# Derivadas Numéricas

Cálculo de derivadas mediante métodos numéricos

**Horacio Useche Losada**
Google Software Developer
horaciouseche@gmail.com

Enero de 2019

Diagramación en LaTeX realizada por el autor bajo Linux Manjaro

# Índice

# Índice de cuadros

# 1. Introducción

La derivada de una función $f(x)$ en el punto $x$ se define mediante la expresión:

$$f'(x) = \lim_{x \to \infty} \frac{f(x+h) - f(x)}{h}$$

sin embargo, podemos aproximar este valor mediante

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

donde, para aproximar la derivada numérica en un punto $x$ debemos tomar una sucesión de puntos $\{h_k\}$ tal que $h_k \to 0$ y calcular el cociente

$$D_k = \frac{f(x+h_k) - f(x)}{h_k} \quad k = 1, 2, \ldots, n.$$

De la sucesión $\{D_1, D_2, \ldots\}$, solo una de las $D_k$ aproximará bien la derivada y por tanto debemos hallar el valor $h_k$ que nos entrega la mejor opción. Las fórmulas para la primera derivada y las derivadas de orden superior, se consiguen mediante el método de coeficientes indeterminados. No vamos a detallar este procedimiento aquí, para lo cual, el lector puede consultar [1] y muchos sitios web.

La idea de este trabajo es presentar el software que nos permite calcular rápida y numéricamente valores de $f', f'', f'''$ y $f^{IV}$ en los puntos donde se les requiera, sobre todo pensando en la estimación del error en problemas que involucran ecuaciones diferenciales ordinarias y ecuaciones en derivadas parciales. El cálculo de estos valores mediante métodos numéricos es de gran ayuda en la resolución de estos problemas, pues ahorra mucho tiempo.

Las rutinas presentadas han sido escritas en lenguaje Go de Google Inc., siguiendo nuestra política de usufructuar al máximo el "C del siglo XXI", que es una herramienta muy rápida, cómoda y con la exactitud suficiente para las aplicaciones propuestas.

Esperamos que este estudio sea de utilidad tanto para matemáticos profesionales como científicos de otrás áreas e ingenieros que requieran calcular el error en sus ecuaciones o las ratas de cambio asociadas con todo un potencial de aplicaciones físicas.

# 2. Derivadas de primer orden

## 2.1. Primer orden con error $O(h^2)$

Dada una función continua[1] $f(x)$, la derivada numérica de primer orden $f'(x)$, calculada en $x$, se obtiene mediante la expresión:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{1}$$

donde empezamos calculando 1 con un valor de $h$ elegido arbitrariamente[2], luego se procede a evaluar 1 con $h$ reducido a la mitad, esto es, $h/2$ y continuamos en cada iteración reduciendo el valor delta a la mitad, luego continuamos con $h/4$, y así sucesivamente.

El error es estimado a través de la expresión:

$$E = O(h^2) = \frac{h^2}{6} f''(\epsilon)$$

### 2.1.1. Software

Para poner en práctica estos conceptos se he escrito el siguiente código Go:

```
package main
import(
"fmt"
"time"
"math"
)
func EvalueFunction(iFunction int,x float64)float64{
 // evalua la funcion de iFunction en X
 var y float64=0
 switch(iFunction){
  case 0:
   sFunction="e^x"
   y=math.Exp(x)
  case 1:
```

[1] Al menos supuestamente continua

[2] De ordinario, un valor de 0.1 es bueno para iniciar.

```
   sFunction="x^3+2"
   y=x*x*x+2
  case 2:
   sFunction="sin x"
   y=math.Sin(x)
 }
 return y
}
//-----------------------------------------------------------
func GetErrorOrden(iOrden int,h float64)float64{
 // evalua el error aproximado
 var y float64=0
 switch(iOrden){
  case 1:
   y=(h*h)/6.0
  case 2:
    y=(h*h*h*h)/30.0
 }
 return y
}
//-------------------------------------------------------------
func CalcDerivada_1(iFunction,iIters int,x,h float64)float64{
 // calcula la derivada en x con incremento h
 var f float64=0
 for i:=0;i<iIters;i++{
  y1:=EvalueFunction(iFunction,x+h)
  y2:=EvalueFunction(iFunction,x-h)
  err:=GetErrorOrden(1,h)
  f=(y1-y2)/(2*h)
  fmt.Printf("h: %.6f f:%.9f err: %.9f\n",h,f,err)
  h/=2.0
 }
 return f
}
//--------------------------------------------------------------
func main() {
 t:= time.Now()
 fmt.Printf("=========DERIVACIÓN NUMÉRICA ===========\n")
```

```
    fmt.Printf("Tiempo antes de ejecutar: %v \n",time.Now())
    iFunction:=0
    x:=0.5
    step:=0.1
    iIters:=10
    CalcDerivada_1(iFunction,iIters,x,step)
    t2 := time.Now()
    fmt.Printf("Begin Time: %d : %d : %d\n",
                                 t.Hour(),t.Minute(),t.Second())
    fmt.Printf("End Time  : %d : %d : %d\n",
                               t2.Hour(),t2.Minute(),t2.Second())
}
```

La función que calcula la derivada es `CalcDerivada_1()` y toma cuatro parámetros con el siguiente significado:

1. `iFunction`, de tipo entero, se refiere al caso de función a evaluar en la estructura `switch`.
2. `iIters`, de tipo entero, se refiere al número de iteraciones a realizar.
3. `x`, de tipo como flotante (`float64`), se refiere al valor puntual donde se evalúa la derivada.
4. `h`, de tipo como flotante (`float64`), se refiere al valor del delta de incremento o paso, para realizar la evaluación.

Ahora bien, esta función cita a la función `EvalueFunction()`, la cual toma dos argumentos:

1. `iFunction`, de tipo entero, se refiere al caso de función a evaluar en la estructura `switch`.
2. `x`, de tipo como flotante (`float64`), se refiere al valor puntual donde se evalúa la derivada.

Y por último está la función `GetErrorOrden()`, la cual toma dos argumentos:

1. `iOrden`, de tipo entero, se refiere al orden de la derivada, ya que se consideran derivadas de orden superior.

2. `h`, de tipo como flotante (`float64`), se refiere al valor del delta de incremento o paso, para realizar la evaluación.

Las estructuras `switch` irán creciendo a medida que consideremos más funciones a evaluar.

## 2.2. Primer orden con error $O(h^4)$

Dada una función continua[3] $f(x)$, la derivada numérica de primer orden $f'(x)$, calculada en $x$, con error del orden $O(h^4)$ se obtiene mediante la expresión:

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + O(h^4) \qquad (2)$$

donde empezamos calculando 2 con un valor de $h$ elegido arbitrariamente[4], luego se procede a evaluar 2 con $h$ reducido a la mitad, esto es, $h/2$ y continuamos en cada iteración reduciendo el valor delta a la mitad, luego continuamos con $h/4$, y así sucesivamente.

El error cometido en este caso viene dado por la expresión:

$$E = O(h^4) = \frac{h^4}{30} f''(\epsilon)$$

sin embargo, el error se aproxima por el orden de magnitud, esto es, $O(h^4) \approx \frac{h^4}{30}$.

### 2.2.1. Software

Para ejecutar la expresión 2, se ha escrito la siguiente rutina:

```
func CalcDerivada_1B(iFunction,iIters int,x,h float64)float64{
 // calcula la primera derivada en x con error O(h^4)
 var f float64=0
 for i:=0;i<iIters;i++{
   y1:=EvalueFunction(iFunction,x+h)
   y2:=EvalueFunction(iFunction,x+2*h)
   ym1:=EvalueFunction(iFunction,x-h)
```

[3] Al menos supuestamente continua

[4] De ordinario, un valor de 0.1 es bueno para iniciar.

```
   ym2:=EvalueFunction(iFunction,x-2*h)
   err:=GetErrorOrden(2,h)
   f=(-y2+8*y1-8*ym1+ym2)/(12*h)
   fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
   h/=2.0
 }
 return f
}
```

sus parámetros son los mismos que para el caso de la rutina con error $O(h^2)$. Para ejecutarla, debemos modificar la cita de la función en `main()`, la cual, deberá aparecer como `CalcDerivada_1B()`, con los mismos parámetros que para el caso `CalcDerivada_1()`.

## 2.3. Ejemplo 1: Evaluar $f'$ para $f(x) = e^x$

Dada la función $f(x) = e^x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con cinco pasos (iteraciones) y por los dos métodos considerados.

