MÉTODOS NUMÉRICOS Y PROGRAMACIÓN **Curso 2026–2027**

Errores **(PRÁCTICA 2)**

1.– Se desea evaluar con un error absoluto inferior a $1\,cm^2$ el área de un recinto cuadrado. Para ello se mide un lado del recinto y se calcula el área elevándolo al cuadrado. Se sabe *"a priori"* que el recinto tiene $1\,m^2$ de superficie aproximadamente. ¿Qué precisión debe tener la regla que se utilice en la medición? ¿Sería posible efectuar correctamente este cálculo en un ordenador utilizando simple precisión?

2.– Se desea dibujar en un plotter la curva $y = f(x)$ para valores de x comprendidos en el intervalo $[a, b]$. Para ello se confecciona la siguiente subrutina:

```
SUBROUTINE CURVA(A,B,N)
IMPLICIT REAL*4 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
DELTA=(B-A)/FLOAT(N)
X=A
Y=F(A)
CALL MOVE(X,Y)
DO I=1,N
   X=A+FLOAT(I)*DELTA
   Y=F(X)
   CALL DRAW(X,Y)
ENDDO
RETURN
END
```

donde la subrutina `MOVE(X,Y)` mueve la pluma sin dibujar hasta el punto de coordenadas (x, y), la subrutina `DRAW(X,Y)` mueve la pluma dibujando una línea recta desde la posición anterior hasta el punto de coordenadas (x, y), y `F(X)` es una función del tipo `REAL*4 FUNCTION` que se confecciona separadamente. Según el manual del compilador FORTRAN, para las variables del tipo `REAL*4` se destinan $m = 24$ bits para almacenar la mantisa (incluido el signo).

Se pide:

a) Explicar muy brevemente como funciona la subrutina.

b) Hallar razonadamente —en primera aproximación— un valor máximo de N a partir del cual no se mejoren los resultados, debido a que la precisión del ordenador no permite discriminar entre dos valores consecutivos de x.

c) Hallar razonadamente —en primera aproximación— un valor máximo de N a partir del cual no se mejoren los resultados, debido a que la precisión del ordenador no permite discriminar entre dos valores consecutivos de $f(x)$.

d) Teniendo en cuenta además que el plotter interpreta las coordenadas (x, y) en centímetros, y que la resolución máxima de la pluma es de 0.1 milímetros, ¿cuál es el valor máximo de N que debe utilizarse en la práctica?

NOTA: Pueden efectuarse las simplificaciones que se juzguen razonables, ya que se habla de "primera aproximación", siempre y cuando se justifiquen.

3.– Presentar diversos ejemplos patológicos de operaciones numéricas en los que se demuestre que en un ordenador digital:

- **a)** No deben restarse números "parecidos".
- **b)** No debe dividirse por números "pequeños".
- **c)** Es preferible sumar en primer lugar los números más pequeños.
- **d)** El orden de los factores altera el producto.

Indicar cuál es la justificación teórica de estas aseveraciones. Realizar los ejemplos en sistema decimal mediante una calculadora.

4.– En un cálculo es preciso evaluar la función:

$$f(x) = 1 - (1-x)(1+x) = x^2$$

para valores de x tales que $|x| << 1$.

Se utiliza un ordenador que utiliza m bits para el almacenamiento de la mantisa (incluido el signo) en coma flotante, y que redondea por aproximación.

- **a)** ¿Se debe calcular $f(x)$ como $1-(1-x)(1+x)$ o como x^2? (En el último caso el cálculo se realiza multiplicando la variable x por sí misma.)
- **b)** Establecer las cotas del error relativo en los dos casos.
- **c)** ¿Qué operación es mejor según el valor de x ?
- **d)** Presentar un ejemplo **numérico** de la conveniencia de usar uno u otro método cuando los cálculos se realizan manualmente con ayuda de una calculadora, y se redondea por aproximación con tres dígitos decimales significativos.

5.– Las n primeras potencias $\varphi^1, \varphi^2, \ldots, \varphi^n$ del número áureo $\varphi = (-1+\sqrt{5})/2$ pueden obtenerse sin realizar ninguna multiplicación, ya que se cumple la relación:

$$\varphi^i = \varphi^{i-2} - \varphi^{i-1}.$$

Demostrar que esta forma de efectuar los cálculos es numéricamente inestable, mientras que la multiplicación reiterada es numéricamente estable.

