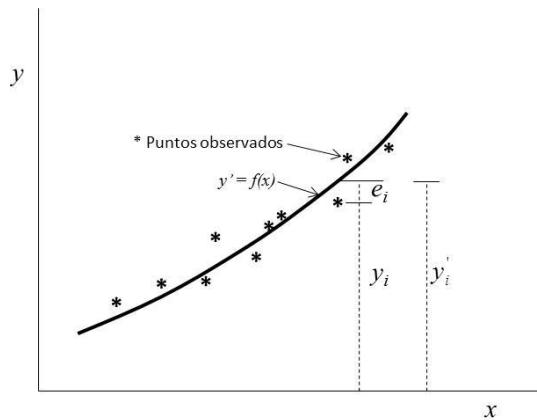


Figura 6.1. Mínimos cuadrados.

El error para cada punto será:

$$e_i = y_i - y'_i \quad (6.2)$$

Notar que la ecuación (6.1) proporciona una estimación de la ecuación real, debiéndose estimar los valores de c_0 , c_1 y c_2 de tal manera que los errores estimados sean lo más pequeños posible.

Una forma de minimizar la suma de los errores, es que la suma de los cuadrados de los errores e_i sea cero. Es decir, se determinan los valores de c_0 , c_1 y c_2 minimizando la siguiente sumatoria:

$$S = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - (c_0 + c_1 x_i + c_2 x_i^2)]^2 \quad (6.3)$$

donde n = número de puntos discretos conocidos.

Para determinar los valores de c_0 , c_1 y c_2 que minimizan la ecuación (6.3), se deriva esta en función en función de cada uno de los parámetros a determinar y se iguala a cero,

$$\begin{aligned}\frac{dS}{dc_0} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_i - c_2 x_i^2)(-1) = 0 \\ \frac{dS}{dc_1} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_i - c_2 x_i^2)(-x_i) = 0 \\ \frac{dS}{dc_2} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_i - c_2 x_i^2)(-x_i^2) = 0\end{aligned}\tag{6.4}$$

Obteniéndose,

$$\begin{aligned}(n)c_0 + (\sum x_i)c_1 + (\sum x_i^2)c_2 &= \sum y_i \\ (\sum x_i)c_0 + (\sum x_i^2)c_1 + (\sum x_i^3)c_2 &= \sum (x_i y_i) \\ (\sum x_i^2)c_0 + (\sum x_i^3)c_1 + (\sum x_i^4)c_2 &= \sum (x_i^2 y_i)\end{aligned}\tag{6.5}$$

Como se aprecia, en el sistema (6.5) las únicas incógnitas son c_0 , c_1 y c_2 , ya que se conocen las sumatorias de los puntos observados, restando resolver tres ecuaciones lineales con tres incógnitas (c_0 , c_1 y c_2). Así se determina la ecuación (6.1) para calcular los valores de y_i .

Generalizando el procedimiento anterior y ajustando un polinomio de grado M , se tiene que,

$$\begin{aligned}(n)c_0 + (\sum x_i)c_1 + \dots + (\sum x_i^M)c_{M-1} &= \sum y_i \\ (\sum x_i)c_0 + (\sum x_i^2)c_1 + \dots + (\sum x_i^{M+1})c_{M+1} &= \sum (x_i y_i) \\ (\sum x_i^2)c_0 + (\sum x_i^3)c_1 + \dots + (\sum x_i^{M+2})c_{M+1} &= \sum (x_i^2 y_i) \\ &\vdots \\ (\sum x_i^M)c_0 + (\sum x_i^{M+1})c_1 + \dots + (\sum x_i^{2M})c_{M+1} &= \sum (x_i^M y_i)\end{aligned}\tag{6.6}$$

Notar que, para un polinomio de grado M , existen $M+1$ ecuaciones que corresponden a $M+1$ parámetros ajustables y se requieren un total de $3M+1$ sumatorias.

6.1.1 Regresión polinomial.

Este tipo de regresión relaciona una variable dependiente con una independiente.

En el análisis del inciso anterior, además de explicar y aplicar el método de mínimos cuadrados, se establecieron las ecuaciones de regresión polinomial, ecuación (6.6).

6.1.2 Regresión lineal múltiple.

Esta regresión considera más de una variable independiente,

$$\hat{y}_i = c_0 + c_1 x_{1_i} + c_2 x_{2_i} + c_3 x_{3_i} \quad (6.7)$$

donde x_1, x_2, \dots, x_n son m variables independientes. Notar que los subíndices i y m se refieren a variables diferentes, no a puntos observados. Si como frecuentemente ocurre, $x_1 = x$, $x_2 = x^2, \dots$ etc. se tiene un caso de regresión lineal múltiple, conocido como regresión polinomial o curvilínea.

Sea la ecuación,

$$\hat{y}_i = c_0 + c_1 x_{1_i} + c_2 x_{2_i} + c_3 x_{3_i} \quad (6.8)$$

Aplicando el principio de mínimos cuadrados para n puntos,

$$\begin{aligned} \frac{dS}{dc_0} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_{1_i} - c_2 x_{2_i} - c_3 x_{3_i})(-1) = 0 \\ \frac{dS}{dc_1} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_{1_i} - c_2 x_{2_i} - c_3 x_{3_i})(-x_{1_i}) = 0 \\ \frac{dS}{dc_2} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_{1_i} - c_2 x_{2_i} - c_3 x_{3_i})(-x_{2_i}) = 0 \\ \frac{dS}{dc_3} &= 2 \sum_{i=1}^n (y_i - c_0 - c_1 x_{1_i} - c_2 x_{2_i} - c_3 x_{3_i})(-x_{3_i}) = 0 \end{aligned} \quad (6.9)$$

El sistema de ecuaciones queda,

$$\begin{aligned} (n)c_0 + (\sum x_{1_i})c_1 + (\sum x_{2_i}^2)c_2 + (\sum x_{3_i})c_3 &= \sum y_i \\ (\sum x_{1_i})c_0 + (\sum x_{1_i}^2)c_1 + [\sum (x_{1_i} x_{2_i})]c_2 + [\sum (x_{1_i} x_{3_i})]c_3 &= \sum (x_{1_i} y_i) \\ (\sum x_{2_i})c_0 + [\sum (x_{1_i} x_{2_i})]c_1 + (\sum x_{2_i}^2)c_2 + (\sum x_{2_i} x_{3_i})c_3 &= \sum (x_{2_i} y_i) \\ (\sum x_{3_i})c_0 + [\sum (x_{1_i} x_{3_i})]c_1 + [\sum (x_{2_i} x_{3_i})]c_2 + (\sum x_{3_i}^2)c_3 &= \sum (x_{3_i} y_i) \end{aligned} \quad (6.10)$$

Generalizando se obtiene,

$$\begin{aligned} (n)c_0 + (\sum x_{1_i})c_1 + (\sum x_{2_i}^2)c_2 + \dots + (\sum x_{m_i})c_m &= \sum y_i \\ (\sum x_{1_i})c_0 + (\sum x_{1_i}^2)c_1 + [\sum (x_{1_i} x_{2_i})]c_2 + \dots + [\sum (x_{1_i} x_{m_i})]c_m &= \sum (x_{1_i} y_i) \\ &\dots \\ (\sum x_{m_i})c_0 + [\sum (x_{1_i} x_{m_i})]c_1 + [\sum (x_{2_i} x_{m_i})]c_2 + \dots + (\sum x_{m_i}^2)c_m &= \sum (x_{m_i} y_i) \end{aligned} \quad (6.11)$$

6.2 Ejemplos.

Ejemplo 6.2.1 Ajustar una curva de segundo grado a los datos mostrado en la siguiente tabla.
Además, estimar y para $x = 1, 1.5$ y 3 .

x	0	2	4	5
y	2	2	10	17

Solución.

Entonces,

$$M = 2$$

$$M + 1 = 3 \text{ ecuaciones}$$

$$M + 1 = 3 \text{ parámetros ajustables}$$

$$3M + 1 = 7 \text{ sumatorias}$$

Y el sistema de ecuaciones queda,

$$(n)c_0 + (\sum x_i)c_1 + (\sum x_i^2)c_2 = \sum y_i$$

$$(\sum x_i)c_0 + (\sum x_i^2)c_1 + (\sum x_i^3)c_2 = \sum(x_i y_i)$$

$$(\sum x_i^2)c_0 + (\sum x_i^3)c_1 + (\sum x_i^4)c_2 = \sum(x_i^2 y_i)$$

También,

$$N = 4, \quad \sum y_i = 31, \quad \sum x_i = 11, \quad \sum(x_i y_i) = 129, \quad \sum x_i^2 = 45, \quad \sum(x_i^2 y_i) = 593, \quad \sum x_i^3 = 197, \quad \sum x_i^4 = 897.$$

Sustituyendo en el sistema,

$$4c_0 + 11c_1 + 45c_2 = 31$$

$$11c_0 + 45c_1 + 197c_2 = 129$$

$$45c_0 + 197c_1 + 897c_2 = 593$$

Resolviendo por Gauss-Jordan se obtiene,

$$c_0 = 2, c_1 = -2, c_2 = 1$$

Por lo la ecuación ajustada queda:

$$y_i = 2 - 2x_i + x_i^2$$

Estimando en los puntos solicitados,

$$y'(1) = 1.0, \quad y'(1.5) = 1.25, \quad y'(3) = 5.0$$

Usando comandos en Octave,

```
>> x=[0 2 4 5];
>> y=[2 2 10 17];
>> p = polyfit(x,y,2)
p =
1.000000000000000 -2.000000000000000 2.000000000000000
>>
```

Entonces, $y_i = 2 - 2x_i + x_i^2$. Si se desea graficar:

```
>> xmin=min(x); xmax=max(x);
>> xx = linspace(xmin,xmax); % genera 100 valores de xx;
>> hold on
>> plot(x,y,'o') % grafica los puntos observados;
>> grid
>> yy = polyval(p,xx);
>> plot(xx,yy) % grafica una curva con los puntos estimados;
>> legend('y', 'yest')
>> hold off
>>
```

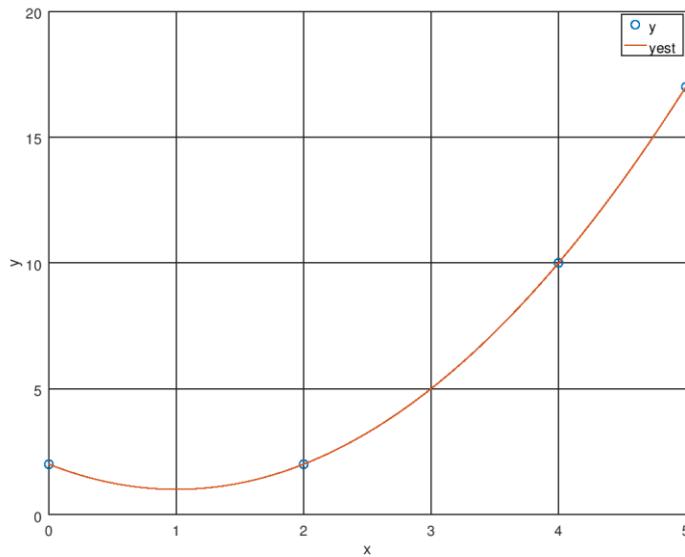


Figura 6.2. Resultado del Ejemplo 6.2.1 usando comandos de Octave.

- Usando código en Octave, *Ajuste_polinomial.m*,

Primero crear la hoja en Excel denominada, en este caso, *Libro1.xlsx* y los datos se insertaron en la Hoja 2 en el rango B18:E18 para x y B19:E19 para y ;

17					
18	x	0	2	4	5
19	y	2	2	10	17
20					

Luego se modican las líneas correspondientes en el código de Octave,

```
13 x=xlsread('Libro1.xlsx','Hoja2','B18:E18');
14 y=xlsread('Libro1.xlsx','Hoja2','B19:E19');
```

Y se ejecuta,

Crear previamente los vectores x y y en excel.
si ya lo creaste oprime Enter si no, oprime Ctrl C y Enter.

Detected XLS interfaces:
Grado del polinomio? 2

Coeficientes del polinomio ajustado

2.0000000000000002

-2.0000000000000002

1.0000000000000000

Coef. de correlación Pearson: 1.0000
Coef. de determinación Pearson: 1.0000
Coef. de correlación Kendall: 0.9129
Coef. de correlación Spearman: 0.9487

Sumatoria de desviaciones al cuadrado: 0.000000
Error cuadrado medio: 0.000000

Lim. sup. Lim. inf.
9.346511 -10.160071
13.863029 -2.812181
20.011829 2.903425
23.696263 5.151195

Introduce el valor de x a estimar.
cuando termines de estimar oprime Ctrl C.

```
x = 1
ans = 1.000000000000001
x = 1.5
ans = 1.2500000000000000
```

```
x = 3
ans = 5.0000000000000001
x =
>>
```

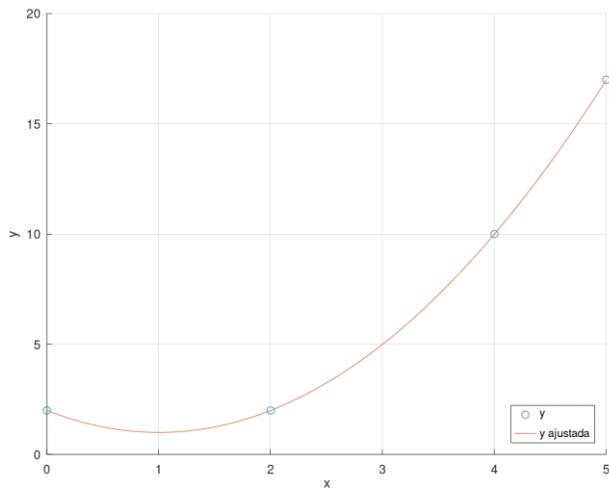


Figura 6.3. Resultados del Ejemplo 6.2.1 usando código de Octave.

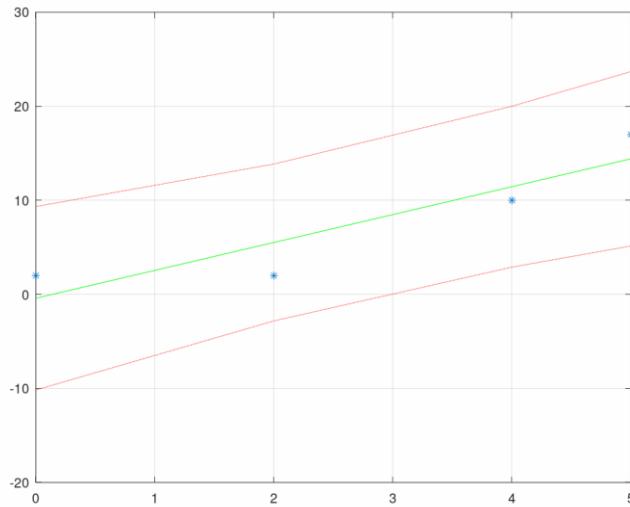


Figura 6.4. Límites de los intervalos de confianza del Ejemplo 6.2.1.

Ejemplo 6.2.2 Suponer la siguiente tabla de datos correspondiente al número de dobleces necesarios para quebrar una aleación forjada en relación al porcentaje de cada uno de los dos elementos aleados presentes en el material.

i	y No. de dobleces	x_1 % del elemento A	x_2 % del elemento B
1	38	1	5
2	40	2	5
3	85	3	5
4	59	4	5
5	40	1	10
6	60	2	10
7	68	3	10
8	53	4	10
9	31	1	15
10	35	2	15
11	42	3	15
12	59	4	15
13	18	1	20
14	34	2	20
15	29	3	20
16	42	4	20

Suponer que se desea ajustar una curva de la forma,

$$y' = c_0 + c_1 x_1 + c_2 x_2$$

Solución.

Entonces, según la ecuación (6.11),

$$(n)c_0 + \left(\sum x_{1_i}\right)c_1 + \left(\sum x_{2_i}^2\right)c_2 = \sum y_i$$

$$\left(\sum x_{1_i}\right)c_0 + \left(\sum x_{1_i}^2\right)c_1 + \left[\sum (x_{1_i} x_{2_i})\right]c_2 = \sum (x_{1_i} y_i)$$

$$\left(\sum x_{2_i}\right)c_0 + \left[\sum (x_{1_i} x_{2_i})\right]c_1 + \left[\sum (x_{2_i}^2)\right]c_2 = \sum (x_{2_i} y_i)$$

Los coeficientes del sistema planteado son:

$$n = 16, \quad \Sigma(x_1 * x_2) = 500, \quad \Sigma(x_1) = 40, \quad \Sigma(x_1 * x) = 1989, \quad \Sigma(x_2) = 200, \quad \Sigma(x_2 y) = 8285,$$

$$\Sigma(x_1^2) = 120, \quad \Sigma(y) = 733, \quad \Sigma(x_2^2) = 3000.$$

Entonces,

$$16c_0 + 40c_1 + 200c_2 = 733$$

$$40c_0 + 120c_1 + 500c_2 = 1989$$

...

$$200c_0 + 500c_1 + 3000c_2 = 8225$$

Resolviendo con Gauss-Jordan,

$$c_0 = 48.1875, c_1 = 7.8250, c_2 = -1.7550$$

Por lo tanto,

$$\dot{y} = 48.1875 + 7.8250x_1 - 1.7550x_2$$

➤ Usando comandos en Octave,

```
>> pkg load statistics
>> x1 = [1; 2; 3; 4; 1; 2; 3; 4; 1; 2; 3; 4; 1; 2; 3; 4];
>> x2 = [5; 5; 5; 10; 10; 10; 15; 15; 15; 15; 20; 20; 20; 20];
>> y = [38; 40; 85; 59; 40; 60; 68; 53; 31; 35; 42; 59; 18; 34; 29; 42];
>> X = [ones(size(x1)) x1 x2];
>> b = regress(y,X)
b =
    4.818750000000003e+01
    7.82499999999995e+00
   -1.755000000000001e+00

>>
```

NOTA: Aquí cabe aclarar que a la matriz X se le denomina “Matriz de diseño”, contiene una primera columna unitaria que representa el efecto común - comando, `ONES (x1)` - y el resto son columnas correspondientes a las variables independientes, en este caso `x1` y `x2`, representando el efecto de las variables predictoras.

```
>> x
x =
    1     1     5
    1     2     5
    1     3     5
    1     4     5
    1     1    10
    1     2    10
    1     3    10
    1     4    10
    1     1    15
    1     2    15
    1     3    15
```

```

1      4    15
1      1    20
1      2    20
1      3    20
1      4    20

```

```
>> y
y =
```

```

38
40
85
59
40
60
68
53
31
35
42
59
18
34
29
42

```

```
>>
```

- Usando código en Octave, *Ajuste_polinomial.m*, resulta lo siguiente:

Se crea previamente la hoja, para este caso, *Libro2.xlsx* y en la Hoja 1, rango A1:D18,

	A	B	C	D
1	Var comun	x1	x2	y
2	1	1	5	38
3	1	2	5	40
4	1	3	5	85
5	1	4	5	59
6	1	1	10	40
7	1	2	10	60
8	1	3	10	68
9	1	4	10	53
10	1	1	15	31
11	1	2	15	35
12	1	3	15	42
13	1	4	15	59
14	1	1	20	18
15	1	2	20	34
16	1	3	20	29
17	1	4	20	42
18				

Luego se modifican las líneas correspondientes en el código,

```

14 x=xlsread('Libro2.xlsx','Hoja1','A2:C17');
15 y=xlsread('Libro2.xlsx','Hoja1','E2:E17');

```

Finalmente se ejecuta,

Crear previamente los vectores x y y en excel.
si ya lo creaste oprime Enter si no, oprime Ctrl C.

Detected XLS interfaces:

```
48.187500
7.825000
-1.755000
>>
```

6.3 Ejercicios propuestos.

- a) Dados los siguientes datos, ajustar un polinomio por mínimos cuadrados, de tal manera que sea el que mejor se ajuste y con el mínimo grado.

x	1.4	2.5	3.1	3.9	5.0	7.1	9.5	11.9	14.1	15.0	16.5	17.2
$f(x)$	3.8	2.6	4.1	5.2	6.2	6.9	7.2	6.7	5.8	3.8	3.8	2.8

- b) Dados los siguientes datos,

x	1.2	2.8	4.3	5.4	6.8	7.9
$f(x)$	2.1	11.5	28.1	41.9	72.3	91.4

Usando el criterio de mínimos cuadrados, ajustar una función de la forma $g(x) = Ax^B$ (determinar A y B).

- c) Dados los siguientes datos,

x	0	0.5	1.25	2.00	2.70	3.00	3.50	3.90	4.75	5.25
$f(x)$	1.37	1.48	2.09	2.77	3.60	4.10	4.88	6.01	7.95	9.90

Ajustar una función de la forma Ae^{Bx} usando el criterio de mínimos cuadrados.

- d) Usar regresión lineal múltiple para ajustar los siguientes datos,

x_1	0	1	2	0	1	2
x_2	2	2	4	4	6	6
y	19	12	11	24	22	15

- e) Usar regresión lineal múltiple para ajustar los siguientes datos,

x_1	1	1	2	2	3	3	4	4
x_2	1	2	1	2	1	2	1	2

y	18.0	12.8	25.7	20.6	35.0	29.8	45.5	40.3
---	------	------	------	------	------	------	------	------

7 Diferenciación numérica

7.1 Introducción

En cálculo diferencial las operaciones para diferenciar una función varían según su grado de complejidad, además de que existen numerosas reglas que hay que respetar para diferentes formas de funciones. Con la aplicación de diferencias finitas, el proceso se simplifica al grado de ejecutar operaciones algebraicas sencillas para la diferenciación, resaltando que se deben tener muy claros conceptos básicos como tangente, pendiente y derivada.

7.2 Diferencias hacia adelante y hacia atrás.

En la figura 7.1 se muestra la gráfica de una función analítica, la cual se usará para desarrollar las diferencias finitas.

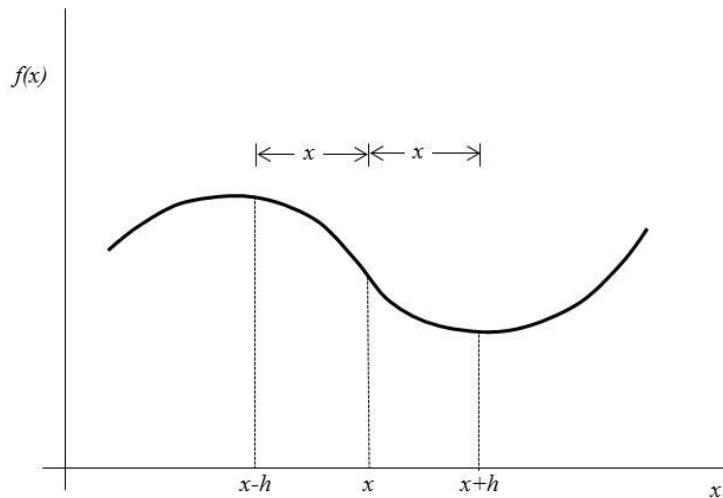


Figura 7.1 Diferencias hacia adelante y hacia atrás.

Para encontrar la primera diferencia hacia adelante se expande $f(x)$ alrededor de la vecindad de x por medio de serie de Taylor,

$$f(x+h) = f(x) + h f'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + \dots \quad (7.1)$$

Despejando $f'(x)$,

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(x) - \frac{h^2}{6} f'''(x) + \dots \quad (7.2)$$

Si se desprecian los términos de orden superior y se usa una nomenclatura más reducida,

$$f(x_j + h) = f_{j+1}$$

$$f(x_j) = f_j$$

$$f(x_j - h) = f_{j-1}$$

La ecuación (6.2) quedaría,

$$f'(x_j) = \frac{f(x_{j+1}) - f(x_j)}{h} + \dots \quad (7.3)$$

Definiendo la primera diferencia hacia delante de f en j como:

$$\Delta f_j \equiv f_{j+1} - f_j \quad (7.4)$$

La ecuación (6.3) queda,

$$f'(x_j) = \frac{\Delta f_j}{h} + \dots \quad (7.5)$$

El término $\Delta f_j/h$ se denomina “primer diferencia hacia adelante” de $f'(x_j)$. Gráficamente, la expresión $(f_{j+1} - f_j)/h$ aproxima la pendiente de la función f para el punto x , por la pendiente de la línea recta que pasa por los puntos en $f(x+h)$ y $f(x)$.

De manera similar se puede proceder para obtener la primera diferencia hacia atrás,

$$f(x-h) = f(x) - h f'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + \dots \quad (7.6)$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{h}{2} f''(x) - \frac{h^2}{6} f'''(x) + \dots \quad (7.7)$$

Si se define la primera diferencia hacia atrás como:

$$\nabla f_j \equiv f_j - f_{j-1} \quad (7.8)$$

Entonces, la ecuación (7.7) queda:

$$f'(x_j) = \frac{\nabla f_j}{h} + \dots \quad (7.9)$$

Y el término $\nabla f_j / h$ se denomina “primera diferencia hacia atrás” de $f'(x_j)$. De manera similar, es la pendiente de la línea recta que conecta los puntos $f(x)$ y $f(x-h)$.

Para aproximar derivadas de mayor orden, se consideran más términos de la serie de Taylor. Por ejemplo: expandiendo hacia adelante alrededor de x , para $f(x+h)$ se obtiene:

$$f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2!}f''(x) + \frac{(2h)^3}{3!}f'''(x) + \dots \quad (7.10)$$

Multiplicando la ecuación (7.1) por 2 y restándola de la ecuación (7.10),

$$\begin{aligned} f(x+2h) - 2f(x+h) &= -f(x) + 2h^2 f''(x) - h^2 f''(x) \\ &\quad + \frac{4h^2}{3} f'''(x) - \frac{h^3}{3} f'''(x) + \dots \end{aligned} \quad (7.11)$$

Despreciando términos de orden mayor y resolviendo para $f'''(x)$,

$$f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} + \dots$$

O también,

$$f''(x_j) = \frac{f_{j+2} - 2f_{j+1} + f_j}{h^2} + \dots \quad (7.12)$$

La ecuación (7.12) es la expresión para la 2da diferencia de f con respecto de x . La segunda diferencia hacia delante se define como:

$$\Delta^2 f_j \equiv f_{j+2} - 2f_{j+1} + f_j \quad (7.13)$$

Por lo que la ecuación (7.12) queda:

$$f''(x_j) = \frac{\Delta^2 f_j}{h^2} + \dots \quad (7.14)$$

De manera similar se procede para determinar una aproximación de la 2da diferencia de f , expandiendo la serie de Taylor hacia atrás para obtener $f(x-h)$ y $f(x-2h)$,

$$f''(x_j) = \frac{f_{j-2} - 2f_{j-1} + f_j}{h^2} + \dots \quad (7.15)$$

Y la 2da diferencia hacia atrás de f en j es:

$$\nabla^2 f_j \equiv f_j - 2f_{j-1} + f_{j-2} \quad (7.16)$$

Generalizando, cualquier diferencia hacia adelante o hacia atrás puede encontrarse con las siguientes fórmulas¹,

$$\begin{aligned} \Delta^n f_j &\equiv \Delta(\Delta^{n-1} f_j) \\ \nabla^n f_j &\equiv \nabla(\nabla^{n-1} f_j) \end{aligned} \quad (7.17)$$

Es decir, se puede encontrar cualquier diferencia tomando las diferencias de las diferencias menores anteriores. Por ejemplo, la 2da diferencia hacia atrás de f para j se puede determinar de la siguiente manera,

$$\nabla^2 f_j = \nabla(\nabla f_j) = \nabla f_j - \nabla f_{j-1} = (f_j - f_{j-1}) - (f_{j-1} - f_{j-2}) = f_j - 2f_{j-1} + f_{j-2}$$

Similarmente se encuentra la 3ra diferencia hacia atrás de f para j ,

$$\begin{aligned}\nabla^3 f_j &= \nabla(\nabla^2 f_j) = \nabla^2 f_j - \nabla^2 f_{j-1} \\ \nabla^3 f_j &= (f_j - 2f_{j-1} + f_{j-2}) - (f_{j-1} - 2f_{j-2} + f_{j-3}) \\ \nabla^3 f_j &= f_j - 3f_{j-1} + 3f_{j-2} - f_{j-3}\end{aligned}$$

Las diferencias hacia adelante y hacia atrás, para derivadas de cualquier orden, están dadas por¹:

$$\begin{aligned}\left. \frac{d^n f}{dx^n} \right|_{x_j} &= \frac{\Delta^n f_j}{h^n} \\ \left. \frac{d^n f}{dx^n} \right|_{x_j} &= \frac{\nabla^n f_j}{h^n}\end{aligned}\tag{7.18}$$

De las ecuaciones (7.18) se puede obtener las aproximaciones de la tabla 7.1 y 7.2.

Tabla 7.1. Diferencias hacia adelante con error de 1er orden

	f_j	f_{j+1}	f_{j+2}	f_{j+3}	f_{j+4}
$hf'(x_j) =$	-1	1			
$h^2 f''(x_j) =$	1	-2	1		
$h^3 f'''(x_j) =$	-1	3	-3	1	
$h^4 f^{IV}(x_j) =$	1	-4	6	-4	1

Tabla 7.2. Diferencias hacia atrás con error de 1er orden

	f_j	f_{j-1}	f_{j-2}	f_{j-3}	f_{j-4}
$hf'(x_j) =$	1	-1			
$h^2 f''(x_j) =$	1	-2	1		
$h^3 f'''(x_j) =$	1	-3	3	-1	
$h^4 f^{IV}(x_j) =$	1	-4	6	-4	1

Las expresiones para las diferencias finitas obtenidas hasta ahora, tienen errores por truncamiento de primer orden, es decir, se trunca la serie de Taylor justamente después del término deseado. Sin embargo, para obtener diferencias con mayor aproximación, se pueden considerar más términos de la serie de Taylor. Por ejemplo, para diferencias hacia adelante,

$$f(x+h) = f(x) + hf'(x) + \frac{h_2}{2!} f''(x) + \frac{h_3}{3!} f'''(x) + \dots \quad (7.19)$$

despejando $f'(x)$,

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(x) - \frac{h^2}{6} f'''(x) + \dots \quad (7.20)$$

Sustituyendo en la ecuación (7.20) la 2da diferencia hacia adelante y despreciando términos de orden mayor,

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} \left[\frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} \right] + \dots \quad (7.21)$$

reduciendo términos.

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \dots$$

usando subíndices,

$$f'(x_j) = \frac{-f_{j+2} + 4f_j + 1 - 3f_j}{2h} + \dots \quad (7.22)$$

La ecuación (7.22) es la aproximación de la diferencia hacia adelante para la primera derivada con un error de segundo orden.

De manera similar se obtienen las diferencias hacia atrás. Entonces, se puede decir que, como en estas últimas diferencias se toman en cuenta más términos de la serie de Taylor, el resultado se aproximará más al valor verdadero de la derivada.

Así, las aproximaciones de las diferencias hacia adelante y hacia atrás con un orden de error de 2do orden se muestran en las tablas 7.3 y 7.4

Tabla 7.3. Diferencias hacia adelante con error de 2do orden

	f_j	f_{j+1}	f_{j+2}	f_{j+3}	f_{j+4}	f_{j+5}
$2hf'(x_j) =$	-3	4	-1			
$h^2 f''(x_j) =$	2	-5	4	-1		
$2h^3 f'''(x_j) =$	-5	18	-24	14	-3	
$h^4 f^{IV}(x_j) =$	3	-14	26	-24	11	-2

Similarmente, las diferencias hacia atrás son:

Tabla 7.4. Diferencias hacia atrás con error de 2do orden

	f_j	f_{j-1}	f_{j-2}	f_{j-3}	f_{j-4}	f_{j-5}
$2hf'(x_j) =$	3	-4	1			
$h^2 f''(x_j) =$	2	-5	4	-1		
$2h^3 f'''(x_j) =$	5	-18	24	-14	3	
$h^4 f^{IV}(x_j) =$	3	-14	26	-24	11	-2

7.3 Diferencias centradas.

Expandiendo la serie de Taylor hacia adelante y hacia atrás, alrededor de x , se obtiene,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + \dots \quad (7.23)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + \dots \quad (7.24)$$

Restando la ecuación (7.24) de la ecuación (7.23),

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3} f'''(x) + \dots$$

Despejando $f'(x)$,

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(x) + \dots$$

O también, despreciando términos de la serie y usando subíndices,

$$f'(x_j) = \frac{f_{j+1} - f_{j-1}}{2h} \quad (7.25)$$

La diferencia expresada por la ecuación (7.25) se denomina diferencia centrada con error de 2do orden. Una expresión para $f''(x)$ con el mismo orden de error puede obtenerse sumando las ecuaciones (7.23) y (7.24),

$$f''(x_j) = \frac{f_{j+1} - 2f_j - f_{j-1}}{h^2} \quad (7.26)$$

Para obtener $f'''(x)$ y $f^{IV}(x)$ se requiere una expresión adicional de la serie de Taylor en cada dirección y algunas manipulaciones similares a las realizadas para obtener $f'(x)$ y $f''(x)$. Las diferencias centradas con error de 2do orden se muestran en la tabla 7.5.

Tabla 7.5. Diferencias centradas con error de 2do orden

	f_{j-2}	f_{j-1}	f_j	f_{j+1}	f_{j+2}
$2h f'(x_j) =$		-1	0	1	
$h^2 f''(x_j) =$		1	-2	1	
$2h^3 f'''(x_j) =$	-1	2	0	-2	1
$h^4 f^{IV}(x_j) =$	1	-4	6	-4	1

Como se puede observar en la tabla 7.5, las expresiones siguen un cierto comportamiento dado por las siguientes fórmulas:

$$\frac{df^n}{dx^n} = \frac{\nabla^n f_{j+n/2} + \Delta f_{j-n/2}}{2h^n}, \quad \text{para } n \text{ par.} \quad (7.27)$$

$$\frac{df^n}{dx^n} = \frac{\nabla^n f_{j+(n-1)/2} + \Delta f_{j-(n-1)/2}}{2h^n}, \quad \text{para } n \text{ impar.}$$

Finalmente, las diferencias centradas con error de 4to orden, cuyo procesamiento es más laborioso, se muestran en la tabla 7.6.

Tabla 7.6. Diferencias centradas con error de 4to orden.

	f_{j-3}	f_{j-2}	f_{j-1}	f_j	f_{j+1}	f_{j+2}	f_{j+3}
$2h f'(x_j) =$		1	-8	0	8	-1	
$h^2 f''(x_j) =$		-1	16	-30	16	-1	
$2h^3 f'''(x_j) =$	1	-8	13	0	-13	8	-1
$h^4 f^{IV}(x_j) =$	-1	12	-39	56	-39	12	-1

7.4. Diferenciación numérica con datos desigualmente espaciados.

En el caso de que tengan datos desigualmente espaciados, se puede aproximar la derivación con base a lo visto en interpolación de Lagrange. En la figura 7.2 se muestran tres puntos unidos por una curva de segundo grado para obtener el interpolante lagrangiano.

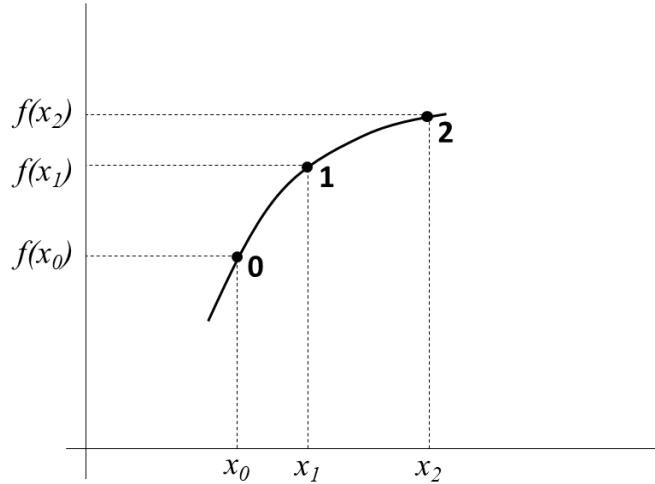


Figura 7.2. Curva para obtener el interpolante lagrangiano.

Entonces, resumiendo interpolación de Lagrange usando la figura 7.2,

$$y = a_0 + a_1x + a_2x^2 \quad \rightarrow \quad y = A_0(x - x_1)(x - x_2) + A_1(x - x_0)(x - x_2) + A_2(x - x_0)(x - x_1)$$

$$A_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)}$$

$$A_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2)}$$

$$A_2 = \frac{y_2}{(x_2 - x_0)(x_2 - x_1)}$$

$$\begin{aligned} y = f(x) &= \frac{y_0}{(x_0 - x_1)(x_0 - x_2)}(x - x_1)(x - x_2) + \\ &\quad \frac{y_1}{(x_1 - x_0)(x_1 - x_2)}(x - x_0)(x - x_2) + \\ &\quad \frac{y_2}{(x_2 - x_0)(x_2 - x_1)}(x - x_0)(x - x_1) \end{aligned}$$

Simplificando se obtiene,

$$\begin{aligned}y' = f'(x) &= y_0 \frac{(2x - x_1 - x_2)}{(x_0 - x_1)(x_0 - x_2)} + \\&\quad y_1 \frac{(2x - x_0 - x_2)}{(x_1 - x_0)(x_1 - x_2)} + \\&\quad y_2 \frac{(2x - x_0 - x_1)}{(x_2 - x_0)(x_2 - x_1)}\end{aligned}\tag{7.28}$$

7.5. Ejemplos.

Ejemplo 7.4.1 Evaluar $\frac{de^x}{dx}$ en $x=1$, usando todas las diferencias vistas y usando incrementos de $h=0.1$ y $h=0.01$.

La derivada de $f'(e^x) = e^x$. Entonces su valor verdadero es $f'(e^1) = 2.718282$

Con $h = 0.1$,

- a) Diferencia hacia adelante con error de 1er orden,

$$f'(x) = \frac{f_{j+1} - f_j}{h} = \frac{e^{1.1} - e^1}{0.1} = 2.858842$$

$$\varepsilon = \frac{|2.718282 - 2.858842|}{2.858842} \times 100 = 5.17\%$$

- b) Diferencia hacia atrás con error de 1er orden,

$$f'(x) = \frac{f_j - f_{j-1}}{h} = \frac{e^1 - e^{0.9}}{0.1} = 2.586787$$

$$\varepsilon = \frac{|2.718282 - 2.586787|}{2.858842} \times 100 = 4.84\%$$

- c) Diferencia hacia delante con error de 2do orden,

$$f'(x) = \frac{-3f_j + 4f_{j+1} - f_{j+2}}{2h} = \frac{-3e^1 + 4e^{1.1} - e^{1.2}}{2(0.1)} = 2.708508$$

$$\varepsilon = \frac{|2.718282 - 2.708505|}{2.858842} \times 100 = 0.36\%$$

- d) Diferencia hacia atrás con error de 2do orden,

$$f'(x) = \frac{3f_j - 4f_{j-1} + f_{j-2}}{2h} = \frac{3e^1 - 4e^{0.9} + e^{0.8}}{2(0.1)} = 2.709870$$

$$\varepsilon = \frac{|2.718282 - 2.709870|}{2.858842} \times 100 = 0.31\%$$

- e) Diferencia centrada con error de 2do orden,

$$f'(x) = \frac{-f_{j-1} + f_{j+1}}{2h} = \frac{-e^{0.9} + e^{1.1}}{2(0.1)} = 2.722815$$

$$\varepsilon = \frac{|2.718282 - 2.722815|}{2.858842} \times 100 = 0.17\%$$

- f) Diferencia centrada con error de 4to orden,

$$f'(x) = \frac{f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2}}{12h} = \frac{e^{0.8} - 8e^{0.9} + 8e^{1.1} - e^{1.2}}{12(0.1)} = 2.718273$$

$$\varepsilon = \frac{|2.718282 - 2.718273|}{2.858842} \times 100 = 0.00033\%$$

Ahora con $h = 0.01$,

- a) Diferencia hacia adelante con error de 1er orden,

$$f'(x) = \frac{f_{j+1} - f_j}{h} = \frac{e^{1.01} - e^1}{0.01} = 2.731919$$

$$\varepsilon = \frac{|2.718282 - 2.731919|}{2.858842} \times 100 = 0.50\%$$

- b) Diferencia hacia atrás con error de 1er orden,

$$f'(x) = \frac{f_j - f_{j-1}}{h} = \frac{e^1 - e^{0.99}}{0.01} = 2.704736$$

$$\varepsilon = \frac{|2.718282 - 2.704736|}{2.858842} \times 100 = 0.50\%$$

- c) Diferencia hacia delante con error de 2do orden,

$$f'(x) = \frac{-3f_j + 4f_{j+1} - f_{j+2}}{2h} = \frac{-3e^1 + 4e^{1.01} - e^{1.02}}{2(0.01)} = 2.718191$$

$$\varepsilon = \frac{|2.718282 - 2.718191|}{2.858842} \times 100 = 0.00335\%$$

- d) Diferencia hacia atrás con error de 2do orden,

$$f'(x) = \frac{3f_j - 4f_{j-1} + f_{j-2}}{2h} = \frac{3e^1 - 4e^{0.99} + e^{0.98}}{2(0.01)} = 2.718192$$

$$\varepsilon = \frac{|2.718282 - 2.718192|}{2.858842} \times 100 = 0.00331\%$$

- e) Diferencia centrada con error de 2do orden,

$$f'(x) = \frac{-f_{j-1} + f_{j+1}}{2h} = \frac{-e^{0.99} + e^{1.01}}{2(0.01)} = 2.718327$$

$$\varepsilon = \frac{|2.718282 - 2.722815|}{2.858842} \times 100 = 0.00166\%$$

- f) Diferencia centrada con error de 4to orden,

$$f'(x) = \frac{f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2}}{12h} = \frac{e^{0.98} - 8e^{0.99} + 8e^{1.01} - e^{1.02}}{12(0.01)} = 2.718282$$

$$\varepsilon = \frac{|2.718282 - 2.718282|}{2.858842} \times 100 = 0.00000\%$$

- Con comandos de Octave se obtiene lo siguiente (solo se calcula la primera derivada en $x=1$):

```
>> pkg load symbolic
>> syms x
>> fx=exp(x);
>> d1=diff(fx,1)
d1 = (sym)

x
e

>> subs(d1,1)
ans = (sym) E
>> d1=function_handle(d1)
d1 =

@ (x) exp (x)

>> d1(1)
ans = 2.7183
>>
```

- En la figura 7.2 se muestran los resultados obtenidos con código en Octave,
Difs_fins_Octave.m.

Figura 7.2. Resultados del ejemplo 7.6 obtenidos con código en Octave.

*** Diferencias Finitas ***.

```
Introduce una función con respecto a x: exp(x)
Incremento de x = 0.1
x = 1
```

```
¿Hasta qué diferencia quieres calcular (1-4)? 1
```

```
Valor analítico: 2.718282
```

```
** OPRIME Enter DESPÚES DEL DESPLIEGUE DE CADA VALOR **
```

* Aproximaciones hacia adelante de primer orden:

2.858842

* Aproximaciones hacia atrás de primer orden:

2.586787

* Aproximaciones hacia adelante de segundo orden:

2.708508

* Aproximaciones hacia atrás de segundo orden:

2.709870

* Aproximaciones centrada de segundo orden:

2.722815

* Aproximaciones centrada de cuarto orden:

2.718273

>>

También, con $h = 0.01$,

*** Diferencias Finitas ***.

Introduce una función con respecto a x: exp(x)

Incremento de x = 0.01

x = 1

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 2.718282

** OPRIME Enter DESPÚES DEL DESPLIEGUE DE CADA VALOR **

* Aproximaciones hacia adelante de primer orden:

2.731919

* Aproximaciones hacia atrás de primer orden:

2.704736

* Aproximaciones hacia adelante de segundo orden:

2.718191

* Aproximaciones hacia atrás de segundo orden:

2.718192

* Aproximaciones centrada de segundo orden:

2.718327

* Aproximaciones centrada de cuarto orden:

2.718282

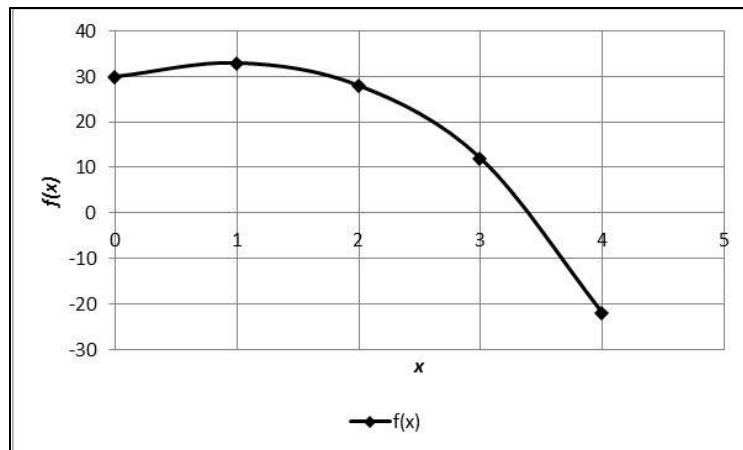
>>

Ejemplo 7.4.2 Dados los siguientes datos igualmente espaciados,

x	0	1	2	3	4
$f(x)$	30	33	28	12	-22

Encontrar $f'(0), f'(2), f'(4)$ y $f''(0)$.

Inicialmente se verifica el comportamiento de los puntos tabulados, de tal manera que se seleccione la diferencia finita apropiada.



En este ejemplo, los valores de las $f(x)$ ya están dados en la tabulación, por lo que solamente se sustituyen en las fórmulas de diferencias finitas.

- 1er diferencia hacia delante de 2do orden:

$$f'(0) = \frac{-3f(0) + 4f(1) - f(2)}{2(1)} = \frac{-3(30) + 4(33) - 28}{2} = 7.00$$

- 1er diferencia centrada de 4to orden:

$$f'(2) = \frac{f(0) - 8f(1) + 8f(3) - f(4)}{12(1)} = \frac{30 - 8(33) + 8(12) - (-22)}{12} = -9.67$$

- 1er diferencia hacia atrás de 2do orden:

$$f'(4) = \frac{3f(4) - 4f(3) + f(2)}{2(1)} = \frac{3(-22) - 4(12) + 28}{2} = -43.00$$

- 2da diferencia hacia delante de 2do orden:

$$f''(0) = \frac{2f(0) - 5f(1) + 4f(2) - f(3)}{1^2} = \frac{2(30) - 5(33) + 4(28) - 12}{1} = -5.00$$

Ejemplo 7.4.3 Para la siguiente tabla, estimar V para $t = 30$ min.

$t, \text{ min}$	0	10	20	30	45	60	75
$V, 10^6 \text{ barriles}$	0.4	0.7	0.77	0.88	1.05	1.17	1.35

Solución.

Con la ecuación (7.28), se obtiene lo siguiente:

$$y' = f'(x) = y_0 \frac{(2x - x_1 - x_2)}{(x_0 - x_1)(x_0 - x_2)} +$$

$$y_1 \frac{(2x - x_0 - x_2)}{(x_1 - x_0)(x_1 - x_2)} +$$

$$y_2 \frac{(2x - x_0 - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Considerando, $x_0 = 20, x_1 = 30, x_2 = 45, y_0 = 0.77, y_1 = 0.88$ y $y_2 = 1.05$, para $y = 30$,

$$y' = f'(x) = 0.77 \frac{(2 \cdot 30 - 30 - 45)}{(20 - 30)(20 - 45)} + \\ 0.88 \frac{(2 \cdot 30 - 20 - 45)}{(30 - 20)(30 - 45)} + \\ 1.05 \frac{(2 \cdot 30 - 20 - 30)}{(45 - 20)(45 - 30)} = 0.011133$$

Usando el código Octave, *Difs_fins_desigual.m*,

```
*** Diferencias Finitas desigualmente espaciadas ***.
```

```
x0: 20
```

```
f(x0): 0.77
```

```
x1: 30
```

```
f(x1): 0.88
```

```
x2: 45
```

```
f(x2): 1.05
```

```
Valor de x donde quieras evaluar la derivada: 30
```

```
0.011133
```

```
>>
```

7.6. Ejercicios propuestos.

- a) Encontrar la primera diferencia de la función $f(x) = \sin(x)$, para $x = 5\pi/8$ usando todas las diferencias finitas expuestas anteriormente. Usar $h = \pi/10$ y $h = \pi/20$. Compare los resultados con la solución real y obtenga conclusiones.
- b) Calcular la primera diferencia de la función $f(x) = [\sin(x)]^{e^x}$ en $x = 1$, usando la diferencia centrada de 4to orden, con $h = 0.1$. Obtenga el error comparando con el valor verdadero y obtenga conclusiones.

8 Integración numérica

8.1 Introducción.

Como es común en métodos numéricos, el principal propósito de la integración numérica es la evaluación de integrales de una forma relativamente sencilla, aprovechando el concepto de obtener el área bajo la curva de la función a resolver.

8.2 Regla trapezoidal.

Suponer que se desea evaluar la integral,

$$I = \int_a^b f(x) dx \quad (8.1)$$

Para ello se divide el intervalo $a \leq x \leq b$ en n subintervalos h , donde

$$h = \frac{b-a}{n} \quad (8.2)$$

Una gráfica posible de la función $f(x)$ se muestra en la figura 8.1.

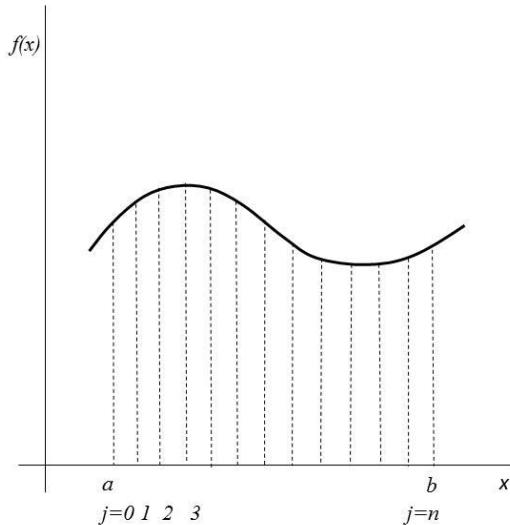


Figura 8.1. Regla trapezoidal.

Considerar una región compuesta por dos paneles de la figura 8.1, como se muestra en la figura 8.2. Como se aprecia, el valor de la integral, que es el área bajo la curva, se approxima dividiendo dicha área en trapecios, de aquí el nombre de regla trapezoidal, entonces,

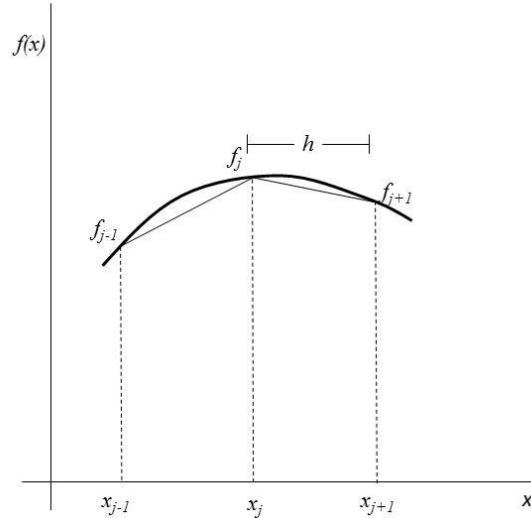


Figura 8.2. Descripción de la regla trapezoidal.

$$\int_{x_{j-1}}^{x_{j+1}} f(x) dx \approx \frac{f_{j-1} + f_j}{2} h + \frac{f_j + f_{j+1}}{2} h = \frac{h}{2} (f_{j-1} + 2f_j + f_{j+1}) \quad (8.3)$$

Extendiendo la ecuación (8.3) a todo el intervalo $a \leq x \leq b$ se obtiene,

$$\int_a^b f(x) dx \approx \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \dots + 2f_{n-1} + f_n) \quad (8.4)$$

Resumiendo, la ecuación (8.4) queda,

$$\int_a^b f(x) dx \approx \frac{h}{2} \left(f_0 + f_n + 2 \sum_{i=1}^{n-1} f_i \right) \quad (8.5)$$

donde $f_0 = f_a$ y $f_n = f_b$.

La ecuación (8.5) se denomina regla trapezoidal sin corrección final.

Para mejorar la precisión del resultado, se consideran las áreas restantes despreciadas por este método. Para ello se puede aproximar más considerando series de Taylor.

Suponiendo que $f(x)$ es una función analítica en la región de interés, entonces $I(x_{j+1})$ puede obtenerse expandiendo la serie de Taylor alrededor de x_j ,

$$I(x_j + h) = I(x_{j+1}) = I(x_j) + h I'(x_j) + \frac{h^2}{2!} I''(x_j) + \frac{h^3}{3!} I'''(x_j) + \dots \quad (8.6)$$

Sabiendo que:

$$I'(x_j) = f(x_j)$$

$$I''(x_j) = f'(x_j)$$

...

La ecuación (8.6) se puede expresar como:

$$I(x_{j+1}) = I(x_j) + h f(x_j) + \frac{h^2}{2!} f'(x_j) + \frac{h^3}{3!} f''(x_j) + \dots \quad (8.7)$$

Sustituyendo la primera derivada de la ecuación (8.7) por una diferencia hacia adelante de primer orden se obtiene:

$$I(x_{j+1}) = I(x_j) + h f(x_j) + \frac{h^2}{2!} \left[\frac{f(x_{j+1}) - f(x_j)}{h} - \frac{h}{2!} f''(x_j) + \dots \right] + \frac{h^3}{3!} f''(x_j) + \dots \quad (8.8)$$

Reduciendo términos,

$$I(x_{j+1}) = I(x_j) + \frac{h}{2} [f(x_{j+1}) + f(x_j)] - \frac{h^3}{12} f''(x_j) + \dots \quad (8.9)$$

Por inspección de la figura 8.2 se puede ver que la expresión $I(x_{j+1}) - I(x_j)$ constituye el área de un panel sencillo entre x_j y x_{j+1} . Es decir,

$$I(x_{j+1}) - I(x_j) = \frac{h}{2} [f(x_{j+1}) + f(x_j)] - \frac{h^3}{12} f''(x_j) + \dots \quad (8.10)$$

Entonces, el miembro izquierdo de la ecuación (8.10) representa el área de un trapecio.

Generalizando la ecuación (8.10) para n paneles, se puede obtener:

$$I = \frac{h}{2} \left[f_0 + f_n + 2 \sum_{j=1}^{n-1} f(x_j) \right] - \frac{h^3}{12} \sum_{j=0}^{n-1} f''(x_j) + \dots \quad (8.11)$$

Entonces, si el primer término del miembro derecho de la ecuación representa la aproximación del área bajo la curva por medio de trapecios, entonces el resto de los términos toman en cuenta el error cometido al considerar segmentos de recta para aproximar la función a través de trapecios.

Procesando el segundo término del miembro derecho se tiene que: aplicando el teorema del valor medio a la sumatoria,

$$\sum_{j=0}^{n-1} f''(x) = n f''(\bar{x}) , \quad \text{donde } a \leq \bar{x} \leq b$$

y

$$n f''(x) = \frac{b-a}{n} f''(x) \quad (8.12)$$

Usando diferencias simples, puede suponerse que,

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

Que sustituida en la ecuación (8.12) se obtiene,

$$\sum_{j=0}^{n-1} f''(x) \approx \frac{1}{h} [f'(b) - f'(a)] \quad (8.13)$$

Entonces, sustituyendo la ecuación (8.13) en la ecuación (8.11),

$$I = \frac{h}{2} \left[f_0 + f_n + 2 \sum_{j=1}^{n-1} f(x_j) \right] - \frac{h^2}{12} [f'(b) - f'(a)] \quad (8.14)$$

La ecuación (8.14) se denomina: regla trapezoidal con corrección final.

8.3 Regla de Simpson.

Este método usa arcos de parábolas para aproximar la $f(x)$, en lugar de segmentos de recta como el método anterior.

Considerar las siguientes series de Taylor expandidas alrededor de x_j :

$$\begin{aligned} I(x_{j+1}) &= I(x_j) + h f(x_j) + \frac{h^2}{2!} f'(x_j) + \frac{h^3}{3!} f''(x_j) + \frac{h^4}{4!} f'''(x_j) + \\ &\quad \frac{h^5}{5!} f^{IV}(x_j) + \frac{h^6}{6!} f^V(x_j) + \end{aligned} \quad (8.15)$$

$$I(x_{j-1}) = I(x_j) - h f(x_j) + \frac{h^2}{2!} f'(x_j) - \frac{h^3}{3!} f''(x_j) + \frac{h^4}{4!} f'''(x_j) - \dots \\ \frac{h^5}{5!} f^{IV}(x_j) + \frac{h^6}{6!} f^V(x_j) + \dots \quad (8.16)$$

Restando la ecuación (8.16) de la ecuación (8.15) se obtiene:

$$I(x_{j+1}) - I(x_{j-1}) = 2h f(x_j) + \frac{h^3}{3} f''(x_j) + \frac{h^5}{60} f^{IV}(x_j) + \dots \quad (8.17)$$

Ahora se sustituye la diferencia centrada de 2do orden de $f''(x_j)$ en la ecuación (8.17),

$$I(x_{j+1}) - I(x_{j-1}) = \frac{h}{3} [f(x_{j-1}) + 4f(x_j) + f(x_{j+1})] - \frac{h^5}{90} f^{IV}(x_j) + \dots \quad (8.18)$$

En la ecuación (8.18) se aprecia que el primer miembro, $I(x_{j+1}) - I(x_{j-1})$, representa el área de dos paneles entre x_{j+1} y x_{j-1} .

Esta ecuación puede generalizarse para n paneles,

$$I = \frac{h}{3} \left[\begin{array}{l} fa + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots \\ 2f(b-2h) + 4f(b-h) + f(b) \end{array} \right] - \frac{h^5}{90} \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-2} f^{IV}(x_j) + \dots$$

Sintetizando y despreciando el último término queda:

$$I = \frac{h}{3} \left(f_0 + f_n + 4 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + 2 \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right) \quad (8.19)$$

La ecuación (8.19) representa la regla de Simpson sin corrección final o regla de Simpson de 1/3.

Finalmente, con el fin de mejorar la exactitud de la regla de Simpson, se puede demostrar que (ver Apéndice C),

$$I = \frac{h}{15} \left\{ 14 \left[\frac{1}{2} (f_0 + f_n) + \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right] + 16 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + h [f'(a) - f'(b)] \right\} \quad (8.20)$$

Siendo la ecuación (8.20) la regla de Simpson con corrección final.

8.3.1 Simpson 3/8.

Se supone que se tienen tres paneles y se pretende ajustar a los cuatro puntos conocidos una ecuación de 3er grado.

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (8.21)$$

En la figura 8.3 se muestra la forma aproximada de la función que se pretende ajustar

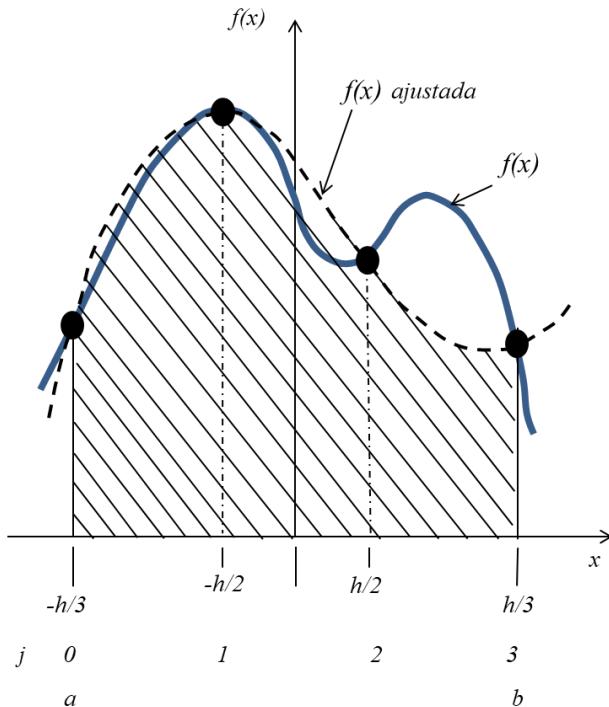


Figura 8.3

Entonces, se requiere determinar los valores de los coeficientes de la ecuación de 3er grado, a_0 , a_1 , a_2 y a_3 , de tal forma que pase por los cuatro puntos previamente conocidos. Para ello se ubica el eje de las ordenadas como se muestra en la figura 8.3. El intervalo de integración será:

$$-\frac{h}{3} \leq x \leq \frac{h}{3}$$

Para calcular el área de los tres paneles, se integra de la siguiente manera,

$$\begin{aligned} I &= \int_{-\frac{3h}{2}}^{3h/2} (a_0 + a_1x + a_2x^2 + a_3x^3)dx = \left[a_0x + a_1\frac{x^2}{2} + a_2\frac{x^3}{3} + a_3\frac{x^4}{4} \right]_{-3h/2}^{3h/2} \\ &= \frac{3a_0h}{2} + \frac{9a_1h^2}{8} + \frac{9a_2h^4}{24} + \frac{81a_3h^4}{64} + \frac{3a_0h}{2} - \frac{9a_1h^2}{8} + \frac{9a_2h^3}{24} - \frac{81a_3h^4}{64} = \end{aligned}$$

$$I = 3a_0h + \frac{9a_2h^3}{4} \quad (8.22)$$

Para calcular el valor de las constantes a_0, a_1, a_2 y a_3 , se procede de manera similar a la interpolación de Lagrange,

$$\begin{aligned} y_0 &= a_0 + a_1x_0 + a_2x_0 + a_3x_0 = a_0 + a_1\left(-\frac{3h}{2}\right) + a_2\left(-\frac{3h}{2}\right)^2 + a_3\left(-\frac{3h}{2}\right)^3 \\ y_1 &= a_0 + a_1x_1 + a_2x_1 + a_3x_1 = a_0 + a_1\left(-\frac{h}{2}\right) + a_2\left(-\frac{h}{2}\right)^2 + a_3\left(-\frac{3}{2}\right)^3 \\ y_2 &= a_0 + a_1x_2 + a_2x_2 + a_3x_2 = a_0 + a_1\left(\frac{h}{2}\right) + a_2\left(\frac{h}{2}\right)^2 + a_3\left(\frac{3}{2}\right)^3 \\ y_3 &= a_0 + a_1x_3 + a_2x_3 + a_3x_3 = a_0 + a_1\left(\frac{3h}{2}\right) + a_2\left(\frac{3h}{2}\right)^2 + a_3\left(\frac{3h}{2}\right)^3 \end{aligned}$$

El anterior es un sistema de cuatro ecuaciones, cuyas incógnitas son los coeficientes de la ecuación a ajustar. Entonces, presentando de manera matricial para resolverla,

$$\begin{bmatrix} 1 & -1.5h & 2.25h^2 & -3.375h^3 \\ 1 & -0.5h & 0.25h^2 & -0.125h^3 \\ 1 & 0.5h & 0.25h^2 & 0.125h^3 \\ 1 & 1.5h & 2.25h^2 & 3.375h^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Para obtener los valores de los coeficientes, se aplica la propiedad de la matriz inversa,

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

donde \mathbf{x} sería la matriz de los coeficientes, \mathbf{A} la matriz de los valores de h y \mathbf{b} la matriz de los valores de y .

Resolviendo con Octave se obtiene lo siguiente:

```
>> pkg load symbolic
>> syms x0 x1 x2 x3 y0 y1 y2 y3 a0 a1 a2 a3 h
>> A=[1 -1.5*h 2.25*h^2 -3.375*h^3; 1 -0.5*h 0.25*h^2 -0.125*h^3; 1 0.5*h 0.25*h^2
0.125*h^3; 1
1.5*h 2.25*h^2 3.375*h^3]
A = (sym 4x4 matrix)
[ 1           2           3 ]
[ -3*h      9*h      -27*h ]
[ 1   -----  -----  ----- ]
[ 2       4       8 ]
[          ]
[ 2           3 ]
[ -h       h       -h ]
[ 1   ---  --  ----- ]
[ 2       4       8 ]
[          ]
[ 2           3 ]
[ h       h       h ]
[ 1   -  --  --  -- ]
[ 2       4       8 ]
```

```

[           2           3 ]
[   3*h   9*h   27*h ]
[1  ---  -----  ----- ]
[   2       4       8   ]
>> x=[a0; a1; a2; a3]
x = (sym 4x1 matrix)
[a0]
[ ]
[a1]
[ ]
[a2]
[ ]
[a3]
>> b=[y0;y1;y2;y3]
b = (sym 4x1 matrix)
[y0]
[ ]
[y1]
[ ]
[y2]
[ ]
[y3]
>> xx=A\b
xx = (sym 4x1 matrix)
[   y0   9*y1   9*y2   y3]
[- -- + ---- + ---- - ---]
[   16     16     16     16]
[   ]
[y0 - 27*y1 + 27*y2 - y3]
[-----]
[      24*h]
[   ]
[   y0 - y1 - y2 + y3 ]
[-----]
[      2]
[   4*h]
[   ]
[-y0 + 3*y1 - 3*y2 + y3]
[-----]
[      3]
[   6*h]
>>

```

Entonces,

$$a_0 = \frac{y_0 + 9y_1 + 9y_2 + y_3}{16}$$

$$a_1 = \frac{y_0 - 27y_1 + 27y_2 - y_3}{24h}$$

$$a_3 = \frac{y_0 - y_1 - y_2 + y_3}{4h^2}$$

$$a_4 = \frac{-y_0 + 3y_1 - 3y_2 + y_3}{6h^3}$$

Sustituyendo a_0 y a_2 en la ecuación (8.22) y procesando algebráicamente se obtiene lo siguiente,

$$I = \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3) \quad (8.23)$$

La ecuación (8.23) es la fórmula de la regla de Simpson de 3/8. También, de la figura 8.3 se observa lo siguiente,

$$h = \frac{b - a}{3}$$

por lo que $b - a = 3h$ y la ecuación 8.23 queda,

$$I = \frac{(b-a)}{8}(y_0 + 3y_1 + 3y_2 + y_3) \quad (8.24)$$

la cual es otra forma de la misma ecuación (D.3)

8.4 Ejemplos.

Ejemplo 8.4.1 Evaluar numéricamente con 6 paneles aplicando las reglas trapezoidal y de Simpson sin y con corrección (por ser un número par de paneles).

$$I = \int_0^{\pi/2} \sin(2\cos(x)) \sin(x)^2 dx = \mathbf{0.507967}$$

(a) Regla trapezoidal sin corrección final.

$$I = \frac{h}{2} \left(f_0 + f_n + 2 \sum_{j=1}^{n-1} f_j \right)$$

Si $f(x) = \sin(2\cos(x)) \sin(x)^2$ se puede obtener la siguiente tabla para los seis paneles,
 $h = (b - a)/n = \pi/12$:

j	x_j	f_j
0	0	0.000000
1	$\pi/12$	0.062668
2	$\pi/6$	0.246757
3	$\pi/4$	0.493883
4	$\pi/3$	0.631103
5	$5\pi/12$	0.461682
6	$\pi/2$	0.000000

Entonces,

$$I = \frac{\pi}{24} \left(f_0 + f_6 + 2 \sum_{j=1}^5 f_j \right) = \frac{\pi}{24} [0 + 0 + 2(1.896093)] = 0.496396$$

$$\text{Error verdadero} = |0.507967 - 0.496396| = 0.011571$$

(b) Regla trapezoidal con corrección final.

$$I = \frac{h}{2} \left(f_0 + f_n + 2 \sum_{j=1}^{n-1} f_j \right) - \frac{h^2}{12} [f'(b) - f'(a)]$$

$$f'(x) = 2\sin(2\cos(x))\sin(x)\cos(x) - 2\cos(2\cos(x))\sin(x)^3$$

$$f'(a) = f'(0) = 0$$

$$f'(b) = f'(\pi/2) = -2$$

Sustituyendo,

$$I = \frac{\pi}{24} [0 + 0 + 2(1.896093)] - \frac{(\pi/12)^2}{12} (-2 - 0) = 0.507819$$

$$\text{Error verdadero} = |0.507967 - 0.507819| = 0.000148$$

(c) Regla de Simpson sin corrección final.

$$I = \frac{h}{3} \left(f_0 + f_n + 4 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + 2 \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right)$$

$$\sum_{\substack{j=1 \\ j \text{ impar}}}^5 f_j = 1.018233$$

$$\sum_{\substack{j=2 \\ j \text{ par}}}^4 f_j = 0.877860$$

$$I = \frac{(\pi/2)/6}{3} [0 + 0 + 4(0.018233)] + 2(0.877860) = 0.508646$$

$$\text{Error verdadero} = |0.507967 - 0.508646| = 0.000679$$

(d) Regla de Simpson con corrección final.

$$I = \frac{h}{15} \left\{ 14 \left[\frac{1}{2} (f_0 + f_n) + \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right] + 16 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + h [f'(a) - f'(b)] \right\}$$

$$I = \frac{(\pi/2)/6}{15} \left\{ 14 \left[\frac{1}{2}(0+0) + (0.877860) \right] + 16(1.018233 + \frac{(\pi/2)}{6}[0 - (-2)]) \right\} = 0.507984$$

$$\text{Error verdadero} = |0.507967 - 0.507984| = 0.000017$$

Como se observa, la regla de Simpson con corrección final es la más exacta.

Vale la pena mencionar que si se tiene dificultad para derivar la función a integrar, se puede usar diferencias finitas. Por ejemplo, considerando la función de este ejemplo y aplicando la diferencia central de cuarto orden con $h=0.1$:

$$f(x) = \sin(2\cos(x))\sin(x)^2$$

$$f'(0) = \frac{f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2}}{12h} = \frac{f(0 - 0.2) - 8f(0 - 0.1) + 8f(0 + 0.1) - f(0 + 0.2)}{12(0.1)} =$$

$$\frac{0.036516 - 8(0.009104) + 8(0.009104) - (0.036516)}{12(0.1)} = \frac{0}{1.2} = 0$$

$$f'(\pi) = \frac{f_{j-2} - 8f_{j-1} + 8f_{j+1} - f_{j+2}}{12h} = \frac{f(\pi - 0.2) - 8f(\pi - 0.1) + 8f(\pi + 0.1) - f(\pi + 0.2)}{12(0.1)} =$$

$$\frac{-0.06516 - 8(-0.009104) + 8(-0.009104) - (-0.036516)}{12(0.1)} = \frac{0}{1.2} = 0$$

➤ Usando comandos de Octave para integrales definidas por una función,

```
>> f=@(x) sin(2.*cos(x)).*sin(x).^2;
>> Integral = quad(f,0,pi/2);
>> fprintf('%10.6f \n',Integral)
0.507967
>>
```

➤ Ahora, elaborando un programa en código de Octave, *Trapezoidal_Octave.m*, y resolviendo, se obtiene lo mostrado en la figura 8.5.

Figura 8.5. Resultados obtenidos con el código en Octave, ejemplo 8.4.1.

*** Regla trapezoidal ***.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Introduce una función con respecto a x $\sin(2\cos(x)) * \sin(x)^2$

Límite inferior = 0

Límite superior = $\pi/2$

Número de paneles = 6

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.000000
1	0.261799	0.062668
2	0.523599	0.246757
3	0.785398	0.493883
4	1.047198	0.631103
5	1.308997	0.461682
6	1.570796	0.000000

I = 0.496396

*** Regla trapezoidal con corrección ***.

$f'(li) = 0.000000$
 $f'(ls) = -2.000000$

I = 0.507819
>>

*** Regla de Simpson ***.

Introduce una función con respecto a x $\sin(2\cos(x)) * \sin(x)^2$
Límite inferior = 0
Límite superior = $\pi/2$

Número de paneles = 6

*** Regla de Simpson sin corrección ***.

0	0.000000	0.000000
1	0.261799	0.062668
2	0.523599	0.246757
3	0.785398	0.493883
4	1.047198	0.631103
5	1.308997	0.461682
6	1.570796	0.000000

suma par: 0.877860 suma impar: 1.018233

I = 0.508646

*** Regla de Simpson con corrección ***.

$f'(a) = 0.000000$ $f'(b) = -2.000000$

```
I = 0.507984
>>
```

Ejemplo 8.4.2 Integrar numéricamente la siguiente función, usando apropiadamente las técnicas numéricas explicadas anteriormente.

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

desde $a=0$ a $b=0.8$. Usar 9 paneles.

Resolviendo analíticamente:

$$\begin{aligned} & \int_0^{0.8} (0.2 + 0.25x - 200x^2 + 675x^3 - 900x^4 + 400x^5) dx \\ &= \left| 0.2x + 12.5x^2 - \frac{200}{3}x^3 + \frac{675}{4}x^4 - 180x^5 + \frac{400}{6}x^6 \right|_0^{0.8} = 1.640533 \end{aligned}$$

Como se observa, se especifica que se deben usar 9 paneles, por lo que solamente se pueden usar la regla trapezoidal sin y con corrección y la regla de Simpson 3/8. No así la regla de Simpson sin y con corrección porque se requiere que el número de paneles sea par.