Evaluamos la expresión 1 con $h = 0.1$, y $x = 0.5$ de manera que conseguimos:

$$\begin{aligned}
f'(0.5) &= \frac{f(0.5+0.1) - f(0.5-0.1)}{2(0.1)} \\
&= \frac{f(0.6) - f(0.4)}{0.2} = \frac{e^{0.6} - e^{0.4}}{0.2} \\
&= \frac{1.8221188 - 1.491824698}{0.2} \\
&= \frac{0.330294103}{0.2} \\
&= 1.651470514
\end{aligned}$$

Este valor corresponde a la derivada $f'(0.5)$ para $h = 0.1$ y aparece en la segunda columna del cuadro 1. Procediendo en idéntica forma se obtienen los demás valores del cuadro 1. De otra parte, para calcular el orden del error en este punto hacemos

$$E(0.1) = \frac{(0.1)^2}{6} = \frac{0.01}{6} = 0.001666667$$

Este procedimiento es justamente lo que se programa en Go y por consiguiente será omitido en el futuro, para ser representado por los cuadros de resultados.

En cuanto al error, debemos advertir que, lo que se calcula con las expresiones

$$E = O(h^2) = \frac{h^2}{6} f''(\epsilon)$$
$$E = O(h^4) = \frac{h^4}{30} f''(\epsilon)$$

es una aproximación del orden de magnitud del error, en lugar del error mismo. Esto nos evita tener que evaluar las derivadas $f''$ y superiores para conseguir un cálculo exacto, lo cual no será necesario cuando podemos conformarnos con una estimación aproximada del orden de magnitud del error.

### 2.3.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 1 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.651470514 | 0.001666667 |
| 2 | 0.050000 | 1.649408324 | 0.000416667 |
| 3 | 0.025000 | 1.648893018 | 0.000104167 |
| 4 | 0.012500 | 1.648764206 | 0.000026042 |
| 5 | 0.006250 | 1.648732005 | 0.000006510 |

Cuadro 1: Evaluación de $f'$ para $f(x) = e^x$ en $x = 0.5$. La solución exacta es 1.648721271. Caso $O(h^2)$.

El error conseguido es de $E = 1.648732005 - 1.648721271 = 0.000010734$. Nótese que el error suministrado para este caso no es exacto sino aproximado, en realidad lo que se entrega en la cuarta columna es información acerca del orden del error, esto para simplificar los cálculos. Un tratamiento más profundo del error en esta clase de cálculos puede ser consultado en [1].

### 2.3.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.648715768 | 0.000003333 |
| 2 | 0.050000 | 1.648720927 | 0.000000208 |
| 3 | 0.025000 | 1.648721249 | 0.000000013 |
| 4 | 0.012500 | 1.648721269 | 0.000000001 |
| 5 | 0.006250 | 1.648721271 | 0.000000000 |

Cuadro 2: Evaluación de $f'$ para $f(x) = e^x$ en $x = 0.5$. La solución exacta es 1.648721271. Caso $O(h^4)$.

El error cometido en este caso es $E = 1.648721271 - 1.648721271 = 0.0$.

## 2.4. Ejemplo 2: Evaluar $f'$ para $f(x) = e^x$ y $h$ fino

Dada la función $f(x) = e^x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con diez iteraciones y usando los dos métodos.

### 2.4.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 3 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.651470514 | 0.001666667 |
| 2 | 0.050000 | 1.649408324 | 0.000416667 |
| 3 | 0.025000 | 1.648893018 | 0.000104167 |
| 4 | 0.012500 | 1.648764206 | 0.000026042 |
| 5 | 0.006250 | 1.648732005 | 0.000006510 |
| 6 | 0.003125 | 1.648723954 | 0.000001628 |
| 7 | 0.001563 | 1.648721942 | 0.000000407 |
| 8 | 0.000781 | 1.648721438 | 0.000000102 |
| 9 | 0.000391 | 1.648721313 | 0.000000025 |
| 10 | 0.000195 | 1.648721281 | 0.000000006 |

Cuadro 3: Evaluación de $f'$ para $f(x) = e^x$ en $x = 0.5$. La solución exacta es 1.648721271. Caso $O(h^2)$.

Nótese que el error conseguido ahora es $E = 1.648721281 - 1.648721271 = 0.00000001029$, es decir, del orden de $1 \times 10^{-8}$, lo cual es muy aceptable para la mayoría de las aplicaciones científicas.

### 2.4.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.648715768 | 0.000003333 |
| 2 | 0.050000 | 1.648720927 | 0.000000208 |
| 3 | 0.025000 | 1.648721249 | 0.000000013 |
| 4 | 0.012500 | 1.648721269 | 0.000000001 |
| 5 | 0.006250 | 1.648721271 | 0.000000000 |
| 6 | 0.003125 | 1.648721271 | 0.000000000 |
| 7 | 0.001563 | 1.648721271 | 0.000000000 |
| 8 | 0.000781 | 1.648721271 | 0.000000000 |
| 9 | 0.000391 | 1.648721271 | 0.000000000 |
| 10 | 0.000195 | 1.648721271 | 0.000000000 |

Cuadro 4: Evaluación de $f'$ para $f(x) = e^x$ en $x = 0.5$. La solución exacta es 1.648721271. Caso $O(h^4)$.

El error no varia después de la 5 iteración.

## 2.5. Ejemplo 3: Evaluar $f'$ para $f(x) = x^3 + 2$

Dada la función $f(x) = x^3+2$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con cinco valores para el delta (iteraciones) y usando los dos métodos.

### 2.5.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 5 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.760000000 | 0.001666667 |
| 2 | 0.050000 | 0.752500000 | 0.000416667 |
| 3 | 0.025000 | 0.750625000 | 0.000104167 |
| 4 | 0.012500 | 0.750156250 | 0.000026042 |
| 5 | 0.006250 | 0.750039063 | 0.000006510 |

Cuadro 5: Evaluación de $f'$ para $f(x) = x^3 + 2$ en $x = 0.5$. La solución exacta es 0.75. Caso $O(h^2)$.

El error conseguido es de $E = 0.750039063 - 0.75 = 0.00003906$, del orden de $3 \times 10^{-5}$.

### 2.5.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.750000000 | 0.000003333 |
| 2 | 0.050000 | 0.750000000 | 0.000000208 |
| 3 | 0.025000 | 0.750000000 | 0.000000013 |
| 4 | 0.012500 | 0.750000000 | 0.000000001 |
| 5 | 0.006250 | 0.750000000 | 0.000000000 |

Cuadro 6: Evaluación de $f'$ para $f(x) = x^3 + 2$ en $x = 0.5$. La solución exacta es 0.75. Caso $O(h^4)$.

El error cometido es $E = 0.750000000 - 0.75 = 0.0$.

## 2.6. Ejemplo 4: Evaluar $f'$ para $f(x) = \sin x$

Dada la función $f(x) = \sin x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con diez valores para el delta (iteraciones) y usando los dos métodos.

### 2.6.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 7 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.876120655 | 0.001666667 |
| 2 | 0.050000 | 0.877216948 | 0.000416667 |
| 3 | 0.025000 | 0.877491150 | 0.000104167 |
| 4 | 0.012500 | 0.877559708 | 0.000026042 |
| 5 | 0.006250 | 0.877576848 | 0.000006510 |
| 6 | 0.003125 | 0.877581134 | 0.000001628 |
| 7 | 0.001563 | 0.877582205 | 0.000000407 |
| 8 | 0.000781 | 0.877582473 | 0.000000102 |
| 9 | 0.000391 | 0.877582540 | 0.000000025 |
| 10 | 0.000195 | 0.877582556 | 0.000000006 |

Cuadro 7: Evaluación de $f'$ para $f(x) = \sin x$ en $x = 0.5$. La solución exacta es 0.877582562. Caso $O(h^2)$.

El error conseguido es de $E = 0.877582556 - 0.877582562 = -0.00000000589$, del orden de $5 \times 10^{-9}$.

### 2.6.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.877579640 | 0.000003333 |
| 2 | 0.050000 | 0.877582379 | 0.000000208 |
| 3 | 0.025000 | 0.877582550 | 0.000000013 |
| 4 | 0.012500 | 0.877582561 | 0.000000001 |
| 5 | 0.006250 | 0.877582562 | 0.000000000 |
| 6 | 0.003125 | 0.877582562 | 0.000000000 |
| 7 | 0.001563 | 0.877582562 | 0.000000000 |
| 8 | 0.000781 | 0.877582562 | 0.000000000 |
| 9 | 0.000391 | 0.877582562 | 0.000000000 |
| 10 | 0.000195 | 0.877582562 | 0.000000000 |

Cuadro 8: Evaluación de $f'$ para $f(x) = \sin x$ en $x = 0.5$. La solución exacta es 0.877582562. Caso $O(h^4)$.