6.– Se desea calcular aproximadamente e^x para diversos valores de x en el intervalo $[0, 10]$ sumando los primeros n términos de la serie de Taylor. El cálculo se efectuará en un ordenador que utiliza 24 bits para la mantisa en coma flotante (incluyendo el signo). Efectuar un estudio sencillo (sin tener en cuenta el efecto de la propagación del error de redondeo) que establezca un valor máximo del número de términos a considerar, por encima del cuál no se obtengan mejoras apreciables en la aproximación.

7.– En un proceso de cálculo se requiere evaluar repetidas veces la función $f(x) = \sqrt{1+x} - 1$. Es sabido, que para valores pequeños de x , existe, entre otras, la aproximación asintótica a la función anterior $f(x) \approx f_0(x) = x/2$. Los cálculos se efectuarán en un ordenador digital. ¿Es posible que los resultados obtenidos empleando la aproximación asintótica f_0 sean mejores que los resultados de los cálculos obtenidos empleando la propia función $f(x)$? En caso afirmativo: ¿para que rango de valores de x?; ¿cuál sería este rango si se utiliza precisión simple (24 bits para la mantisa, incluido el signo)? Ya sea el caso afirmativo o negativo, presentar diversos ejemplos numéricos que corroboren el resultado, empleando la base decimal y realizando las operaciones en coma flotante con tres dígitos.

8.– En un proceso de cálculo es necesario evaluar la función $f(x) = e^{-1/x}$ para diversos valores de $x \in [0.05, 0.10]$ con un error relativo $r_f \leq 10^{-8}$. Si los valores de x se conocen con un error relativo $r_x \leq 10^{-5}$, y suponiendo que podamos aumentar la precisión del ordenador hasta donde sea necesario, ¿Puede obtenerse un resultado satisfactorio? En caso contrario, ¿Con qué precisión sería necesario conocer los datos?

En cualquier caso, ¿qué tipo de precisión —simple o doble— debe usarse si los cálculos se efectúan en un ordenador digital? Se supondrá que en simple y en doble precisión se utilizan 24 y 53 bits, respectivamente, para almacenar la mantisa (incluido el signo).

9.– En un proceso de cálculo es preciso evaluar varios millones de veces la función $f(x) = \ln(1+x)$, siempre para valores de x positivos ($x > 0$) y pequeños ($x << 1$). La función del compilador FORTRAN de que se dispone es muy precisa, pero requiere un tiempo de computación relativamente elevado (del orden del equivalente a varias decenas de operaciones elementales).

Para reducir el tiempo de computación se valora la posibilidad de aproximar la función utilizando los dos primeros términos de su desarrollo de Taylor, admitiéndose esta simplificación cuando el error relativo sea inferior (en valor absoluto) a cien veces el error de almacenamiento de la máquina. Se confeccionan las siguientes subrutinas:

```
SUBROUTINE EXACTA(X,Y1)
REAL*4 X,Z,Y1
Z=1.+X
Y1=ALOG(Z)
RETURN
END
```

```
SUBROUTINE APROX(X,Y2)
REAL*4 X,T,Y2
T=1.-X/2.
Y2=X*T
RETURN
END
```

Sabiendo que el ordenador trabaja en coma flotante redondeando por aproximación y destinando $m = 24$ bits a la mantisa (incluido 1 bit para el signo), y que los valores de `X` son exactos (en el sentido de que su error inherente es nulo, aunque en general estarán afectados del correspondiente error de almacenamiento), se pide:

- **a)** Analizar instrucción por instrucción las subrutinas anteriores, obteniendo los errores relativos de las variables en cada paso.
- **b)** Obtener el error relativo del valor `Y1` calculado mediante la subrutina `EXACTA` respecto al valor exacto de $f(x)$. Obtener una cota del error y explicar qué sucede cuando x tiende a cero.
- **c)** Obtener el error relativo del valor `Y2` calculado mediante la subrutina `APROX` respecto al valor exacto de $f(x)$. Obtener una cota del error y explicar qué sucede cuando x tiende a cero.
- **d)** Discutir para que rango de valores de `X` puede utilizarse la subrutina `APROX` en lugar de la subrutina `EXACTA`.
- **e)** ¿Es posible que la subrutina `APROX` proporcione resultados más precisos que la subrutina `EXACTA`? En caso afirmativo, ¿para qué valores de `X`?

