Tema 2 - Operaciones matemáticas básicas

Técnicas de Interpolación

M. en C. Gustavo Contreras Mayén

Facultad de Ciencias - UNAM

22 de febrero de 2018

Objetivo General del Tema 2

Al concluir el Tema 2 del curso, el alumno empleará las técnicas más comunes de operaciones matemáticas necesarias para la solución de problemas numéricos aplicados en la física.

Objetivo General del Tema 2

Este tema es una base importante de todo el curso, ya que se establecen las tareas de:

1. Interpolación.

Objetivo General del Tema 2

Este tema es una base importante de todo el curso, ya que se establecen las tareas de:

1. Interpolación.
2. Diferenciación.

Objetivo General del Tema 2

Este tema es una base importante de todo el curso, ya que se establecen las tareas de:

1. Interpolación.
2. Diferenciación.
3. Integración numérica.

Objetivo General del Tema 2

Este tema es una base importante de todo el curso, ya que se establecen las tareas de:

1. Interpolación.
2. Diferenciación.
3. Integración numérica.
4. Cálculo de raíces.

Objetivo General del Tema 2

De tal manera que en los temas posteriores, se emplearán éstas técnicas, y en algunos casos, se extenderán de manera específica.

Iniciamos

Dados los $n+1$ pares de datos (x_i, y_i), con $i = 0, 1, \ldots, n$, queremos estimar el valor de $y(x)$

Interpolación

Una parte importante en el trabajo del físico es la interpretación de datos experimentales o cálculos teóricos.

Interpolación

Normalmente cuando hacemos mediciones, tenemos un conjunto discreto de puntos que representan nuestro experimento.

Para facilitar el trabajo, suponemos que el experimento puede representarse por un par de valores: una *variable independiente* x la cual podemos controlar y una cantidad y, que se mide en el punto x.

Interpolación vs ajuste con curvas

Hay una diferencia entre interpolación y el ajuste de curvas.

En la interpolación se *construye una curva a través de los puntos de datos*.

Interpolación vs ajuste con curvas

El ajuste de curvas se aplica a los datos que contienen una dispersión (ruido), por lo general debida a errores de medición.

En este caso, *queremos encontrar una curva suave que se aproxima a los datos en algún sentido*. De tal manera que la curva no toca necesariamente los puntos de datos.

Interpolación vs ajuste con curvas

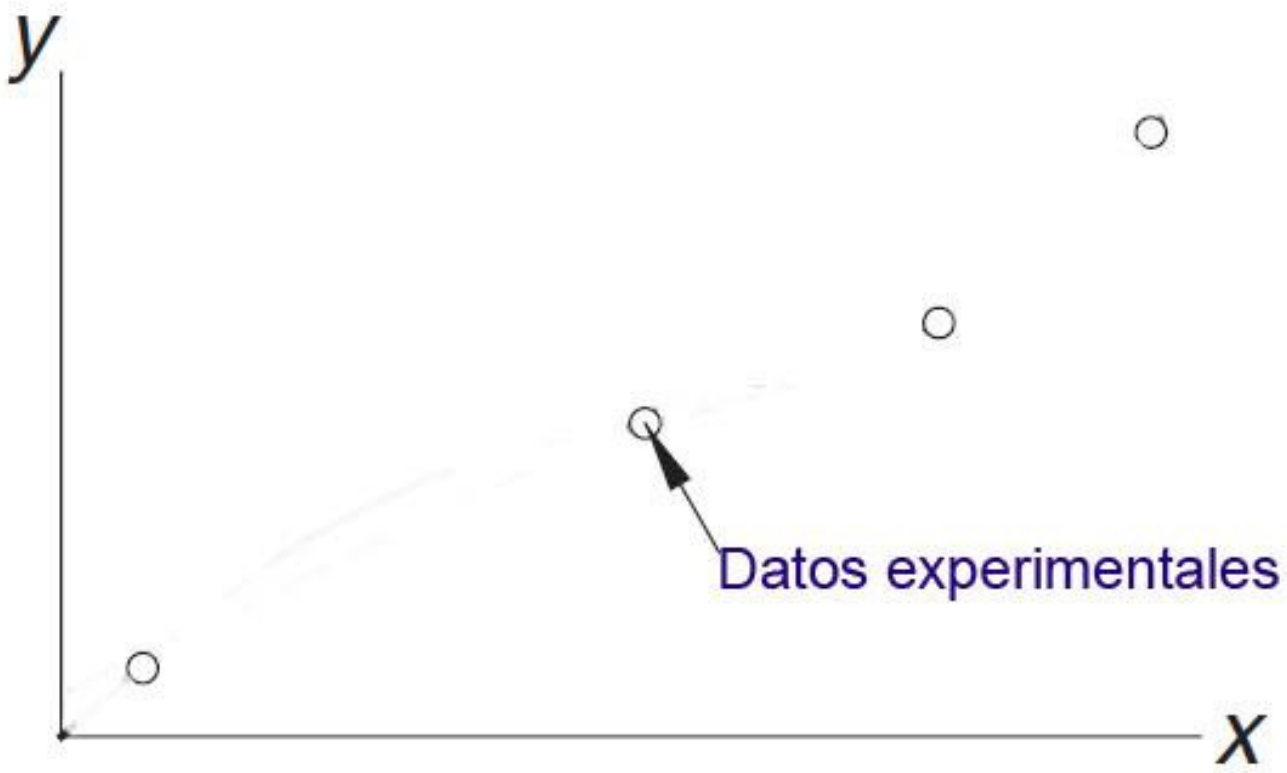


Figura 1: *Tenemos un conjunto de datos experimentales obtenidos en el laboratorio.*

Interpolación vs ajuste con curvas

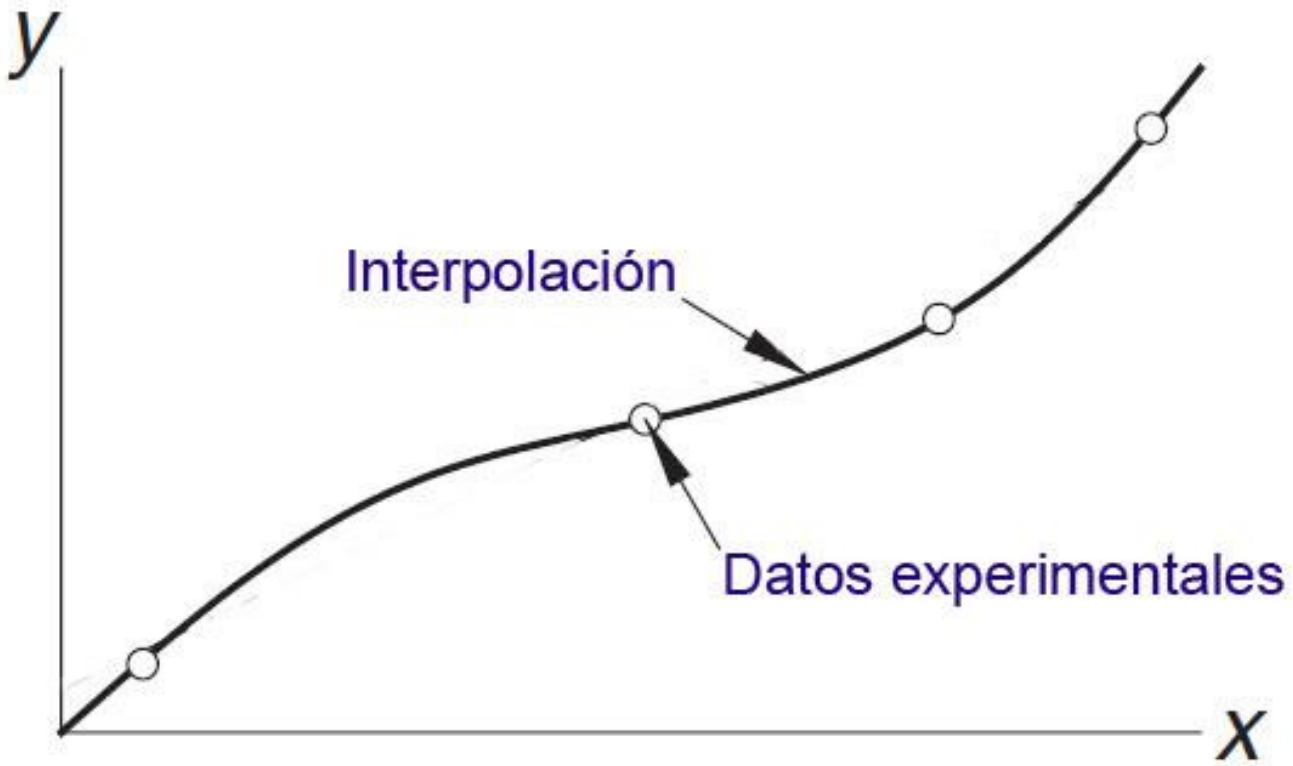


Figura 2: *Con la interpolación se construye una curva suave que "toca" a los puntos experimentales*