## 2.7. Ejemplo 5: Evaluar $f'$ para $f(x) = \ln x$

Dada la función $f(x) = \ln x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con cinco valores para el delta y usando los dos métodos.

### 2.7.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 9 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 2.027325541 | 0.001666667 |
| 2 | 0.050000 | 2.006706955 | 0.000416667 |
| 3 | 0.025000 | 2.001669171 | 0.000104167 |
| 4 | 0.012500 | 2.000416823 | 0.000026042 |
| 5 | 0.006250 | 2.000104176 | 0.000006510 |

Cuadro 9: Evaluación de $f'$ para $f(x) = \ln x$ en $x = 0.5$. La solución exacta es 2.0. Caso $O(h^2)$.

El error conseguido es de $E = 2.000104176 - 2.0 = 0.000104176$, del orden de $1 \times 10^{-4}$.

### 2.7.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.997019170 | 0.000003333 |
| 2 | 0.050000 | 1.999834093 | 0.000000208 |
| 3 | 0.025000 | 1.999989910 | 0.000000013 |
| 4 | 0.012500 | 1.999999374 | 0.000000001 |
| 5 | 0.006250 | 1.999999961 | 0.000000000 |

Cuadro 10: Evaluación de $f'$ para $f(x) = \ln x$ en $x = 0.5$. La solución exacta es 2.0. Caso $O(h^4)$.

El error cometido es de $E = 1.999999961 - 2.0 = 0.000000039$, es decir del orden de $3.9 \times 10^{-8}$.

## 2.8. Ejemplo 6: Evaluar $f'$ para $f(x) = e^{-x^2} \ln x$

Dada la función $f(x) = e^{-x^2} \ln x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con 5 valores para el delta y usando los dos métodos.

### 2.8.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 11 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 2.122102558 | 0.001666667 |
| 2 | 0.050000 | 2.103477371 | 0.000416667 |
| 3 | 0.025000 | 2.098931080 | 0.000104167 |
| 4 | 0.012500 | 2.097801178 | 0.000026042 |
| 5 | 0.006250 | 2.097519117 | 0.000006510 |

Cuadro 11: Evaluación de $f'$ para $f(x) = e^{-x^2} \ln x$ en $x = 0.5$. La solución exacta es 2.097425133. Caso $O(h^2)$.

El error conseguido es de $E = 2.097519117 - 2.097425133 = 0.000093984$, del orden de $9.3 \times 10^{-5}$.

### 2.8.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 2.094603571 | 0.000003333 |
| 2 | 0.050000 | 2.097268975 | 0.000000208 |
| 3 | 0.025000 | 2.097415650 | 0.000000013 |
| 4 | 0.012500 | 2.097424545 | 0.000000001 |
| 5 | 0.006250 | 2.097425096 | 0.000000000 |

Cuadro 12: Evaluación de $f'$ para $f(x) = e^{-x^2} \ln x$ en $x = 0.5$. La solución exacta es 2.097425133. Caso $O(h^4)$.

Donde el error cometido es $E = 2.097425096 - 2.097425133 = 0.000000037$.

## 2.9. Ejemplo 7: Evaluar $f'$ para $f(x) = \sin x \cos x$

Dada la función $f(x) = \sin x \cos x$, se pide evaluar $f'(x)$ en $x = 0.5$ empezando con $h = 0.1$ con 5 valores para el delta y usando los dos métodos.

### 2.9.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 13 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.536707488 | 0.001666667 |
| 2 | 0.050000 | 0.539402252 | 0.000416667 |
| 3 | 0.025000 | 0.540077208 | 0.000104167 |
| 4 | 0.012500 | 0.540246026 | 0.000026042 |
| 5 | 0.006250 | 0.540288236 | 0.000006510 |

Cuadro 13: Evaluación de $f'$ para $f(x) = \sin x \cos x$ en $x = 0.5$. La solución exacta es 0.540302306. Caso $O(h^2)$.

El error conseguido es de $E = 0.540288236 - 0.540302306 = 0.00001407$, del orden de $1.4 \times 10^{-5}$.

### 2.9.2. caso $O(h^4)$

| **No** | $h$ | $f'(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.540273627 | 0.000003333 |
| 2 | 0.050000 | 0.540300507 | 0.000000208 |
| 3 | 0.025000 | 0.540302193 | 0.000000013 |
| 4 | 0.012500 | 0.540302299 | 0.000000001 |
| 5 | 0.006250 | 0.540302305 | 0.000000000 |

Cuadro 14: Evaluación de $f'$ para $f(x) = \sin x \cos x$ en $x = 0.5$. La solución exacta es 0.540302306. Caso $O(h^4)$.

El error es $E = 0.540302305 - 0.540302306 = 0.000000001$, del orden de $1 \times 10^{-9}$.

# 3. Derivadas de segundo orden

## 3.1. Segundo orden con error $O(h^2)$

Dada una función continua[5] $f(x)$, la derivada numérica de segundo orden $f''(x)$, calculada en $x$, se obtiene mediante la expresión:

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

donde empezamos calculando 3 con un valor de $h$ elegido arbitrariamente[6], luego se procede a evaluar 3 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer orden.

### 3.1.1. Software

Para evaluar $f''(x)$ agregamos al código escrito la siguiente función:

```
func CalcDerivada_2(iFunction,iIters int,x,h float64)float64{
 // calcula la segunda derivada en x con incremento h
 var f float64=0
 for i:=0;i<iIters;i++{
   y1:=EvalueFunction(iFunction,x+h)
   y2:=EvalueFunction(iFunction,x-h)
   y:=EvalueFunction(iFunction,x)
   err:=GetErrorOrden(2,h)
   f=(y1-2*y+y2)/(h*h)
   fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
   h/=2.0
}
return f
}
```

la anterior función se encarga de materializar la fórmula dada en 3. El resto del código no sufre cambios, con excepción de la cita a la función `CalcDerivada_2()` en la función `main()`, que debe sustituir a la función usada para las derivadas de primer orden.

[5] Al menos supuestamente continua

[6] De ordinario, un valor de 0.1 es bueno para iniciar.

El error en este caso viene dado por una expresión de la forma:

$$E = \frac{h^2}{3!} f'''(\epsilon)$$

en la cual hacemos $f'''(\epsilon) = 1$, para simplificar los cálculos con el entendido de que nos interesa conocer es el orden del error más que el error mismo con exactitud.

## 3.2. Segundo orden con error $O(h^4)$

Dada una función continua[7] $f(x)$, la derivada numérica de segundo orden $f''(x)$, calculada en $x$, con un error del orden de $O(h^4)$ se obtiene mediante la expresión:

$$f''(x) = \frac{-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)}{12h^2} + O(h^4) \quad (4)$$

donde empezamos calculando 4 con un valor de $h$ elegido arbitrariamente[8], luego se procede a evaluar 4 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer orden.

De otra parte, el error cometido en este caso viene dado, por

$$E = O(h^4) = \frac{h^4}{30} f'''(\epsilon)$$

el cual, de nuevo, aproximamos al valor $E = O(h^4) \approx \frac{h^4}{30}$, por cuanto solo nos interesa el orden de magnitud del error.

El error cometido con la expresión 4, del tipo $O(h^4)$, es sustancialmente menor que el error cometido con la expresión 3, que es del tipo $O(h^2)$.

La idea de despreciar el efecto de la tercera derivada en la expresión anterior y asumirla con un valor unitario, es simplificar los cálculos. De lo contrario, nos veríamos obligados a calcular $f'''$ lo cual podría ser muy dispendiosos dependiendo de la forma de $f(x)$.

[7] Al menos supuestamente continua

[8] De ordinario, un valor de 0.1 es bueno para iniciar.