Inicialmente se puede evaluar la función a integrar, $f(x)=0.2+25x-200x^2+675x^3-900x^4+400x^5$, en todos los valores de x , desde $j=0$ hasta $j=9$. Lo cual se considera hace más eficiente el cálculo.

j	x_j	f_j
0	0.000000	0.200000
1	0.088889	1.262083
2	0.177778	1.288095
3	0.266667	1.432724
4	0.355556	2.034909
5	0.444444	2.884211
6	0.533333	3.487177
7	0.622222	3.333705
8	0.711111	2.163412
9	0.800000	0.232000

(a) Regla trapezoidal sin corrección final.

$$I = \frac{h}{2} \left(f_0 + f_n + 2 \sum_{j=1}^8 f_j \right) = \frac{(0.8/9)}{2} [0.200000 + 0.232000 + 2(17.886317)] = 1.609095$$

(b) Regla trapezoidal con corrección final.

$$I = \frac{h}{2} \left(f_0 + f_n + 2 \sum_{j=1}^8 f_j \right) - \frac{h^2}{12} [f'(b) - f'(a)]$$

$$f'(x) = 25 - 400x + 2025x^2 - 3600x^3 + 2000x^4$$

$$f'(a) = f'(0) = 25$$

$$f'(b) = f'(0.8) = -23$$

$$I = 1.609095 - \frac{(0.8/9)^2}{12} (-23 - 25) = \mathbf{1.650700}$$

(c) Regla de Simpson 3/8.

➤ Regla de Simpson sin corrección final, usando los primeros 6 paneles para que n sea par.

$$I = \frac{h}{3} \left(f_0 + f_n + 4 \sum_{\substack{j=i \\ j \text{ impar}}}^5 f_j + 2 \sum_{\substack{j=2 \\ j \text{ par}}}^4 f_j \right)$$

j	x_j	f_j
0	0.000000	0.200000
1	0.088889	1.262083
2	0.177778	1.288095
3	0.266667	1.432724
4	0.355556	2.034909
5	0.444444	2.884211
6	0.533333	3.487177

$$I = \frac{(0.533333/6)}{3} [0.000000 + 0.533333 + 4(3.323005) + 2(5.579018)] = 0.967385$$

➤ Regla de Simpson 3/8 para los tres últimos paneles.

$$I = (b-a) \left[\frac{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)}{8} \right]$$

j	x_j	f_j
0	0.533333	3.487177
1	0.622222	3.333705
2	0.711111	2.163412
3	0.800000	0.232000

$$I = (0.800000 - 0.533333) \left[\frac{3.487177 + 3(3.333705) + 3(2.163412) + 0.232000}{8} \right] = 0.673684$$

Entonces, el resultado se obtiene sumando las áreas de ambas integrales:

$$I = 0.967386 + 0.673684 = \mathbf{1.641070}$$

➤ Resolviendo con comandos de Octave,

```
>> f=@(x) 0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
>> Integral = quad(f,0,0.8);
>> fprintf('%10.6f \n',Integral)
1.640533
>>
```

➤ Resolviendo con código de Octave, *Simpson_Octave.m*, se obtiene lo mostrado en la figura 8.6

Figura 8.6. Resultados obtenidos en código de Octave para el ejemplo 8.4.2.

*** Regla trapezoidal ***.

```
Introduce una función con respecto a x 0.2+25*x-200*x^2+675*x^3-900*x^4+400*x^5
Límite inferior = 0
Límite superior = 0.8
```

Número de paneles = 9

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.200000
1	0.088889	1.262083
2	0.177778	1.288095
3	0.266667	1.432724
4	0.355556	2.034909
5	0.444444	2.884211
6	0.533333	3.487177
7	0.622222	3.333705
8	0.711111	2.163412

```
9      0.800000      0.232000  
  
I = 1.609095  
  
*** Regla trapezoidal con corrección ***.  
  
f'(li)= 25.000000  
f'(ls)= -23.000000  
  
I = 1.640700  
>>
```

*** Regla de Simpson ***.

```
Introduce una función con respecto a x 0.2+25*x-200*x^2+675*x^3-900*x^4+400*x^5  
Límite inferior = 0  
Límite superior = 0.8  
  
Número de paneles = 9
```

*** Regla de Simpson 3/8 ***.

0	0.000000	0.200000
1	0.088889	1.262083
2	0.177778	1.288095
3	0.266667	1.432724
4	0.355556	2.034909
5	0.444444	2.884211
6	0.533333	3.487177
6	0.533333	3.487177
7	0.622222	3.333705
8	0.711111	2.163412
9	0.800000	0.232000

```
I = 1.641070  
>>
```

8.5. Ejercicios propuestos.

- a) Aplicar las reglas trapezial y de Simpson, con y sin corrección, comparando con la solución real y concluyendo. Usar 4 paneles.

$$\int_0^{\pi/2} \frac{\sin(x)}{\sqrt{1-0.25\sin^2(x)}} dx$$

- b) Aplicar las reglas trapezial y de Simpson, con y sin corrección, comparando con la solución real y concluyendo. Usar 10 paneles.

$$\int_{-1.1\pi}^{1.1\pi} \cos(x)^2 dx$$

- c) Dados los siguientes datos igualmente espaciados:

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11
<i>x_i</i>	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
<i>f(x_i)</i>	93	87	68	55	42	37	35	39	48	53	51	39

Determinar $\int_0^{1.1} f(x) dx$

9 Ecuaciones diferenciales ordinarias de primer orden.

9.1 Introducción.

Muchos problemas prácticos en ciencias e ingeniería requieren la solución de un sistema de ecuaciones diferenciales ordinarias simultáneas más que solamente una ecuación. Tales sistemas pueden ser representados de manera general:

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, \dots, y_n)$$

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2, \dots, y_n)$$

...

$$\frac{dy_n}{dt} = f_n(t, y_1, y_2, \dots, y_n)$$

En cuanto a la forma general del problema con valor inicial, queda de la siguiente forma:

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, \dots, y_n)$$

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2, \dots, y_n)$$

...

$$\frac{dy_n}{dt} = f_n(t, y_1, y_2, \dots, y_n)$$

$$y_1(0) = y_{1,0}$$

$$y_2(0) = y_{2,0}$$

...

$$y_n(0) = y_{n,0}$$

9.2. Método de Euler.

En la figura 9.1 se aproxima la derivada de la función $f(x,y)$ a través de la tangente, o también la pendiente, de la recta que pasa por los puntos x_i, y_i . Resulta obvio que cuando h tiende a cero, la tangente representa a la derivada en el punto x_i, y_i .

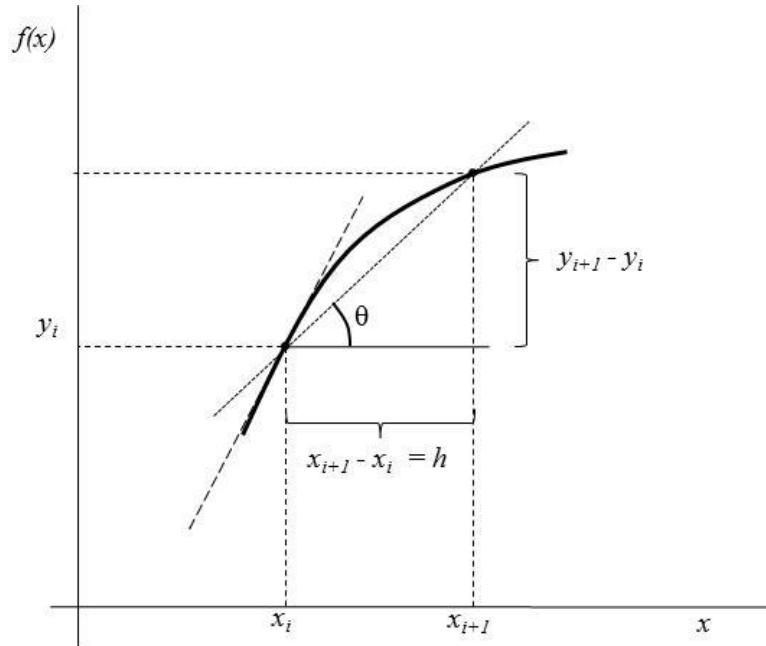


Figura 9.1. Método de Euler.

Entonces, sea la ecuación diferencial ordinaria de primer orden,

$$y' = \frac{dy}{dx} = f(x, y) \quad (9.1)$$

De la figura 9.1 se obtiene la tangente,

$$\operatorname{tg}(\theta) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = y' = f(x_i, y_i)$$

Procesando la ecuación anterior se obtiene:

$$y_{i+1} = y_i + h f(x_i, y_i) \quad (9.2)$$

La ecuación (9.2) se denomina ecuación de Euler, con la cual se puede resolver una ecuación diferencial ordinaria de primer orden de una manera iterativa.

9.3. Método de Runge-Kutta de 4to orden.

Las ecuaciones usadas para aplicar el método de Runge-Kutta de 4to orden, rk4, son las siguientes:

$$y_{i+1} = y_i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (9.3)$$

donde

$$k_1 = f(x_i, y_i) \quad (9.4)$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \quad (9.5)$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \quad (9.6)$$

$$k_4 = f(x_i + h, y_i + h k_3) \quad (9.7)$$

9.4. Sistemas de ecuaciones diferenciales ordinarias de primer orden.

Considerar la solución del siguiente sistema de n ecuaciones diferenciales ordinarias de primer orden,

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(x, y_1, y_2, y_3, \dots, y_n) \\ \frac{dy_2}{dx} &= f_2(x, y_1, y_2, y_3, \dots, y_n) \\ &\dots \\ \frac{dy_n}{dx} &= f_n(x, y_1, y_2, y_3, \dots, y_n) \end{aligned} \quad (9.8)$$

Con condiciones iniciales dadas para un punto común x_0 , es decir,

$$\begin{aligned} y_1(0) &= y_{1,0} \\ y_2(0) &= y_{2,0} \\ &\dots \\ y_n(0) &= y_{n,0} \end{aligned} \quad (9.9)$$

En el apartado de Ejemplos, se resuelve un problema al respecto.

9.5 Ejemplos.

Ejemplo 9.5.1 Resolver usando los métodos de Euler y de Runge Kutta de 4to orden.

$$\frac{dy}{dx} = \frac{1}{2}(1+x)y^2, \quad y(0) = 1$$

Evaluar en $x=0.5$. Con $h=0.1$ y 0.01 .

➤ Usando Octave para determinar la solución analítica se obtiene:

```
>> pkg load symbolic
>> warning('off','all')
>> syms y(x)
>> DE=diff(y,x)-0.5*(1+x)*y.^2==0
DE = (sym)
```

$$\frac{dy}{dx} - \frac{1}{2} - \frac{0.5(1+x)y^2}{2} = 0$$

```
>> sol=dsolve(DE,y(0)==1)
sol = (sym)
```

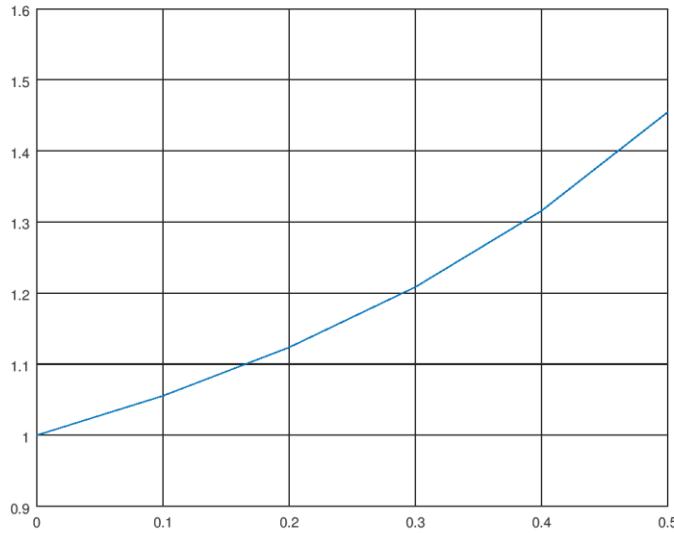
$$y(x) = \frac{-4}{x^2 + 2x - 4}$$

>>

Eso decir, $f(x) = \frac{4}{4-2x-x^2}$

➤ También se pueden usar comandos de Octave para obtener la solución numérica,

```
>> x=[0:0.1:0.5];
>> fun=@(y,x) 0.5*(1+x)*y.^2;
>> F=lsode(fun,1,x);
>> plot(x,F)
>> for i=1:6
fprintf('%4.2f %8.6f',x(i),F(i));
endfor
0.00 1.000000
0.10 1.055409
0.20 1.123596
0.30 1.208459
0.40 1.315790
0.50 1.454546
>>
```



➤ Método de Euler.

Aplicando el procedimiento por medio de la ecuación recursiva de Euler (9.2), se obtiene lo siguiente:

$$y_{i+1} = y_i + h f(x_i, y_i) = y_i + h \left[\frac{1}{2} (1 + x_i) y_i^2 \right]$$

Considerando $h = 0.1$,

$$i = 0, x_0 = 0, y_0 = 1$$

$$y_1 = y_0 + h f(x_0, y_0) = 1 + 0.1 \left[0.5 \bullet (1 + 0)(1)^2 \right] = 1.050000$$

$$i = 1, x_1 = 0.1, y_0 = 1.050000$$

$$y_2 = y_1 + h f(x_1, y_1) = 1.05 + 0.1 \left[0.5 \bullet (1 + 0.1)(1.05)^2 \right] = 1.110638$$

$$i = 2, x_2 = 0.2, y_0 = 1.110638$$

$$y_3 = y_2 + h f(x_2, y_2) = 1.110638 + 0.1 \left[0.5 \bullet (1 + 0.2)(1.110638)^2 \right] = 1.184648$$

...

En la tabla 9.1 y figura 9.2 se muestran los resultados.

<i>i</i>	<i>x_i</i>	<i>y_i</i>	<i>f(x_i, y_i)</i>	Sol. Analítica	ϵ
0	0.000000	1.000000	0.050000	1.000000	0.00%
1	0.100000	1.050000	0.060638	1.055409	0.51%
2	0.200000	1.110638	0.074011	1.123596	1.15%
3	0.300000	1.184648	0.091220	1.208459	1.97%
4	0.400000	1.275869	0.113949	1.315789	3.03%
5	0.500000	1.389818		1.454545	4.45%

Tabla 9.1. Resultados del ejemplo 9.5.1 usando Euler, $h=0.1$.

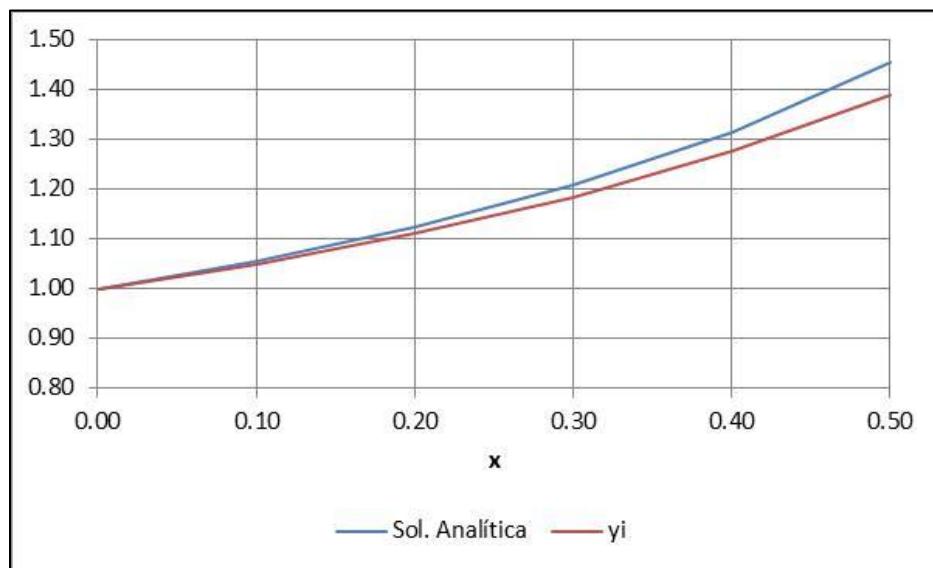


Figura 9.2. Grafica de los resultados del ejemplo 9.5.1 usando Euler, $h=0.01$.

Como se aprecia en la tabla 9.1, el error relativo porcentual para $x = 0.5$, con $h=0.1$, es de 4.45%.

$$\epsilon = \frac{\text{valor verdadero} - \text{valor estimado}}{\text{valor verdadero}} \times 100$$

Por otro lado, si se utiliza $h = 0.01$, se obtienen los siguientes resultados:

$$i = 0, x_0 = 0, y_0 = 1$$

$$y_1 = y_0 + h f(x_0, y_0) = 1 + 0.01 [0.5 \bullet (1+0)(1)^2] = 1.005000$$

$$i = 1, x_1 = 0.01, y_1 = 1.005000$$

$$y_2 = y_1 + h f(x_1, y_1) = 1.005 + 0.01 [0.5 \bullet (1 + 0.01)(1.005)^2] = 1.010101$$

$$i = 2, x_2 = 0.02, y_2 = 1.010101$$

$$y_3 = y_2 + h f(x_2, y_2) = 1.010101 + 0.01 [0.5 \bullet (1 + 0.02)(1.010101)^2] = 1.015304$$

...

En la tabla 9.2 y figura 9.3 se muestran los resultados utilizando $h = 0.01$. Como se aprecia, el error disminuye sustancialmente al disminuir h ; desde 4.45% hasta 0.52%.

i	x_i	y_i	$f(x_i, y_i)$	Sol. Analítica	ε
0	0.000000	1.000000	0.005000	1.000000	0.00%
1	0.010000	1.005000	0.005101	1.005050	0.01%
2	0.020000	1.010101	0.005204	1.010203	0.01%
3	0.030000	1.015304	0.005309	1.015460	0.02%
4	0.040000	1.020613	0.005417	1.020825	0.02%
5	0.050000	1.026030	0.005527	1.026299	0.03%
...					
45	0.450000	1.374620	0.013699	1.380500	0.43%
46	0.460000	1.388319	0.014070	1.394506	0.44%
47	0.470000	1.402389	0.014455	1.408897	0.46%
48	0.480000	1.416845	0.014855	1.423690	0.48%
49	0.490000	1.431700	0.015271	1.438901	0.50%
50	0.500000	1.446970		1.454545	0.52%

Tabla 9.2. Resultados del ejemplo 9.5.1 usando Euler, $h=0.01$.

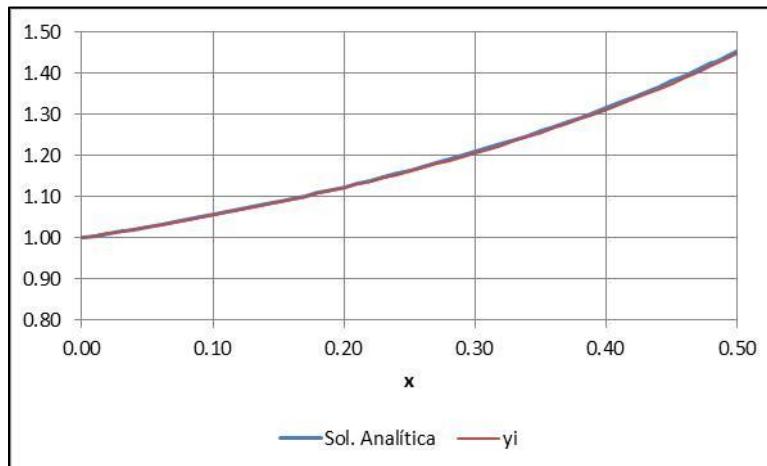


Figura 9.3. Grafica de los resultados del ejemplo 9.5.1 usando Euler, $h=0.01$.

- Resolviendo con código de Octave se obtiene lo mostrado en la figura 9.4.

Figura 9.4. Resultados del ejemplo 9.5.1, obtenidos con código en Octave, *Euler_1_Octave.m*.

Método de Euler.

```
Introduce una función con respecto a x  0.5*(1+x)*y.^2
```

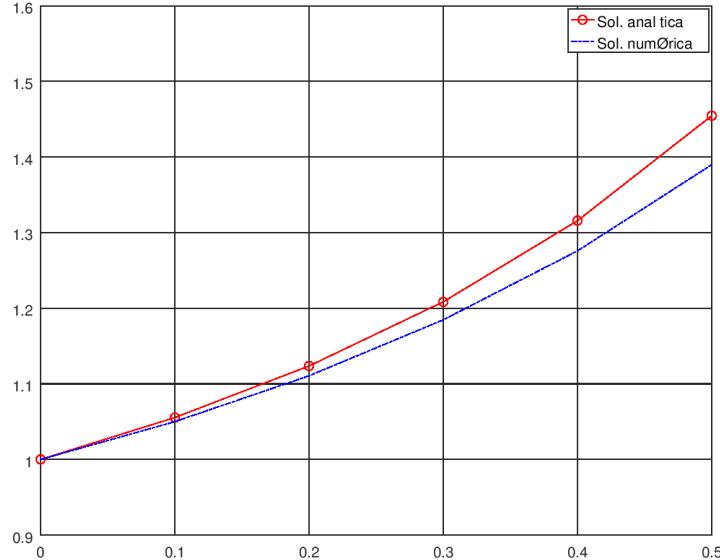
```
Introduce la solución analítica (Oprime Enter si no existe)  4./(4-2.*x-x.^2)
```

Datos:

```
x0 = 0
xn = 0.5
y0 = 1
Incremento de x = 0.1
```

i	x(i)	y(i)	y(i).verd.	e
0	0.000000	1.000000	1.000000	0.00
1	0.100000	1.050000	1.055409	0.51
2	0.200000	1.110638	1.123596	1.15
3	0.300000	1.184648	1.208459	1.97
4	0.400000	1.275869	1.315789	3.03
5	0.500000	1.389818	1.454545	4.45

>>



Ahora, con $h=0.01$,

Método de Euler.

```
Introduce una función con respecto a x  0.5*(1+x)*y.^2
```

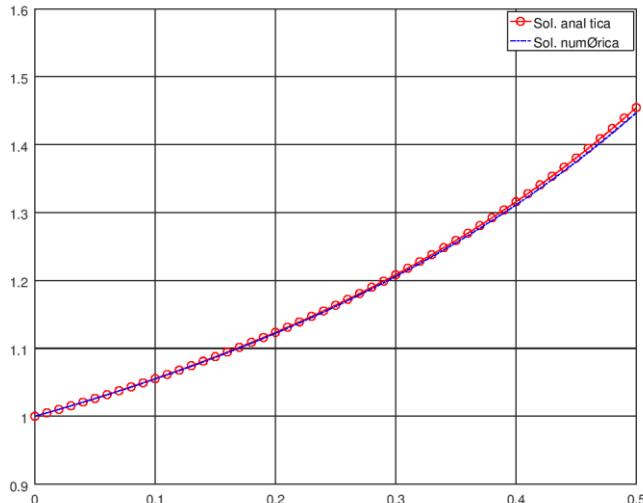
Introduce la solución analítica (Oprime Enter si no existe) $4 ./ (4 - 2.*x - x.^2)$

Datos:

```
x0 = 0
xn = 0.5
y0 = 1
Incremento de x = 0.01
```

i	x(i)	y(i)	y(i) verd.	e
0	0.000000	1.000000	1.000000	0.00
1	0.010000	1.005000	1.005050	0.01
2	0.020000	1.010101	1.010203	0.01
3	0.030000	1.015304	1.015460	0.02
4	0.040000	1.020613	1.020825	0.02
...				
45	0.450000	1.374620	1.380500	0.43
46	0.460000	1.388319	1.394506	0.44
47	0.470000	1.402389	1.408897	0.46
48	0.480000	1.416845	1.423690	0.48
49	0.490000	1.431700	1.438901	0.50
50	0.500000	1.446970	1.454545	0.52

>>



Se confirma la validez de los resultados obtenidos anteriormente.

➤ Método de Runge-Kutta de 4to orden.

Aplicando las ecuaciones (9.3) a (9.7) se obtiene lo siguiente:

Con $h = 0.1$,

$$i = 0, \quad x_0 = 0, \quad y_0 = 1$$

$$k_1 = f(x_0, y_0) = 0.5(1+x_0)y_0^2 = 0.5(1+0)1^2 = 0.5$$

$$\begin{aligned} k_2 &= f(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_1) = 0.5 \left[1 + \left(x_0 + \frac{h}{2} \right) \right] \left(y_0 + \frac{h}{2}k_1 \right)^2 = \\ &\quad 0.5 \left[1 + \left(0 + \frac{0.1}{2} \right) \right] \left(1 + \frac{0.1}{2}(0.5) \right)^2 = 0.551578 \end{aligned}$$

$$\begin{aligned} k_3 &= f(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}k_2) = 0.5 \left[1 + \left(x_0 + \frac{h}{2} \right) \right] \left(y_0 + \frac{h}{2}k_2 \right)^2 = \\ &\quad 0.5 \left[1 + \left(0 + \frac{0.1}{2} \right) \right] \left(1 + \frac{0.1}{2}(0.551578) \right)^2 = 0.554357 \end{aligned}$$

$$\begin{aligned} k_4 &= f(x_0 + h, y_0 + hk_3) = 0.5 [1 + (x_0 + h)] (y_0 + hk_3)^2 = \\ &\quad 0.5 [1 + (0 + 0.1)] (1 + 0.1(0.554357))^2 = 0.612670 \end{aligned}$$

$$\begin{aligned} y_1 &= y_0 + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) = 1 + \frac{0.1}{6} (0.5 + 0.551578 + 0.554357 + 0.612670) = \\ &= \mathbf{1.055409} \end{aligned}$$

$$i = 1, \quad x_1 = 0.1, \quad y_1 = 1.055409$$

$$k_1 = f(x_1, y_1) = 0.5(1+x_1)y_1^2 = 0.5(1+0.1)1.055409^2 = 0.612638$$

$$\begin{aligned} k_2 &= f(x_1 + \frac{h}{2}, y_1 + \frac{h}{2}k_1) = 0.5 \left[1 + \left(x_1 + \frac{h}{2} \right) \right] \left(y_1 + \frac{h}{2}k_1 \right)^2 = \\ &\quad 0.5 \left[1 + \left(0.1 + \frac{0.1}{2} \right) \right] \left(1.055409 + \frac{0.1}{2}(0.612638) \right)^2 = 0.678204 \\ k_3 &= f(x_1 + \frac{h}{2}, y_1 + \frac{h}{2}k_2) = 0.5 \left[1 + \left(x_1 + \frac{h}{2} \right) \right] \left(y_1 + \frac{h}{2}k_2 \right)^2 = \\ &\quad 0.5 \left[1 + \left(0.1 + \frac{0.1}{2} \right) \right] \left(1.055409 + \frac{0.1}{2}(0.678204) \right)^2 = 0.682304 \end{aligned}$$

$$k_4 = f(x_1 + h, y_1 + hk_3) = 0.5[1 + (x_1 + h)][(y_1 + hk_3)^2] =$$

$$0.5[1 + (0.1 + 0.1)][(1.055409 + 0.1(0.682304))^2] = 0.757539$$

$$y_2 = y_1 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1 + \frac{0.1}{6}(0.612638 + 0.678204 + 0.682304 + 0.757539) =$$

$$= 1.123596$$

$$i = 2, \quad x_0 = 0.2, \quad y_0 = 1.123596$$

$$k_1 = f(x_2, y_2) = 0.5(1 + x_2)y_2^2 = 0.5(1 + 0.2)1.123596^2 = 0.757480$$

$$k_2 = f(x_2 + \frac{h}{2}, y_2 + \frac{h}{2}k_1) = 0.5\left[1 + \left(x_2 + \frac{h}{2}\right)\right]\left(y_2 + \frac{h}{2}k_1\right)^2 =$$

$$0.5\left[1 + \left(0.2 + \frac{0.1}{2}\right)\right]\left(1 + \frac{0.1}{2}(0.757480)\right)^2 = 0.843132$$

$$k_3 = f(x_2 + \frac{h}{2}, y_2 + \frac{h}{2}k_2) = 0.5\left[1 + \left(x_2 + \frac{h}{2}\right)\right]\left(y_2 + \frac{h}{2}k_2\right)^2 =$$

$$0.5\left[1 + \left(0.2 + \frac{0.1}{2}\right)\right]\left(1 + \frac{0.1}{2}(0.843132)\right)^2 = 0.849361$$

$$k_4 = f(x_2 + h, y_2 + hk_3) = 0.5[1 + (x_2 + h)][(y_2 + hk_3)^2] =$$

$$0.5[1 + (0.2 + 0.1)][(1 + 0.1(0.849361))^2] = 0.949357$$

$$y_3 = y_2 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.123596 + \frac{0.1}{6}(0.757480 + 0.843132 + 0.849361 + 0.949357) =$$

$$= 1.208459$$

En la tabla 9.3 y figura 9.11 se muestran los resultados.

i	x_i	y_i	k_1	k_2	k_3	k_4	Sol. Analítica	ϵ
0	0.0	1.000000	0.500000	0.551578	0.554357	0.612670	1.000000	0.000000%
1	0.1	1.055409	0.612638	0.678204	0.682304	0.757539	1.055409	0.000003%
2	0.2	1.123596	0.757480	0.843132	0.849361	0.949357	1.123596	0.000006%
3	0.3	1.208459	0.949243	1.064704	1.074514	1.212135	1.208459	0.000008%
4	0.4	1.315790	1.211911	1.373466	1.389635	1.587230	1.315789	0.000005%
5	0.5	1.454545					1.454545	0.000015%

Tabla 9.3. Resultados aplicando rk4 en el ejemplo 9.5.1.

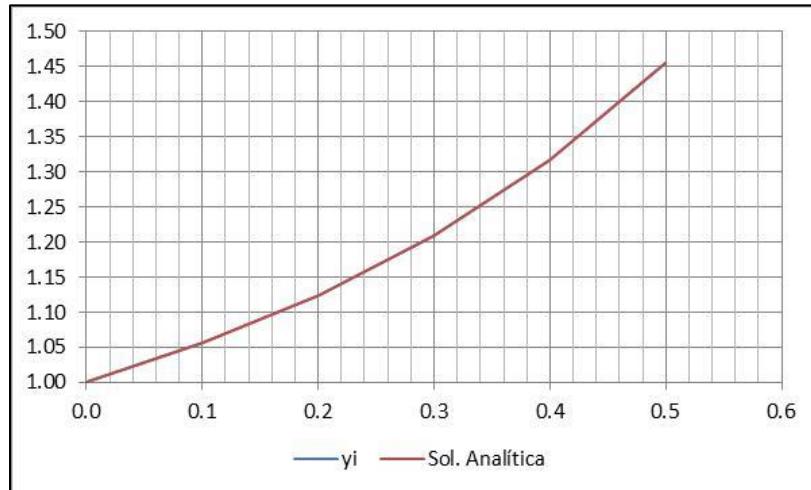


Figura 9.11. Resultados aplicando rk4 en el ejemplo 9.5.1.

Se observa que el método rk4 se obtiene una precisión importante con $h=0.1$.

- Resolviendo con código de Octave, *RK4_1_Octave.m*, se obtienen los resultados mostrados en la figura 9.12.

Figura 9.12. Resultados obtenidos con código en Octave, ejemplo 9.5.1.

Método de Runge-Kutta de 4to orden.

```
Introduce una función con respecto a x  0.5*(1+x)*y.^2
```

```
Introduce la solución analítica (Oprime Enter si no existe)  4./(4-2.*x-x.^2)
```

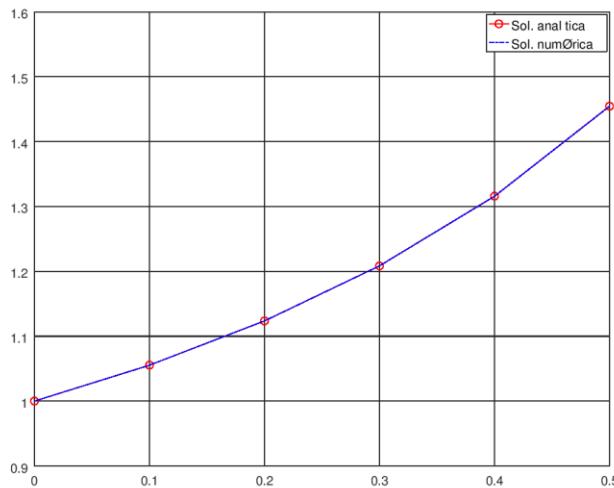
Datos:

```
x0 = 0
xn = 0.5
y0 = 1
```

Incremento de $x = 0.1$

i	x(i)	y(i)	k1	k2	k3	k4
0	0.00000000	1.00000000	0.50000000	0.55157813	0.55435716	0.61266950
1	0.10000000	1.05540900	0.61263849	0.67820381	0.68230437	0.75753935
2	0.20000000	1.12359557	0.75748021	0.84313224	0.84936135	0.94935678
3	0.30000000	1.20845931	0.94924303	1.06470363	1.07451426	1.21213474
4	0.40000000	1.31578953	1.21191147	1.37346607	1.38963458	1.58722970
5	0.50000000	1.45454524				

>>



Ejemplo 9.5.2 Un paracaidista con una masa de 69.1 kg. salta de un globo aerostático fijo. Calcular la velocidad y la altura antes de abrir el paracaídas. El coeficiente de resistencia es de aproximadamente 0.25 kg/s.

La aceleración de un cuerpo en caída se puede calcular con la segunda ley de Newton,

$$a = \frac{dv}{dt} = \frac{F}{m}$$

donde v es la velocidad de caída que depende del tiempo t , F la fuerza resultante ejercida sobre el cuerpo y m la masa del cuerpo.

La fuerza resultante se obtiene restando a la fuerza de gravedad (mg) la fuerza de resistencia o fricción ($c_d v^2$). Donde c_d es el coeficiente de resistencia o de fricción. Entonces,

$$a = \frac{dv}{dt} = \frac{F}{m} = \frac{mg - c_d v^2}{m} = g - \frac{c_d}{m} v^2$$

Para obtener la variación de la altura al ir cayendo el cuerpo, y , se puede proceder de la siguiente forma considerando que $v = \frac{dy}{dt}$,

$$\frac{dy}{dt} = v$$

$$\frac{dv}{dy} = g - \frac{c_d}{m} v^2$$

$$t = 0, \quad y(0) = v(0) = 0$$

Con lo cual se tienen dos ecuaciones diferenciales ordinarias de primer orden, las cuales suponen que el cuerpo salta desde un estado inmóvil.

➤ Resolviendo con Euler.

Las funciones son:

$$f_1(t, y, v) = v$$

$$f_2(t, y, v) = g - \frac{c}{m} v^2$$

Y los planteamientos de Euler,

$$y_{i+1} = y_i + h f_1(t_i, y_i, v_i)$$

$$v_{i+1} = v_i + h f_2(t_i, y_i, v_i)$$

Iniciando el proceso iterativo se obtiene lo siguiente:

$$i=0, t_0=0, y_0=0, v_0=0$$

$$f_1(t_0, y_0, v_0) = v_0 = 0$$

$$f_2(t_0, y_0, v_0) = g - \frac{c}{m} v_0^2 = 9.81 - \frac{0.25}{68.1} 0^2 = 9.81$$

$$y_1 = y_0 + h f_1(t_0, y_0, v_0) = 0 + 2(0) = 0$$

$$v_1 = v_0 + h f_2(t_0, y_0, v_0) = 0 + 2(9.81) = 19.62$$

$$i=1, t_1=2, y_1=0, v_1=19.62$$

$$f_1(t_1, y_1, v_1) = v_1 = 19.62$$

$$f_2(t_1, y_1, v_1) = g - \frac{c}{m} v_1^2 = 9.81 - \frac{0.25}{68.1} (19.62)^2 = 8.396841$$

$$y_2 = y_1 + h f_1(t_1, y_1, v_1) = 0 + 2(19.62) = 39.240000$$

$$v_2 = v_1 + h f_2(t_1, y_1, v_1) = 19.62 + 2(8.396841) = 36.413683$$

$$i=2, t_2=4, y_2=39.240000, v_2=36.413683$$

$$f_1(t_2, y_2, v_2) = v_2 = 36.413683$$

$$f_2(t_2, y_2, v_2) = g - \frac{c}{m} v_2^2 = 9.81 - \frac{0.25}{68.1} (36.413683)^2 = 4.942319$$

$$y_3 = y_2 + h f_1(t_2, y_2, v_2) = 39.24 + 2(36.413683) = 112.067366$$

$$v_3 = v_2 + h f_2(t_2, y_2, v_2) = 36.413683 + 2(4.942319) = 46.298321$$

En la tabla 9.4 y en las gráficas 9.3 y 9.4 se muestran los resultados.

<i>i</i>	<i>t_i</i>	<i>y_i</i>	Sol. Analítica	ϵ	<i>v_i</i>	Sol. Analítica	ϵ
0	0.00	0.000000	0.000000		0.000000	0.000000	
1	2.00	0.000000	19.166286	100.00%	19.620000	18.729189	4.76%
2	4.00	39.240000	71.930366	45.45%	36.413683	33.111825	9.97%
3	6.00	112.067366	147.946180	24.25%	46.298321	42.076227	10.03%
4	8.00	204.664007	237.510440	13.83%	50.180182	46.957495	6.86%
5	10.00	305.024371	334.178167	8.72%	51.312292	49.421367	3.83%

Tabla 9.4. Resultados del ejemplo 9.5.2 aplicando Euler.

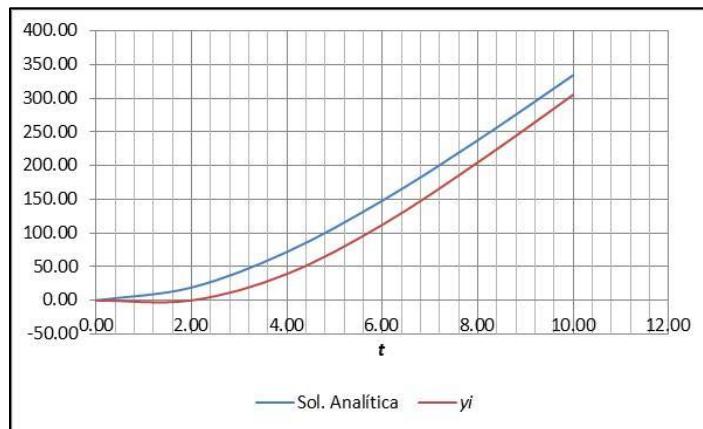


Figura 9.3. Grafica de *t* vs *y* del ejemplo 9.5.2.

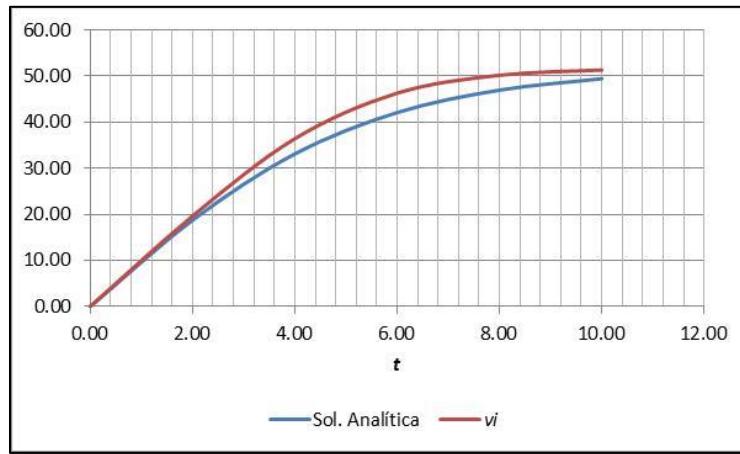


Figura 9.4. Grafica de t vs v del ejemplo 9.5.2.

➤ Resolviendo con Runge-Kutta de 4to orden, rk4.

$$k_1 = f(t_i, v_i)$$

$$l_1 = f(t_i, v_i)$$

$$k_2 = f\left(t_i + \frac{h}{2}, v_i + \frac{h}{2}k_1\right)$$

$$l_2 = f\left(t_i + \frac{h}{2}, v_i + \frac{h}{2}l_1\right)$$

$$k_3 = f\left(t_i + \frac{h}{2}, v_i + \frac{h}{2}k_2\right)$$

$$l_3 = f\left(t_i + \frac{h}{2}, v_i + \frac{h}{2}l_2\right)$$

$$k_4 = f(t_i + h, v_i + hk_3)$$

$$l_4 = f(t_i + h, v_i + hl_3)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$v_{i+1} = v_i + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4)$$

donde k servirá para calcular los valores de y_i y l los de v_i . Entonces, iniciando con el proceso iterativo,

Para $i = 0$, $t_0 = 0$, $y_0 = 0$, $v_0 = 0$,

$$k_1 = f_1(t_0, v_0, y_0) = v_0 = 0$$

$$l_1 = f_2(t_0, v_0, y_0) = g - \frac{c_v}{m}v_0^2 = 9.81 - \frac{0.25}{68.1}0^2 = 9.81$$

$$k_2 = f_1\left(t_0 + \frac{h}{2}, y_0 + k_1 \frac{h}{2}, v_0 + l_1 \frac{h}{2}\right) = v_0 + l_1 \frac{h}{2} = 0 + 9.81 \frac{2}{2} = 9.81$$

$$l_2 = f_2\left(t_0 + \frac{h}{2}, y_0 + k_1 \frac{h}{2}, v_0 + l_1 \frac{h}{2}\right) = g - \frac{c_v}{m}\left(v_0 + l_1 \frac{h}{2}\right) = 9.81 - \frac{0.25}{68.1}\left(0 + 9.81 \frac{2}{2}\right)^2 \\ = 9.456710$$

$$k_3 = f_1\left(t_0 + \frac{h}{2}, y_0 + k_2 \frac{h}{2}, v_0 + l_2 \frac{h}{2}\right) = v_0 + l_2 \frac{h}{2} = 0 + 9.456710 \frac{2}{2} = 9.456710$$

$$l_3 = f_2\left(t_0 + \frac{h}{2}, y_0 + k_2 \frac{h}{2}, v_0 + l_2 \frac{h}{2}\right) = g - \frac{c_v}{m} \left(v_0 + l_2 \frac{h}{2}\right)^2 = 9.81 - \frac{0.25}{68.1} \left(0 + 9.456710 \frac{2}{2}\right)^2 = 9.481698$$

$$k_4 = f_1(t_0 + h, y_0 + k_3 h, v_0 + l_3 h) = v_0 + l_3 h = 0 + 9.481698(2) = 18.963397$$

$$l_4 = f_2(t_0 + h, y_0 + k_3 h, v_0 + l_3 h) = g - \frac{c_v}{m} (v_0 + l_3 h)^2 = 9.81 - \frac{0.25}{68.1} (0 + 9.481698(2))^2 = 8.489844$$

$$y_1 = y_0 + \left(\frac{h}{6}\right)(k_1 + 2k_2 + 2k_3 + 2k_4) = 0 + \left(\frac{2}{6}\right)[0 + 2(9.81) + 2(9.456710) + 18.963397] = 19.165606$$

$$v_1 = v_0 + \left(\frac{h}{6}\right)(l_1 + 2l_2 + 2l_3 + 2l_4) = 0 + \left(\frac{2}{6}\right)[9.81 + 2(9.456710) + 2(9.481698) + 8.489844] = 18.724554$$

Para $i = 1$, $t_1 = 2$, $y_1 = 19.165606$, $v_1 = 17.724554$,

$$k_1 = f_1(t_1, y_1, v_1) = v_1 = 18.725554$$

$$l_1 = f_2(t_1, y_1, v_1) = g - \frac{c_v}{m} v_1^2 = 9.81 - \frac{0.25}{68.1} (18.725554)^2 = 8.522752$$

$$k_2 = f_1\left(t_0 + \frac{h}{2}, y_1 + k_1 \frac{h}{2}, v_1 + l_1 \frac{h}{2}\right) = v_1 + l_1 \frac{h}{2} = 18.725554 + 18.725554 \left(\frac{2}{2}\right) = 27.248306$$

$$l_2 = f_2\left(t_0 + \frac{h}{2}, y_1 + k_1 \frac{h}{2}, v_1 + l_1 \frac{h}{2}\right) = g - \frac{c_v}{m} \left(v_1 + l_1 \frac{h}{2}\right)^2 = 9.81 - \frac{0.25}{68.1} \left[\left(18.725554 + 8.522752 \frac{2}{2}\right) \right]^2 = 7.084339$$

$$k_3 = f_1\left(t_0 + \frac{h}{2}, y_1 + k_2 \frac{h}{2}, v_1 + l_2 \frac{h}{2}\right) = v_1 + l_2 \frac{h}{2} = 18.725554 + 7.084339 \left(\frac{2}{2}\right) = 25.809892$$

$$l_3 = f_2\left(t_0 + \frac{h}{2}, y_1 + k_2 \frac{h}{2}, v_1 + l_2 \frac{h}{2}\right) = g - \frac{c_v}{m} \left(v_1 + l_2 \frac{h}{2}\right)^2 = 9.81 - \frac{0.25}{68.1} [(18.725554 + 7.084339 * 2)]^2 = 7.364513$$

$$k_4 = f_1(t_0 + h, y_1 + k_3 h, v_1 + l_3 h) = v_1 + l_3 h = 18.725554 + 7.364513(2) = 33.454581$$

$$l_4 = f_2(t_0 + h, y_1 + k_3 h, v_1 + l_3 h) = g - \frac{c_v}{m} (v_1 + l_3 h)^2 = 9.81 - \frac{0.25}{68.1} (18.725554 + 7.364513 * 2)^2 = 5.701303$$

$$y_2 = y_1 + \left(\frac{h}{6}\right)(k_1 + 2k_2 + 2k_3 + 2k_4) = 19.165606 + \left(\frac{2}{6}\right)[18.725554 + 2(27.248306) + 2(25.809892) + 33.454581] = 71.931116$$

$$v_2 = v_1 + \left(\frac{h}{6}\right)(l_1 + 2l_2 + 2l_3 + 2l_4) = 18.522752 + \left(\frac{2}{6}\right)[8.522752 + 2(7.084339) + 2(7.364513) + 5.701303] = 33.099474$$

Para $i = 2$, $t_2 = 6$, $y_2 = 71.931116$, $v_2 = 33.099474$,

$$k_1 = f_1(t_2, y_2, v_2) = v_2 = 33.099474$$

$$l_1 = f_2(t_2, y_2, v_2) = g - \frac{c_v}{m} v_2^2 = 9.81 - \frac{0.25}{68.1} (33.099474)^2 = 5.788065$$

$$k_2 = f_1\left(t_2 + \frac{h}{2}, y_2 + k_1 \frac{h}{2}, v_2 + l_1 \frac{h}{2}\right) = v_2 + l_1 \frac{h}{2} = 33.099474 + 5.788065 \left(\frac{2}{2}\right) = 38.887538$$

$$l_2 = f_2\left(t_2 + \frac{h}{2}, y_2 + k_1 \frac{h}{2}, v_2 + l_1 \frac{h}{2}\right) = g - \frac{c_v}{m} \left(v_2 + l_1 \frac{h}{2}\right)^2 = 9.81 - \frac{0.25}{68.1} \left(33.099474 + 5.788065 \left(\frac{2}{2}\right)\right)^2 = 4.258456$$

$$k_3 = f_1\left(t_2 + \frac{h}{2}, y_2 + k_2 \frac{h}{2}, v_2 + l_2 \frac{h}{2}\right) = v_2 + l_2 \frac{h}{2} = 33.099474 + 4.258456 \left(\frac{2}{2}\right) = 37.357929$$

$$l_3 = f_2\left(t_2 + \frac{h}{2}, y_2 + k_2 \frac{h}{2}, v_2 + l_2 \frac{h}{2}\right) = g - \frac{c_v}{m} \left(v_2 + l_2 \frac{h}{2}\right)^2 = 9.81 - \frac{0.25}{68.1} \left(33.099474 + 4.258456 \left(\frac{2}{2}\right)\right)^2 = 4.686597$$

$$k_4 = f_1(t_2 + h, y_2 + k_3 h, v_2 + l_3 h) = v_2 + l_3 h = 33.099474 + 4.686597(2) = 42.472668$$

$$l_4 = f_2(t_2 + h, y_2 + k_3 h, v_2 + l_3 h) = g - \frac{c_v}{m} (v_2 + l_3 h)^2 = 9.81 - \frac{0.25}{68.1} (33.099474 + 4.686597(2))^2 = 3.187652$$

$$y_3 = y_2 + \left(\frac{h}{6}\right)(k_1 + 2k_2 + 2k_3 + 2k_4) = 71.931116 + \left(\frac{2}{6}\right)[33.099474 + 2(38.887538) + 2(37.357929) + 42.472668] = 147.952142$$

$$v_3 = v_2 + \left(\frac{h}{6}\right)(l_1 + 2l_2 + 2l_3 + 2l_4) = 33.099474 + \left(\frac{2}{6}\right)[5.788065 + 2(4.258456) + 2(4.686597) + 3.187652] = 42.054748$$

Los resultados finales se muestran en la tabla 9.5 y en las figuras 9.20 y 9.21.

Tabla 9.5. Resultados del ejemplo 9.5.2 usando rk4.

t_i	y_i	v_i	k_1	l_1	k_2	l_2	k_3	l_3	k_4	l_4	Sol. Analítica y_i	ϵ	Sol. Analítica y_i	ϵ
0.00	0.000000	0.000000	0.000000	9.810000	9.810000	9.456710	9.456710	9.481698	18.963397	8.489844	0.000000		0.000000	
2.00	19.165606	18.725554	18.725554	8.522752	27.248306	7.084339	25.808892	7.364513	33.454581	5.701303	19.166286	0.004%	18.729189	0.02%
4.00	71.931116	33.099474	33.099474	5.788065	38.887538	4.258456	37.357929	4.686597	42.472668	3.187652	71.930366	0.001%	33.111825	0.04%
6.00	147.952142	42.054748	42.054748	3.317335	45.372083	2.252636	44.307384	2.603156	47.261060	1.610265	147.946180	0.004%	42.076227	0.05%
8.00	237.510390	46.934476	46.934476	1.723197	48.657674	1.118483	48.052959	1.333176	49.600828	0.778274	237.510440	0.000%	46.957495	0.05%
10.00	334.162580	49.402739	49.402739	0.850269	50.253009	0.539204	49.941943	0.653621	50.709981	0.369831	334.178167	0.005%	49.421367	0.04%

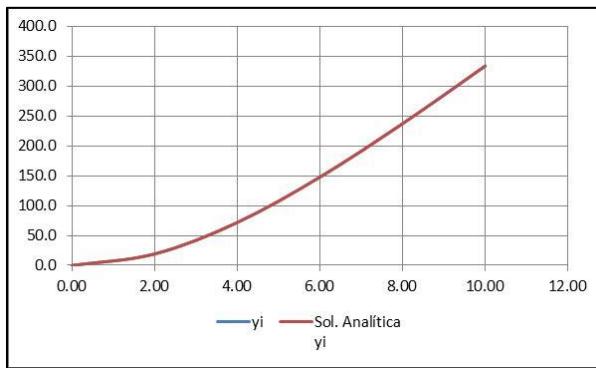


Figura 9.20. Gráfica de los resultados del ejemplo 9.5.2, t vs y .

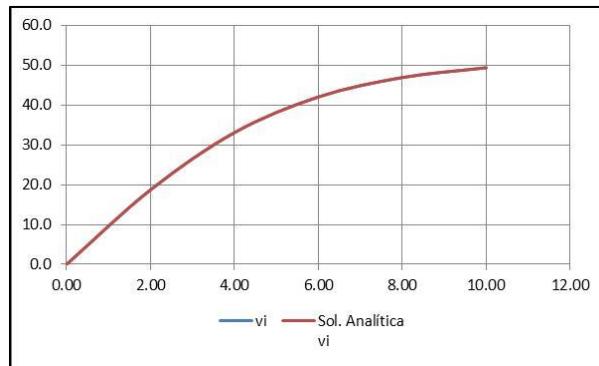


Figura 9.21. Gráfica de los resultados del ejemplo 9.5.2, t vs v .

➤ Resolviendo analíticamente con comandos de Octave se obtiene lo siguiente:

```
>> syms y(t) v(t) c m g
>> ode_sys=[diff(y(t))==v(t); diff(v(t))==g-(c/m)*v(t)^2];
>> soln = dsolve (ode_sys);
>> soln{1}
ans = (sym)

y(t) = C1 - -----
          /           g*m
          m*log|-g*m + -----| \
          |           /   / \   / -2*C2 - 2*t\ \
          |           2| \ c * \ / g *log\ e           / || |
          |           tanh |-----|| |
          |           |           /   / \   / |
          \           \           2*\ / m           // |

v(t) = -----
          -\ / g * \ / m
          /   / \   / -2*C2 - 2*t\ \
          | \ / c * \ / g *log\ e           / || |
          \ / c *tanh|-----|| |
          |           /   / \   / |
          \           2*\ / m           / |
```

>>

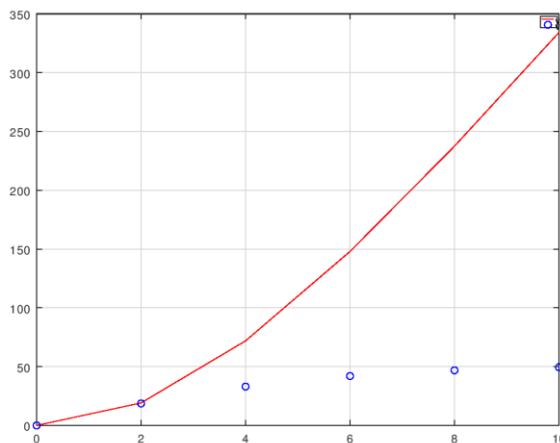
➤ Resolviendo numéricamente con comandos de Octave se obtiene lo siguiente:

```
>> function fun=fcn(y,t)
g=9.81;c=0.25;m=68.1;
fun(1)=y(2);
```

```

fun(2)=g-(c/m)*y(2)^2;
endfunction
>> y0=[0,0];
>> t=[0:2:10];
>> y=lsode("fcn",y0,t);
>> figure(1)
>> plot(t,y(:,1),'r-',t,y(:,2),'bo')
>> legend('y','v')
>> grid
>> for i=1:length(t)
fprintf('%4.1f %8.6f %8.6f',t(i),y(i,1),y(i,2));
fprintf('\n')
endfor
0.0 0.000000 0.000000
2.0 19.166286 18.729189
4.0 71.930363 33.111824
6.0 147.946171 42.076227
8.0 237.510431 46.957497
10.0 334.178162 49.421369
>>

```



➤ Resolviendo con código de Octave, *Euler_2_Octave.m* y *RK4_2_Octave.m*,

Figura 9.7. Resultados del ejemplo 9.5.2 usando código Octave para Euler y rk4.

Método de Euler.

Introduce la función $y=f(t,y,v)$ v

Introduce la solución analítica $y=f(t,y,v)$ [Enter si no existe]

Introduce la función $v=f(t,y,v)$ $9.81 - (0.25/68.1)*v^2$

Datos:

```

t0 = 0
tn = 10
y0 = 0
v0 = 0

```

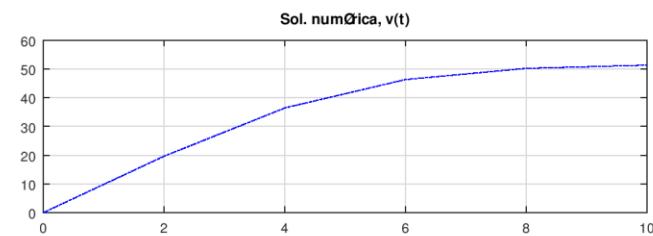
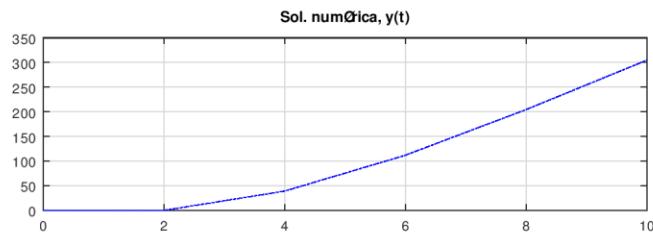
Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Incremento de $t = 2$

i	t(i)	y(i)	v(i)
0	0.00000	0.000000	0.000000
1	2.00000	0.000000	19.620000
2	4.00000	39.240000	36.413683
3	6.00000	112.067366	46.298321
4	8.00000	204.664007	50.180182
5	10.00000	305.024371	51.312292

>>



Método de Runge-Kutta de cuarto orden.

Introduce la función $y=f(t,y,v)$ v

Introduce la solución analítica $y=f(t,y,v)$ [Enter si no existe]

Introduce la función $v=f(t,y,v)$ $9.81 - (0.25/68.1) * v^2$

Datos:

$t_0 = 0$

$t_n = 12$

$y_0 = 0$

$v_0 = 0$

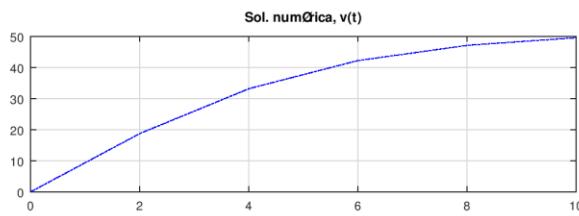
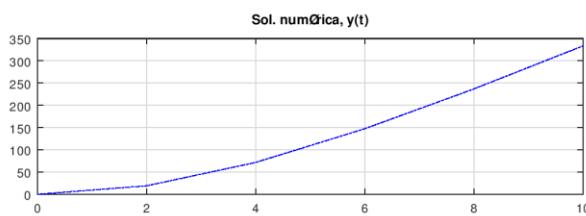
Incremento de $t = 2$

i	t(i)	y(i)	v(i)	k1	k2	k3	k4
11							
12	0.000000	0.000000	0.000000	0.000000	9.810000	9.456710	
13							
14							
0							
18.913421							
	9.810000						
		9.481698					
			8.496793				
9.456710							

```

1      2.000000   19.148947   18.727870   18.727870   27.250304   25.811809
32.895748   8.522433
    7.083939   7.364150   5.837422
2      4.000000   71.731562   33.146548   33.146548   38.923165   37.394827
41.643107   5.776617
    4.248279   4.676472   3.443817
3      6.000000   147.540108   42.169860   42.169860   45.451603   44.395982
46.622105   3.281743
    2.226122   2.574305   1.830482
4      8.000000   237.035820   47.074220   47.074220   48.749190   48.159978
49.245736   1.674970
    1.085758   1.295376   0.907127
5      10.000000   333.748584   49.522342
>>

```



9.6 Ejercicios propuestos.

- (a) Resolver el siguiente problema con valor inicial en el intervalo $t=0$ a 2 , donde $y(0)=1$. Despliegue los resultados en una gráfica.

$$\frac{dy}{dt} = y t^2 - 1.5 y$$

- i. Analíticamente
- ii. Usando el método de Euler con $h=0.5$ y $h=0.25$
- iii. Usando rk4 con $h=0.5$

- (b) Resolver el siguiente par de ecuaciones diferenciales ordinarias en el intervalo $t=0$ a 0.4 , usando $h=0.1$. Las condiciones iniciales son $y(0)=0$ y $z(0)=4$. Obtener la solución con (a) método de Euler y (b) método de Runge-Kutta de 4to orden. Elabore una gráfica con los resultados.

$$\frac{dy}{dx} = -2y + 5e^{-t}$$

$$\frac{dz}{dt} = -\frac{yz^2}{2}$$

Apéndice A. Introducción a la dinámica de sistemas.

1) MODELOS PRESA-DEPREDADOR².

Antecedente. Ingenieros y científicos se ocupan de una variedad de problemas que incluyen sistemas de ecuaciones diferenciales ordinarias no lineales. Este caso de estudio se enfoca a dos de estas aplicaciones. La primera se refiere a los modelos de presa-depredador que se utilizan para estudiar las interacciones de las especies. La segunda son ecuaciones derivadas de la dinámica de fluidos que se utilizan para simular la atmósfera.

Los modelos de presa-depredador se desarrollaron de forma independiente en la primera parte del siglo XX por el matemático italiano Vito Volterra y el biólogo americano Alfred Lotka. Estas ecuaciones comúnmente se denominan ecuaciones Lotka-Volterra. La versión más simple son los siguientes pares de ODE's:

$$\frac{dx}{dt} = ax - bxy \quad (\text{A.1})$$

$$\frac{dy}{dt} = -cy + dxy \quad (\text{A.2})$$

donde x y y = número de presas y depredadores, respectivamente, a = tasa de crecimiento de la presa, c = tasa de mortalidad del depredador y b y d = tasas que caracterizan el efecto de las interacciones presa-depredador sobre la muerte de las presas y el crecimiento del depredador, respectivamente. La multiplicación los términos (es decir, los que implican xy) son lo que hacen tales ecuaciones no lineales.

Entonces, cuando hay alimento ilimitado y ausencia de depredadores, se tiene que $dx/dt = ax$, $a > 0$, por lo que el número de presas, $x(t)$, crecerá exponencialmente. Cuando existen depredadores, el número de presas decrece debido a su consumo por parte de depredadores, y , con una tasa proporcional a xy , quedando $-bxy$. Por otro lado, cuando existe ausencia de presas, el número de depredadores decrece exponencialmente hasta cero, $dy/dt = -cy$. En cambio, cuando existen presas, la densidad de depredadores $y(t)$ aumentará con una tasa de crecimiento proporcional según en número de encuentros presa-depredador, dxy .

A continuación se resolverá un ejercicio usando los métodos numéricos vistos anteriormente, Euler y Runge-Kutta de 4to orden, así como conceptos básicos de dinámica de sistemas.

Ejercicio A.1 Se requiere simular la dinámica del comportamiento presa-depredador, suponiendo los siguientes parámetros: $a = 1.2$, $b = 0.6$, $c = 0.8$ y $d = 0.3$. Suponer las condiciones iniciales de $x = 2$ e $y = 1$ e integrar desde $t = 0$ a 30, utilizando un tamaño de paso de $h = 0.0625$.

Resolviendo con Euler aplicando Octave (figura A.1), se obtiene los siguientes resultados (figuras A.2 y A.3):

Figura A.1. Código en Octave, *Euler*.

```
% Método de Euler para resolver modelo presa-depredador;
%
clc; clear;
%
% Caso de estudio 22.6.
fx=@(t,x,y) 1.2*x-0.6*x*y;
fy=@(t,x,y) -0.8*y+0.3*x*y;
%
disp('Método de Euler.');
disp(' ');
% Lectura de datos
%
disp('Datos:');
t0=input('t0 = ');
tn=input('tn = ');
x0=input('x0 = ');
y0=input('y0 = ');
dt=input('Incremento de t = ');
disp(' ');
disp('    i      t(i)      x(i)      y(i)');
ndt=(tn-t0)/dt;
t = [t0:dt:tn];
n = length(t);
x = zeros(1,n);
y=x;
iter=0;
fprintf('%4.0f %12.4f %12.6f %12.6f',iter,t0,x0,y0);fprintf('\n');
for i = 1:ndt
    if i==1
        x(i) = x0 + dt * fx(t0, x0, y0);
        y(i) = y0 + dt * fy(t0, x0, y0);
    else
        x(i) = x(i-1) + dt * fx(t(i-1), x(i-1), y(i-1));
        y(i) = y(i-1) + dt * fy(t(i-1), x(i-1), y(i-1));
    end
    fprintf('%4.0f %12.4f %12.6f %12.6f',i,t(i),x(i),y(i));fprintf('\n');
end
hold on
plot(t,x,'-r',t,y,'-.b'); grid;
legend('Presa','Depredador')
hold off
```

Figura A.2. Resultados arrojados por el código en Octave.

Método de Euler.

Datos:

```
t0 = 0
tn = 30
x0 = 2
y0 = 1
Incremento de t = 0.0625
```

i	t(i)	x(i)	y(i)
0	0.0000	2.000000	1.000000
1	0.0625	2.075000	0.987500
2	0.1250	2.153785	0.976545
3	0.1875	2.236446	0.967154
4	0.2500	2.323068	0.959352
5	0.3125	2.413724	0.953172
.	.	.	.
475	29.6875	2.558825	0.274585
476	29.7500	2.724388	0.274029
477	29.8125	2.900722	0.274326
478	29.8750	3.088435	0.275530
479	29.9375	3.288157	0.277709
480	30.0000	3.500526	0.280945

>>

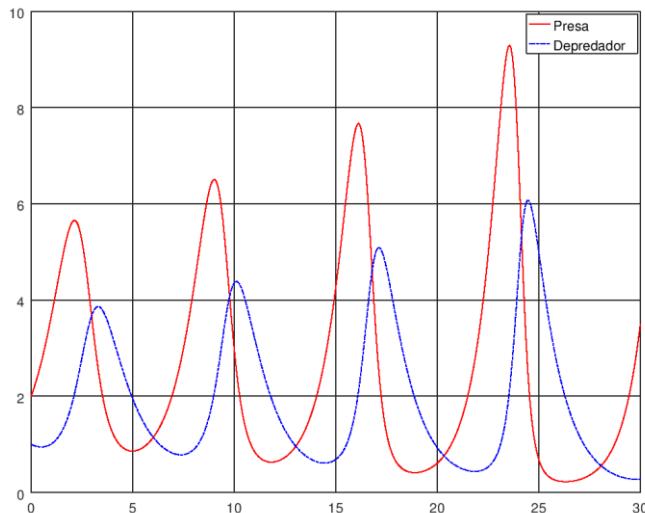


Figura A.3. Gráfica de los resultados arrojados por el código de la figura H.1.

Ahora, resolviendo con *rk4* aplicando Octave (figura A.4), se obtienen los siguientes resultados (figuras A.5 y A.6):

Figura A.4. Código en Octave.

```
% Runge-Kutta 4to para dos ODE's.
% Octubre del 2017
%
clear;clc;
%
% Caso de estudio 22.6 modelo presa-depredador.
fx=@(t,x,y) 1.2*x-0.6*x*y;
fy=@(t,x,y) -0.8*y+0.3*x*y;
%
disp('Método de Runge-Kutta de cuarto orden.');
disp(' ');
% Lectura de datos
%
disp('Datos:');
t0=input('t0 = ');
tn=input('tn = ');
x0=input('x0 = ');
y0=input('y0 = ');
dt=input('Incremento de t = ');
ndt=(tn-t0)/dt;
t = [t0:dt:tn];
n = length(t);
x = zeros(1,n);
y=x;
xact = x0;
yact = y0;
%
disp(' ');
disp(' i      t(i)      x(i)      y(i)      k1      k2      k3');
k4      11      12      13      14')
i=0;
while i<=ndt
    tant=t(i+1); xant=xact; yant=yact;
    k1 = fx(tant,xant,yant);
    l1 = fy(tant,xant,yant);
    k2 = fx(tant+dt/2,xant+dt*k1/2,yant+dt*l1/2);
    l2 = fy(tant+dt/2,xant+dt*k1/2,yant+dt*l1/2);
    k3 = fx(tant+dt/2,xant+dt*k2/2,yant+dt*l2/2);
    l3 = fy(tant+dt/2,xant+dt*k2/2,yant+dt*l2/2);
    k4 = fx(tant+dt,xant+dt*k2,yant+dt*l2);
    l4 = fy(tant+dt,xant+dt*k2,yant+dt*l2);
    xact = xant+dt*(k1+2*k2+2*k3+k4)/6;
    yact = yant+dt*(l1+2*l2+2*l3+l4)/6;
    fprintf('%4.0f %12.4f %12.6f %12.6f %12.6f %12.6f %12.6f %12.6f\n',i,tant,xant,yant,k1,k2,k3,k4,11,12,13,14);
    i=i+1;
    x(i)=xant; y(i)=yact;
end
x(i)=xact; y(i)=yact;
hold on
plot(t,x,'-r',t,y,'-b'); grid;
legend('Presas','Depredadores')
hold off
```

Figura A.5. Resultados arrojados por el código de la figura A.4.

Método de Runge-Kutta de cuarto orden.																																																																																																																																																																																																																	
Datos:																																																																																																																																																																																																																	
$t_0 = 0$																																																																																																																																																																																																																	
$t_n = 30$																																																																																																																																																																																																																	
$x_0 = 2$																																																																																																																																																																																																																	
$y_0 = 1$																																																																																																																																																																																																																	
Incremento de $t = 0.0625$																																																																																																																																																																																																																	
<table border="1"> <thead> <tr> <th>i</th><th>t(i)</th><th>x(i)</th><th>y(i)</th><th>k1</th><th>k2</th><th>k3</th><th>k4</th><th>11</th><th>12</th><th>13</th><th>14</th><th></th><th></th></tr> </thead> <tbody> <tr><td>0</td><td>0.0000</td><td>2.00000</td><td>1.00000</td><td>1.20000</td><td>1.230141</td><td>1.230234</td><td>1.260739</td><td>-0.20000</td><td>-0.187570</td><td>-0.187363</td><td>-0.174861</td><td></td><td></td></tr> <tr><td>1</td><td>0.0625</td><td>2.076891</td><td>0.988284</td><td>1.260734</td><td>1.291588</td><td>1.291677</td><td>1.322866</td><td>-0.174860</td><td>-0.162277</td><td>-0.162057</td><td>-0.149377</td><td></td><td></td></tr> <tr><td>2</td><td>0.1250</td><td>2.157621</td><td>0.978150</td><td>1.322859</td><td>1.354364</td><td>1.354443</td><td>1.386243</td><td>-0.149377</td><td>-0.136591</td><td>-0.136359</td><td>-0.123450</td><td></td><td></td></tr> <tr><td>3</td><td>0.1875</td><td>2.242274</td><td>0.969621</td><td>1.386235</td><td>1.418307</td><td>1.418370</td><td>1.450688</td><td>-0.123450</td><td>-0.110408</td><td>-0.110164</td><td>-0.096970</td><td></td><td></td></tr> <tr><td>4</td><td>0.2500</td><td>2.330923</td><td>0.962730</td><td>1.450678</td><td>1.483212</td><td>1.483252</td><td>1.515971</td><td>-0.096969</td><td>-0.083612</td><td>-0.083355</td><td>-0.069814</td><td></td><td></td></tr> <tr><td>5</td><td>0.3125</td><td>2.423627</td><td>0.957514</td><td>1.515958</td><td>1.548825</td><td>1.548832</td><td>1.581807</td><td>-0.069814</td><td>-0.056078</td><td>-0.055809</td><td>-0.041854</td><td></td><td></td></tr> <tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td></tr> <tr><td>475</td><td>29.6875</td><td>3.842767</td><td>3.392046</td><td>-3.209585</td><td>-3.209794</td><td>-3.203474</td><td>-3.193199</td><td>1.196816</td><td>1.106820</td><td>1.105906</td><td>1.012914</td><td></td><td></td></tr> <tr><td>476</td><td>29.7500</td><td>3.642462</td><td>3.461163</td><td>-3.193337</td><td>-3.173153</td><td>-3.167386</td><td>-3.138022</td><td>1.013215</td><td>0.917918</td><td>0.917796</td><td>0.820669</td><td></td><td></td></tr> <tr><td>477</td><td>29.8125</td><td>3.444415</td><td>3.518510</td><td>-3.138227</td><td>-3.100385</td><td>-3.095311</td><td>-3.049770</td><td>0.820955</td><td>0.722668</td><td>0.723298</td><td>0.624329</td><td></td><td></td></tr> <tr><td>478</td><td>29.8750</td><td>3.250880</td><td>3.563689</td><td>-3.050019</td><td>-2.997550</td><td>-2.993245</td><td>-2.934624</td><td>0.624586</td><td>0.525549</td><td>0.526856</td><td>0.428203</td><td></td><td></td></tr> <tr><td>479</td><td>29.9375</td><td>3.063732</td><td>3.596581</td><td>-2.934897</td><td>-2.870912</td><td>-2.867394</td><td>-2.798778</td><td>0.428423</td><td>0.330691</td><td>0.332575</td><td>0.236170</td><td></td><td></td></tr> <tr><td>480</td><td>30.0000</td><td>2.884458</td><td>3.617321</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>														i	t(i)	x(i)	y(i)	k1	k2	k3	k4	11	12	13	14			0	0.0000	2.00000	1.00000	1.20000	1.230141	1.230234	1.260739	-0.20000	-0.187570	-0.187363	-0.174861			1	0.0625	2.076891	0.988284	1.260734	1.291588	1.291677	1.322866	-0.174860	-0.162277	-0.162057	-0.149377			2	0.1250	2.157621	0.978150	1.322859	1.354364	1.354443	1.386243	-0.149377	-0.136591	-0.136359	-0.123450			3	0.1875	2.242274	0.969621	1.386235	1.418307	1.418370	1.450688	-0.123450	-0.110408	-0.110164	-0.096970			4	0.2500	2.330923	0.962730	1.450678	1.483212	1.483252	1.515971	-0.096969	-0.083612	-0.083355	-0.069814			5	0.3125	2.423627	0.957514	1.515958	1.548825	1.548832	1.581807	-0.069814	-0.056078	-0.055809	-0.041854			475	29.6875	3.842767	3.392046	-3.209585	-3.209794	-3.203474	-3.193199	1.196816	1.106820	1.105906	1.012914			476	29.7500	3.642462	3.461163	-3.193337	-3.173153	-3.167386	-3.138022	1.013215	0.917918	0.917796	0.820669			477	29.8125	3.444415	3.518510	-3.138227	-3.100385	-3.095311	-3.049770	0.820955	0.722668	0.723298	0.624329			478	29.8750	3.250880	3.563689	-3.050019	-2.997550	-2.993245	-2.934624	0.624586	0.525549	0.526856	0.428203			479	29.9375	3.063732	3.596581	-2.934897	-2.870912	-2.867394	-2.798778	0.428423	0.330691	0.332575	0.236170			480	30.0000	2.884458	3.617321										
i	t(i)	x(i)	y(i)	k1	k2	k3	k4	11	12	13	14																																																																																																																																																																																																						
0	0.0000	2.00000	1.00000	1.20000	1.230141	1.230234	1.260739	-0.20000	-0.187570	-0.187363	-0.174861																																																																																																																																																																																																						
1	0.0625	2.076891	0.988284	1.260734	1.291588	1.291677	1.322866	-0.174860	-0.162277	-0.162057	-0.149377																																																																																																																																																																																																						
2	0.1250	2.157621	0.978150	1.322859	1.354364	1.354443	1.386243	-0.149377	-0.136591	-0.136359	-0.123450																																																																																																																																																																																																						
3	0.1875	2.242274	0.969621	1.386235	1.418307	1.418370	1.450688	-0.123450	-0.110408	-0.110164	-0.096970																																																																																																																																																																																																						
4	0.2500	2.330923	0.962730	1.450678	1.483212	1.483252	1.515971	-0.096969	-0.083612	-0.083355	-0.069814																																																																																																																																																																																																						
5	0.3125	2.423627	0.957514	1.515958	1.548825	1.548832	1.581807	-0.069814	-0.056078	-0.055809	-0.041854																																																																																																																																																																																																						
.																																																																																																																																																																																																				
475	29.6875	3.842767	3.392046	-3.209585	-3.209794	-3.203474	-3.193199	1.196816	1.106820	1.105906	1.012914																																																																																																																																																																																																						
476	29.7500	3.642462	3.461163	-3.193337	-3.173153	-3.167386	-3.138022	1.013215	0.917918	0.917796	0.820669																																																																																																																																																																																																						
477	29.8125	3.444415	3.518510	-3.138227	-3.100385	-3.095311	-3.049770	0.820955	0.722668	0.723298	0.624329																																																																																																																																																																																																						
478	29.8750	3.250880	3.563689	-3.050019	-2.997550	-2.993245	-2.934624	0.624586	0.525549	0.526856	0.428203																																																																																																																																																																																																						
479	29.9375	3.063732	3.596581	-2.934897	-2.870912	-2.867394	-2.798778	0.428423	0.330691	0.332575	0.236170																																																																																																																																																																																																						
480	30.0000	2.884458	3.617321																																																																																																																																																																																																														
>>																																																																																																																																																																																																																	

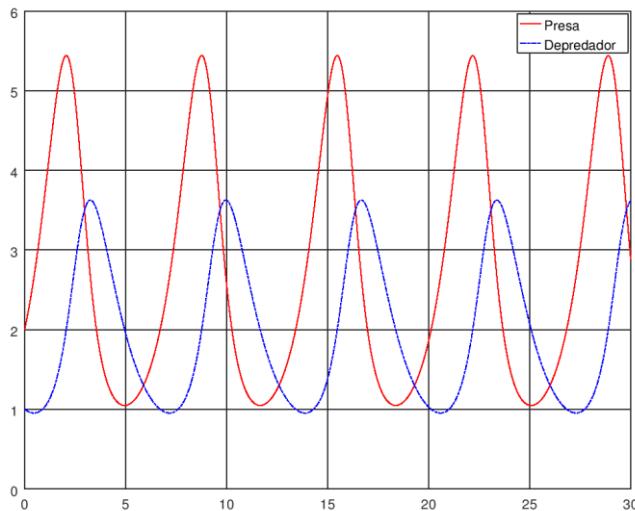


Figura A.6. Gráfica de los resultados arrojados por el código de la figura A.4.