Interpolación vs ajuste con curvas

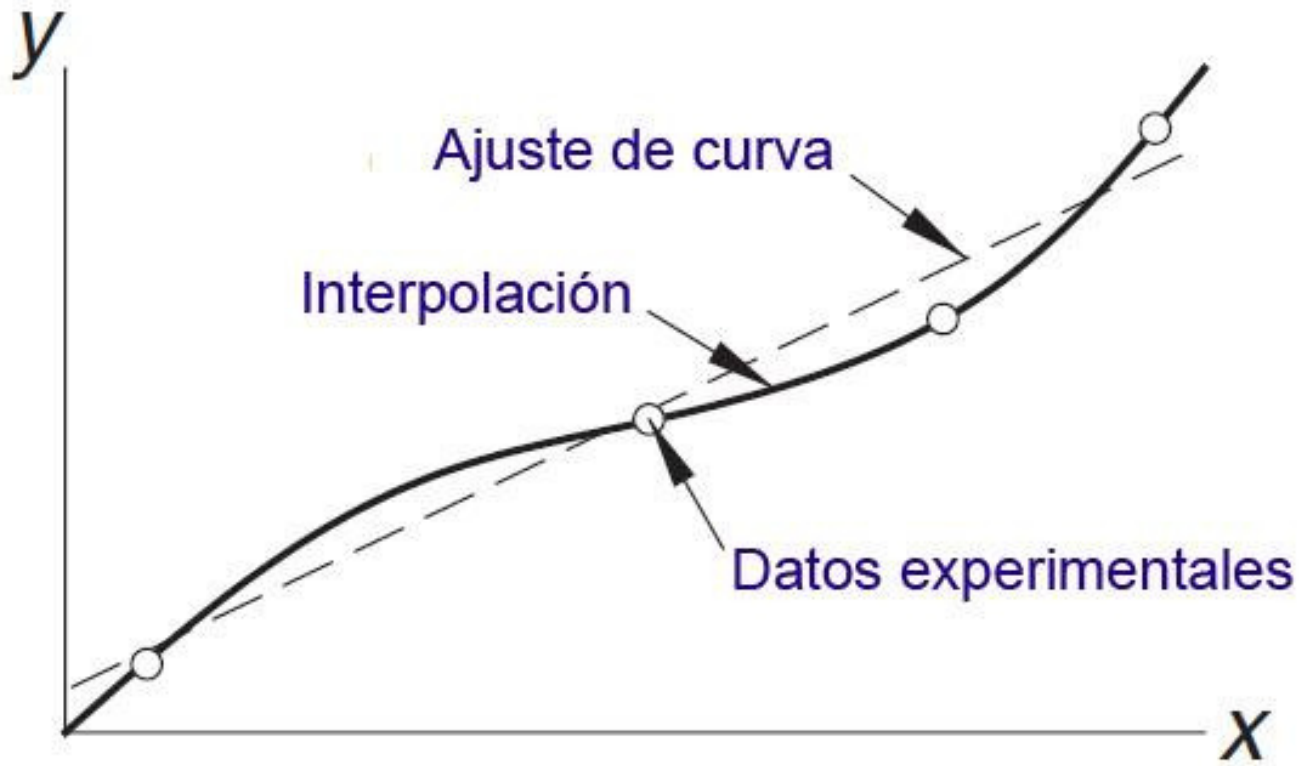


Figura 3: *La técnica de ajuste de curvas es una aproximación al conjunto de datos.*

Ejemplo

Tomemos como ejemplo una fuente radioactiva y un detector, el cual contabiliza el número de decaimientos.

Para determinar la vida media de la fuente, debemos de contar los decaimientos N_0, N_1, N_2, . . ., N_k, en los tiempos t_0, t_1, t_2, . . ., t_k

Ejemplo

En este ejemplo, la variable independiente es t, siendo la forma apropiada para resolver el problema.

Sin embargo, tenemos un conjunto discreto de pares de números (t_k, N_k) en el rango de (t_0, t_k)

Ejemplo

Con la intención de obtener información del experimento, deberíamos de encontrar una función analítica que nos devuelva el valor de N para cualquier punto arbitrario t.

Ejemplo

Pero a veces, el tratar de encontrar una función analítica es imposible, o el pensar en utilizar una función conocida, nos podría llevar mucho tiempo para calcularla, más si nuestro interés se basa en una pequeña vecindad de la variable independiente.

Ejemplo 2

Supongamos que tenemos una fuente radioactiva de ^{241}Am, una fuente de rayos α. Su vida media es $\tau_{\frac{1}{2}} = 430$ años.

Obviamente no podríamos determinar su vida media midiéndola, ya que el decaimiento es lento y quizá lo que podríamos hacer es medir cada lunes durante algunos meses: después de cinco meses (por ejemplo) podríamos detener las mediciones y revisar los datos.

Ejemplo 2

Una pregunta que nos podemos plantear es: ¿cuál fue la actividad el miércoles de la tercera semana de mediciones?

Ya que ese día está dentro del rango de mediciones (t_0, t_k)

Ejemplo 2

Lo que podríamos hacer es usar técnicas de interpolación para determinar ese valor.

Ejemplo 2

Lo que podríamos hacer es usar técnicas de interpolación para determinar ese valor.

Si lo que queremos, es el caso contrario, conocer la actividad luego de ocho meses posteriores a la última medición, lo que deberíamos de hacer es extrapolar a ese punto a partir de las mediciones previas.

Objetivo de la interpolación.

La idea central de la interpolación es seleccionar una función $g(x)$ tal que $g(x_i) = f_i$ para cada dato i, es una buena aproximación para cualquier otro dato x entre el conjunto original de datos.

Objetivo de la interpolación.

Pero ¿cómo podemos considerar una buena aproximación al conjunto de datos, si no tenemos la función original?

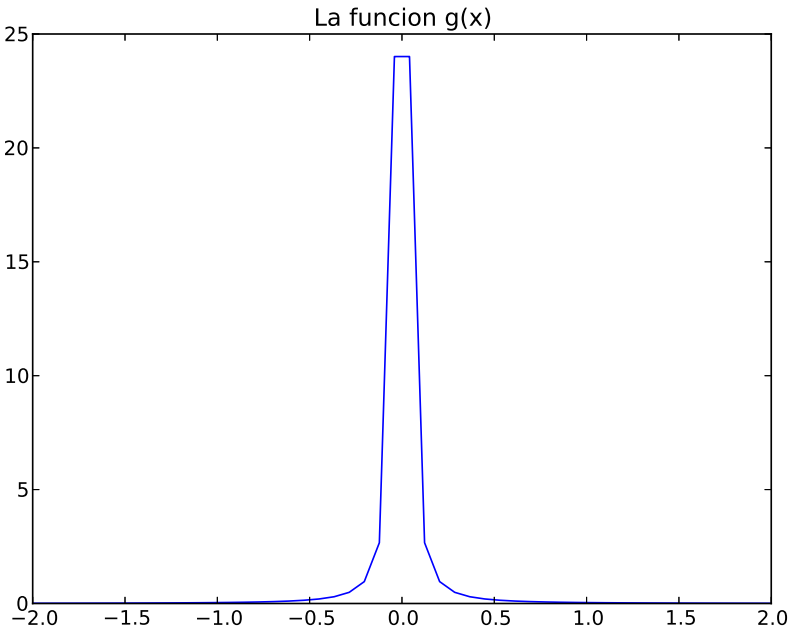
Dado que los puntos ser pueden interpolar por una familia infinita de funciones, para ello debemos de contar con algún criterio o guía para seleccionar una función razonable.

La regla para esos métodos se basa en la suavidad al ajuste de las funciones de interpolación.

La regla para esos métodos se basa en la suavidad al ajuste de las funciones de interpolación.