Nota: El desarrollo de Taylor (con el resto de Lagrange) de la función $f(x)$ es:

$$f(x) = -\sum_{i=1}^{n}(-1)^i \frac{x^i}{i} + R_n(x); \qquad R_n(x) = -\left(\frac{-1}{1+\xi}\right)^{n+1} \frac{x^{n+1}}{n+1}; \qquad \xi \in (0, x).$$

10.– Un programa FORTRAN incluye la instrucción `X=`n, donde `X` es una variable real de tipo `REAL*4` (simple precisión) o de tipo `REAL*8` (doble precisión), y n es una constante entera positiva de tipo `INTEGER*4`.

Deducir cuál es el valor de n a partir del cual se pueden producir errores de almacenamiento en simple precisión y en doble precisión.

11.– Realizar un programa FORTRAN que encuentre la solución del problema anterior mediante experimentación numérica.

12.– En un ordenador digital se calcula la suma

$$S_n = \overbrace{a + a + \ldots + a}^{\text{“}n\text{” veces}},$$

donde a es un número cualquiera cuyo error inherente es r_a^I. Las operaciones se realizan en coma flotante, destinando m bits a la mantisa (incluido el signo). Se pide:

- **a)** Calcular el error total del resultado que se obtiene al realizar la operación anterior. Indicar qué parte del error total se debe a la propagación del error inherente del dato a y qué parte se debe a la propagación de los errores de almacenamiento de las operaciones intermedias.
- **b)** Analizar cómo crece el error máximo al aumentar el valor de n. Discutir en qué medida este valor máximo puede considerarse exagerado.
- **c)** Repetir los apartados anteriores suponiendo que el cálculo se realiza en la forma $S_n = n{\cdot}a$. Comparar los resultados obtenidos en los dos casos.
- **d)** Relacionar estas conclusiones teóricas con el plantamiento del problema 8 de la práctica anterior y realizar una predicción teórica del PK a partir del cuál pueden producirse los problemas a los que hace referencia aquel ejercicio. En su caso, comparar la predicción teórica con los resultados "experimentales" obtenidos por simulación numérica.

13.– Un ingeniero programa habitualmente en FORTRAN y utiliza determinado ordenador digital. El ingeniero desea conocer cuál es el error de máquina r_M con el que se realizan los cálculos, tanto para las variables de tipo `REAL*4` (simple precisión) como para las variables de tipo `REAL*8` (doble precisión).

El ingeniero no tiene acceso a los manuales ni a ningún tipo de documentación en la que se explique cómo se almacenan los datos y cuántos bits se destinan al almacenamiento de la mantisa en cada caso.

Se pide:

- **a)** Relacionar el error de máquina r_M con el valor $\varepsilon > 0$ (*"machine epsilon"*), correspondiente al menor número real positivo tal que $\mathcal{A}(1+\varepsilon) \neq \mathcal{A}(1)$, siendo $\mathcal{A}(x)$ el valor almacenado correspondiente a x.
- **b)** Realizar un programa FORTRAN que permita obtener experimentalmente el valor del ε (*"machine epsilon"*) para los dos tipos de precisión (simple y doble) en el ordenador en el que se ejecute el programa.
- **c)** Probar el programa en un ordenador y comparar los resultados obtenidos con las predicciones teóricas. Explicar las discrepancias, en su caso.

14.– En un proceso de cálculo es preciso evaluar la función $f(x) = e^x - e^{-x}$, para valores de x pequeños en valor absoluto ($|x| << 1$). Para evaluar la función se confeccionan las dos siguientes subrutinas que representan dos modos alternativos de evaluar la función f:

```
SUBROUTINE EXACTA(X,Y1)
REAL*4 X,A,B,Y1
A=EXP(X)
B=EXP(-X)
Y1=A-B
RETURN
END
```

```
SUBROUTINE APROX(X,Y2)
REAL*4 X,Y2
Y2=2.*X
RETURN
END
```

La subrutina EXACTA corresponde a la evaluación de la función utilizando directamenta su expresión, mientras que la subrutina APROX corresponde a utilizar los dos primeros términos de su desarrollo en serie de Taylor.

Sabiendo que el ordenador trabaja en coma flotante redondeando por aproximación y destinando "m=24" bits a la mantisa (incluido 1 bit para el signo), y que los valores de `X` son exactos (en el sentido de que su error inherente es nulo, aunque en general estarán afectados del correspondiente error de almacenamiento), se pide:

- **a)** Analizar instrucción por instrucción las subrutinas anteriores, obteniendo los errores relativos de las variables en cada paso.
- **b)** Obtener el error relativo del valor `Y1` calculado mediante la subrutina `EXACTA` respecto al valor exacto de $f(x)$. Explicar qué sucede cuando x tiende a cero.
- **c)** Obtener el error relativo del valor `Y2` calculado mediante la subrutina `APROX` respecto al valor exacto de $f(x)$. Explicar qué sucede cuando x tiende a cero.
- **d)** ¿Es posible que la subrutina `APROX` proporcione resultados más precisos que la subrutina `EXACTA`? ¿Por qué?