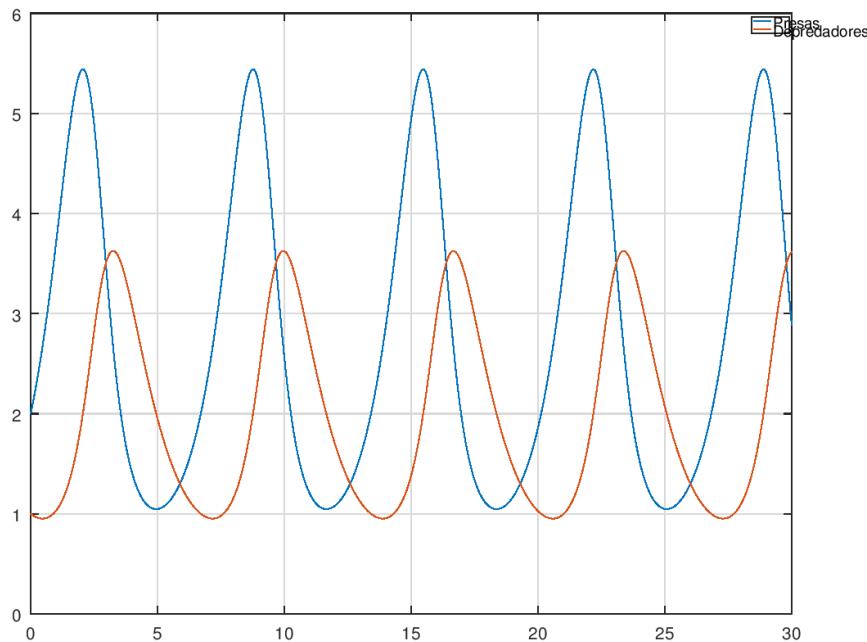
Lo primero que se observa en los resultados obtenidos hasta ahora, es que el método de Euler diverge, es decir, tanto las presas como los depredadores crecen de manera oscilatoria; el cálculo es inestable. Esto se puede resolver disminuyendo el incremento de tiempo.

Por otro lado, el método de *rk4* arroja resultados apropiados; mostrando que el sistema se estabiliza con el tiempo, esto se debe al orden mayor que el método de Euler.

También, debido a que la población de depredadores es inicialmente pequeña, la presa crece exponencialmente. En cierto punto, las presas llegan a ser tan numerosas que la población depredadora comienza a crecer. Eventualmente, el aumento de los depredadores causa que las presas disminuyan. Esta disminución, a su vez, conduce a una disminución de los depredadores. Eventualmente, el proceso se repite. Se observa que, como era de esperarse, el pico de los depredadores-presa. También, se advierte que el proceso tiene un período fijo, esto es, se repite en un tiempo determinado (*rk4*).

Otra alternativa para resolver este problema, es usar comandos de Octave,

```
>> function fun=fcn(y,t)
fun(1) = 1.2*y(1) - 0.6*y(1)*y(2);
fun(2) = -0.8*y(2) + 0.3*y(1)*y(2);
endfunction
>> t=[0:0.0625:30];
>> y0=[2, 1];
>> y=lsode("fcn",y0,t);
>> for i=1:length(t)
fprintf('%8.4f %8.6f %8.6f',t(i),y(i,1),y(i,2));
fprintf('\n')
endfor
0.0000 2.000000 1.000000
0.0625 2.076890 0.988284
0.1250 2.157621 0.978150
0.1875 2.242274 0.969621
0.2500 2.330922 0.962730
0.3125 2.423626 0.957514
.
.
.
29.6250 4.043893 3.311768
29.6875 3.843864 3.392043
29.7500 3.643495 3.461233
29.8125 3.445376 3.518651
29.8750 3.251761 3.563897
29.9375 3.064531 3.596851
30.0000 2.885174 3.617647
>> plot(t,y(:,1))
>> hold on
>> plot(t,y(:,2))
>> grid
>> legend('Presas','Depredadores')
>>
```



Dinámica de sistemas. Perspectiva ecosistémica.

En la figura A.7 se muestran algunos símbolos propuestos por Odum⁴ para representar la dinámica de un ecosistema y que posibilitan para representar el ejemplo actual.

En la figura A.8 se aprecia el ecosistema propuesto para el presente ejemplo, usando algunos de los símbolos propuestos por Odum⁴.

Se inicia con un sistema cerrado que recibe energía externa, E (un sistema cerrado solamente puede entrar y/o salir energía). Dentro del sistema se encuentran los símbolos de productor y de consumidor, los cuales incluyen los depósitos de cada uno de ellos, respectivamente. También se observan las diferentes interacciones como procesos de transferencia de energía y de biomasa entre la fuente de energía, la presa y el depredador.

Finalmente, se muestran las relaciones que complementan la estructura del ecosistema en estudio, las cuales están representadas como ecuaciones que reflejan la interacción entre las partes del ecosistema.

Resolviendo ahora este ejemplo desde la perspectiva de dinámica de sistemas; se pueden establecer las siguientes **variables** involucradas (incluyendo su definición, tipo y valor):

$$E = \text{energía constante entrante (auxiliar)} = 1200 \text{ J/mes}$$

$$x = \text{inventario de presas (biomasa) (nivel o estado)}, x_0=2 \text{ presa}$$

⁴ Odum, H.T. 1988. Ecosistemas y Políticas Públicas. Ecological Economics Program. University of Florida.

a = tasa de crecimiento de la presa (constante) = $K_I * E = 1.2 \text{ 1/mes}$

K_I = coef. de aprovechamiento de E por la presa (constante = 0.001 1/J)

$f_I(t)$ = flujo de entrada de presas = $a * P(t) = K_I * E * x(t)$ presa/mes

y = inventario de depredadores (biomasa) (nivel o estado) = 1 depredador

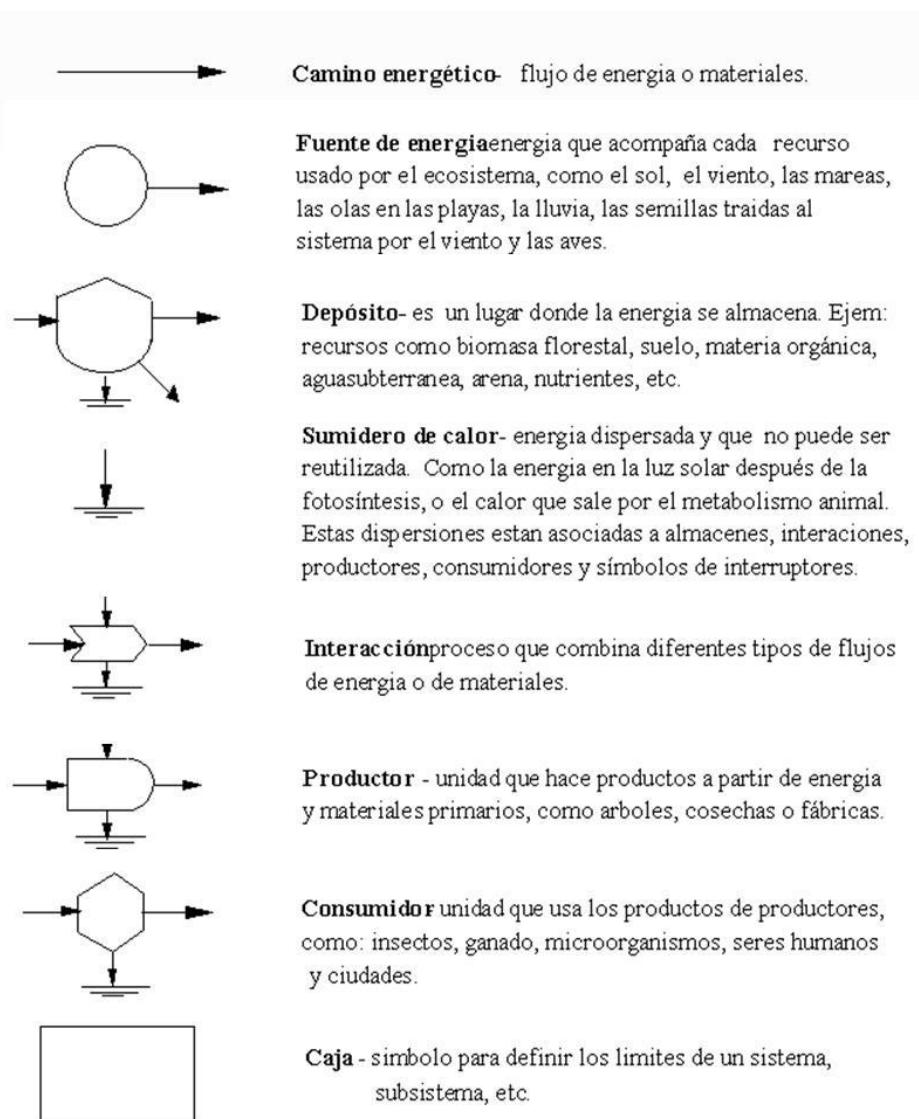


Figura A.7. Simbología de Odum para dinámica de ecosistemas.

c = tasa de crecimiento del depredador (constante) = 0.8 1/mes

$f_2(t)$ = Flujo de entrada de depredadores = $-c * y(t)$ depredador/mes

b = tasa de proporcionalidad de xy sobre la presa (encuentros presa-depredador) (constante) = 0.6 1/mes

d = tasa de proporcionalidad de xy sobre el depredador (encuentros presa-depredador) (constante) = 0.3 1/mes

$f_3(t)$ = decremento de presas por presencia de depredadores = $-bx(t)y(t)$ presa/mes

$f_4(t)$ = incremento de depredadores por presencia de presas = $y(t)x(t)$ depredador/mes

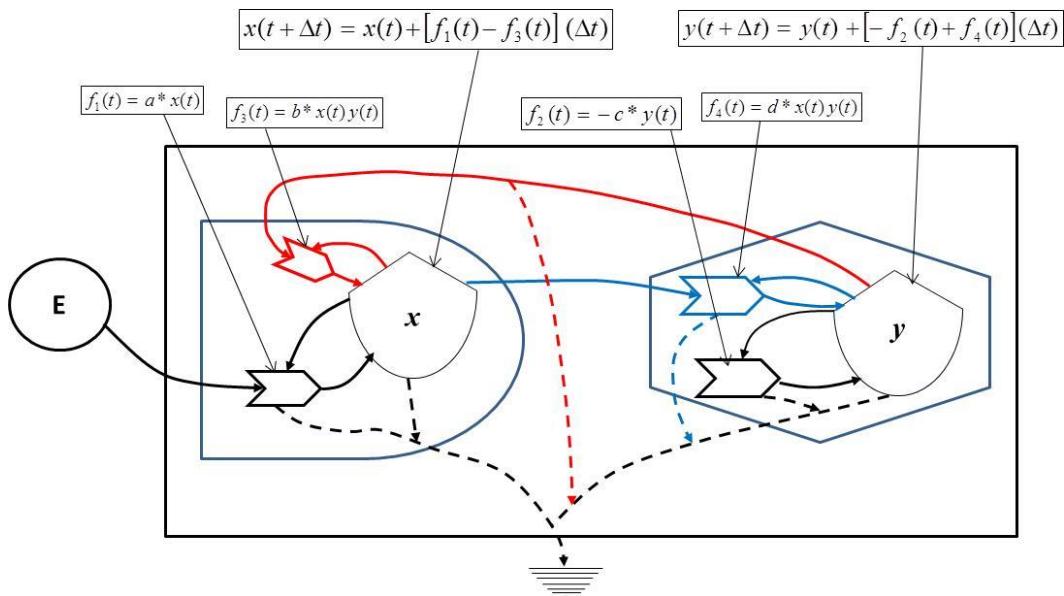


Figura A.9. Ecosistema del ejemplo actual representado con simbología de Odum.

Estas variables se relacionan dinámicamente entre ellas, influenciándose positiva o negativamente, estableciéndose sus **relaciones de influencia**. Esto se muestra en la figura A.9. Siendo una influencia positiva (+) si se aprecia que cuando una variable se incrementa (o decremente) la otra variable con la que se relaciona también se incrementa (o decremente). Y es una influencia negativa cuando una variable se incrementa (o se decremente) la otra variable con la que se relaciona se decremente (o se incremente).

Con base en las relaciones de influencia se construye un **diagrama de influencias** que permite visualizar esquemáticamente las influencias que se ejercen entre las diferentes partes de la estructura del ecosistema en estudio, figura A.10.

Para complementar la esquematización del ejemplo, se elabora un **diagrama de Forrester**, el cual contiene simbología utilizada a nivel mundial para esquematizar problemas de diversas áreas del conocimiento. Ver figura A.11.

- o R1: $E \xrightarrow{+} f_1(t)$
- o R2: $K_1 \xrightarrow{+} f_1(t)$
- o R3: $f_1(t) \xrightarrow{+} x(t)$
- o R4: $x(t) \xrightarrow{+} f_1(t)$
- o R5: $x(t) \xrightarrow{+} f_2(t)$
- o R6: $f_2(t) \xrightarrow{-} x(t)$
- o R7: $K_2 \xrightarrow{+} f_2(t)$
- o R8: $x(t) \xrightarrow{+} f_3(t)$
- o R9: $K_3 \xrightarrow{+} f_3(t)$
- o R10: $f_3(t) \xrightarrow{+} y(t)$
- o R11: $K_4 \xrightarrow{+} f_4(t)$
- o R12: $f_4(t) \xrightarrow{-} y(t)$
- o R13: $y(t) \xrightarrow{+} f_4(t)$
- o R14: $y(t) \xrightarrow{+} f_2(t)$

Figura A.9. Relaciones de influencia.

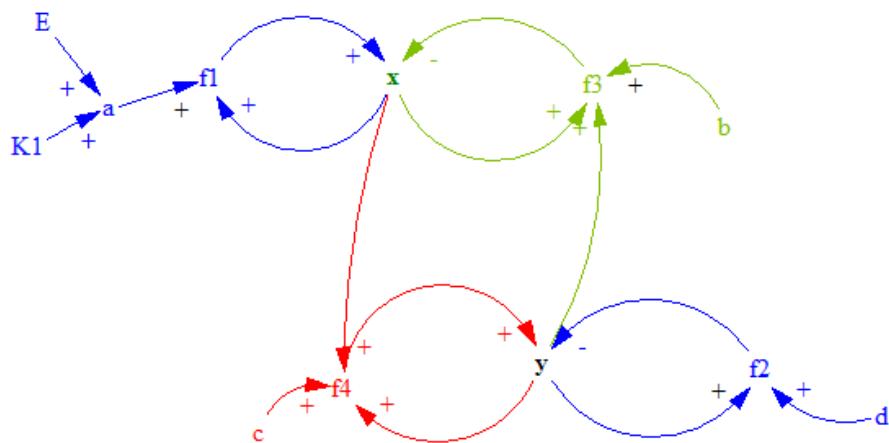


Figura A.10. Diagrama de influencias.

Uno de los paquetes de software más populares para conocer la dinámica de un sistema planteado es Vensim, además de Stella por supuesto. De hecho, las figuras A.10 y A.11 son producto de dicho paquete. Entonces, resolviendo este ecosistema con Vensim aplicando *Euler* se obtiene lo mostrado en la Tabla A.1 y figura A.12.

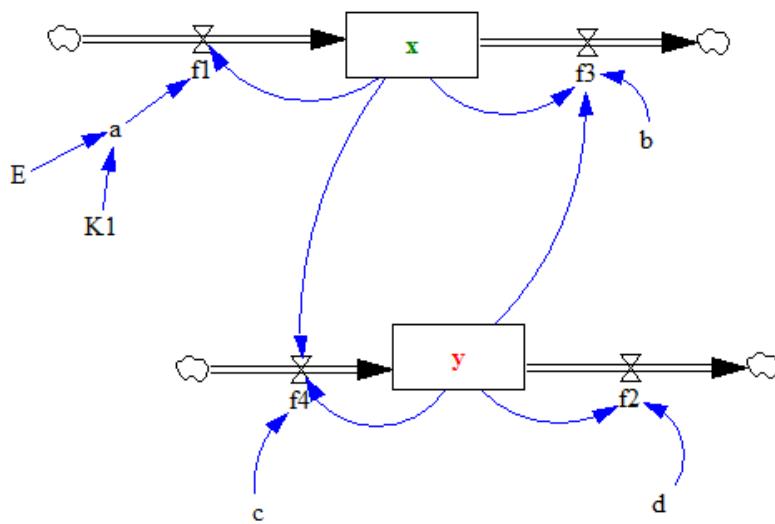


Figura A.11. Diagrama de Forrester.

Time (mes)	x	y
0.0000	2.0000	1.0000
0.0625	2.0750	0.9875
0.1250	2.1538	0.9765
0.1875	2.2364	0.9672
0.2500	2.3231	0.9594
0.3125	2.4137	0.9532
...
29.6875	2.5588	0.2746
29.7500	2.7244	0.2740
29.8125	2.9007	0.2743
29.8750	3.0884	0.2755
29.9375	3.2882	0.2777
30.0000	3.5005	0.2809

Tabla A.1. Resultados del modelo presa y depredador, Euler.

Modelo presa-depredador.

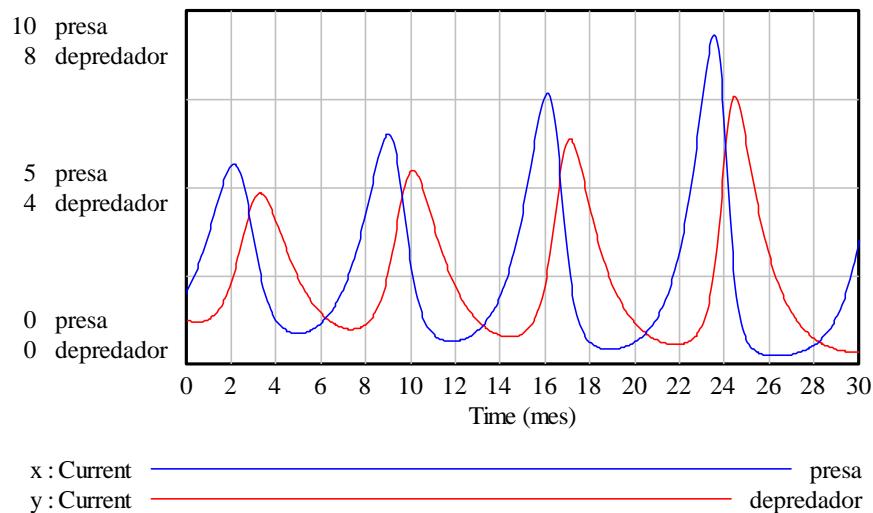


Figura A.12. Gráfica de tiempo entre presa vs depredador (*Euler*).

Como se aprecia, el modelo también es inestable, por lo que a continuación se resuelve con *rk4*.

Tabla A.2 y figura A.13.

Time (mes)	x	y
0.0000	2.0000	1.0000
0.0625	2.0769	0.9883
0.1250	2.1576	0.9781
0.1875	2.2423	0.9696
0.2500	2.3309	0.9627
0.3125	2.4236	0.9575
...
29.6875	3.8438	3.3920
29.7500	3.6435	3.4612
29.8125	3.4454	3.5187
29.8750	3.2517	3.5639
29.9375	3.0645	3.5968
30.0000	2.8852	3.6176

Tabla A.2. Resultados del modelo presa y depredador, *rk4*.

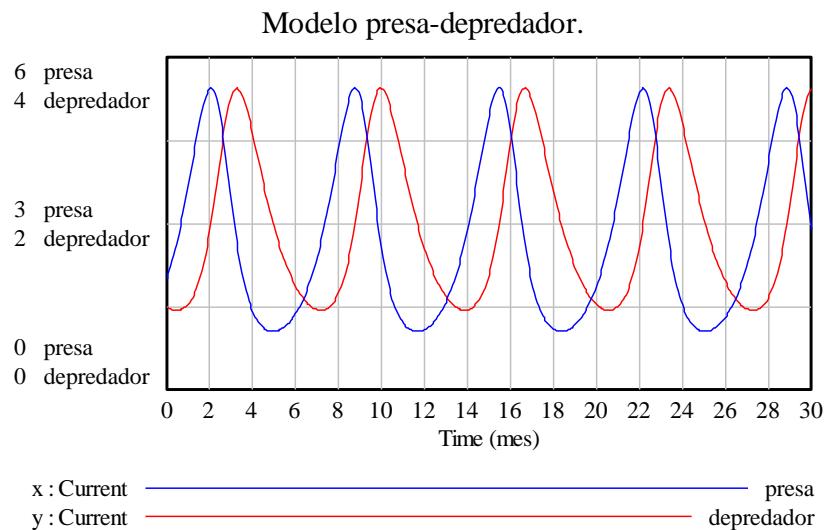
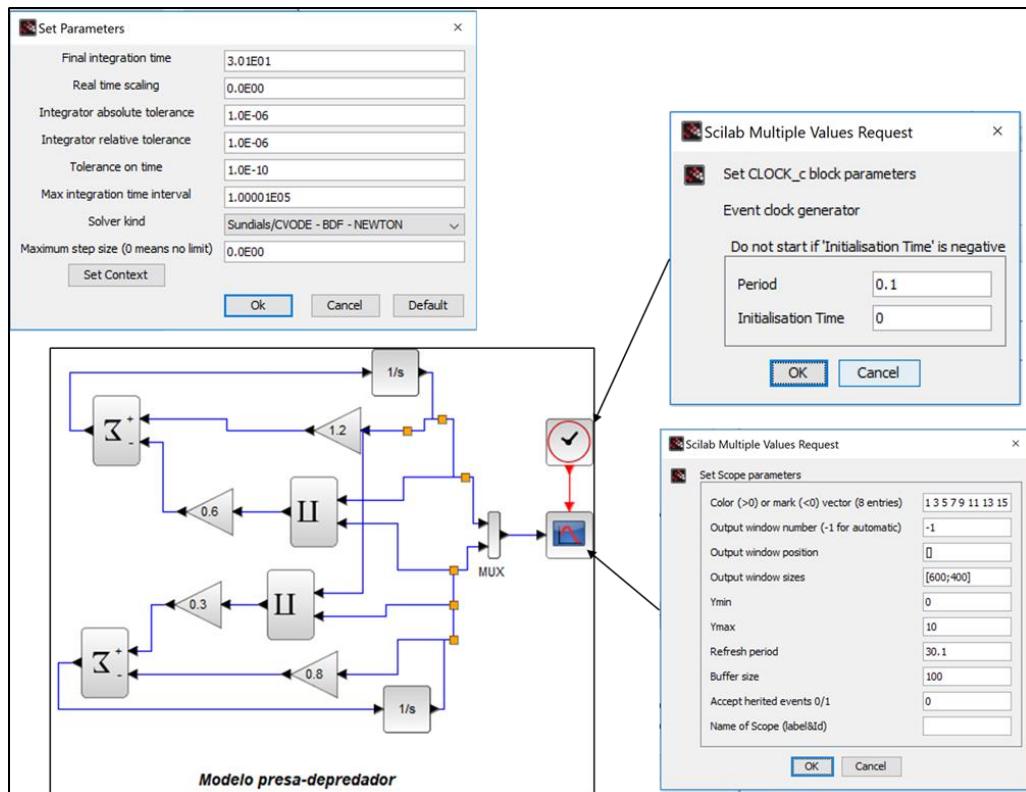
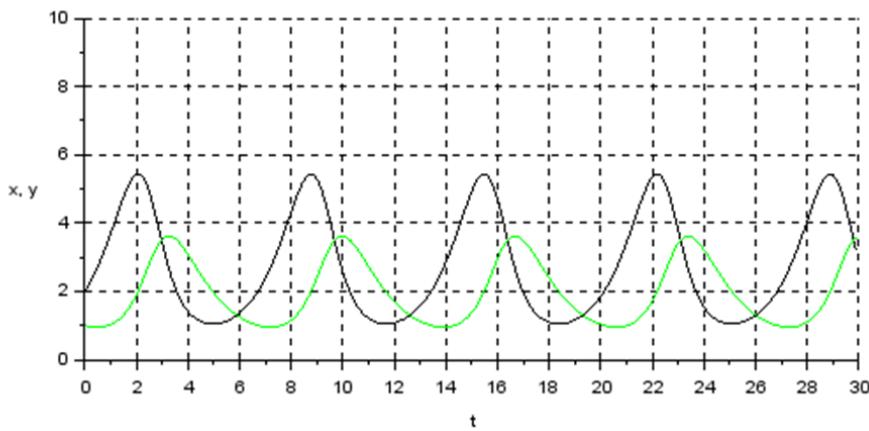


Figura A.13. Gráfica de tiempo entre presa vs depredador (rk4).

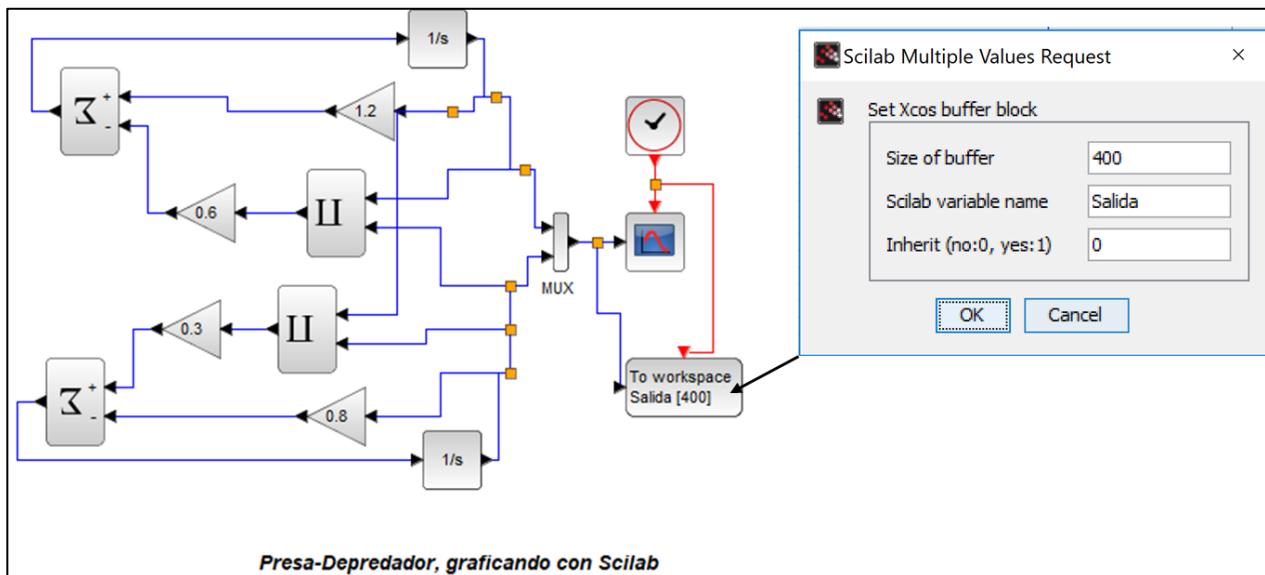
Se comprueba que con *rk4* el sistema se estabiliza, similar a los resultados con métodos numéricos.

➤ **Usando Xcos de Scilab:**

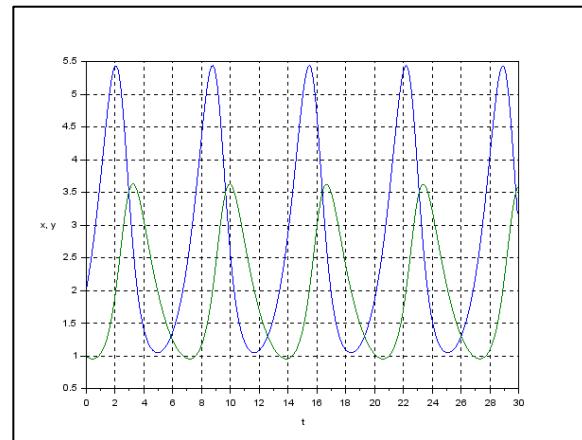




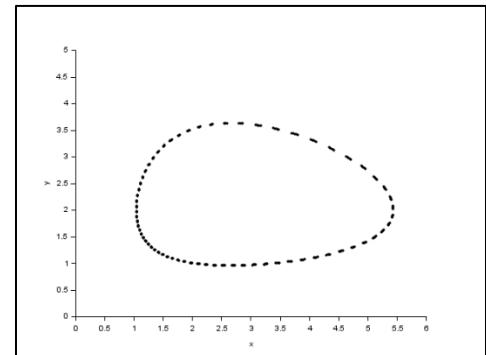
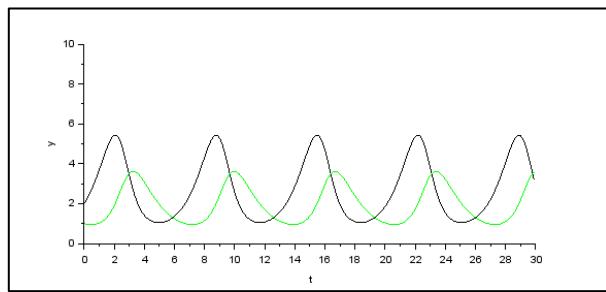
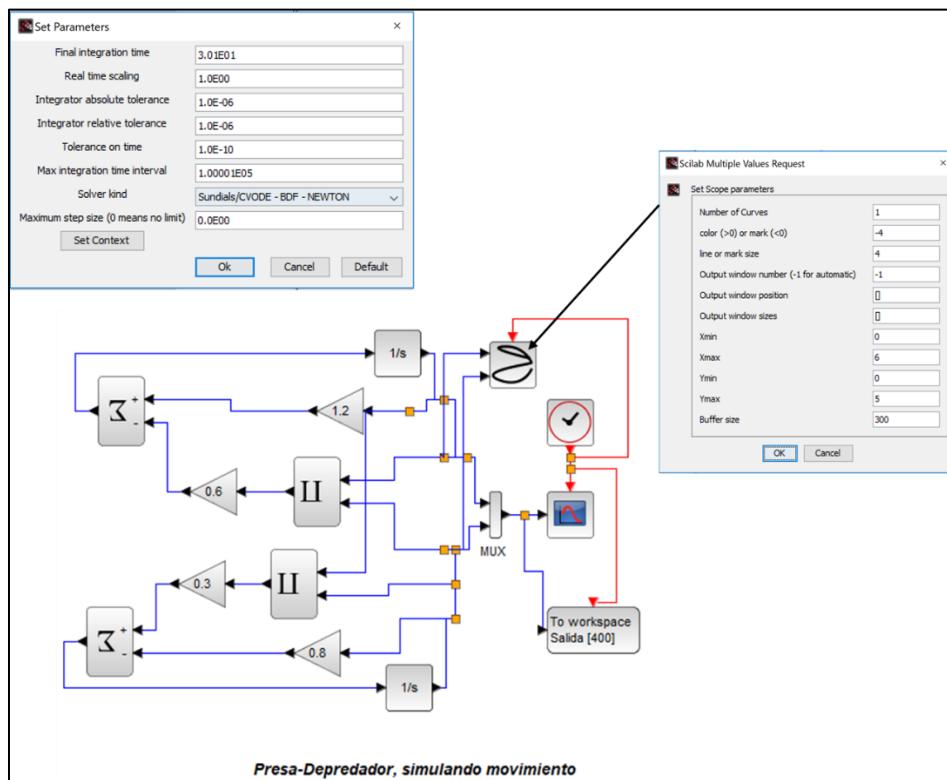
Graficando con Scilab:



```
Scilab 6.0.2 Console
--> Salida
Salida =
values: [301x2 constant]
time: [301x1 constant]
--> plot(Salida.time,Salida.values)
-->
```



Suponiendo movimiento:



Conclusiones.

Se resolvió el sistema de dos ecuaciones diferenciales que representan el comportamiento más simple en la relación dinámica presa vs depredador, aplicando Euler y Runke-Kutta de 4to orden mediante métodos numéricos.

Se representó el comportamiento dinámico del ejemplo usando simbología propia de dinámica de ecosistemas.

Se convirtió la representación anterior a una simbología más universal en cuanto a su aplicación a problemas de diversas áreas del conocimiento mediante diagramas de Forrester.

Se resolvió el mismo sistema de ecuaciones planteado inicialmente y resuelto por métodos numéricos, usando Vensim. Obteniendo resultados muy similares en ambos procedimientos.

Se propone una metodología para migrar de la simbología usada para dinámica de ecosistemas, a la simbología de dinámica de sistemas, ofreciendo así una herramienta más general en cuanto a su aplicación en diversas áreas del conocimiento. Procurando así enriquecer la interdisciplinariedad.

Mediante la metodología aplicada de dinámica de sistemas, se pueden resolver una o más ecuaciones diferenciales ordinarias, lo cual redundará en una conceptualización más integral del comportamiento dinámico de los sistemas en cualquier área del conocimiento que se aplique.

2) MODELO DE CRECIMIENTO POBLACIONAL.

Se desea estudiar el crecimiento poblacional de un hato de cerdos de la Sociedad Ganadera de la Región sur de Sonora, para posteriormente enviarlos a Granjas Ojai quien se dedica al procesamiento y distribución a mercados de exportación (Japón y China recientemente). Determine, empleando la metodología de DS⁵.

¿Cuántos puercos habría durante los próximos 60 meses?

Hipótesis dinámicas (supuestos):

1. La población actual en noviembre de 2016 es de 100 hembras.
2. Existe un proceso de inseminación artificial que permite tener una tasa de nacimiento del 12% mensual (TN).
3. La tasa de mortandad es del 5% mensual. (TM)

Las ecuaciones diferenciales que simulan el comportamiento dinámico de este proyecto son:

$$\frac{dNac}{dt} = TN * Pob(t) , \quad \frac{dMu}{dt} = TM * Pob(t)$$

$$\frac{dPob}{dt} = \frac{dNac}{dt} - \frac{dMu}{dt} = TN * Pob(t) - TM * Pob(t) = (TN - TM)Pob(t) = 0.07 * Pob(t)$$

Resolviendo con comandos de Octave se obtiene lo siguiente:

```
>> pkg load symbolic  
>> warning('off','all')  
>> syms t Pob(t) TN TM
```

⁵ Lagarda, L.E.A. Taller: Simulación de la cadena de suministros con dinámica de sistemas. Desarrollo de modelos con software Stella. 2017. Documento técnico interno ITSON.

```
>> DE=diff(Pob,t)-(TN-TM)*Pob==0
DE = (sym)

d
- (-TM + TN)*Pob(t) + --(Pob(t)) = 0
dt

>> sol=dsolve(DE,Pob(0)==100)
sol =
{
(sym)

-TN*t t*(-TM + TN)
Pob(t) = Pob(t)*e *e

}
>>
```

Entonces, considerando las condiciones iniciales, $t=0$, $Pob(0) = 100$, la función solución queda:

$$Pob(t) = 100 e^{(TN-TM)t}.$$

Y para $TN=0.12$, $TM=0.05$ y 60 meses, el resultado es:

$$Pob(60) = 100 e^{(0.12-0.05)(60)} = 6,669 \text{ vientos}$$

❖ Métodos numéricos.

Resolviendo con el código en Octave, considerando $t=x$ y $Pob=y$, se obtiene lo siguiente:

Método de Euler.

Introduce una función con respecto a x 0.07*y

Introduce la solución analítica (Oprime Enter si no existe) 100*exp(0.07*x)

Datos:

x0 = 0

xn = 60

y0 = 100

Incremento de x = 1

i	x(i)	y(i)	y(i)verd.	e
0	0.000000	100.000000	100.000000	0.00
1	1.000000	107.000000	107.250818	0.23
2	2.000000	114.490000	115.027380	0.47
3	3.000000	122.504300	123.367806	0.70
4	4.000000	131.079601	132.312981	0.93
5	5.000000	140.255173	141.906755	1.16
55	55.000000	4131.500148	4699.306323	12.08
56	56.000000	4420.705159	5040.044478	12.29
57	57.000000	4730.154520	5405.488936	12.49
58	58.000000	5061.265336	5797.431108	12.70
59	59.000000	5415.553910	6217.792293	12.90
60	60.000000	5794.642683	6668.633104	13.11

>>

En la figura A.14 se muestran los resultados anteriores en forma gráfica.

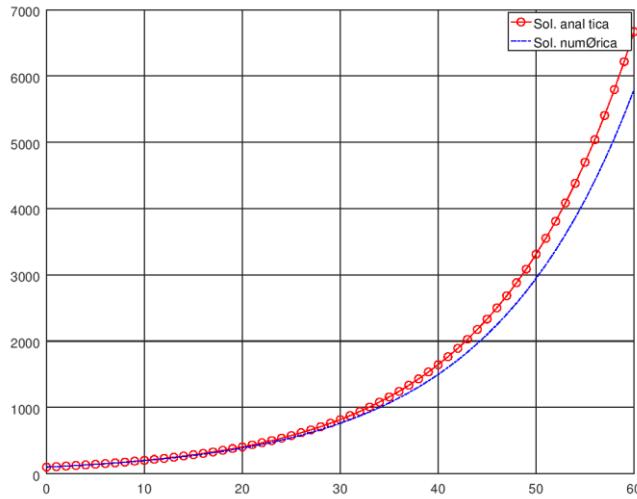


Figura A.14. Comportamiento gráfico de los resultados obtenidos con el método de Euler.

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x 0.07*y

Introduce la solución analítica (Oprime Enter si no existe) 100*exp(0.07*x)

Datos:

x0 = 0

xn = 60

y0 = 100

Incremento de x = 1

i	x(i)	y(i)	k1	k2	k3	k4
0	0.00000000	100.00000000	7.00000000	7.24500000	7.25357500	7.50775025
1	1.00000000	107.25081671	7.50755717	7.77032167	7.77951843	8.05212346
2	2.00000000	115.02737685	8.05191638	8.33373345	8.34359705	8.63596817
3	3.00000000	123.36780111	8.63574608	8.93799719	8.94857598	9.26214640
4	4.00000000	132.31297424	9.26190820	9.58607498	9.59742082	9.93372765
5	5.00000000	141.90674548	9.93347218	10.28114371	10.29331221	10.65400404
54	54.00000000	4381.60104731	306.71207331	317.44699588	317.82271817	328.95966358
55	55.00000000	4699.30290814	328.95120357	340.46449569	340.86746092	352.81192583
56	56.00000000	5040.04074858	352.80285240	365.15095223	365.58313573	378.39367190
57	57.00000000	5405.48486529	378.38394057	391.62737849	392.09089882	405.83030349
58	58.00000000	5797.42666507	405.81986655	420.02356188	420.52069122	435.25631494
59	59.00000000	6217.78744635	435.24512124	450.47870049	451.01187576	466.81595255
60	60.00000000	6668.62781740				

>>

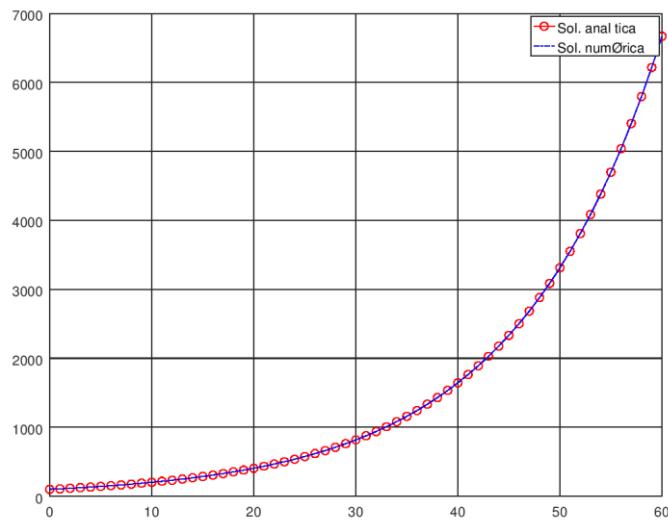


Figura A.14. Comportamiento gráfico de los resultados obtenidos con el método de rk4.

Rk4 es más exacto, como se esperaba.

❖ **Dinámica de sistemas.**

- Objetivo: Determinar el número de puercos que habría dentro de 60 meses.
- Variables:
 - *Población de puercos, Pob* (nivel o estado)
 - *Nacimientos, Nac* (auxiliar)
 - *Muertes, MU* (auxiliar)
 - *Tasa de natalidad, TN* (constante)
 - *Tasa de mortalidad, TM* (constante)
- Enunciados:
 - R1: Si aumenta *Pob*, aumenta *Nac*
 - R2: Si disminuye *Pob*, disminuye *Nac*
 - R3: Si aumenta *TN*, aumenta *Nac*
 - R4: Si aumenta *Pob*, aumenta *MU*
 - R5: Si aumenta *MU*, disminuye *Pob*

- R6: Si aumenta TM , aumenta Mu

- Relaciones de influencia:

$$\begin{array}{l} R1: Pob \xrightarrow{+} Nac \\ R2: Nac \xrightarrow{+} Pob \\ R3: TN \xrightarrow{+} Nac \\ R4: Pob \xrightarrow{+} Mu \\ R5: Mu \xrightarrow{-} Pob \\ R6: TM \xrightarrow{+} Mu \end{array}$$

- Diagrama de influencias:

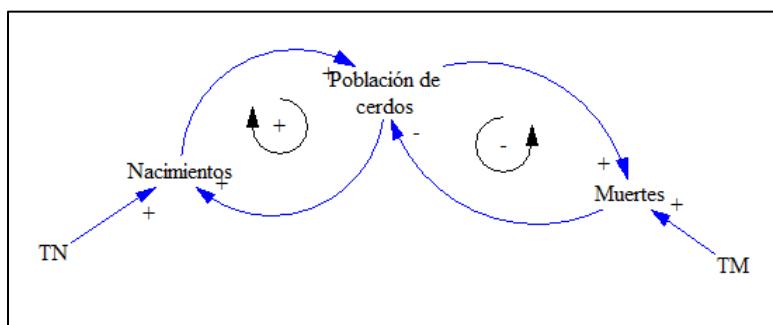


Figura A.15. Diagrama de influencias del ejemplo (2).

- Diagrama de Forrester:

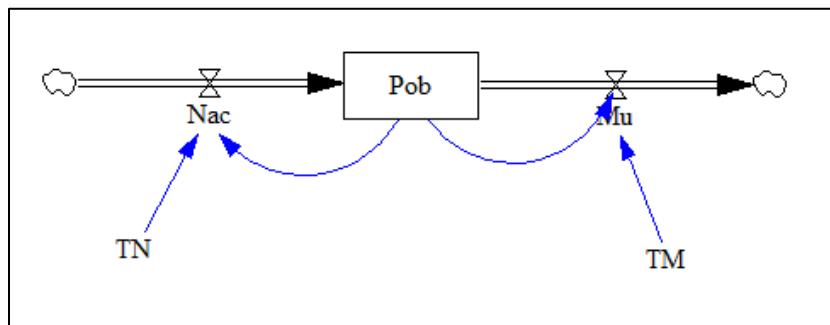


Figura A.16. Diagrama de Forrester del ejemplo (2).

- Ecuaciones:

$$Nac(t) = TN * Pob(t)$$

$$Mu(t) = TM * Pob(t)$$

$$Pob(t+dt) = Pob(t) + Nac(t) - Mu(t)$$

- Resolviendo con Vensim (Euler):

Time (Month)	"Población de cerdos"	Runs: Población de cerdos
0	Current	100
1		107
2		114.49
3		122.504
4		131.08
5		140.255
55		4131.5
56		4420.7
57		4730.15
58		5061.26
59		5415.55
60		5794.64

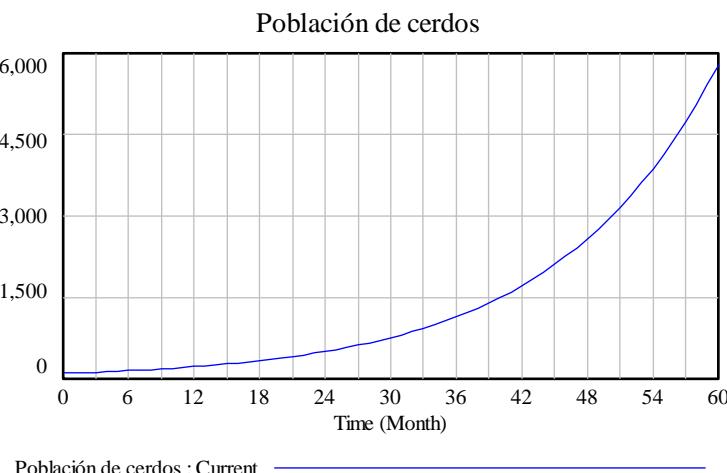


Figura A.17. Resultado gráfico del ejemplo (2).

- Resolviendo con Vensim (rk4):

Time (Month)	"Población de cerdos"	Runs: Población de cerdos
0	Current	100
1		107.251
2		115.027
3		123.368
4		132.313
5		141.907
55		4699.31
56		5040.05
57		5405.49
58		5797.43
59		6217.79
60		6668.63

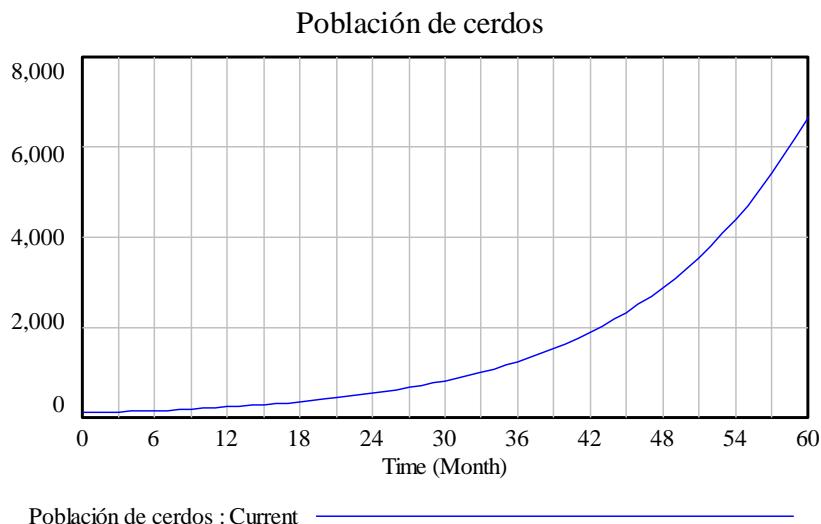


Figura A.19. Resultado gráfico del ejemplo (2), usando Vensim.

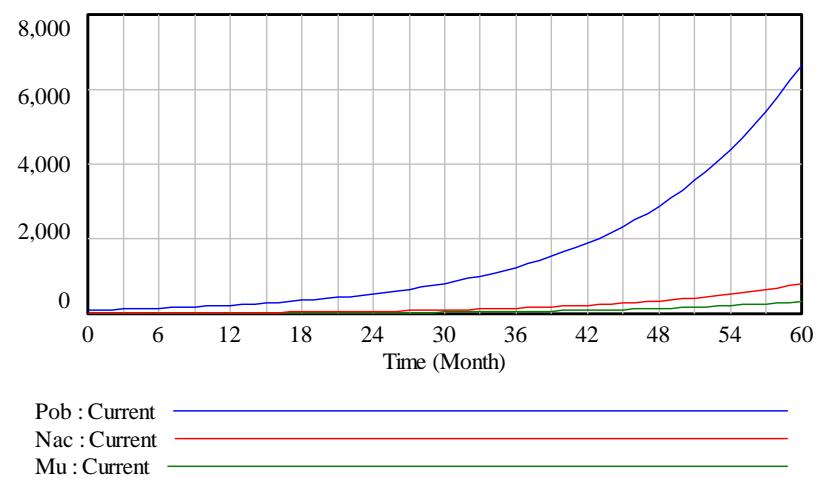
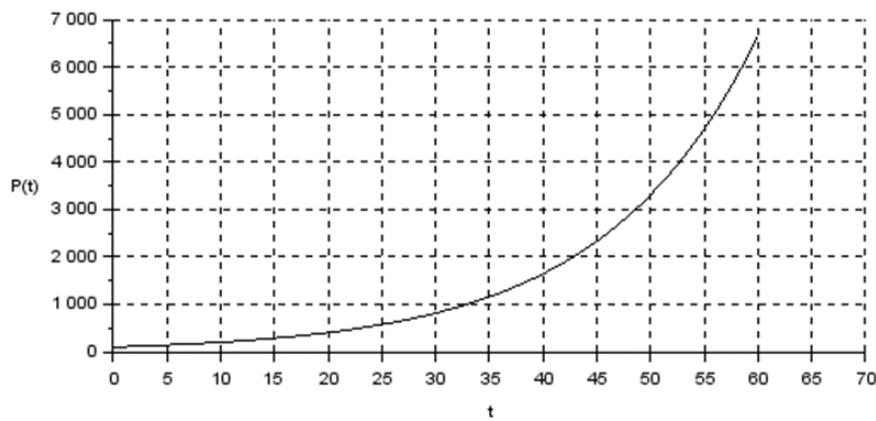
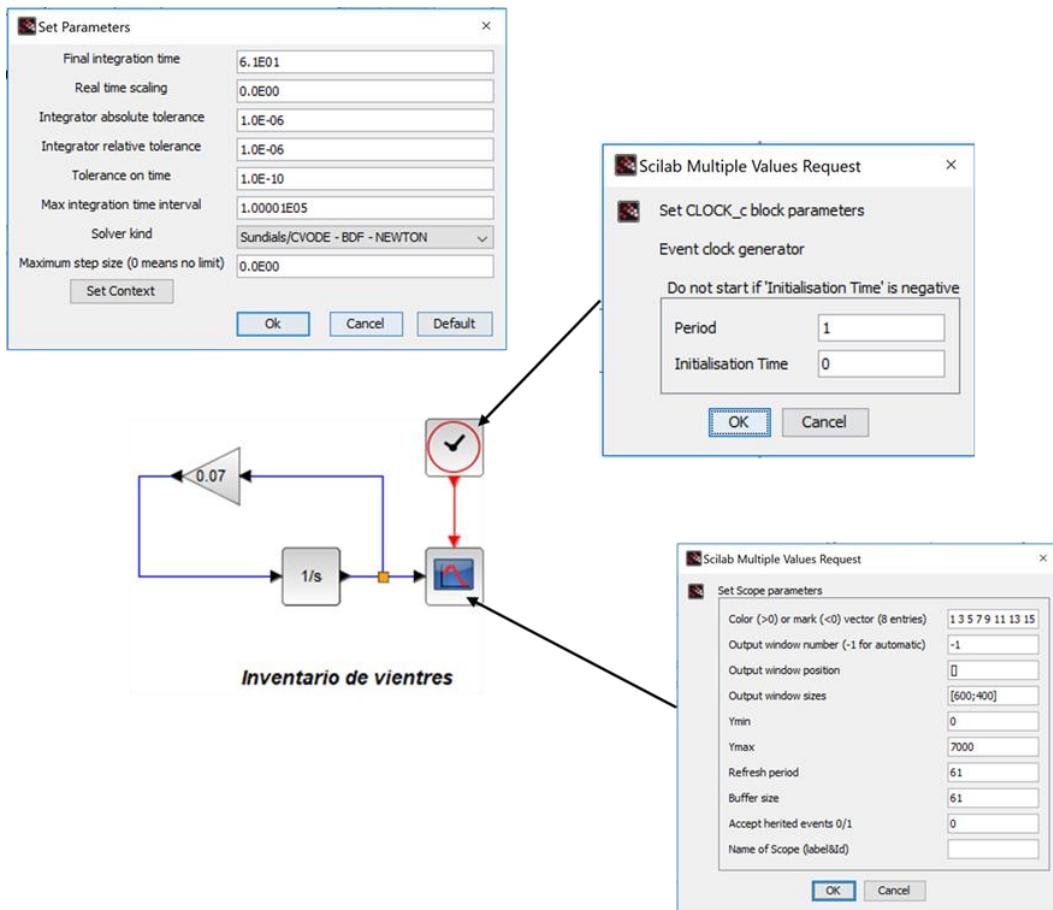


Figura A.19. Resultado gráfico del ejemplo (2), usando Vensim. Cont.

➤ **Usando Xcos de Scilab:**



3)². MODELO DE CRECIMIENTO POBLACIONAL.

El crecimiento poblacional de los organismos tiene muchas aplicaciones ingenieriles y científicas. Uno de los modelos más simples supone que la tasa de cambio de la población, p , es proporcional a la población existente en cualquier tiempo t :

$$\frac{dp}{dt} = k_g p$$

donde k_g = tasa de crecimiento. La población mundial en millones desde 1950 hasta el 2000 fue

<i>t</i>	1950	1955	1960	1965	1970	1975	1980	1985	1990	1995	2000
<i>p</i>	2560	2780	3040	3350	3710	4090	4450	4850	5280	5690	6080

- (a) Usar los datos para estimar el valor de k_g .
- (b) Usar rk4 con el resultado de (a) para simular la población mundial desde 1950 hasta el 2050 con intervalos de 5 años. Desplegar los resultados por medio de una gráfica.

- (a) Resolviendo la ecuación diferencial, propuesta para la simulación, se obtiene lo siguiente con comandos de Octave:

```
>> pkg load symbolic
>> warning('off','all')
>> syms y(x) k
>> DE=diff(y,x)-k*y==0
DE = (sym)

d
-k*y(x) + --(y(x)) = 0
dx

>> sol=dsolve(DE)
sol = (sym)

k*x
y(x) = C1*e

>>
```

Entonces, para $t=0$, $p=p_0$ y $p(0) = p_0 = C_1 e^{k_g(0)}$ $\therefore C_1 = p_0 \rightarrow p(t) = p_0 e^{k_g t}$

Tomando el logaritmo de esta ecuación resulta lo siguiente:

$$\ln(p) = \ln(p_0) + k_g t$$

Graficando,

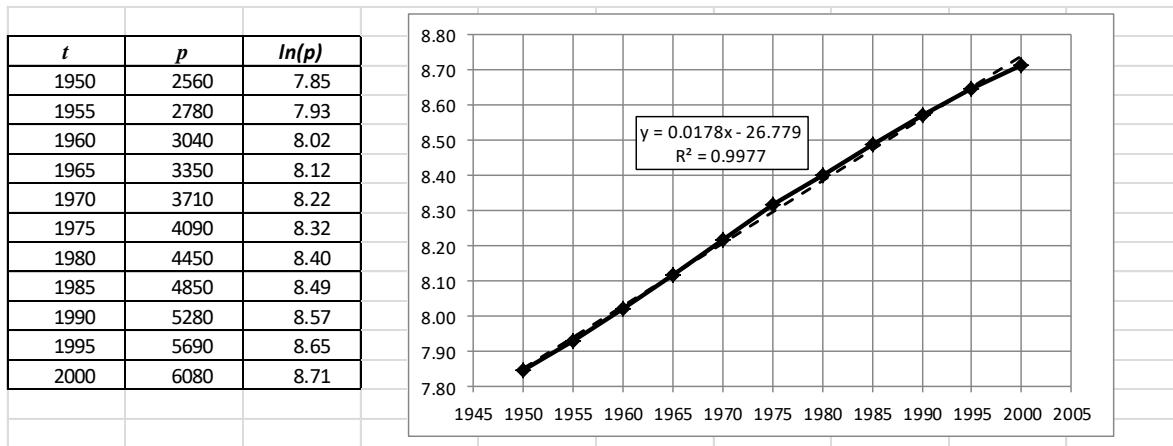


Figura A.20. Resultado para obtener k_g , usando la solución analítica.

Obteniéndose $k_g = 0.0178/\text{año}$.

(b) Aplicando rk4 con el código de Octave (se intercambia t por x y p por y):

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x $0.0178*y$

Introduce la solución analítica (Oprime Enter si no existe) $2560*\exp(0.0178*x)$

Datos:

```
x0 = 1950
xn = 2050
y0 = 2560
Incremento de x = 5
```

i	x(i)	y(i)	k1	k2	k3	k4
0	1950	2560	45.56800000	47.59577600	47.68601203	49.81205507
1	1955	2798	49.80949720	52.02601982	52.12465508	54.44859150
2	1960	3059	54.44579553	56.86863343	56.97644972	59.51669956
3	1965	3343	59.51364334	62.16200047	62.27985236	65.05655020
4	1970	3655	65.05320952	67.94807734	68.07689896	71.11205352
5	1975	3995	71.10840188	74.27272576	74.41353818	77.73120678
6	1980	4367	77.72721524	81.18607632	81.33999564	84.96647485
7	1985	4773	84.96211178	88.74292575	88.91117197	92.87520608
8	1990	5217	92.87043689	97.00317133	97.18707802	101.52008683
9	1995	5703	101.51487372	106.03228560	106.23331043	110.96963835
10	2000	6234	110.96394000	115.90183533	116.12157167	121.29875988
11	2005	6814	121.29253112	126.69004876	126.93023829	132.58932233
12	2010	7448	132.58251380	138.48243566	138.74498219	144.93081721
13	2015	8142	144.92337494	151.37246512	151.65944964	158.42106596
14	2020	8900	158.41293095	165.46230638	165.77600358	173.16699527
15	2025	9728	173.15810305	180.86363864	181.20653497	189.28548466
16	2030	10633	189.27576475	197.69853628	198.07334962	206.90429287
17	2035	11623	206.89366822	216.10043646	216.51013765	226.16307047

```

18      2040      12705 226.15145688 236.21519671 236.66303313 247.21446683
19      2045      13888 247.20177223 258.20225110 258.69177241 270.22533998
20      2050      15180
>>

```

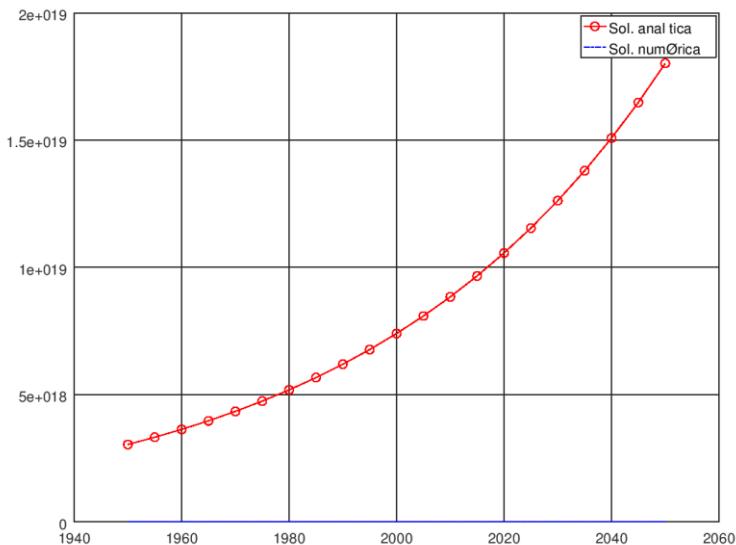


Figura A.21. Resultados en forma gráfica, usando código en Octave.

➤ **Dinámica de sistemas.**

Se ejecutará con Vensim. Debido a que el incremento de tiempo es de 5 años, el valor de $K_g = 5 * 0.0178 = 0.089/5$ años.

- Diagrama de influencias.

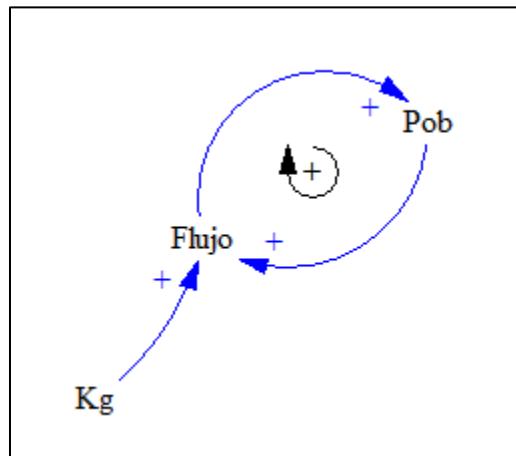


Figura A.22. Diagrama de influencias del ejemplo (3).

- Diagrama de Forrester.

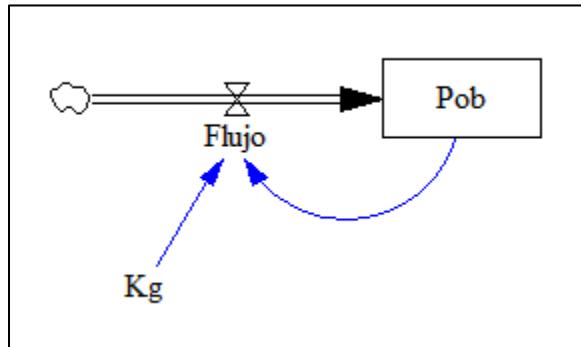


Figura A.23. Diagrama de Forrester del ejemplo (3).

- Ecuaciones:

$$Flujo(t) = Kg * Pob(t)$$

$$Pob(t+dt) = Pob(t) + Flujo(t)$$

- Simulación.

Time (Month)	"Pob"	Runs:	Pob
0	Current	2560	
1			2798.29
2			3058.75
3			3343.46
4			3654.68
5			3994.85
6			4366.7
7			4773.15
8			5217.44
9			5703.08
10			6233.93
11			6814.19
12			7448.46
13			8141.77
14			8899.61
15			9727.99
16			10633.5
17			11623.2

18	12705.1
19	13887.7
20	15180.4

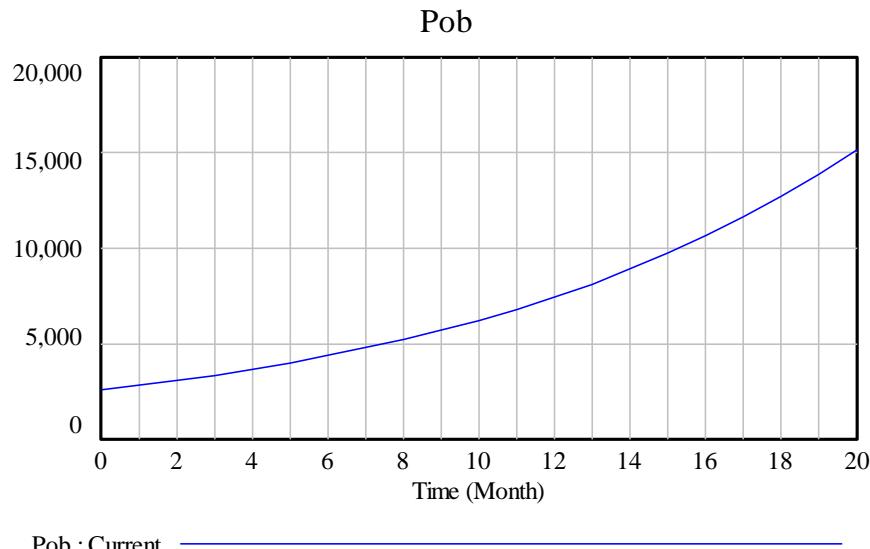
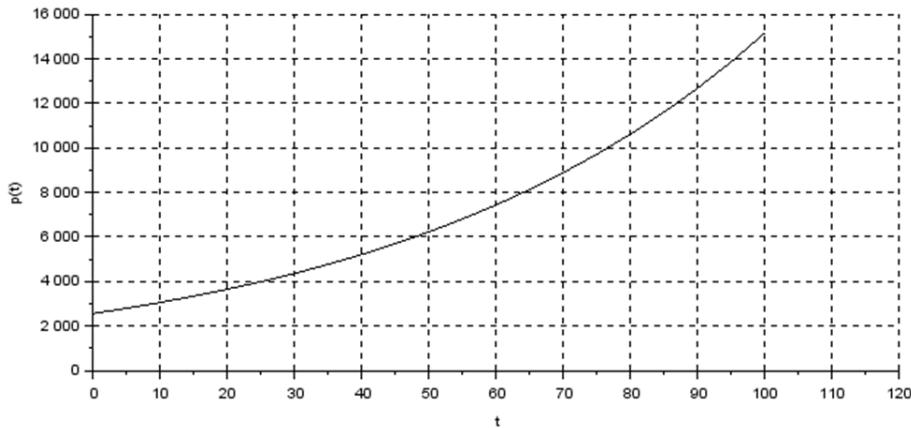
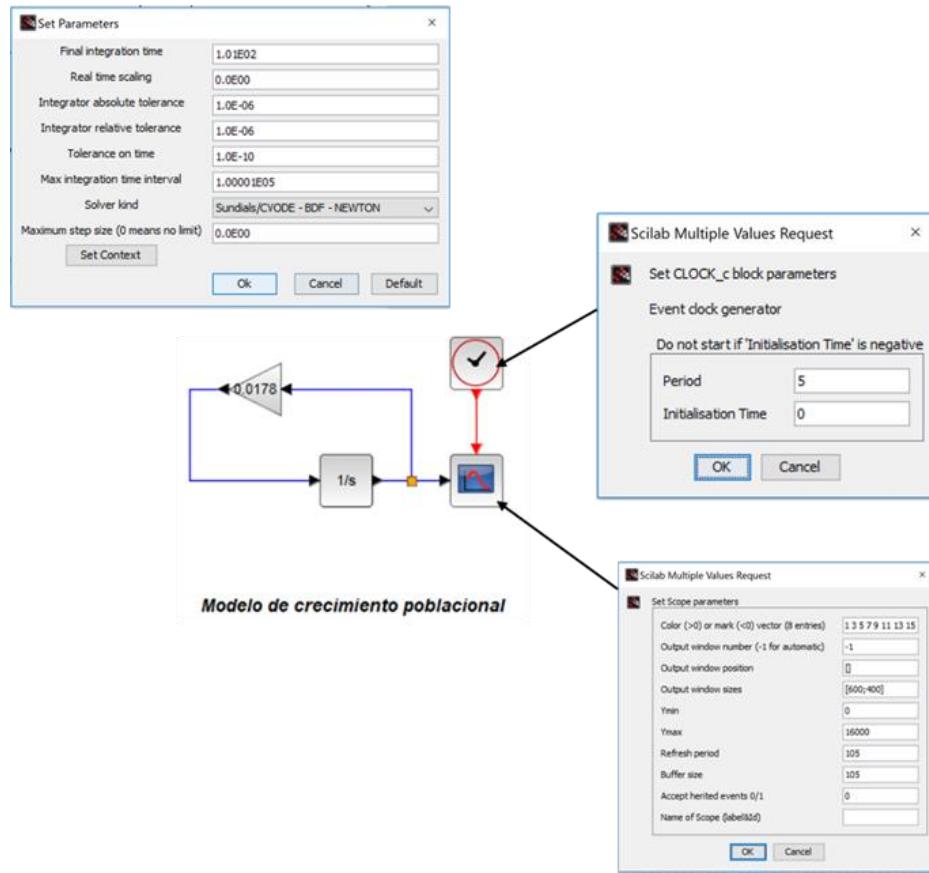


Figura A.24. Resultados gráficos del ejemplo (3) usando Vensim.

➤ **Usando Xcos de Scilab:**



4)². MODELO DE CRECIMIENTO POBLACIONAL.