Pero esto no podría funcionar para todo tipo de funciones, consideremos la función:

$$g(x) = \frac{1}{25x^2}$$



Consideración previa

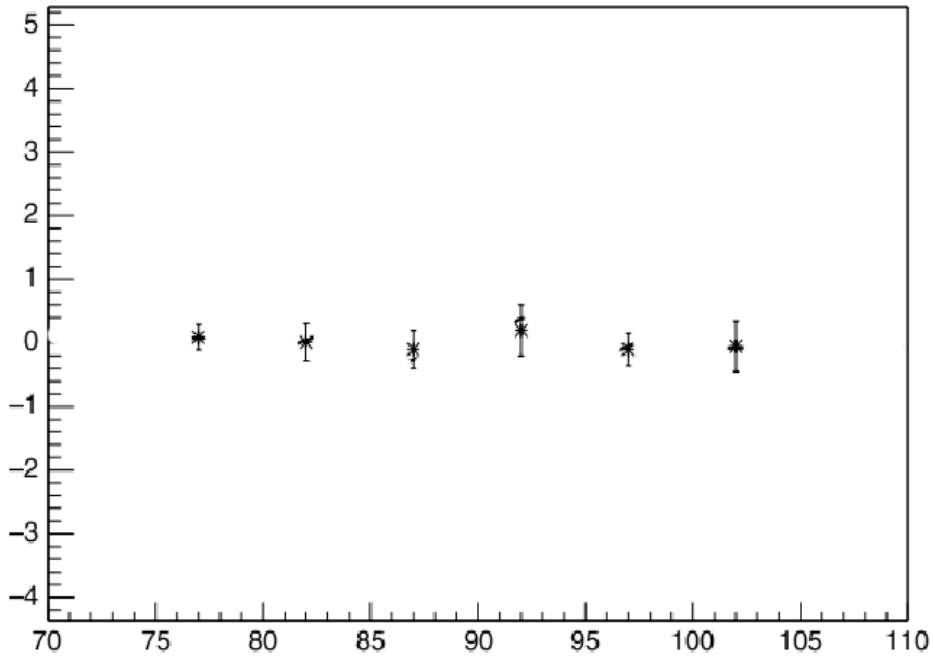
Antes de entrar de lleno a la revisión de las técnicas de interpolación, es necesario mencionar lo siguiente: dado que contamos con un conjunto finito de puntos, debemos de tener cuidado en el espaciamiento de la variable independiente.

Consideración previa

Si los puntos se alejan unos de otros, perderemos información para aquellos valores entre éstos puntos y la predicción de la interpolación ya no será la esperada.

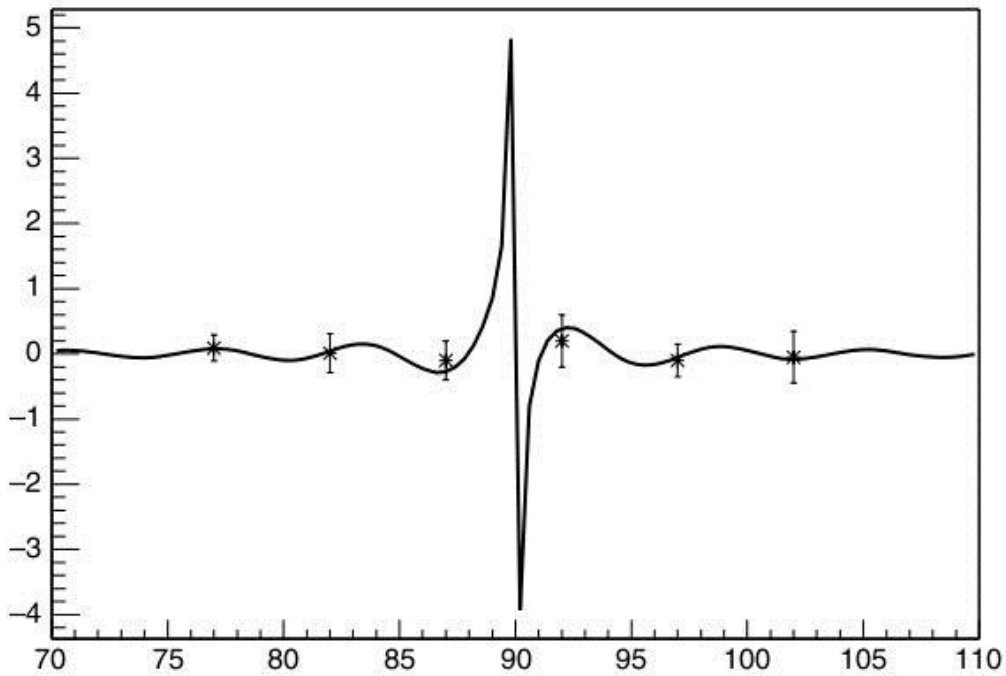
Ejemplo 3

Supongamos que tenemos seis mediciones como se indican en la siguiente figura, podemos ver claramente un comportamiento oscilatorio de la función, juzgando por los puntos y de acuerdo a las barras de error, una línea recta es la que probablemente nos ajustaría los puntos.



Gráfica de una interpolación

Las técnicas de interpolación hacen lo que pidamos, pero no necesariamente son consistentes con un fenómeno o modelo.



Interpolación de Lagrange

La técnica más sencilla de interpolación, es usando polinomios. Siempre es posible construir un *único* polinomio de grado n que pasa a través de $n + 1$ puntos.

Una manera de obtener este polinomio es usando la fórmula de Lagrange

Fórmula de Lagrange

$$P_n(x) = \sum_{i=0}^{n} y_i \, \ell_i(x) \tag{1}$$

donde n es el grado del polinomio y

$$\begin{aligned} \ell_i(x) &= \frac{x - x_0}{x_i - x_0} \cdot \frac{x - x_1}{x_i - x_1} \cdots \frac{x - x_{i+1}}{x_i - x_{i+1}} \cdot \frac{x - x_n}{x_i - x_n} \\ &= \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n \end{aligned} \tag{2}$$

se llaman *funciones cardinales*.

Por ejemplo, si $n = 1$

La interpolación es una línea recta:

$$P_1(x) = y_0\,\ell_0(x) + y_1\,\ell_1(x)$$

y las funciones cardinales son

$$\ell_0(x) = \frac{x - x_1}{x_0 - x_1} \qquad \ell_1(x) = \frac{x - x_0}{x_1 - x_0}$$

Con $n = 2$

La interpolación es parabólica:

$$P_2(x) = y_0\,\ell_0(x) + y_1\,\ell_1(x) + y_2\,\ell_2(x)$$

y las $\mathcal{L}_i(x)$ son:

$$\begin{aligned}
\ell_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \\
\ell_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \\
\ell_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}
\end{aligned}$$

Propiedad de las funciones cardinales

Las funciones cardinales son polinomios de orden n que tienen la propiedad

$$\ell_i(x_j) = \begin{cases} 0 \text{ si } i \neq j \\ 1 \text{ si } i = j \end{cases} = \delta_{ij} \tag{3}$$

donde δ_{ij} es la delta de Kronecker.

Fórmula de Lagrange

Para probar que el polinomio de interpolación pasa por los puntos experimentales, sustituimos $x = x_j$ en la definición de $P_n(x)$ (ec. 1) y usamos la ec. (3, entonces:

$$P_n(x_j) = \sum_{i=0}^{n} y_i \, \ell_i(x_j) = \sum_{i=0}^{n} y_i \, \delta_{ij} = y_j$$

Error en la fórmula de Lagrange

Se puede demostrar que el error en el polinomio de interpolación es

$$f(x) - P_n(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_n)}{(n+1)!} f^{(n+1)}(\xi)$$

donde $\xi \in (x_0, x_n)$, y éste valor normalmente no se conoce.

Debemos de notar que mientras más lejos están los datos de x, contribuye a que el error se incremente.

Ejemplo

Veamos el cambio de la presión del vapor de ^{4}He como función de la temperatura, de acuerdo a la literatura tenemos que:

Temperatura [K]	Presión de vapor [kPa]
2.3	6.38512
2.7	13.6218
2.9	18.6760
3.2	28.2599
3.5	40.4082
3.7	49.9945