### 3.2.1. Software

La rutina implementada para calcular $f''$ con error de $O(h^4)$ es la siguiente:

```
func CalcDerivada_2B(iFunction,iIters int,x,h float64)float64{
 // calcula la segunda derivada en x con O(h^4)
 var f float64=0
 for i:=0;i<iIters;i++{
   y0:=EvalueFunction(iFunction,x)
   y1:=EvalueFunction(iFunction,x+h)
   y2:=EvalueFunction(iFunction,x+2*h)
   ym1:=EvalueFunction(iFunction,x-h)
   ym2:=EvalueFunction(iFunction,x-2*h)
   err:=GetErrorOrden(2,h)
   f=(-y2+16*y1-30*y0+16*ym1-ym2)/(12*h*h)
   fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
   h/=2.0
 }
 return f
}
```

Una salida típicas de esta rutina sería:

```
====================DERIVACIÓN NUMÉRICA =====================
Tiempo antes de ejecutar: 2019-01-28 11:52:00.891615637 -0500
1 & 0.100000 & 3.669292587 & 0.000003333\\ \hline
2 & 0.050000 & 3.669296413 & 0.000000208\\ \hline
3 & 0.025000 & 3.669296652 & 0.000000013\\ \hline
4 & 0.012500 & 3.669296667 & 0.000000001\\ \hline
5 & 0.006250 & 3.669296668 & 0.000000000\\ \hline
6 & 0.003125 & 3.669296668 & 0.000000000\\ \hline
7 & 0.001563 & 3.669296668 & 0.000000000\\ \hline
8 & 0.000781 & 3.669296667 & 0.000000000\\ \hline
Begin Time: 11 : 52 : 0
End Time  : 11 : 52 : 0
```

## 3.3. Ejemplo 8: Evaluar $f''$ para $f(x) = e^x$

Dada la función $f(x) = e^x$, se pide evaluar $f''$ en $x = 1.3$ empezando con $h = 0.1$ con 5 valores para el delta.

### 3.3.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 15 con los resultados de la evaluación pedida.

| **No** | $h$ | $f''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 3.672355434 | 0.000041667 |
| 2 | 0.050000 | 3.670061168 | 0.000005208 |
| 3 | 0.025000 | 3.669487781 | 0.000000651 |
| 4 | 0.012500 | 3.669344445 | 0.000000081 |
| 5 | 0.006250 | 3.669308612 | 0.000000010 |

Cuadro 15: Evaluación de $f''$ para $f(x) = e^x$ en $x = 1.3$. La solución exacta es 3.669296668. Caso $O(h^2)$.

El error conseguido es de $E = 3.669308612 - 3.669296668 = 0.000011944$, del orden de $1.1 \times 10^{-5}$.

### 3.3.2. caso $O(h^4)$

| **No** | $h$ | $f''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 3.669292587 | 0.000003333 |
| 2 | 0.050000 | 3.669296413 | 0.000000208 |
| 3 | 0.025000 | 3.669296652 | 0.000000013 |
| 4 | 0.012500 | 3.669296667 | 0.000000001 |
| 5 | 0.006250 | 3.669296668 | 0.000000000 |

Cuadro 16: Evaluación de $f''$ para $f(x) = e^x$ en $x = 1.3$. La solución exacta es 3.669296668. Caso $O(h^4)$.

El error cometido es $E = 3.669296668 - 3.669296668 = 0.0$ exactamente.

## 3.4. Ejemplo 9: Evaluar $f''$ para $f(x) = x^3 + 2$

Dada la función $f(x) = x^3 + 2$, se pide evaluar $f''$ en $x = 1.3$ empezando con $h = 0.1$ con cinco valores para el delta.

### 3.4.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 15 con los resultados de la evaluación pedida.

| **No** | $h$ | $f''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 7.800000000 | 0.000041667 |
| 2 | 0.050000 | 7.800000000 | 0.000005208 |
| 3 | 0.025000 | 7.800000000 | 0.000000651 |
| 4 | 0.012500 | 7.800000000 | 0.000000081 |
| 5 | 0.006250 | 7.800000000 | 0.000000010 |

Cuadro 17: Evaluación de $f''$ para $f(x) = x^3 + 2$ en $x = 1.3$. La solución exacta es 7.8. Caso $O(h^2)$

.

El error conseguido es de $E = 7.800000000 - 7.8 = -0.0$, operación exacta. En efecto, observando el cuadro 17 notamos que el valor obtenido es exacto, en todos los casos, a pesar de que la columna del error marca valores no nulos.

### 3.4.2. caso $O(h^4)$

| **No** | $h$ | $f''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 7.800000000 | 0.000003333 |
| 2 | 0.050000 | 7.800000000 | 0.000000208 |
| 3 | 0.025000 | 7.800000000 | 0.000000013 |
| 4 | 0.012500 | 7.800000000 | 0.000000001 |
| 5 | 0.006250 | 7.800000000 | 0.000000000 |

Cuadro 18: Evaluación de $f''$ para $f(x) = x^3 + 2$ en $x = 1.3$. La solución exacta es 7.8. Caso $O(h^4)$

.

## 3.5. Ejemplo 10: Evaluar $f''$ para $f(x) = e^{-x^2} \ln x$

Dada la función $f(x) = e^{-x^2} \ln x$, se pide evaluar $f''$ en $x = 1.75$ empezando con $h = 0.1$ con 5 valores para el delta.

### 3.5.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 19 con los resultados de la evaluación pedida.

| **No** | $h$ | $f''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.063289608 | 0.001666667 |
| 2 | 0.050000 | 0.065266555 | 0.000416667 |
| 3 | 0.025000 | 0.065760170 | 0.000104167 |
| 4 | 0.012500 | 0.065883534 | 0.000026042 |
| 5 | 0.006250 | 0.065914373 | 0.000006510 |

Cuadro 19: Evaluación de $f''$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.75$. La solución exacta se desconoce. Caso $O(h^2)$.

El error conseguido es del orden de $1 \times 10^{-8}$. No obstante, debemos "confiar"en que este orden esta cerca al error verdadero, para no tener que calcular la segunda derivada de la función. En lo sucesivo prescindimos de la solución exacta y confiamos en que el orden de magnitud del error es carcano al error verdadero.

### 3.5.2. caso $O(h^4)$

| **No** | $h$ | $f''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 0.065938488 | 0.000003333 |
| 2 | 0.050000 | 0.065925538 | 0.000000208 |
| 3 | 0.025000 | 0.065924708 | 0.000000013 |
| 4 | 0.012500 | 0.065924656 | 0.000000001 |
| 5 | 0.006250 | 0.065924652 | 0.000000000 |

Cuadro 20: Evaluación de $f''$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.75$. La solución exacta se desconoce, pero podría ser 0.065924652. Caso $O(h^4)$.

El error, o mejor, el orden del error es de $1 \times 10^{-9}$.

## 3.6. Ejemplo 11: Evaluar $f''$ para $f(x) = \sin x \cos x$

Dada la función $f(x) = \sin x \cos x$, se pide evaluar $f''$ en $x = 0.785$ empezando con $h = 0.1$ con 15 valores para el delta.

### 3.6.1. caso $O(h^2)$

Usando el software dado arriba se construye el cuadro 21 con los resultados de la evaluación pedida.

| **No** | $h$ | $f''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | -1.993341584 | 0.001666667 |
| 2 | 0.050000 | -1.998333255 | 0.000416667 |
| 3 | 0.025000 | -1.999582734 | 0.000104167 |
| 4 | 0.012500 | -1.999895201 | 0.000026042 |
| 5 | 0.006250 | -1.999973324 | 0.000006510 |

Cuadro 21: Evaluación de $f''$ para $f(x) = \sin x \cos x$ en $x = 0.785$, con 5 valores para $h$. Caso $O(h^2)$.

El error conseguido es del orden de $1 \times 10^{-6}$. En este ejemplo, la solución exacta parece ser -2.0 o muy cercano a ello, se arriesgaría a calcularlo con exactitud ?

### 3.6.2. caso $O(h^4)$

| **No** | $h$ | $f''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | -1.999963937 | 0.000003333 |
| 2 | 0.050000 | -1.999997146 | 0.000000208 |
| 3 | 0.025000 | -1.999999227 | 0.000000013 |
| 4 | 0.012500 | -1.999999357 | 0.000000001 |
| 5 | 0.006250 | -1.999999365 | 0.000000000 |

Cuadro 22: Evaluación de $f''$ para $f(x) = \sin x \cos x$ en $x = 0.785$, con 5 valores para $h$. Caso $O(h^4)$.

El orden de magnitud del error es de $1 \times 10^{-9}$.

# 4. Derivadas de tercer orden

## 4.1. Tercer orden con error $O(h^2)$

Dada una función continua[9] $f(x)$, la derivada numérica de tercer orden $f'''(x)$, calculada en $x$, se obtiene mediante la expresión:

$$f'''(x) = \frac{f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)}{2h^3} + O(h^2) \qquad (5)$$

donde empezamos calculando 5 con un valor de $h$ elegido arbitrariamente, luego se procede a evaluar 5 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer y segundo orden.