Aunque el modelo del problema 22.4 funciona adecuadamente cuando la población crece ilimitadamente, no se cumple cuando existen factores como la escasez de alimento, contaminación y falta de espacio que inhibe el crecimiento. En tales casos, la tasa de crecimiento no es constante, pero puede ser formulada como

$$k_g = k_{gm} \left(1 - p / p_{máx}\right)$$

donde k_{gm} = tasa máxima de crecimiento bajo condiciones ilimitadas, p = población y $p_{máx}$ = población máxima. Notar que $p_{máx}$ es llamada algunas veces *capacidad de carga*. Entonces, para una densidad de población baja $p \ll p_{máx}$, $kg \rightarrow k_{gm}$. A medida que p se aproxima a $p_{máx}$, la tasa de crecimiento se aproxima a cero. Usando esta formulación de tasa de crecimiento, la tasa de cambio de población puede ser modelada como

$$\frac{dp}{dt} = k_{gm} (1 - p / p_{máx}) p$$

Esto se refiere a un modelo logístico. La solución analítica de este modelo es

$$p = p_0 \frac{p_{máx}}{p_0 + (p_{máx} - p_0) e^{-k_{gm} t}}$$

Simular al población mundial desde 1950 hasta 2050 usando (a) la solución analítica y (b) el método de rk4 con un intervalo de 5 años. Emplear las siguientes condiciones iniciales y parámetros para la simulación: p_0 (en 1950) = 2,560 millones de habitantes, k_{gm} = 0.026/año y $p_{máx}$ = 12,000 millones de habitantes. Despliegue los resultados en forma gráfica junto con los resultados del ejemplo 22.4

Método de Runge-Kutta de 4to orden.

```
Introduce una función con respecto a x  0.026*(1-y/12000)*y
```

```
Introduce la solución analítica (Oprime Enter si no existe)
12000/(exp(log(59/16) - (13*x)/500) + 1)
```

Datos:

```
x0 = 0
xn = 100
y0 = 2560
Incremento de x = 5
```

i	x(i)	y(i)	k1	k2	k3	k4	y(i).verd
0	0.000000	2560.000000	52.360533	54.274710	54.343281	56.250955	2560.000000
1	5.000000	2831.539557	56.248527	58.136412	58.198290	60.060373	2831.539737
2	10.000000	3122.354810	60.058176	61.881614	61.935448	63.712005	3122.355168
3	15.000000	3431.858398	63.710072	65.427619	65.472400	67.120965	3431.858929
4	20.000000	3759.050960	67.119319	68.687766	68.722959	70.200266	3759.051655
5	25.000000	4102.501824	70.198918	71.575211	71.600861	72.864910	4102.502668
6	30.000000	4460.348467	72.863859	74.007301	74.024099	75.036427	4460.349445
7	35.000000	4830.317705	75.035661	75.910235	75.919529	76.647494	4830.318798
8	40.000000	5209.769940	76.646996	77.223603	77.227338	77.646202	5209.771131
9	45.000000	5595.765840	77.645955	77.904341	77.904929	77.999531	5595.767111
10	50.000000	5985.152529	77.999522	77.929682	77.929818	77.695635	5985.153865
11	55.000000	6374.664327	77.695858	77.298755	77.301200	76.744676	6374.665716
12	60.000000	6761.031362	76.745134	76.032651	76.039999	75.178114	6761.032796
13	65.000000	7141.088485	75.178820	74.172940	74.187409	73.046521	7141.089958
14	70.000000	7511.876850	73.047495	71.778817	71.802084	70.416192	7511.878362
15	75.000000	7870.731424	70.417455	68.923209	68.956311	67.364922	7870.732976
16	80.000000	8215.349272	67.366493	65.688268	65.731568	63.977394	8215.350868
17	85.000000	8543.835571	63.979285	62.160699	62.213922	60.340619	8543.837217
18	90.000000	8854.726525	60.342829	58.427346	58.489665	56.539796	8854.728226

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

```

19    95.000000  9146.990397   56.542311   54.571355   54.641516   52.654866  9146.992158
20    100.000000  9420.009497
>>

```

Introduce una función con respecto a x $0.0178 * (1 - y / 12000) * y$

Introduce la solución analítica (Oprime Enter si no existe)
 $30.72e6 / (2560 + 9440 * \exp(-0.0178 * x))$

Datos:

```

x0 = 0
xn = 100
y0 = 2560
Incremento de x = 5

```

i	x(i)	y(i)	k1	k2	k3	k4	y(i).verd
0	0	2560	35.84682667	36.74948573	36.77190806	37.67303147	2560
1	5	2744	37.67247761	38.56911567	38.59013583	39.48116704	2744
2	10	2937	39.48064702	40.36317721	40.38257460	41.25511767	2937
3	15	3139	41.25463537	42.11437963	42.13196097	42.97709200	3139
4	20	3349	42.97665058	43.80446689	43.82007611	44.62848635	3349
5	25	3568	44.62808816	45.41452907	45.42805670	46.19022582	3568
6	30	3795	46.18987233	46.92537234	46.93676403	47.64316005	3795
7	35	4030	47.64285186	48.31793931	48.32720273	48.96850330	4030
8	40	4272	48.96824017	49.57376736	49.58097684	50.14830462	4272
9	45	4519	50.14808562	50.67546879	50.68076725	51.16593028	4519
10	50	4773	51.16575395	51.60721357	51.61081128	52.00653823	4773
11	55	5031	52.00640278	52.35519330	52.35736323	52.65752251	5031
12	60	5292	52.65742606	52.90804401	52.90911388	53.10890576	5292
13	65	5557	53.10884658	53.25720648	53.25754767	53.35365938	5557
14	70	5823	53.35363610	53.39720461	53.39721862	53.38793399	5823
15	75	6090	53.38794579	53.32582682	53.32592981	53.21118739	6090
16	80	6357	53.21123407	53.04419991	53.04480639	52.82620266	6357
17	85	6622	52.82628475	52.55675150	52.55825805	52.23899557	6622
18	90	6885	52.23911422	51.87106286	51.87383296	51.45861636	6885
19	95	7144	51.45877326	50.99762081	51.00197150	50.49685708	7144
20	100	7399					7399

>>

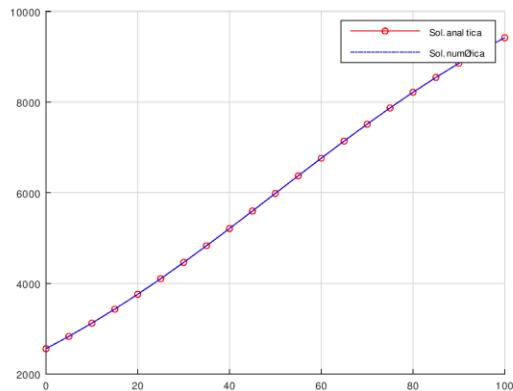


Figura A.25. Gráfica de resultados del ejemplo (4) aplicando rk4 con código en Octave.

➤ **Dinámica de sistemas.**

- Diagrama de influencias

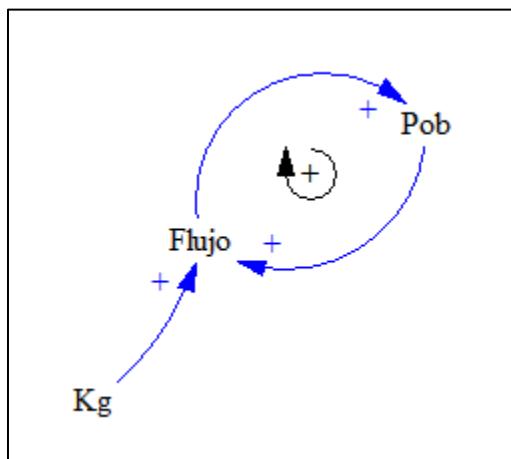


Figura A.26. Diagrama de influencias del ejemplo (4).

- Diagrama de Forrester.

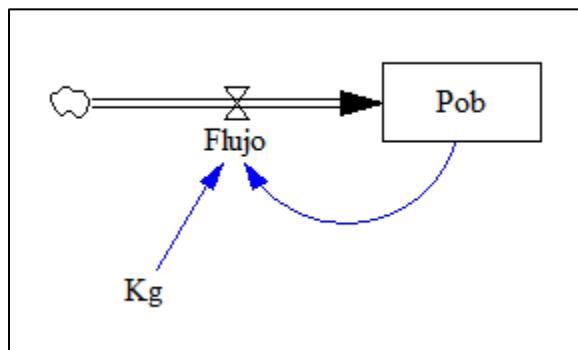


Figura A.27. Diagrama de Forrester del ejemplo (4).

- Ecuaciones:

$$Flujo(t) = Kg * (1 - Pob(t)/pmax) * Pob(t)$$

$$Pob(t+dt) = Pob(t) + Flujo(t)$$

- Simulación.

Time (Month)	"Pob"	Runs:	Pob
0	Current	2560	
1		2743.8	
2		2936.7	
3		3138.55	
4		3349.16	

5	3568.2
6	3795.29
7	4029.92
8	4271.5
9	4519.36
10	4772.71
11	5030.72
12	5292.46
13	5556.96
14	5823.21
15	6090.15
16	6356.73
17	6621.91
18	6884.66
19	7143.98
20	7398.95

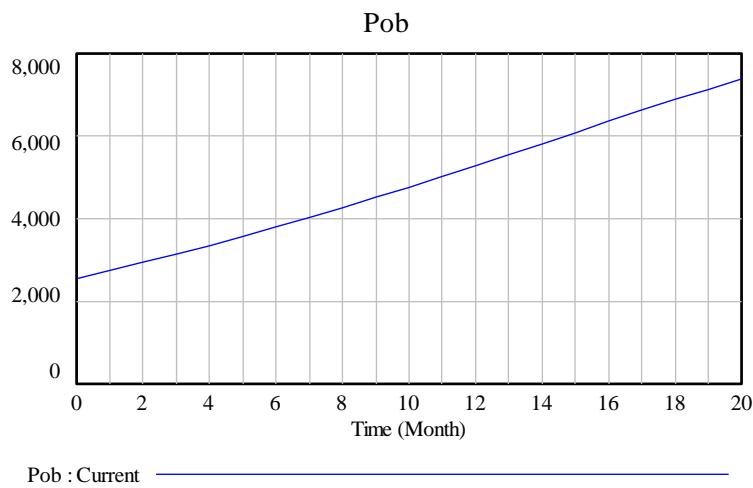
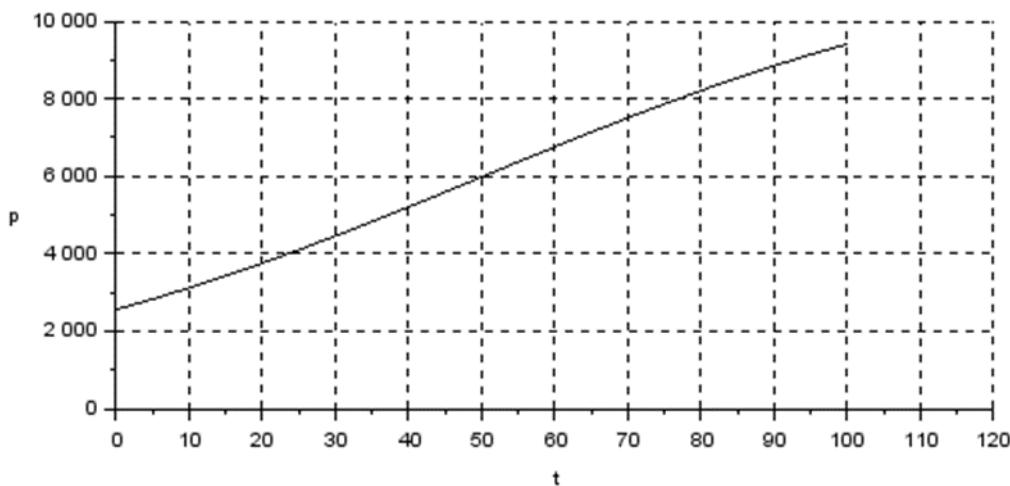
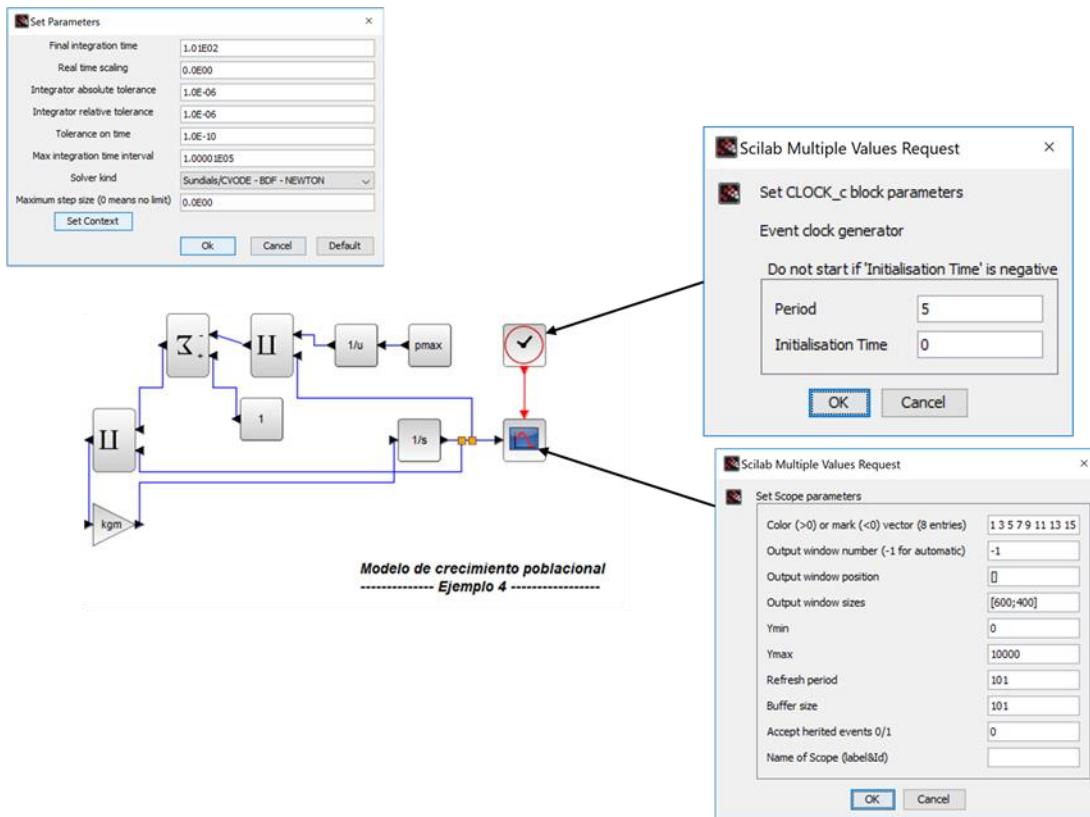


Figura A.28. Gráfica de resultados del ejemplo (4) usando Vensim.

➤ **Usando Xcos de Scilab:**



5)². MODELO DE TRAYECTORIA DE UN PROYECTIL.

Suponer que se lanza un proyectil hacia arriba desde la superficie de la Tierra. Suponer que la única fuerza actuando sobre el objeto es la fuerza hacia debajo de gravedad. Bajo estas condiciones, se usa un balance de fuerzas para derivar:

$$\frac{dv}{dt} = -g(0) \frac{R^2}{(R+x)^2}$$

donde v = velocidad de caída (m/s), t = tiempo (s), x = altitud (m) medida hacia arriba de la superficie terrestre, $g(0)$ = aceleración gravitacional en la superficie terrestre ($\approx 9.81 \text{ m/s}^2$) y R = radio de la Tierra ($\approx 6.37 \times 10^6 \text{ m}$). Reconociendo que $dx/dt = v$, usar el método de Euler para determinar la altura máxima que podría alcanzar si $v(0) = 1500 \text{ m/s}$.

Entonces,

$$\frac{dx}{dt} = f_1(x, v, t) = v$$

$$\frac{dv}{dt} = f_2(x, v, t) = -g \frac{R^2}{(R+x)^2} \quad v(0) = 1500, \quad x(0) = 0$$

Asignando a las variables los componentes del vector \mathbf{y} , se tiene que: $y(1) = x$, $y(2) = v$. Entonces,

$$f_1(x, v, t) = y(2)$$

$$f_2(x, v, t) = -g \frac{R^2}{(R+y(1))^2} \quad , \quad y(1)_0 = 0, \quad y(2)_0 = 1500$$

➤ Resolviendo numéricamente con comandos de Octave.

```
>> function fun=fcn(y,t)
g=9.81; R=6.37e6;
fun(1)=y(2);
fun(2)=-g*R^2/(R+y(1))^2;
endfunction
>> y0=[0,1500];
>> t=[0:5:200];
>> y=lsode ("fcn",y0,t);
>> plot(t,y(:,1),'r-',t,y(:,2),'b')
>> legend('x','v')
>> grid
>> for i=1:41
fprintf('%4.1f %8.6f %8.6f',t(i),y(i,1),y(i,2));
fprintf('\n')
endfor
0.0 0.000000 1500.000000
5.0 7377.470374 1451.007056
10.0 14510.256133 1402.125458
15.0 21398.903682 1353.351096
20.0 28043.939036 1304.679904
25.0 34445.867986 1256.107863
30.0 40605.176317 1207.630995
```

```

35.0 46522.330072 1159.245364
40.0 52197.775600 1110.947072
45.0 57631.939933 1062.732260
50.0 62825.230859 1014.597101
55.0 67778.037225 966.537803
60.0 72490.728938 918.550607
65.0 76963.657172 870.631782
70.0 81197.154520 822.777627
75.0 85191.535109 774.984466
80.0 88947.094762 727.248651
85.0 92464.111116 679.566555
90.0 95742.843777 631.934575
95.0 98783.534166 584.349126
100.0 101586.406141 536.806645
105.0 104151.665822 489.303584
110.0 106479.501186 441.836413
115.0 108570.082973 394.401615
120.0 110423.564555 346.995686
125.0 112040.081340 299.615134
130.0 113419.751412 252.256478
135.0 114562.675863 204.916245
140.0 115468.938069 157.590970
145.0 116138.603932 110.277193
150.0 116571.722568 62.971459
155.0 116768.325644 15.670317
160.0 116728.427306 -31.629682
165.0 116452.024747 -78.931988
170.0 115939.097715 -126.240050
175.0 115189.608805 -173.557321
180.0 114203.503308 -220.887258
185.0 112980.709216 -268.233320
190.0 111521.137298 -315.598977
195.0 109824.680792 -362.987703
200.0 107891.215759 -410.402983
>>

```

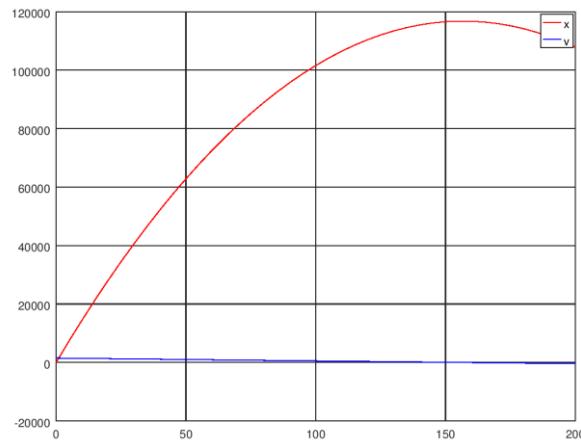


Figura A.29. Gráfica de resultados del ejemplo (5) usando comandos de Octave.

Para que se observe mejor,

```
>> plot(t,y(:,1), 'r-')
>> legend('x')
>> grid
```

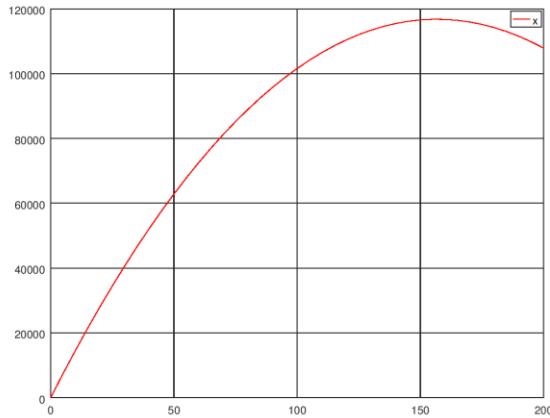


Figura A.29. Gráfica de resultados del ejemplo (5) usando comandos de Octave. Cont.

```
>> plot(t,y(:,2), 'b')
>> legend('v')
>> grid
```

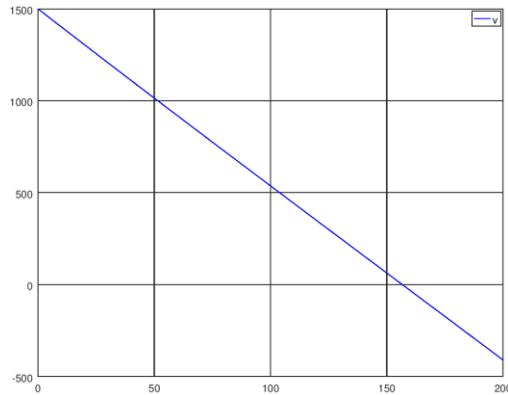


Figura A.29. Gráfica de resultados del ejemplo (5) usando comandos de Octave. Cont.

- Resolviendo con Euler, usando el código propuesto se cambia $f(t, x, v)$ por $f(t, x, y)$, se obtiene lo siguiente:

Método de Euler.

Datos:

```
t0 = 0
tn = 200
x0 = 0
y0 = 1500
Incremento de t = 5
```

i	t(i)	x(i)	y(i)
0	0.0000	0.000000	1500.000000
1	5.0000	7500.000000	1450.950000
2	10.0000	14754.750000	1402.015299
3	15.0000	21764.826493	1353.191739
4	20.0000	28530.785191	1304.475214
5	25.0000	35053.161263	1255.861663
6	30.0000	41332.469579	1207.347069
7	35.0000	47369.204925	1158.927460
8	40.0000	53163.842227	1110.598904
9	45.0000	58716.836745	1062.357507
10	50.0000	64028.624278	1014.199413
11	55.0000	69099.621345	966.120804
12	60.0000	73930.225365	918.117891
13	65.0000	78520.814822	870.186921
14	70.0000	82871.749429	822.324170
15	75.0000	86983.370278	774.525940
16	80.0000	90855.999979	726.788565
17	85.0000	94489.942802	679.108399
18	90.0000	97885.484798	631.481824
19	95.0000	101042.893920	583.905243
20	100.0000	103962.420136	536.375078
21	105.0000	106644.295528	488.887773
22	110.0000	109088.734392	441.439787
23	115.0000	111295.933325	394.027596
24	120.0000	113266.071306	346.647693
25	125.0000	114999.309769	299.296580
26	130.0000	116495.792670	251.970775
27	135.0000	117755.646547	204.666805
28	140.0000	118778.980571	157.381205
29	145.0000	119565.886594	110.110518
30	150.0000	120116.439182	62.851294
31	155.0000	120430.695651	15.600088
32	160.0000	120508.696089	-31.646543
33	165.0000	120350.463374	-78.892038
34	170.0000	119956.003183	-126.139837
35	175.0000	119325.303999	-173.393379
36	180.0000	118458.337103	-220.656107
37	185.0000	117355.056567	-267.931466
38	190.0000	116015.399236	-315.222907
39	195.0000	114439.284703	-362.533884
40	200.0000	112626.615280	-409.867864

>>

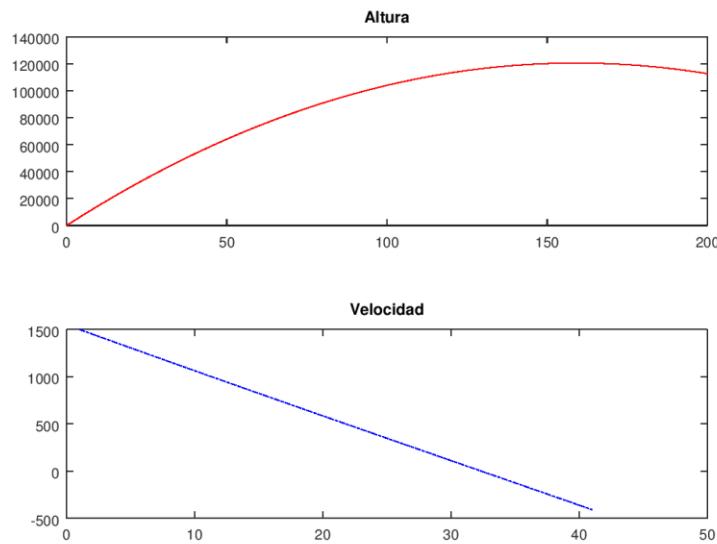


Figura A.30. Gráfica de resultados del ejemplo (5) usando código en Octave para Euler.

Ejecutando de nuevo con incremento de tiempo de 0.5 se obtiene lo siguiente:

```
313      156.5000 117151.143296      1.478209
314      157.0000 117151.882401     -3.251232
```

Es decir, la altura máxima se alcanza entre 156.5 y 157.0 s. y es de aproximadamente 117,152 m.

➤ **Dinámica de sistemas.**

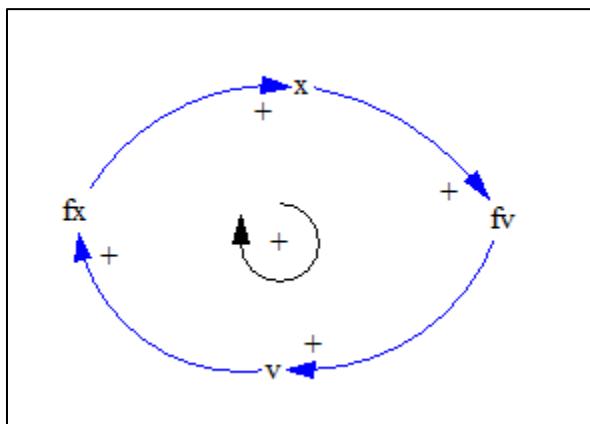


Figura A.31. Diagrama de influencias del ejemplo (5).

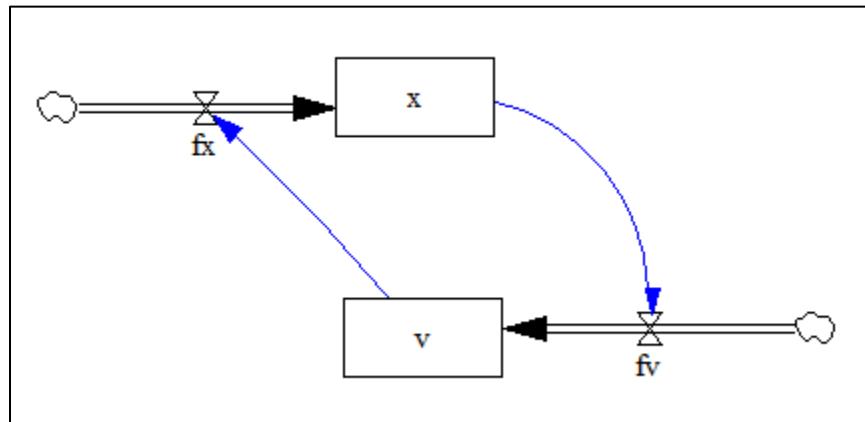
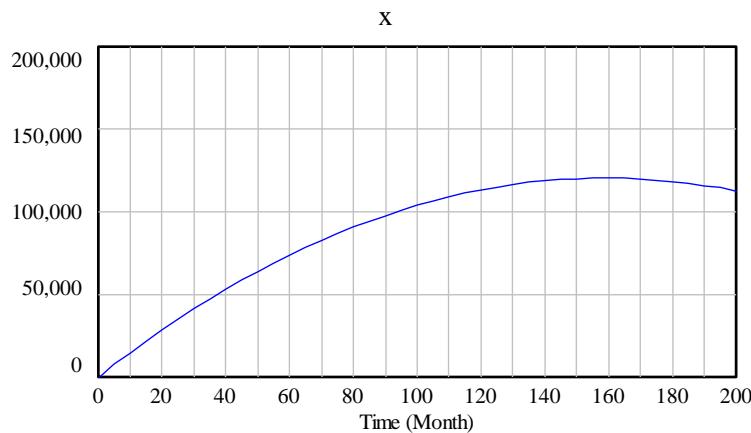


Figura A.32. Diagrama de Forrester del ejemplo (5).

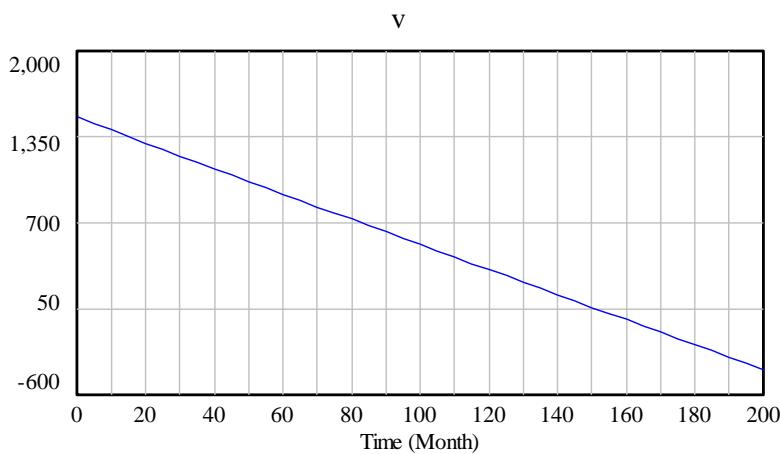
Time (Month)	"x"	Runs: x
0	0	
5	7500	
10	14754.8	
15	21764.8	
20	28530.8	
25	35053.2	
30	41332.5	
35	47369.2	
40	53163.8	
45	58716.8	
50	64028.6	
55	69099.6	
60	73930.2	
65	78520.8	
70	82871.7	
75	86983.3	
80	90856	
85	94489.9	
90	97885.5	
95	101043	
100	103962	
105	106644	
110	109089	
115	111296	
120	113266	
125	114999	
130	116496	
135	117756	
140	118779	
145	119566	
150	120116	

155	120431
160	120509
165	120350
170	119956
175	119325
180	118458
185	117355
190	116015
195	114439
200	112627



x : Ejem_2_26

Figura A.33.Resultado gráfico del ejemplo (5) usando Vensim.



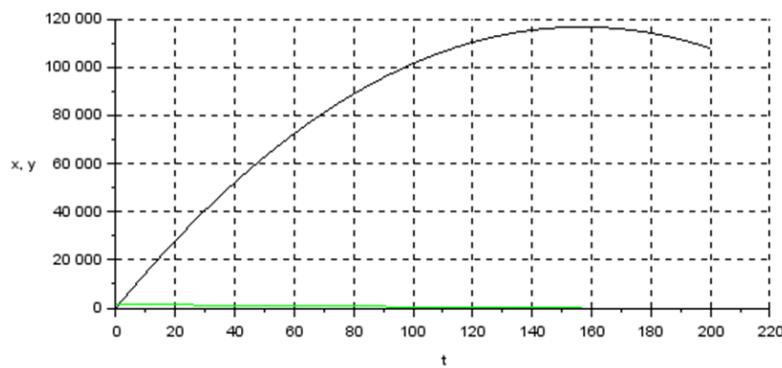
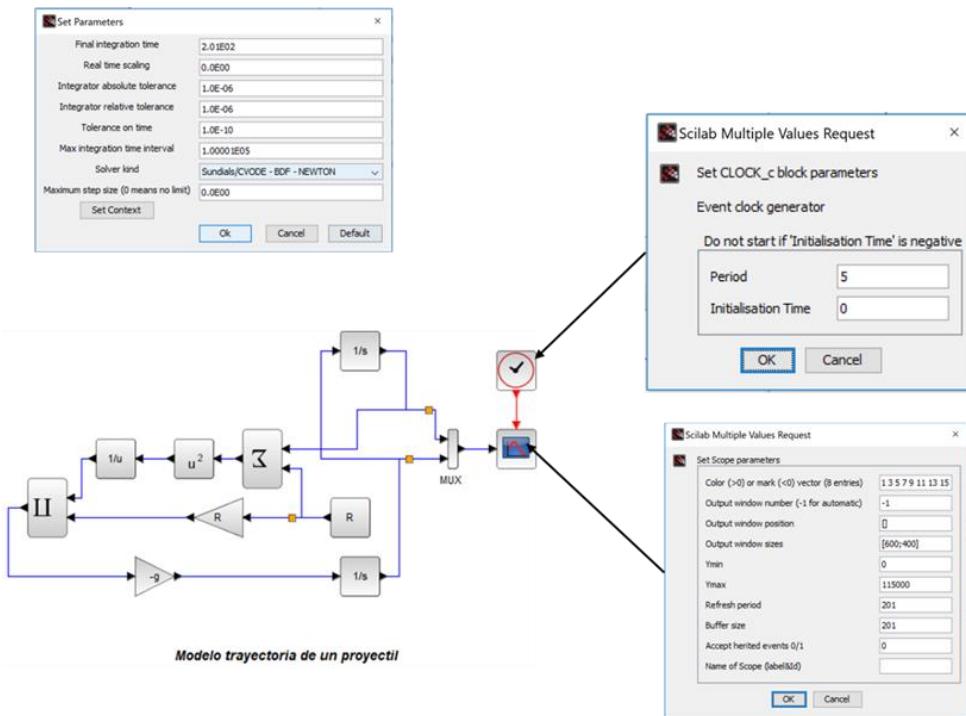
v : Ejem_2_26

Figura A.33.Resultado gráfico del ejemplo (5) usando Vensim. Cont.

Con incremento de tiempo de 0.5 se obtiene que la velocidad cambia en:

```
Time (Month) "v"      Runs: v
0          Ejem_2_26    1500
0.5        1495.09
...
156.5      1.47854
157        -3.2509
```

➤ **Usando Xcos de Scilab:**



6)². MODELO PARA UN CIRCUITO ELECTRÓNICO.

La ecuación de Van der Pol es un modelo de un circuito electrónico que surgió en los días de los tubos de vacío:

$$\frac{d^2y}{dt^2} - (1-y^2)\frac{dy}{dt} + y = 0$$

Dadas las condiciones iniciales, $y(0) = y'(0) = I$, resolver esta ecuación de $t = 0$ a 10 usando el método de Euler con un paso de (a) 0.25 y (b) 0.125. Graficar ambas soluciones en una misma gráfica.

$$\frac{dy}{dt} = f_1(y, z, t) = z$$

$$\frac{dz}{dt} = f_2(y, z, t) = (1-y^2)z - y \quad , \quad y(0) = z(0) = 1$$

Asignando a las variables los componentes del vector \mathbf{y} , se tiene que: $y(1) = y$, $y(2) = z$. Entonces,

$$f_1(y, z, t) = y(2)$$

$$f_2(y, z, t) = (1-y^2)y(2) - y(1) \quad , \quad y(1)_0 = y(2)_0 = 1$$

```
>> function fun=fcn(y,t)
fun(1)=y(2);
fun(2)=(1-y(1)^2)*y(2)-y(1);
endfunction
>> y0=[1,1];
>> t=[0:0.25:10];
>> y=lsode("fcn",y0,t);
>> for i=1:41
fprintf('%4.2f %8.6f %8.6f',t(i),y(i,1),y(i,2));
fprintf('\n')
endfor
0.00 1.000000 1.000000
0.25 1.211981 0.673681
0.50 1.331264 0.279973
0.75 1.355270 -0.077016
1.00 1.298482 -0.367035
1.25 1.175527 -0.612217
1.50 0.992836 -0.851946
1.75 0.746399 -1.129618
2.00 0.421175 -1.488952
2.25 -0.006582 -1.946019
2.50 -0.549871 -2.366566
2.75 -1.149395 -2.302646
```

```

3.00 -1.634813 -1.485462
3.25 -1.881065 -0.524932
3.50 -1.928782 0.079883
3.75 -1.865620 0.391589
4.00 -1.743955 0.568923
4.25 -1.584614 0.703345
4.50 -1.391940 0.841442
4.75 -1.160942 1.015058
5.00 -0.878655 1.258107
5.25 -0.522584 1.612630
5.50 -0.060691 2.102495
5.75 0.529597 2.587159
6.00 1.187087 2.521679
6.25 1.711394 1.565389
6.50 1.962541 0.502911
6.75 2.002272 -0.112291
7.00 1.933024 -0.406838
7.25 1.809764 -0.566947
7.50 1.652796 -0.686546
7.75 1.466191 -0.809294
8.00 1.245559 -0.963189
8.25 0.979582 -1.177624
8.50 0.648483 -1.491435
8.75 0.222310 -1.941563
9.00 -0.329624 -2.467255
9.25 -0.984159 -2.652799
9.50 -1.576489 -1.932960
9.75 -1.914874 -0.794490
10.00 -2.008257 -0.034150
>> plot(t,y(:,1),'r-')
>> grid
>>

```

Se procede similarmente para un paso de 0.125 y luego se grafica,

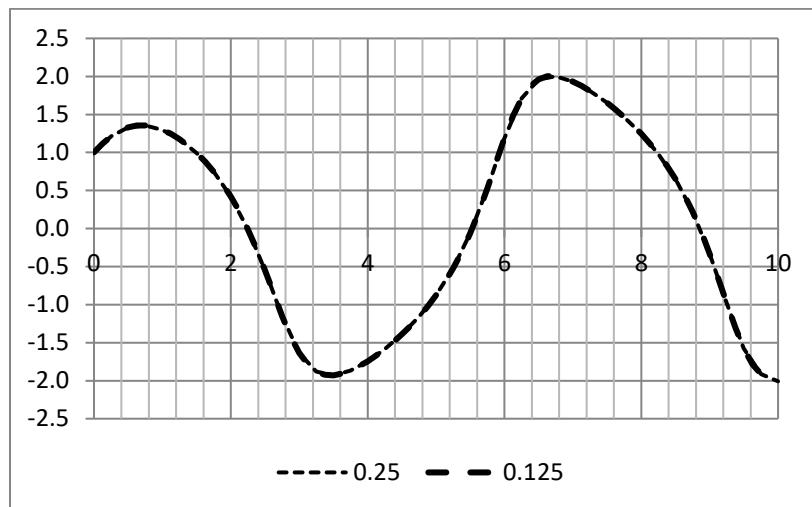


Figura A.34.Resultado gráfico del ejemplo (6) usando comandos de Octave.

➤ Aplicando Euler con código en Octave.

Método de Euler.

Datos:

```
t0 = 0  
tn = 10  
x0 = 1  
y0 = 1  
Incremento de t = 0.25
```

i	t(i)	x(i)	y(i)
0	0.0000	1.000000	1.000000
1	0.2500	1.250000	0.750000
2	0.5000	1.437500	0.332031
3	0.7500	1.520508	-0.115864
4	1.0000	1.491542	-0.457989
5	1.2500	1.377045	-0.690650
6	1.5000	1.204382	-0.880162
7	1.7500	0.984342	-1.082121
8	2.0000	0.713811	-1.336613
9	2.2500	0.379658	-1.678959
10	2.5000	-0.040082	-2.133112
11	2.7500	-0.573360	-2.655512
12	3.0000	-1.237238	-2.957806
13	3.2500	-1.976689	-2.256028
14	3.5000	-2.540696	-0.122118
15	3.7500	-2.571226	0.679599
16	4.0000	-2.401326	0.369064
17	4.2500	-2.309060	0.529622
18	4.5000	-2.176655	0.533338
19	4.7500	-2.043320	0.579121
20	5.0000	-1.898540	0.630251
21	5.2500	-1.740977	0.694521
22	5.5000	-1.567347	0.777122
23	5.7500	-1.373066	0.885974
24	6.0000	-1.151573	1.033150
25	6.2500	-0.893285	1.236811
26	6.5000	-0.584083	1.522604
27	6.7500	-0.203432	1.919415
28	7.0000	0.276422	2.430269
29	7.2500	0.883989	2.922306
30	7.5000	1.614566	2.860986
31	7.7500	2.329813	1.308070
32	8.0000	2.656830	-0.722425
33	8.2500	2.476224	-0.292385
34	8.5000	2.403127	-0.536334
35	8.7500	2.269044	-0.496864
36	9.0000	2.144828	-0.548807
37	9.2500	2.007626	-0.591048

```

38      9.5000    1.859864   -0.645153
39      9.7500    1.698576   -0.713496
40     10.0000    1.520202   -0.801877
>>

```

Haciendo lo mismo para un paso de 0.125 y graficando ambas.

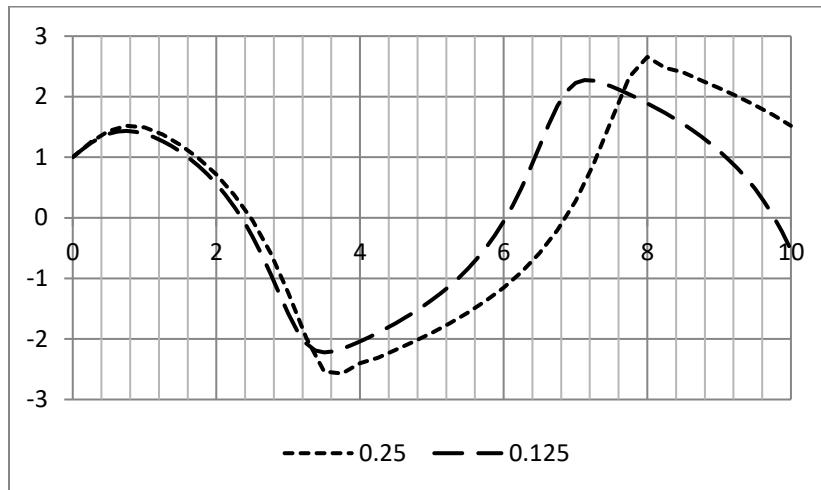


Figura A.35.Resultado gráfico del ejemplo (6) usando código de Octave para Euler.

- Aplicando rk4 en código de Octave con paso igual a 0.125.

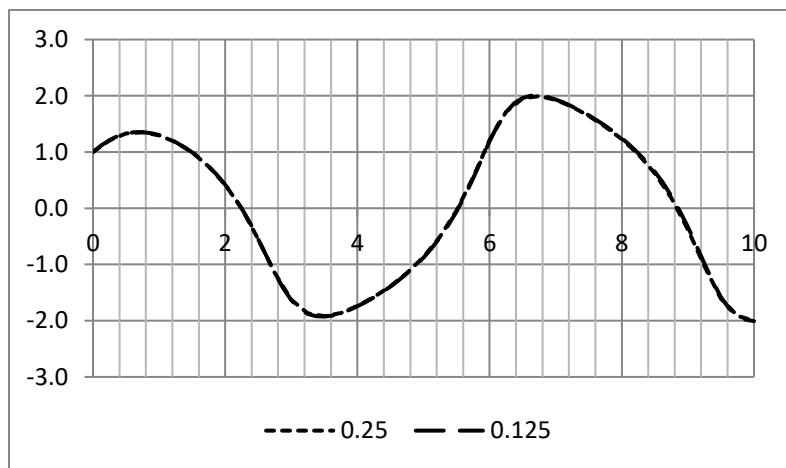


Figura A.35.Resultado gráfico del ejemplo (6) usando código de Octave para rk4. Cont.

➤ Dinámica de sistemas (Euler).

Paso = 0.25,

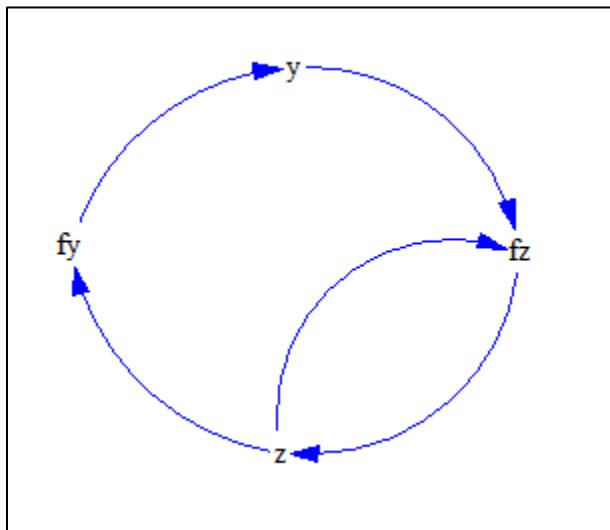


Figura A.36. Diagrama de influencias para en ejemplo (6).

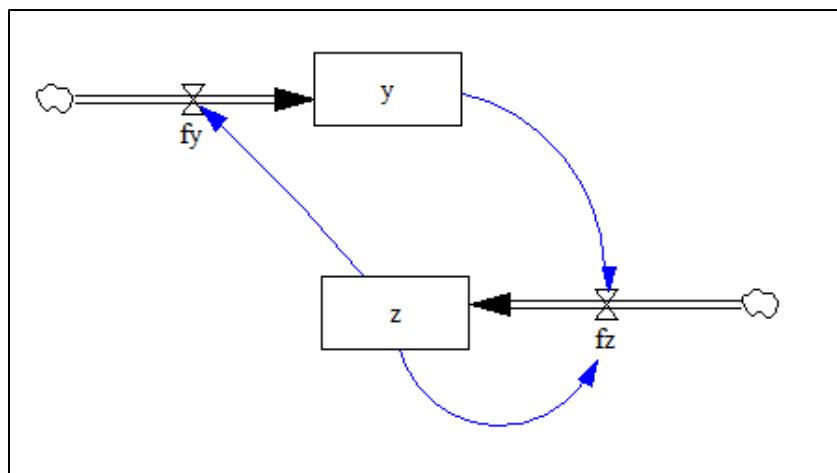


Figura A.37. Diagrama de Forrester para en ejemplo (6).

Time (Month)	"y"	Runs:
0	Current	1
0.25		1.25
0.5		1.4375

0.75	1.52051
1	1.49154
1.25	1.37704
1.5	1.20438
1.75	0.984342
2	0.713811
2.25	0.379658
2.5	-0.0400817
2.75	-0.573359
3	-1.23724
3.25	-1.97669
3.5	-2.5407
3.75	-2.57123
4	-2.40133
4.25	-2.30906
4.5	-2.17665
4.75	-2.04332
5	-1.89854
5.25	-1.74098
5.5	-1.56735
5.75	-1.37307
6	-1.15157
6.25	-0.893285
6.5	-0.584082
6.75	-0.203432
7	0.276422
7.25	0.883989
7.5	1.61457
7.75	2.32981
8	2.65683
8.25	2.47622
8.5	2.40313
8.75	2.26904
9	2.14483
9.25	2.00763
9.5	1.85986
9.75	1.69858
10	1.5202

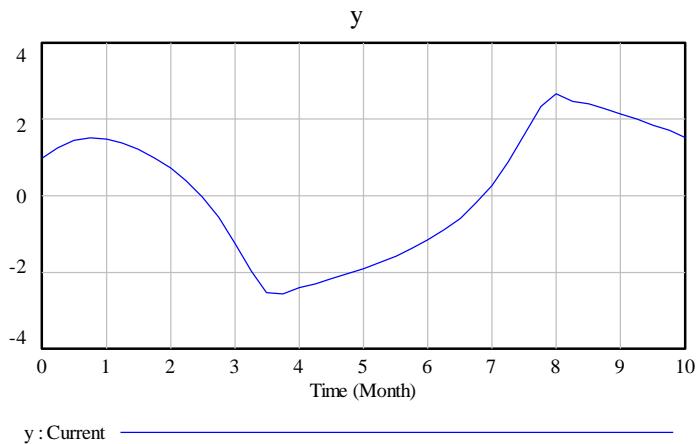


Figura A.38. Gráfica de los resultados del ejemplo (6) usando Vensim para Euler. Paso = 0.25.

Paso = 0.125,

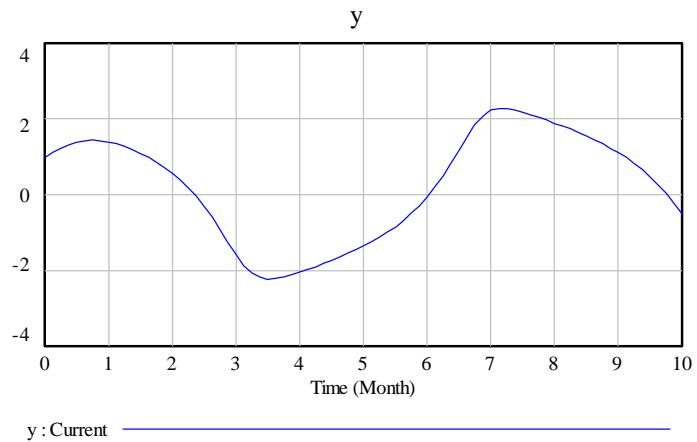


Figura A.39. Gráfica de los resultados del ejemplo (6) usando Vensim para Euler. Paso = 0.125.

Euler ambos pasos:

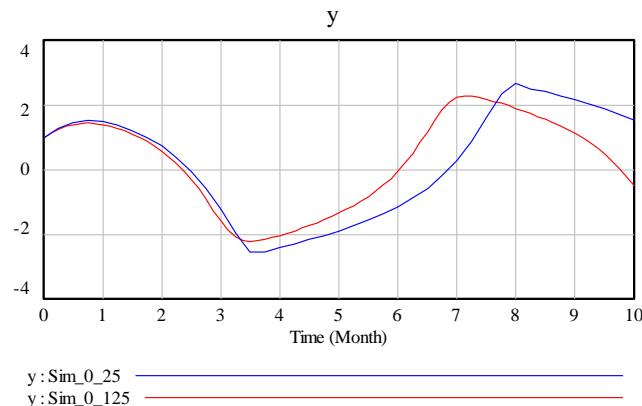


Figura A.40. Gráfica de los resultados del ejemplo (6) usando Vensim para Euler. Ambos pasos.

Aplicando rk4 con ambos pasos se obtiene lo siguiente:

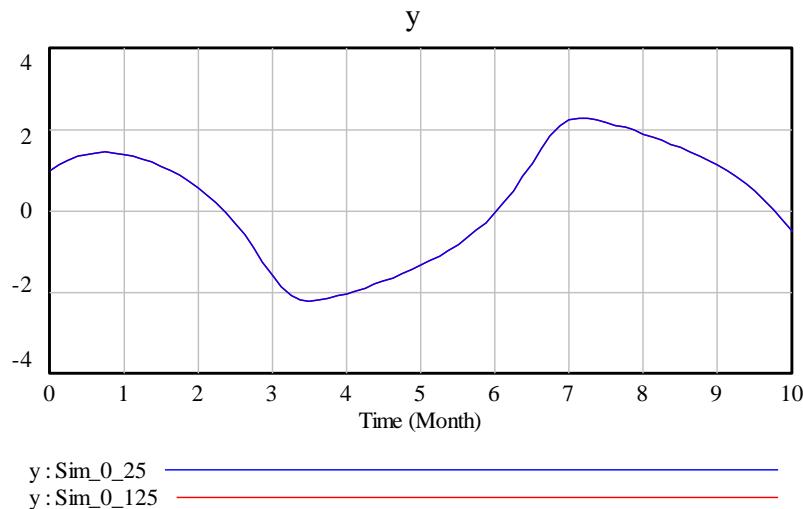
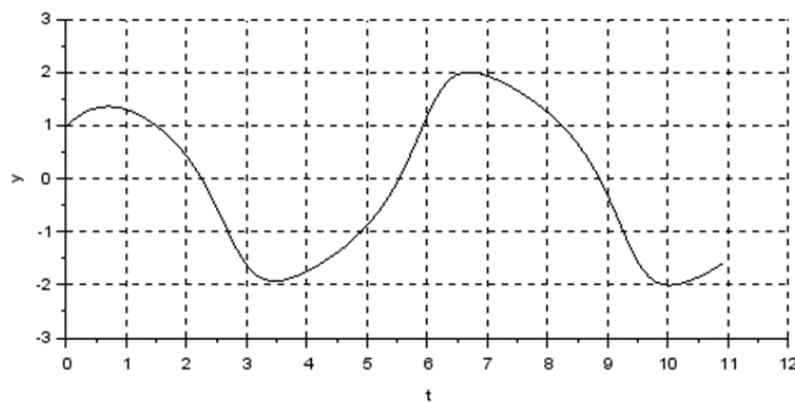
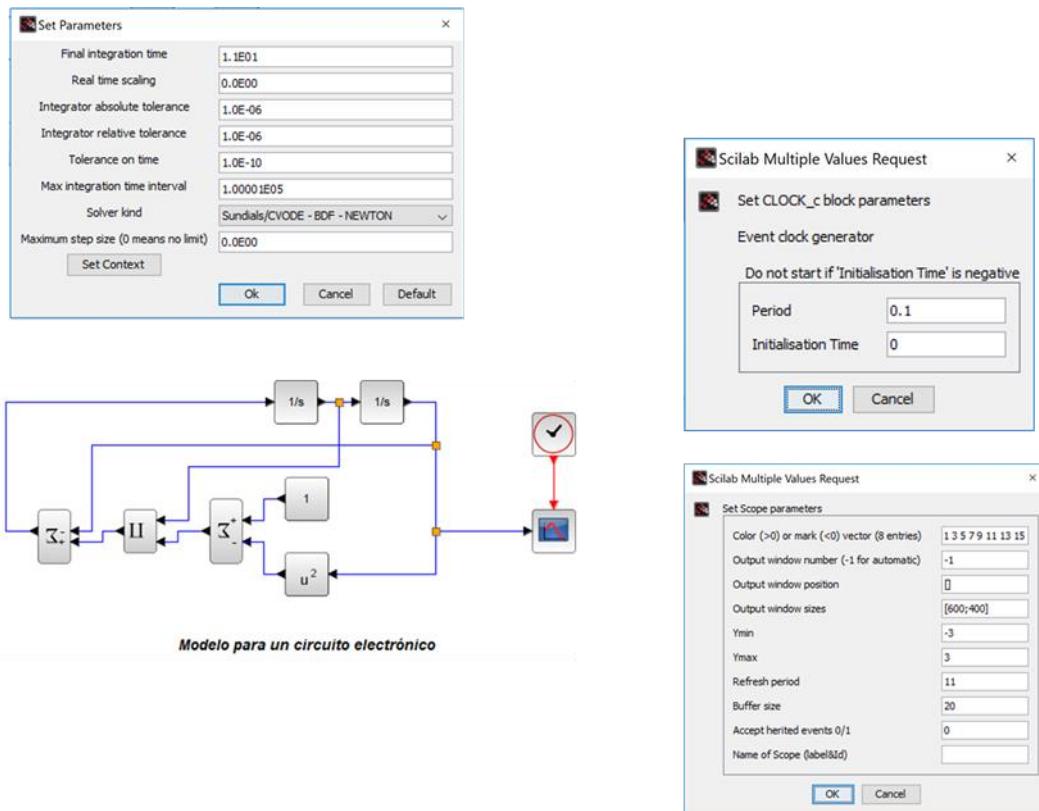


Figura A.41. Gráfica de los resultados del ejemplo (6) usando Vensim para rk4. Ambos pasos.

➤ **Usando Xcos de Scilab:**



7)². **Modelo de un sistema amortiguado de resorte-masa.**

El movimiento de un sistema amortiguado de resorte-masa se describe por la siguiente ecuación diferencial ordinaria:

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

donde x = desplazamiento desde la posición de equilibrio (m), t = tiempo (s), $m = 20 \text{ kg}$ y c = coeficiente de amortiguación ($\text{N}\cdot\text{s}/\text{m}$). El coeficiente de amortiguación c varía entre tres valores: 5 (sub-amortiguado), 40 (críticamente amortiguado) y 200 (sobre-amortiguado). La constante del resorte $k = 20 \text{ N/m}$. La velocidad inicial es cero y el desplazamiento inicial $x = 1 \text{ m}$. Resolver esta ecuación usando un método numérico en un período de tiempo de $0 \leq t \leq 15 \text{ s}$. Graficar el desplazamiento vs tiempo para cada uno de los tres coeficientes de amortiguación en una misma gráfica.

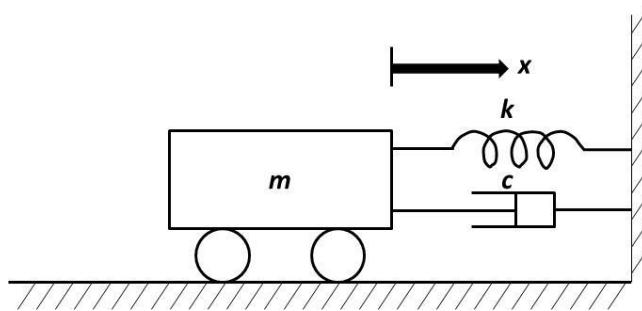


Figura A.42. Figura del ejemplo (7).

$$\frac{dx}{dt} = f_1(x, z, t) = z$$

$$\frac{dz}{dt} = f_2(x, z, t) = -\frac{c}{m}z - \frac{k}{m}x \quad , \quad x(0) = 1, z(0) = 0$$

Asignando a las variables los componentes del vector \mathbf{y} , se tiene que: $y(1) = x$, $y(2) = z$. Entonces,

$$f_1(x, z, t) = y(2)$$

$$f_2(x, z, t) = -\frac{c}{m}y(2) - \frac{k}{m}y(1) \quad , \quad y(1)_0 = 1, y(2)_0 = 0$$

```
>> function fun=fcn(y, t)
fun(1)=y(2);
m=20; k=20; c=5;
fun(2)=-(c/m)*y(2)-(k/m)*y(1);
endfunction
>> y0=[1, 0];
>> t=[0:0.1:15];
>> y=lsode ("fcn", y0, t);
>> y(:,1)
```

```
ans =
1.0000000
0.9950455
0.9803945
0.9564308
0.9236209
...
-0.0203135
-0.0359347
-0.0508155
-0.0648275
-0.0778539
-0.0897903
```

```
>>
```

Así se sigue con los valores restantes del coeficiente de amortiguación. Finalmente se grafican.

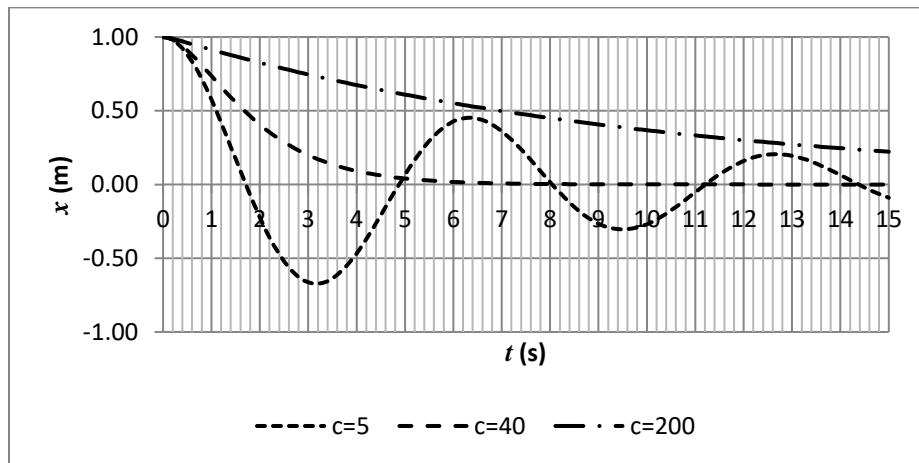


Figura A.43. Gráfica de los resultados del ejemplo (7) usando comandos de Octave.

➤ Usando código en Octave para aplicar rk4 se obtiene el siguiente resultado:

Método de Runge-Kutta de cuarto orden.

Datos:

```
t0 = 0
tn = 15
x0 = 1
y0 = 0
Incremento de t = 0.1
```

i	t(i)	x(i)	y(i)	k1	k2	k3	k4	11	12	13	14
0	0.0000	1.000000	0.000000	0.000000	-0.050000	-0.049375	-0.098750	-1.000000	-0.987500	-0.985156	-0.970313
1	0.1000	0.995042	-0.098594	-0.098594	-0.147113	-0.146260	-0.193927	-0.970393	-0.953334	-0.951121	-0.931849
2	0.2000	0.980387	-0.193780	-0.193780	-0.240377	-0.239310	-0.284840	-0.931942	-0.910604	-0.908541	-0.885140
3	0.3000	0.956421	-0.284702	-0.284702	-0.328965	-0.327700	-0.370697	-0.885245	-0.859944	-0.858047	-0.830850
4	0.4000	0.923609	-0.370570	-0.370570	-0.412119	-0.410673	-0.450775	-0.830966	-0.802050	-0.800334	-0.769703
5	0.5000	0.882493	-0.450661	-0.450661	-0.489152	-0.487545	-0.524428	-0.769828	-0.737672	-0.736149	-0.702471
145	14.5000	-0.020262	-0.159379	-0.159379	-0.156373	-0.156012	-0.152646	0.060107	0.067324	0.067084	0.074061
146	14.6000	-0.035875	-0.152662	-0.152662	-0.148960	-0.148625	-0.144587	0.074041	0.080748	0.080479	0.086918
147	14.7000	-0.050749	-0.144605	-0.144605	-0.140260	-0.139953	-0.135301	0.086900	0.093044	0.092750	0.098600
148	14.8000	-0.064754	-0.135321	-0.135321	-0.130391	-0.130115	-0.124909	0.098585	0.104118	0.103803	0.109021
149	14.9000	-0.077775	-0.124930	-0.124930	-0.119479	-0.119235	-0.113541	0.109008	0.113891	0.113558	0.118108
150	15.0000	-0.089707	-0.113563								

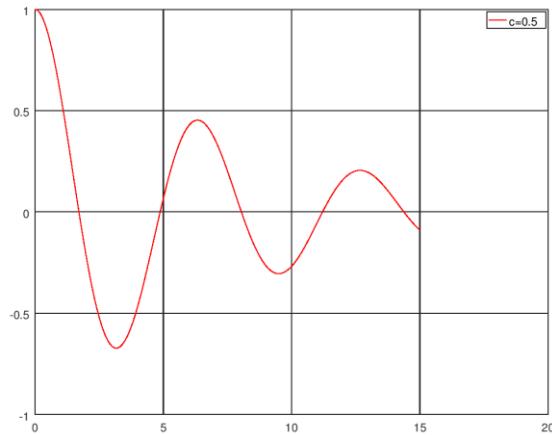


Figura A.44. Gráfica de los resultados del ejemplo (7) usando código de Octave para rk4.

Repetiendo el procedimiento para los otros valores del coeficiente de amortiguación se obtiene la siguiente gráfica:

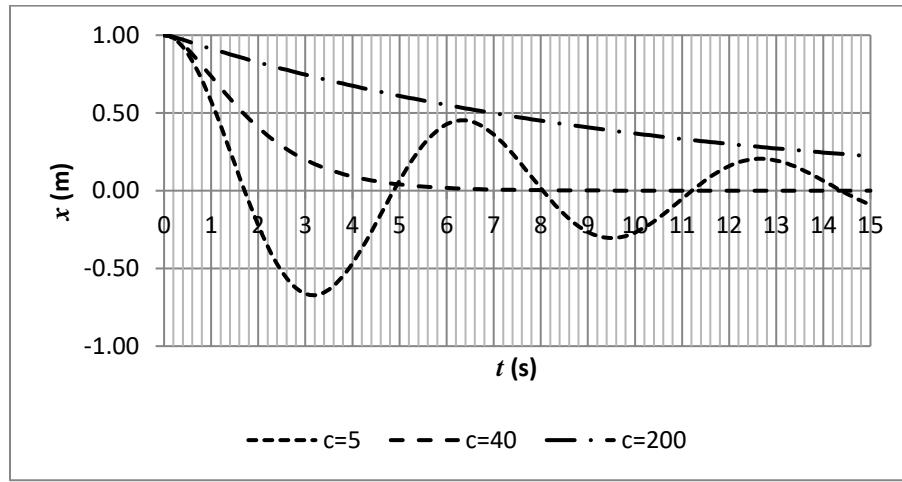


Figura A.44. Gráfica de los resultados del ejemplo (7) usando código de Octave para rk4. Cont.

➤ **Dinámica de sistemas.**

- Diagrama de influencias.

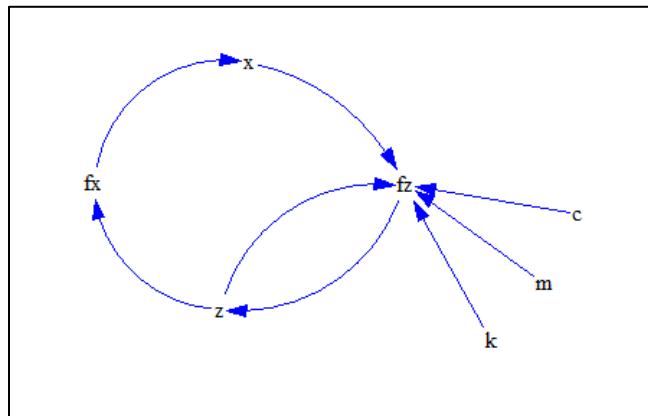


Figura A.45. Diagrama de influencias del ejemplo (7).

- Diagrama de Forrester.

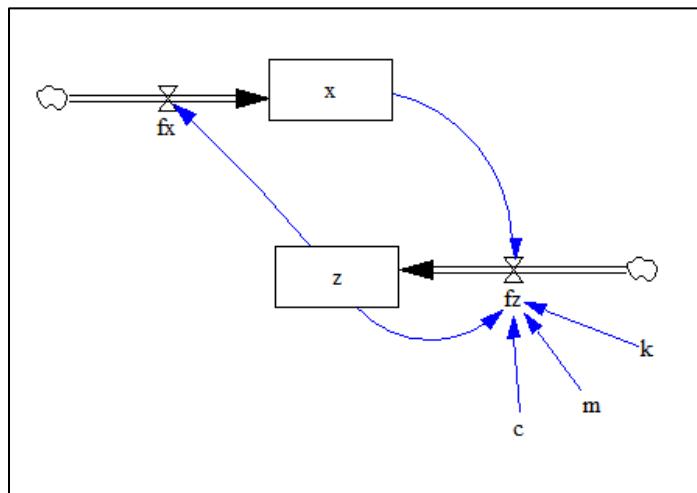


Figura A.46. Diagrama de Forrester del ejemplo (7).

- Gráfica solicitada:

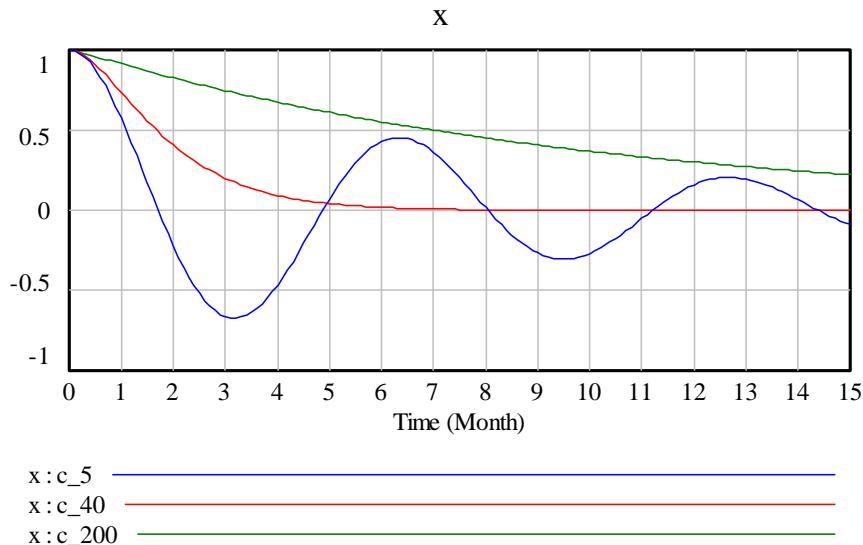
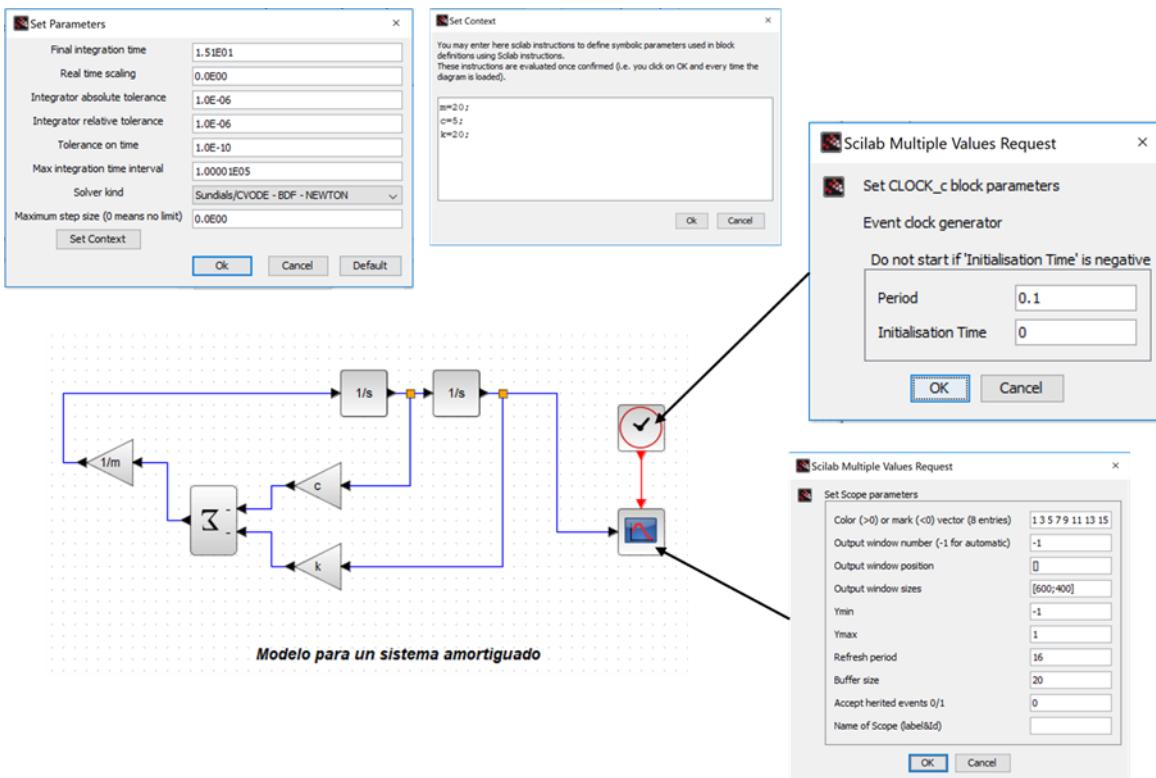
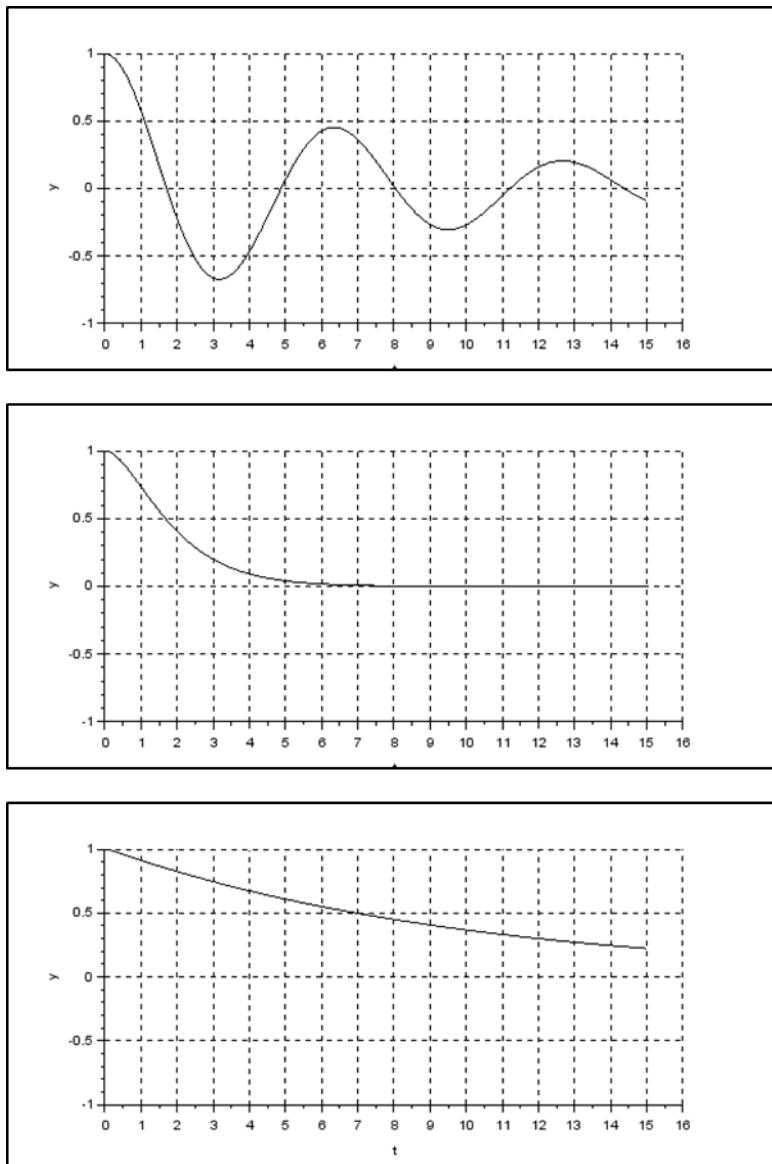


Figura A.47. Gráfica de los resultados del ejemplo (7) usando Vensim.

➤ **Usando Xcos de Scilab:**





8)². Modelo de un sistema de reactores.

Se tiene el arreglo de un sistema de reactores mostrado en la figura. Los volúmenes de cada reactor son: $V_1 = 50 \text{ m}^3$, $V_2 = 20 \text{ m}^3$, $V_3 = 40 \text{ m}^3$, $V_4 = 80 \text{ m}^3$ y $V_5 = 100 \text{ m}^3$. Suponer que para $t = 0$ todas las concentraciones en los reactores son cero. Calcular cómo se incrementarán sus concentraciones en la siguiente hora.

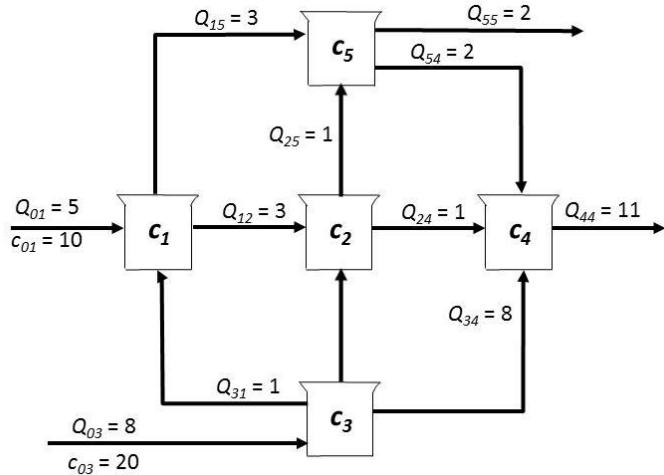


Figura A.48. Figura del ejemplo (8).