Gráfica de los puntos experimentales

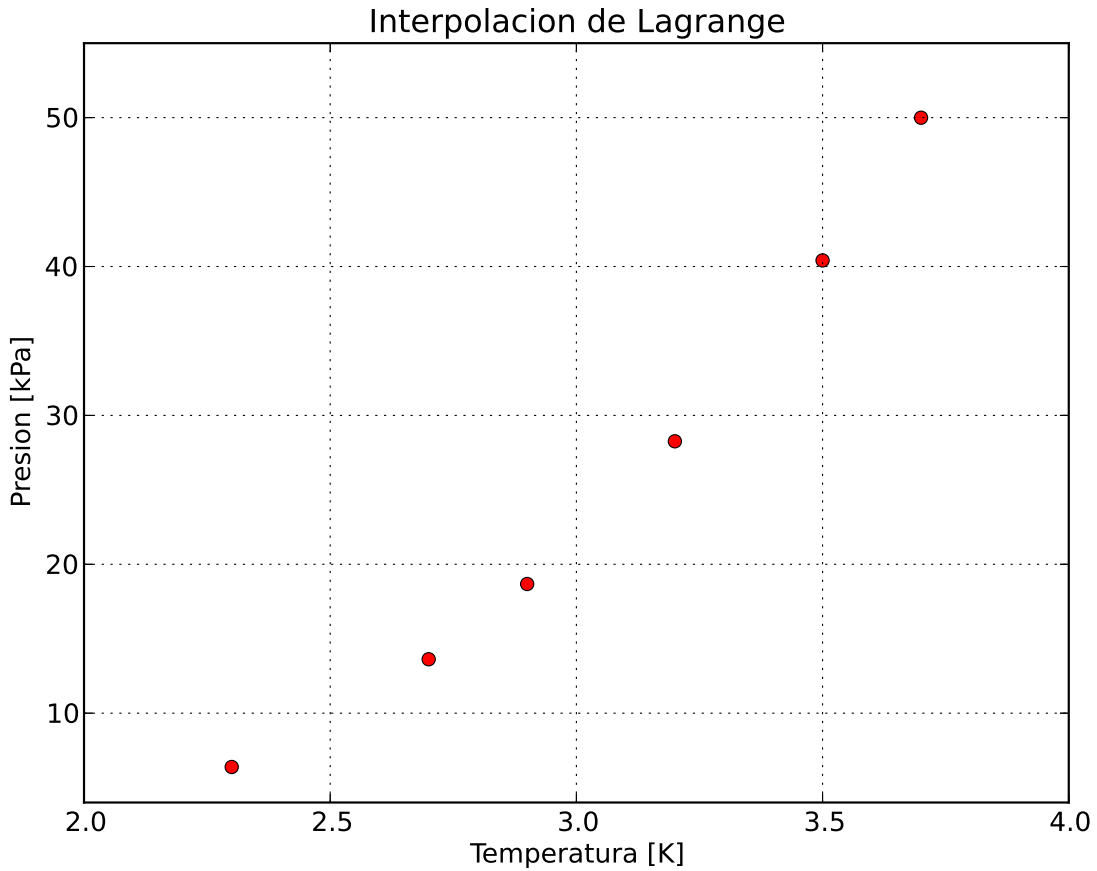
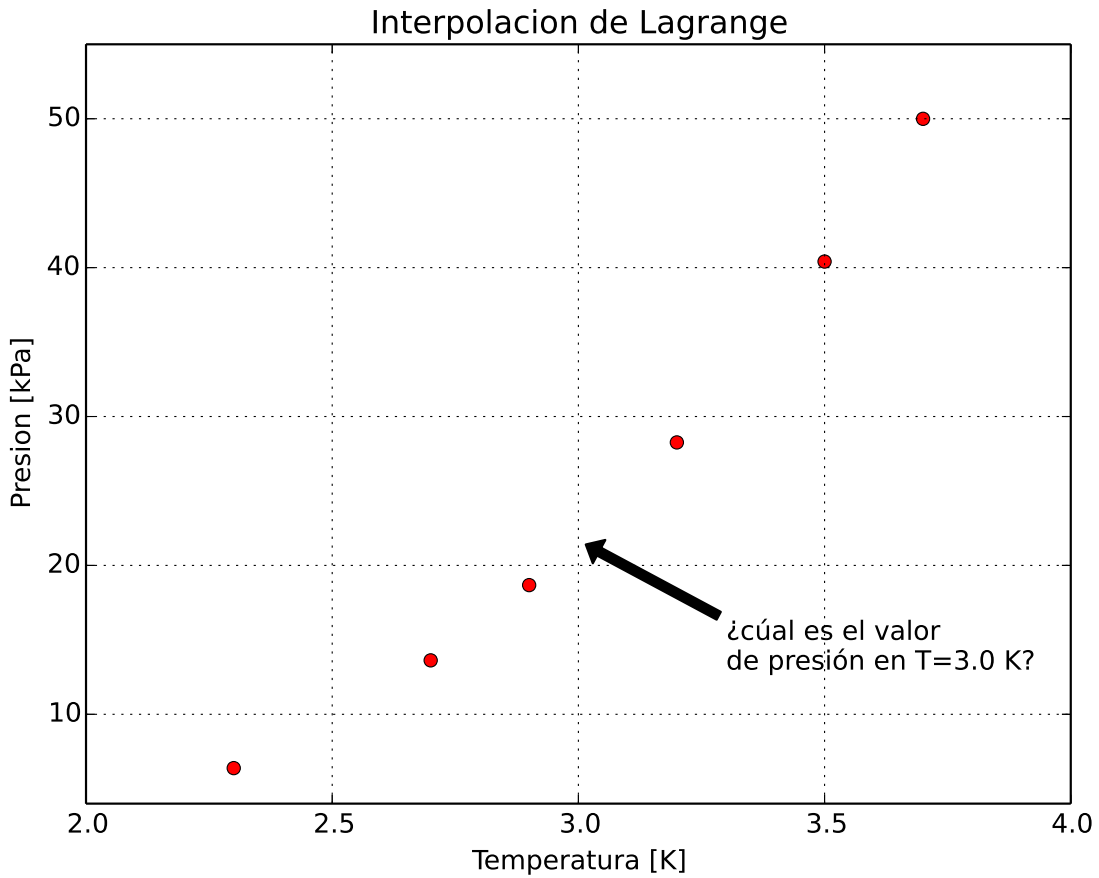


Figura 4: *El conjunto de datos experimentales.*

Pregunta para resolver

¿Cuál es el valor de presión a una temperatura de 3 K?



Solución

Tomamos lo que ya conocemos para $n = 1$

$$P_1(x) = y_0\, \ell_0(x) + y_1\, \ell 1(x)$$

Solución

Tomamos lo que ya conocemos para $n = 1$

$$P_1(x) = y_0\, \ell_0(x) + y_1\, \ell 1(x)$$

y las funciones cardinales son

$$\ell_0(x) = \frac{x - x_1}{x_0 - x_1} \qquad \ell_1(x) = \frac{x - x_0}{x_1 - x_0}$$

Solución

En nuestro ejemplo, tenemos que:

$$(x_0, y_0) = (2.9, 18.6760)$$

$$(x_1, y_1) = (3.2, 28.2599)$$

Solución

En nuestro ejemplo, tenemos que:

$$(x_0, y_0) = (2.9, 18.6760)$$

$$(x_1, y_1) = (3.2, 28.2599)$$

Con la interpolación lineal, tenemos que para una temperatura de 3.0 K, la presión tiene un valor de 21.87 kPa

Interpolación cuadrática

Con la intención de mejorar nuestro resultado, podemos usar un polinomio de segundo orden, es decir $n = 2$

$$P_2(x) = y_0\,\ell_0(x) + y_1\,\ell_1(x) + y_2\,\ell_2(x)$$

Interpolación cuadrática

Las funciones cardinales $\ell_i(x)$ son:

$$\ell_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

$$\ell_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$\ell_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Usando los datos experimentales

Usando los valores de la tabla anterior:

$$(x_0, y_0) = (2.7, 13.6218)$$
$$(x_1, y_1) = (2.9, 18.6760)$$
$$(x_2, y_2) = (3.2, 28.2599)$$

Usando los datos experimentales

Usando los valores de la tabla anterior:

$$(x_0, y_0) = (2.7, 13.6218)$$
$$(x_1, y_1) = (2.9, 18.6760)$$
$$(x_2, y_2) = (3.2, 28.2599)$$

A una temperatura de 3.0 K, la presión de vapor tiene un valor de 21.671 kPa

Usando los datos experimentales

Usando los valores de la tabla anterior:

$$(x_0, y_0) = (2.7, 13.6218)$$
$$(x_1, y_1) = (2.9, 18.6760)$$
$$(x_2, y_2) = (3.2, 28.2599)$$

A una temperatura de 3.0 K, la presión de vapor tiene un valor de 21.671 kPa

El siguiente paso es usar cuatro puntos y construir el polinomio de orden 3.

