Temas: **Programación en MATLAB**: Sentencias, expresiones y variables. Estructuras de control. Operadores relacionales y lógicos. Programación de funciones.
Aritmética finita: Representación de números en computadora,dígitos significativos, errores de redondeo, precisión.

1. Evaluar el siguiente polinomio:

$$3.45x^7 - 1.32x^6 + 2.51x^5 + 0.73x^4 - 4.16x^3 + 2.27x^2 + 1.98x + 4.57$$

para valores de x=0.23, x=-0.18

a) Calculando cada término por separado y sumándolos finalmente. ¿ Cuántos productos por **x** se necesitaría realizar, si las potencias se calcularan como productos de **x**. (P/ej. $x^3 = x*x*x$) ?

b) Calculando cada término por separado y sumándolos finalmente ¿ Cuántos productos por **x** se necesitaría realizar, si las potencias se calcularan en forma anidada ? (P/ej. $x^4 = x * x^3$ y $x^3 = x * x^2$, etc.)

c) ¿Cuántos productos por **x** se necesitaría realizar, si el polinomio se reescribiera de la siguiente forma?

$$4.57 + x(1.98 + x(2.27 + x(-4.16 + x(0.73 + x(2.51 + x(-1.32 + x(3.45)))))))$$

2. Escribir una función en MATLAB y evaluar con el polinomio y los puntos del ejercicio anterior.

a) Que permita evaluar un polinomio de grado **n**. Los valores de los coeficientes se enviarán a la función en la forma de un vector, comenzando por el coeficiente de menor grado.

b) Generalizar la función anterior para poder evaluar el polinomio en varios puntos simultáneamente.

3. Convertir de Binario a decimal

a) Numeros binarios enteros con 8 digitos

Ejemplo: 01000010 $\rightarrow 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 64 + 2 = 66$

1) 00000001
2) 01010101
3) 00100111

b) Numeros binarios con 8 dígitos

Ejemplo: .01010000 $\rightarrow 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 0 \cdot 2^{-6} + 0 \cdot 2^{-7} + 0 \cdot 2^{-8} = 1/4 + 1/16 = 0.3125$

1) .00000001
2) .01010101
3) .00100111

c) Numeros binarios con 4 dígitos para la parte entera y 4 digitos para la parte decimal

> Ejemplo: 0001.1000 $\rightarrow 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} = 1 + 1/2 = 1.5$

1) 0000.0001
2) 0101.0101
3) 0010.0111

d) Numeros binarios en formato flotante normalizado, con 16 bits, de los cuales uno es el bit de signo, 4 son bits de exponente (con exceso $z = 2^{4-1} - 1 = 7$) y 11 bits son de mantisa (mantisa=$0.1d_1d_2d_3...d_n$).

> Ejemplo: Signo = 0, Exponente = 1001, Mantisa = 10100000000
> Resultado:
>
> - Calculo el exponente en decimal: 1001 $\rightarrow 1 \cdot 2^3 + 1 \cdot 2^0 = 8 + 1 = 9$
> - Resto el exceso: $9 - 7$ (exceso) $= 2$
> - Armo el numero $0.1d_1d_2d_3...d_n \times 2^{exp}$: $.110100000000 \cdot 2^2$
> - Elimino el exponente corriendo el cero: 11.0100000000
> - Convierto a decimal: $1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-2} = 2 + 1 + 1/4 = 3.25$

1) Signo = 0, Exponente = 0010, Mantisa = 01010110101
2) Signo = 0, Exponente = 1110, Mantisa = 11100000000
3) Signo = 1, Exponente = 1000, Mantisa = 10000000000

4. Convertir de Decimal a Binario

a) A numeros enteros binarios con 8 digitos

Ejemplo: 66 a binario con

- $66 = 2 * 33 + 0$
- $33 = 2 * 16 + 1$
- $16 = 2 * 8 + 0$
- $8 = 2 * 4 + 0$
- $4 = 2 * 2 + 0$
- $2 = 2 * 1 + 0$
- $1 = 2 * 0 + 1$

Por lo tanto, tomando los "restos" de los productos, de abajo para arriba, el numero es 1000010, para tener 8 bits le agregamos un cero a la izquierda: 01000010

1) 10
2) 127
3) 64

b) A numeros binarios con 8 dígitos decimales

Ejemplo: 0.3125

- $0.3125 * 2 = 0.625$
- $0.625 * 2 = 1.25$
- $0.25 * 2 = 0.5$
- $0.5 * 2 = 1.0$

Termina cuando la parte decimal es cero, o se alcanzan los 8 digitos. Se toman como resultado las partes enteras de los resultados, de arriba para abajo. Por lo tanto el resultado es 0.0101. Se agregan ceros a la derecha para tener 8 digitos decimales binarios, quedando 0.01010000.