El balance de masa para el primer reactor puede escribirse como:

$$V_1 \frac{dc_1}{dt} = Q_{01}c_{01} + Q_{31}c_3 - Q_{12}c_1 - Q_{15}c_1$$

y sustituyendo parámetros,

$$\frac{dc_1}{dt} = -0.12c_1 + 0.02c_3 + 1$$

Similarmente, los balances pueden desarrollarse para los otros reactores como:

$$\frac{dc_2}{dt} = 0.15c_1 - 0.15c_2$$

$$\frac{dc_3}{dt} = 0.025c_2 - 0.225c_3 + 4$$

$$\frac{dc_4}{dt} = 0.0125c_2 + 0.1c_3 - 0.1375c_4 + 0.025c_5$$

$$\frac{dc_5}{dt} = 0.03c_1 + 0.01c_2 - 0.04c_5$$

➤ Resolviendo con rk4 usando comandos de Octave.

```
>> function fun=fcn(y,t)
fun(1)=-0.12*y(1)+0.02*y(3)+1;
```

```

fun(2)=0.15*y(1)-0.15*y(2);
fun(3)=0.025*y(2)-0.225*y(3)+4;
fun(4)=0.0125*y(2)+0.1*y(3)-0.1375*y(4)+0.025*y(5);
fun(5)=0.03*y(1)+0.01*y(2)-0.04*y(5);
endfunction
>> y0=[0,0,0,0,0];
>> t=[0:90];
>> y=lsode("fcn",y0,t);
>> for i=1:91
fprintf('%5.1f %10.6f %10.6f %10.6f %10.6f
%10.6f',t(i),y(i,1),y(i,2),y(i,3),y(i,4),y(i,5));
fprintf('\n')
endfor
0.0    0.000000    0.000000    0.000000    0.000000    0.000000
1.0    0.978025    0.070349    3.582497    0.177849    0.014825
2.0    1.905806    0.263641    6.446231    0.634208    0.058424
3.0    2.776947    0.555251    8.738480    1.275385    0.129170
4.0    3.588250    0.923335   10.576333    2.031674    0.225132
5.0    4.338846    1.348821   12.052829    2.851785    0.344197
6.0    5.029526    1.815299   13.241864    3.698505    0.484161
.
.
.
84.0  11.507914  11.504300  19.055531  16.812008  10.799652
85.0  11.508066  11.504815  19.055639  16.819541  10.827393
86.0  11.508204  11.505277  19.055735  16.826757  10.854055
87.0  11.508327  11.505694  19.055822  16.833669  10.879680
88.0  11.508438  11.506069  19.055900  16.840293  10.904307
89.0  11.508538  11.506406  19.055971  16.846640  10.927975
90.0  11.508628  11.506710  19.056034  16.852724  10.950721
>>

```

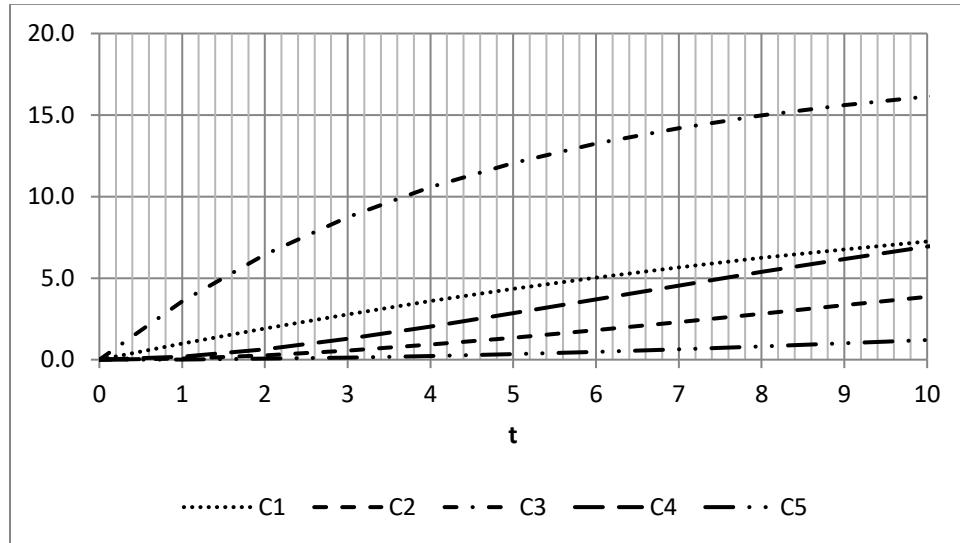


Figura A.49. Gráfica de resultados del ejemplo (8) usando código en Octave para rk4.

➤ **Dinámica de sistemas.**

- **Diagrama de Forrester:**

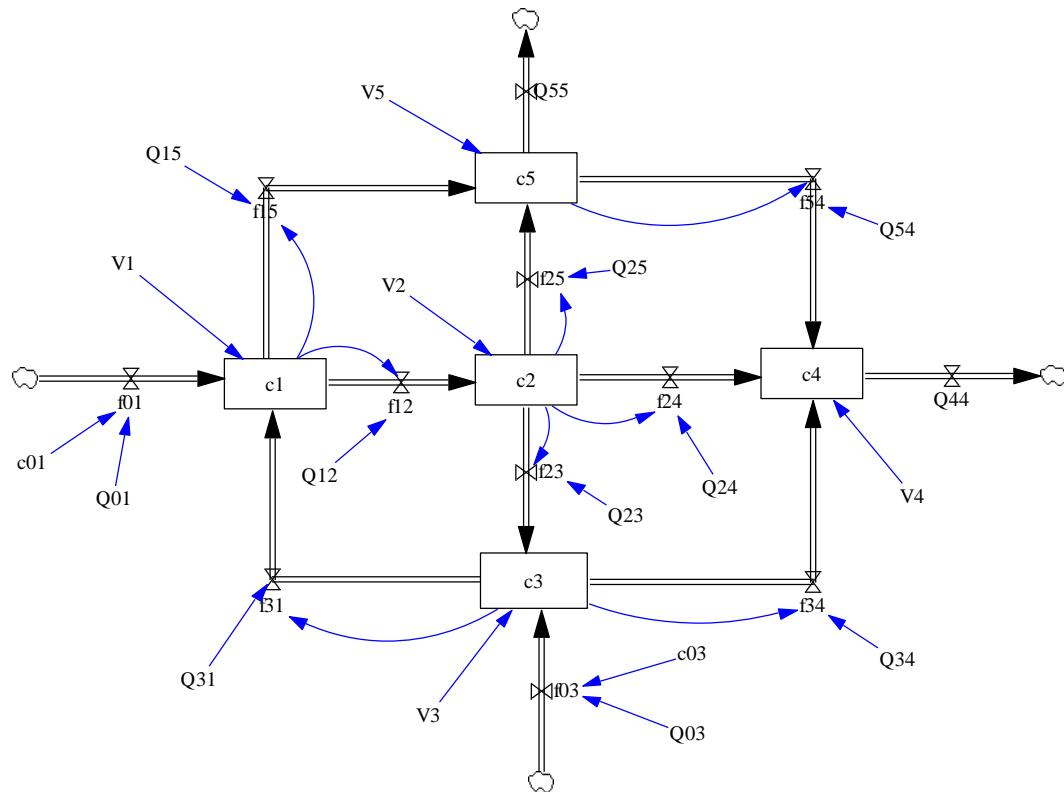
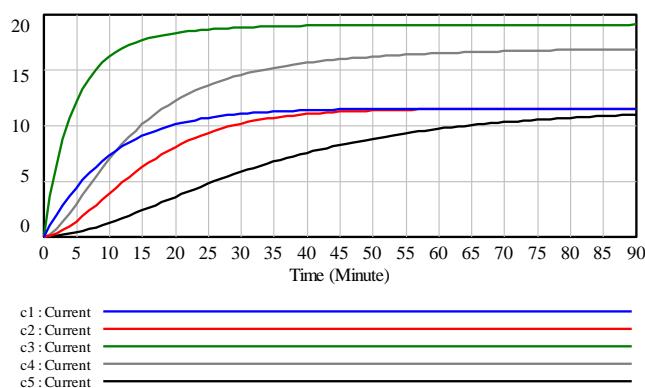


Figura A.50. Diagrama de Forrester para el ejemplo (8).

- **Simulación.**



O también,

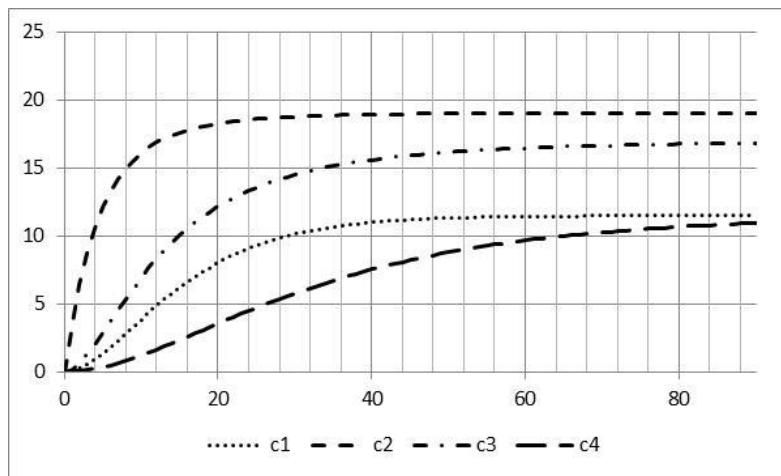


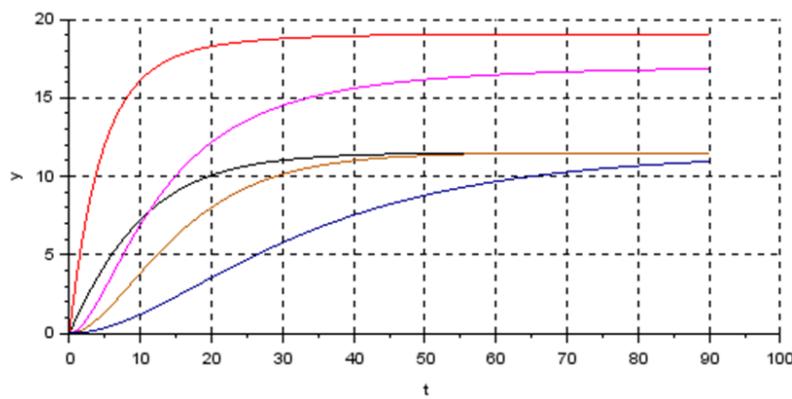
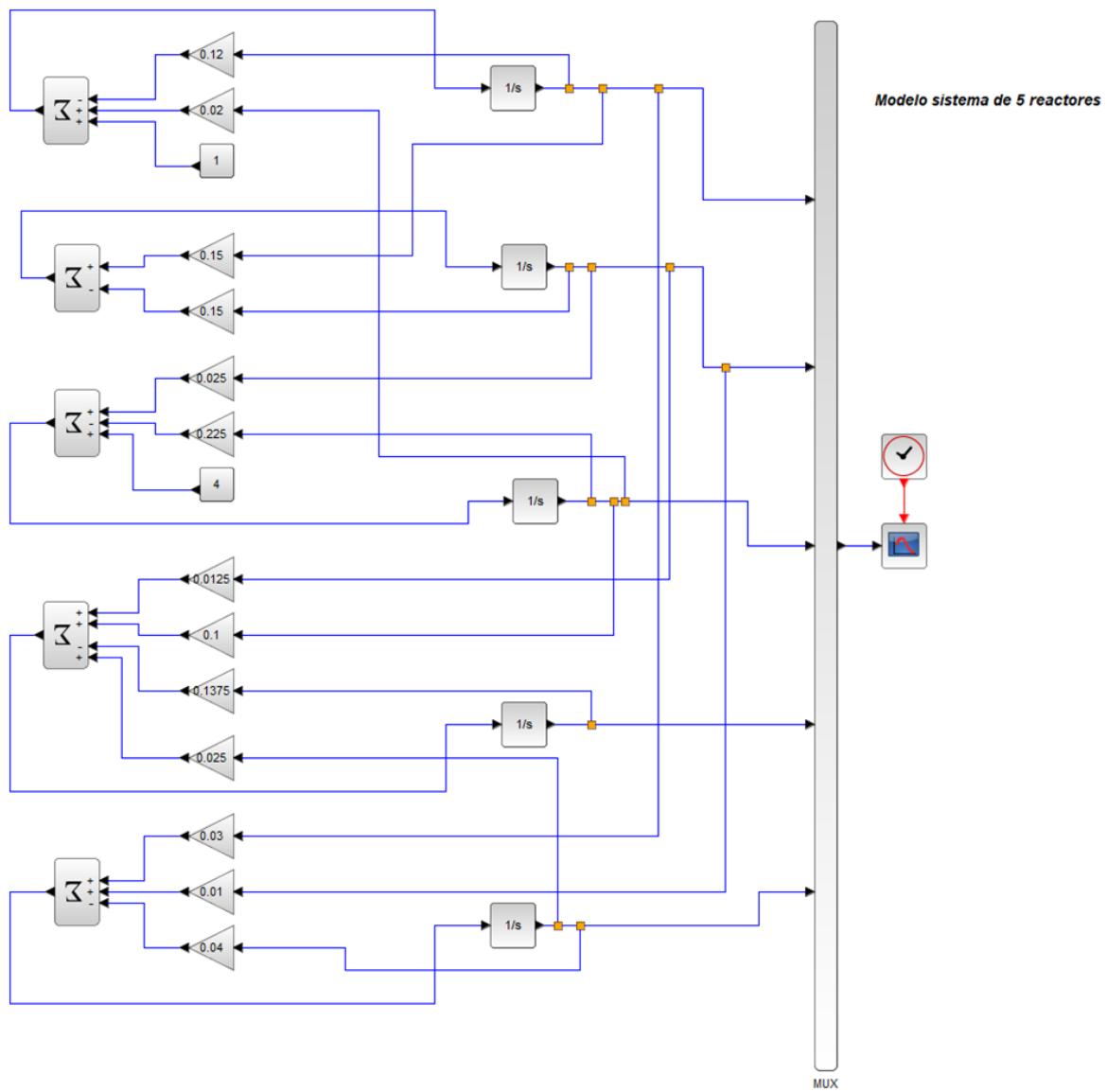
Figura A.51. Gráfica de los resultados del ejemplo (8).

Time (Minute)	c1	c2	c3	c4	c5
0	0	0	0	0	0
1	0.9780	0.0703	3.582	0.1778	0.0148
2	1.905	0.2636	6.446	0.6342	0.0584
3	2.776	0.5552	8.738	1.275	0.1291
4	3.588	0.9233	10.57	2.031	0.2251
5	4.338	1.348	12.05	2.851	0.3441
	...				
85	11.50	11.50	19.05	16.81	10.82
86	11.50	11.50	19.05	16.82	10.85
87	11.50	11.50	19.05	16.83	10.87
88	11.50	11.50	19.05	16.84	10.90
89	11.50	11.50	19.05	16.84	10.92
90	11.50	11.50	19.05	16.85	10.95

Conclusiones.

Se obtienen resultados muy similares usando métodos numéricos y dinámica de sistemas.

➤ Xcos de Scilab:



Apéndice B. Ejemplos prácticos usando Octave.

Raíces de ecuaciones.

B.1² La concentración de saturación del oxígeno disuelto en agua fresca puede ser calculada con la siguiente ecuación:

$$\ln O_{sf} = -139.34411 + \frac{1.575701 \times 10^5}{T_a} - \frac{6.642308 \times 10^7}{T_a^2} + \frac{1.243800 \times 10^{10}}{T_a^3} - \frac{8.621949 \times 10^{11}}{T_a^4}$$

donde O_{sf} = concentración de saturación del oxígeno disuelto en agua fresca a 1 atm (mg/l) y T_a = temperatura absoluta en (°K). Recordar que $T_a = T + 273.5$, donde T = temperatura (°C). De acuerdo con esta ecuación, la saturación disminuye con el incremento de temperatura. Para aguas naturales típicas en climas templados, la ecuación puede ser usada para determinar la concentración de oxígeno en rangos desde 14.621 mg/l a 0°C hasta 6.949 mg/l a 35 °C. Determinar T como una función de la concentración dada de oxígeno, usando el método de bisección. Probar esta función para $O_{sf} = 8, 10$ y 14 mg/l.

Solución.

Considerando la variable independiente $t = x$, y $O_{sf} = y$, se puede usar el código en Octave incluido en el material del presente libro.

➤ Usando comando de Octave

```
>> fx= @(x) -139.34411+1.575701e5/x-6.642308e7/x.^2+1.243800e10/x.^3-
8.621949e11/x.^4-log(8);
>> fzero(fx,273)
ans = 299.93
>> fzero(fx,273.5)
ans = 299.93
>>

>> fx= @(x) -139.34411+1.575701e5/x-6.642308e7/x.^2+1.243800e10/x.^3-
8.621949e11/x.^4-log(10);
>> fzero(fx,273.5)
ans = 288.54
>>

>> fx= @(x) -139.34411+1.575701e5/x-6.642308e7/x.^2+1.243800e10/x.^3-
8.621949e11/x.^4-log(14);
>> fzero(fx,273.5)
ans = 274.70
>>
```

➤ Aplicando Bisección.

Método de Bisección.

Introduce una función con respecto a x $-139.34411+1.575701e5/x-6.642308e7/x^2+1.243800e10/x^3-8.621949e11/x^4-\log(8)$
 Valores para graficar la $f(x)$

```
x inicial? 273
x final? 308
```

Método de Bisección.

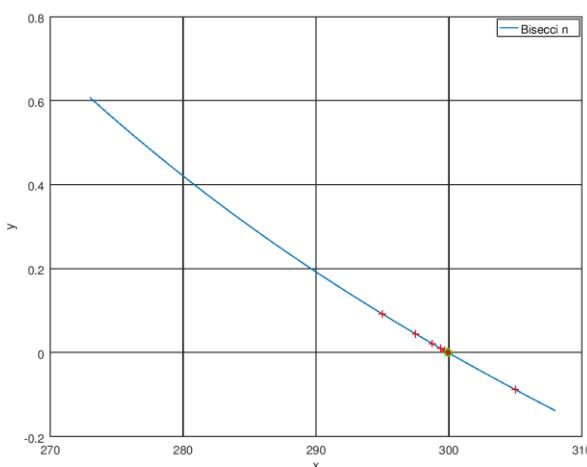
Datos:

```
a = 295
b = 305
Iteraciones máximas = 100
Tolerancia = 0.0001
```

i	a	b	xm	$f(b) * f(xm)$	$f(xm)$
0	295.000000	305.000000	300.000000	0.000111	-0.001255
1	295.000000	300.000000	297.500000	-0.000056	0.044433
2	297.500000	300.000000	298.750000	-0.000027	0.021392
3	298.750000	300.000000	299.375000	-0.000013	0.010021
4	299.375000	300.000000	299.687500	-0.000005	0.004371
5	299.687500	300.000000	299.843750	-0.000002	0.001555
6	299.843750	300.000000	299.921875	-0.000000	0.000149
7	299.921875	300.000000	299.960938	0.000001	-0.000553
8	299.921875	299.960938	299.941406	0.000000	-0.000202
9	299.921875	299.941406	299.931641	0.000000	-0.000027

Raíz: 299.931641

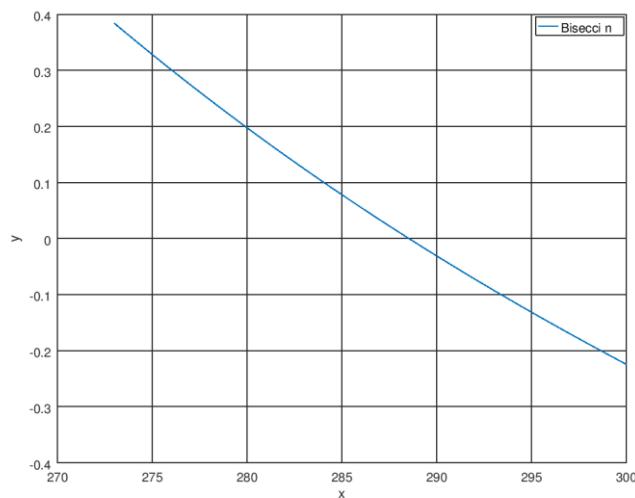
>>



Método de Bisección.

Introduce una función con respecto a x $-139.34411+1.575701e5/x-6.642308e7/x^2+1.243800e10/x^3-8.621949e11/x^4-\log(10)$
 Valores para graficar la $f(x)$

```
x inicial? 273
x final? 300
```



Método de Bisección.

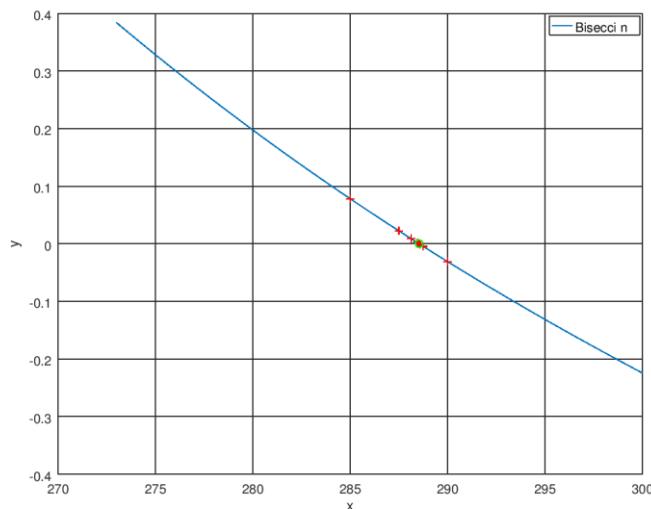
Datos:

```
a = 285
b = 290
Iteraciones máximas = 50
Tolerancia = 0.0001
```

i	a	b	xm	f (b) * f (xm)	f (xm)
0	285.000000	290.000000	287.500000	-0.000695	0.022459
1	287.500000	290.000000	288.750000	0.000140	-0.004532
2	287.500000	288.750000	288.125000	-0.000040	0.008891
3	288.125000	288.750000	288.437500	-0.000010	0.002161
4	288.437500	288.750000	288.593750	0.000005	-0.001190
5	288.437500	288.593750	288.515625	-0.000001	0.000484
6	288.515625	288.593750	288.554688	0.000000	-0.000353
7	288.515625	288.554688	288.535156	-0.000000	0.000065

Raíz:**288.535156**

>>



Método de Bisección.

Introduce una función con respecto a x $-139.34411+1.575701e5/x-6.642308e7/x^2+1.243800e10/x^3-8.621949e11/x^4-\log(14)$
Valores para graficar la $f(x)$

```
x inicial? 273
x final? 300
```

Método de Bisección.

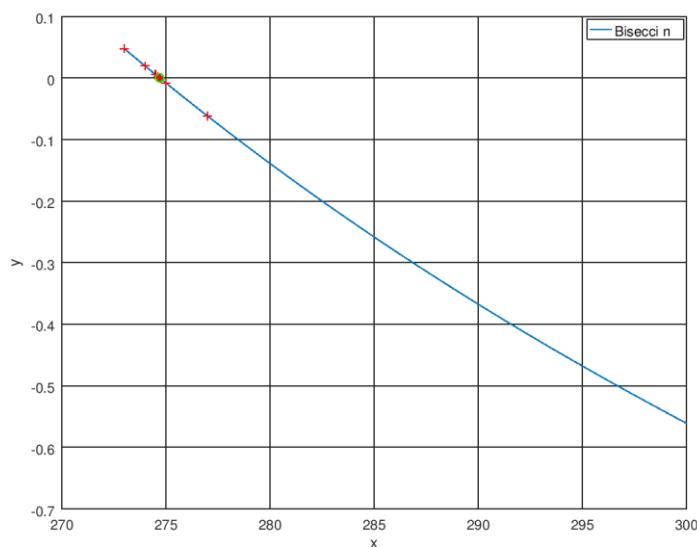
Datos:

```
a = 273
b = 277
Iteraciones máximas = 50
Tolerancia = 0.0001
```

i	a	b	xm	$f(b) * f(xm)$	$f(xm)$
0	273.000000	277.000000	275.000000	0.000501	-0.008105
1	273.000000	275.000000	274.000000	-0.000158	0.019515
2	274.000000	275.000000	274.500000	-0.000046	0.005642
3	274.500000	275.000000	274.750000	0.000010	-0.001247
4	274.500000	274.750000	274.625000	-0.000003	0.002194
5	274.625000	274.750000	274.687500	-0.000001	0.000473
6	274.687500	274.750000	274.718750	0.000000	-0.000387
7	274.687500	274.718750	274.703125	-0.000000	0.000043

Raíz:**274.703125**

>>



Resumiendo, los resultados (cercanos a 300) son:

<i>Osf, mg/l</i>	8	10	14
<i>T, °K</i>	300	289	275

B.2² Un proyecto de diseño de ingeniería química requiere estimar con precisión el volumen molal (*y*) del alcohol etílico para un número de diferentes temperaturas y combinaciones de presión para poder seleccionar recipientes de contención apropiados. Se sabe que la ley de los gases ideales se representa por la siguiente fórmula:

$$pV = nRT$$

donde *p* es la presión absoluta, *V* es el volumen, *n* es el número de moles, *R* es la constante universal de los gases y *T* es la temperatura absoluta.

Una ecuación alternativa del estado de los gases está dada por,

$$\left(p + \frac{a}{v^2} \right) (v - b) = RT$$

Conocida como la ecuación de van der Waals, donde *v* = *V/N* es el volumen molal y *a* y *b* son constantes empíricas que dependen de un gas en particular.

Considerar el alcohol etílico (*a* = 12.02 y *b* = 0.08407) a una temperatura de 375 °K y *p* = 2.0 atm. Comparar los resultados con la ley del estado ideal.

Solución.

- Ley de los gases ideales.

$$v = \frac{V}{n} = \frac{RT}{p} = 0.082054 \frac{L \text{ atm}}{\text{mol } ^\circ\text{K}} \frac{375^\circ\text{K}}{2 \text{ atm}} = 15.3851^\circ\text{K/mol}$$

- van der Waals

$$f(v) = \left(p + \frac{a}{v^2} \right) (v - b) - RT = 0$$

$$f(v) = \left(2 + \frac{12.02}{v^2} \right) (v - 0.08407) - 0.082054(375)$$

Aplicando bisección con el código de Octave se obtiene lo siguiente:

Método de Bisección.

```
Introduce una función con respecto a x  (2+12.02/x^2)*(x-0.08407)-30.7703
Valores para graficar la f(x)
```

```
x inicial? 10
x final? 30
```

Método de Bisección.

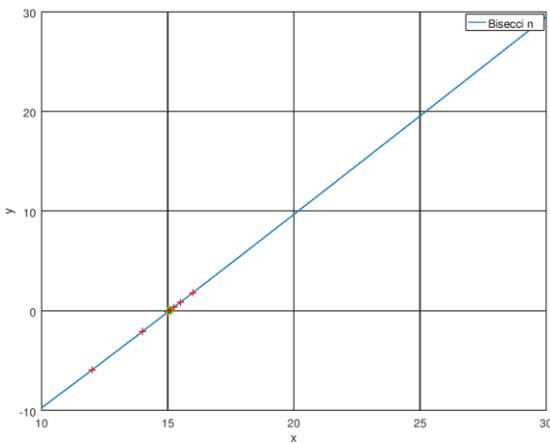
Datos:

```
a = 12
b = 16
Iteraciones máximas = 20
Tolerancia = 0.0001
```

i	a	b	xm	f(b) * f(xm)	f(xm)
0	12.000000	16.000000	14.000000	-3.771523	-2.085024
1	14.000000	16.000000	15.000000	-0.256131	-0.141598
2	15.000000	16.000000	15.500000	1.506489	0.832838
3	15.000000	15.500000	15.250000	0.287672	0.345412
4	15.000000	15.250000	15.125000	0.035181	0.101853
5	15.000000	15.125000	15.062500	-0.002025	-0.019886
6	15.062500	15.125000	15.093750	0.004174	0.040981
7	15.062500	15.093750	15.078125	0.000432	0.010547
8	15.062500	15.078125	15.070312	-0.000049	-0.004670
9	15.070312	15.078125	15.074219	0.000031	0.002938
10	15.070312	15.074219	15.072266	-0.000003	-0.000866
11	15.072266	15.074219	15.073242	0.000003	0.001036
12	15.072266	15.074219	15.072754	0.000000	0.000085

Raíz: **15.072754**

>>



Y con el código en Octave para Newton-Raphson,

Método de Newton-Raphson.

Introduce una función con respecto a x $(2+12.02/x^2) * (x-0.08407) - 30.7703$

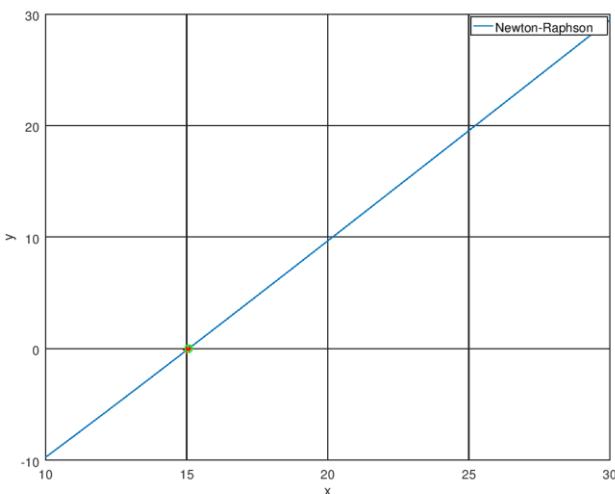
Valores para graficar la $f(x)$

desde x = 10
hasta x = 30

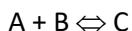
x inicial? 15
tolerancia? 0.0001
Máximas iteraciones? 20

i	x	$f(x)$	$f'(x)$	delta
0	15.000000	-0.141600	1.947177	0.07272093
1	15.072721	0.000018	1.947682	-0.00000946
2	15.072711			

>>



B3². Una reacción química reversible



Puede caracterizarse por la relación de equilibrio

$$K = \frac{c_c}{c_a^2 c_b}$$

donde la nomenclatura c_i representa la concentración del constituyente i . Donde x representa el número de moles producidos C . La ley de conservación de masa puede ser usada para reformular la ecuación de equilibrio como

$$K = \frac{(c_{c,0} + x)}{(c_{a,0} - 2x)^2 (c_{b,0} - x)}$$

donde el subíndice 0 designa la concentración inicial de cada constituyente. Si $K = 0.015$, $c_{a,0} = 42$, $c_{b,0} = 30$ y $c_{c,0} = 4$, determinar el valor de x .

Solución.

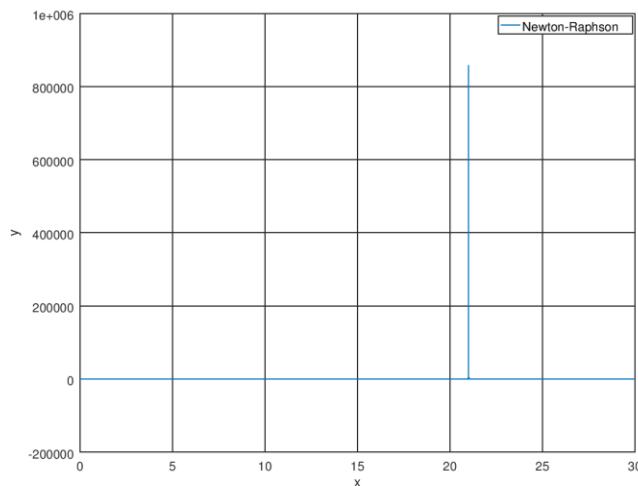
Resolviendo con Newton-Raphson,

Método de Newton-Raphson.

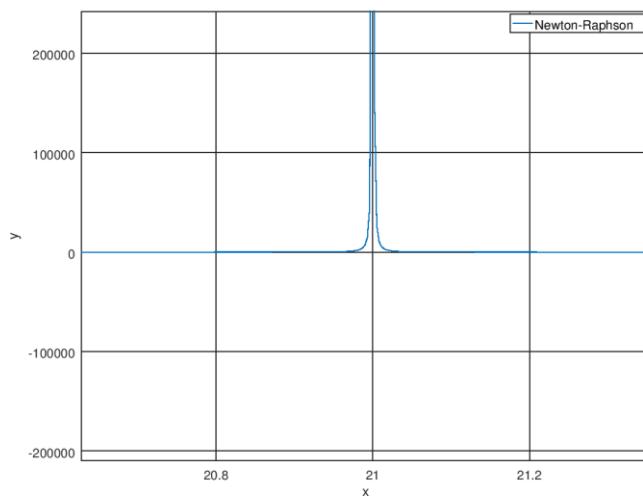
```
Introduce una función con respecto a x      (4+x) / ((42-2*x)^2*(30-x))-0.015
```

Valores para graficar la $f(x)$

```
desde x = 0
hasta x = 30
```



Haciendo un acercamiento,



x inicial? 20.5

Tolerancia? 0.0001

Máximas iteraciones? 20

i	x	$f(x)$	$f'(x)$	delta
0	20.500000	2.563947	10.692521	-0.23978886
1	20.260211	1.122809	3.239759	-0.34657187
2	19.913639	0.487230	0.995406	-0.48947926
3	19.424160	0.207979	0.313601	-0.66319844

```

4 18.760962 0.085990 0.103631 -0.82977106
5 17.931191 0.033239 0.037635 -0.88319798
6 17.047993 0.011012 0.016408 -0.67114017
7 16.376852 0.002495 0.009711 -0.25695548
8 16.119897 0.000217 0.008089 -0.02676696
9 16.093130 0.000002 0.007940 -0.00025155
10 16.092878 0.000000 0.007939 -0.00000002
11 16.092878

```

>>

B4². La caída de presión en una sección de tubo puede ser calculada como

$$\Delta p = f \frac{L \rho V^2}{2D}$$

donde Δp = caída de presión (Pa), f = factor de fricción, L = longitud del tubo (m), ρ = densidad (kg/m^3), V = velocidad (m/s) y D = diámetro (m). Para flujo turbulento, la ecuación de Colebrook provee un medio para calcular el factor de fricción

$$\frac{1}{\sqrt{f}} = -2.0 \log \left(\frac{\epsilon}{3.7D} + \frac{2.51}{Re \sqrt{f}} \right)$$

donde ϵ = rugosidad (m) y Re = número de Reynolds,

$$Re = \frac{\rho V D}{\mu}$$

donde μ = viscosidad dinámica (N s/m^2)

- (a) Determinar Δp para un tubo horizontal de 0.2 m. de largo de tubería estrecha lisa dado $\rho = 1.23 \text{ kg/m}^3$, $\mu = 1.79 \times 10^{-5} \text{ N s/m}^2$, $D = 0.005 \text{ m}$, $V = 40 \text{ m/s}$, y $\epsilon = 0.0015 \text{ mm}$. Utilizar un método numérico para determinar el factor de fricción. Observar que para los tubos lisos con $Re < 10^5$, una buena suposición inicial se puede obtener usando la fórmula de Blasius, $f = 0.316/Re^{0.25}$.

$$\Delta p = f \frac{L \rho V^2}{2D} = f \frac{(0.2)(1.23)(40)^2}{2(0.005)} = 39360 f$$

$$Re = \frac{(1.23)(40)(0.005)}{1.79 \times 10^{-5}} = 13743$$

$$f(f) = -2.0 \log \left(\frac{\varepsilon}{3.7D} + \frac{2.51}{\operatorname{Re} \sqrt{f}} \right) - \frac{1}{\sqrt{f}} = -2.0 \log \left(\frac{0.0015}{3.7(0.005)} + \frac{2.51}{(13743)\sqrt{f}} \right) - \frac{1}{\sqrt{f}} = 0$$

Aplicando Newton-Raphson,

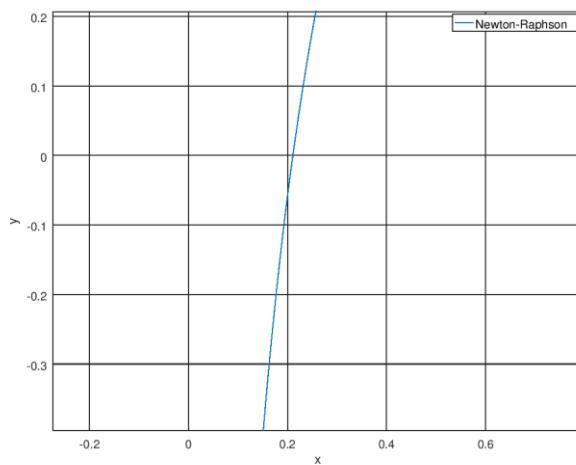
Método de Newton-Raphson.

Introduce una función con respecto a x

$$2 * \log10(0.0015 / (3.7 * 0.005) + 2.51 / (13743 * \sqrt{x})) - 1 / \sqrt{x}$$

Valores para graficar la f(x)

desde x = 0
 hasta x = 0.5



x inicial? 0.2
 tolerancia? 0.0001
 Máximas iteraciones? 50

i	x	f(x)	f'(x)	delta
0	0.200000	-0.058271	5.601052	0.01040358
1	0.210404	-0.002179	5.190809	0.00041980
2	0.210823	-0.000003	5.175313	0.00000063
3	0.210824			

>>

Entonces,

$$\Delta p = f \frac{L \rho V^2}{2D} = f \frac{(0.2)(1.23)(40)^2}{2(0.005)} = 39360 f = 39360(0.210824) = 8298 \text{ Pa}$$

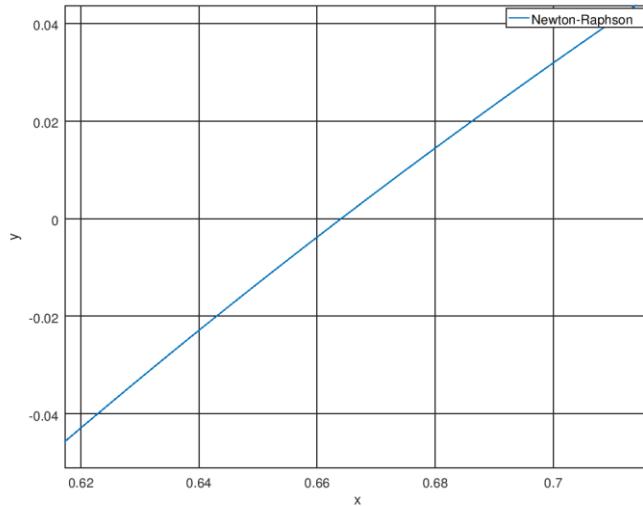
(b) Repetir el cálculo para un tubo de acero comercial ($\varepsilon = 0.045 \text{ mm}$).

Método de Newton-Raphson.

Introduce una función con respecto a x -
 $2*\log10(0.0045/(3.7*0.005)+2.51/(13743*sqrt(x)))-1/sqrt(x)$

Valores para graficar la $f(x)$

desde x = 0.5
 hasta x = 1



x inicial? 0.66
 tolerancia? 0.0001
 Máximas iteraciones? 20

i	x	f(x)	f'(x)	delta
0	0.660000	-0.003799	0.933119	0.00407117
1	0.664071	-0.000017	0.924551	0.00001891
2	0.664090			

>>

$$\Delta p = f \frac{L \rho V^2}{2 D} = f \frac{(0.2)(1.23)(40)^2}{2(0.005)} = 39360 f = 39360(0.66409) = 26139 \text{ Pa}$$

B.5². La concentración de una bacteria contaminante c en un lago disminuye según

$$c = 70e^{-1.5t} + 25e^{-0.075t}$$

Determinar el tiempo requerido para que la concentración de la bacteria sea reducida a 9.

Solución.

```
>> fc= @(t) 70*exp (-1.5*t)+25*exp (-0.075*t)-9;
>> fzero(fc,5)
ans = 13.622
>>
```

Aplicando el método de Bisección,

Método de Bisección.

Introduce una función con respecto a x $70 \cdot \exp(-1.5 \cdot x) + 25 \cdot \exp(-0.075 \cdot x) - 9$
 Valores para graficar la $f(x)$

```
x inicial? 0
x final? 20
```

Método de Bisección.

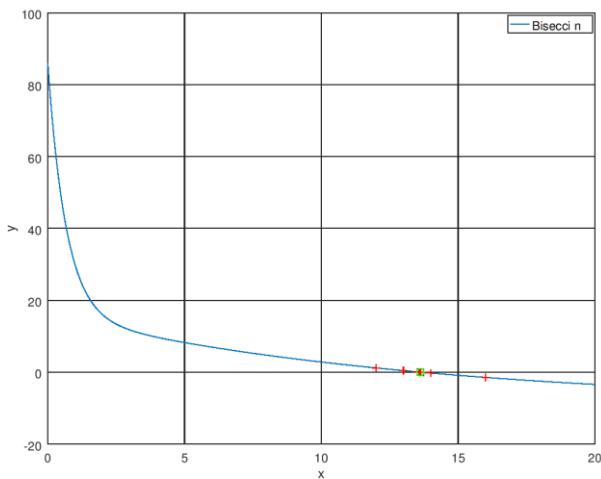
Datos:

```
a = 12
b = 16
Iteraciones máximas = 50
Tolerancia = 0.0001
```

i	a	b	xm	$f(b) * f(xm)$	$f(xm)$
0	12.000000	16.000000	14.000000	0.369824	-0.251556
1	12.000000	14.000000	13.000000	-0.108121	0.429809
2	13.000000	14.000000	13.500000	-0.020814	0.082739
3	13.500000	14.000000	13.750000	0.021628	-0.085975
4	13.500000	13.750000	13.625000	0.000173	-0.002013
5	13.500000	13.625000	13.562500	-0.000081	0.040264
6	13.562500	13.625000	13.593750	-0.000038	0.019100
7	13.593750	13.625000	13.609375	-0.000017	0.008537
8	13.609375	13.625000	13.617188	-0.000007	0.003260
9	13.617188	13.625000	13.621094	-0.000001	0.000623
10	13.621094	13.625000	13.623047	0.000001	-0.000695
11	13.621094	13.623047	13.622070	0.000000	-0.000036

Raíz: **13.622070**

>>



Y el de Newton-Raphson,

Método de Newton-Raphson.

Introduce una función con respecto a x $70 \cdot \exp(-1.5 \cdot x) + 25 \cdot \exp(-0.075 \cdot x) - 9$

Valores para graficar la $f(x)$

```

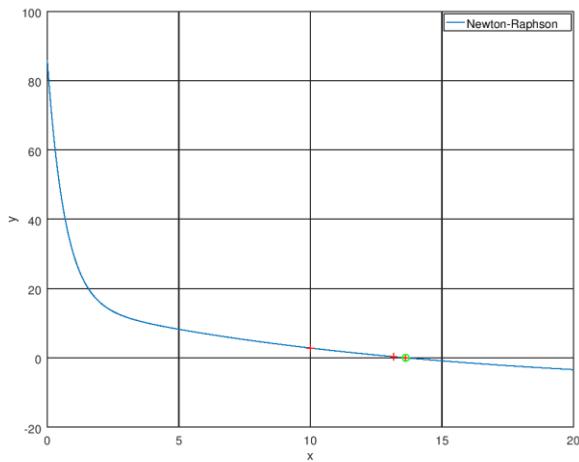
desde x = 0
hasta x = 20

x inicial? 10
tolerancia? 0.0001
Máximas iteraciones? 50

      i          x        f(x)        f'(x)       delta
0  10.000000  2.809185  -0.885719  3.17164241
1  13.171642  0.309195  -0.698190  0.44285285
2  13.614495  0.005078  -0.675381  0.00751939
3  13.622015  0.000001  -0.675000  0.00000212
4  13.622017

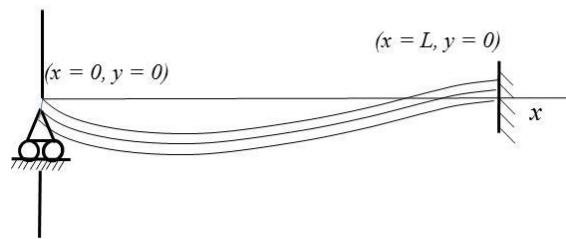
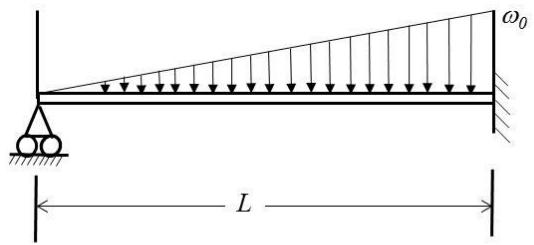
>>

```



B.6². La figura muestra una viga uniforme sujeta a una carga distribuida linealmente incrementada. La ecuación para la curva elástica resultante es:

$$y = \frac{\omega_0}{120EI L} \left(-x^5 + 2L^2 x^3 - L^4 x \right)$$



Determinar el punto de máxima deflexión (que es el valor de x donde $dy/dx = 0$). Usar los siguientes valores de los parámetros en el cálculo: $L = 450$ cm, $E = 50,000$ kN/cm², $I = 30,000$ cm⁴ y $\omega_0 = 1.75$ kN/cm.

Entonces

$$\frac{dy}{dx} = \frac{w_0}{120EI L} (-5x^4 + 6L^2 x^2 - L^4) = 0$$

Solución.

Aplicando Newton-Raphson,

Método de Newton-Raphson.

Introduce una función con respecto a x $(1.75 / (120 * 50000 * 30000 * 450)) * (-5 * x^4 + 6 * 450^2 * x^2 - 450^4)$

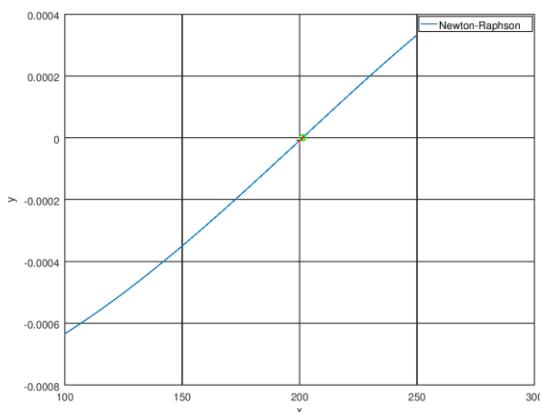
Valores para graficar la $f(x)$

```
desde x = 100
hasta x = 250

x inicial? 200
tolerancia? 0.0001
Máximas iteraciones? 50

      i          x        f(x)        f'(x)        delta
  0 200.000000  -0.000009  0.000007  1.24616564
  1 201.246166    0.000000  0.000007 -0.00004767
  2 201.246118
```

>>



Y con Bisección,

Método de Bisección.

```
Introduce una función con respecto a x  (1.75/(120*50000*30000*450))*(-5*x^4+6*450^2*x^2-450^4)
Valores para graficar la f(x)
```

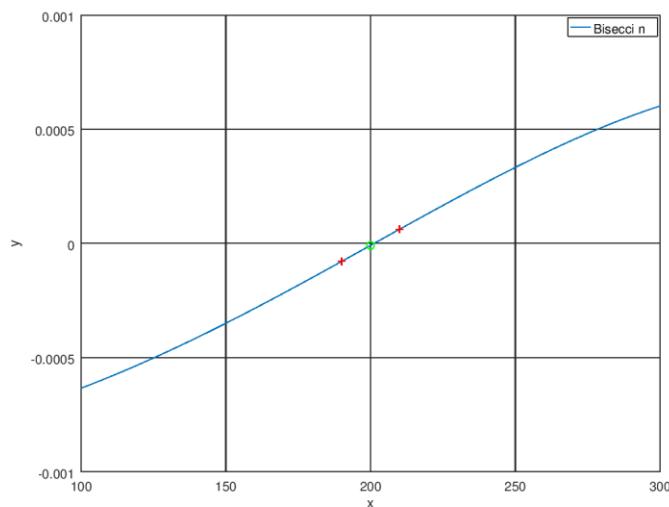
```
x inicial? 100
x final? 300
```

Método de Bisección.

Datos:

```
a = 190
b = 210
Iteraciones máximas = 50
Tolerancia = 0.0001
```

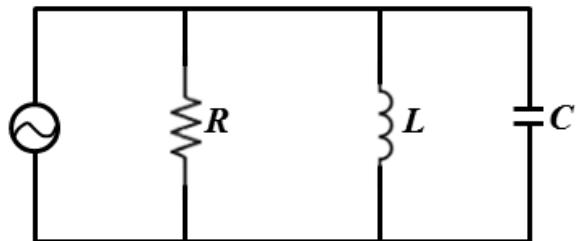
```
i          a          b          xm        f (b) *f (xm)      f (xm)
0 190.000000 210.000000 200.000000 -0.000000 -0.000009
Raíz:200.000000
>>
```



B.7². La figura muestra un circuito con un resistor, un inductor y un capacitor en paralelo. Las reglas de Kirchhoff pueden usarse para expresar la impedancia del sistema como

$$\frac{1}{Z} = \sqrt{\frac{1}{R^2} + \left(\omega C - \frac{1}{\omega L} \right)}$$

donde Z = impedancia (Ω) y ω = frecuencia angular. Encontrar ω que resulta en una impedancia de 75Ω . Usar los siguientes parámetros: $R = 225 \Omega$, $C = 0.6 \times 10^{-6} F$ y $L = 0.5 H$.



$$f(\omega) = \sqrt{\frac{1}{R^2} + \left(\omega C - \frac{1}{\omega L} \right)} - \frac{1}{Z} = 0$$

Solución.

Aplicando Newton-Raphson,

Método de Newton-Raphson.

```
Introduce una función con respecto a x    sqrt((1/225^2)+(0.6e-6*x-1/(0.5*x)))-1/75
```

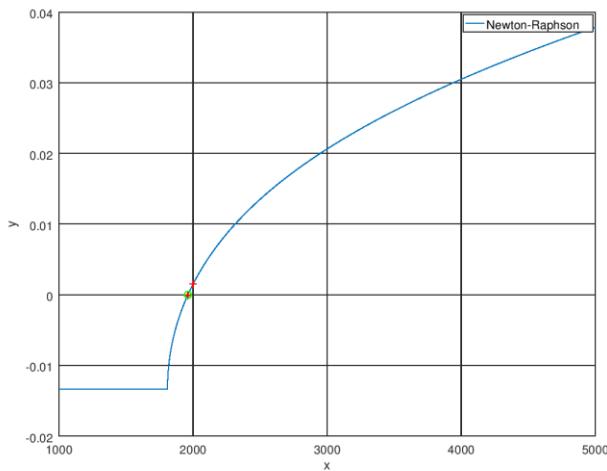
Valores para graficar la $f(x)$

```
desde x = 1000
hasta x = 5000

x inicial? 2000
tolerancia? 0.0001
Máximas iteraciones? 50
```

i	x	f(x)	f'(x)	delta
0	2000.000000	0.001491	0.000037	-40.17958868
1	1959.820411	-0.000099	0.000042	2.34167009
2	1962.162081	-0.000000	0.000042	0.01008130
3	1962.172163	-0.000000	0.000042	0.00000018
4	1962.172163			

>>



Y con Bisección,

Método de Bisección.

Introduce una función con respecto a x `sqrt((1/225^2)+(0.6e-6*x-1/(0.5*x)))-1/75`
Valores para graficar la f(x)

```
x inicial? 1000
x final? 5000
```

Método de Bisección.

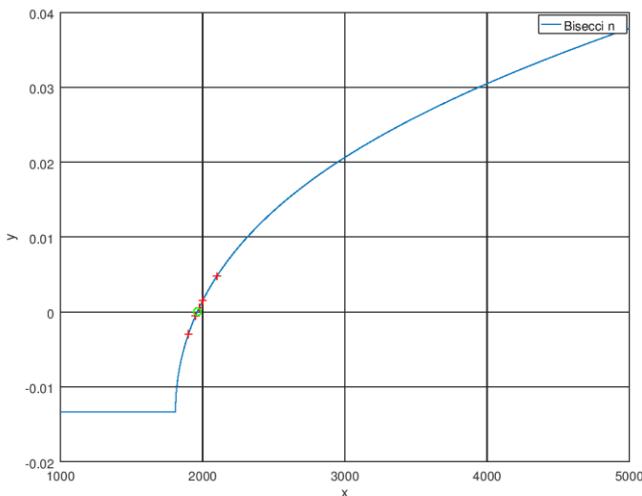
Datos:

```
a = 1900
b = 2100
Iteraciones máximas = 100
Tolerancia = 0.0001
```

i	a	b	xm	f (b) *f (xm)	f (xm)
0	1900.000000	2100.000000	2000.000000	0.000007	0.001491
1	1900.000000	2000.000000	1950.000000	-0.000001	-0.000523
2	1950.000000	2000.000000	1975.000000	0.000001	0.000526
3	1950.000000	1975.000000	1962.500000	0.000000	0.000014

Raíz:1962.500000

>>



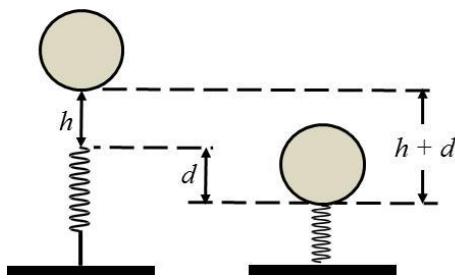
B.8². Los sistemas mecánicos reales pueden involucrar la deflección de resortes no lineales. En la figura una masa m es arrojada una distancia h sobre sobre un resorte no lineal. La fuerza de resistencia F del resorte está dada por

$$F = -(k_1 d + k_2 d^{3/2})$$

La conservación de energía puede ser usada para mostrar que

$$0 = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2}k_1 d^2 - m g d - m g h$$

Resolver para d , dados los siguientes valores de los parámetros: $k_1 = 40,000 \text{ g/s}^2$, $k_2 = 40 \text{ g/(s}^2 \text{ m}^{0.5}\text{)}$, $m = 95 \text{ g}$, $g = 9.81 \text{ m/s}^2$ y $h = 0.43 \text{ m}$.



Solución.

Aplicando Newton-Raphson,

Método de Newton-Raphson.

Introduce una función con respecto a x $16*x^{2.5}+20000*x^2-931.95*x-400.7385$

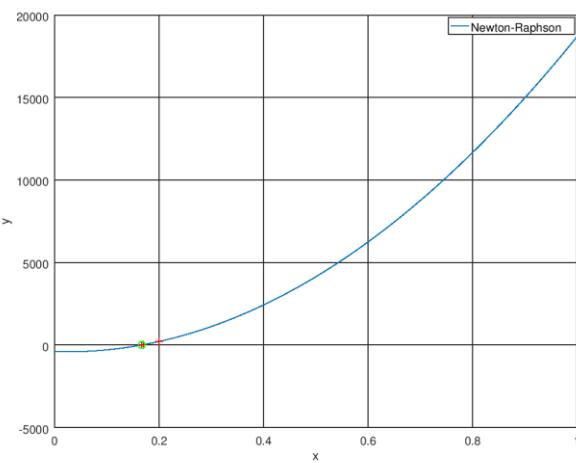
Valores para graficar la $f(x)$

```
desde x = 0
hasta x = 1

x inicial? 0.2
tolerancia? 0.0001
Máximas iteraciones? 20

i          x        f(x)        f'(x)        delta
0  0.200000  213.157755  7071.627709  -0.03014267
1  0.169857   18.183491  5865.143299  -0.00310026
2  0.166757    0.192351  5741.056436  -0.00003350
3  0.166724
```

>>



Y con Bisección,

Método de Bisección.

Introduce una función con respecto a x $16*x^{2.5}+20000*x^2-931.95*x-400.7385$
Valores para graficar la $f(x)$

x inicial? 0

x final? 1

Método de Bisección.

Datos:

```
a = 0.1
b = 0.2
Iteraciones máximas = 50
Tolerancia = 0.0001
```

i	a	b	xm	f (b) *f (xm)	f (xm)
0	0.100000	0.200000	0.150000	-19267.661224	-90.391573
1	0.150000	0.200000	0.175000	10418.132794	48.875232
2	0.150000	0.175000	0.162500	-1167.387832	-23.885060
3	0.162500	0.175000	0.168750	572.492919	11.713355
4	0.162500	0.168750	0.165625	-73.574911	-6.281284
5	0.165625	0.168750	0.167188	31.241590	2.667177
6	0.165625	0.167188	0.166406	-4.852310	-1.819268
7	0.166406	0.167188	0.166797	1.122617	0.420901
8	0.166406	0.166797	0.166602	-0.294608	-0.699947
9	0.166602	0.166797	0.166699	-0.058806	-0.139714
10	0.166699	0.166797	0.166748	0.059156	0.140546
11	0.166699	0.166748	0.166724	0.000057	0.000404
12	0.166699	0.166724	0.166711	-0.000028	-0.069658
13	0.166711	0.166724	0.166718	-0.000014	-0.034628
14	0.166718	0.166724	0.166721	-0.000007	-0.017112
15	0.166721	0.166724	0.166722	-0.000003	-0.008354
16	0.166722	0.166724	0.166723	-0.000002	-0.003975
17	0.166723	0.166724	0.166723	-0.000001	-0.001786
18	0.166723	0.166724	0.166723	-0.000000	-0.000691
19	0.166723	0.166724	0.166724	-0.000000	-0.000143
20	0.166724	0.166724	0.166724	0.000000	0.000130
21	0.166724	0.166724	0.166724	-0.000000	-0.000007

Raíz: **0.166724**

>>

B.9². La velocidad ascendente de un cohete puede calcularse por medio de la siguiente fórmula:

$$v = u \ln\left(\frac{m_0}{m_0 - qt}\right) - gt$$

donde v = velocidad ascendente, u = velocidad a la cual el combustible es expelido del cohete, m_0 = masa inicial del cohete en el tiempo $t = 0$, q = tasa de consumo de combustible y g = aceleración de la gravedad. Si $u = 2200$ m/s, $m_0 = 160,000$ kg y $q = 2680$ kg/s. Calcular el tiempo en el cual $v = 1000$ m/s.

Solución.

$$f(t) = u \ln\left(\frac{m_0}{m_0 - qt}\right) - gt - v$$

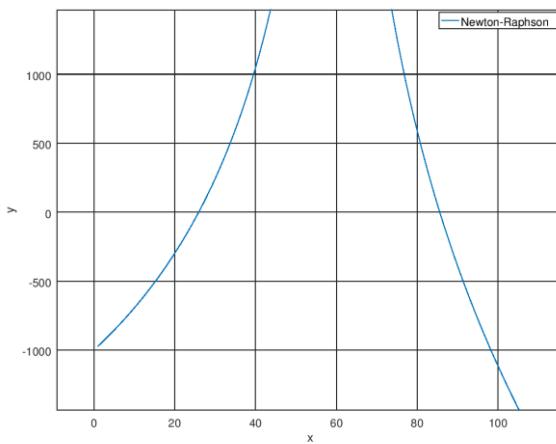
$$f(t) = 2200 \ln\left(\frac{160000}{160000 - 2680t}\right) - 9.81t - 1000$$

Método de Newton-Raphson.

```
Introduce una función con respecto a x    2200*log(160000/(160000-2680*x))-9.81*x-1000
```

Valores para graficar la f(x)

```
desde x = 1
hasta x = 1000
```



```
x inicial? 30
tolerancia? 0.0001
Máximas iteraciones? 50
```

i	x	f(x)	f'(x)	delta
0	30.000000	241.651389	64.260352	-3.76050524
1	26.239495	16.273122	55.936224	-0.29092278
2	25.948572	0.082668	55.369545	-0.00149302
3	25.947079	0.000002	55.366662	-0.00000004
4	25.947079			

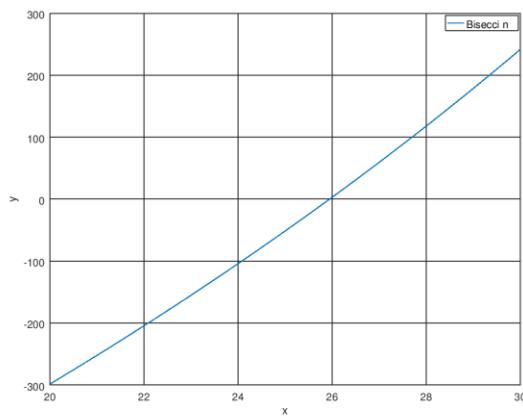
>>

Aplicando Bisección,

Método de Bisección.

```
Introduce una función con respecto a x 2200*log(160000/(160000-2680*x))-9.81*x-1000
Valores para graficar la f(x)
```

```
x inicial? 20
x final? 30
```



Método de Bisección.

Datos:

```
a = 25
b = 27
Iteraciones máximas = 50
Tolerancia = 0.0001
```

i	a	b	xm	f (b) *f (xm)	f (xm)
0	25.000000	27.000000	26.000000	174.176795	2.932770
1	25.000000	26.000000	25.500000	-72.034636	-24.561980
2	25.500000	26.000000	25.750000	-31.891671	-10.874249
3	25.750000	26.000000	25.875000	-11.689319	-3.985761
4	25.875000	26.000000	25.937500	-1.555144	-0.530265
5	25.937500	26.000000	25.968750	3.520229	1.200309
6	25.937500	25.968750	25.953125	0.401847	0.334786
7	25.937500	25.953125	25.945312	-0.032741	-0.097798
8	25.945312	25.953125	25.949219	0.039665	0.118479
9	25.945312	25.949219	25.947266	0.001225	0.010337

```

10 25.945312 25.947266 25.946289 -0.000452 -0.043732
11 25.946289 25.947266 25.946777 -0.000173 -0.016698
12 25.946777 25.947266 25.947021 -0.000033 -0.003180
13 25.947021 25.947266 25.947144 0.000037 0.003578
14 25.947021 25.947144 25.947083 0.000001 0.000199
15 25.947021 25.947083 25.947052 -0.000000 -0.001491
16 25.947052 25.947083 25.947067 -0.000000 -0.000646
17 25.947067 25.947083 25.947075 -0.000000 -0.000224
18 25.947075 25.947083 25.947079 -0.000000 -0.000012

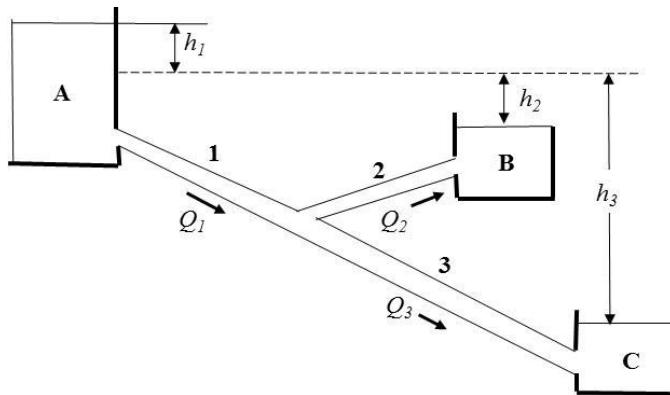
```

Raíz: **25.947079**

>>

B.10². La figura muestra tres reservorios conectados por tubos circulares. Las tuberías, que se hacen del acero cubierto de asfalto ($\epsilon = 0.0012 \text{ m}$), tienen las características siguientes:

Tubo	1	2	3
Longitud, m	1 800	500	1 400
Diámetro, m	0.4	0.25	0.2
Flujo, m^3/s	?	0.1	?



Si las elevaciones de la superficie del agua en los depósitos A y C son 200 y 172.5 m, respectivamente, determinar la elevación en el reservorio B y el flujo en las tuberías 1 y 3. Tener en cuenta que la viscosidad cinemática del agua es de $1 \times 10^{-6} \text{ m}^2/\text{s}$ y utilizar la ecuación Colebrook para determinar el factor de fricción. ($\rho = 1\,000 \text{ kg/m}^3$).

Solución.

$$\Delta p_2 = f \frac{L \rho V^2}{2D} = f \frac{(500)(1000)(2.04)^2}{2(0.25)} = 4.148 \times 10^6 f$$

$$Re_2 = \frac{\rho V D}{\mu} = \frac{(1000)(0.1/0.0491)(0.25)}{1 \times 10^{-6}} = 509.30 \times 10^6$$

$$f(f) = -2.0 \log \left(\frac{\varepsilon}{3.7 D} + \frac{2.51}{Re \sqrt{f}} \right) - \frac{1}{\sqrt{f}} = -2.0 \log \left(\frac{0.0012}{3.7(0.25)} + \frac{2.51}{(509.30 \times 10^6) \sqrt{f}} \right) - \frac{1}{\sqrt{f}} = 0$$

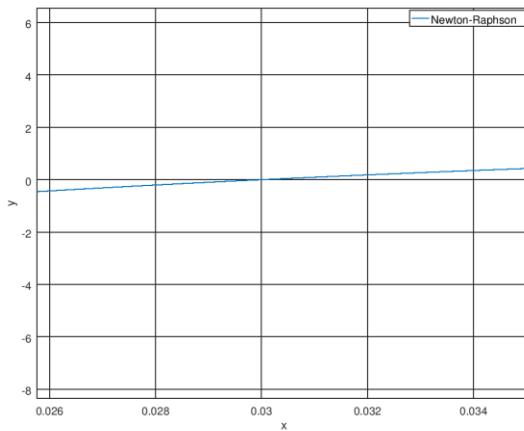
Aplicando Newton-Raphson,

Método de Newton-Raphson.

```
Introduce una función con respecto a x      -
2*log10(0.0012/(3.7*0.25)+2.51/(509.3e6*sqrt(x)))-1/sqrt(x)
```

Valores para graficar la $f(x)$

```
desde x = 0
hasta x = 0.1
```



```
x inicial? 0.03
tolerancia? 0.0001
Máximas iteraciones? 20
```

i	x	f(x)	f'(x)	delta
0	0.030000	0.000399	96.225362	-0.00000415
1	0.029996			

>>

Luego,

$$\Delta p = f \frac{L_2 \rho V_2^2}{2 D_2} = f \frac{(500)(1000)(2.037)^2}{2(0.25)} = 4.1494 \times 10^6 \quad f = 39360(0.03) = 124481 \text{ Pa} = 12.70 \text{ m.}$$

También se pudo haber calculado con Darcy-Weisbach,

$$h_{f_2} = f \frac{L_2}{D_2} \frac{V_2^2}{2g} = 0.03 \frac{500}{0.25} \frac{(2.037)^2}{19.62} = 12.69 \text{ m}$$

$$h_{f_1} = f \frac{L_1}{D_1} \frac{V_1^2}{2g} = 0.03 \frac{1800}{0.40} \frac{V_1^2}{19.62} = 6.8807 V_1^2$$

$$h_{f_3} = f \frac{L_3}{D_3} \frac{V_3^2}{2g} = 0.03 \frac{1400}{0.20} \frac{V_3^2}{19.62} = 10.7034 V_3^2$$

$$h_{f_{A-C}} = 200 - 172.5 = 27.50 \text{ m} = h_{f_1} + h_{f_3} = 6.8807 V_1^2 + 10.7034 V_3^2$$

$$Q_1 = Q_2 + Q_3$$

$$A_1 V_1 = A_2 V_2 + A_3 V_3$$

$$V_1 = \frac{A_2 V_2 + A_3 V_3}{A_1} = \frac{0.1000 + 0.0314 V_3}{0.1257}$$

$$h_{f_{A-C}} = 27.50 \text{ m} = 6.8807 V_1^2 + 10.7034 V_3^2 = 6.8807 \left(\frac{0.10000 + 0.0314 V_3}{0.1257} \right)^2 + 10.7034 V_3^2$$

$$f(V_3) = 6.8807 \left(\frac{0.10000 + 0.0314 V_3}{0.1257} \right)^2 + 10.7034 V_3^2 - 27.5 = 0$$

Método de Newton-Raphson.

Introduce una función con respecto a x
 $6.8807*((0.1+0.0314*x)/0.1257)^2+10.7034*x^2-27.5$

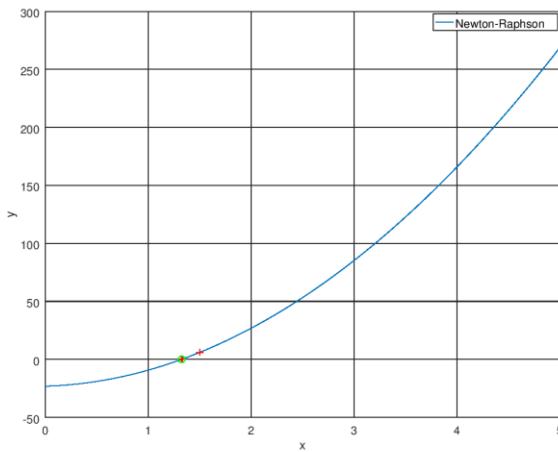
Valores para graficar la $f(x)$

```
desde x = 0
hasta x = 5

x inicial? 1.5
Tolerancia? 0.0001
Máximas iteraciones? 20
```

i	x	f(x)	f'(x)	delta
0	1.500000	6.005605	36.133053	-0.16620808
1	1.333792	0.307544	32.432344	-0.00948263
2	1.324309	0.001001	32.221208	-0.00003107
3	1.324278			

>>



Luego,

$$V_3 = 1.3243 \text{ m/s}$$

$$h_{f_3} = 10.7034(1.3243)^2 = 18.7713 \text{ m}$$

$$V_1 = \frac{0.1000 + 0.0314 V_3}{0.1257} = \frac{0.1000 + 0.0314(1.3243)}{0.1257} = 1.1264 \text{ m}$$

$$h_{f_1} = 6.8807 V_1^2 = 6.8807(1.1264)^2 = 8.7294 \text{ m/s}$$

Entonces,

$$h_{f_{A-C}} = 27.50 \text{ m} = 6.8807 V_1^2 + 10.7034 V_3^2 = 6.8807(1.1264)^2 + 10.7034(1.3243)^2 = 27.5017 \approx 27.5$$

Finalmente,

$$Q_1 = A_1 V_1 = 0.1415 \text{ m}^3/\text{s}, \quad Q_3 = A_3 V_3 = 0.0416 \text{ m}^3/\text{s}, \quad h_B = 200 - 8.7294 - 12.69 = 178.58 \text{ m}$$

B.11². Muchos campos de la ingeniería requieren estimaciones exactas de la población. Por ejemplo, los ingenieros de transporte podrían tener la necesidad de determinar por separado las tendencias de crecimiento de la población de una ciudad y un suburbio adyacente. La población del área urbana está declinando con el tiempo según

$$P_u(t) = P_{u,\max} e^{-k_u t} + P_{u,\min}$$

mientras que la población suburbana está creciendo como

$$P_s(t) = \frac{P_{s,\max}}{1 + (P_{s,\max}/P_0 - 1)e^{-k_s t}}$$

donde $P_{u,\max}$, k_u , $P_{s,\max}$ y k_s = parámetros derivados empíricamente. Determinar el tiempo y los valores correspondientes de $P_u(t)$ y $P_s(t)$ cuando los suburbios son 20% mayores que la ciudad. Los valores de los parámetros son $P_{u,\max} = 75,000$ hab, $k_u = 0.045/\text{año}$, $P_{u,\min} = 100,000$ hab, $P_{s,\max} = 300,000$ hab, $P_0 = 10,000$ hab, $k_s = 0.08/\text{año}$.

Solución.

$$P_s(t) = 1.2 P_u(t)$$

$$\frac{P_{s,\max}}{1 + (P_{s,\max}/P_0 - 1)e^{-k_s t}} = 1.2 \left(P_{u,\max} e^{-k_u t} + P_{u,\min} \right)$$

$$f(t) = \frac{P_{s,\max}}{1 + (P_{s,\max}/P_0 - 1)e^{-k_s t}} - 1.2 \left(P_{u,\max} e^{-k_u t} + P_{u,\min} \right) = 0$$

$$f(t) = \frac{300000}{1 + 29e^{-0.08t}} - 1.2 \left(75000 e^{-0.045t} + 100000 \right) = 0$$

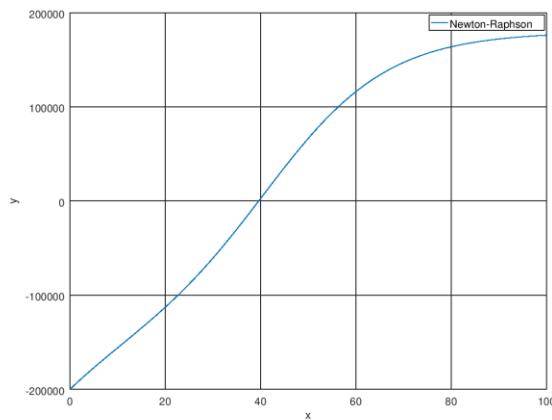
Aplicando Newton-Raphson,

Método de Newton-Raphson.

```
Introduce una función con respecto a x    300000/(1+(29*exp(-0.08*x)))-  
1.2*(75000*exp(-0.045*x))+100000
```

Valores para graficar la $f(x)$

```
desde x = 0  
hasta x = 100
```



```
x inicial? 40  
Tolerancia? 0.0001  
Máximas iteraciones? 20
```

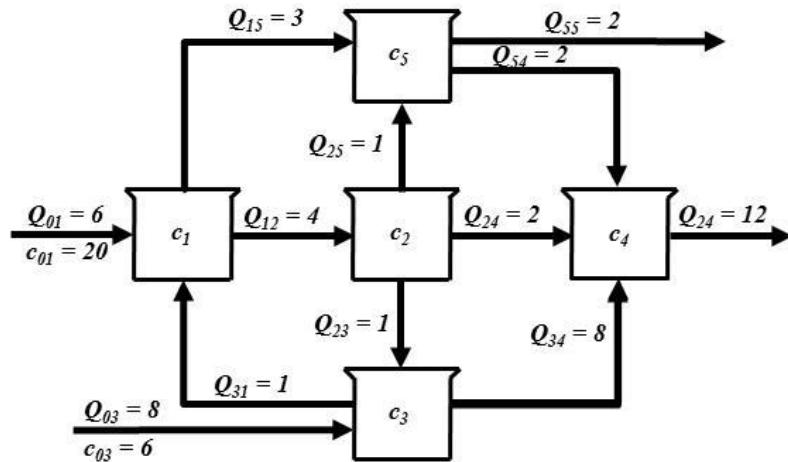
i	x	$f(x)$	$f'(x)$	delta
0	40.000000	2605.095251	6627.673712	-0.39306329
1	39.606937	0.920324	6622.548093	-0.00013897
2	39.606798	0.000000	6622.545813	-0.00000000
3	39.606798			

>>

- $t = 39.61$ año
- $P_u(39.61) = 75000e^{-0.045(39.61)} + 100000 = 112617$ hab
- $P_s(39.61) = \frac{300000}{1 + 29e^{-0.08(39.61)}} = 135162$ hab
- $P_s \cong 1.2 P_u$

Sistemas de ecuaciones lineales.

B12². Se tienen cinco reactores interconectados. Plantear y resolver las cinco ecuaciones diferenciales simultáneas, necesarias para caracterizar el sistema de reactores con flujo estable. Realizar el mismo cálculo del ejemplo 12.1, pero cambiando c_{01} a 20 y c_{03} a 6. También cambiar los siguientes flujos: $Q_{01} = 6$, $Q_{12} = 4$, $Q_{24} = 2$ y $Q_{44} = 12$.