Ejercicio

A partir de la siguiente tabla de datos,

x	P(x)
1	0.671
2	0.620
3	0.567
4	0.512

Ejercicio

A partir de la siguiente tabla de datos,

x	P(x)
1	0.671
2	0.620
3	0.567
4	0.512

Usa el algoritmo de interpolación de Lagrange con un polinomio de grado $n = 3$, para estimar el valor de $P(x)$ en los siguientes puntos:

$x = 1.5, 2.5, 3.5$

Solución al problema I

Código 1: Código para la interpolación de Lagrange

```
1  import numpy as np
2  n = 3
3  x0 = np.array([1.5, 2.5, 3.5])
4  x = np.array([1., 2., 3., 4.])
5  f = np.array([0.671, 0.620, 0.567, 0.512])
6
7  for k in x0:
8      yres = 0
9      for i in range(0, n + 1):
10         z = 1.0
11         for j in range(0, n + 1):
12             if i != j:
```

Solución al problema II

```
13                    z = z * (k - x[j])/(x[i] - x[j
    ])
14          yres = yres + z*f[i]
15
16     print ('El polinomio evaluado en P(',k,')
    =',  yres)
```

Solución (en la terminal y con una gráfica)

x	P(x)
1	0.671
1.5	0.64575
2	0.620
2.5	0.59375
3	0.567
3.5	0.53975
4	0.512

Solución con una gráfica

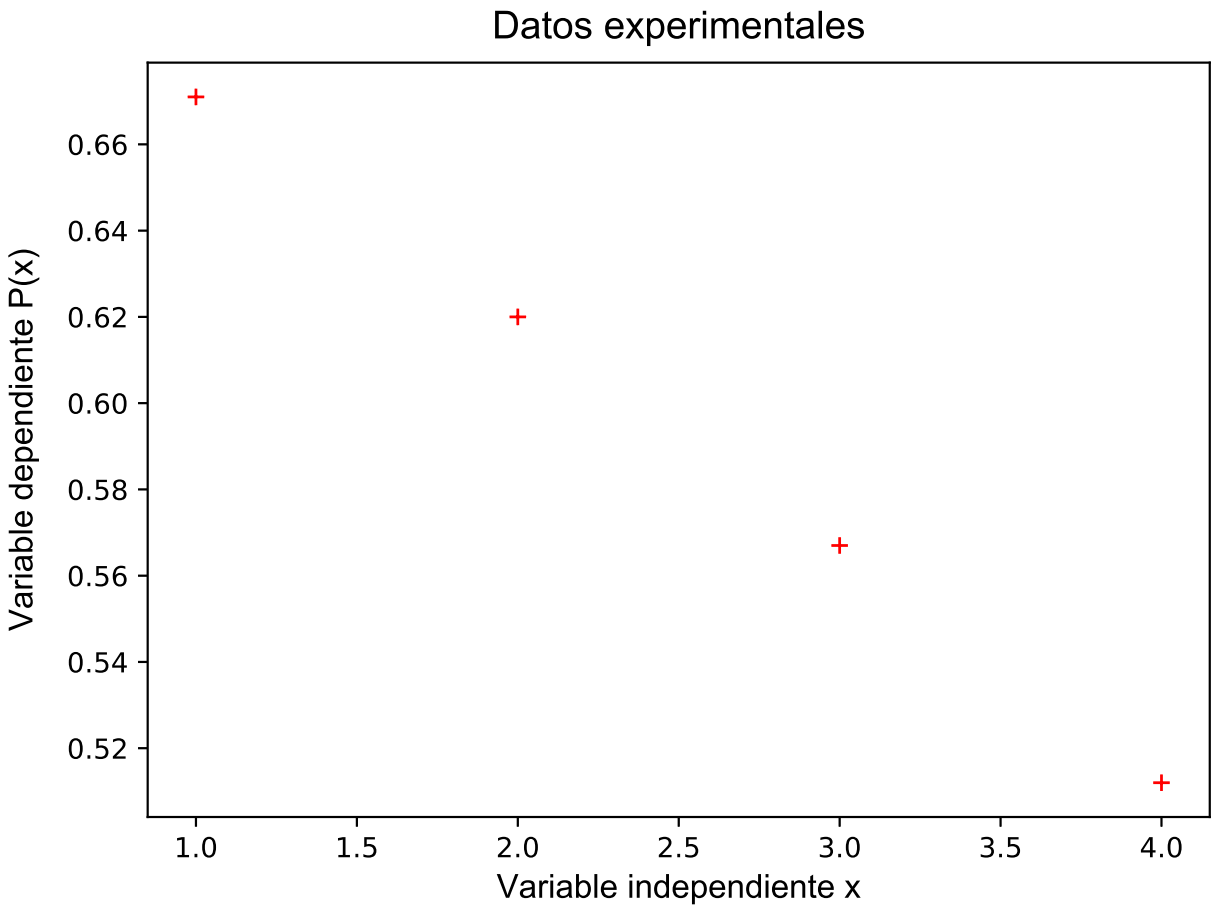


Figura 5: *Gráfica con los datos experimentales*

Solución con una gráfica

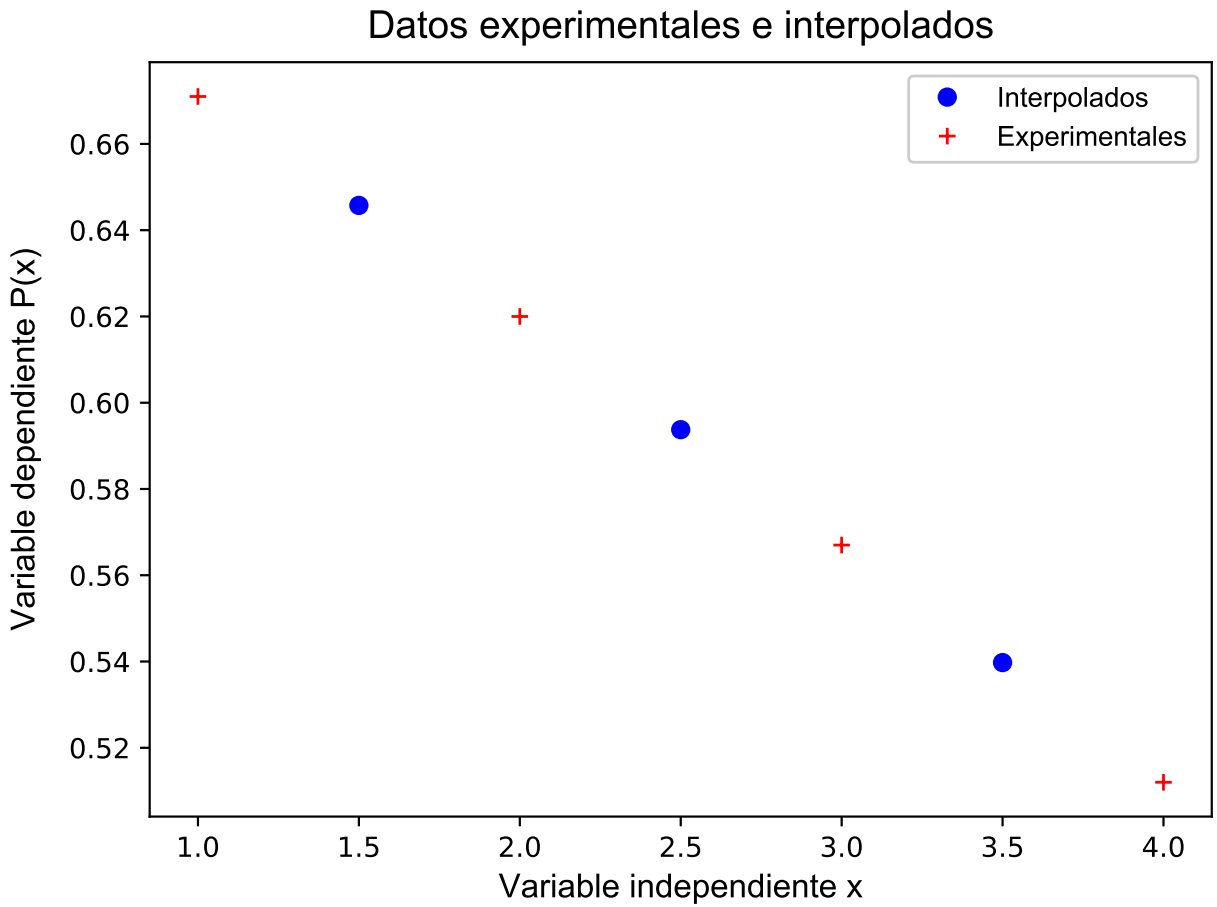


Figura 6: *Gráfica con los datos experimentales y el valor de interpolación -bola azul-*

Ejercicios

1. Ajusta $x\,\mathrm{sin}(x)$ en $[0, \pi/2]$ con un polinomio de interpolación de Lagrange de orden $n = 4$, utilizando puntos con igual separación.

   Calcula el error de cada interpolación en cada incremento de $\pi/16$, muestra una gráfica.

Ejercicios

2. Ajusta $\sin(x)$ en $[0, 2\pi]$ con el polinomio de interpolación de Lagrange de orden $n = 4$ y $n = 8$, utilizando puntos con igual separación (5 y 9 puntos respectivamente).