Nota: El desarrollo de Taylor de la función exponencial es:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

15.– En un programa de ordenador es preciso evaluar la función

$$f(x) = \text{sen}(2x) - 2x$$

para valores pequeños de x en valor absoluto ($|x| << 1$). Para ello se realiza la siguiente subrutina en lenguaje FORTRAN:

```
SUBROUTINE CALCULAR(X,F)
IMPLICIT REAL*8 (A-H, O-Z)
A = 2.D+00 * X
B = SIN(A)
F = B - A
RETURN
END
```

Se pide:

a) Analizar instrucción por instrucción la subrutina CALCULAR, obteniendo los errores relativos de las variables en cada paso. Explicitar el error relativo del valor F y analizar su comportamiento cuando el valor almacenado en la variable X tiende a cero. ¿Es numéricamente estable esta forma de realizar los cálculos? ¿Por qué?

b) Proponer un procedimiento alternativo que sea lo más sencillo posible y que permita evaluar la función $f(x)$ de forma numéricamente estable y que sea asintóticamente equivalente (para valores pequeños de x en valor absoluto). Justificar razonadamente la propuesta. Modificar la subrutina CALCULAR de acuerdo con el planteamiento propuesto y realizar de nuevo el apartado a).

c) Discutir cuál de las dos subrutinas resultantes es preferible en función del valor de x. Justificar la respuesta.

16.– En un programa de ordenador es preciso evaluar la función

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

para valores pequeños de x en valor absoluto ($|x| << 1$). Para ello se realiza la siguiente subrutina en lenguaje FORTRAN:

```
SUBROUTINE CALCULAR(X,F)
IMPLICIT REAL*8 (A-H, O-Z)
A = 2.D+00 * X
B = EXP(A)
F = B - 1.D+00
RETURN
END
```

Se pide:

a) Analizar instrucción por instrucción la subrutina CALCULAR, obteniendo los errores relativos de las variables en cada paso. Explicitar el error relativo del valor F y analizar su comportamiento cuando el valor almacenado en la variable X tiende a cero. ¿Es numéricamente estable esta forma de realizar los cálculos? ¿Por qué?

b) Proponer un procedimiento alternativo que sea lo más sencillo posible y que permita evaluar la función $f(x)$ de forma numéricamente estable y que sea asintóticamente equivalente (para valores pequeños de x en valor absoluto). Justificar razonadamente la propuesta. Modificar la subrutina CALCULAR de acuerdo con el planteamiento propuesto y realizar de nuevo el apartado **a)**.

c) Discutir cuál de las dos subrutinas resultantes es preferible en función del valor de x. Justificar la respuesta.

17.– En un programa de ordenador es preciso evaluar la función

$$f(x) = \frac{1}{\text{sen}(x)} - \frac{1}{\tan(x)}$$

para valores pequeños de x en valor absoluto ($|x| << 1$). Para ello se realiza la siguiente subrutina en lenguaje FORTRAN:

```
SUBROUTINE CALCULAR(X,F)
IMPLICIT REAL*8 (A-H, O-Z)
A = SIN(X)
B = TAN(X)
C = 1/A
D = 1/B
F = C - D
RETURN
END
```

Se pide:

a) Analizar instrucción por instrucción la subrutina CALCULAR, obteniendo los errores relativos de las variables en cada paso. Explicitar el error relativo del valor F y analizar su comportamiento cuando el valor almacenado en la variable X tiende a cero. ¿Es numéricamente estable esta forma de realizar los cálculos? ¿Por qué?

b) Proponer un procedimiento alternativo que sea lo más sencillo posible y que permita evaluar la función $f(x)$ de forma numéricamente estable y que sea asintóticamente equivalente (para valores pequeños de x en valor absoluto). Justificar razonadamente la propuesta. Modificar la subrutina CALCULAR de acuerdo con el planteamiento propuesto y realizar de nuevo el apartado **a)**.

c) Discutir cuál de las dos subrutinas resultantes es preferible en función del valor de x. Justificar la respuesta.

Nota: $f(x) = x/2 + x^3/24 + x^5/240 + O(x^7)$