Para el reactor 1:

$$Q_{01} c_{01} + Q_{31} c_3 = Q_{15} c_1 + Q_{12} c_1$$

$$6(20) + 1c_3 = 3c_1 + 4c_1 = 7c_1$$

$$6c_1 - c_3 = 120$$

Reactor 2:

$$Q_{12} c_1 = Q_{23} c_2 + Q_{25} c_2 + Q_{24} c_2$$

$$4c_1 = 1c_2 + 1c_2 + 2c_2$$

$$4c_1 - 4c_2 = 0$$

Reactor 3:

$$Q_{03} c_{03} + Q_{23} c_2 = Q_{31} c_3 + Q_{34} c_3$$

$$8(6) + 1c_2 = 1c_3 + 8c_3$$

$$-c_2 + 9c_3 = 48$$

Reactor 4:

$$Q_{24} c_2 + Q_{34} c_3 + Q_{54} c_5 = Q_{44} c_4$$

$$2c_2 + 8c_3 + 2c_5 = 12c_4$$

$$2c_2 + 8c_3 - 12c_4 + 2c_5 = 0$$

Reactor 5:

$$Q_{15} c_1 + Q_{25} c_2 = Q_{54} c_5 + Q_{55} c_5$$

$$3c_1 + 1c_2 = 2c_5 + 2c_5$$

$$3c_1 + 1c_2 - 4c_5 = 0$$

$$\begin{bmatrix} 7 & 0 & -1 & 0 & 0 \\ 4 & -4 & 0 & 0 & 0 \\ 0 & -1 & 9 & 0 & 0 \\ 0 & 2 & 8 & -12 & 2 \\ 3 & 1 & 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 120 \\ 0 \\ 48 \\ 0 \\ 0 \end{bmatrix}$$

Al tratar de resolver con Gauss-Jordan se tienen contratiempos al proceder con los elementos pivotes, como se puede ver a continuación,

Detected XLS interfaces:

Método de Gauss-Jordan.

$$\begin{array}{cccccc} 7 & 0 & -1 & 0 & 0 & 120 \\ 4 & -4 & 0 & 0 & 0 & 0 \\ 0 & -1 & 9 & 0 & 0 & 48 \\ 0 & 2 & 8 & -12 & 2 & 0 \\ 3 & 1 & 0 & 0 & -4 & 0 \end{array}$$

Sustitución hacia adelante:

1.00000	0.00000	-0.14286	0.00000	0.00000	17.14286
1.00000	-1.00000	0.00000	0.00000	0.00000	0.00000
NaN	-Inf	Inf	NaN	NaN	Inf
NaN	Inf	Inf	-Inf	Inf	NaN
1.00000	0.33333	0.00000	0.00000	-1.33333	0.00000

>>

Entonces, se procede a resolver por medio de la matriz inversa,

A =

7	0	-1	0	0
4	-4	0	0	0
0	-1	9	0	0
0	2	8	-12	2
3	1	0	0	-4

b =

120
0
48
0
0

A_inversa =

0.14516	-0.00403	0.01613	0.00000	0.00000
0.14516	-0.25403	0.01613	0.00000	0.00000
0.01613	-0.02823	0.11290	0.00000	0.00000
0.05914	-0.07224	0.08065	-0.08333	-0.04167
0.14516	-0.06653	0.01613	0.00000	-0.25000

x =

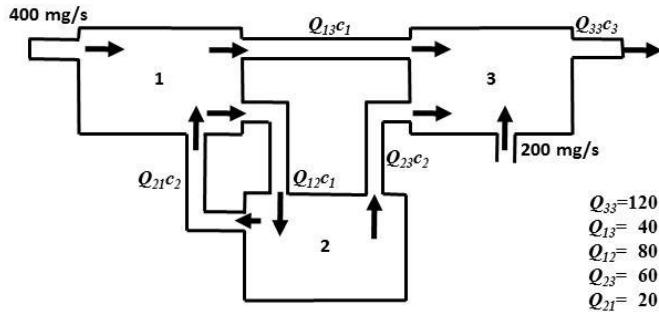
18.1935
18.1935
7.3548
10.9677
18.1935

>>

Con este método si se obtiene la solución. Por otro lado, cada uno de los elementos a_{ij} de la matriz inversa significa el cambio en la concentración del reactor i debido a un cambio unitario en la carga al reactor j . Así, los ceros en la columna 4 indican que una carga al reactor 4 no tendrá impacto en los reactores 1, 2, 3 y 5. Esto es consistente con la configuración del sistema mostrado en la figura, lo que indica que el flujo del reactor 4 no se vuelve a alimentar en ninguno de los otros reactores. Por el contrario, las cargas en cualquiera de los primeros tres reactores afectarán a todo el sistema como se indica en la falta de ceros en las primeras tres columnas. Dicha información es de gran utilidad para los ingenieros que diseñan y administran dichos sistemas.

B.13². La figura muestra tres reactores unidos por tubos. Como se indica, la tasa de transferencia de químicos a través de cada tubo es igual a la tasa de flujo (Q , m³/s) multiplicada por la concentración del reactor del cual se origina el flujo (c , mg/m³). Si el sistema está en un estado

estable, la transferencia a cada reactor equilibrará la transferencia hacia fuera. Desarrollar las ecuaciones de balance para los reactores y resolver las tres ecuaciones algebraicas simultáneas para sus concentraciones.



Haciendo el balance en el sistema de reactores:

Reactor 1:

$$400 + Q_{21} c_2 = Q_{12} c_1 + Q_{13} c_1$$

$$400 + 20c_2 = 80c_1 + 40c_1$$

$$120c_1 - 20c_2 + 0c_3 = 400$$

Reactor 2:

$$Q_{12} c_1 = Q_{21} c_2 + Q_{23} c_2$$

$$80c_1 = 20c_2 + 60c_2$$

$$80c_1 - 80c_2 + 0c_3 = 0$$

Reactor 3:

$$Q_{13} c_1 + Q_{23} c_2 + 200 = Q_{33} c_3$$

$$40c_1 + 60c_2 + 200 = 120c_3$$

$$40c_1 + 60c_2 - 120c_3 = -200$$

➤ Aplicando Gauss-Jordan,

Detected XLS interfaces:

Método de Gauss-Jordan.

```
120   -20      0    400
 80   -80      0     0
 40    60    120   -200
```

Sustitución hacia adelante:

```
1.00000  -0.16667  0.00000  3.33333
1.00000  -1.00000  0.00000  0.00000
1.00000   1.50000  3.00000 -5.00000

1.00000  -0.16667  0.00000  3.33333
0.00000   0.83333  0.00000  3.33333
0.00000  -1.66667  -3.00000  8.33333

1.00000  -0.16667  0.00000  3.33333
0.00000   1.00000  0.00000  4.00000
-0.00000   1.00000  1.80000 -5.00000

1.00000  -0.16667  0.00000  3.33333
0.00000   1.00000  0.00000  4.00000
0.00000   0.00000  -1.80000  9.00000

1.00000  -0.16667  0.00000  3.33333
0.00000   1.00000  0.00000  4.00000
-0.00000  -0.00000  1.00000 -5.00000

1.00000  -0.16667  0.00000  3.33333
0.00000   1.00000  0.00000  4.00000
-0.00000  -0.00000  1.00000 -5.00000
```

Hasta aquí es eliminación de Gauss, previo a Gauss-Jordan. De este paso se puede obtener la solución [4, 4, -5].

Sustitución inversa:

```
Inf  -Inf  NaN  Inf
NaN  Inf  NaN  Inf
-0   -0   1   -5

-Inf  Inf  NaN  -Inf
NaN  -Inf  NaN  -Inf
-0   -0   1   -5

NaN  NaN  NaN  NaN
NaN  NaN  NaN  NaN
-0   -0   1   -5

NaN  NaN  NaN  NaN
NaN  NaN  NaN  NaN
```

```
-0      -0      1      -5
```

Matriz aumentada final:

```
1      NaN      NaN      NaN  
NaN      NaN      NaN      NaN  
-0      -0      1      -5
```

b =

```
NaN  
NaN  
-5
```

>>

➤ Aplicando matriz inversa,

Detected XLS interfaces:

Método de Gauss-Jordan.

A =

```
120    -20      0  
 80    -80      0  
 40     60    120
```

b =

```
400  
 0  
-200
```

A_inversa =

```
0.010000  -0.002500  -0.000000  
0.010000  -0.015000  -0.000000  
-0.008333   0.008333   0.008333
```

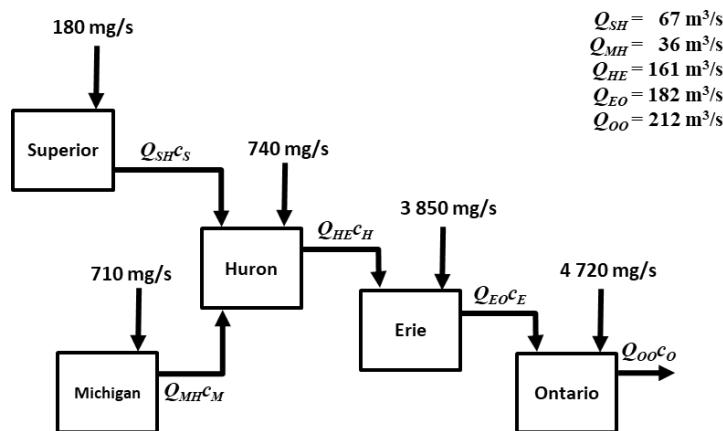
Se aprecia que la carga del reactor
3 no impacta en los reactores 1 y 2.

x =

```
4.0000  
4.0000  
-5.0000
```

>>

B.14². Similar al ejemplo anterior, determinar la concentración de cloruro en cada uno de los grandes lagos usando la información mostrada en la figura.



Solución.

➤ Aplicando Gauss-Jordan:

Detected XLS interfaces:

Método de Gauss-Jordan.

$$\begin{array}{ccccccc} 67 & 0 & 0 & 0 & 0 & 180 \\ 67 & -161 & 36 & 0 & 0 & -740 \\ 0 & 0 & 36 & 0 & 0 & 710 \\ 0 & 161 & 0 & -182 & 0 & -3850 \\ 0 & 0 & 0 & 182 & -212 & -4720 \end{array}$$

Sustitución hacia adelante:

$$\begin{array}{ccccccc} 1.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 2.68657 \\ 1.00000 & -2.40299 & 0.53731 & 0.00000 & 0.00000 & -11.04478 \\ \text{NaN} & \text{NaN} & \text{Inf} & \text{NaN} & \text{NaN} & \text{Inf} \\ \text{NaN} & \text{Inf} & \text{NaN} & -\text{Inf} & \text{NaN} & -\text{Inf} \\ \text{NaN} & \text{NaN} & \text{NaN} & \text{Inf} & -\text{Inf} & -\text{Inf} \end{array}$$

➤ Aplicando matriz inversa:

A =

```
67      0      0      0      0
67   -161     36      0      0
 0      0     36      0      0
 0    161      0   -182      0
 0      0      0    182   -212
```

b =

```
180
-740
710
-3850
-4720
```

A_inversa =

```
0.014925  0.000000 -0.000000  0.000000  0.000000
0.006211 -0.006211  0.006211  0.000000  0.000000
0.000000  0.000000  0.027778  0.000000  0.000000
0.005495 -0.005495  0.005495 -0.005495  0.000000
0.004717 -0.004717  0.004717 -0.004717 -0.004717
```

x =

```
2.6866
10.1242
19.7222
30.1099
48.1132
```

>>

B.15². El bajo río Colorado consiste en una serie de cuatro reservorios como se muestra en la figura. Puede escribirse los balances de masa para cada reservorio, resultando el siguiente sistema de ecuaciones algebraicas simultáneas:

$$\begin{bmatrix} 13.442 & 0 & 0 & 0 \\ -13.442 & 12.252 & 0 & 0 \\ 0 & -12.252 & 12.377 & 0 \\ 0 & 0 & -12.377 & 11.797 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 750.5 \\ 300 \\ 102 \\ 30 \end{bmatrix}$$

donde el vector del miembro derecho consiste en la carga de cloruro en cada uno de los cuatro lagos y c_1, c_2, c_3 y c_4 = concentración de cloruro resultante para los lagos Powell, Mead, Mohave y Havasu, respectivamente.

- (a) Usar la matriz inversa para resolver las concentraciones en cada uno de los lagos.
- (b) ¿Cuánto debe reducirse la carga en el lago Powell para que la concentración del lago Havasu sea 75?

Solución.

Usando código en Octave,

A =

```
13.44200    0.00000    0.00000    0.00000
-13.44200   12.25200   0.00000    0.00000
  0.00000   -12.25200  12.37700   0.00000
  0.00000    0.00000   -12.37700  11.79700
```

b =

```
750.500
300.000
102.000
 30.000
```

A_inversa =

```
0.074394   0.000000   0.000000   0.000000
0.081619   0.081619   0.000000   0.000000
0.080795   0.080795   0.080795   0.000000
0.084767   0.084767   0.084767   0.084767
```

x =

```
55.832 mg/m³ en el lago Powell
85.741 mg/m³ en el lago Mead
93.116 mg/m³ en el lago Mohave
100.237 mg/m³ en el lago Havasu
```

>>

El elemento de la matriz inversa que se debe incluir es el a_{41} , ya que por cada unidad que cambie la carga en el reservorio 1 (Powell), el reservorio 4 (Havasu) cambiará 0.084767. Entonces, para que la carga en Havasu disminuya de 100.237 a 75 se requiere que en el reservorio Powell la carga se reduzca $(100.237-75)/0.084767 = 297.72 \text{ mg/m}^3$.

B.16². Un ingeniero civil implicado en la construcción requiere 4 800, 5 810, y 5 690 m³ de arena, grava fina, y grava gruesa, respectivamente, para un proyecto de edificio. Hay tres bancos de materiales de los cuales pueden ser obtenidos. La composición de estos bancos es

	Arena %	Grava fina %	Grava gruesa %
Banco 1	52	30	18
Banco 2	20	50	30
Banco 2	25	20	55

¿Cuántos metros cúbicos se deben transportar de cada banco para satisfacer las necesidades del ingeniero?

Solución.

$$0.52x + 0.30y + 0.18z = 4800$$

$$0.20x + 0.50y + 0.30z = 5810$$

$$0.25x + 0.20y + 0.55z = 5690$$

Usando código en Octave,

Detected XLS interfaces:

Método de Gauss-Jordan.

```
5.2000e-001 3.0000e-001 1.8000e-001 4.8000e+003
2.0000e-001 5.0000e-001 3.0000e-001 5.8100e+003
2.5000e-001 2.0000e-001 5.5000e-001 5.6900e+003
```

Sustitución hacia adelante:

```
1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
1.0000e+000 2.5000e+000 1.5000e+000 2.9050e+004
1.0000e+000 8.0000e-001 2.2000e+000 2.2760e+004

1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
0.0000e+000 -1.9231e+000 -1.1538e+000 -1.9819e+004
0.0000e+000 -2.2308e-001 -1.8538e+000 -1.3529e+004
```

```

1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
-0.0000e+000 1.0000e+000 6.0000e-001 1.0306e+004
-0.0000e+000 1.0000e+000 8.3103e+000 6.0648e+004

1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
-0.0000e+000 1.0000e+000 6.0000e-001 1.0306e+004
0.0000e+000 0.0000e+000 -7.7103e+000 -5.0342e+004

1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
-0.0000e+000 1.0000e+000 6.0000e-001 1.0306e+004
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

1.0000e+000 5.7692e-001 3.4615e-001 9.2308e+003
-0.0000e+000 1.0000e+000 6.0000e-001 1.0306e+004
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

```

Sustitución inversa:

```

2.8889e+000 1.6667e+000 1.0000e+000 2.6667e+004
-0.0000e+000 1.6667e+000 1.0000e+000 1.7177e+004
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

-2.8889e+000 -1.6667e+000 0.0000e+000 -2.0137e+004
0.0000e+000 -1.6667e+000 0.0000e+000 -1.0647e+004
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

1.7333e+000 1.0000e+000 -0.0000e+000 1.2082e+004
-0.0000e+000 1.0000e+000 -0.0000e+000 6.3885e+003
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

-1.7333e+000 0.0000e+000 0.0000e+000 -5.6940e+003
-0.0000e+000 1.0000e+000 -0.0000e+000 6.3885e+003
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

```

Matriz aumentada final:

```

1.0000e+000 0.0000e+000 0.0000e+000 3.2850e+003
-0.0000e+000 1.0000e+000 -0.0000e+000 6.3885e+003
-0.0000e+000 -0.0000e+000 1.0000e+000 6.5292e+003

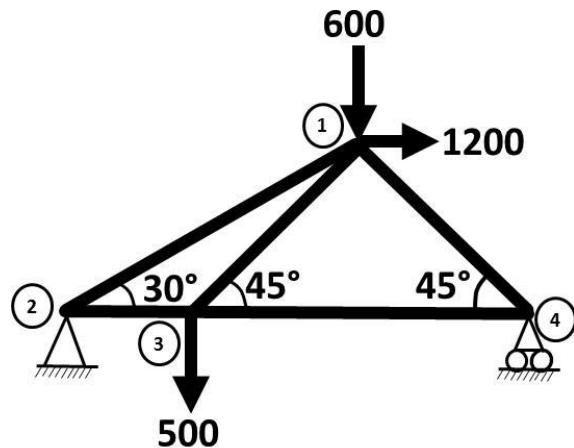
```

b =

3285.0 m³
6388.5 m³
6529.2 m³

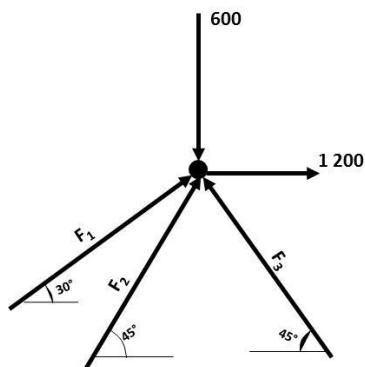
>>

B.17². Determinar las fuerzas y las reacciones asociadas con un entramado estáticamente mostrado en la siguiente figura.



Solución.

Nudo 1.



$$+ \rightarrow \sum F_x = 0$$

$$1200 - F_3 \cos 45^\circ + F_2 \cos 45^\circ - F_1 \cos 30^\circ = 0$$

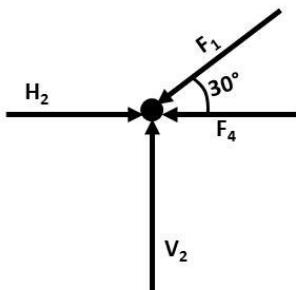
$$0.8660F_1 + 0.7071F_2 - 0.7071F_3 = -1200$$

$$+ \uparrow \sum F_y = 0$$

$$-600 + F_3 \operatorname{sen} 45^\circ + F_2 \operatorname{sen} 45^\circ + F_1 \operatorname{sen} 30^\circ = 0$$

$$0.5F_1 + 0.7071F_2 + 0.7071F_3 = 600$$

Nudo 2.



$$+ \rightarrow \sum F_x = 0$$

$$H_2 - F_4 - F_1 \operatorname{sen} 30^\circ = 0$$

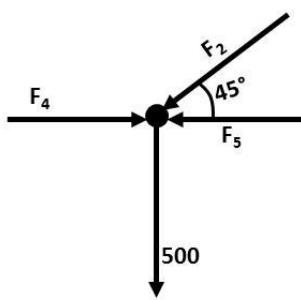
$$0.5F_1 + F_4 - H_2 = 0$$

$$+ \uparrow \sum F_y = 0$$

$$V_2 - F_1 \cos 30^\circ = 0$$

$$0.8660F_1 - V_2 = 0$$

Nudo 3.



$$+ \rightarrow \sum F_x = 0$$

$$F_4 - F_5 - F_2 \sin 45^\circ = 0$$

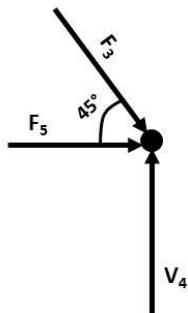
$$0.7071F_2 - F_4 + F_5 = 0$$

$$+ \uparrow \sum F_y = 0$$

$$-500 - F_2 \cos 45^\circ = 0$$

$$0.7071F_2 = -500$$

Nudo 4.



$$+ \rightarrow \sum F_x = 0$$

$$F_5 + F_3 \sin 45^\circ = 0$$

$$0.7071F_3 + F_5 = 0$$

$$+ \uparrow \sum F_y = 0$$

$$V_4 - F_3 \cos 45^\circ = 0$$

$$0.7071F_3 - V_4 = 0$$

Usando Gauss-Jordan en código de Octave,

Método de Gauss-Jordan.

```

8.6600e-001 7.0710e-001 -7.0710e-001 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 -1.2000e+003
5.0000e-001 7.0710e-001 7.0710e-001 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 6.0000e+002
5.0000e-001 0.0000e+000 0.0000e+000 1.0000e+000 0.0000e+000 -1.0000e+000
0.0000e+000 0.0000e+000 0.0000e+000

```

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

```

8.6600e-001 0.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000 -
1.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 7.0710e-001 0.0000e+000 -1.0000e+000 1.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 7.0710e-001 0.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 -5.0000e+002
0.0000e+000 0.0000e+000 7.0710e-001 0.0000e+000 1.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 7.0710e-001 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 -1.0000e+000 0.0000e+000

```

Sustitución hacia adelante:

```

1.0000e+000 8.1651e-001 -8.1651e-001 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 -1.3857e+003
1.0000e+000 1.4142e+000 1.4142e+000 0.0000e+000 0.0000e+000 0.0000e+000
0.0000e+000 0.0000e+000 1.2000e+003
1.0000e+000 0.0000e+000 0.0000e+000 2.0000e+000 0.0000e+000 -2.0000e+000
0.0000e+000 0.0000e+000 0.0000e+000
1.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000 0.0000e+000 -
1.1547e+000 0.0000e+000 0.0000e+000
NaN Inf NaN -Inf Inf NaN
NaN NaN NaN NaN NaN NaN
NaN NaN -Inf NaN NaN NaN
NaN NaN NaN Inf NaN NaN
NaN NaN NaN Inf NaN NaN
NaN -Inf NaN NaN NaN NaN

```

>>

Entonces, se usa matriz inversa,

A =

```

0.86600 0.70710 -0.70710 0.00000 0.00000 0.00000 0.00000 0.00000
0.50000 0.70710 0.70710 0.00000 0.00000 0.00000 0.00000 0.00000
0.50000 0.00000 0.00000 1.00000 0.00000 -1.00000 0.00000 0.00000
0.86600 0.00000 0.00000 0.00000 0.00000 0.00000 -1.00000 0.00000
0.00000 0.70710 0.00000 -1.00000 1.00000 0.00000 0.00000 0.00000
0.00000 0.70710 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.70710 0.00000 1.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.70710 0.00000 0.00000 0.00000 0.00000 -1.00000

```

b =

```

-1200
600

```

```
0
0
0
-500
0
0

A_inversa =
```

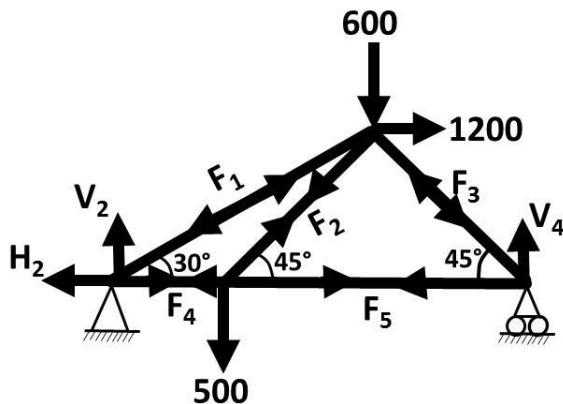
0.73206	0.73206	0.00000	0.00000	0.00000	-1.46413	-0.00000	0.00000
0.00000	-0.00000	0.00000	0.00000	0.00000	1.41423	0.00000	0.00000
-0.51765	0.89657	0.00000	0.00000	0.00000	-0.37892	-0.00000	0.00000
0.36603	-0.63397	0.00000	-0.00000	-1.00000	1.26794	1.00000	0.00000
0.36603	-0.63397	0.00000	-0.00000	0.00000	0.26794	1.00000	0.00000
0.73206	-0.26794	-1.00000	-0.00000	-1.00000	0.53587	1.00000	0.00000
0.63397	0.63397	0.00000	-1.00000	0.00000	-1.26794	-0.00000	0.00000
-0.36603	0.63397	0.00000	0.00000	0.00000	-0.26794	-0.00000	-1.00000

```
x =
```

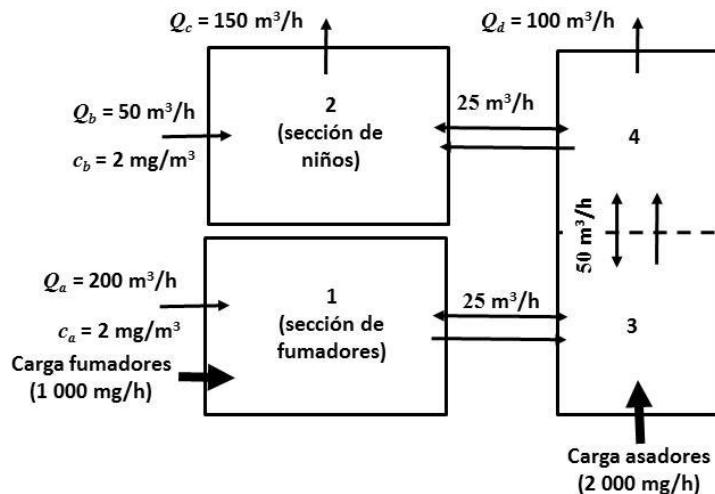
292.83
-707.11
1348.59
-1453.59
-953.59
-1307.17
253.59
953.59

Finalmente,

$F_1 = 292.83$, $F_2 = -707.11$, $F_3 = 1348.59$, $F_4 = -1453.59$, $F_5 = -953.59$, $H_2 = -1307.17$, $V_2 = 253.59$ y
 $V_4 = 953.59$.



B.18². Como su nombre lo indica, la contaminación atmosférica interior trata de la contaminación del aire en espacios cerrados como hogares, oficinas, áreas de trabajo, etc. Suponga que está diseñando un sistema de ventilación para un restaurante como se muestra en la figura. El área de servicio del restaurante consta de dos habitaciones cuadradas y una habitación alargada. La habitación 1 y la habitación 3 tienen fuentes de monóxido de carbono de los fumadores y una rejilla defectuosa, respectivamente. Los balances de masa de estado estacionario se pueden escribir para cada habitación.



- (a) Resolver para estado estable la concentración de monóxido de carbono en cada cuarto.
- (b) Determinar qué porcentaje de monóxido de carbono en la sección de niños es debido a (i) los fumadores, (ii) el asador y (iii) el aire de entrada.
- (c) Si las carga de los fumadores y asadores se incrementa a 2 000 y 5 000 mg/h, respectivamente, usar la matriz inversa para determinar el incremento en la concentración de la sección de niños.

- (d) ¿Cómo cambia la concentración en el área de los niños si se construye una pantalla para que la mezcla entre las áreas 2 y 4 se disminuya a $5 \text{ m}^3/\text{h}$?

Solución.

(a)

Sección de fumadores (1),

$$0 = W_{fumadores} + Q_a C_a - Q_a c_1 + E_{13}(c_3 - c_1)$$

(carga) + (entrada) - (salida) + (mezclado)

sustituyendo parámetros

$$0 = 1000 + 200(2) - 200c_1 + 25c_3 - 25c_1$$

$$225c_1 - 25c_3 = 1400$$

Para la sección de niños (2),

$$0 = Q_b c_b - Q_c c_2 + Q_{42} c_4 + E_{24}(c_4 - c_2)$$

$$50(2) - 150c_2 + 100c_4 + 25c_4 - 25c_2 = 0$$

$$-175c_2 + 125c_4 = -100$$

Para la sección 3 (asadores),

$$0 = 2000 + E_{31}(c_1 - c_3) + Q_a c_1 + E_{34}(c_4 - c_3) - Q_{34} c_3$$

$$25c_1 - 25c_3 + 200c_1 + 50c_4 - 50c_3 - 100c_3 = -2000$$

$$225c_1 - 175c_3 + 50c_4 = -2000$$

Para la sección 4 (asadores),

$$0 = Q_{34} c_3 + E_{43}(c_3 - c_4) + E_{42}(c_2 - c_4) - Q_d c_4$$

$$100c_3 + 50c_3 - 50c_4 + 25c_2 - 25c_4 - 100c_4 = 0$$

$$25c_2 + 150c_3 - 175c_4 = 0$$

Aplicando matriz inversa,

Detected XLS interfaces:

Matriz Inversa.

A =

```
225      0     -25      0
  0    -175      0    125
225      0    -175     50
  0     25    150   -175
```

b =

```
1400
-100
-2000
  0
```

A_inversa =

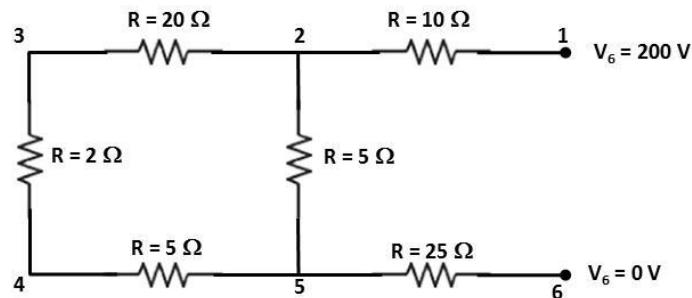
```
5.5309e-003 -4.9383e-005 -1.0864e-003 -3.4568e-004
6.6667e-003 -6.6667e-003 -6.6667e-003 -6.6667e-003
9.7778e-003 -4.4444e-004 -9.7778e-003 -3.1111e-003
9.3333e-003 -1.3333e-003 -9.3333e-003 -9.3333e-003
```

x =

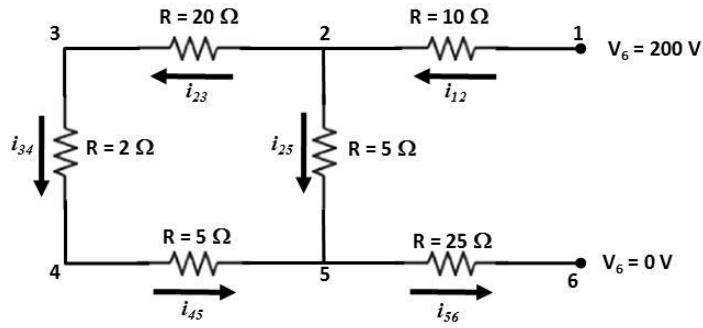
```
9.9210 mg/m³
23.3333 mg/m³
33.2889 mg/m³
31.8667 mg/m³
```

>>

B16². Realizar los mismos cálculos de la sección 12.3, pero con el circuito mostrado en la figura.



Solución.



$$i_{12} - i_{23} - i_{25} = 0$$

$$i_{23} - i_{34} = 0$$

$$i_{34} - i_{45} = 0$$

$$i_{45} + i_{25} - i_{56} = 0$$

$$i_{23} R_{23} + i_{34} R_{34} + i_{45} R_{45} - i_{25} R_{25} = 0$$

$$20i_{23} + 2i_{34} + 5i_{45} - 5i_{25} = 0$$

$$i_{12} R_{12} + i_{25} R_{25} + i_{56} R_{56} = 200 - 0 = 0$$

$$10i_{12} + 5i_{25} + 25i_{56} = 200$$

Aplicando matriz inversa,

Detected XLS interfaces:

Matriz Inversa.

A =

$$\begin{matrix} 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{matrix}$$

```
0      1      0      0      1      -1
0     -5     20      2      5      0
10      5      0      0      0     25
```

b =

```
0
0
0
0
0
200
```

A_inversa =

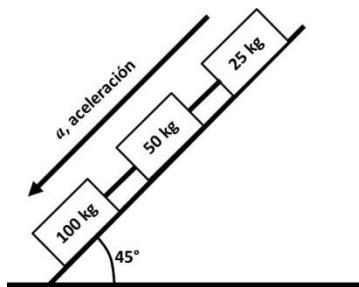
```
0.7450199  0.6653386  0.6573705  0.6374502  0.0039841  0.0254980
-0.2151394  0.3426295  0.3984064  0.5378486  -0.0278884  0.0215139
-0.0398406  0.3227092  0.2589641  0.0996016  0.0318725  0.0039841
-0.0398406  -0.6772908  0.2589641  0.0996016  0.0318725  0.0039841
-0.0398406  -0.6772908  -0.7410359  0.0996016  0.0318725  0.0039841
-0.2549801  -0.3346614  -0.3426295  -0.3625498  0.0039841  0.0254980
```

x =

```
5.09960
4.30279
0.79681
0.79681
0.79681
5.09960
```

>>

B17². Tres bloques son conectados por una cuerda sin peso y el resto en un plano inclinado (ver figura).



Se tienen las siguientes ecuaciones simultáneas (diagramas de cuerpo libre)

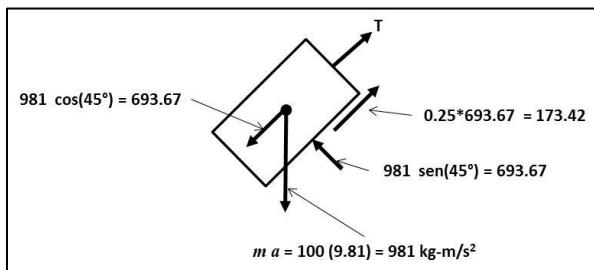
$$100a + T = 519.72$$

$$50a - T + R = 216.55$$

$$25a - R = 108.28$$

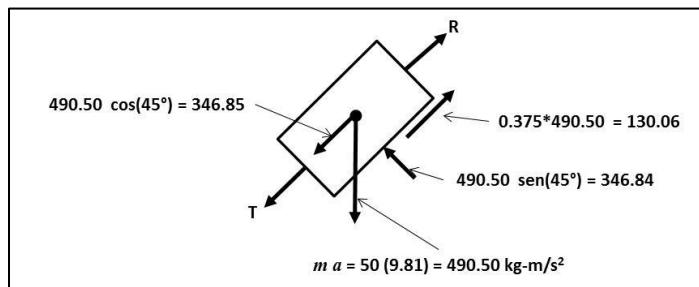
Solucionar para la aceleración a y las tensiones T y R en las dos cuerdas.

Solución.



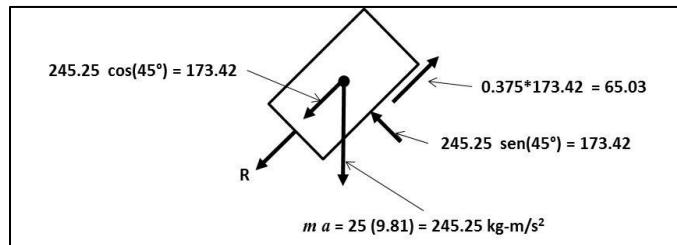
$$100a + T + 173.42 - 693.67 = 0$$

$$100a + T = 520.25$$



$$50a - T + R + 130.06 - 346.85 = 0$$

$$100a - T + R = 216.78$$



$$25a - R + 65.03 - 173.42 + 65.03 = 0$$

$$25a - R = 108.39$$

Aplicando matriz inversa con código Octave,

Detected XLS interfaces:

Matriz Inversa.

A =

$$\begin{matrix} 100 & 1 & 0 \\ 50 & -1 & 1 \\ 25 & 0 & -1 \end{matrix}$$

b =

$$\begin{matrix} 520.25 \\ 216.78 \\ 108.39 \end{matrix}$$

A_inversa =

$$\begin{matrix} 0.0057143 & 0.0057143 & 0.0057143 \\ 0.4285714 & -0.5714286 & -0.5714286 \\ 0.1428571 & 0.1428571 & -0.8571429 \end{matrix}$$

x =

$$\begin{matrix} 4.8310 & a \\ 37.1529 & T \\ 12.3843 & R \end{matrix}$$

>>

Sistemas de ecuaciones no lineales.

B.18². Resolver la siguiente ecuación obteniendo sus máximos, iniciando con $x = -1$ y $y = 1$.

$$f(x, y) = 2xy + 2x - x^2 - 2y^2$$

Solución.

Derivando parcialmente se obtiene,

$$\frac{\partial f}{\partial x} = 2y + 2 - 2x = 0$$

$$\frac{\partial f}{\partial y} = 2x - 4y = 0$$

Resolviendo con código en Octave,

```
Introduce una función con respecto a x1 2*x2+2-2*x1
```

```
Introduce una función con respecto a x2 2*x1-4*x2
```

```
Método de Newton-Raphson para ecuaciones no lineales.
```

```
Valores iniciales
```

```
x1 = -1
```

```
x2 = 1
```

```
Tolerancia = 0.0001
```

```
Iteraciones = 30
```

```
Jacobiano:
```

```
-2 2  
2 -4
```

```
Determinante del Jacobiano
```

```
4
```

```
Inversa del Jacobiano
```

```
-1.00000 -0.50000  
-0.50000 -0.50000
```

k	x1,k	x2,k	-f1,k	-f2,k	f11,k	f12,k	f21,k
f22,k	h						
1,k	h2,k						
0	-1.000000	1.000000	-6.000000	6.000000	-2.000000	2.000000	
2.000000	-4.000000	3.0					
00000	-0.000000						
1	2.000000	1.000000	-0.000000	-0.000000	-2.000000	2.000000	
2.000000	-4.000000	0.0					
00000	0.000000						
2	2.000000	1.000000					

```
Raíces: 2.000000 1.000000
```

```
>>
```

O también usando comandos de Octave,

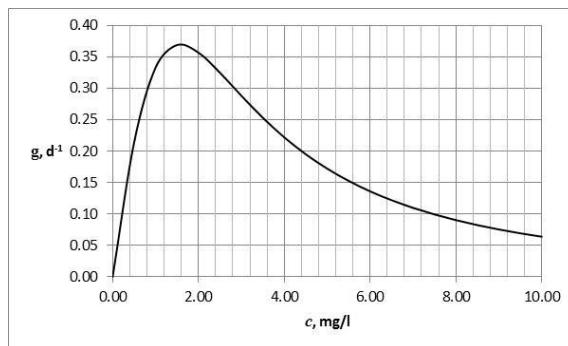
```
>> function [f]=ecuaciones(x)
f(1)=2*x(2)+2-2*x(1);
f(2)=2*x(1)-4*x(2)
endfunction
>> x0=[-1,1];
>> [x,feval]=fsolve('ecuaciones', x0)
f =
6   -6
f =
6.0000  -6.0000
f =
6.0000  -6.0000
f =
6.0000  -6.0000
f =
6.0000  -6.0000
f =
2.07768    0.27572
f =
2.07767    0.27573
f =
2.07769    0.27570
f =
2.07769    0.27569
f =
2.07767    0.27574
f =
0.32189    0.22069
f =
2.1694e-011  0.0000e+000
```

```
x =
2.00000  1.00000
feval =
2.1694e-011  0.0000e+000
>>
```

B.19². La tasa de crecimiento específica de una levadura que produce un antibiótico es una función de la concentración de alimentos c ,

$$g = \frac{2c}{4 + 0.8c + c^2 + 0.2c^3}$$

Como se muestra en la figura, el crecimiento va de cero a concentraciones muy bajas debido a la limitación de alimentos. También va de cero a altas concentraciones debido a efectos de toxicidad. Encontrar el valor de c en el que el crecimiento es un máximo.



Solución.

Debido a que el máximo de una función se puede determinar igualando a cero su primera derivada, utilizando Octave se puede determinar lo siguiente,

```
>> syms g(c)
>> g=2*c/(4+0.8*c+c^2+0.2*c^3);
>> diff(g)
ans = (sym)

      /      2          \
      |  3*c          4 |
2*c*| - ----- - 2*c - - | +
      \      5          5/          2
----- + -----
      / 3           2          3
                  \      c      2    4*c
```

$$\frac{|c^2 - 4*c|}{|5 - c - \frac{4}{5}|} = \frac{c^2 - 4*c}{5 - c - \frac{4}{5}}$$

```
>> simplify(ans)
ans = (sym)

      3      2
    - 20*c  - 50*c  + 200
-----
      6      5      4      3      2
c  + 10*c  + 33*c  + 80*c  + 216*c  + 160*c + 400

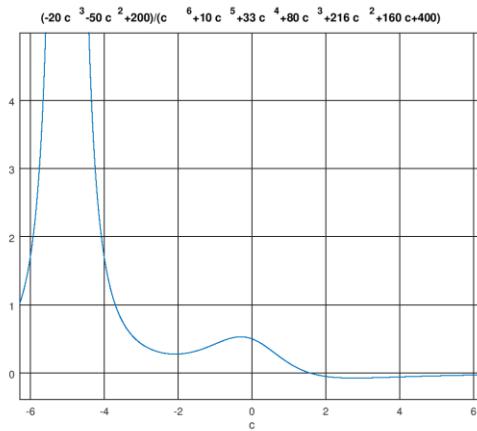
>>
```

Entonces, la derivada es:

$$\frac{dg}{dc} = \frac{-20c^3 - 50c^2 + 200}{c^6 + 10c^5 + 33c^4 + 80c^3 + 216c^2 + 160c + 400}$$

Si se desea graficar con Octave,

```
>> ezplot('2*c/(4+0.8*c+c.^2+0.2*c.^3)', [0, 10])
>> ezplot('(-20*c.^3-50*c.^2+200)/(c.^6+10*c.^5+33*c.^4+80*c.^3+216*c.^2+160*c+400)')
>>
```

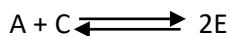
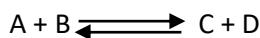


Resolviendo con Octave,

```
>> fc=@(c) (-20*c.^3-
50*c.^2+200)/(c.^6+10*c.^5+33*c.^4+80*c.^3+216*c.^2+160*c+400);
```

```
>> sol=fzero(fc,1)
sol = 1.5679
>>
```

B.20⁶. En un reactor se efectúan las siguientes reacciones en fase gaseosa:



A la temperatura de la reacción, las constantes de equilibrio son $k_{p1} = 2.6$ y $k_{p2} = 3.1$. Las composiciones iniciales son 2 mol/l de A y 1 mol/l de B.

Calcular la composición a la salida del reactor, asumiendo que se alcanza el equilibrio.

Solución.

Si x_1 = moles de A convertidos en la reacción (1) y x_2 = moles de A convertidos en la reacción (2), entonces en el equilibrio se tiene:

Aplicando la ley de acción de masas,

Para la reacción (1)

$$2.6 = \frac{(x_1 - x_2)x_1}{(2 - x_1 - x_2)(1 - x_1)}$$

Para la reacción (2)

$$3.1 = \frac{(2x_2)^2}{(2 - x_1 - x_2)(x_1 - x_2)}$$

Debido a los valores iniciales de moles en A y en B, se suponen valores iniciales de $x_1 = 0.8$ y $x_2 = 0.4$.

Ejecutando con código en Octave,

```
Introduce una función con respecto a x1 (x1-x2)*x1/((2-x1-x2)*(1-x1))-2.6
```

```
Introduce una función con respecto a x2 (2*x2)^2/((2-x1-x2)*(x1-x2))-3.1
```

Método de Newton-Raphson para ecuaciones no lineales.

⁶ Nieves, A. y Domínguez, F. MÉTODOS NUMÉRICOS APLICADOS A LA INGENIERÍA. 2da ed. CECSA. México, 2005.

Valores iniciales

x1 = 0.8

x2 = 0.4

Tolerancia = 0.000001

Iteraciones = 50

Jacobiano:

20.0000	-2.5000
-2.5000	17.5000

Determinante del Jacobiano

343.75

Inversa del Jacobiano

0.0509091	0.0072727
0.0072727	0.0581818

k	x1,k	x2,k	-f1,k	-f2,k	f11,k	f12,k	f21,k	f22,k	h1,k	h2,k
0	0.800000	0.400000	0.600000	1.100000	20.000000	-2.500000	-2.500000	17.500000	0.038545	0.068364
1	0.838545	0.468364	-0.173967	-0.319960	31.985005	-3.491212	-4.304236	28.776823	-0.006763	-0.012130
2	0.831782	0.456233	-0.008162	-0.013846	29.248489	-3.281701	-3.917974	26.315160	-0.000344	-0.000577
3	0.831439	0.455656	-0.000020	-0.000029	29.117916	-3.271872	-3.901086	26.204855	-0.000001	-0.000001
4	0.831438	0.455655	-0.000000	-0.000000	29.117607	-3.271850	-3.901052	26.204625	-0.000000	-0.000000
5	0.831438	0.455655								

Raíces: 0.831438 0.455655

>>

Entonces, $x_1 = 0.831438$ y $x_2 = 0.455655$.

B21⁶. La presión p requerida para sumergir una lámina circular de radio r en un terreno suave, donde la base dura del terreno se encuentra a una distancia $D > d$ debajo de la superficie, puede aproximarse mediante una ecuación de la forma:

$$p = k_1 e^{(k_2 r)} + k_3 r$$

donde k_1 , k_2 y k_3 son constantes que, con $k_2 > 0$, dependen de d y de la consistencia del terreno pero no del radio de la lámina.

Encontrar los valores de k_1 , k_2 y k_3 si se supone que una lámina de radio de 1 pul requiere una presión de 10 lb/pul² para sumergirse 1 pie en el terreno lodoso; una lámina de 2 pul, requiere de una presión de 12 lb/pul² para sumergirse 1 pie; y una lámina de 3 pul requiere una presión de 15 lb/pul² (suponiendo que el lodo tiene un espesor de más de 1 pie).

Solución.

Sustituyendo los valores en la ecuación para los tres casos, se obtiene:

$$10 = k_1 e^{(k_2)} + k_3$$

$$12 = k_1 e^{(2k_2)} + 2k_3$$

$$15 = k_1 e^{(3k_2)} + 3k_3$$

Resolviendo con comandos de Octave,

```
>> pkg load symbolic
>> syms k
>> f=@(k) [k(1)*exp(k(2))+k(3)-10; k(1)*exp(2*k(2))+2*k(3)-12;
k(1)*exp(3*k(2))+3*k(3)-15];
>> fsolve(f, [1,1,1])
ans =
8.77129  0.25970 -1.37228
>>
```

Es decir, $k_1 = 8.771286$, $k_2 = 0.259695$ y $k_3 = -1.372281$.

Interpolación polinomial y Aproximación funcional.

9.22². Se realiza un experimento y se determinan los siguientes valores de capacidad de calor c para varias temperaturas T de un gas:

T	-50	-30	0	60	90	110
c	1270	1280	1350	1480	1580	1700

Usar regresión para determinar un modelo para predecir c como función de T .

Solución.

Usando comandos de Octave y ajustando un polinomio de segundo grado,

```
>> T=[-50 -30 0 60 90 100];
>> c=[1270 1280 1350 1480 1580 1700];
>> polyfit(T,c,2)
ans =
0.012770      1.816372     1330.951759
>>
```

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Grado del polinomio? 2

Coeficientes del polinomio ajustado

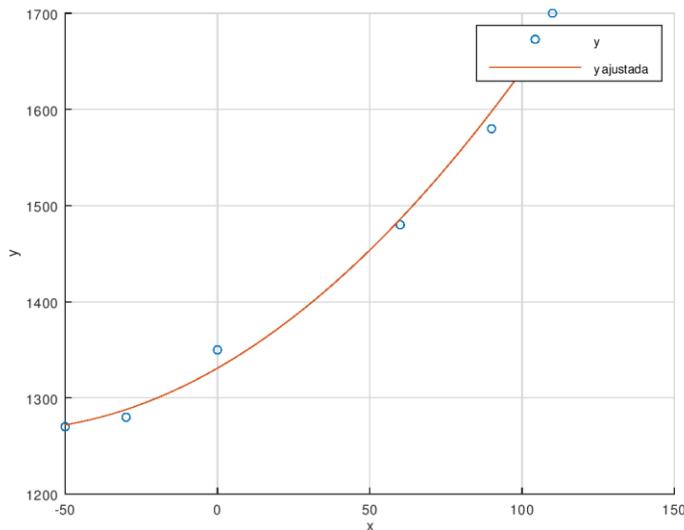
```
1330.951759
 1.816372
 0.012770
```

Coef. de determinación: 0.9934

Introduce el valor de x a estimar.
cuando termines de estimar oprime Ctrl C.

```
x =
>>
```

Entonces, $c = 1330.951759 + 1.816372T + 0.012770T^2$



9.23². Un reactor está termalmente estratificado como se muestra en la siguiente tabla:

Profundidad, m	0	0.5	1.0	1.5	2.0	2.5	3.0
Temperatura, °C	70	68	55	22	13	11	10

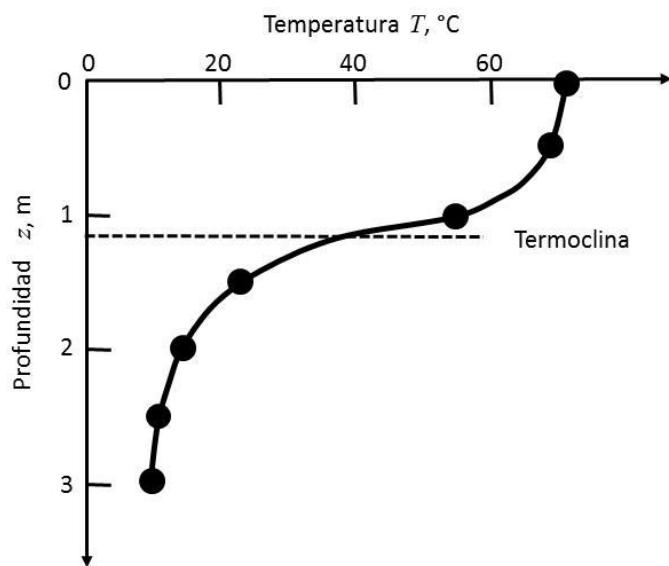
Como se muestra en la figura, el tanque puede ser idealizado como dos zonas separadas por un gradiente de temperatura fuerte o termoclina. La profundidad de este gradiente puede ser

definido como el punto de inflexión de la curva temperatura-profundidad, esto es, el punto en el cual $d^2T/dz^2 = 0$.

En esta profundidad, el flujo de calor desde la superficie hacia la capa del fondo puede ser calculada con la ley de Fourier,

$$J = -k \frac{dT}{dz}$$

Ajustar una curva cúbica en estos datos para determinar la profundidad termoclina. Si $k = 0.02$ cal/(s cm °C) calcular el flujo a través de la interfase.



Solución.

Usando comandos de Octave,

```
>> z=[0 0.5 1.0 1.5 2.0 2.5 3.0];
>> T=[70 68 55 22 13 11 10];
>> polyfit(P,T,3)
ans =
    8.6667   -33.8571      3.9048     71.5000
>>
```

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

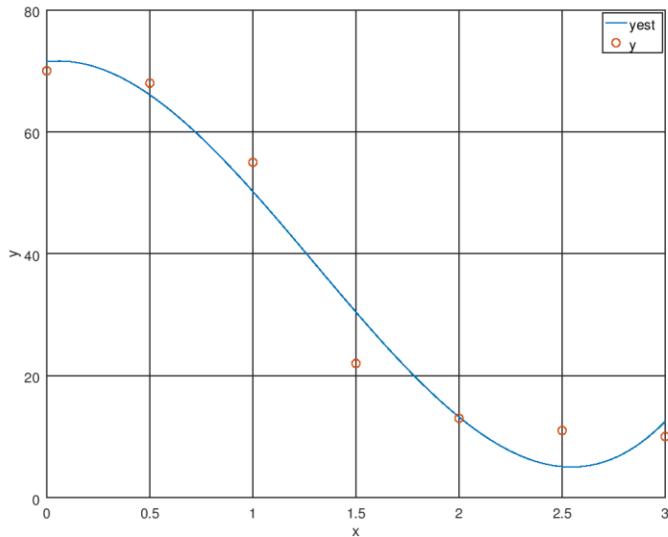
Detected XLS interfaces:

Grado del polinomio = 3

p =

8.666666666666667 -33.85714285714285 3.90476190476187 71.500000000000001

>>



Entonces, $T = 71.50000 + 3.9048z - 33.8571z^2 + 8.6667z^3$.

También, $dT/dz = 3.9048 - 67.7142z + 26.0000z^2$.

Y, por lo tanto, $J = -0.02(3.9048 - 67.7142z + 26.0000z^2)$

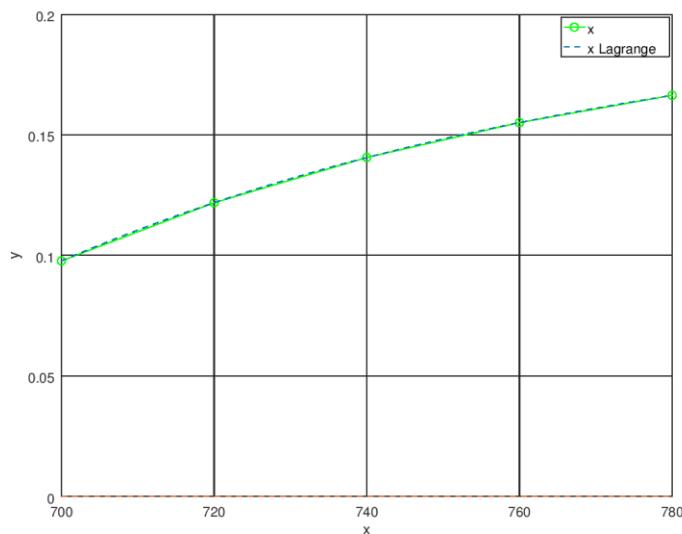
9.24². El volumen específico de un vapor sobrecalentado se enumera en las tablas del vapor para las varias temperaturas. Por ejemplo, a una presión de 3 000 lb/in², absoluta:

$T, ^\circ F$	700	720	740	760	780
$v, \text{pie}^3/\text{lb}$	0.0977	0.12184	0.14060	0.15509	0.16643

Determinar v para $T = 750$ °F.

Solución.

Usando código en Octave,



Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
5 Aes

```
0.000000
-0.000000
0.000000
-0.000000
0.000000
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 750
0.148309
```

```
x1 =
>>
```

9.25². En la enfermedad de Alzheimer, el número de neuronas en la corteza disminuye a medida que la enfermedad progresó. Se tomaron los siguientes datos para determinar el número de receptores de neurotransmisores que quedaron en un cerebro enfermo. El neurotransmisor libre ($[F]$) fue incubado con el tejido y la concentración que limitó específicamente a un receptor ($[B]$) fue medido. Entonces el atascamiento es específico a un receptor, el límite de la concentración está relacionado con la concentración libre por la relación siguiente:

$$[B] = \frac{B_{\max} [F]}{K + [F]}$$

Usando los datos de abajo, determinar los parámetros que minimizan la suma de los cuadrados de los residuos. También calcular r^2 .

$[F]$, nM	0.1	0.5	1	5	10	20	50
$[B]$, nM	10.57	36.61	52.93	82.65	89.46	94.35	101.00

Solución.

$$[B] = \frac{B_{\max} [F]}{K + [F]}, \quad \frac{1}{[B]} = \frac{1}{B_{\max}} + \frac{K}{B_{\max}} \frac{1}{[F]}$$

$$y = a + mx$$

$1/[F]$	10.00	2.00	1.00	0.20	0.10	0.05	0.02
$1/[B]$	0.0946	0.0273	0.0189	0.0121	0.0112	0.0106	0.0099

Resolviendo con código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

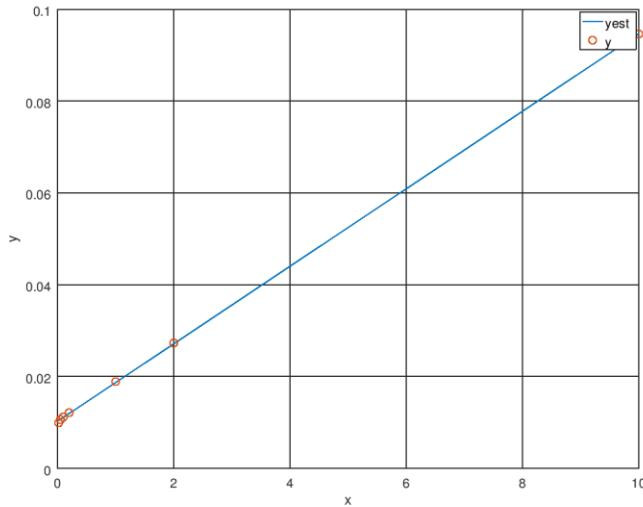
Detected XLS interfaces:

Grado del polinomio = 1

p =

0.00844218739522837 0.01024576721858996

```
Correlación Pearson: 1.0000
Correlación Kendall: 1.0000
Correlación Spearman: 1.0000
>>
```



$$1/B_{máx} = 0.0102, B_{máx} = 97.6013 \text{ nM.}$$

$$K/B_{máx} = 0.0084, K = 0.8240.$$

$$\text{Finalmente, } [B] = \frac{B_{máx} [F]}{K + [F]} = \frac{97.6013 [F]}{0.8240 + [F]}$$

9.26². Los esfuerzos de corte, en kilopascales (kPa), de nueve muestras tomadas en varias profundidades en un estrato de arcilla se enumeran a continuación. Estimar el esfuerzo de cizallamiento a una profundidad de 4.5 m.

Profundidad, m	1.9	3.1	4.2	5.1	5.8	6.9	8.1	9.3	10.0
Esfuerzo, kPa	14.4	28.7	19.2	43.1	33.5	52.7	71.8	62.2	76.6

Solución.

Crear previamente los vectores X y Y en excel.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

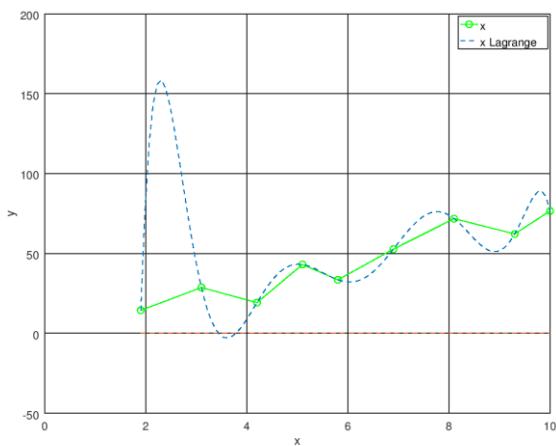
Valores de las
9 Aes

```
0.000225
-0.004954
0.016920
-0.096187
0.076377
-0.058113
0.031458
-0.008970
0.002785
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 4.5
33.289583 KPa
```

```
x1 =
>>
```



Usando ajuste polinomial,

Crear previamente los vectores x y y en excel.

si ya lo creaste oprime Enter si no, oprime Ctrl C y Enter.

Detected XLS interfaces:

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Grado del polinomio? 8

Coeficientes del polinomio ajustado

-31825.039504

54853.837038

-39003.353846

15066.802774

-3479.241526

494.222840

-42.345240

2.007822

-0.040460

Coef. de determinación Pearson: 1.0000

Coef. de correlación Kendall: 1.0000

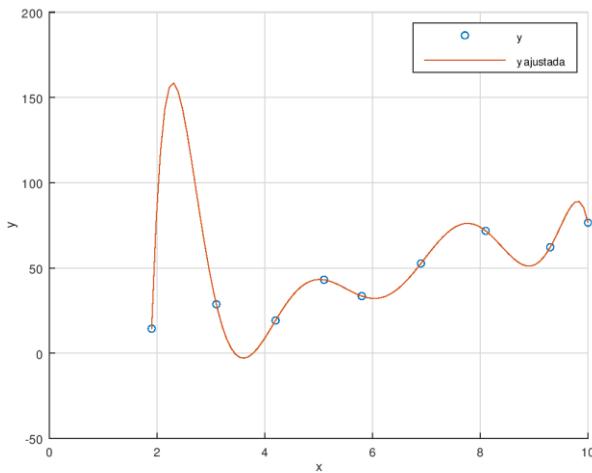
Coef. de correlación Spearman: 1.0000

Introduce el valor de x a estimar.

cuando termines de estimar oprime Ctrl C.

x =

>>



O también se puede ajustar una recta de una manera más gruesa a la serie de datos usando código en Octave y luego estimar para los valores requeridos.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 1
p =
```

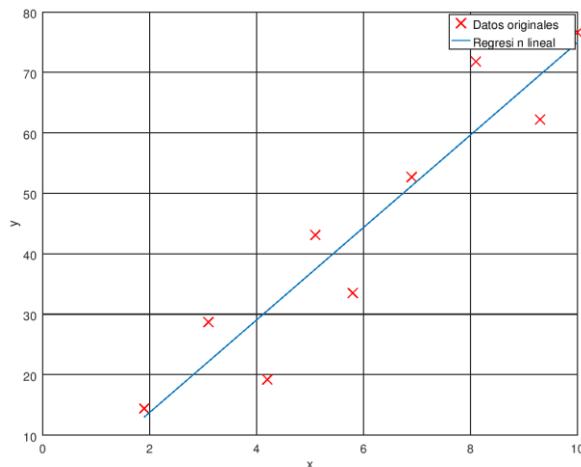
```
7.64914407730448 -1.54593753392928
```

```
Correlación Pearson: 0.9390
Correlación Kendall: 0.8333
Correlación Spearman: 0.9500
```

>>

Es decir,

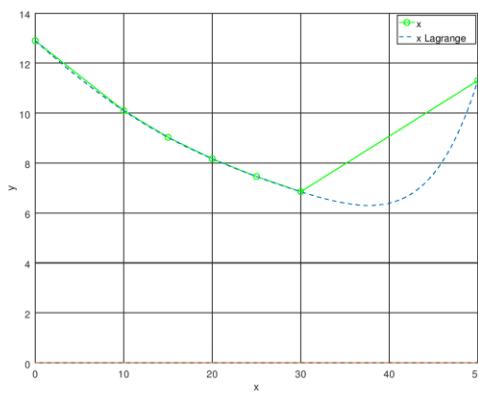
$$\text{Esfuerzo} = -1.5459 + 7.6491 * \text{Profundidad} = -1.5459 + 7.6491 * 4.5 = 32.88 \text{ kPA}$$



9.27². La concentración de saturación de oxígeno disuelto en el agua en función de la concentración de la temperatura y del cloruro se enumera en la tabla. Utilizar interpolación para estimar el nivel de oxígeno disuelto para $T = 8^{\circ}\text{C}$ con cloruro = 10 g/l.

$T, ^{\circ}\text{C}$	$c = 10 \text{ g/l}$
0	12.9
50	11.3
10	10.1
15	9.03
20	8.17
25	7.46
30	6.85

Solución.



Crear previamente los vectores X y Y en excel.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
7 Aes

0.000000
-0.000002
0.000005
-0.000005
0.000003
-0.000001
0.000000

Cuando termines de interpolar oprime Ctrl c

**x1 = 8
10.592914**

x1 =

9.28². En la ingeniería de recursos hídricos, el dimensionamiento de los embalses depende de las estimaciones exactas del escurrimiento de agua en el río que se está captando. Para algunos ríos, los registros históricos a largo plazo de tales datos de escurrimiento son difíciles de obtener. En cambio, los datos meteorológicos sobre la precipitación están a menudo disponibles por muchos años pasados. Por lo tanto, a menudo es útil determinar una relación entre el escurrimiento y la precipitación. Esta relación se puede utilizar para estimar el escurrimiento durante años cuando sólo se realizaron mediciones de precipitación. Los siguientes datos están disponibles para un río:

Precipitación, cm	88.9	108.5	104.1	139.7	127	94	116.8	99.1
Escurrimiento, m ³ /s	14.6	16.7	15.3	23.2	19.5	16.1	18.1	16.6

Predecir el escurrimiento anual si la precipitación es 120 cm. usando regresión lineal.

Solución.

Usando regresión lineal en código de Octave,

Crear previamente los vectores X y Y en excel.

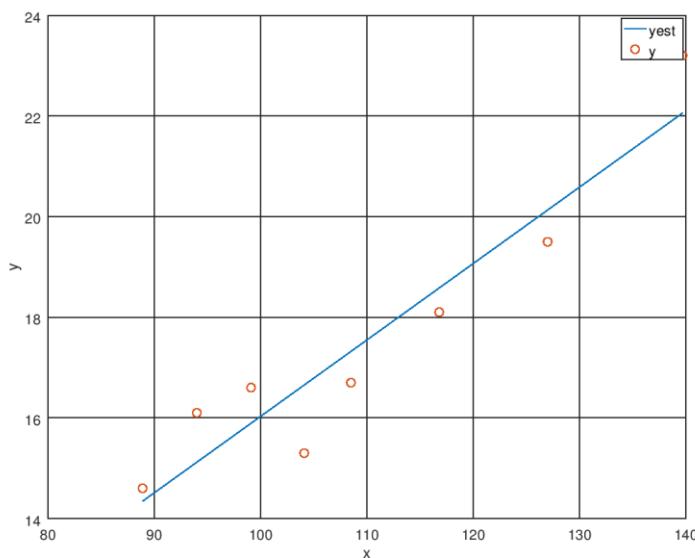
En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Grado del polinomio = 1
 $p =$

0.151870788172619 0.842782613202885

>>



Entonces, se ajusta la siguiente ecuación, con un 95.06% de coef. de regresión.

$$\text{Escurreimiento} = 0.8428 + 0.1519 * \text{precipitación} = 0.8428 + 0.1519 * 120 = 19.0708 \text{ m}^3/\text{s}$$

9.29². Se está trabajando en una comunidad que tiene agua contaminada. En $t = 0$ se adiciona un desinfectante en una cisterna que está contaminada con bacterias. Se tienen las siguientes mediciones.

t (h)	2	4	6	8	10
c (#/100 ml)	430	190	80	35	16

Si el agua está apta para beber cuando la concentración cae por debajo de 5 #/100 ml, estimar el tiempo en el cual la concentración caerá debajo del límite.

Solución.

NOTA: Como se observa, el valor de 5 #/100 ml no es parte de la tabla de mediciones, por lo que para resolver se tendría que extrapolar debido a que el valor mínimo medido es de 16 #/100 ml, lo cual no es parte del presente tema.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
5 Aes

A =

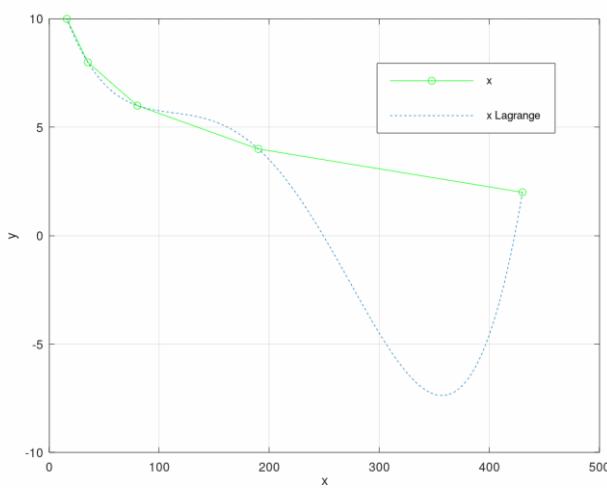
```
0.00000000014560
-0.00000000561791
0.00000005411255
-0.00000015282524
0.00000011416076
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 5
11.628238
```

x1 =

>>



Como se observa en la figura, debido a que Lagrange ajusta considerando que la función pase por todos los puntos, se corre el riesgo de que arroje resultados ilógicos, como en este ejemplo que se pueden obtener resultados negativos. Por ello tal vez es más adecuado convertir los valores medidos de concentración a una escala logarítmica - $\ln(c)$ -.

t (h)	2	4	6	8	10
$\ln C$ (#/100 ml)	6.0638	5.2470	4.3820	3.5553	2.7726

Y el valor del tiempo deseado sería para $\ln(5) = 1.6094$.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
5 Aes

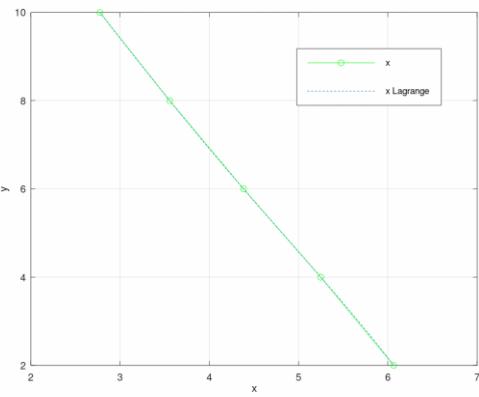
A =

0.17637
-1.35256
3.10000
-2.91343
0.97469

Cuando termines de interpolar oprime Ctrl c

x1 = 1.6094
12.875284

x1 =
>>



Entonces, el tiempo aproximado será de 12.88 h.

9.30². Se mide la caída de voltaje V a través de una resistencia para diferentes valores de corriente i . Los resultados son:

i	0.25	0.75	1.25	1.5	2.0
V	-0.45	-0.6	0.7	1.88	6.0

Estimar la caída de voltaje para $i = 1.15$.

Resultados.

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
5 Aes

```
-0.411429
2.560000
7.466667
-16.042667
7.314286
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 1.15  
0.337286
```

```
x1 =  
>>
```

9.31². La corriente en un cable se midió con gran precisión como una función del tiempo:

<i>t</i>	0	0.1250	0.2500	0.3750	0.5000
<i>i</i>	0	6.24	7.75	4.85	0.0000

Determinar *i* en *t* = 0.23.

Solución.

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

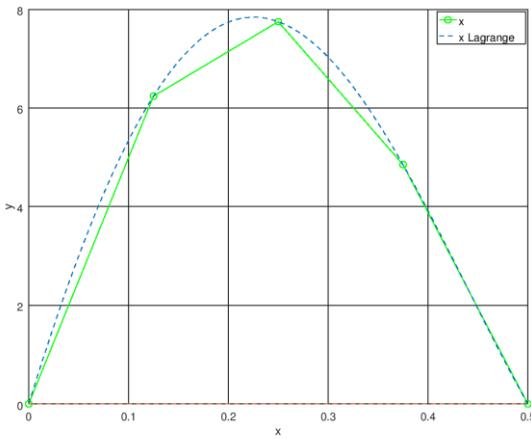
Valores de las
5 Aes

0.000000
-4259.840000
7936.000000
-3310.933333
0.000000

Cuando termines de interpolar oprime Ctrl c

```
x1 = 0.23  
7.838646
```

```
x1 =  
>>
```



9.32². Los siguientes datos fueron tomados de un experimento que midió la corriente en un cable para varios voltajes aplicados:

V, v	2	3	4	5	7	10
i, A	5.2	7.8	10.7	13	19.3	27.5

Determinar la corriente para un voltaje de 3.5 v, aplicando regresión lineal.

Solución.

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 1
p =
```

```
2.808171206225681 -0.592217898832687
>>
```

Luego, $i = -0.59222 + 2.80817 \cdot V = -0.59222 + 2.80817 \cdot 3.5 = 9.2364 \text{ A}$

9.33². La viscosidad dinámica del agua $\mu(10^{-3} \text{ N}\cdot\text{s}/\text{m}^2)$ se relaciona con la temperatura $T(\text{°C})$ de la siguiente manera:

T	0	5	10	20	30	40
μ	1.787	1.519	1.307	1.002	0.7975	0.6529

Estimar μ en $T = 7.5\text{°C}$.

Solución.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

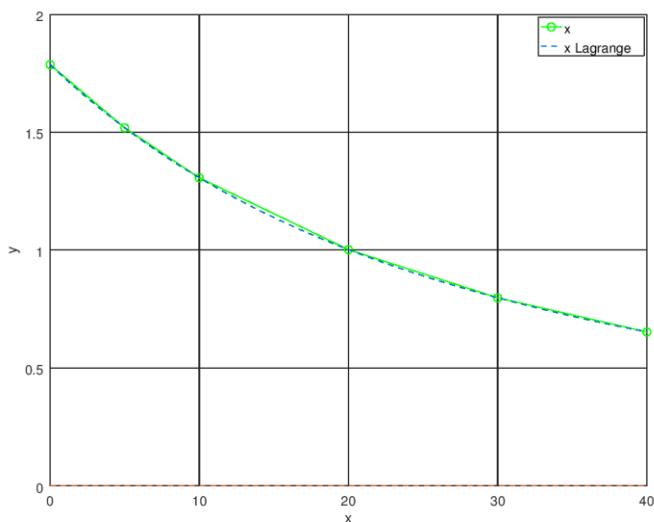
Valores de las
6 Aes

-0.000001
0.000005
-0.000004
0.000002
-0.000001
0.000000

Cuando termines de interpolar oprime Ctrl c

x1 = 7.5
1.406863

x1 =
>>



9.34². Se realiza un experimento para definir la relación entre un esfuerzo aplicado y el tiempo de fractura para un tipo de acero inoxidable. Se aplican 8 diferentes valores de esfuerzo resultando los datos siguientes:

Esfuerzo aplicado x , kg/mm ²	5	10	15	20	25	30	35	40
Tiempo de fractura y , h.	40	30	25	40	18	20	22	15

Estimar el tiempo de fractura para un esfuerzo aplicado de 20 kg/mm².