   Grafica los polinomios de interpolación junto con $\sin(x)$ y las distribuciones de sus errores.

Hints para resolver los ejercicios

Tomemos en cuenta los siguiente:

1. Hay que dividir el intervalo $[0, \pi/2]$ con cinco espacios.

Hints para resolver los ejercicios

Tomemos en cuenta los siguiente:

1. Hay que dividir el intervalo $[0, \pi/2]$ con cinco espacios.
2. Se evalúa la función $x\ sin(x)$ en el conjunto de puntos que obtuvimos.

Hints para resolver los ejercicios

Tomemos en cuenta los siguiente:

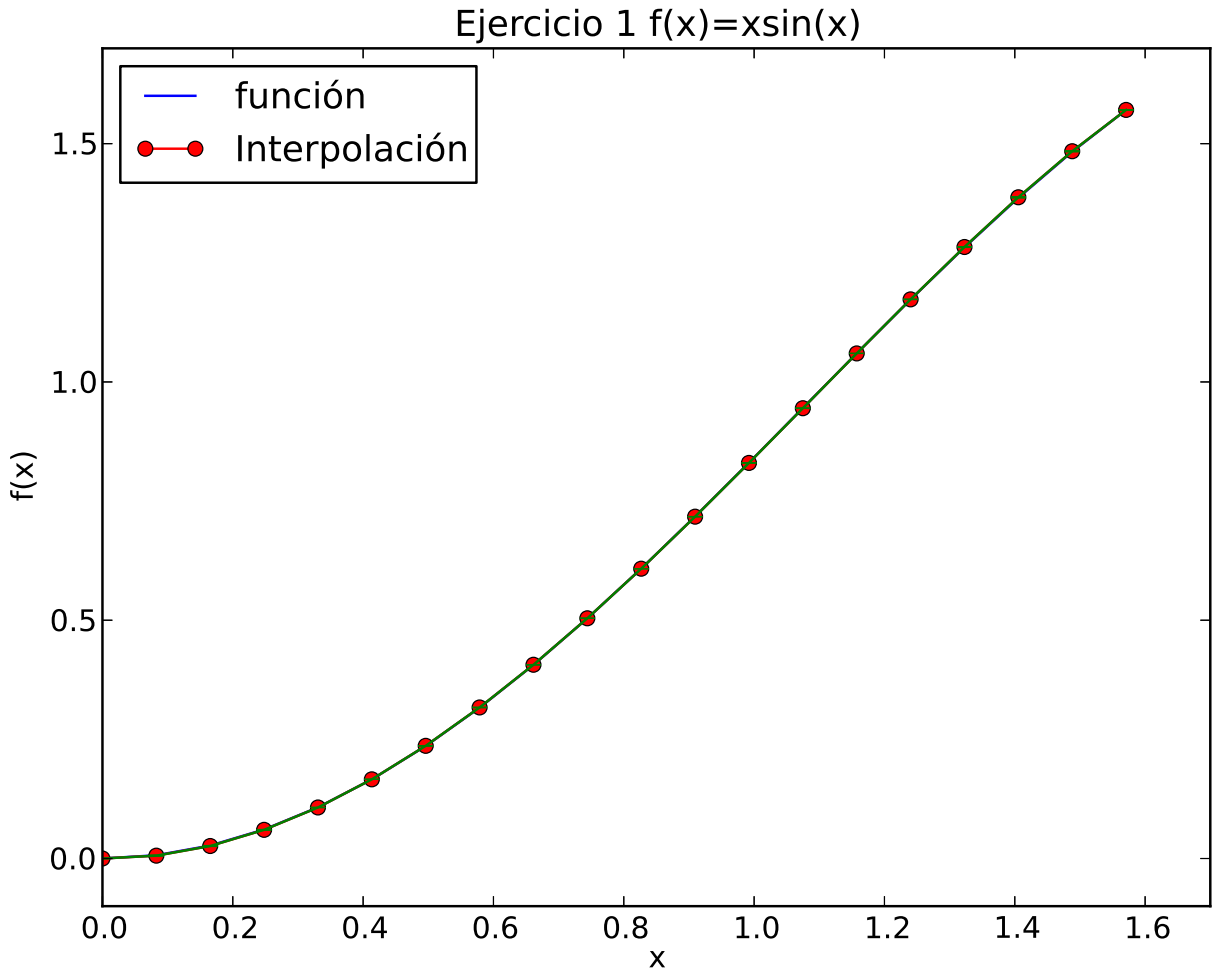
1. Hay que dividir el intervalo $[0, \pi/2]$ con cinco espacios.
2. Se evalúa la función $x\ sin(x)$ en el conjunto de puntos que obtuvimos.
3. Creamos un conjunto de puntos para ajustar con el método de Lagrange.

Hints para resolver los ejercicios

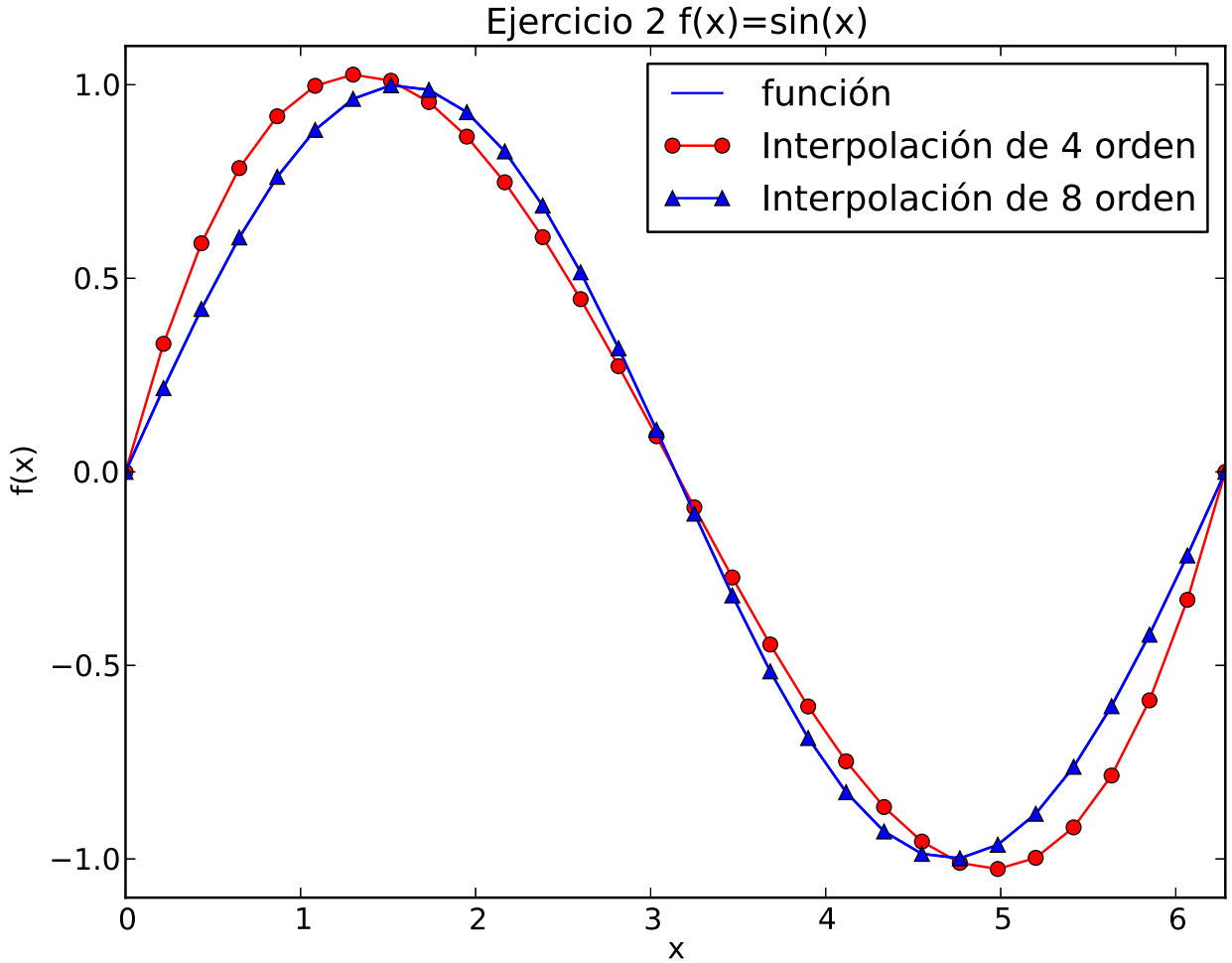
Tomemos en cuenta los siguiente:

1. Hay que dividir el intervalo $[0, \pi/2]$ con cinco espacios.
2. Se evalúa la función $x\ sin(x)$ en el conjunto de puntos que obtuvimos.
3. Creamos un conjunto de puntos para ajustar con el método de Lagrange.
4. La manera más fácil es con un `linspace`.