### 4.1.1. Software

Para evaluar $f'''(x)$ agregamos al código escrito la siguiente función:

```
func CalcDerivada_3(iFunction,iIters int,x,h float64)float64{
 // calcula la tercera derivada en x con incremento h
 var f float64=0
 for i:=0;i<iIters;i++{
  y2:=EvalueFunction(iFunction,x+2*h)
  y1:=EvalueFunction(iFunction,x+h)
  ym2:=EvalueFunction(iFunction,x-2*h)
  ym1:=EvalueFunction(iFunction,x-h)
  err:=GetErrorOrden(3,h)
  f=(y2-2*y1+2*ym1-ym2)/(2*h*h*h)
  fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
  h/=2.0
 }
 return f
}
```

esta función se encarga de materializar la fórmula dada en 5. El resto del código no sufre cambios, con excepción de la cita a la función `CalcDerivada_3()` en la función `main()`, que debe sustituir a la función usada para las derivadas de segundo orden.

[9] Al menos supuestamente continua

El error en este caso viene dado por una expresión de la forma:

$$E = \frac{h^2}{3!} f^{IV}(\epsilon)$$

en la cual hacemos $f^{IV}(\epsilon) = 1$, para simplificar los cálculos con el entendido de que nos interesa conocer es el orden del error más que el error mismo.

## 4.2. Tercer orden con error $O(h^4)$

Dada una función continua[10] $f(x)$, la derivada numérica de tercer orden $f'''(x)$, calculada en $x$, con error de orden $O(h^4)$ se obtiene mediante la expresión:

$$f'''(x) = \frac{-f(x+3h) + 8f(x+2h) - 13f(x+h)}{8h^3} + \frac{13f(x-h) - 8f(x-2h) + f(x-3h)}{8h^3} + O(h^4) \quad (6)$$

donde empezamos calculando 6 con un valor de $h$ elegido arbitrariamente, luego se procede a evaluar 6 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer y segundo orden.

Asimismo, el error cometido en este caso viene dado, por

$$E = O(h^4) = \frac{h^4}{30} f^{IV}(\epsilon)$$

donde, una vez más, hacemos $f^{IV}(\epsilon) = 1$, con el fin de simplificar los cálculos y considerar únicamente el orden de magnitud del error.

### 4.2.1. Software

Para evaluar la expresión 6 se ha escrito la rutina:

```
func CalcDerivada_3B(iFunction,iIters int,x,h float64)float64{
 // calcula la tercera derivada en x con O(h^4)
 var f float64=0
 for i:=0;i<iIters;i++{
   y3:=EvalueFunction(iFunction,x+3*h)
```

[10] Al menos supuestamente continua

```
    y2:=EvalueFunction(iFunction,x+2*h)
    y1:=EvalueFunction(iFunction,x+h)
    ym3:=EvalueFunction(iFunction,x-3*h)
    ym2:=EvalueFunction(iFunction,x-2*h)
    ym1:=EvalueFunction(iFunction,x-h)
    err:=GetErrorOrden(2,h)
    f=(-y3+8*y2-13*y1+13*ym1-8*ym2+ym3)/(8*h*h*h)
    fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
    h/=2.0
  }
 return f
}
```

Una ejecución típica de la rutina `CalcDerivada_3B()` se puede apreciar a continuación:

```
===================DERIVACIÓN NUMÉRICA ======================
Tiempo antes de ejecutar: 2019-01-28 14:25:23.610652489 -0500
I: 0 h: 0.100000 f:2.192394128 err: 0.000003333
I: 1 h: 0.050000 f:2.192406141 err: 0.000000208
I: 2 h: 0.025000 f:2.192406891 err: 0.000000013
I: 3 h: 0.012500 f:2.192406938 err: 0.000000001
I: 4 h: 0.006250 f:2.192406942 err: 0.000000000
I: 5 h: 0.003125 f:2.192406910 err: 0.000000000
Begin Time: 14 : 25 : 23
End Time  : 14 : 25 : 23
```

En la cual hemos usado en la función `main()` una cita para la función `CalcDerivada_3B()`, sustituyendo a su similar del orden $O(h^2)$. La cita puede verse como sigue:

```
iFunction:=0
x:=0.785
step:=0.1
iIters:=6
CalcDerivada_3B(iFunction,iIters,x,step)
```

Justamente se usó para producir la salida de arriba.

## 4.3. Ejemplo 12: Evaluar $f'''$ para $f(x) = e^x$

Dada la función $f(x) = e^x$, se pide evaluar $f'''$ en $x = 1.5$ empezando con $h = 0.1$ con 5 valores para $h$.

### 4.3.1. caso $O(h^2)$

Como de costumbre, construimos el cuadro 23 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 4.492904504 | 0.001666667 |
| 2 | 0.050000 | 4.484490826 | 0.000416667 |
| 3 | 0.025000 | 4.482389378 | 0.000104167 |
| 4 | 0.012500 | 4.481864139 | 0.000026042 |
| 5 | 0.006250 | 4.481732834 | 0.000006510 |

Cuadro 23: Evaluación de $f'''$ para $f(x) = e^x$ en $x = 1.5$, con 8 valores para $h$. La solución exacta es $f''' = 4.48168907$. Caso $O(h^2)$.

El error verdadero es $E = 4.481732834 - 4.48168907 = 0.00004376$, es decir, del orden de $4.3 \times 10^{-5}$.

### 4.3.2. caso $O(h^4)$

| **No** | $h$ | $f'''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 4.481662879 | 0.000003333 |
| 2 | 0.050000 | 4.481687436 | 0.000000208 |
| 3 | 0.025000 | 4.481688968 | 0.000000013 |
| 4 | 0.012500 | 4.481689063 | 0.000000001 |
| 5 | 0.006250 | 4.481689065 | 0.000000000 |

Cuadro 24: Evaluación de $f'''$ para $f(x) = e^x$ en $x = 1.5$, con 8 valores para $h$. La solución exacta es $f''' = 4.48168907$. Caso $O(h^4)$.

El error cometido es $E = 4.481689065 - 4.48168907 = 0.000000005338$, es del orden de $5.3 \times 10^{-9}$.

## 4.4. Ejemplo 13: Evaluar $f'''$ para $f(x) = x^5 + 5$

Dada la función $f(x) = e^x$, se pide evaluar $f'''$ en $x = 0.3$ empezando con $h = 0.1$ con 5 valores para $h$.

### 4.4.1. caso $O(h^2)$

Construimos el cuadro 25 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 5.700000000 | 0.001666667 |
| 2 | 0.050000 | 5.475000000 | 0.000416667 |
| 3 | 0.025000 | 5.418750000 | 0.000104167 |
| 4 | 0.012500 | 5.404687500 | 0.000026042 |
| 5 | 0.006250 | 5.401171877 | 0.000006510 |

Cuadro 25: Evaluación de $f'''$ para $f(x) = x^5 + 5$ en $x = 0.3$, con 9 valores para $h$. La solución exacta es $f''' = 5.4$. Caso $O(h^2)$.

El error verdadero es $E = 5.401171877 - 5.4 = 0.001171877$, es decir, del orden de $1.1 \times 10^{-3}$.

### 4.4.2. caso $O(h^4)$

| **No** | $h$ | $f'''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 5.400000000 | 0.000003333 |
| 2 | 0.050000 | 5.400000000 | 0.000000208 |
| 3 | 0.025000 | 5.400000000 | 0.000000013 |
| 4 | 0.012500 | 5.399999999 | 0.000000001 |
| 5 | 0.006250 | 5.400000005 | 0.000000000 |

Cuadro 26: Evaluación de $f'''$ para $f(x) = x^5 + 5$ en $x = 0.3$, con 9 valores para $h$. La solución exacta es $f''' = 5.4$. Caso $O(h^4)$.

El error cometido vale $E = 5.400000005 - 5.4 = 0.000000005$, es decir es el orden de $5 \times 10^{-9}$.