Solución.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Valores de las
8 Aes

```
-0.000000
0.000001
-0.000001
0.000004
-0.000002
0.000001
-0.000000
0.000000
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 20  
40.000000  
  
x1 =  
>>
```

9.35². La aceleración debida a la gravedad en una altitud y y sobre la superficie de la Tierra está dada por

$y, \text{ m}$	0	30,000	60,000	90,000	120,000
$g, \text{ m/s}^2$	9.8100	9.7487	9.6879	0.6278	9.5682

Calcular g en $y = 55,000 \text{ m}$.

Solución.

Aplicando código en Octave,

Crear previamente los vectores X y Y en excel.

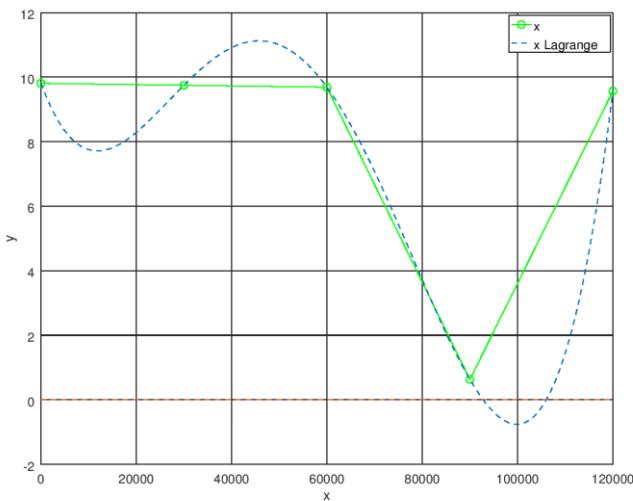
En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Valores de las  
5 Aes  
  
0.000000  
-0.000000  
0.000000  
-0.000000  
0.000000
```

Cuando termines de interpolar oprime Ctrl c

```
x1 = 55000  
10.525531  
  
x1 =  
>>
```



9.36². Desarrollar ecuaciones para ajustar el calor específico ideal c_p (kJ/kg K), como una función de la temperatura T (K), para algunos gases listados en la tabla.

Gas	250 K	300 K	350 K	450 K	550 K	650 K	800 K	900 K	1 000 K
H ₂	14.051	14.307	14.427	14.501	14.530	14.571	14.695	14.822	14.983
CO ₂	0.791	0.846	0.895	0.978	1.046	1.102	1.169	1.204	1.234
O ₂	0.913	0.918	0.928	0.956	0.988	1.017	1.054	1.074	1.090
N ₂	1.039	1.039	1.041	1.049	1.065	1.086	1.121	1.145	1.167

Solución.

Resolviendo con código en Octave,

- Para H₂.

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Grado del polinomio = 3
p =

```
6.67495334384018e-009 -1.27542998171262e-005 8.37538917819111e-003
1.27133442660017e+001
```

Correlación Pearson: 0.9874

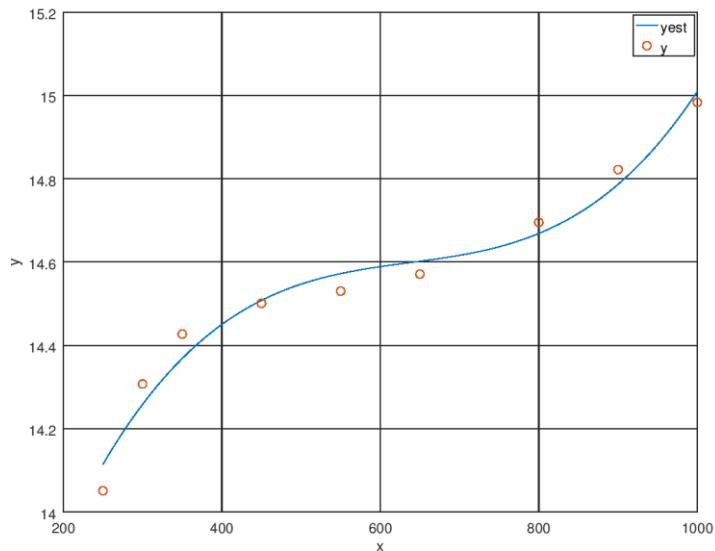
pval: 7.33063e-007

ans =

scalar structure containing the fields:

```
method = Pearson's product moment correlation
params = 7
stat = 16.4996470342773
dist = t
pval = 7.33063244329202e-007
alternative = !=
```

>>



- Para CO₂:

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

Grado del polinomio = 2

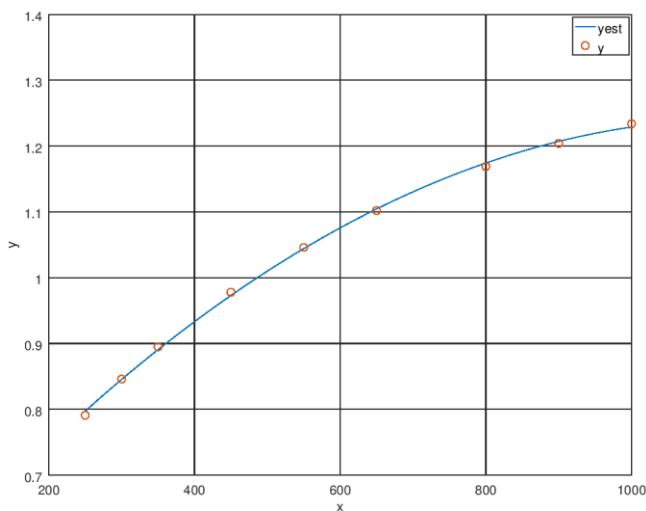
p =

```
-5.49579878082270e-007 1.26239121679756e-003 5.16087401834256e-001
```

```
Correlación Pearson: 0.9996
pval: 3.5516e-012
ans =
scalar structure containing the fields:

method = Pearson's product moment correlation
params = 7
stat = 95.8242292472235
dist = t
pval = 3.55160345577588e-012
alternative = !=

>>
```



- Para O₂:

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 1
p =
```

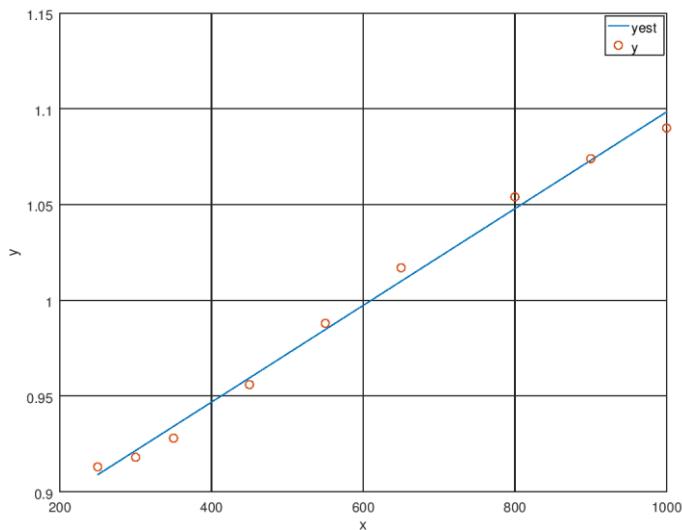
```
2.52655367231639e-004 8.45728813559322e-001
```

```
Correlación Pearson:  0.9967
pval: 6.6431e-009
ans =

scalar structure containing the fields:

method = Pearson's product moment correlation
params = 7
stat = 32.5785469121032
dist = t
pval = 6.64309918185779e-009
alternative = !=

>>
```



- N₂:

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

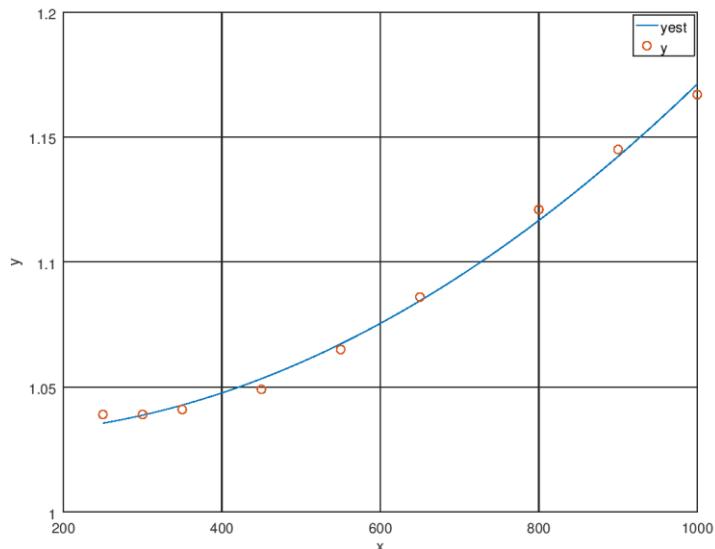
```
Grado del polinomio = 2
p =
1.67170080729309e-007 -2.80787888772932e-005 1.03209165797133e+000
```

```
Correlación Pearson: 0.9978
pval: 1.63243e-009
ans =

scalar structure containing the fields:

method = Pearson's product moment correlation
params = 7
stat = 39.8499804432810
dist = t
pval = 1.63243329964757e-009
alternative = !=

>>
```



Diferenciación numérica.

9.37². Los siguientes datos fueron recopilados cuando un gran tanque para almacenar petróleo fue cargado:

$t, \text{ min}$	0	10	20	30	45	60	75
$V, 10^6 \text{ barriles}$	0.4	0.7	0.77	0.88	1.05	1.17	1.35

Calcular la tasa de flujo Q (esto es, dV/dt) para cada tiempo con orden de h^2 .

Solución.

Inicialmente se realiza una aproximación polinomial de los datos, para posteriormente obtener las diferencias numéricas.

Usando código en Octave,

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 3
p =
```

Columns 1 through 3:

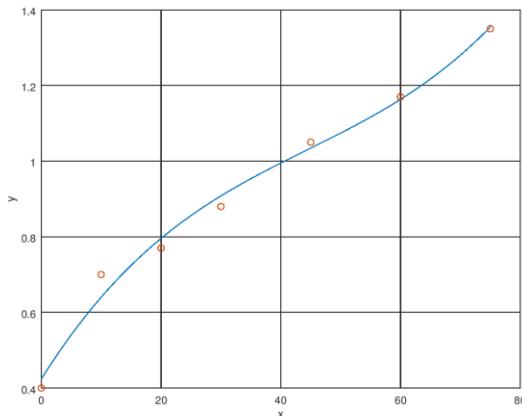
```
2.95116286677543e-006 -3.92695432384404e-004 2.52624229019223e-002
```

Column 4:

```
4.23853732874023e-001
```

Correlación: 0.9951

>>



Entonces, se acepta el siguiente polinomio:

$$V = 2.9512 \times 10^{-6} t^3 - 3.9270 \times 10^{-4} t^2 + 2.5262 \times 10^{-2} t + 4.2385 \times 10^{-1}$$

*** Diferencias Finitas ***.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Introduce una función con respecto a x: $2.9512e-6x^3 - 3.9270e-4x^2 + 2.5262e-2x + 4.2385e-1$

Incremento de x = 0.1
x = 0

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.025262

* Aproximaciones hacia adelante de segundo orden:

0.025262

* Aproximaciones hacia atrás de segundo orden:

0.025262

* Aproximaciones centrada de segundo orden:

0.025262

>>

*** Diferencias Finitas ***.

Introduce una función con respecto a x: $2.9512e-6x^3 - 3.9270e-4x^2 + 2.5262e-2x + 4.2385e-1$

Incremento de x = 0.1
x = 10

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.018293

* Aproximaciones hacia adelante de segundo orden:

0.018293

* Aproximaciones hacia atrás de segundo orden:

0.018293

* Aproximaciones centrada de segundo orden:

```
0.018293  
>>
```

*** Diferencias Finitas ***.

Introduce una función con respecto a x: 2.9512e-6*x^3-3.9270e-4*x^2+2.5262e-
2*x+4.2385e-1

Incremento de x = 0.1
x = 20

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.013095

* Aproximaciones hacia adelante de segundo orden:

0.013095

* Aproximaciones hacia atrás de segundo orden:

0.013095

* Aproximaciones centrada de segundo orden:

0.013095

>>

*** Diferencias Finitas ***.

Introduce una función con respecto a x: 2.9512e-6*x^3-3.9270e-4*x^2+2.5262e-
2*x+4.2385e-1

Incremento de x = 0.1
x = 20

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.013095

* Aproximaciones hacia adelante de segundo orden:

0.013095

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

* Aproximaciones hacia atrás de segundo orden:

0.013095

* Aproximaciones centrada de segundo orden:

0.013095

>>

*** Diferencias Finitas ***.

Introduce una función con respecto a x: 2.9512e-6*x^3-3.9270e-4*x^2+2.5262e-2*x+4.2385e-1

Incremento de x = 0.1
x = 30

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.009668

* Aproximaciones hacia adelante de segundo orden:

0.009668

* Aproximaciones hacia atrás de segundo orden:

0.009668

* Aproximaciones centrada de segundo orden:

0.009668

>>

*** Diferencias Finitas ***.

Introduce una función con respecto a x: 2.9512e-6*x^3-3.9270e-4*x^2+2.5262e-2*x+4.2385e-1

Incremento de x = 0.1
x = 45

¿Hasta qué diferencia quieres calcular (1-4)? 1

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Valor analítico: 0.007848

* Aproximaciones hacia adelante de segundo orden:

0.007847

* Aproximaciones hacia atrás de segundo orden:

0.007847

* Aproximaciones centrada de segundo orden:

0.007848

>>

*** Diferencias Finitas ***.

Introduce una función con respecto a x: 2.9512e-6*x^3-3.9270e-4*x^2+2.5262e-2*x+4.2385e-1

Incremento de x = 0.1
x = 60

¿Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: 0.010011

* Aproximaciones hacia adelante de segundo orden:

0.010011

* Aproximaciones hacia atrás de segundo orden:

0.010011

* Aproximaciones centrada de segundo orden:

0.010011

>>

*** Diferencias Finitas ***.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Introduce una función con respecto a x: $2.9512e-6x^3 - 3.9270e-4x^2 + 2.5262e-2x + 4.2385e-1$

Incremento de x = 0.1
x = 75

¿Hasta qué diferencia quieres calcular (1-4) ? 1

Valor analítico: 0.016158

* Aproximaciones hacia adelante de segundo orden:

0.016158

* Aproximaciones hacia atrás de segundo orden:

0.016158

* Aproximaciones centrada de segundo orden:

0.016159

>>

9.38². Una investigadora ha diseñado un nuevo parche transdérmico para suministrar insulina a través de la piel a los pacientes diabéticos de una manera controlada, eliminando la necesidad de inyecciones dolorosas. Ella ha recogido los siguientes datos sobre el flujo masivo de insulina que se está entregando a través del parche (y la piel) en función del tiempo:

Flujo, mg/cm ² /h	Tiempo, h	Flujo, mg/cm ² /h	Tiempo, h
15	0	8	5
14	1	5	10
12	2	2.5	15
11	3	2	20
9	4	1	24

Recordar que el flujo másico es el caudal a través de un área o $(1/A) dm/dt$. Proporcionar la mejor estimación posible para la cantidad de droga entregada a través de la piel en 24 horas usando un parche de 12 cm².

Solución.

Ajustando polinomialmente,

Crear previamente los vectores X y Y en excel.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

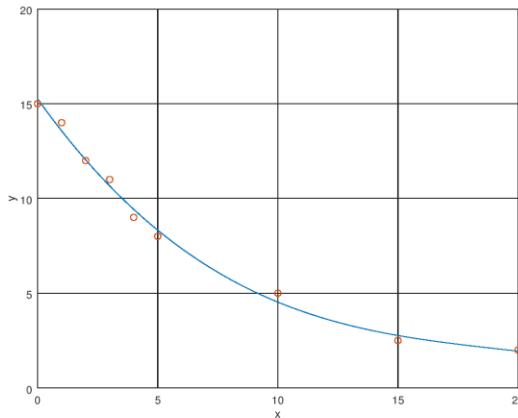
Grado del polinomio = 3

p =

```
-1.46508369953115e-003 8.46454060734041e-002 -1.77265252864800e+000  
1.52550877328575e+001
```

Correlación: 0.9974

>>



Entonces, se acepta el siguiente polinomio:

$$m = -1.4651 \times 10^{-3} t^3 + 8.4645 \times 10^{-2} t^2 - 1.7727 t + 1.5255 \times 10^{-1}$$

Aplicando diferencia numérica para $t = 24$ h,

*** Diferencias Finitas ***.

Introduce una función con respecto a x: $-1.4651e-3*x^3+8.4645e-2*x^2-$
 $1.7727*x+1.5255e1$

Incremento de x = 0.1
x = 24

?Hasta qué diferencia quieres calcular (1-4)? 1

Valor analítico: -0.241433

* Aproximaciones centrada de cuarto orden:

-0.241433

>>

Entonces, cuando $t = 24$ h, se entrega un flujo másico de: $0.2414 \cdot 12 = \mathbf{1.8924 \text{ mg/h}}$.

9.39². Se miden las siguientes desviaciones a lo largo de una viga uniforme con soporte simple,

$x, \text{ m}$	0	0.375	0.75	1.125	1.5	1.875	2.25	2.625	3
$y, \text{ m}$	0	-0.2571	-0.9484	-1.9689	-3.2262	-4.6414	-6.1503	-7.7051	-9.275

Usar diferenciación numérica para calcular la pendiente, el momento (en N m), el cortante (en N) y la carga distribuida (en N/m). Usar los siguientes parámetros en el cálculo: $E = 200 \text{ GPa}$ y $I = 0.0004 \text{ m}^4$.

Para analizar las vigas sujetas a cargas distribuidas se pueden utilizar las siguientes relaciones:

$$\frac{dy}{dx} = \theta(x), \quad \frac{d\theta}{dx} = \frac{M(x)}{EI}, \quad \frac{dM}{dx} = V(x), \quad \frac{dV}{dx} = -w(x)$$

donde x = distancia en la viga (m), y = deflexión (m), $\theta(x)$ = pendiente (m/m), E = módulo de elasticidad ($\text{Pa} = \text{N/m}^2$), I = momento de inercia (m^4), $M(x)$ = momento (N m), $V(x)$ = cortante (N) y $w(x)$ = carga distribuida (N/m).

Solución.

Aproximando polinomialmente:

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

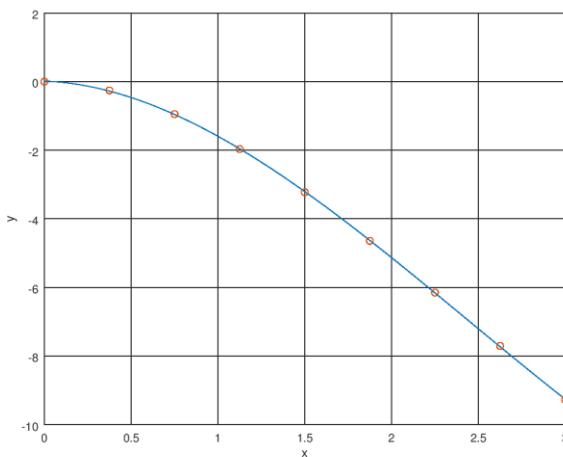
Detected XLS interfaces:

Grado del polinomio = 3
 $p =$

0.2228818555929664 -1.6320915824915805 -0.2011468013468053
0.0145696969696986

Correlación: 1.0000

>>



Entonces, las desviaciones a lo largo de la viga se comportan según la siguiente función:

$$y = 0.2229x^3 - 1.6321x^2 - 0.2011x + 0.0146$$

$$\theta(x) = \frac{dy}{dx} = 0.6686x^2 - 3.2642x - 0.2011$$

$$M(x) = EI \frac{d\theta}{dx} = 200 \times 10^6 (0.0004) (1.3373x - 3.2642) = 80000(1.3373x - 3.2642)$$

$$V(x) = \frac{dM}{dx} = 80000(1.3373) = 106983$$

$$\frac{dV}{dx} = -w(x)$$

9.40². Los datos siguientes para el calor específico del benceno fueron generados con un polinomio de grado n . Usar diferenciación numérica para determinar n .

$T, {}^\circ\text{K}$	300	400	500	600	700	800	900	1000
$C_p, \text{kJ}/(\text{kmol } {}^\circ\text{K})$	82.888	112.136	136.933	157.744	175.036	189.273	200.923	210.45

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,

si ya lo creaste, oprime Enter

Detected XLS interfaces:

Grado del polinomio = 3

p =

```
7.76338383838422e-008 -3.15727056277064e-004 4.84765652958158e-001 -
3.62224047619058e+001
```

Correlación Pearson: 1.0000

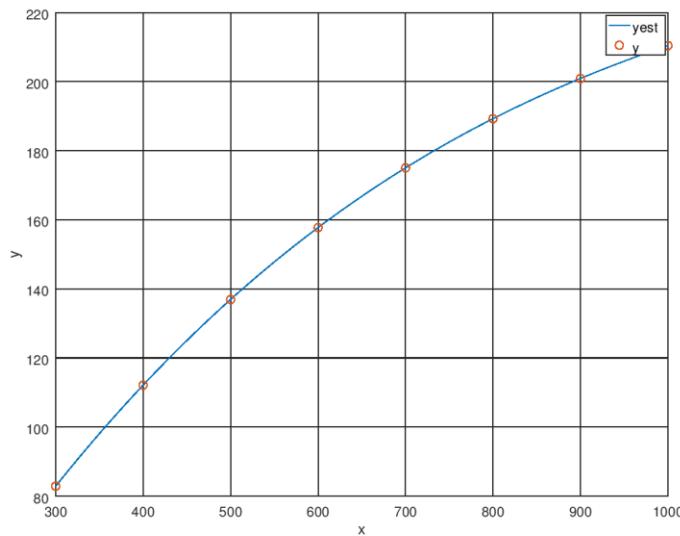
pval: 0

ans =

scalar structure containing the fields:

```
method = Pearson's product moment correlation
params = 6
stat = 558194.897262097
dist = t
pval = 0
alternative = !=
```

>>



9.41⁶. En la siguiente tabla

<i>t</i>	0	1	2	3	4	5
<i>T</i>	93.1	85.9	78.8	75.1	69.8	66.7

donde T representa la temperatura ($^{\circ}\text{C}$) de una salmuera utilizada como refrigerante y t (min) es el tiempo. Encuentre la velocidad de enfriamiento en los tiempos $t = 2.5$ y $t = 4$ min.

Solución.

Ajustando una función a la serie de datos, se obtiene lo siguiente:

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 5
p =
```

Columns 1 through 4:

```
0.975833333333318 -11.77083333333121 48.44583333332267 -
77.52916666664139
```

Columns 5 and 6:

```
32.67833333330414 93.10000000001415
```

Correlación Pearson: 1.0000

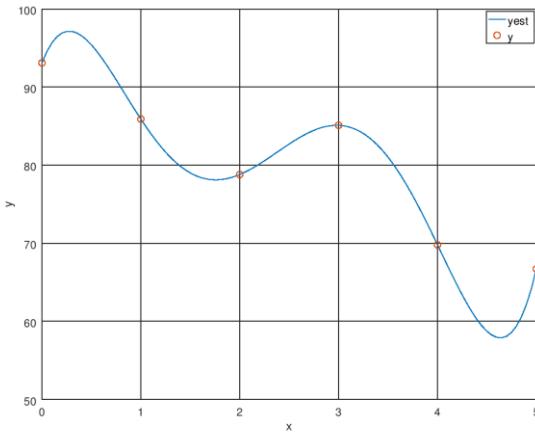
pval: 0

ans =

scalar structure containing the fields:

```
method = Pearson's product moment correlation
params = 4
stat = Inf
dist = t
pval = 0
alternative = !=
```

>>



Entonces, $dT / dt = 93.1 + 32.6783x - 77.5292x^2 + 48.4458x^3 - 11.7708x^4 + 0.9758x^5$

Crear previamente los vectores X y Y en excel.

En caso de no haberlo hecho aún, oprime Ctrl c y créalo,
si ya lo creaste, oprime Enter

Detected XLS interfaces:

```
Grado del polinomio = 4
p =
```

Columns 1 through 4:

```
0.0104166666666681    -0.145833333333441    1.127083333333456    -
8.5988095238095124
```

Column 5:

```
93.1678571428571871
```

Correlación Pearson: 0.9988

pval: 2.07001e-006

ans =

scalar structure containing the fields:

```
method = Pearson's product moment correlation
params = 4
stat = 41.2210719455531
dist = t
```

```

pval = 2.07000837892934e-006
alternative = !=

>> syms f(x)
>> f=93.1679-8.5988*x+1.1271*x^2-0.1458*x^3+0.0104*x^4;
>> diff(f)
ans = (sym)

      3      2
26*x   2187*x   1420*pi*x   1436
----- - ----- + ----- - -----
       625      5000      1979      167

>> deriv=diff(f)
deriv = (sym)

      3      2
26*x   2187*x   1420*pi*x   1436
----- - ----- + ----- - -----
       625      5000      1979      167

>> deriv=function_handle(deriv)
deriv =

@(x) 26 * x .^ 3 / 625 - 2187 * x .^ 2 / 5000 + 1420 * pi * x / 1979 - 1436 / 167

>> deriv(2.5)
ans = -5.04705268816372
>> deriv(4)
ans = -3.91800286393620
>>

```

Integración numérica.

9.42². La integración provee un medio para calcular cuánta masa entra o sale de un reactor en un tiempo especificado,

$$M = \int_{t_1}^{t_2} Q c dt$$

donde t_1 y t_2 = tiempos inicial y final, respectivamente. Así, la integral representa la suma del producto de la concentración de los tiempos de flujo para dar la masa total entrando o saliendo de t_1 a t_2 . Si la tasa de flujo es constante, Q puede ser movida fuera de la integral:

$$M = Q \int_{t_1}^{t_2} c dt$$

Usar integración numérica para evaluar esta ecuación para los datos listados abajo. Notar que $Q = 4 \text{ m}^3/\text{min}$.

$t, \text{ min}$	0	10	20	30	35	40	45	50
$c, \text{ mg/m}^3$	10	35	55	52	40	37	32	34

Solución.

Para resolver este ejemplo primero se aproximarán polinomialmente una función a la serie de datos y posteriormente se aplicarán las técnicas numéricas vistas.

Crear previamente los vectores x y y en excel.
si ya lo creaste oprime Enter si no, oprime Ctrl C y Enter.

Detected XLS interfaces:
Grado del polinomio? 6

Coeficientes del polinomio ajustado

10.006

1.9865

0.046225

0.0045961

-0.00053795

0.000013380

-0.00000010141

Coef. de correlación Pearson: 0.9959
Coef. de determinación Pearson: 0.9918
Coef. de correlación Kendall: 1.0000
Coef. de correlación Spearman: 1.0000

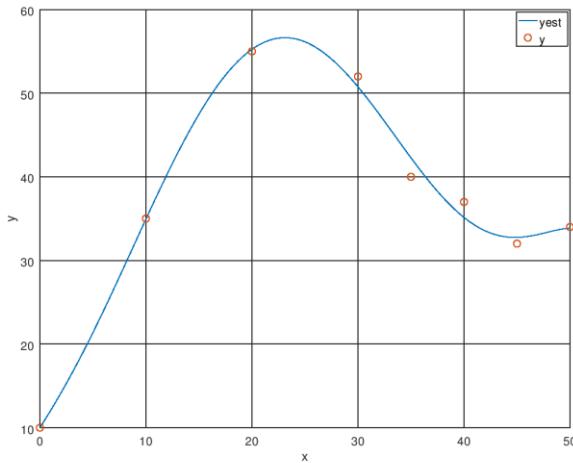
Sumatoria de desviaciones al cuadrado: 10.798392

Error cuadrado medio: 1.161809

Lim. sup.	Lim. inf.
64.037299	-5.048995
63.964727	0.158080
64.898618	4.358692
67.004878	7.386935
68.519884	8.439181
70.336586	9.189729
72.439621	9.653946
74.807322	9.853497

Introduce el valor de x a estimar.
cuando termines de estimar oprime Ctrl C.

```
x =
>>
```



$$y = 10.0059 + 1.986528x + 4.622496 \times 10^{-2}x^2 - 5.379551 \times 10^{-4}x^3 + 1.337974 \times 10^{-5}x^4 - 1.014083 \times 10^{-7}x^5$$

Integrando con comandos en Octave,

```
>> f =@(x) 1.00059072166269e+001 + 1.98652754208209 * x + 4.62249575748035e-002 *
x.^2 + 4.596103680655842e-03 * x.^3 - 5.379512739258502e-04 * x.^4 +
1.337974040745910e-05 * x.^5 - 1.014083267317531e-07*x.^6;
>> Integral = quad(f,0,50);
>> fprintf('%10.6f \n',Integral)
1994.132437
>>
```

Ahora, numéricamente,

```
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x    1.00059072166269e+001 + 1.98652754208209
* x + 4.62249
575748035e-002 * x.^2 + 4.596103680655842e-03 * x.^3 - 5.379512739258502e-04 *
x.^4 + 1.337
974040745910e-05 * x.^5 - 1.014083267317531e-07*x.^6
Límite inferior = 0
Límite superior = 50
```

Número de paneles = 50

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	10.005907
1	1.000000	12.042730
2	2.000000	14.192444
3	3.000000	16.465211
4	4.000000	18.861335
5	5.000000	21.372686
6	6.000000	23.984050
7	7.000000	26.674403
8	8.000000	29.418119
9	9.000000	32.186099
10	10.000000	34.946827
11	11.000000	37.667360
12	12.000000	40.314235
13	13.000000	42.854311
14	14.000000	45.255532
15	15.000000	47.487625
16	16.000000	49.522714
17	17.000000	51.335869
18	18.000000	52.905583
19	19.000000	54.214168
20	20.000000	55.248086
21	21.000000	55.998202
22	22.000000	56.459966
23	23.000000	56.633525
24	24.000000	56.523754
25	25.000000	56.140221
26	26.000000	55.497077
27	27.000000	54.612872
28	28.000000	53.510300
29	29.000000	52.215868
30	30.000000	50.759495
31	31.000000	49.174035
32	32.000000	47.494732
33	33.000000	45.758592
34	34.000000	44.003699
35	35.000000	42.268437
36	36.000000	40.590657
37	37.000000	39.006758
38	38.000000	37.550707
39	39.000000	36.252973
40	40.000000	35.139400
41	41.000000	34.229996
42	42.000000	33.537660
43	43.000000	33.066827
44	44.000000	32.812046
45	45.000000	32.756479
46	46.000000	32.870334
47	47.000000	33.109219
48	48.000000	33.412427

```
49 49.000000 33.701145  
50 50.000000 33.876593
```

I = 1993.978045

*** Regla trapezoidal con corrección ***.

```
f'(li)= 1.986527  
f'(ls)= 0.080510
```

I = 1994.136880

>>

Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.
*** Regla de Simpson ***.

```
Introduce una función con respecto a x 1.00059072166269e+001 + 1.98652754208209  
* x + 4.62249  
575748035e-002 * x .^ 2 + 4.596103680655842e-03 * x .^ 3 -5.379512739258502e-04 *  
x .^ 4 + 1.337  
974040745910e-05 * x .^ 5-1.014083267317531e-07*x.^6  
Límite inferior = 0  
Límite superior = 50
```

Número de paneles = 50

*** Regla de Simpson sin corrección ***.

```
0 0.000000 10.005907  
1 1.000000 12.042730  
2 2.000000 14.192444  
3 3.000000 16.465211  
4 4.000000 18.861335  
5 5.000000 21.372686  
6 6.000000 23.984050  
7 7.000000 26.674403  
8 8.000000 29.418119  
9 9.000000 32.186099  
10 10.000000 34.946827  
11 11.000000 37.667360  
12 12.000000 40.314235  
13 13.000000 42.854311  
14 14.000000 45.255532  
15 15.000000 47.487625  
16 16.000000 49.522714  
17 17.000000 51.335869  
18 18.000000 52.905583  
19 19.000000 54.214168  
20 20.000000 55.248086  
21 21.000000 55.998202  
22 22.000000 56.459966
```

23	23.000000	56.633525
24	24.000000	56.523754
25	25.000000	56.140221
26	26.000000	55.497077
27	27.000000	54.612872
28	28.000000	53.510300
29	29.000000	52.215868
30	30.000000	50.759495
31	31.000000	49.174035
32	32.000000	47.494732
33	33.000000	45.758592
34	34.000000	44.003699
35	35.000000	42.268437
36	36.000000	40.590657
37	37.000000	39.006758
38	38.000000	37.550707
39	39.000000	36.252973
40	40.000000	35.139400
41	41.000000	34.229996
42	42.000000	33.537660
43	43.000000	33.066827
44	44.000000	32.812046
45	45.000000	32.756479
46	46.000000	32.870334
47	47.000000	33.109219
48	48.000000	33.412427
49	49.000000	33.701145
50	50.000000	33.876593

suma par: 974.811180 suma impar: 997.225615

I = 1994.135773

*** Regla de Simpson con corrección ***.

f'(a) = 1.986527 f'(b) = 0.080510

I = 1994.136658

>>

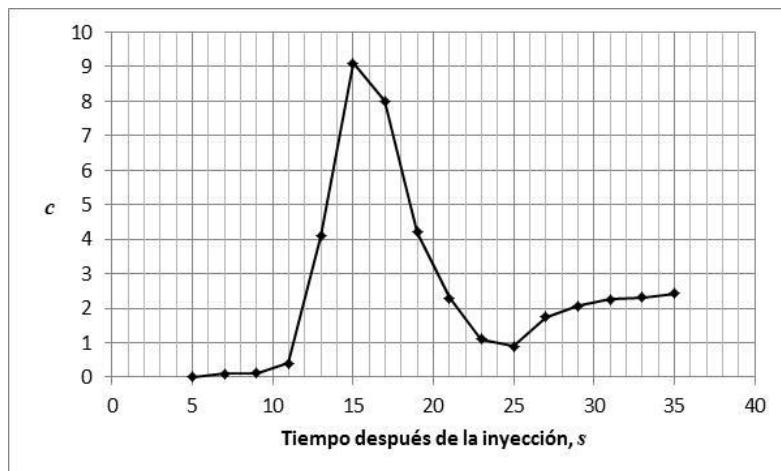
9.43². La técnica estándar para determinar el flujo cardíaco es el método de dilución del indicador desarrollado por Hamilton. Un extremo de un catéter pequeño se inserta en la arteria radial y el otro extremo está conectado a un densitómetro, que puede registrar automáticamente la concentración del colorante en la sangre. Se inyecta rápidamente una cantidad conocida de colorante, 5.6 mg, y se obtienen los siguientes datos:

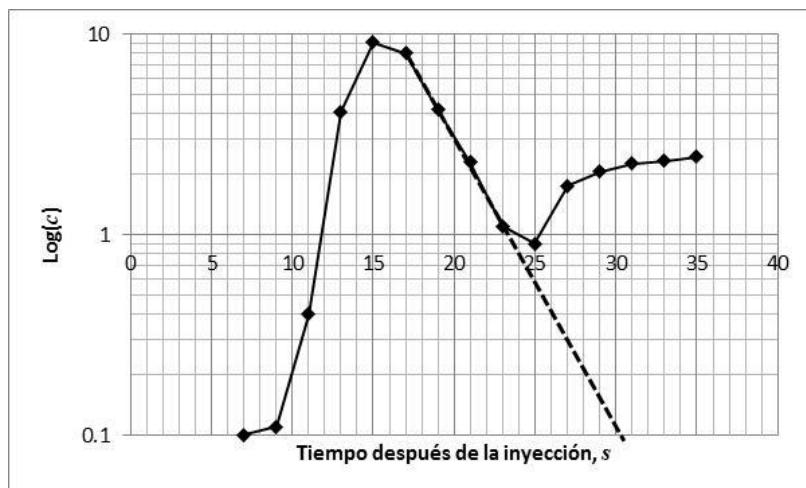
Tiempo, s	Concentración, mg/l	Tiempo, s	Concentración, mg/l
5	0	21	2.3
7	0.1	23	1.1
9	0.11	25	0.9
11	0.4	27	1.75
13	4.1	29	2.06
15	9.1	31	2.25
17	8	33	2.32
19	4.2	35	2.43

Graficando los datos anteriores resulta en la curva de dilución del tinte en la primera figura. La concentración alcanza un valor máximo en unos 15 segundos y luego se cae, seguida de un aumento debido a la recirculación del colorante. La curva se vuelve a trazar en un gráfico semi-log. Observar que una línea recta aproxima la extremidad descendente de la curva de dilución. Para separar el efecto de la recirculación, los analistas amplían la porción de línea recta. El gasto cardíaco puede calcularse a partir de

$$C = \frac{M}{A} \times 60 \text{ s/min}$$

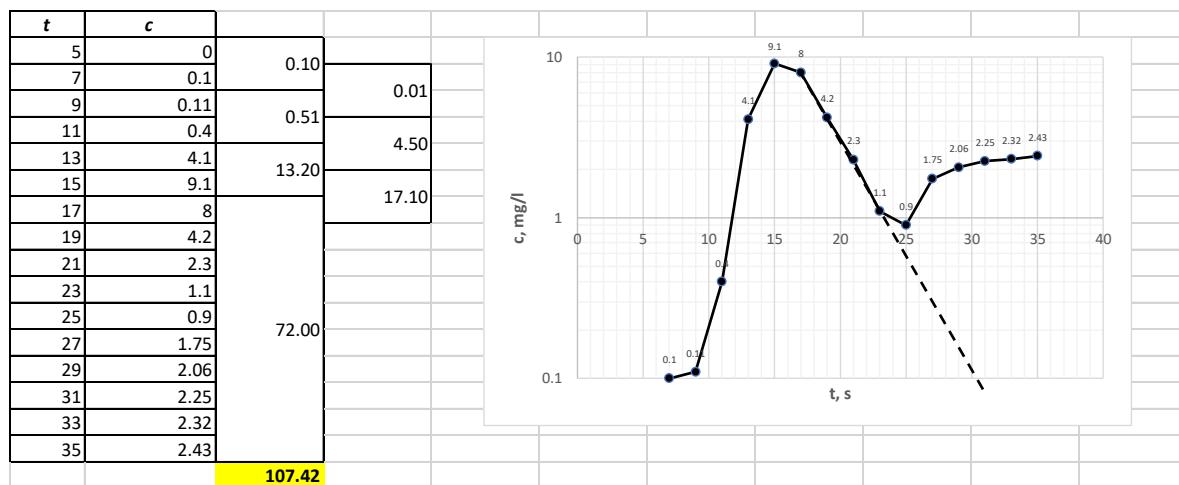
donde C = salida cardiaca (l/min), M = cantidad de líquido diluido inyectado (mg) y A = área bajo la curva con corrección lineal. Calcular la salida cardíaca de este paciente usando la regla trapezoidal con $h = 2 \text{ s}$.




Solución.

Se aprecia que la línea recta intersecta el eje de las abscisas en $t = 30.5$ s, aproximadamente. Entonces. El área bajo de la curva desde $s = 5$ s hasta $t = 30.5$ s se puede obtener con la regla trapezoidal o también ajustando una curva mediante aproximación funcional y luego integrando numéricamente.

Entonces, encontrando las áreas debajo de la gráfica con ordenadas logarítmicas se obtiene un área de 107.42 mg-s/l, como se muestra a continuación,



Por otro lado, ajustando un polinomio de 4to grado desde $t = 7$ hasta $t = 17$ usando Octave, se obtiene lo siguiente:

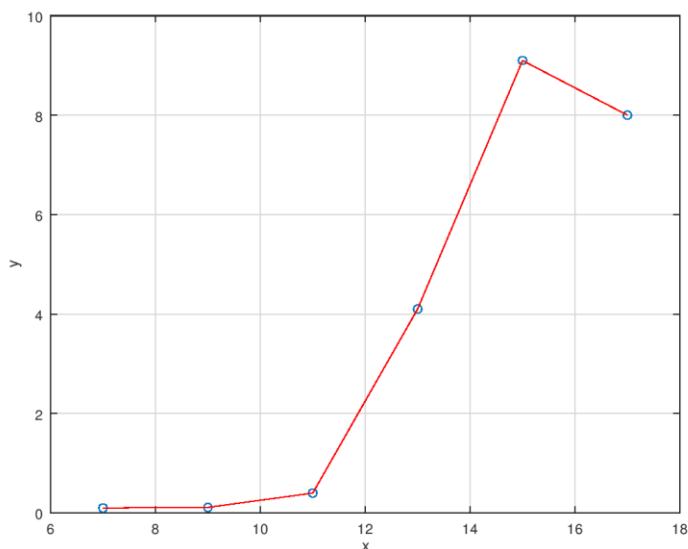
Crear previamente los vectores x y y en excel.
si ya lo creaste oprime Enter si no, oprime Ctrl C.

Detected XLS interfaces:
Grado del polinomio? 4

Coeficientes del polinomio ajustado

-166.638805
67.195607
-9.821849
0.613877
-0.013711

Coef. de determinación: 1.0000
>>



Siendo la ecuación ajustada,

$$y = -166.638805 + 67.195607*x -9.821849*x^2 +0.613877*x^3 -0.013711*x^4$$

Al integrar por la regla de Simpson, también usando Octave:

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.

```
Introduce una función con respecto a x      -166.638805 + 67.195607*x -9.821849*x^2
+0.613877*x^3 -
0.013711*x^4
Límite inferior = 7
Límite superior = 17
```

```
Número de paneles = 10
```

```
*** Regla de Simpson sin corrección ***.
```

0	7.000000	0.099588
1	8.000000	0.472541
2	9.000000	0.110424
3	10.000000	-0.100543
4	11.000000	0.396794
5	12.000000	1.830526
6	13.000000	4.099680
7	14.000000	6.774218
8	15.000000	9.095039
9	16.000000	9.973979
10	17.000000	7.993809

```
suma par: 13.701937    suma impar: 18.950721
```

```
I = 37.100052
```

```
*** Regla de Simpson con corrección ***.
```

```
f(a) = 1.118160    f(b) = -3.964401
```

```
I = 37.118333
```

```
>>
```

Entonces, el área resultante es: 37.12 mg-s/l

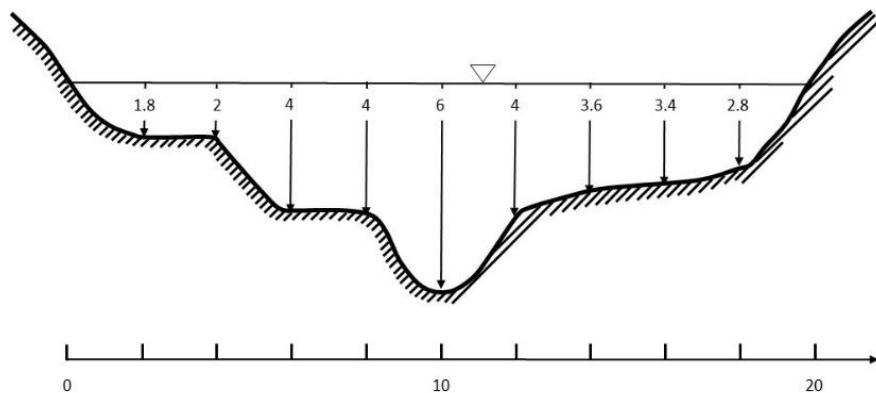
Y el área total será: $0.1 + 31.12 + 72 = 103.22 \text{ mg-s/l}$. Un poco menor, 4.20 (3.91%), que la obtenida con el procedimiento anterior. Entonces, sustituyendo ambos resultados se obtiene:

$$C = \frac{5.6}{107.42} \times 60 = 3.13 \text{ l/min}$$

$$C = \frac{5.6}{103.22} \times 60 = 3.26 \text{ l/min}$$

9.44². Se requieren áreas transversales de una corriente (A) para una serie de trabajos en ingeniería de recursos hídricos, que incluyen predicción de inundaciones y diseño de reservorios. A

menos que haya disponibles dispositivos de sonido electrónico para obtener perfiles continuos del fondo del canal, el ingeniero debe confiar en mediciones de profundidad discretas para calcular A. En la figura se muestra un ejemplo de una sección transversal típica del flujo. Los puntos de datos representan lugares donde se ancló un bote y se tomaron lecturas de profundidad. Utilizar la regla trapezoidal para dos incrementos ($h = 4$ y 2 m) y la regla de Simpson de $1/3$ ($h = 2$ m) para estimar el área de la sección transversal a partir de estos datos.



Solución.

$\int_a^b f(x) dx \approx \frac{h}{2} \left(f_0 + f_n + 2 \sum_{i=1}^{n-1} f_i \right)$	$I = \frac{h}{3} \left(f_0 + f_n + 4 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + 2 \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right)$																																																																								
<table border="1"> <thead> <tr> <th>i</th><th>x_i</th><th>y_i</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1.8</td></tr> <tr><td>2</td><td>4</td><td>2</td></tr> <tr><td>3</td><td>6</td><td>4</td></tr> <tr><td>4</td><td>8</td><td>4</td></tr> <tr><td>5</td><td>10</td><td>6</td></tr> <tr><td>6</td><td>12</td><td>4</td></tr> <tr><td>7</td><td>14</td><td>3.6</td></tr> <tr><td>8</td><td>16</td><td>3.4</td></tr> <tr><td>9</td><td>18</td><td>2.8</td></tr> <tr><td>10</td><td>20</td><td>0</td></tr> </tbody> </table> <p style="text-align: right;">$I = 63.20$</p>	i	x_i	y_i	0	0	0	1	2	1.8	2	4	2	3	6	4	4	8	4	5	10	6	6	12	4	7	14	3.6	8	16	3.4	9	18	2.8	10	20	0	<table border="1"> <thead> <tr> <th>i</th><th>x_i</th><th>y_i</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1.8</td></tr> <tr><td>2</td><td>4</td><td>2</td></tr> <tr><td>3</td><td>6</td><td>4</td></tr> <tr><td>4</td><td>8</td><td>4</td></tr> <tr><td>5</td><td>10</td><td>6</td></tr> <tr><td>6</td><td>12</td><td>4</td></tr> <tr><td>7</td><td>14</td><td>3.6</td></tr> <tr><td>8</td><td>16</td><td>3.4</td></tr> <tr><td>9</td><td>18</td><td>2.8</td></tr> <tr><td>10</td><td>20</td><td>0</td></tr> </tbody> </table> <p style="text-align: right;">Suma_par = 13.4 Suma_impar = 18.2 $I = 66.4$</p>	i	x_i	y_i	0	0	0	1	2	1.8	2	4	2	3	6	4	4	8	4	5	10	6	6	12	4	7	14	3.6	8	16	3.4	9	18	2.8	10	20	0
i	x_i	y_i																																																																							
0	0	0																																																																							
1	2	1.8																																																																							
2	4	2																																																																							
3	6	4																																																																							
4	8	4																																																																							
5	10	6																																																																							
6	12	4																																																																							
7	14	3.6																																																																							
8	16	3.4																																																																							
9	18	2.8																																																																							
10	20	0																																																																							
i	x_i	y_i																																																																							
0	0	0																																																																							
1	2	1.8																																																																							
2	4	2																																																																							
3	6	4																																																																							
4	8	4																																																																							
5	10	6																																																																							
6	12	4																																																																							
7	14	3.6																																																																							
8	16	3.4																																																																							
9	18	2.8																																																																							
10	20	0																																																																							
<table border="1"> <thead> <tr> <th>i</th><th>x_i</th><th>y_i</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1.8</td></tr> <tr><td>2</td><td>4</td><td>2</td></tr> <tr><td>3</td><td>6</td><td>4</td></tr> <tr><td>4</td><td>8</td><td>4</td></tr> <tr><td>5</td><td>10</td><td>6</td></tr> <tr><td>6</td><td>12</td><td>4</td></tr> <tr><td>7</td><td>14</td><td>3.6</td></tr> <tr><td>8</td><td>16</td><td>3.4</td></tr> <tr><td>9</td><td>18</td><td>2.8</td></tr> <tr><td>10</td><td>20</td><td>0</td></tr> </tbody> </table> <p style="text-align: right;">$I = 53.6$</p>	i	x_i	y_i	0	0	0	1	2	1.8	2	4	2	3	6	4	4	8	4	5	10	6	6	12	4	7	14	3.6	8	16	3.4	9	18	2.8	10	20	0	<table border="1"> <thead> <tr> <th></th><th></th><th></th></tr> </thead> <tbody> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td></td><td>12</td></tr> <tr><td></td><td>16</td><td></td></tr> <tr><td></td><td></td><td>14.8</td></tr> <tr><td></td><td>6.8</td><td></td></tr> </tbody> </table>					4				12		16				14.8		6.8																			
i	x_i	y_i																																																																							
0	0	0																																																																							
1	2	1.8																																																																							
2	4	2																																																																							
3	6	4																																																																							
4	8	4																																																																							
5	10	6																																																																							
6	12	4																																																																							
7	14	3.6																																																																							
8	16	3.4																																																																							
9	18	2.8																																																																							
10	20	0																																																																							
	4																																																																								
		12																																																																							
	16																																																																								
		14.8																																																																							
	6.8																																																																								

Considerando que la técnica de la regla de Simpson ajusta a través de paráolas, tal vez sea la más indicada para representar el área solicitada.

9.45². Como se describió en el problema anterior, el área transversal de un canal se puede calcular como:

$$A_c = \int_0^B H(y) dy$$

donde B = ancho total del canal (m), H = tirante (m) y y = distancia desde el banco (m). En forma similar, el flujo promedio Q (m^3/s) se puede calcular como:

$$Q = \int_0^B U(y) H(y) dy$$

donde U es la velocidad (m/s). Usar estas relaciones y un método numérico para determinar A_c y Q para los siguientes datos:

Solución.

Ajustando polinomialmente para el área transversal, A_c , y para el gasto, Q ,

Crear previamente los vectores x y y en excel.
si ya lo creaste oprime Enter si no, oprime Ctrl C.

Detected XLS interfaces:
Grado del polinomio? 5

Coeficientes del polinomio ajustado

0.500000
4.883056
-4.457315
1.470671
-0.200324
0.009468

Coef. de determinación: 1.0000
>>

$$H(y) = 0.500000 + 4.883056 * y - 4.457315 * y^2 + 1.470671 * y^3 - 0.200324 * y^4 + 0.009468 * y^5$$

Crear previamente los vectores x y y en excel.

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

si ya lo creaste oprime Enter si no, oprime Ctrl C.

Detected XLS interfaces:
Grado del polinomio? 5

Coeficientes del polinomio ajustado

```
0.030000  
0.408794  
-0.391247  
0.128024  
-0.016987  
0.000781
```

Coef. de determinación: 1.0000
>>

$$U(y) = 0.030000 + 0.408794 \cdot x - 0.391247 \cdot x^2 + 0.128024 \cdot x^3 - 0.016987 \cdot x^4 + 0.000781 \cdot x^5$$

$$(0.500000 + 4.883056 \cdot x - 4.457315 \cdot x^2 + 1.470671 \cdot x^3 - 0.200324 \cdot x^4 + 0.009468 \cdot x^5) * (0.030000 + 0.408794 \cdot x - 0.391247 \cdot x^2 + 0.128024 \cdot x^3 - 0.016987 \cdot x^4 + 0.000781 \cdot x^5)$$

Entonces, de la fórmula para determinar el área transversal,

$$A_c = \int_0^9 H(y) dy$$

Resolviendo con las reglas trapezoidal y de Simpson se obtiene lo siguiente:

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.

Introduce una función con respecto a x 0.500000+4.883056*x-
4.457315*x^2+1.470671*x^3-0.200324*
x^4+0.009468*x^5
Límite inferior = 0
Límite superior = 9

Número de paneles = 10

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.500000
1	0.900000	2.230601
2	1.800000	1.500732

```
3    2.700000    0.850146
4    3.600000    1.005996
5    4.500000    1.553729
6    5.400000    1.607974
7    6.300000    0.483433
8    7.200000    -1.634226
9    8.100000    -3.017480
10   9.000000    0.273942
```

```
I = 4.471090
```

```
*** Regla trapezoidal con corrección ***.
```

```
f(li)= 4.883055
f(ls)= 8.477277
```

```
I = 4.228480
>>
```

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x  0.500000+4.883056*x-
4.457315*x^2+1.470671*x^3-0.200324*
x^4+0.009468*x^5
Límite inferior = 0
Límite superior = 9
```

```
Número de paneles = 10
```

```
*** Regla de Simpson sin corrección ***.
```

```
0    0.000000    0.500000
1    0.900000    2.230601
2    1.800000    1.500732
3    2.700000    0.850146
4    3.600000    1.005996
5    4.500000    1.553729
6    5.400000    1.607974
7    6.300000    0.483433
8    7.200000    -1.634226
9    8.100000    -3.017480
10   9.000000    0.273942
suma par: 2.480476  suma impar: 2.100430
```

```
I = 4.240984
```

```
*** Regla de Simpson con corrección ***.
```

```
f(a) = 4.883055   f(b) = 8.477277
I = 4.230981
>>
```

Considerando que generalmente la regla de Simpson con corrección es la más exacta, se sugiere tomar $A_c = 4.23 \text{ m}^2$.

Por otro lado, el gasto se calcula con la fórmula

$$Q = \int_0^9 U(y) H(y) dy$$

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x (0.500000+4.883056*x-
4.457315*x^2+1.470671*x^3-0.200324
*x^4+0.009468*x^5)*( 0.030000+0.408794*x-0.391247*x^2+0.128024*x^3-
0.016987*x^4+0.000781*x^5)
Lmite inferior = 0
Lmite superior = 9
```

Número de paneles = 10

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.015000
1	0.900000	0.365038
2	1.800000	0.121949
3	2.700000	0.009144
4	3.600000	0.023402
5	4.500000	0.137415
6	5.400000	0.208772
7	6.300000	0.038772
8	7.200000	0.103178
9	8.100000	0.552073
10	9.000000	0.003616

I = 1.412147

*** Regla trapezoidal con corrección ***.

```
f(li)= 0.350889
f(ls)= 0.266060
```

```
I = 1.417873
>>
```

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x (0.500000+4.883056*x-
4.457315*x^2+1.470671*x^3-0.200324
*x^4+0.009468*x^5)*( 0.030000+0.408794*x-0.391247*x^2+0.128024*x^3-
0.016987*x^4+0.000781*x^5)
Límite inferior = 0
límite superior = 9
```

Número de paneles = 10

```
*** Regla de Simpson sin corrección ***.
```

0	0.000000	0.015000
1	0.900000	0.365038
2	1.800000	0.121949
3	2.700000	0.009144
4	3.600000	0.023402
5	4.500000	0.137415
6	5.400000	0.208772
7	6.300000	0.038772
8	7.200000	0.103178
9	8.100000	0.552073
10	9.000000	0.003616
suma par:	0.457302	suma impar: 1.102443

I = 1.602897

```
*** Regla de Simpson con corrección ***.
```

f(a) = 0.350889 f(b) = 0.266060

I = 1.454878
>>

Se opta por proponer $Q = 1.45 \text{ m}^3/\text{s}$.

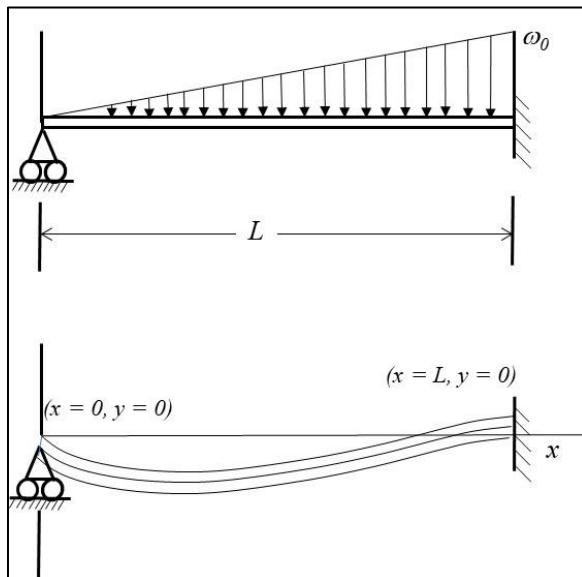
9.46². Las siguientes relaciones puedes ser usadas para analizar vigas uniformes sujetas a cargas distribuidas

$$\frac{dy}{dx} = \theta(x) \quad \frac{d\theta}{dx} = \frac{M(x)}{E I} \quad \frac{dM}{dx} = V(x) \quad \frac{dV}{dx} = -\omega(x)$$

donde x = distancia a lo largo de la viga (m), y = deflexión (m), $\theta(x)$ = pendiente (m/m), E = módulo de elasticidad ($\text{Pa} = \text{N}/\text{m}^2$), I = momento de inercia (m^4), $M(x)$ = momento (N m), $V(x)$ = cortante y ω = carga distribuida (N/m). Para el caso de un incremento en la carga lineal (ver figura), la pendiente puede ser calculada analíticamente como

$$\theta(x) = \frac{\omega_0}{120 E I L} (-5x^4 + 6L^2x^2 - L^4)$$

Usar integración numérica para calcular la deflexión (en m). Para calcular la pendiente usar intervalos igualmente espaciados de $\Delta x = 0.125$ m a lo largo de 3 m de viga. Usar los siguientes parámetros en los cálculos: $E = 200 \text{ GPa}$, $I = 0.0003 \text{ m}^4$ y $\omega_0 = 2.5 \text{ kN/cm}$. Además, la deflexión en las puntas de la viga son $y(0) = y(L) = 0$.



Solución.

$$\frac{dy}{dx} = \theta(x) = \frac{\omega_0}{120 E I L} (-5x^4 + 6L^2x^2 - L^4)$$

$$\begin{aligned} \frac{dy}{dx} = \theta(x) &= \frac{250}{120 (200 \times 10^{-6})(0.0003)(3)} (-5x^4 + 54x^2 - 9) \\ &= 11.57407 \times 10^{-6} (-5x^4 + 54x^2 - 9) \end{aligned}$$

Integrando numéricamente con las reglas Trapezoidal y de Simpson.

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.  
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x 11.57407e-6*(-5*x^4+54*x^2-9)  
Límite inferior = 0  
Límite superior = 3
```

```
Número de paneles = 240
```

```
*** Regla trapezoidal sin corrección ***.
```

i	xi	fi
0	0.000000	-0.000104
1	0.012500	-0.000104
2	0.025000	-0.000104
3	0.037500	-0.000103
4	0.050000	-0.000103
5	0.062500	-0.000102
...		
235	2.937500	0.000980
236	2.950000	0.000952
237	2.962500	0.000924
238	2.975000	0.000894
239	2.987500	0.000864
240	3.000000	0.000833

```
I = 0.002500
```

```
*** Regla trapezoidal con corrección ***.
```

```
f(li)= 0.000000  
f(ls)= -0.002500
```

```
I = 0.002500  
>>
```

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.  
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x 11.57407e-6*(-5*x^4+54*x^2-9)  
Límite inferior = 0  
Límite superior = 3
```

Número de paneles = 240

*** Regla de Simpson sin corrección ***.

```

0      0.000000   -0.000104
1      0.012500   -0.000104
2      0.025000   -0.000104
3      0.037500   -0.000103
4      0.050000   -0.000103
5      0.062500   -0.000102

...
235    2.937500   0.000980
236    2.950000   0.000952
237    2.962500   0.000924
238    2.975000   0.000894
239    2.987500   0.000864
240    3.000000   0.000833
suma par: 0.099630  suma impar: 0.100003

```

I = 0.002500

*** Regla de Simpson con corrección ***.

f(a) = 0.000000 f(b) = -0.002500

I = 0.002500
>>

Entonces, la deflexión es 0.0025 m.

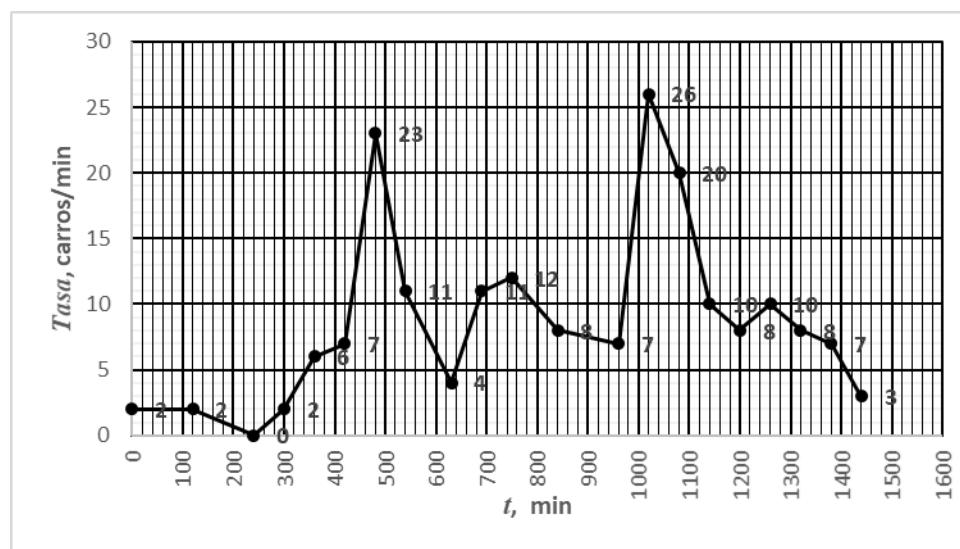
9.47². Un estudio de ingeniería de transporte requiere el cálculo del número total de automóviles que pasan por una intersección en un período de 24 horas. Un individuo visita la intersección en varios momentos durante el transcurso de un día y cuenta la cantidad de autos que pasan a través de la intersección en un minuto. Utiliza los datos resumidos en tabla para estimar el número total de automóviles que pasan por la intersección por día.

<i>Tiempo</i>	<i>Taza</i>	<i>Tiempo</i>	<i>Taza</i>	<i>Tiempo</i>	<i>Taza</i>
12:00 media noche	2	9.00 A.M.	11	6.00 A.M.	20
2.00 A.M.	2	10:30 A.M.	4	7.00 A.M.	10
4.00 A.M.	0	11:30 A.M.	11	8.00 A.M.	8
5.00 A.M.	2	12:30 A.M.	12	9.00 A.M.	10
6:00 A.M.	6	2.00 A.M.	8	10.00 A.M.	8
7:00 A.M.	7	4.00 A.M.	7	11.00 A.M.	7
8:00 A.M.	23	5.00 A.M.	26	12.00 media noche	3

Solución.

Debido a que se tienen incrementos de tiempo desiguales, se optará por determinar el área de cada trapecio bajo la curva. Obteniendo la integral al sumar todas las áreas parciales de cada trapecio.

Graficando los datos se obtiene lo siguiente:



Y calculando,

<i>i</i>	<i>t_i</i> (min)	Tasa (Carros/min)	Trapecios (carros)
0	0	2	
1	120	2	240
2	240	0	120
3	300	2	60
4	360	6	240
5	420	7	390
6	480	23	900
7	540	11	1,020
8	630	4	675
9	690	11	450
10	750	12	690
11	840	8	900

12	960	7	900
13	1020	26	990
14	1080	20	1,380
15	1140	10	900
16	1200	8	540
17	1260	10	540
18	1320	8	540
19	1380	7	450
20	1440	3	300

Total:	12,225
--------	---------------

Por lo tanto, el número estimado de automóviles que pasan por la intersección diariamente son 12,225 carros.

9.48². La presión del agua sobre la cara aguas arriba de una presa se muestra en la figura. La presión puede expresarse como:

$$p(z) = \rho g(D - z)$$

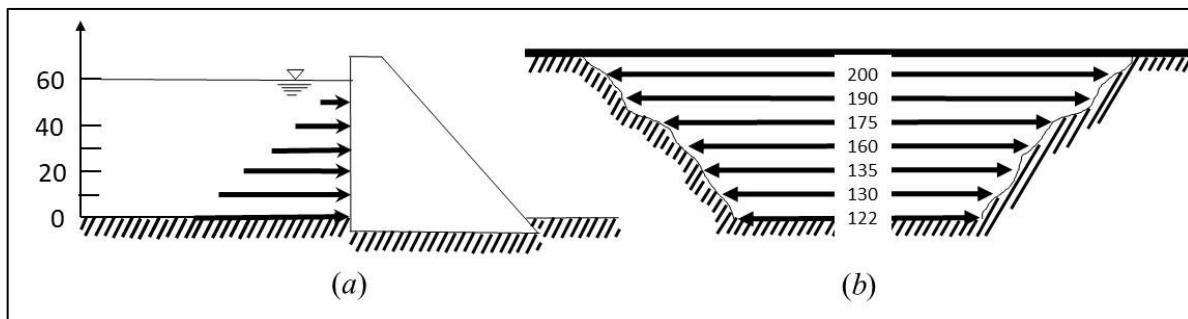
donde $p(z)$ = presión en pascales (o N/m^2) ejercida en una elevación z metros sobre el fondo del reservorio; ρ = densidad del agua, la cual se supone constante de 10^3 kg/m^3 ; g = aceleración debido a la gravedad (9.81 m/s^2) y D = elevación de la superficie del agua sobre el fondo del reservorio (m). Como se observa en la ecuación anterior, la presión se incrementa con la profundidad como se muestra en la figura. Despreciando la presión atmosférica (porque se ejerce en ambos lados de la presa cancelándose), la fuerza total f_t puede determinarse multiplicando la presión por el área de la presa, como se muestra en la figura. Debido a que la presión y el área varían con la elevación, la fuerza total se obtiene evaluando

$$f_t = \int_0^D \rho g(D - z) \omega(z) dz$$

donde $\omega(z)$ = ancho de la cara de la presa (m) en la elevación z . La línea de acción puede también ser obtenida evaluando

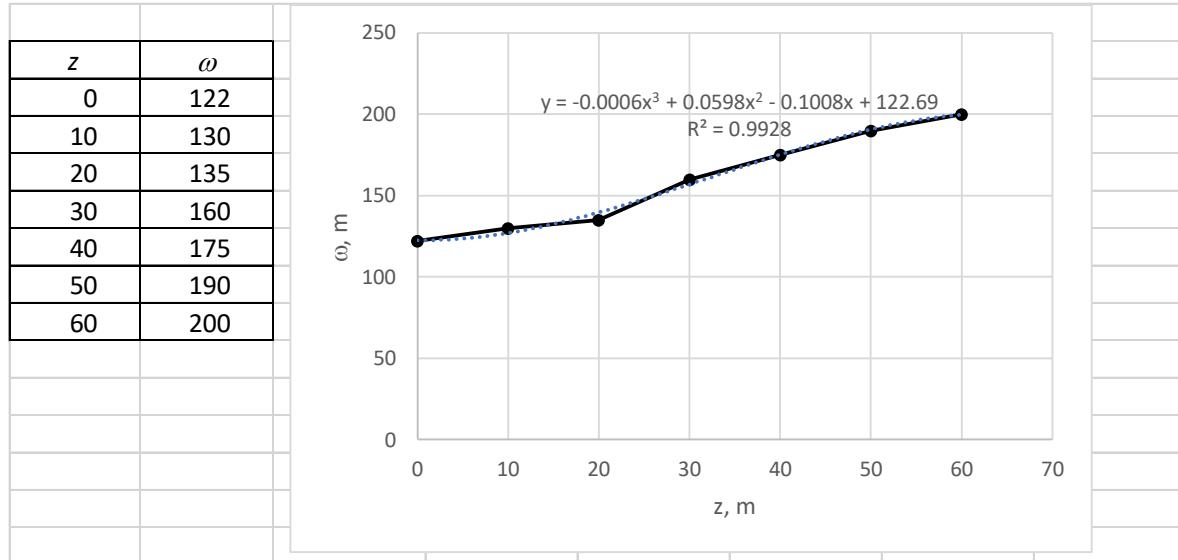
$$d = \frac{\int_0^D \rho g z \omega(z)(D - z) dz}{\int_0^D \rho g \omega(z)(D - z) dz}$$

Usar la regla de Simpson para calcular f_t y d . Comprobar los resultados usando la regla trapezoidal.



Solución.

Ajustando un polinomio de 3er grado a los valores de $\omega(z)$,



Y resolviendo,

$$f_t = \int_0^D \rho g(D-z) \omega(z) dz$$

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.

Introduce una función con respecto a x $9810 * (60-x) * (-0.0006*x^3+0.0598*x^2-0.1008*x+122.69)$

Límite inferior = 0

Límite superior = 60

Número de paneles = 6

*** Regla de Simpson sin corrección ***.

0	0.000000	72215334.000000
1	10.000000	62323911.000000
2	20.000000	54855165.600000
3	30.000000	46289269.800000
4	40.000000	34519035.600000
5	50.000000	18849915.000000
6	60.000000	0.000000

suma par: 89374201.200000 suma impar: 127463095.800000

I = 2536053732.000000

*** Regla de Simpson con corrección ***.

f(a) = -1262919.780000 f(b) = -1984778.820000

I = 2535582852.000000

>>

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.

Introduce una función con respecto a x 9810*(60-x)*(-0.0006*x^3+0.0598*x^2-0.1008*x+122.69)
Límite inferior = 0
Límite superior = 60

Número de paneles = 6

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	72215334.000000
1	10.000000	62323911.000000
2	20.000000	54855165.600000
3	30.000000	46289269.800000
4	40.000000	34519035.600000
5	50.000000	18849915.000000
6	60.000000	0.000000

I = 2529449640.000000

*** Regla trapezoidal con corrección ***.

```
f(li)= -1262919.780000  
f(ls)= -1984778.820000
```

```
I = 2535465132.000000  
>>
```

Por lo tanto, la fuerza total es de 2,535,465,132 N.

Ahora, para calcular d ,

$$d = \frac{\int_0^D \rho g z \omega(z) (D - z) dz}{\int_0^D \rho g \omega(z) (D - z) dz}$$

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.  
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x    9810*(60-x)*x*(-0.0006*x^3+0.0598*x^2-  
0.1008*x+122.69)  
Lmite inferior = 0  
Lmite superior = 60  
  
Número de paneles = 6
```

*** Regla de Simpson sin corrección ***.

```
0      0.000000      0.000000  
1     10.000000    623239110.000000  
2     20.000000   1097103312.000000  
3     30.000000   1388678094.000000  
4     40.000000   1380761424.000000  
5     50.000000   942495750.000000  
6     60.000000      0.000000  
suma par: 2477864736.000000    suma impar: 2954412954.000000
```

I = 55911270960.000000

*** Regla de Simpson con corrección ***.

```
f(a) = 72215334.000000    f(b) = -119086729.200000
```

```
I = 55915822800.000000  
>>
```

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x      9810*(60-x)*x*(-0.0006*x^3+0.0598*x^2-
0.1008*x+122.69)
```

```
Límite inferior = 0
Límite superior = 60
```

```
Número de paneles = 6
```

```
*** Regla trapezoidal sin corrección ***.
```

i	xi	fi
0	0.000000	0.000000
1	10.000000	623239110.000000
2	20.000000	1097103312.000000
3	30.000000	1388678094.000000
4	40.000000	1380761424.000000
5	50.000000	942495750.000000
6	60.000000	0.000000

```
I = 54322776900.000000
```

```
*** Regla trapezoidal con corrección ***.
```

```
f(li)= 72215334.000000
f(ls)= -119086729.200000
```

```
I = 55916960760.000000
>>
```

$$d = \frac{\int_0^D \rho g z \omega(z) (D-z) dz}{\int_0^D \rho g \omega(z) (D-z) dz} = \frac{55916960760}{2535465132} = 22.05 \text{ m.}$$

9.49². Suponer que la corriente a través de un resistor se describe por la ecuación

$$i(t) = (60-t)^2 + (60-t) \operatorname{sen}(\sqrt{t})$$

y la resistencia es una función de la corriente,

$$R = 10i + 2i^{2/3}$$

Calcular el voltaje promedio entre t = 0 a 60 usando integración numérica.

Solución.

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.  
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x      (60-x)^2+(60-x)*sin(x^0.5)  
Límite inferior = 0  
Límite superior = 60
```

```
Número de paneles = 10
```

```
*** Regla de Simpson sin corrección ***.
```

0	0.000000	3600.000000
1	6.000000	2950.460512
2	12.000000	2288.786536
3	18.000000	1726.549345
4	24.000000	1260.624872
5	30.000000	878.355135
6	36.000000	569.294028
7	42.000000	327.532911
8	48.000000	151.214553
9	54.000000	41.249559
10	60.000000	0.000000

```
suma par: 4269.919990    suma impar: 5924.147462
```

I = 71672.859658

```
*** Regla de Simpson con corrección ***.
```

```
f(a) =           Inf   f(b) = -0.994172  
I =           Inf  
>>
```

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.  
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x      (60-x)^2+(60-x)*sin(x^0.5)  
Límite inferior = 0  
Límite superior = 60
```

```
Número de paneles = 10
```

```
*** Regla trapezoidal sin corrección ***.
```

i	xi	fi
0	0.000000	3600.000000
1	6.000000	2950.460512
2	12.000000	2288.786536
3	18.000000	1726.549345
4	24.000000	1260.624872
5	30.000000	878.355135
6	36.000000	569.294028
7	42.000000	327.532911
8	48.000000	151.214553
9	54.000000	41.249559
10	60.000000	0.000000

I = 71964.404712

*** Regla trapezoidal con corrección ***.