Solución Ejercicio 1



Solución Ejercicio 2



Consideraciones importantes

La técnica de interpolación de Lagrange supone que el espaciamiento entre los puntos es la misma, por lo que cuando se presentan puntos que no cumplen ésta condición, la técnica ya no aplicaría.

Desventajas interpolación de Lagrange

A continuación se presentan algunas de las desventajas de la primera técnica de interpolación, al interpolación de Lagrange que hemos estudiado.

Desventajas interpolación Lagrange

- ✗ La cantidad de cálculos necesarios para una interpolación es grande.

Desventajas interpolación Lagrange

- ✗ La cantidad de cálculos necesarios para una interpolación es grande.
- ✗ La interpolación para otro valor de x necesita la misma cantidad de cálculos adicionales, ya que no se pueden utilizar partes de la aplicación previa.

Desventajas interpolación Lagrange

- ✗ Cuando el número de datos tiene que incrementarse o decrementarse, no se pueden utilizar los resultados de los cálculos previos.

Desventajas interpolación Lagrange

- ✗ Cuando el número de datos tiene que incrementarse o decrementarse, no se pueden utilizar los resultados de los cálculos previos.
- ✗ La evaluación del error no es fácil.

Interpolación de Newton

Aunque el método de interpolación de Lagrange es conceptualmente sencillo, no es en sí, un algoritmo eficiente.

Un mejor método computacional se obtiene con el Método de Newton.

Interpolación de Newton

El polinomio de interpolación de Newton se escribe de la forma:

$$P_n(x) = a_0 + (x - x_0)\, a_1 + (x - x_0)(x - x_1)\, a_2 + \ldots + \\ + (x - x_0)(x - x_1) \ldots (x - x_{n-1})\, a_n$$

Interpolación de Newton

El polinomio de interpolación de Newton se escribe de la forma:

$$P_n(x) = a_0 + (x - x_0)\, a_1 + (x - x_0)(x - x_1)\, a_2 + \ldots + \\ + (x - x_0)(x - x_1) \ldots (x - x_{n-1})\, a_n$$

Este polinomio nos permite contar con un procedimiento de evaluación más eficiente.

Evaluación más eficiente

Por ejemplo, con cuatro pares de datos ($n = 3$), tenemos que el polinomio de interpolación de Newton es:

$$\begin{aligned} P_3(x) =& a_0 + (x - x_0)\, a_1 (x - x_0)(x - x_1)\, a_2 + \\ &+ (x - x_0)(x - x_1)(x - x_2)\, a_3 \end{aligned}$$

$$P_3(x) = a_0 + \{a_1 + (x - x_1)\,[a_2 + (x - x_2) a_3]\}$$

Evaluación del polinomio

$$P_3(x) = a_0 + \{a_1 + (x - x_1)\,[a_2 + (x - x_2)a_3]\}$$

Que puede ser evaluado hacia atrás con las siguientes relaciones de recurrencia:

$$\begin{aligned} P_0 &= a_3 \\ P_1 &= a_2 + (x - x_2)\,P_0(x) \\ P_2 &= a_1 + (x - x_1)\,P_1(x) \\ P_3 &= a_0 + (x - x_0)\,P_2(x) \end{aligned}$$

Evaluación del polinomio

Evaluación del polinomio

Para un n arbitrario, tenemos:

Evaluación del polinomio

Para un n arbitrario, tenemos:

$$\begin{aligned} P_0(x) &= a_n \\ P_k &= a_{n-k} + (x - x_{n-k})\, P_{k-1}(x), \quad k = 1, 2, \ldots, n \end{aligned} \tag{4}$$

Estructurando el código en `python`

Definimos `xDatos` para las coordenadas x del conjunto de puntos y n al grado de polinomio, podemos usar el siguiente algoritmo para calcular $P_n(x)$:

Código 2: Nombre Codigo

```
1 p = a[n]
2 for k in range(1, n + 1):
3     p = a[n-k] + (x - xDatos[n-k]) *
    p
```

Coeficientes del polinomio

Los coeficientes de P_n se calculan forzando que el polinomio pase a través del conjunto de puntos $y_i = P_n(x_i), \quad i = 0, 1, \ldots, n.$

Coeficientes del polinomio

De tal manera que tenemos un sistema de ecuaciones simultáneas:

$$
\begin{aligned}
y_0 &= a_0 \\
y_1 &= a_0 + (x_1 - x_0)\, a_1 \\
y_2 &= a_0 + (x_2 - x_0)\, a_1 + (x_2 - x_0)(x_2 - x_1)\, a_2 \\
&\vdots \\
y_n &= a_0 + (x_n - x_0)\, a_1 + \ldots + \\
&\quad + (x_n - x_0)(x_n - x_1) \ldots (x_n - x_{n-1})\, a_n
\end{aligned}
$$

Diferencias divididas

Se introducen las diferencias divididas, de la siguiente forma:

$$\nabla y_i = \frac{y_i - y_0}{x_i - x_0}, \qquad i = 1, 2, \ldots, n$$

Diferencias divididas

Se introducen las diferencias divididas, de la siguiente forma:

$$\nabla y_i = \frac{y_i - y_0}{x_i - x_0}, \qquad i = 1, 2, \ldots, n$$

$$\nabla^2 y_i = \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \qquad i = 1, 2, \ldots, n$$

Diferencias divididas

Se introducen las diferencias divididas, de la siguiente forma:

$$\nabla y_i = \frac{y_i - y_0}{x_i - x_0}, \qquad i = 1, 2, \ldots, n$$

$$\nabla^2 y_i = \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \qquad i = 1, 2, \ldots, n$$

$$\nabla^3 y_i = \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2}, \qquad i = 1, 2, \ldots, n$$

Diferencias divididas

Se introducen las diferencias divididas, de la siguiente forma:

$$\begin{aligned}
\nabla y_i &= \frac{y_i - y_0}{x_i - x_0}, \qquad i = 1, 2, \ldots, n \\
\nabla^2 y_i &= \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \qquad i = 1, 2, \ldots, n \\
\nabla^3 y_i &= \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2}, \qquad i = 1, 2, \ldots, n \\
&\vdots
\end{aligned}$$

Diferencias divididas

Se introducen las diferencias divididas, de la siguiente forma:

$$
\begin{aligned}
\nabla y_i &= \frac{y_i - y_0}{x_i - x_0}, \qquad i = 1, 2, \ldots, n \\
\nabla^2 y_i &= \frac{\nabla y_i - \nabla y_1}{x_i - x_1}, \qquad i = 1, 2, \ldots, n \\
\nabla^3 y_i &= \frac{\nabla^2 y_i - \nabla^2 y_2}{x_i - x_2}, \qquad i = 1, 2, \ldots, n \\
&\vdots \\
\nabla^n y_i &= \frac{\nabla^{n-1} y_n - \nabla^{n-1} y_{n-1}}{x_n - x_{n-1}}
\end{aligned}
$$

Diferencias divididas

La solución al sistema de ecuaciones es:

$$\begin{aligned} a_0 &= y_0 \\ a_1 &= \nabla y_1 \\ a_2 &= \nabla^2 y_2 \\ &\vdots \\ a_n &= \nabla^n y_n \end{aligned}$$

Diferencias divididas

Si los coeficientes se calculan a mano, es conveniente escribirlos con el siguiente formato: (con $n = 4$)

x_0	y_0				
x_1	y_1	∇y_1			
x_2	y_2	∇y_2	$\nabla^2 y_2$		
x_3	y_3	∇y_3	$\nabla^2 y_3$	$\nabla^3 y_3$	
x_4	y_4	∇y_4	$\nabla^2 y_4$	$\nabla^3 y_4$	$\nabla^4 y_4$

Diferencias divididas

Los términos en la diagonal

$$(y_0, \nabla y_1, \nabla^2 y_2, \nabla^3 y_3, \nabla^4 y_4)$$

Diferencias divididas

Los términos en la diagonal

$$(y_0, \nabla y_1, \nabla^2 y_2, \nabla^3 y_3, \nabla^4 y_4)$$

son los coeficientes del polinomio.

Diferencias divididas

Si los puntos de datos se enumeran en un orden diferente, las entradas de la tabla van a cambiar, pero el polinomio resultante será el mismo.

Recordemos que un polinomio de interpolación de grado n con $n + 1$ datos diferentes, es único.

