Tarea 3

Angel Caceres Licona

May 18, 2020

1 1 Complete la tabla 1

k	a_k	b_k	m_k	$f(m_k)$	$\frac{\lvert m_k - a\rvert}{\lvert a\rvert}$
0	1.8	2	1.9	-	0.9025
1	1.9	2	1.95	+	0.950625
2	1.9	1.95	1.925	-	0.92640625
3	1.925	1.95	1.9375	+	0.9384765625
4	1.925	1.9375	1.93125	-	0.932431640625
5	1.93125	1.9375	1.934375	+	0.9354516601562499

2 2 Elabore un programa que sume el numero 0.0001 10,000 veces consigo mismo...

```java
public strictfp class Suma {

    public static void main(String[] args) {
    Float numerof = 0.0001
    Double numerod = 0.0001
        for(int i = 1; i < 10000; i++) {
            numerof = numerof + 0.0001
            numerod = numerod + 0.0001
        }
    System.out.println("El resultado de sumar 0.0001 10,000 veces con precision sencilla es " + numerof)
    System.out.println("El resultado de sumar 0.0001 10,000 veces con precision doble es " + numerod)
    }

}
```

La salida del programa es la siguiente:

```
El resultado de sumar 0.0001 10,000 veces con precision
sencilla es 1.0000535
El resultado de sumar 0.0001 10,000 veces con precision
doble es 0.9999999999999062
```

a) Mi resultado es diferente de 1.
b) Hice el programa con 0.00001 y 0.000001 y obtuve los siguientes resultados:

```
El resultado de sumar 0.00001 10,000 veces con precision
sencilla es 0.0999915
El resultado de sumar 0.00001 10,000 veces con precision
doble es 0.09999999999999393
```

```
El resultado de sumar 0.000001 10,000 veces con precision
 sencilla es 0.009999673
El resultado de sumar 0.000001 10,000 veces con precision
 doble es 0.009999999999999105
```

c) En este caso creo que es posible obtener resultados menores que 1 porque el error que se va acumulando excede la precisión de la variable usada. Entonces en cada paso va guardando un numero menor que el que debería obtenerse y ese error se va acumulando.

3 Evalúe la expresión $\frac{A}{1-\cos x}$ en un valor cercano $x = 0$...

Para esto usaremos la serie de Taylor

$$f(a^*) - f(a) \approx f'(a)(a^* - a) \tag{1}$$

considerando

$$f'(a) = -\frac{A\sin(x)}{(1-\cos(x))^2} \tag{2}$$

y tomaré $a^* = 0.0001$. Sustituyendo obtenemos:

$$f(0.0001) \approx \frac{1-\sin(0.0001)}{(1-\cos(0.0001))^2}(0.0001-0) - \frac{1}{1-(\cos 0.0001)} \tag{3}$$

Para un valor arbitrario de A, en este caso 1 obtengo el siguiente valor: $3.9999x10^{12}$ y el valor de $f(0.0001) = 200,000,000$ Yo creo que la aproximación podría mejorarse tomando más términos de la serie de Taylor.

4 Se desea evaluar la expresión $f(x) = e^{5x}$...

Tomamos la expresión

$$|\epsilon_f| \approx |f'(a^*)||\epsilon_a| \tag{4}$$

Sustituimos los valores:

$$\begin{aligned} |\epsilon_f| &\approx |5e^5 1.01||0.01| \\ \epsilon_f &\approx 7.49486 \end{aligned} \tag{5}$$

El resultado de de $e^1 - e^{1.01}$ es $148.413 - 156.022 = 7.609$ que es similar a lo que predice la expresion anterior.

5 Codifique el siguiene algoritmo usando precision sencilla...

Los resultados son los siguientes:

```
Ingrese el numero: 0.2
El resultado es: 0.2
El resultado es: -5.551115E-17
Ingrese el numero: 0.25
El resultado es: 0.25
El resultado es: -8.326673E-17
Ingrese el numero: 1
El resultado es: 1.0
El resultado es: 0.0
Ingrese el numero: 1.8
El resultado es: 1.8
El resultado es: 0.0
Ingrese el numero: 2.5
El resultado es: 2.5
El resultado es: 0.0
Ingrese el numero: 3.14159
El resultado es: 3.14159
El resultado es: 0.0
Ingrese el numero: 0.08205
El resultado es: 0.08205
El resultado es: -2.7755576E-17
```

Lo que veo es que mientras es mas pequeño el numero que estoy evaluando el resultado que incluye la resta es menos preciso. Al cambiar a precision doble obtengo los siguientes resultados:

```
Ingrese el numero: 0.2
El resultado es: 0.20000000298023218
El resultado es: -5.551115123125783E-17
Ingrese el numero: 0.25
El resultado es: 0.24999999999999992
El resultado es: -8.326672684688674E-17
Ingrese el numero: 1
El resultado es: 1.0
El resultado es: 0.0
Ingrese el numero: 1.8
El resultado es: 1.7999999523162842
El resultado es: 0.0
Ingrese el numero: 2.5
El resultado es: 2.5
El resultado es: 0.0
Ingrese el numero: 3.14159
El resultado es: 3.141590118408203
El resultado es: 0.0
Ingrese el numero: 0.008205
El resultado es: 0.008205000311136284
El resultado es: 3.8163916471489756E-17
```

En el caso del resultado que incluye la resta del valor original, veo que aunque la primer operación no da el mismo resultado la resta si da cero, lo cual me da a entender que hay un error que rebasa la precision de la variable.

En el caso en el que se está elevando al cuadrado curiosamente si obtuve los resultados esperados para precision sencilla

```
Ingrese el numero: 0.2
El resultado es: 0.2
El resultado es: 0.0
Ingrese el numero: 0.25
El resultado es: 0.25
El resultado es: 0.0
Ingrese el numero: 1
El resultado es: 1.0
El resultado es: 0.0
Ingrese el numero: 1.9
El resultado es: 1.9
El resultado es: 0.0
Ingrese el numero: 2.5
El resultado es: 2.5
El resultado es: 0.0
Ingrese el numero: 3.14159
El resultado es: 3.14159
El resultado es: 0.0
Ingrese el numero: 0.008205
El resultado es: 0.008205
El resultado es: 0.0
```

y para precision doble

```
Ingrese el numero: 0.2
El resultado es: 0.20000000298023218
El resultado es: -5.551115123125783E-17
Ingrese el numero: 0.25
El resultado es: 0.24999999999999992
El resultado es: -8.326672684688674E-17
Ingrese el numero: 1
El resultado es: 1.0
El resultado es: 0.0
Ingrese el numero: 1.8
El resultado es: 1.7999999523162842
El resultado es: 0.0
Ingrese el numero: 2.5
El resultado es: 2.5
El resultado es: 0.0
Ingrese el numero: 3.14159
El resultado es: 3.141590118408203
El resultado es: 0.0
Ingrese el numero: 0.008205
El resultado es: 0.008205000311136284
El resultado es: 3.8163916471489756E-17
```

El codigo es el siguiente:

```
public strictfp class algoriitmoTarea2 {

    public static void main(String[] args) {
    Double numero = 0.01
    Double resultado = 0.0

    while(numero > 0){
        print "\nIngrese el numero: "
        numero = System.in.newReader().readLine() as Double
        if(numero < 0){
            break
        }
        resultado =  Math.sqrt(Math.pow(numero,2))
        print "El resultado es: " + resultado
        resultado = Math.sqrt(Math.pow(numero,2)) - numero
        print "\nEl resultado es: " + resultado
    }

    }

}
```