```
f(li)=      Inf
f(ls)= -0.994172

I =      Inf
>>
```

Considerando el resultado de la regla de Simpson,

$$V = i R = 71673 [10(71673) + 2(71673)^{2/3}] = 51.76 \times 10^9 \text{ v}$$

9.50². Muchos problemas de ingeniería implican el cálculo del trabajo. La fórmula general es

$$\text{Trabajo} = \text{fuerza} \times \text{distancia}$$

En aplicaciones simples las fuerzas permanecen constantes durante el desplazamiento en x .

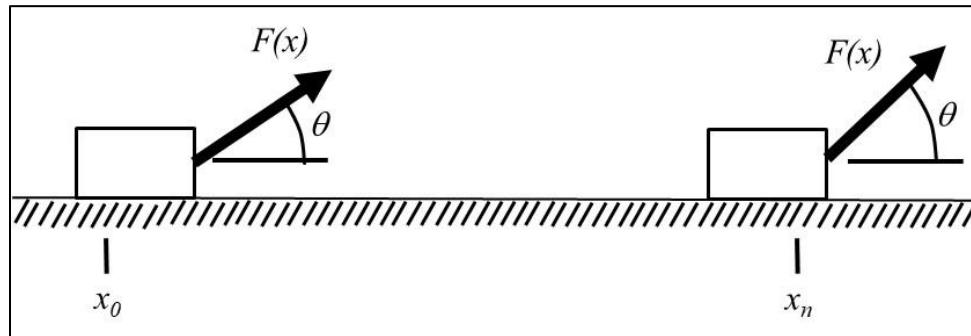
Aunque este cálculo simple es útil para introducir en el concepto, las condiciones de los problemas reales suelen ser más complejas. Por ejemplo, suponer que la fuerza varía durante el transcurso del cálculo. En tales casos, la ecuación de trabajo se vuelve a expresar como

$$W = \int_{x_0}^{x_n} f(x) dx$$

donde W = trabajo [$F \bullet L$], x_0 y x_n = posiciones inicial y final, respectivamente, y $F(x)$ una fuerza que varía en función de la posición. Sin embargo, en el contexto de un problema realista, la fuerza

podría no expresarse de esa manera. De hecho, al analizar los datos medidos, la fuerza podría estar disponible solo en forma tabular. Para tales casos, la integración numérica es la única opción viable para la evaluación.

Además, la complejidad se incrementa si el ángulo entre la fuerza y la dirección del movimiento también varía en función de la posición.



La ecuación del trabajo puede modificarse para considerar este efecto,

$$W = \int_{x_0}^{x_n} F(x) \cos[\theta(x)] dx$$

Si $F(x)$ y $\theta(x)$ son funciones simples, la ecuación se puede resolver analíticamente. Sin embargo, si la relación funcional es complicada, los métodos numéricos pueden ser aplicados como una alternativa muy viable.

Suponer que $F(x)$ y $\theta(x)$ varían de la siguiente forma,

$$F(x) = 1.6x - 0.045x^2$$

$$\theta(x) = 0.8 + 0.125x - 0.009x^2 + 0.0002x^3$$

donde la fuerza está dada en libras y el cuerpo se desplaza 30 pies.

Determinar el trabajo aplicando las reglas trapezoidal y de Simpson con 4, 8 y 16 paneles.

Solución.

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x    (1.6*x-0.045*x^2)*(0.8+0.125*x-
0.009*x^2+0.0002*x^3)
Límite inferior = 0
Límite superior = 30
```

Número de paneles = 4

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.000000
1	7.500000	12.457324
2	15.000000	18.384375
3	22.500000	17.638770
4	30.000000	13.875000

I = 415.634766

*** Regla trapezoidal con corrección ***.

$$\begin{aligned}f(l_i) &= 1.280000 \\f(l_s) &= -1.097500\end{aligned}$$

I = 426.779297

>>

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.

*** Regla trapezoidal ***.

Introduce una función con respecto a x $(1.6*x-0.045*x^2)*(0.8+0.125*x-0.009*x^2+0.0002*x^3)$
Límite inferior = 0
Límite superior = 30

Número de paneles = 8

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.000000
1	3.750000	6.186942
2	7.500000	12.457324
3	11.250000	16.635361
4	15.000000	18.384375
5	18.750000	18.405899
6	22.500000	17.638770
7	26.250000	16.458224
8	30.000000	13.875000

I = 424.141479

*** Regla trapezoidal con corrección ***.

$$\begin{aligned}f(l_i) &= 1.280000 \\f(l_s) &= -1.097500\end{aligned}$$

I = 426.927612

>>

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.

Introduce una función con respecto a x $(1.6*x - 0.045*x^2) * (0.8 + 0.125*x - 0.009*x^2 + 0.0002*x^3)$
Límite inferior = 0
Límite superior = 30

Número de paneles = 16

*** Regla trapezoidal sin corrección ***.

i	xi	fi
0	0.000000	0.000000
1	1.875000	2.853314
2	3.750000	6.186942
3	5.625000	9.500179
4	7.500000	12.457324
5	9.375000	14.862647
6	11.250000	16.635361
7	13.125000	17.784597
8	15.000000	18.384375
9	16.875000	18.548572
10	18.750000	18.405899
11	20.625000	18.074869
12	22.500000	17.638770
13	24.375000	17.120637
14	26.250000	16.458224
15	28.125000	15.478977
16	30.000000	13.875000

I = 426.240349

*** Regla trapezoidal con corrección ***.

f(li)= 1.280000
f(ls)= -1.097500

I = 426.936882

>>

Como se puede apreciar, entre más paneles se considerar, los resultados de las reglas trapezoidales sin y con corrección se aproximan entre ellas.

Por otro lado, aplicando la regla de Simpson,

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.

Introduce una función con respecto a x $(1.6*x - 0.045*x^2) * (0.8 + 0.125*x - 0.009*x^2 + 0.0002*x^3)$
Límite inferior = 0
Límite superior = 30

Número de paneles = 4

*** Regla de Simpson sin corrección ***.

0	0.000000	0.000000
1	7.500000	12.457324
2	15.000000	18.384375
3	22.500000	17.638770
4	30.000000	13.875000

suma par: 18.384375 suma impar: 30.096094

I = 427.570313

*** Regla de Simpson con corrección ***.

f(a) = 1.280000 f(b) = -1.097500

I = 426.937500

>>

Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.

Introduce una función con respecto a x $(1.6*x - 0.045*x^2) * (0.8 + 0.125*x - 0.009*x^2 + 0.0002*x^3)$
Límite inferior = 0
Límite superior = 30

Número de paneles = 8

*** Regla de Simpson sin corrección ***.

0	0.000000	0.000000
1	3.750000	6.186942
2	7.500000	12.457324
3	11.250000	16.635361
4	15.000000	18.384375
5	18.750000	18.405899

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

```
6   22.500000  17.638770
7   26.250000  16.458224
8   30.000000  13.875000
suma par: 48.480469  suma impar: 57.686426
```

I = 426.977051

*** Regla de Simpson con corrección ***.

```
f(a) = 1.280000  f(b) = -1.097500
```

I = 426.937500

>>

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla de Simpson ***.
```

```
Introduce una función con respecto a x  (1.6*x-0.045*x^2)*(0.8+0.125*x-
0.009*x^2+0.0002*x^3)
Límite inferior = 0
Límite superior = 30
```

Número de paneles = 16

*** Regla de Simpson sin corrección ***.

```
0   0.000000  0.000000
1   1.875000  2.853314
2   3.750000  6.186942
3   5.625000  9.500179
4   7.500000  12.457324
5   9.375000  14.862647
6   11.250000 16.635361
7   13.125000 17.784597
8   15.000000 18.384375
9   16.875000 18.548572
10  18.750000 18.405899
11  20.625000 18.074869
12  22.500000 17.638770
13  24.375000 17.120637
14  26.250000 16.458224
15  28.125000 15.478977
16  30.000000 13.875000
suma par: 106.166895  suma impar: 114.223792
```

I = 426.939972

*** Regla de Simpson con corrección ***.

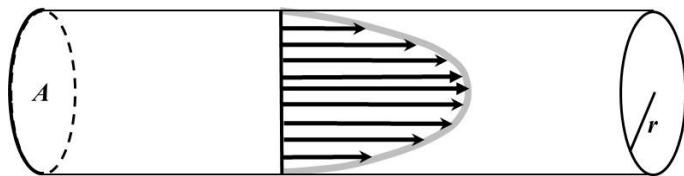
```
f(a) = 1.280000   f(b) = -1.097500
```

```
I = 426.937500
>>
```

Aplicando Simpson sin y con corrección, al considerar 16 paneles, la solución es muy similar.

9.51². Si la distribución de la velocidad de un fluido fluyendo a través de un tubo se asemeja a la mostrada en la figura, la tasa de flujo Q puede calcularse por $Q = \int_0^r v(2\pi r)dr$, donde la v es la velocidad y A es el área de la sección transversal del tubo. Para un tubo circular, $A = \pi r^2$ y $dA = 2\pi r dr$. Entonces,

$$Q = \int_0^r v(2\pi r)dr$$



donde r es la distancia radial hacia afuera desde el centro del tubo. Si la distribución de la velocidad está dada por

$$v = 2 \left(1 - \frac{r}{r_0}\right)^{1/6}$$

donde r_0 es el radio total, 3 cm en este caso, calcular Q usando la regla trapezoidal.

Solución.

```
Symbolic pkg v2.7.0: Python communication link active, SymPy v1.2.
*** Regla trapezoidal ***.
```

```
Introduce una función con respecto a x    4*pi*x*(1-x/0.03)^(1/6)
Límite inferior = 0
Límite superior = 0.03
```

```
Número de paneles = 10
```

```
*** Regla trapezoidal sin corrección ***.
```

i	xi	fi
0	0.000000	0.000000
1	0.003000	0.037043
2	0.006000	0.072646

3	0.009000	0.106570
4	0.012000	0.138489
5	0.015000	0.167930
6	0.018000	0.194160
7	0.021000	0.215915
8	0.024000	0.230635
9	0.027000	0.231157
10	0.030000	0.000000

I = 0.004184

*** Regla trapezoidal con corrección ***.

f(li)= 12.566371
f(ls)= -Inf

I = Inf
>>

Entonces, $Q = 2 * 0.004184 = 0.008368 \text{ m}^3/\text{s.}$

Ecuaciones diferenciales.

9.52⁶. En un tanque perfectamente agitado se tienen 400 l de una salmuera en la cual están disueltos 25 kg de sal común (NaCl), en cierto momento se hace llegar al tanque un gasto de 80 l/min de una salmuera que contiene 0.5 kg de sal común por litro. Si se tiene un gasto de salida de 80 l/min determinar:

- a) ¿Qué cantidad de sal hay en el tanque, transcurridos 10 minutos?
- b) ¿Qué cantidad de sal hay en el tanque, transcurrido un tiempo muy grande?

Solución.

a)

$$\frac{dx}{dt} = \text{masa de sal que entra-masa de sal que sale}$$

x = cantidad de sal en el tanque después de t minutos, kg.

Sustituyendo los datos,

$$\frac{dx}{dt} = 80(0.5) - 80\left(\frac{x}{400}\right) = 40 - 0.2x, \quad x(0) = 25$$

Resolviendo por rk4 en código Octave,

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x 40-0.2*y

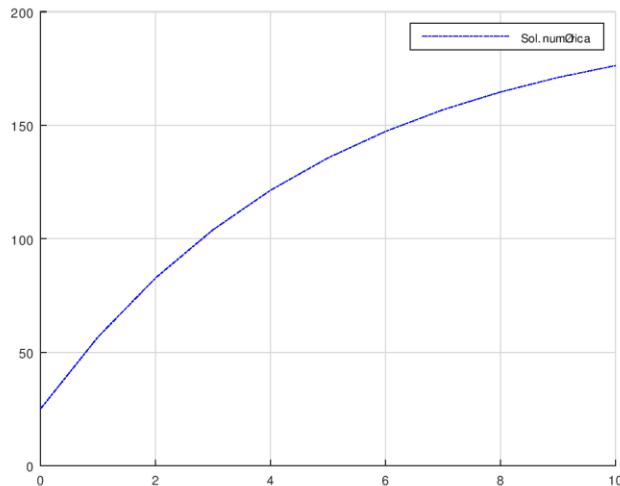
Introduce la solución analítica (Oprime Enter si no existe)

Datos:

```
x0 = 0
xn = 10
y0 = 25
Incremento de x = 1
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.00000000	25.00000000	35.00000000	31.50000000	31.85000000	28.63000000
1	1.00000000	56.72166667	28.65566667	25.79010000	26.07665667	23.44033533
2	2.00000000	82.69325256	23.46134949	21.11521454	21.34982803	19.19138388
3	3.00000000	103.95705564	19.20858887	17.28772998	17.47981587	15.71262570
4	4.00000000	121.36644002	15.72671200	14.15404080	14.31130792	12.86445041
5	5.00000000	135.62008333	12.87598333	11.58838500	11.71714483	10.53255437
6	6.00000000	147.29001622	10.54199676	9.48779708	9.59321705	8.62335335
7	7.00000000	156.84457928	8.63108414	7.76797573	7.85428657	7.06022683
8	8.00000000	164.66721854	7.06655629	6.35990066	6.43056622	5.78044305
9	9.00000000	171.07187406	5.78562519	5.20706267	5.26491892	4.73264140
10	10.00000000	176.31557902				

>>



Entonces, $x(10) = 176.32 \text{ kg}$

- b) Considerando que se estabiliza al no variar con el tiempo, $\frac{dx}{dt} = 0$

$$\frac{dx}{dt} = 40 - 0.2x = 0$$

Por lo tanto, $x = 200 \text{ kg}$ después de mucho tiempo.

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x $40 - 0.2 * y$

Introduce la solución analítica (Oprime Enter si no existe)

Datos:

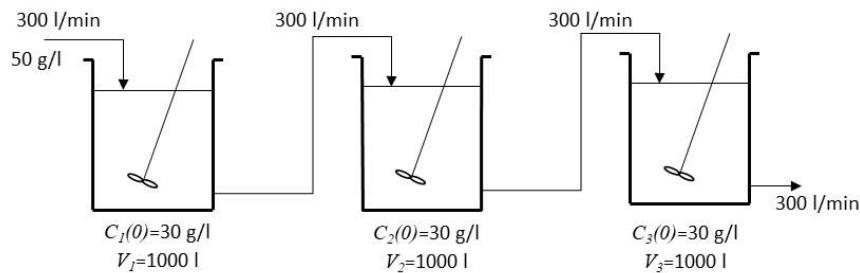
```
x0 = 0
xn = 50
y0 = 25
Incremento de x = 1
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.00000000	25.00000000	35.00000000	31.50000000	31.85000000	28.63000000
1	1.00000000	56.72166667	28.65566667	25.79010000	26.07665667	23.44033533
2	2.00000000	82.69325256	23.46134949	21.11521454	21.34982803	19.19138388
3	3.00000000	103.95705564	19.20858887	17.28772998	17.47981587	15.71262570
4	4.00000000	121.36644002	15.72671200	14.15404080	14.31130792	12.86445041
5	5.00000000	135.62008333	12.87598333	11.58838500	11.71714483	10.53255437
. . .						
45	45.00000000	199.97840022	0.00431996	0.00388796	0.00393116	0.00353372
46	46.00000000	199.98231554	0.00353689	0.00318320	0.00321857	0.00289318
47	47.00000000	199.98552114	0.00289577	0.00260619	0.00263515	0.00236874
48	48.00000000	199.98814568	0.00237086	0.00213378	0.00215749	0.00193937
49	49.00000000	199.99029447	0.00194111	0.00174700	0.00176641	0.00158782
50	50.00000000	199.99205376				

>>

9.53⁶. Se tienen tres tanques de 1000 litros de capacidad cada uno, perfectamente agitados. Los tres recipientes están completamente llenos con una solución cuya concentración es de 30 g/l. A partir de cierto momento se alimenta al primer tanque con una solución que contiene 50 g/l con un gasto de 300 l/min (el cual fluye igual al resto de los tanques y hacia afuera del sistema).

Calcular al concentración en cada tanque después de 10 minutos de haber empezado a agregar la solución al primero.



Solución.

Balance para el primer tanque:

$$\text{Variación del soluto} = \text{Entradas-Salidas}$$

$$V_1 \frac{dC_1}{dt} = 300(50) - 300C_1$$

$$\frac{dC_1}{dt} = 15 - 0.3C_1 \quad y \quad C_1(0) = 30$$

Balance para el segundo tanque:

$$V_2 \frac{dC_2}{dt} = (1000) \frac{dC_2}{dt} = 300C_1 - 300C_2$$

$$\frac{dC_2}{dt} = 0.3(C_1 - C_2) \quad y \quad C_2(0) = 30$$

Balance para el tercer tanque:

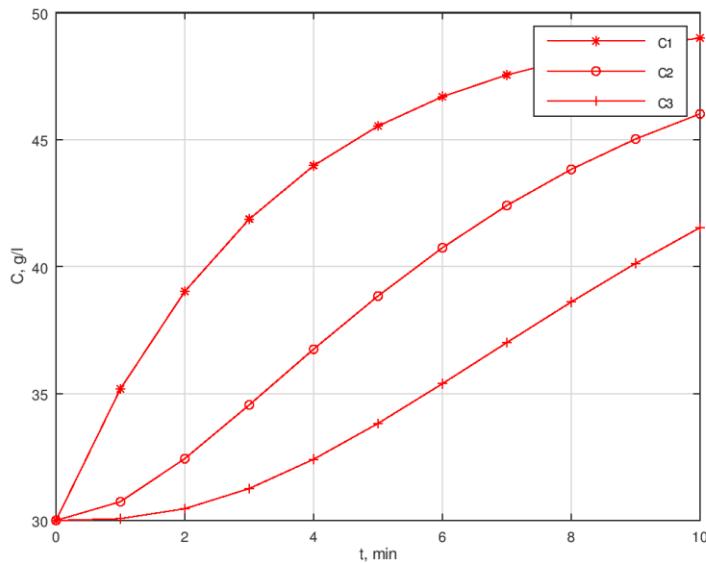
$$V_3 \frac{dC_3}{dt} = (1000) \frac{dC_3}{dt} = 300C_2 - 300C_3$$

$$\frac{dC_3}{dt} = 0.3(C_2 - C_3) \quad y \quad C_3(0) = 30$$

Por lo tanto, se tienen tres ecuaciones diferenciales ordinarias con sus condiciones iniciales.

Resolviendo con código en Octave,

```
>> function fun=fcn(c,t)
fun(1)=15-0.3*c(1);
fun(2)=0.3*(c(1)-c(2));
fun(3)=0.3*(c(2)-c(3));
endfunction
>> y0=[30,30,30];
>> t=[0:10];
>> y=lsode("fcn",y0,t);
>> for i=1:length(t)
fprintf('%4.1f %8.4f %8.4f %8.4f',t(i),y(i,1),y(i,2),y(i,3));
fprintf('\n')
endfor
0.0 30.0000 30.0000 30.0000
1.0 35.1836 30.7387 30.0720
2.0 39.0238 32.4380 30.4623
3.0 41.8686 34.5504 31.2571
4.0 43.9761 36.7475 32.4103
5.0 45.5374 38.8435 33.8231
6.0 46.6940 40.7433 35.3876
7.0 47.5509 42.4077 37.0074
8.0 48.1856 43.8312 38.6058
9.0 48.6559 45.0268 40.1275
10.0 49.0043 46.0170 41.5362
>> plot(t,y(:,1),'r-*',t,y(:,2),'r-o',t,y(:,3),'r-+')
>> xlabel('t, min')
>> ylabel('C, g/l')
>> legend('C1','C2','C3')
>> grid
>>
```



9.54⁶. Un tanque perfectamente agitado contiene 400 litros de salmuera en la cual están disueltos 10 kg de sal. Si se hace llegar 1.0 l/min de una salmuera que contiene 3 kg de sal en cada 5 l y por el fondo se extraen 8 l/min de una salmuera, determinar la concentración de sal en el tanque en distintos tiempos.

Solución.

Elaborando un balance,

$$\frac{d(Vc)}{dt} = 1(600) - 8c$$

$$\frac{d(Vc)}{dt} = V \frac{dc}{dt} + c \frac{dV}{dt} = 600 - 8c$$

Debido a que entran 1 l/min y salen 8 l/min, la ecuación de la variación del volumen en el tanque es: $V = 400 - 7t$. Entonces,

$$\frac{d(Vc)}{dt} = (400 - 7t) \frac{dc}{dt} - 7c = 600 - 8c$$

$$\frac{dc}{dt} - 7c = \frac{600 - c}{400 - 7t}$$

Resolviendo con código en Octave,

Método de Euler.

Introduce una función con respecto a x (600-y)/(400-7*x)

Introduce la solución analítica (Oprime Enter si no existe)

Datos:

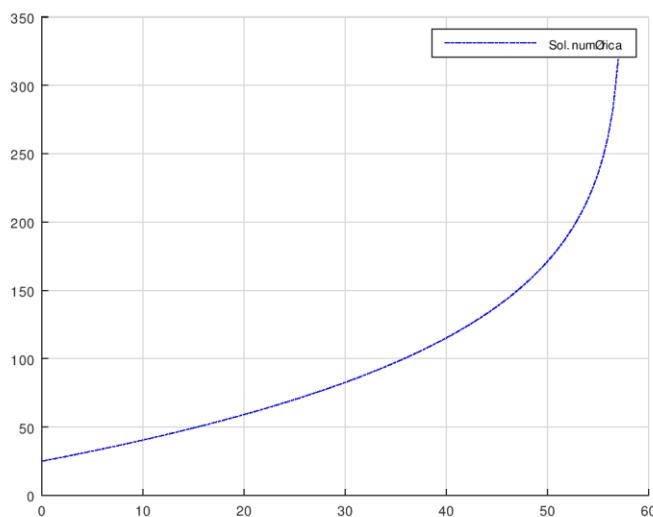
```
x0 = 0
xn = 57
y0 = 25
Incremento de x = 0.5
```

i	x(i)	y(i)
0	0.000000	25.000000
1	0.500000	25.718750
2	1.000000	26.442938

```

3   1.500000  27.172655
4   2.000000  27.907991
5   2.500000  28.649043
.
.
109 54.500000 225.328078
110 55.000000 235.454346
111 55.500000 247.605868
112 56.000000 262.927352
113 56.500000 283.994392
114 57.000000 319.106126
>>

```



Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x $(600-y)/(400-7*x)$

Introduce la solución analítica (Oprime Enter si no existe)

Datos:

```

x0 = 0
xn = 57
y0 = 25
Incremento de x = 0.5

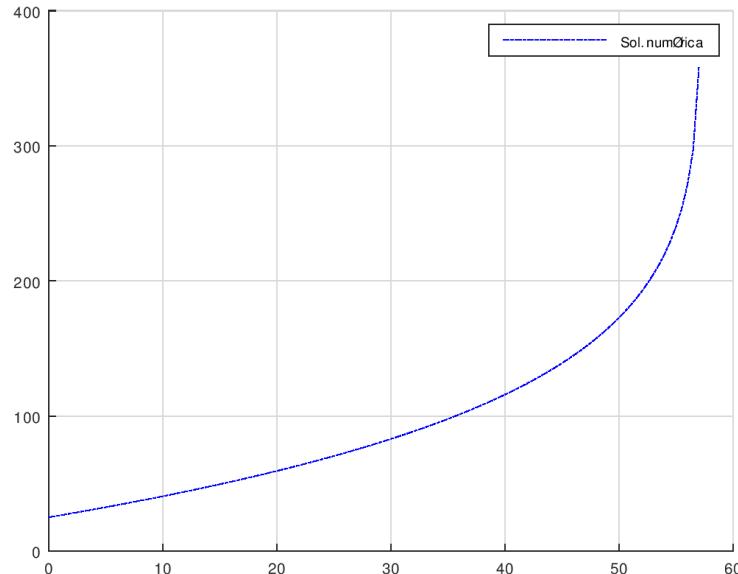
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.00000000	25.00000000	1.43750000	1.44291431	1.44291091	1.44836960
1	0.50000000	25.72146000	1.44836958	1.45387320	1.45386972	1.45941884
2	1.00000000	26.44839953	1.45941883	1.46501405	1.46501047	1.47065236
3	1.50000000	27.18090955	1.47065235	1.47634153	1.47633786	1.48207493
4	2.00000000	27.91908338	1.48207491	1.48786050	1.48785674	1.49369147
5	2.50000000	28.66301679	1.49369146	1.49957600	1.49957213	1.50550712
.
109	54.50000000	229.34543113	20.03538210	21.82959542	21.80281612	23.98354405

```

110 55.00000000 240.28574357 23.98095043 26.69577501 26.64455190 30.12104178
111 55.50000000 253.68429740 30.11440892 34.74739491 34.62860040 41.12517530
112 56.00000000 271.18359531 41.10205059 50.96654273 50.57196304 67.45120515
113 56.50000000 297.15278425 67.29938128 104.00813470 100.67097530 252.51172810
114 57.00000000 357.91689503
>>

```



9.55⁶. Considerar un sistema ecológico compuesto solamente de coyotes (y) y de correcaminos (x), donde los primeros se alimentan de los segundos. Los tamaños de las poblaciones cambian de acuerdo con las ecuaciones

$$\frac{dx}{dt} = k_1x - k_2xy$$

$$\frac{dy}{dt} = k_3xy - k_4y$$

Si no hay coyotes (y) los correcaminos se reproducen con una velocidad de crecimiento k_1x ; si no hay correcaminos, la especie de coyotes desaparece con velocidad k_4y . El término xy representa la interacción de las dos especies y las constantes k_2 y k_3 dependen de la habilidad de los depredadores para atrapar a los correcaminos y de la habilidad de estos para huir.

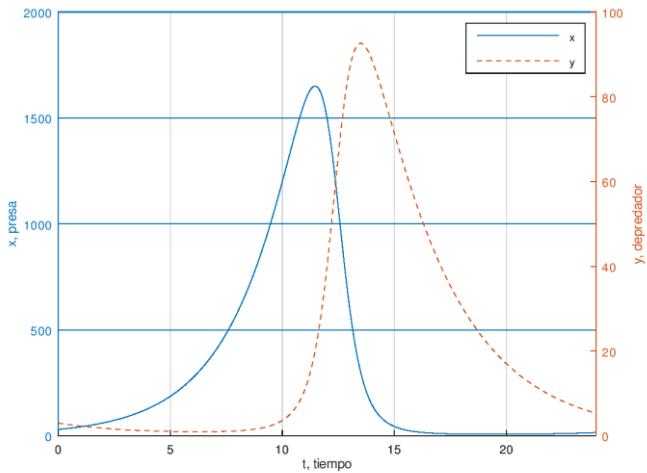
Las poblaciones de los coyotes cambian cíclicamente. Calcular el ciclo al resolver el modelo con $k_1 = 0.4$, $k_2 = 0.02$, $k_3 = 0.001$ y $k_4 = 0.3$. Usar $x(0) = 30$ y $y(0) = 3$ como condiciones iniciales.

Solución.

Resolviendo con comandos de Octave,

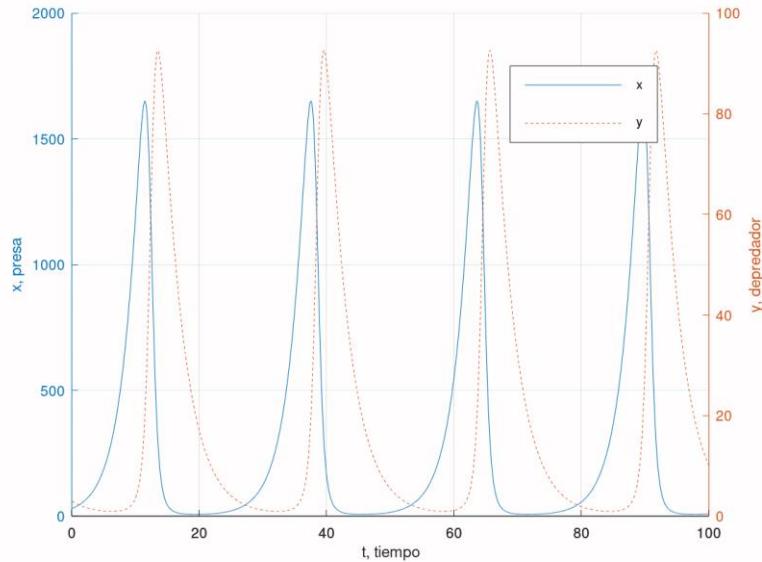
```
>> function fun=fcn(y,t)
```

```
k1=0.4; k2=0.02; k3=0.001; k4=0.3;
fun(1)=k1*y(1)-k2*y(1)*y(2);
fun(2)=k3*y(1)*y(2)-k4*y(2);
endfunction
>> t=[0:24];
>> y0=[30,3];
>> y=lsode("fcn",y0,t);
>> for i=1:length(t)
fprintf('%4.0f %8.0f %8.0f',t(i),y(i,1),y(i,2));
fprintf('\n')
endfor
    0      30      3
    1      42      2
    2      61      2
    3      88      1
    4     128      1
    5     186      1
    6     273      1
    7     399      1
    8     583      1
    9     845      2
   10    1199      4
   11    1575     11
   12    1497     40
   13    605      87
   14    146      89
   15    44       71
   16    19       54
   17    11       41
   18     8       31
   19     7       23
   20     7       17
   21     8       13
   22     9       9
   23    12       7
   24    16       5
>>
>> t=[0:0.1:24];
>> y0=[30,3];
>> y=lsode("fcn",y0,t);
>> clf;
>> hold on
>> [hax, h1, h2] = plotyy (t, y(:,1), t, y(:,2));
>> set ([h2], "linestyle", "--");
>> xlabel (hax(1), "t, tiempo");
>> ylabel (hax(1), "x, presa")
>> ylabel (hax(2), "y, depredador")
>> grid
>> legend("x", "y")
>>
```

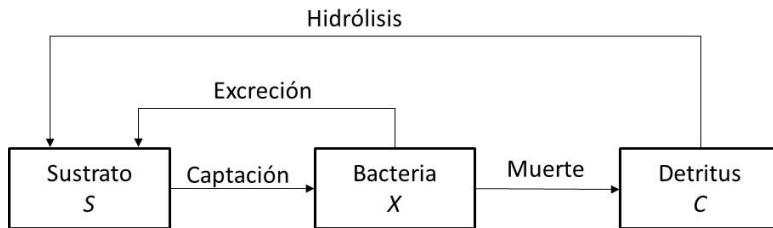


Y para varios ciclos,

```
>> t=[0:0.1:100];
>> y=lsode ("fcn",y0,t);
>> clf;
>> hold on
>> [hax, h1, h2] = plotyy (t, y(:,1), t, y(:,2));
>> set ([h2], "linestyle", "--");
>> xlabel (hax(1), "t, tiempo");
>> ylabel (hax(1), "x, presa")
>> ylabel (hax(2), "y, depredador")
>> grid
>>
```



9.56². Las bacterias que crecen en un reactor utilizan una fuente de alimento soluble (sustrato) como se representa en la figura. La absorción del sustrato está representada por un modelo logístico con limitación de Michaelis-Menten. La muerte de la bacteria produce detritus que posteriormente se convierte en el sustrato por hidrólisis. Además, las bacterias también excretan algún sustrato directamente. La muerte, la hidrólisis y la excreción se simulan como reacciones de primer orden.



donde X , C y S = las concentraciones (mg / l) de bacterias, detritus y sustrato, respectivamente; μ_{max} = tasa máxima de crecimiento (/ d), K = la capacidad de carga logística (mg / l); K_s = la constante de semisaturación de Michaelis-Menten (mg / l), tasa de muerte k_d = (/ d); tasa de excreción k_e = (/ d); y tasa de hidrólisis de k_h = (/ d). Simula las concentraciones de $t = 0$ a 100 d, dadas las condiciones iniciales $X(0) = 1$ mg / l, $S(0) = 100$ mg / l y $C(0) = 0$ mg / l. Emplee los siguientes parámetros en su cálculo: $\mu_{max} = 10$ / d, $K = 10$ mg / l, $K_s = 10$ mg / l, $k_d = 0.1$ / d, $k_e = 0.1$ / d, y $k_h = 0.1$ / d.

Solución.

Resolviendo comandos de Octave,

```

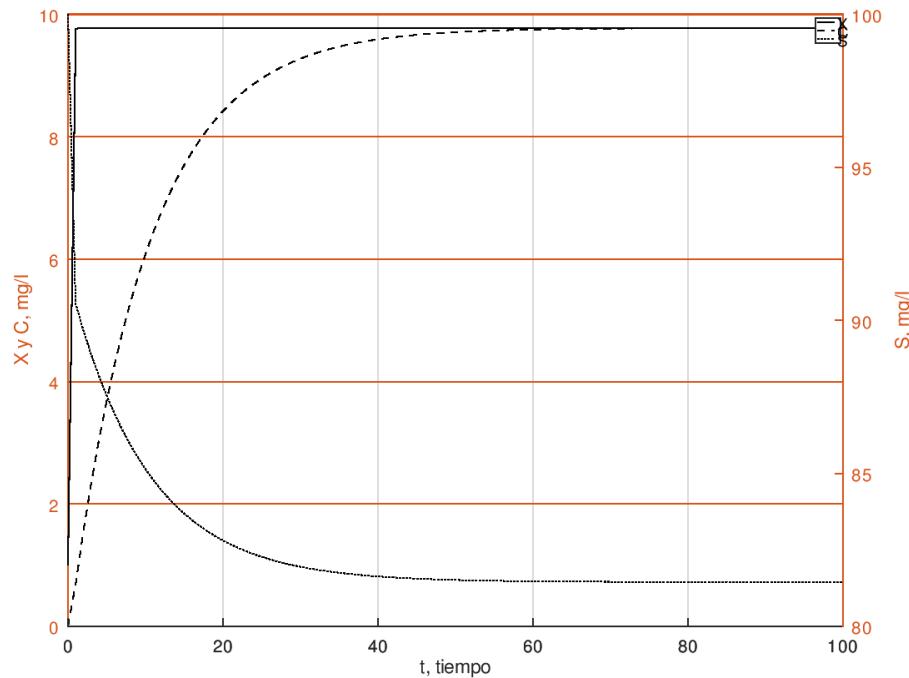
>> function fun=fcn(y,t)
Umax=10; K=10; Ks=10; kd=0.1; ke=0.1; kh=0.1;
fun(1) = Umax*(1-y(1)/K)*(y(3)/(Ks+y(3)))*y(1) - kd*y(1) - ke*y(1);
fun(2) = kd*y(1) - kh*y(2);
fun(3) = ke*y(1) + kh*y(2) - Umax*(1-y(1)/K)*(y(3)/(Ks+y(3)))*y(1);
endfunction
>> t=[0:100];
>> y0=[1,0,100];
>> y=lsode("fcn",y0,t);
>> for i=1:length(t)
fprintf('%4.0f %8.4f %8.4f %8.4f',t(i),y(i,1),y(i,2),y(i,3));
fprintf('\n')
endfor
0 1.0000 0.0000 100.0000
1 9.7654 0.6987 90.5359
2 9.7777 1.5625 89.6598
3 9.7775 2.3443 88.8782
4 9.7773 3.0517 88.1710
5 9.7772 3.6917 87.5312
.
.
.

```

```

95  9.7754  9.7747  81.4499
96  9.7754  9.7748  81.4498
97  9.7754  9.7748  81.4497
98  9.7754  9.7749  81.4497
99  9.7754  9.7749  81.4496
100 9.7754  9.7750  81.4496

>>
>> clf;
>> hold on
>> plot(t,y(:,1),"color","k")
>> [hax, h1, h2] = plotyy (t, y(:,2), t, y(:,3));
>> grid
>> set ([h1], "linestyle", "--");
>> set ([h1], "color", "k");
>> set ([h2], "color", "k");
>> set ([h2], "linestyle", ":" );
>> legend('X', 'C', 'S')
>> xlabel ("t, tiempo");
>> ylabel("X, mg/l")
>> ylabel (hax(1), "X y C, mg/l")
>> ylabel (hax(2), "S, mg/l")
>>
    
```

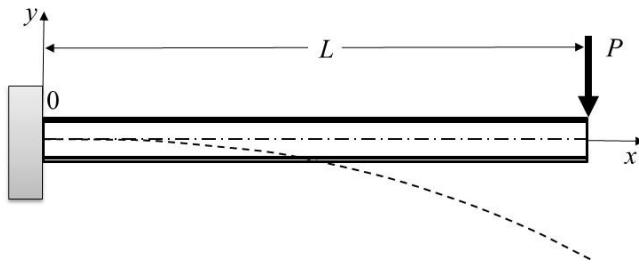


9.56². La ecuación diferencial básica de la curva elástica para una viga cantilever (figura) se da como

$$EI \frac{d^2y}{dx^2} = -P(L - x)$$

donde E = módulo de elasticidad e I = momento de inercia. Resolver para la deflexión de la viga usando métodos numéricos. Aplicar los siguientes valores de los parámetros: $E = 30,000$ ksi, $I = 800$ pul 2 , $P = 1$ kip, $L = 1$ pie. Comparar los resultados obtenidos con al solución analítica,

$$y = -\frac{PLx^2}{2EI} + \frac{Px^3}{6EI}$$



Solución.

$$EI \frac{d^2y}{dx^2} = -P(L - x)$$

Haciendo, $z = \frac{dy}{dt}$, queda el siguiente sistema,

$$\frac{dy}{dx} = z$$

$$\frac{dz}{dx} = -\frac{P}{EI} (L - x)$$

Comprobando la solución analítica usando comandos de Octave,

```
>> pkg load symbolic
>> syms i(t) P E I L
>> DE = diff(y, 2) == -(P/(E*I))*(L-x)
DE = (sym)


$$\frac{d^2y(x)}{dx^2} = \frac{-P*(L - x)}{E*I}$$


>> dsolve(DE, y(0) == 0, diff(y)(0) == 0)
ans = (sym)
```

$$y(x) = -\frac{L^2 P^2 x^2}{2 \cdot E^2 I} + \frac{P^3 x^3}{6 \cdot E^3 I}$$

>>

Ahora, resolviendo el problema con comandos de Octave, $y(1) = y$, $y(2) = z$,

```
>> function fun=fcn(y,t)
E=3e8; I=38.58e-3; P=1000; L=10;
fun(1) = y(2);
fun(2) = -P*(10-t)/(E*I);
endfunction
>> t=[0:0.5:10];
>> y0=[0, 0];
>> y=lsode("fcn",y0,t);
>> yreal=-432e-6*t.^2+14.4e-6*t.^3;
>> for i=1:length(t)
fprintf('%4.1f %8.6f %8.6f %8.6f',t(i),y(i,1),y(i,2),yreal(i));
fprintf('\n')
endfor
0.0 0.000000 0.000000 0.000000
0.5 -0.000106 -0.000421 -0.000106
1.0 -0.000418 -0.000821 -0.000418
1.5 -0.000923 -0.001199 -0.000923
2.0 -0.001613 -0.001555 -0.001613
2.5 -0.002475 -0.001890 -0.002475
3.0 -0.003499 -0.002203 -0.003499
3.5 -0.004675 -0.002495 -0.004675
4.0 -0.005990 -0.002765 -0.005990
4.5 -0.007436 -0.003013 -0.007436
5.0 -0.009000 -0.003240 -0.009000
5.5 -0.010672 -0.003445 -0.010672
6.0 -0.012442 -0.003629 -0.012442
6.5 -0.014297 -0.003791 -0.014297
7.0 -0.016229 -0.003931 -0.016229
7.5 -0.018225 -0.004050 -0.018225
8.0 -0.020275 -0.004147 -0.020275
8.5 -0.022369 -0.004223 -0.022369
9.0 -0.024495 -0.004277 -0.024494
9.5 -0.026642 -0.004309 -0.026642
10.0 -0.028800 -0.004320 -0.028800
>>
```

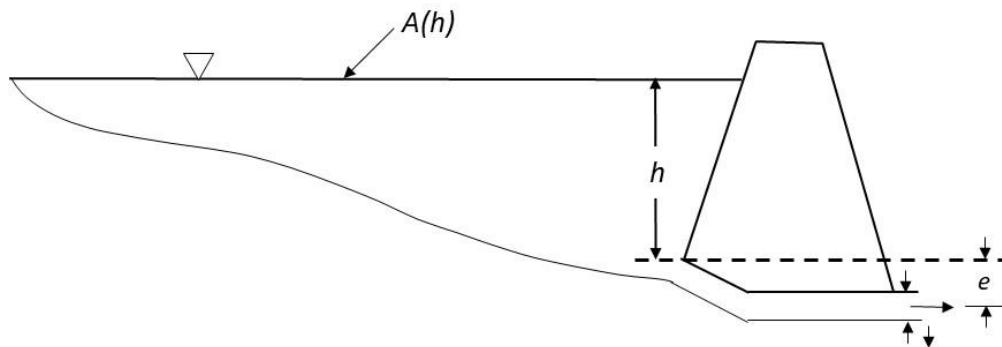
Los valores numéricos y los analíticos mostrados, respectivamente, en las columnas 2 y 4, confirman la similitud entre ambos resultados.

9.57². Un estanque drena a través de una tubería, como se muestra en la figura. Bajo una serie de supuestos que simplifican el problema, la siguiente ecuación diferencial describe cómo cambia la profundidad con el tiempo:

$$\frac{dh}{dt} = - \frac{\pi d^2}{4A(h)} \sqrt{2g(h+e)}$$

donde h = profundidad (m), t = tiempo (s), d = diámetro del tubo (m), $A(h)$ = área de la superficie del estanque en función de la profundidad (m^2), $g = 9.81 \text{ m/s}^2$ y e = tirante del tubo bajo el fondo del estanque (m). Con base en la siguiente tabla tirante-área, resolver esta ecuación diferencial dado que $h(0) = 6 \text{ m}$, $d = 0.25 \text{ m}$, $e = 1 \text{ m}$.

$h, \text{ m}$	6	5	4	3	2	1	0
$A(h), 10^4 \text{ m}^2$	1.17	0.97	0.67	0.45	0.32	0.18	0



Solución.

Usando código de Octave, se ejecuta el programa para $t = 19 \text{ h}$ (68,400 s).

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x -
 $(0.25*\pi*0.25^2/(129*y^2+1171*y+186))*(19.62*(y+1))^{0.5}$

Introduce la solución analítica (Oprime Enter si no existe)

Datos:

```
x0 = 0
xn = 68400
y0 = 6
Incremento de x = 3600
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.000000	6.000000	-0.000049	-0.000049	-0.000049	-0.000050
1	3600.000000	5.822850	-0.000050	-0.000051	-0.000051	-0.000051
2	7200.000000	5.640476	-0.000051	-0.000052	-0.000052	-0.000053
3	10800.000000	5.452415	-0.000053	-0.000054	-0.000054	-0.000055
4	14400.000000	5.258131	-0.000055	-0.000056	-0.000056	-0.000057
5	18000.000000	5.056998	-0.000057	-0.000058	-0.000058	-0.000059
6	21600.000000	4.848278	-0.000059	-0.000060	-0.000060	-0.000062
7	25200.000000	4.631085	-0.000062	-0.000063	-0.000063	-0.000064
8	28800.000000	4.404347	-0.000064	-0.000066	-0.000066	-0.000068

```

9 32400.000000 4.166738 -0.000068 -0.000069 -0.000069 -0.000071
10 36000.000000 3.916591 -0.000071 -0.000074 -0.000074 -0.000076
11 39600.000000 3.651751 -0.000076 -0.000078 -0.000078 -0.000081
12 43200.000000 3.369350 -0.000081 -0.000084 -0.000084 -0.000088
13 46800.000000 3.065419 -0.000088 -0.000092 -0.000092 -0.000097
14 50400.000000 2.734162 -0.000097 -0.000102 -0.000102 -0.000108
15 54000.000000 2.366486 -0.000108 -0.000116 -0.000117 -0.000126
16 57600.000000 1.946445 -0.000126 -0.000139 -0.000140 -0.000159
17 61200.000000 1.440273 -0.000159 -0.000187 -0.000193 -0.000254
18 64800.000000 0.737489 -0.000256 -0.000472 -0.003699 -0.000000
19 68400.000000 -4.422208

```

>>

Como se observa, después de 18 horas, el estanque se vacía completamente.

9.58². Para un circuito simple RL , la ley del voltaje de Kirchhoff requiere que (si se considera la ley de Ohm)

$$L \frac{di}{dt} + Ri = 0$$

donde i = corriente, L = inductancia y R = resistencia. Resolver para i , si $L = 1$, $R = 1.5$ e $i(0) = 0.5$. Resolver analíticamente y numéricamente y comparar su resultados.

Solución.

Con comandos de octave se puede obtener la solución analítica,

```

>> pkg load symbolic
>> syms i(t)
>> DE=diff(i,t)+1.5*i==0;
>> sol=dsolve(DE,i(0)==0.5)
sol = (sym)

```

$$i(t) = \frac{e^{-\frac{3t}{2}}}{2}$$

>>

Entonces, la solución analítica es: $i(t) = 0.5 e^{-1.5 t}$

Y resolviendo numéricamente con código de octave,

Introducción a los métodos numéricos + Octave

Salvador Díaz Maldonado, 2022.

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x -1.5*y

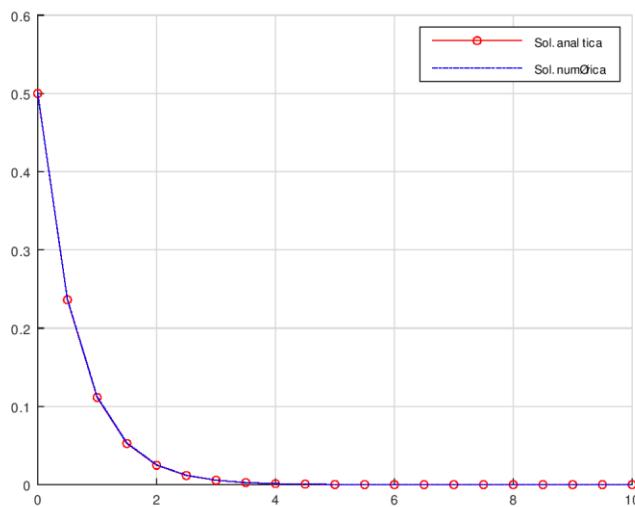
Introduce la solución analítica (Oprime Enter si no existe) 0.5*exp(-1.5*x)

Datos:

```
x0 = 0
xn = 10
y0 = 0.5
Incremento de x = 0.5
```

i	x(i)	y(i)	k1	k2	k3	k4	y(i).verd
0	0.000000	0.500000	-0.750000	-0.468750	-0.574219	-0.319336	0.500000
1	0.500000	0.237061	-0.355591	-0.222244	-0.272249	-0.151404	0.236183
2	1.000000	0.112395	-0.168593	-0.105371	-0.129079	-0.071784	0.111565
3	1.500000	0.053289	-0.079934	-0.049958	-0.061199	-0.034034	0.052700
4	2.000000	0.025265	-0.037898	-0.023686	-0.029016	-0.016136	0.024894
5	2.500000	0.011979	-0.017968	-0.011230	-0.013757	-0.007651	0.011759
6	3.000000	0.005679	-0.008519	-0.005324	-0.006522	-0.003627	0.005554
7	3.500000	0.002693	-0.004039	-0.002524	-0.003092	-0.001720	0.002624
8	4.000000	0.001277	-0.001915	-0.001197	-0.001466	-0.000815	0.001239
9	4.500000	0.000605	-0.000908	-0.000567	-0.000695	-0.000387	0.000585
10	5.000000	0.000287	-0.000430	-0.000269	-0.000330	-0.000183	0.000277
11	5.500000	0.000136	-0.000204	-0.000128	-0.000156	-0.000087	0.000131
12	6.000000	0.000065	-0.000097	-0.000060	-0.000074	-0.000041	0.000062
13	6.500000	0.000031	-0.000046	-0.000029	-0.000035	-0.000020	0.000029
14	7.000000	0.000015	-0.000022	-0.000014	-0.000017	-0.000009	0.000014
15	7.500000	0.000007	-0.000010	-0.000006	-0.000008	-0.000004	0.000007
16	8.000000	0.000003	-0.000005	-0.000003	-0.000004	-0.000002	0.000003
17	8.500000	0.000002	-0.000002	-0.000001	-0.000002	-0.000001	0.000001
18	9.000000	0.000001	-0.000001	-0.000001	-0.000001	-0.000000	0.000001
19	9.500000	0.000000	-0.000001	-0.000000	-0.000000	-0.000000	0.000000
20	10.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

>>



Como se aprecia, los valores numéricos y analítico son muy similares.

9.59². En contraste con el problema anterior 9.58, las resistencias no siempre pueden obedecer la ley de Ohms. Por ejemplo, la caída de voltaje puede ser no lineal y la dinámica del circuito puede representarse por la siguiente ecuación:

$$L \frac{di}{dt} + R \left[\frac{1}{I} - \left(\frac{1}{I^3} \right) \right] = 0$$

donde I es una corriente de referencia conocida e igual a 1 y todos los otros parámetros como se definieron en el problema anterior. Determinar i como función del tiempo bajo las mismas condiciones especificadas en el problema anterior.

Solución.

Resolviendo con código de Octave.

Método de Runge-Kutta de 4to orden.

Introduce una función con respecto a x $-1.5 * (y - y^3)$

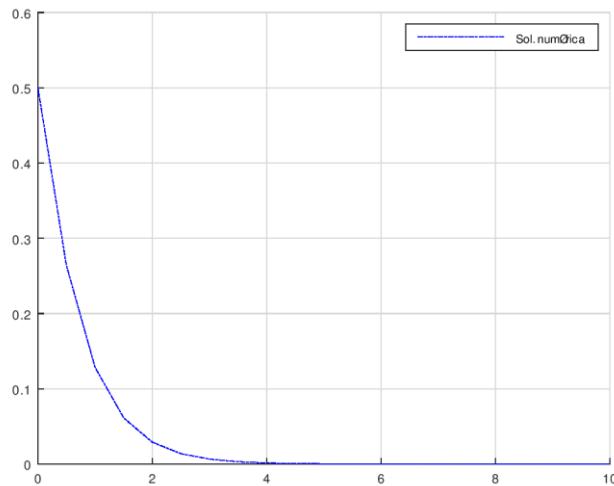
Introduce la solución analítica (Oprime Enter si no existe)

Datos:

```
x0 = 0
xn = 10
y0 = 0.5
Incremento de x = 0.5
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.000000	0.500000	-0.562500	-0.469442	-0.489924	-0.357674
1	0.500000	0.263424	-0.367717	-0.249677	-0.289326	-0.175630
2	1.000000	0.128312	-0.189299	-0.120684	-0.145793	-0.082867
3	1.500000	0.061218	-0.091483	-0.057437	-0.070134	-0.039200
4	2.000000	0.029066	-0.043563	-0.027254	-0.033362	-0.018575
5	2.500000	0.013785	-0.020674	-0.012924	-0.015830	-0.008805
6	3.000000	0.006536	-0.009804	-0.006128	-0.007506	-0.004175
7	3.500000	0.003099	-0.004649	-0.002905	-0.003559	-0.001979
8	4.000000	0.001469	-0.002204	-0.001378	-0.001687	-0.000938
9	4.500000	0.000697	-0.001045	-0.000653	-0.000800	-0.000445
10	5.000000	0.000330	-0.000495	-0.000310	-0.000379	-0.000211
11	5.500000	0.000157	-0.000235	-0.000147	-0.000180	-0.000100
12	6.000000	0.000074	-0.000111	-0.000070	-0.000085	-0.000047
13	6.500000	0.000035	-0.000053	-0.000033	-0.000040	-0.000022
14	7.000000	0.000017	-0.000025	-0.000016	-0.000019	-0.000011
15	7.500000	0.000008	-0.000012	-0.000007	-0.000009	-0.000005
16	8.000000	0.000004	-0.000006	-0.000004	-0.000004	-0.000002
17	8.500000	0.000002	-0.000003	-0.000002	-0.000002	-0.000001
18	9.000000	0.000001	-0.000001	-0.000001	-0.000001	-0.000001
19	9.500000	0.000000	-0.000001	-0.000000	-0.000000	-0.000000
20	10.000000	0.000000	-0.000000	-0.000000	-0.000000	-0.000000

>>



9.60². La tasa de flujo de calor (conducción) entre dos puntos en un cilindro calentado en un extremo está dado por

$$\frac{dQ}{dt} = \lambda A \frac{dT}{dx}$$

donde λ = una constante, A = área de la sección transversal del cilindro, Q = flujo de calor, T = temperatura, t = tiempo y x = distancia desde el extremo caliente. Debido a que la ecuación involucra dos derivadas, se puede simplificar esta ecuación haciendo

$$\frac{dT}{dx} = \frac{100(L-x)(20-t)}{100-xt}$$

donde L es la longitud de la barra. Combinar las dos ecuaciones para calcular el flujo de calor desde $t = 0$ hasta 25 s. Las condiciones iniciales son $Q(0) = 0$ y los parámetros son $\lambda = 0.5$ cal-cm, $A = 12 \text{ cm}^2$, $L = 20 \text{ cm}$ y $x = 2.5 \text{ cm}$. Graficar los resultados.

Solución.

Método de Runge-Kutta de 4to orden.

```
Introduce una función con respecto a x  0.5*12*100*(20-2.5)*(20-x)/(100-2.5*x)
```

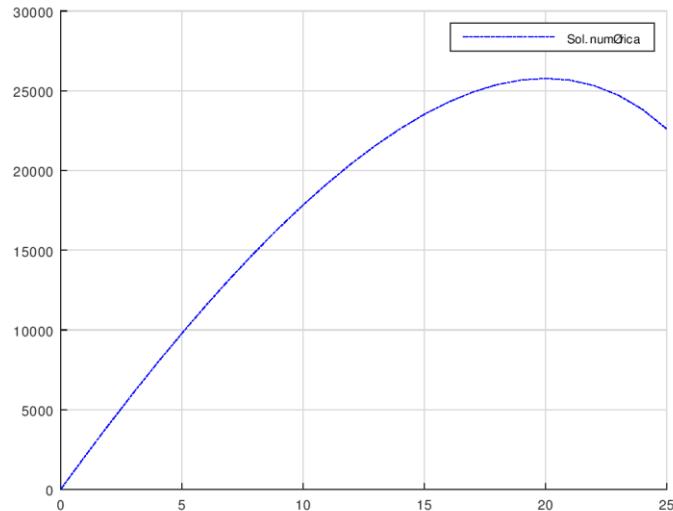
Introduce la solución analítica (Oprime Enter si no existe)

Datos:

```
x0 = 0
xn = 25
y0 = 0
Incremento de x = 1
```

i	x(i)	y(i)	k1	k2	k3	k4
0	0.000000	0.000000	2100.000000	2073.417722	2073.417722	2046.153846
1	1.000000	2073.304122	2046.153846	2018.181818	2018.181818	1989.473684
2	2.000000	4091.363256	1989.473684	1960.000000	1960.000000	1929.729730
3	3.000000	6051.230492	1929.729730	1898.630137	1898.630137	1866.666667
4	4.000000	7949.716649	1866.666667	1833.802817	1833.802817	1800.000000
5	5.000000	9783.362971	1800.000000	1765.217391	1765.217391	1729.411765
6	6.000000	11548.409860	1729.411765	1692.537313	1692.537313	1654.545455
7	7.000000	13240.760938	1654.545455	1615.384615	1615.384615	1575.000000
8	8.000000	14855.941591	1575.000000	1533.333333	1533.333333	1490.322581
9	9.000000	16389.050910	1490.322581	1445.901639	1445.901639	1400.000000
10	10.000000	17834.705766	1400.000000	1352.542373	1352.542373	1303.448276
11	11.000000	19186.975394	1303.448276	1252.631579	1252.631579	1200.000000
12	12.000000	20439.304493	1200.000000	1145.454545	1145.454545	1088.888889
13	13.000000	21584.422338	1088.888889	1030.188679	1030.188679	969.230769
14	14.000000	22614.234734	969.230769	905.882353	905.882353	840.000000
15	15.000000	23519.694764	840.000000	771.428571	771.428571	700.000000
16	16.000000	24290.647145	700.000000	625.531915	625.531915	547.826087
17	17.000000	24915.639436	547.826087	466.666667	466.666667	381.818182
18	18.000000	25381.691259	381.818182	293.023256	293.023256	200.000000
19	19.000000	25674.009793	200.000000	102.439024	102.439024	0.000000
20	20.000000	25775.635809	0.000000	-107.692308	-107.692308	-221.052632
21	21.000000	25666.998832	-221.052632	-340.540541	-340.540541	-466.666667
22	22.000000	25325.351922	-466.666667	-600.000000	-600.000000	-741.176471
23	23.000000	24724.044733	-741.176471	-890.909091	-890.909091	-1050.000000
24	24.000000	23831.575927	-1050.000000	-1219.354839	-1219.354839	-1400.000000
25	25.000000	22610.339368				

>>



9.61². La siguiente ecuación diferencial ordinaria describe el movimiento de un sistema masa-resorte amortiguado (figura):

$$m \frac{d^2x}{dt^2} + a \left| \frac{dx}{dt} \right| \frac{dx}{dt} + bx^3 = 0$$

donde x = desplazamiento desde el punto de equilibrio, t = tiempo, $m = 1$ kg masa y $a = 5$ N/(m/s)². El término de amortiguamiento es no lineal y representa el amortiguamiento por aire.

El resorte es un resorte cúbico y es también no lineal con $b = 5 \text{ N/m}^3$. Las condiciones iniciales son:

$$\text{Velocidad inicial} \quad \frac{dx}{dt} = 0.5 \text{ m/s}$$

$$\text{Desplazamiento inicial } x = 1 \text{ m}$$

Resolver esta ecuación usando un método numérico para un período de tiempo $0 \leq t \leq 8 \text{ s}$.

Graficar el desplazamiento y la velocidad contra el tiempo y la gráfica plano-fase (velocidad vs desplazamiento) para todos los siguientes casos:

- (a) Una ecuación lineal similar

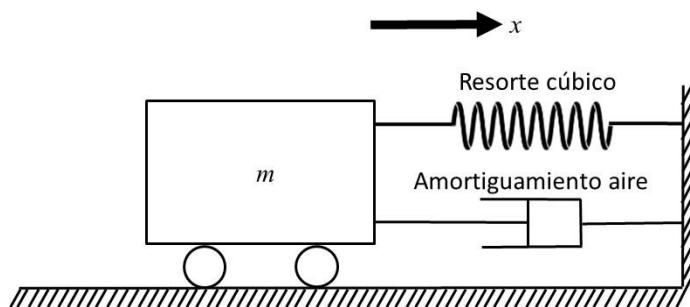
$$m \frac{d^2x}{dt^2} + 2 \frac{dx}{dt} + 5x = 0$$

- (b) La ecuación no lineal con solo un término resorte no lineal

$$\frac{d^2x}{dt^2} + 2 \frac{dx}{dt} + bx^3 = 0$$

- (c) La ecuación no lineal con solo un término de amortiguamiento no lineal.

$$m \frac{d^2x}{dt^2} + a \left| \frac{dx}{dt} \right| \frac{dx}{dt} + 5x = 0$$



- (a) Una ecuación lineal similar

$$m \frac{d^2x}{dt^2} + 2 \frac{dx}{dt} + 5x = 0$$

Procesando,

$$\frac{dx}{dt} = z$$

$$\frac{dx}{dt} = -2z - 5x$$

Usando comandos de Octave,

```
>> function fun=fcn(x,t)
fun(1) = x(2);
fun(2) = -2*x(2)-5*x(1);
endfunction
>> t=[0:0.5:8];
>> x0=[1, 0.5];
>> x=lsode("fcn",x0,t);
>> for i=1:length(t)
fprintf('%4.1f %8.4f %8.4f',t(i),x(i,1),x(i,2));
fprintf('\n')
endfor
0.0    1.0000    0.5000
0.5    0.7105   -1.2397
1.0    0.0978   -0.9965
1.5   -0.1973   -0.1970
2.0   -0.1653    0.2374
2.5   -0.0358    0.2281
3.0    0.0374    0.0622
3.5    0.0376   -0.0432
4.0    0.0109   -0.0512
4.5   -0.0067   -0.0177
5.0   -0.0084    0.0073
5.5   -0.0030    0.0112
6.0    0.0011    0.0047
6.5    0.0018   -0.0011
7.0    0.0008   -0.0024
7.5   -0.0002   -0.0012
8.0   -0.0004    0.0001
>>
```

(b) La ecuación no lineal con solo un término no lineal de resorte,

$$\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + 5x^3 = 0$$

Procesando

$$\frac{dx}{dt} = z$$

$$\frac{dx}{dt} = -2z - 5x^3$$

Usando comandos de Octave,

```

>> function fun=fcn(x,t)
fun(1) = x(2);
fun(2) = -2*x(2)-5*x(1)^3;
endfunction
>> t=[0:0.5:8];
>> x0=[1, 0.5];
>> x=lsode ("fcn",x0,t);
>> for i=1:length(t)
fprintf('%4.1f %8.4f %8.4f',t(i),x(i,1),x(i,2));
fprintf('\n')
endfor
0.0    1.0000    0.5000
0.5    0.7301   -1.0179
1.0    0.3183   -0.5697
1.5    0.1314   -0.2239
2.0    0.0601   -0.0835
2.5    0.0337   -0.0308
3.0    0.0239   -0.0114
3.5    0.0203   -0.0042
4.0    0.0190   -0.0016
4.5    0.0185   -0.0006
5.0    0.0183   -0.0002
5.5    0.0182   -0.0001
6.0    0.0182   -0.0000
6.5    0.0182   -0.0000
7.0    0.0182   -0.0000
7.5    0.0182   -0.0000
8.0    0.0182   -0.0000
>>

```

(c) La ecuación no lineal con solo un término no lineal de amortiguamiento

$$\frac{d^2x}{dt^2} + 2 \left| \frac{dx}{dt} \right| \frac{dx}{dt} + 5x = 0$$

Procesando

$$\frac{dx}{dt} = z$$

$$\frac{dx}{dt} = -2|z|z - 5x$$

Usando comandos de Octave,

```

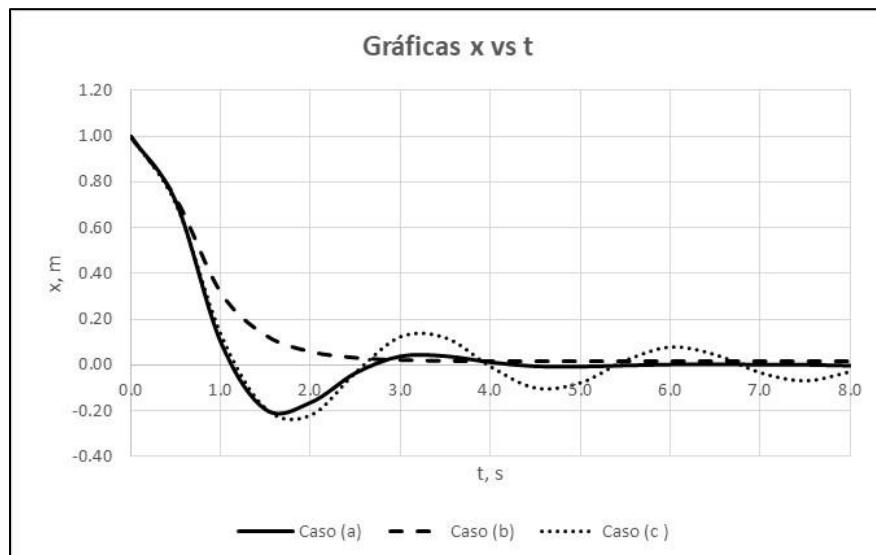
>> function fun=fcn(x,t)
fun(1) = x(2);
fun(2) = -2*abs(x(2))*x(2)-5*x(1);
endfunction
>> t=[0:0.5:8];

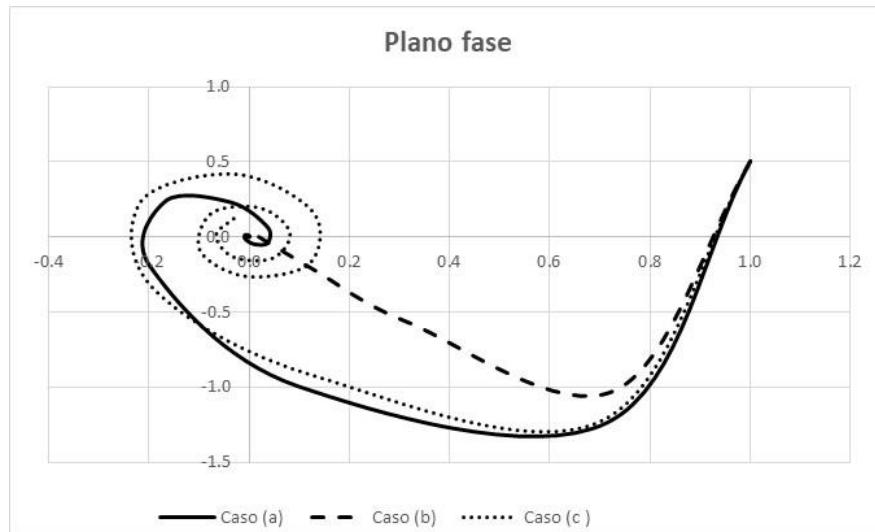
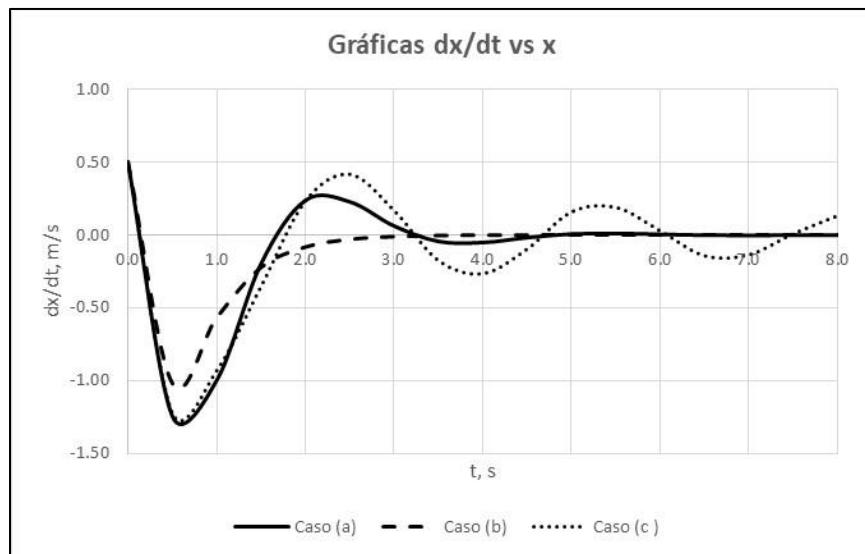
```

```

>> x0=[1, 0.5];
>> x=lsode ("fcn",x0,t);
>> for i=1:length(t)
fprintf('%4.1f %8.4f %8.4f',t(i),x(i,1),x(i,2));
fprintf('\n')
endfor
0.0    1.0000    0.5000
0.5    0.7052   -1.2235
1.0    0.1356   -0.9342
1.5   -0.1894   -0.3557
2.0   -0.2173    0.2319
2.5   -0.0299    0.4161
3.0    0.1259    0.1670
3.5    0.1210   -0.1766
4.0   -0.0050   -0.2665
4.5   -0.0984   -0.0793
5.0   -0.0753    0.1574
5.5    0.0221    0.1889
6.0    0.0808    0.0271
6.5    0.0470   -0.1438
7.0   -0.0316   -0.1375
7.5   -0.0669    0.0078
8.0   -0.0272    0.1309
>>

```





Apéndice C. Derivación de la regla de Simpson con corrección final.

Se sabe que el área de 2 paneles esta dada por:

$$I(x_j + h) - I(x_j - h) = I_{j+1} - I_{j-1} = 2hf(x_j) + \frac{h^3}{3!}f''(x_j) + \frac{h^5}{5!}f^{IV}(x_j) + \dots \quad (\text{C.1})$$

Aproximando polinomialmente con una función de 4º grado a los puntos de la siguiente figura B.1,

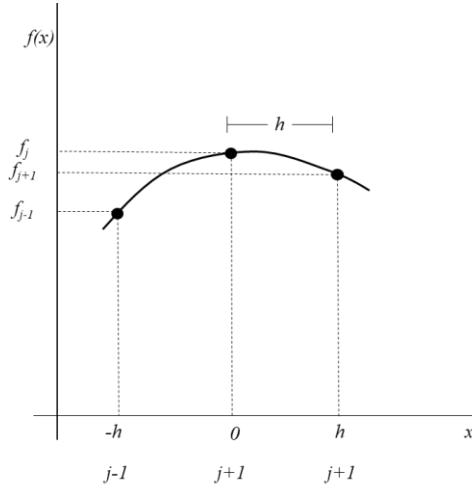


Figura B.1

$$f(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E \quad (\text{C.2})$$

evaluando en $j-1, j$ y $j+1$,

$$f(-h) = f_{j-1} = Ah^4 - Bh^3 + Ch^2 - Dh + E \quad (\text{C.3})$$

$$f(h) = f_j = E \quad (\text{C.4})$$

$$f(h) = f_{j+1} = Ah^4 + Bh^3 + Ch^2 + Dh + E \quad (\text{C.5})$$

sumando (C.3) y (C.4),

$$\begin{aligned} f_{j-1} + f_{j+1} &= 2Ah^4 + 2Ch^2 + 2f_j \\ 2Ah^4 + 2Ch^2 &= f_{j-1} - 2f_j + f_{j+1} \end{aligned} \quad (\text{C.6})$$

Suponiendo que se conocen las derivadas de $f(x)$ al final de cada doble panel, derivando la ecuación (C.2),

$$f'(x) = 4Ax^3 + 3Bx^2 + 2Cx + D$$

Entonces,

$$f'(-h) = f'_{j-1} = -4Ah^3 + 3Ch^2 - 2Ch + D \quad (\text{C.7})$$

$$f'(-h) = f'_{j-1} = -4Ah^3 + 3Ch^2 - 2Ch + D \quad (\text{C.8})$$

restando (C.8) de (C.7) y multiplicando por h ,

$$8Ah^4 + 4Ch^2 = (f'_{j+1} - f'_{j-1})h \quad (\text{C.9})$$

multiplicando por 2 la ecuación (C.6) y restándola de (C.9),

$$4Ah^4 + 4Ch^2 = 2f_{j-1} - 4f_j + 2f_{j+1}$$

$$4Ah^4 = (f'_{j+1} - f'_{j-1})h - 2f_{j-1} + 4f_j - 2f_{j+2} \quad (\text{C.10})$$

multiplicando la ecuación (C.6) por 4 y restándole la ecuación (C.9),

$$8Ah^4 + 8Ch^2 = 4f_{j-1} - 8f_j + 4f_{j+1}$$

$$4Ch^2 = 4f_{j-1} - 8f_j + 4f_{j+1} - (f'_{j+1} - f'_{j-1})h \quad (\text{C.11})$$

Por otro lado,

$$f''(x) = 12Ax^2 + 6Bx + 2C$$

$$f''(0) = f''_j = 2C \quad (\text{C.12})$$

despejando C de la ecuación (C.11) y procesando con la ecuación (C.12),

$$f''_j = \frac{4f_{j-1} - 8f_j + 4f_{j+1}}{2h^2} - \frac{f'_{j+1} - f'_{j-1}}{2h} \quad (\text{C.13})$$

también, derivando la ecuación (C.2),

$$f'''(x) = 24Ax + 6B \quad (\text{C.14})$$

$$f^{IV}(x) = 24A$$

Despejando A de la ecuación (C.10) y sustituyéndola en la ecuación (C.14),

$$f^{IV}(0) = f_j^{IV} = \frac{-12f_{j-1} + 24f_j - 12f_{j+1}}{h^4} + \frac{6(f'_{j+1} - f'_{j-1})}{h^4} \quad (\text{C.15})$$

Sustituyendo las ecuaciones (C.13) y (C.15) en la ecuación (C.1),

$$I_{j+1} - I_{j-1} = 2hf_j + \frac{h^3}{6} \left[\frac{1}{2h^2} (4f_{j-1} - 8f_j + 4f_{j+1}) - (f'_{j+1} - f'_{j-1}) \right] \\ + \frac{h^5}{60} \left[\frac{1}{h^4} (-12f_{j-1} + 24f_j - 12f_{j+1}) + 6(f'_{j+1} - f'_{j-1}) \right] \quad (\text{C.16})$$

procesando la ecuación (C.16),

$$I_{j+1} - I_{j-1} = \frac{h}{15} [7f_{j-1} + 16f_j + 7f_{j+1} + h(f'_{j-1} - f'_{j+1})] \quad (\text{C.17})$$

La ecuación (C.17) representa el área de dos párteles. Para n paneles (desde a hasta b) se tiene,

$$I = \frac{h}{15} [7f(a) + 16f(a+h) + 14f(a+2h) + 16f(a+3h) + \cdots + 16f(b-3h) \\ + 14f(b-2h) + 16f(b-h) - hf(b)]$$

también,

$$I = \frac{h}{15} \left\{ 14 \left[\frac{1}{2} f(a) + f(a+2h) + f(a+4h) + \cdots + f(b-2h) + \frac{1}{2} f(b) \right] \right. \\ \left. + 16[f(a+h) + f(a+3h) + \cdots + f(b-3h) + f(b-h)] + h[f'(a) - f'(b)] \right\}$$

y finalmente,

$$I = \frac{h}{15} \left\{ 14 \left[\frac{1}{2} (f_0 + f_n) + \sum_{\substack{j=2 \\ j \text{ par}}}^{n-2} f_j \right] + 16 \sum_{\substack{j=1 \\ j \text{ impar}}}^{n-1} f_j + h[f'(a) - f'(b)] \right\} \quad (\text{C.18})$$