Estructurando el código

Las operaciones en la computadora se pueden realizar con un arreglo unidimensional a, usando el siguiente algoritmo (tomando la notación $m = n + 1$ = número de puntos):

Estructurando el código

Código 3: Calculando los coeficientes del polinomio

```
1 a = yDatos.copy()
2 for k in range(1, m):
3     for i in range(k, m):
4         a[i] = (a[i] - a[k-1])/(
    xDatos[i] - xDatos[k-1])
```

Estructurando el código

Inicialmente el arreglo a contiene las coordenadas y del conjunto de datos, es decir, la segunda columna de la tabla.

Estructurando el código

Cada vez que pasa por el bucle externo, se genera la siguiente columna, por lo que se sobre-escriben los elementos de a, por tanto, al concluir el bucle, a contiene los elementos de la diagonal, que son los coeficientes del polinomio.

Módulos propios de interpolación con `python`

Para facilitar el mantenimiento y la lectura cuando los programas son demasiado largos, es posible “dividirlos” en módulos, agrupando elementos relacionados.

Módulo `newtonPoli`

El módulo **`newtonPoli`** incluye dos funciones que se requieren para la interpolación de Newton.

1. La función **`coeffts`**.

Módulo `newtonPoli`

El módulo **`newtonPoli`** incluye dos funciones que se requieren para la interpolación de Newton.

1. La función **`coeffts`**.
2. La función **`evalPoli`**

La función `coeffts`

Dados el conjuntos de puntos en los arreglos `xDatos` y `yDatos`, la función **`coeffts`** devuelve el arreglo a con los coeficientes.

La función `evalPoli`

Una vez que ya conocemos los coeficientes, el polinomio $P_n(x)$ puede evaluarse para cualquier valor de x con la función `evalPoli`.

Código del módulo

Código 4: Funciones `coeffts` del módulo `newtonPoli`

```
1 def coeffts(xDatos, yDatos):
2     m = len(xDatos)
3     a = yDatos.copy()
4     for k in range(1, m):
5         a[k:m] = (a[k:m] - a[k-1])/(
    xDatos[k:m] - xDatos[k-1])
6     return a
```

Código del módulo

Código 5: Funciones `evalPoli` del módulo `newtonPoli`

```
1 def evalPoli(a, xDatos, x):
2     n = len(xDatos) - 1
3     p = a[n]
4     for k in range(1, n+1):
5         p = a[n-k] + (x - xDatos[n-k
    ]) * p
6     return p
```

Ejemplo

Los datos que se muestran en la siguiente tabla:

x	0.15	2.30	3.15	4.85	6.25	7.95
y	4.79867	4.49013	4.2243	3.47313	2.66674	1.51909

Se obtuvieron de la función

$$f(x) = 4.8 \cos\left(\frac{\pi\, x}{20}\right)$$

Ejemplo

Con ese conjunto de datos, interpola mediante el polinomio de Newton en

$$x = 0, 0.5, 1.0, 1.5, \ldots, 7.5, 8.0$$

Compara los resultados con el valor “exacto” de los valores $y_i = f(x_i)$

Solución al ejemplo

¿Qué necesitamos?

Abriendo un archivo en Spyder, llamamos a la librería **`numpy`** y también al módulo **`newtonPoli`** que contiene las funciones para resolver el polinomio de interpolación:

Llamada a las librerías

Código 6: Llamando a las librerías y módulos

```
1 from numpy import *
2 from newtonPoli import *
```

Definiendo los arreglos

Hay que crear los arreglos `xDatos` y `yDatos`, el arreglo a se obtiene de la función **`coeffts`**.

Código 7: Nombre Codigo

```
1 xDatos = array([0.15,2.3,...,7.95])
2
3 yDatos = array([4.79867,4.49013,...,
    1.51909])
4
5 a = coeffts(xDatos, yDatos)
```

Evaluación de los puntos

La siguiente parte es proporcionar el rango de puntos en donde queremos evaluar mediante el método de Newton, a través de la función **`evalPoli`**:

Evaluación de los puntos

Código 8: Evaluando los puntos y el error relativo

```
1 for x in arange(0.0, 8.1, 0.5):
2     y = evalPoli(a, xDatos, x)
3     yExacta = 4.8 * cos(pi * x/20.0)
4     print ('{:1.1f} \t {:1.5f} \t {:
  1.5f} \t {:1.5E}'.format(x, y,
  yExacta, errorRelativo(yExacta, y)
  ))
```

Al ejecutar el código obtenemos lo siguiente:

x	yInterp	yExacta	erroRelativo
0.0	4.80003	4.80000	$5.22802E-04$
0.5	4.78518	4.78520	$5.16392E-04$
1.0	4.74088	4.74090	$5.70846E-04$
1.5	4.66736	4.66738	$3.19661E-04$
$\vdots$			
7.0	2.17915	2.17915	$3.43797E-04$
7.5	1.83687	1.83688	$6.76648E-04$
8.0	1.48329	1.48328	$2.67576E-04$

Graficando la función exacta y los puntos

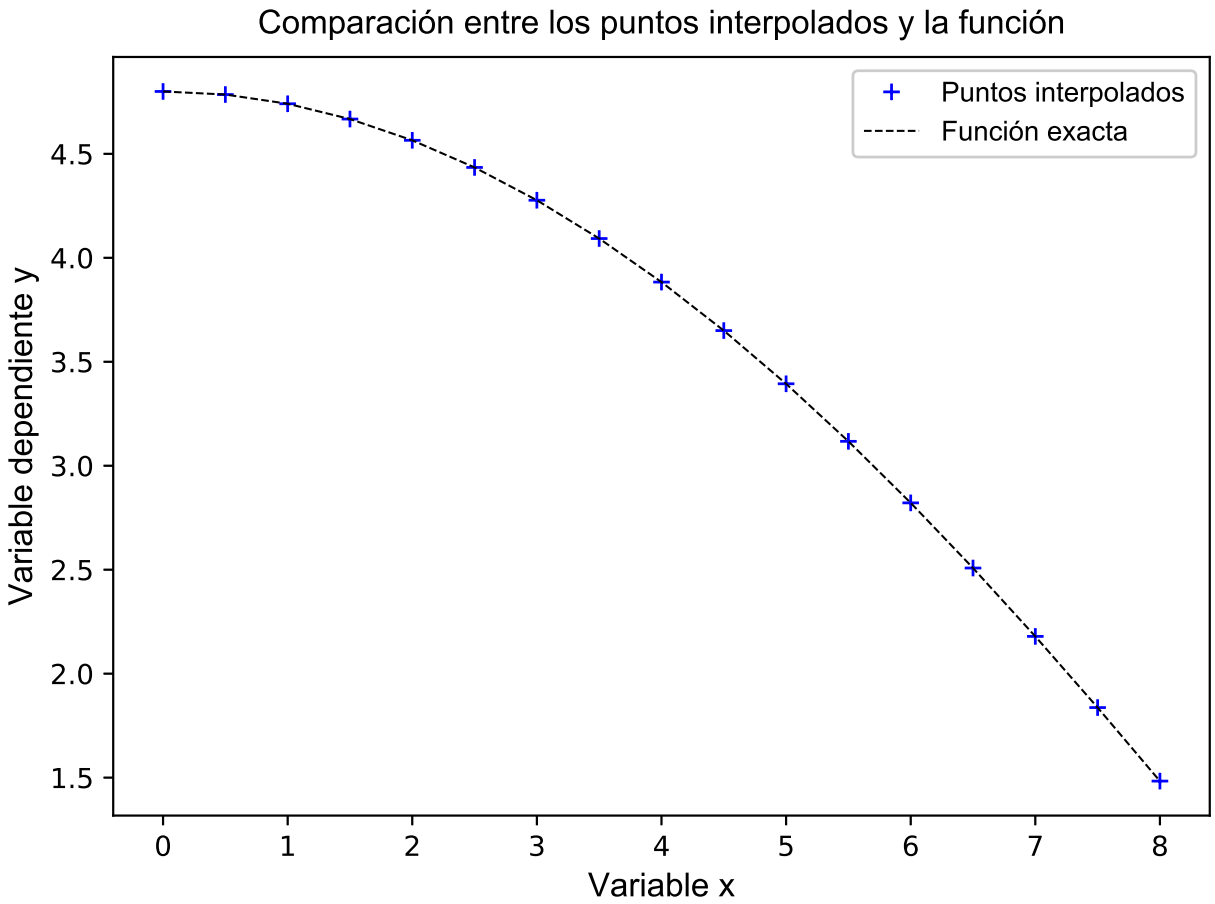


Figura 7: *Gráfica donde se visualizan los datos interpolados y la función.*