## 4.5. Ejemplo 14: Evaluar $f'''$ para $f(x) = e^{-x^2} \ln x$

Dada la función $f(x) = e^x$, se pide evaluar $f'''$ en $x = 1.5$ empezando con $h = 0.1$ con 5 valores para $h$

### 4.5.1. caso $O(h^2)$

Construimos los cuadros 27 y 28 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.600929021 | 0.001666667 |
| 2 | 0.050000 | 1.581575958 | 0.000416667 |
| 3 | 0.025000 | 1.576666404 | 0.000104167 |
| 4 | 0.012500 | 1.575434505 | 0.000026042 |
| 5 | 0.006250 | 1.575126248 | 0.000006510 |

Cuadro 27: Evaluación de $f'''$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.5$, con 7 valores para $h$. No se dispone de la solución exacta. Caso $O(h^2)$.

No se dispone de la solución exacta, pero sabemos que el orden de magnitud del error es de $1 \times 10^{-5}$.

### 4.5.2. caso $O(h^4)$

| **No** | $h$ | $f'''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 1.575946095 | 0.000003333 |
| 2 | 0.050000 | 1.575082981 | 0.000000208 |
| 3 | 0.025000 | 1.575027218 | 0.000000013 |
| 4 | 0.012500 | 1.575023705 | 0.000000001 |
| 5 | 0.006250 | 1.575023485 | 0.000000000 |

Cuadro 28: Evaluación de $f'''$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.5$, con 7 valores para $h$. No se dispone de la solución exacta. Caso $O(h^2)$.

No se dispone de la solución exacta, pero sabemos que el orden de magnitud del error es de $1 \times 10^{-7}$.

## 4.6. Ejemplo 15: Evaluar $f'''$ para $f(x) = \sin x \cos x$

Dada la función $f(x) = \sin x \cos x$, se pide evaluar $f'''$ en $x = 3.141592$ empezando con $h = 0.1$ con 5 valores para $h$.

### 4.6.1. caso $O(h^2)$

Construimos los cuadros 29 y 30 con los resultados de la evaluación pedida.

| **No** | $h$ | $f'''(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | -3.960159641 | 0.001666667 |
| 2 | 0.050000 | -3.990009994 | 0.000416667 |
| 3 | 0.025000 | -3.997500625 | 0.000104167 |
| 4 | 0.012500 | -3.999375039 | 0.000026042 |
| 5 | 0.006250 | -3.999843752 | 0.000006510 |

Cuadro 29: Evaluación de $f'''$ para $f(x) = \sin x \cos x$ en $x = 3.141592$, con 5 valores para $h$. No se dispone de la solución exacta. Caso $O(h^2)$.

No se dispone de la solución exacta, pero nos inclinamos a “creer”que el valor exacto es -4.0 o en todo caso, muy cercano a tal valor. El orden de magnitud del error es de $1 \times 10^{-6}$.

### 4.6.2. caso $O(h^4)$

| **No** | $h$ | $f'''(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | -3.999629408 | 0.000003333 |
| 2 | 0.050000 | -3.999976710 | 0.000000208 |
| 3 | 0.025000 | -3.999998542 | 0.000000013 |
| 4 | 0.012500 | -3.999999909 | 0.000000001 |
| 5 | 0.006250 | -3.999999994 | 0.000000000 |

Cuadro 30: Evaluación de $f'''$ para $f(x) = \sin x \cos x$ en $x = 3.141592$, con 5 valores para $h$. No se dispone de la solución exacta. Caso $O(h^4)$.

El orden de magnitud del error es de $1 \times 10^{-8}$.

# 5. Derivadas de cuarto orden

## 5.1. Cuarto con error $O(h^2)$

Dada una función continua $f(x)$, la derivada numérica de cuarto orden $f^{IV}(x)$, calculada en $x$, se obtiene mediante la expresión:

$$f^{IV}(x) = \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + O(h^2) \tag{7}$$

donde empezamos calculando 7 con un valor de $h$ elegido arbitrariamente, luego se procede a evaluar 7 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer, segundo, y tercer orden.

## 5.2. Software

Para evaluar $f^{IV}(x)$ agregamos al código escrito la siguiente función:

```
func CalcDerivada_4(iFunction,iIters int,x,h float64)float64{
 // calcula la cuarta derivada en x con incremento h
 var f float64=0
 for i:=0;i<iIters;i++{
 y2:=EvalueFunction(iFunction,x+2*h)
 y1:=EvalueFunction(iFunction,x+h)
 y0:=EvalueFunction(iFunction,x)
 ym1:=EvalueFunction(iFunction,x-h)
 ym2:=EvalueFunction(iFunction,x-2*h)
 err:=GetErrorOrden(4,h)
 f=(y2-4*y1+6*y0-4*ym1+ym2)/(h*h*h*h)
 fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
 h/=2.0
 }
 return f
}
```

esta función se encarga de materializar la fórmula dada en 7. El resto del código no sufre cambios, con excepción de la cita a la función `CalcDerivada_4()` en la función `main()`, que debe sustituir a la función usada para las derivadas de tercer orden.

El error en este caso viene dado por una expresión de la forma:

$$E = \frac{h^2}{6} f^V(\epsilon)$$

en la cual hacemos $f^V(\epsilon) = 1$, para simplificar los cálculos con el entendido de que nos interesa conocer es el orden del error más que el error mismo.

## 5.3. Cuarto orden con error $O(h^4)$

Dada una función continua $f(x)$, la derivada numérica de cuarto orden $f^{IV}(x)$, calculada en $x$, se obtiene mediante la expresión:

$$f^{IV}(x) = \frac{-f(x+3h) + 12f(x+2h) - 39f(x+h) + 56f(x)}{6h^4} + \frac{-39f(x-h) + 12f(x-2h) - f(x-h)}{6h^4} + O(h^4) \quad (8)$$

donde empezamos calculando 8 con un valor de $h$ elegido arbitrariamente, luego se procede a evaluar 8 con $h$ reducido sucesivamente a la mitad, como en el caso de las derivadas de primer, segundo, y tercer orden.

Asimismo, el error cometido en este caso viene dado, por

$$E = O(h^4) = \frac{h^4}{30} f^V(\epsilon)$$

donde, una vez más, hacemos $f^V(\epsilon) = 1$, con el fin de simplificar los cálculos y considerar únicamente el orden de magnitud del error.

### 5.3.1. Software

Para ejecutar la expresión 8 se ha escrito la rutina:

```
func CalcDerivada_4B(iFunction,iIters int,x,h float64)float64{
 // calcula la cuarta derivada en x con incremento h
 var f float64=0
 for i:=0;i<iIters;i++{
  y3:=EvalueFunction(iFunction,x+3*h)
  y2:=EvalueFunction(iFunction,x+2*h)
  y1:=EvalueFunction(iFunction,x+h)
```

```
  y0:=EvalueFunction(iFunction,x)
  ym1:=EvalueFunction(iFunction,x-h)
  ym2:=EvalueFunction(iFunction,x-2*h)
  ym3:=EvalueFunction(iFunction,x-3*h)
  err:=GetErrorOrden(2,h)
  f=(-y3+12*y2-39*y1+56*y0-39*ym1+12*ym2-ym3)/(6*h*h*h*h)
  fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
  h/=2.0
 }
 return f
}
```

cuyos parámetros son como en el caso de su similar de orden $O(h^2)$. Una salida típica de esta rutina produce un resultado similar a:

```
=================DERIVACIÓN NUMÉRICA ===================
Tiempo antes de ejecutar: 2019-01-28 15:10:14.043683692
I: 0 h: 0.100000 f:1.648716455 err: 0.000003333
I: 1 h: 0.050000 f:1.648720970 err: 0.000000208
I: 2 h: 0.025000 f:1.648721256 err: 0.000000013
I: 3 h: 0.012500 f:1.648721259 err: 0.000000001
I: 4 h: 0.006250 f:1.648720402 err: 0.000000000
I: 5 h: 0.003125 f:1.648704832 err: 0.000000000
Begin Time: 15 : 10 : 14
End Time  : 15 : 10 : 14
```

donde la cita en la función `main()` es similar a:

```
iFunction:=0
x:=0.5
step:=0.1
iIters:=6
CalcDerivada_4B(iFunction,iIters,x,step)
```

el error es, como se ha señalado reiteradamente, aproximado y lo entrega la rutina `GetErrorOrden()`, la cual contempla los dos casos considerados $O(h^2)$ y $O(h^4)$.

## 5.4. Ejemplo 16: Evaluar $f^{IV}$ para $f(x) = e^x$

Dada la función $f(x) = e^x$, se pide evaluar $f^{IV}$ en $x = 3.14$ empezando con $h = 0.1$ con 5 valores para $h$.

### 5.4.1. caso $O(h^2)$

Construimos los cuadros 31 y 32 con los resultados de la evaluación pedida.

