Elementos de Cálculo Numérico (M) - Cálculo Numérico

`Primer Cuatrimestre 2013`

Práctica de Programación - Octave - Matlab.

1. **Fibonacci** La sucesión de Fibonacci está dada por la recurrencia: $a_1 = a_2 = 1$, $a_{n+2} = a_{n+1} + a_n$, para $n \geq 1$. Hacer un programa que calcule los primeros 100 términos de la sucesión y los guarde en un vector. Calcular el vector con los cocientes $\frac{a_{n+1}}{a_n}$ y graficarlo.

2. **Positividad y Negatividad** Considerar la función $f(x) = 15x^4 - 20x^3 - 180x^2 - 10$. Se desea determinar (aproximadamente) los intervalos de positividad y negatividad de f en el intervalo $I = [-3, 5]$. Para ello se propone el siguiente procedimiento: generar una discretización suficientemente fina del intervalo I (por ejemplo, con paso 0.01. Generar el vector `F` cuya coordenada i es el valor de f evaluada en el i-ésimo punto de la grilla. Recorrer `F` reemplazando cada uno de sus valores por su respectivo signo. Es decir: si `F(i)` es positivo, se reemplaza por 1 y si es negativo, por -1. Graficar el vector resultante.

3. **Números Primos.** Hacer un programa que detecte los números primos entre 1 y 100.

   Sugerencia: ver `help mod`. Una forma (burda, pero posible) de resolver este ejercicio es: generar el vector de números naturales del 1 al 100. Recorrer este vector y para cada casillero estudiar si el número correspondiente es divisible por los anteriores o no. Si lo es se lo deja igual, si no, se lo reemplaza por un 1. De este modo, al finalizar el programa, los casilleros que tienen un 1 son números compuestos, y los que no, son números primos.

   Una manera un poco más prolija es recorrer los números de 1 a 100 y estudiar si son divisibles por los *primos* menores: si el número es primo, lo guardo en un vector `p` y si no, no. Tener en cuenta que si uno tiene dos vectores fila `x` e `y` y la operación `[x y]` genera el vector fila que es la *concatenación* de `x` e `y`. De este modo se puede ir agrandando `p`, agregandole un número cada vez que se encuentra un primo.

4. **Divisores** ¿Qué numero entre 1 y 1000 tiene la mayor cantidad de divisores?¿Qué cantidad de divisores tiene? Escribir un programa que recorra estos números y, para cada uno de ellos, calcule el número de divisores. Luego, determinar cuál es el que tiene más divisores y cuántos tiene.

5. **Factorial** Escribir una función `f = factorial(a)` que reciba como input un número entero `a` y devuelva su factorial.

6. **Reverso** Escribir una función `y = reverso(y)` que reciba el vector $x = (x_1, x_2, \ldots, x_n)$ y devuelva el vector invertido $y = (x_n, \ldots, x_2, x_1)$.