1) 0.1
2) 0.05
3) 0.10921

c) Numeros binarios con 4 dígitos para la parte entera y 4 digitos para la parte decimal

1) 2.123
2) 7.102
3) 1.023

d) A numeros binarios en formato flotante normalizado, con 16 bits, de los cuales uno es el bit de signo, 4 son bits de exponente (con exceso $z = 2^{4-1} - 1 = 7$) y 11 bits son de mantisa (mantisa=$0.1d_1d_2d_3...d_n$).
   1) 2.5
   2) 0.0121
   3) 1021.2

5. Un número binario puede representarse en *punto flotante normalizado* (NBPFN) por medio de la siguiente expresión, donde **s** representa el *signo del número*, **f** la *fracción de la mantisa normalizada* y **e** el *exponente (con signo incluido)*.
En nuestro caso particular, MATLAB utiliza el formato **IEEE 754**. Por ejemplo, para el caso de números de doble precisión, tenemos:

$$NBPFN_{IEEE754} = (-1)^s * 2^{e-1023} * (1.f)$$

a) Exprese la *cantidad de bits* y los rangos de **e** y **f** utilizados para la *representación binaria* de números en *simple precisión*.

b) Idem con *doble precisión*.

c) Debido a las limitaciones de representación del formato antes mencionado. ¿ Cuál es la diferencia que existe entre el valor **1** y la representación del *número real* inmediatamente superior ?

6. Supongamos que tenemos un sistema de representación de números en *punto flotante normalizado* que consta de solo **8 bits**, distribuidos de la siguiente manera: **1 bit** para el *signo del número* , **4 bits** para la *fracción de la mantisa normalizada* y **3 bits** para el *exponente (exceso* $2^{3-1}-1 = 2^2-1 = 3$*)*.
(Asumimos que no hay representacion de numeros especiales, excepto "0-0000-000"=0)
Para este sistema de representación, calcule y exprese en formato binario y decimal:

a) El mayor número positivo que se puede representar.

b) El número positivo mas pequeño que se puede representar.

c) ¿ Cuál es la diferencia que existe entre el valor **1** y la representación del *número real* inmediatamente superior ?

7. Dado los siguientes números reales:

i) 17.93, ii) -0.938, iii) 15.34, iv) 10.1, v) 3.57

a) Transforme manualmente los siguientes números reales a binario y Realice el paso inverso para comprobar los resultados.

b) Cuántos dígitos binarios necesitaría para representar exactamente cada número?

c) Cuál es el error cometido en cada caso si la mantisa esta limitada a 8 bits?

8. Calcular, manualmente (con ayuda de Matlab o la calculadora para hacer las cuentas) y usando las cuatro definiciones de la teoría (error absoluto vs. error relativo, 10^{-t} vs. $0.5 * 10^{-t}$), con cuantas cifras significativas es la aproximación $\hat{x}$ a x.

a) $x = 3.141592$ a $\hat{x} = 3.1415$.

b) $x = 3.141592$ a $\hat{x} = 3.14$.

c) $x = 0.3141592$ a $\hat{x} = 0.314$.

d) $x = 314.1592$ a $\hat{x} = 314$.

e) $x = 0.0011174755$ a $\hat{x} = 0.00111748$.

f) $x = 11.174755$ a $\hat{x} = 11.1748$.

9. Escribir un programa Matlab que reciba dos parametros (a=valor exacto y b=aproximacion), y calcule con cuantas cifras significativas es la aproximación de "a" por "b". Comparar con los resultados obtenidos en el ejercicio 8.

10. Una vez obtenido los resultados, comparar los valores entre si. Que conclusiones puede sacar comparando, en el ejercicio 8?:

    a) Los incisos (a) y (b) (relativo a truncamiento)?

    b) Los incisos (b), (c) y (d) (relativo a la coma decimal)?

    c) Los incisos (e), y (f) (relativo a la coma decimal)?

11. Dados los siguientes valores exactos, calcular su *valor redondeado*, su *error absoluto*, su *error relativo*, y las *cifras significativas* de la aproximación del valor exacto por su valor redondeado.

    a) Valor exacto = 0.42331082513075, redondear a **6** decimales.

    b) Valor exacto = 0.31288191422707, redondear a **9** decimales.

12. Cuando dos números muy próximos entre sí se restan, se produce un fenómeno denominado *cancelación sustractiva*. Las siguientes funciones $f(x)$ y $g(x)$ son algebraicamente equivalentes. Calcule manualmente, con ayuda de Matlab o la calculadora, las expresiones con solo 6 cifras significativas (usando redondeo en cada operación), y ademas sin perdida de cifras significativas. Calcular el error absoluto y relativo producido por usar menos cifras significativas. Comparar $g(x)$ y $f(x)$ obtenidos sin perdida de cifras. Porque son diferentes?

    a) $f(500)$ y $g(500)$ para

$$f(x) = x(\sqrt{x+1} - \sqrt{x}) \qquad para \quad x \gg 1$$

$$g(x) = x/(\sqrt{x+1} + \sqrt{x}) \qquad para \quad x \gg 1$$

Errores de truncamiento:

a) La serie de Maclaurin para la función "cos(x)" es

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - ... = \sum_{k=0}^{n} \frac{(-1)^k x^{2k}}{(2k)!}$$

Aplicar la serie a $x = 3 * \pi/4$ con $n = 1, 2, 3, 4, 5$ y calcular el error absoluto y relativo respecto a $cos(x)$ calculado con Matlab.

b) La serie de Maclaurin para la función "sin(x)" es

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - ... = \sum_{k=1}^{n} \frac{(-1)^{k-1}\, x^{2k-1}}{(2k-1)!}$$

Aplicar la serie a $x = 3 * \pi/4$ con $n = 1, 2, 3, 4, 5$ y calcular el error absoluto y relativo respecto a $sin(x)$ calculado con Matlab.