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 23.142402196 | 0.001666667 |
| 2 | 0.050000 | 23.113495275 | 0.000416667 |
| 3 | 0.025000 | 23.106273602 | 0.000104167 |
| 4 | 0.012500 | 23.104467837 | 0.000026042 |
| 5 | 0.006250 | 23.104017600 | 0.000006510 |

Cuadro 31: Evaluación de $f^{IV}$ para $f(x) = e^x$ en $x = 3.14$, con 6 valores para $h$. La solución exacta es 23.10386686. Caso $O(h^2)$.

El error verdadero es $E = 23.104017600 - 23.10386686 = 0.00015074$, es decir, es del orden de $1.5 \times 10^{-4}$.

### 5.4.2. caso $O(h^4)$

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 23.103799373 | 0.000003333 |
| 2 | 0.050000 | 23.103862640 | 0.000000208 |
| 3 | 0.025000 | 23.103866555 | 0.000000013 |
| 4 | 0.012500 | 23.103864150 | 0.000000001 |
| 5 | 0.006250 | 23.103868589 | 0.000000000 |

Cuadro 32: Evaluación de $f^{IV}$ para $f(x) = e^x$ en $x = 3.14$, con 5 valores para $h$. La solución exacta es 23.10386686. Caso $O(h^4)$.

El error cometido ahora vale $E = 23.103868589 - 23.10386686 = 0.00000172$, es del orden de $1.7 \times 10^{-6}$.

## 5.5. Ejemplo 17: Evaluar $f^{IV}$ para $f(x) = x^5 + 5$

Dada la función $f(x) = x^5 + 5$, se pide evaluar $f^{IV}$ en $x = 0.2$ empezando con $h = 0.1$ con 5 valores para $h$.

### 5.5.1. caso $O(h^2)$

Construimos los cuadros 33 y 34 con los resultados de la evaluación pedida.

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 24.000000000 | 0.001666667 |
| 2 | 0.050000 | 24.000000000 | 0.000416667 |
| 3 | 0.025000 | 24.000000012 | 0.000104167 |
| 4 | 0.012500 | 23.999999976 | 0.000026042 |
| 5 | 0.006250 | 24.000001140 | 0.000006510 |

Cuadro 33: Evaluación de $f^{IV}$ para $f(x) = x^5 + 5$ en $x = 0.2$, con 5 valores para $h$. La solución exacta es 24. Caso $O(h^2)$.

El error verdadero es $E = 24.000001140 - 24 = 0.00000114$, es decir, es del orden de $1.4 \times 10^{-6}$.

### 5.5.2. caso $O(h^4)$

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 24.000000000 | 0.000003333 |
| 2 | 0.050000 | 24.000000000 | 0.000000208 |
| 3 | 0.025000 | 24.000000024 | 0.000000013 |
| 4 | 0.012500 | 23.999999861 | 0.000000001 |
| 5 | 0.006250 | 24.000000752 | 0.000000000 |

Cuadro 34: Evaluación de $f^{IV}$ para $f(x) = x^5 + 5$ en $x = 0.2$, con 5 valores para $h$. La solución exacta es 24. Caso $O(h^2)$.

El error verdadero es $E = 24.000000752 - 24 = 0.00000075$, es decir, es del orden de $7.5 \times 10^{-7}$.

## 5.6. Ejemplo 18: Evaluar $f^{IV}$ para $f(x) = e^{-x^2} \ln x$

Dada la función $f(x) = e^{-x^2} \ln x$, se pide evaluar $f^{IV}$ en $x = 1.57$ empezando con $h = 0.1$ con 5 valores para $h$.

### 5.6.1. caso $O(h^2)$

Construimos los cuadros 35 y 36 con los resultados de la evaluación pedida.

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | -5.221827899 | 0.001666667 |
| 2 | 0.050000 | -5.232904133 | 0.000416667 |
| 3 | 0.025000 | -5.235615571 | 0.000104167 |
| 4 | 0.012500 | -5.236289725 | 0.000026042 |
| 5 | 0.006250 | -5.236457991 | 0.000006510 |

Cuadro 35: Evaluación de $f^{IV}$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.57$, con 6 valores para $h$. La solución exacta se desconoce. Caso $O(h^2)$.

El error verdadero se desconoce pero su magnitud es del orden de $1 \times 10^{-6}$.

### 5.6.2. caso $O(h^4)$

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | -5.237234923 | 0.000003333 |
| 2 | 0.050000 | -5.236562608 | 0.000000208 |
| 3 | 0.025000 | -5.236517196 | 0.000000013 |
| 4 | 0.012500 | -5.236514305 | 0.000000001 |
| 5 | 0.006250 | -5.236514088 | 0.000000000 |

Cuadro 36: Evaluación de $f^{IV}$ para $f(x) = e^{-x^2} \ln x$ en $x = 1.57$, con 6 valores para $h$. La solución exacta se desconoce. Caso $O(h^4)$.

El valor exacto de $f^{IV}$ se desconoce pero creemos que es muy cercano a -5.236514 y su magnitud es del orden de $1 \times 10^{-8}$.

## 5.7. Ejemplo 19: Evaluar $f^{IV}$ para $f(x) = \sin x \cos x$

Dada la función $f(x) = \sin x \cos x$, se pide evaluar $f^{IV}$ en $x = 0.14$ empezando con $h = 0.1$ con 6 valores para $h$.

### 5.7.1. caso $O(h^2)$

Construimos los cuadros 37 y 38 con los resultados de la evaluación pedida.

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^2)$ |
|---|---|---|---|
| 1 | 0.100000 | 2.196150358 | 0.001666667 |
| 2 | 0.050000 | 2.207163209 | 0.000416667 |
| 3 | 0.025000 | 2.209924176 | 0.000104167 |
| 4 | 0.012500 | 2.210614903 | 0.000026042 |
| 5 | 0.006250 | 2.210787570 | 0.000006510 |
| 6 | 0.003125 | 2.210831735 | 0.000001628 |

Cuadro 37: Evaluación de $f^{IV}$ para $f(x) = \sin x \cos x$ en $x = 0.14$, con 6 valores para $h$. La solución exacta se desconoce. Caso $O(h^2)$.

También aquí, el error verdadero se desconoce pero su magnitud es del orden de $1 \times 10^{-6}$.

### 5.7.2. caso $O(h^4)$

| **No** | $h$ | $f^{IV}(x)$ | Error $O(h^4)$ |
|---|---|---|---|
| 1 | 0.100000 | 2.210742622 | 0.000003333 |
| 2 | 0.050000 | 2.210838750 | 0.000000208 |
| 3 | 0.025000 | 2.210844786 | 0.000000013 |
| 4 | 0.012500 | 2.210845167 | 0.000000001 |
| 5 | 0.006250 | 2.210845129 | 0.000000000 |
| 6 | 0.003125 | 2.210846578 | 0.000000000 |

Cuadro 38: Evaluación de $f^{IV}$ para $f(x) = \sin x \cos x$ en $x = 0.14$, con 6 valores para $h$. La solución exacta se desconoce. Caso $O(h^2)$.

El valor exacto de $f^{IV}$ se desconoce pero debe estar muy carca a 2.2108465. El orden de magnitud del error es de $1 \times 10^{-8}$.

# 6. Últimas obras del autor

- **N-ésimo primo**. Primer millón de números primos calculados con una fórmula para el n-ésimo primo. PDF publicado en facebook. 2019. 25 p.
- **Raíces**. Cálculo de ceros de función por métodos numéricos. PDF publicado en facebook. 2019. 50 p.
- **Integrales Imposibles**. Aproximación de integrales no elementales mediante métodos numéricos. PDF publicado en facebook. 2019. 53 p.
- **Integrales Especiales**. Aproximación de integrales de funciones especiales mediante métodos numéricos. PDF publicado en facebook. 2019. 35 p.
- **Funciones y Series**. Aproximación de funciones matemáticas mediante representación en series. PDF publicado en facebook. 2019. 17 p.
- **Un millón de precisiones entorno a** $\varphi$. Cálculo del número áureo con múltiple precisión. PDF publicado en facebook. 2018. 47 p.
- **Un millón de precisiones entorno a** $e$. Cálculo del número de Euler con múltiple precisión. PDF publicado en facebook. 2018. 25 p.
- **Un millón de precisiones entorno a** $\pi$. Cálculo de $\pi$ con múltiple precisión. PDF publicado en facebook. 2018. 29 p.

# A. Software completo

Presentamos a continuación una versión completa del código Go escrito con ocasión de este estudio.

```
package main

import(
"fmt"
"time"
"math"
)
```

```
//------------------------- VARIABLES GLOBALES------------
var fX[] float64
var fY[] float64
var sFunction string
//------------------------------------------------------------
func check(e error) {
if e != nil {
panic(e)
}
}
//--------------------------------------------------------------
func EvalueFunction(iFunction int,x float64)float64{
// evalua la funcion de iFunction en X
var y float64=0
switch(iFunction){
case 0:
sFunction="e^x"
y=math.Exp(x)
case 1:
sFunction="x^3+2"
y=x*x*x+2
case 2:
sFunction="sin x"
y=math.Sin(x)
case 3:
sFunction="ln x"
y=math.Log(x)
case 4:
sFunction="e^(-x^2)ln x"
y=math.Exp(-x*x)*math.Log(x)
case 5:
sFunction="sin x cos x"
y=math.Sin(x)*math.Cos(x)
case 6:
sFunction="x^5+5"
y=x*x*x*x*x+5
}
return y
```

```
}
//------------------------------------------------------------
func GetErrorOrden(iOrden int,h float64)float64{
// evalua el error aproximado
var y float64=0
switch(iOrden){
case 1:
y=(h*h)/6.0
case 2:
y=(h*h*h*h)/30.0
}
return y
}
//------------------------------------------------------------
func CalcDerivada_1(iFunction,iIters int,x,h float64)float64{
// calcula la primera derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y1:=EvalueFunction(iFunction,x+h)
y2:=EvalueFunction(iFunction,x-h)
err:=GetErrorOrden(1,h)
f=(y1-y2)/(2*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
}
//-------------------------------------------------------------
func CalcDerivada_1B(iFunction,iIters int,x,h float64)float64{
// calcula la primera derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y1:=EvalueFunction(iFunction,x+h)
y2:=EvalueFunction(iFunction,x+2*h)
ym1:=EvalueFunction(iFunction,x-h)
ym2:=EvalueFunction(iFunction,x-2*h)
err:=GetErrorOrden(2,h)
```

```
f=(-y2+8*y1-8*ym1+ym2)/(12*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
}
//------------------------------------------------------------
func CalcDerivada_2(iFunction,iIters int,x,h float64)float64{
// calcula la segunda derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y1:=EvalueFunction(iFunction,x+h)
y2:=EvalueFunction(iFunction,x-h)
y:=EvalueFunction(iFunction,x)
err:=GetErrorOrden(1,h)
f=(y1-2*y+y2)/(h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
}
//-------------------------------------------------------------
func CalcDerivada_2B(iFunction,iIters int,x,h float64)float64{
// calcula la segunda derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y0:=EvalueFunction(iFunction,x)
y1:=EvalueFunction(iFunction,x+h)
y2:=EvalueFunction(iFunction,x+2*h)
ym1:=EvalueFunction(iFunction,x-h)
ym2:=EvalueFunction(iFunction,x-2*h)
err:=GetErrorOrden(2,h)
f=(-y2+16*y1-30*y0+16*ym1-ym2)/(12*h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
```

```
}
return f
}
//----------------------------------------------------------
func CalcDerivada_3(iFunction,iIters int,x,h float64)float64{
// calcula la tercera derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y2:=EvalueFunction(iFunction,x+2*h)
y1:=EvalueFunction(iFunction,x+h)
ym2:=EvalueFunction(iFunction,x-2*h)
ym1:=EvalueFunction(iFunction,x-h)
err:=GetErrorOrden(1,h)
f=(y2-2*y1+2*ym1-ym2)/(2*h*h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
}
//-----------------------------------------------------------
func CalcDerivada_3B(iFunction,iIters int,x,h float64)float64{
// calcula la tercera derivada en x con O(h^4)
var f float64=0
for i:=0;i<iIters;i++{
y3:=EvalueFunction(iFunction,x+3*h)
y2:=EvalueFunction(iFunction,x+2*h)
y1:=EvalueFunction(iFunction,x+h)
ym3:=EvalueFunction(iFunction,x-3*h)
ym2:=EvalueFunction(iFunction,x-2*h)
ym1:=EvalueFunction(iFunction,x-h)
err:=GetErrorOrden(2,h)
f=(-y3+8*y2-13*y1+13*ym1-8*ym2+ym3)/(8*h*h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
```

```
}
//-------------------------------------------------------------
func CalcDerivada_4(iFunction,iIters int,x,h float64)float64{
// calcula la cuarta derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y2:=EvalueFunction(iFunction,x+2*h)
y1:=EvalueFunction(iFunction,x+h)
y0:=EvalueFunction(iFunction,x)
ym1:=EvalueFunction(iFunction,x-h)
ym2:=EvalueFunction(iFunction,x-2*h)
err:=GetErrorOrden(1,h)
f=(y2-4*y1+6*y0-4*ym1+ym2)/(h*h*h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
}
//--------------------------------------------------------------
func CalcDerivada_4B(iFunction,iIters int,x,h float64)float64{
// calcula la cuarta derivada en x con incremento h
var f float64=0
for i:=0;i<iIters;i++{
y3:=EvalueFunction(iFunction,x+3*h)
y2:=EvalueFunction(iFunction,x+2*h)
y1:=EvalueFunction(iFunction,x+h)
y0:=EvalueFunction(iFunction,x)
ym1:=EvalueFunction(iFunction,x-h)
ym2:=EvalueFunction(iFunction,x-2*h)
ym3:=EvalueFunction(iFunction,x-3*h)
err:=GetErrorOrden(2,h)
f=(-y3+12*y2-39*y1+56*y0-39*ym1+12*ym2-ym3)/(6*h*h*h*h)
//fmt.Printf("I: %d h: %.6f f:%.9f err: %.9f\n",i,h,f,err)
fmt.Printf("%d & %.6f & %.9f & %.9f\\\\ \\hline\n",i+1,h,f,err)
h/=2.0
}
return f
```

```
}
//---------------------------------------------------------------
func main() {
t:= time.Now()
fmt.Printf("=========DERIVACIÓN NUMÉRICA ===========\n")
fmt.Printf("Tiempo antes de ejecutar: %v \n",time.Now())
iFunction:=5
x:=0.14
step:=0.1
iIters:=6
CalcDerivada_4B(iFunction,iIters,x,step)
t2 := time.Now()
fmt.Printf("Begin Time: %d : %d : %d\n",
                             t.Hour(),t.Minute(),t.Second())
fmt.Printf("End Time  : %d : %d : %d\n",
                           t2.Hour(),t2.Minute(),t2.Second())
}
```

# Referencias


[1] GERALD, F.C. & WHEATLEY, O.P. *Análisis Numérico con Aplicaciones.* Pearson Educación. 2000. 698 p.

[2] USECHE, H. *Integrales Imposibles.* Aproximación de integrales no elementales mediante métodos numéricos. PDF publicado en facebook. 2019. 53 p.

[3] USECHE, H. *Integrales Especiales.* Aproximación de integrales de funciones especiales mediante métodos numéricos. PDF publicado en facebook. 2019. 35 p.

[4] USECHE, H. *Funciones y Series.* Aproximación de funciones matemáticas mediante representación en series. PDF publicado en facebook. 2019. 17 p.

[5] USECHE, H. *Raíces.* Primer millón de números primos calculados con una fórmula para el n-ésimo primo. PDF publicado en facebook. 2019. 25 p.

[6] USECHE, H. *Un Millón de Precisiones Entorno a $\varphi$*. Cálculo del número áureo con múltiple precisión. PDF publicado en facebook. 2018. 47 p.

[7] USECHE, H. *Un millón de precisiones entorno a e*. Cálculo del número de Euler con múltiple precisión. PDF publicado en facebook. 2018. 25 p.

[8] USECHE, H. *Un millón de precisiones entorno a $\pi$*. Cálculo de $\pi$ con múltiple precisión. PDF publicado en facebook. 2018. 29 p.

[9] WIKIPEDIA. *Integración numérica.* https://es.wikipedia.org/wiki/Integracion_numerica. 2018.

[10] WIKIPEDIA. *Regla del trapecio.* http/es.m.wikipedia.org/wiki/Regla_del_trapecio. 2018.

[11] WIKIPEDIA. *Regla de Simpson.* https://es.wikipedia.org/wiki/Regla_de_Simpson. 2018.

[9] WIKIPEDIA. *Función gamma.* http/es.m.wikipedia.org/wiki/Funcion_gamma