Cálculo Numérico 2022
Trabajo Práctico 0

Introducción a Octave

Trabajos de Laboratorio en Cálculo Numérico 2022: En el presente dictado de la materia, se utilizará *exclusivamente* el software para cálculo científico `Octave`. Los trabajos prácticos y demás actividades de Laboratorio estarán específicamente orientadas al uso de dicho programa, con el objetivo de que el alumno aprenda a utilizarlo como herramienta para desarrollar a nivel práctico los contenidos de la asignatura. Además, las evaluaciones parciales involucran el conocimiento de dicho lenguaje para poder resolver los ejercicios de programación.

IMPORTANTE: MODALIDAD DE LAS PRÁCTICAS

- **PLATAFORMA**: http://e-fich.unl.edu.ar/moodle.
  En ella encontrarán las guías de ejercicios, las presentaciones en diapositivas de la teoría y otros materiales de interés.
- **PRÁCTICOS DE EVALUACIÓN CONTINUA**: En fechas a convenir, se realizarán trabajos prácticos que pueden consistir en resolver un problema propuesto a través del aula virtual, o un trabajo a entregar, utilizando los conocimientos previamente adquiridos.
- **FECHA DE TRABAJOS PRÁCTICOS**: La fecha se indicará desde la cátedra. En caso de que el alumno no lo realice, se considera el práctico desaprobado.

Introducción: El software científico `Octave` presenta las siguientes características: posee una amplia variedad de librerías de funciones orientadas al cálculo científico, es interactivo, programable, de libre uso con la condición de hacer referencia a sus autores y disponible tanto para plataformas Windows y Linux. El sitio oficial de `Octave` es http://www.octave.org. Allí se encuentra información general, manuales, FAQs (*Frequently Asked Questions*), referencias sobre reportes, diferencias y similitudes con Matlab, lista de errores, etc. y se pueden obtener la versiones binarias o los fuentes para las diferentes plataformas.

Ejercicio 1: Realice las siguientes operaciones utilizando las funciones apropiadas de Octave

(a) $5^2 - \frac{1}{2^3} - \sqrt{3^2 + (2 \times 2)^2} =$ 19.875

(b) $\sin(\pi/6) - \arctan(0.5) =$ 0.036352

(c) $\ln(3 + \frac{1}{5}) - e^2 =$ -6.2259

Nota: En la línea de comandos de Octave, se pueden recuperar instrucciones ejecutadas anteriormente pulsando la flecha dirigida hacia arriba, lo que evita la escritura reiterada de una misma instrucción.

Ejercicio 2: Sea la función

$$y = \frac{\sin(2x)}{x(x+1)}$$

y = @(x) sin(2*x)./(x.*(x+1))

(para evitar errores de compilación colocar el punto . antes de una multiplicación, división o potencias entre funciones de x)

Halle el valor numérico de y, para $x = -4$, $x = -\pi/8$, $x = \sqrt{2}/4$, $x = \pi/2$ y $x = 9\pi/5$. Saque conclusiones sobre el dominio de la función. ¿Se podrán calcular los valores de y correspondientes a $x = 0$ y a $x = -1$? Intente calcularlos y justifique su respuesta.

x = -4.0000 -0.3927 0.3536 1.5708 5.6549
y(x) = -8.2447e-02 2.9650e+00 1.3575e+00 3.0326e-17 -2.5272e-02

y(0) = NaN

en cualquier caso significa que la función no está definida en esos puntos

y(-1) = Inf

y

Ejercicio 3: Los siguientes ejemplos definen diferentes tipos de arreglos. Pruébelos y saque conclusiones:

(a) `[1 2 3 -4]` -Vector fila de 4 elementos

x = linspace(1,2,10) vector fila desde -1 hasta 2 con 10 valores equidistantes

(b) `[1 2 3 -4]'` -Vector columna de 4 elementos

(c) `-2.5:0.5:1` -Serie de elementos desde el -2.5 hasta el 1 con paso 0.5

(d) `(-2.5:0.5:1)'` -Serie de valores en columna desde el -2.5 hasta el 1 con paso 0.5

(e) `[-3:2:4]` Vector fila con valores desde el -3 hasta el 3 (no llega a 4) con paso 2

Nota: Cuando las operaciones aritméticas $+$, $-$, $*$ y $/$ se utilizan entre matrices (donde un vector columna se puede interpretar como una matriz de $n \times 1$), debe tenerse en cuenta la compatibilidad de las dimensiones de las mismas, para que tales operaciones tengan sentido. Se presentan a continuación las operaciones correspondientes a la multiplicación, división y potenciación elemento a elemento. De esta manera, dadas dos matrices A con elementos A_{ij} y B con elementos B_{ij} se tiene que

Declarar una matriz 2x3:
A = [1 2 3; 4 5 6]

(a) `A.*B` da como resultado una matriz `C` cuyos elementos son $C_{ij} = A_{ij} \cdot B_{ij}$ -multiplica componente a componente. A Y B DEBEN SER DEL MISMO TAMAÑO

(b) `A./B` da como resultado una matriz `D` cuyos elementos son $D_{ij} = A_{ij}/B_{ij}$ -Divide componente a componente. A Y B DEBEN SER DEL MISMO TAMAÑO

(c) `A.^n` resulta ser otra matriz cuyos elementos son A_{ij}^n -Eleva a la n cada componente de la matriz A

Ejercicio 4: Considere los arreglos

z*x --> me retorna una matriz de 3x3

x*z -->me retorna "una matriz" de 1x1 (básicamente es el producto punto entre vectores)

$$x = \begin{pmatrix} -0.5 & 7 & 2 \end{pmatrix} \qquad \text{y} \qquad z = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}$$

(m) w = [x', z] -Genera una matriz de 3x2 con los valores de x en la primer columna y los valores de z en la segunda columna.

Investigue qué realizan las siguientes operaciones:

Para el producto punto entre x e y:
x*y'
(x debe ser un vector fila e y un vector columna)

(a) `2*x` -Multiplica cada componente de x por 2

(b) `y = x-1` -A cada componente de x le resta 1 Ahora 'y' es un vector fila.

(c) `x.*y` -Retorna un vector donde se multiplica la componente i-esima de cada vector

(d) `x./y` -Retorna un vector donde se divide la componente i-esima de x por la de y

(e) `y.^2` .Eleva al cuadrado cada componente de y

(f) `x'+5*z` -Multiplica por 5 cada componente de z y luego suma componente a componente con el vector (columna) x'

(g) `x+5*z` -Multiplica cada componente de z por 5, luego retorna una matriz de 3x3. Donde las columnas son las componentes de z MAS la componente i-esima de x

(h) `y-z'/3` - Divide por 3 cada valor del vector (fila ahora) z' y hace la resta (componente a componente)

(i) `v = [x,y]` -Concatena los valores del vector x seguido los de y y los guarda en v

(j) `v(2:5)` -Retorna los valores del vector v, desde la posición 2 a la 5 inclusive. Posición inicial empieza en 1

(k) `v(5:6)+(z(1:2))'` -Toma las posiciones 5 y 6 del vector v, y las suma con las posiciones 1 y 2 del vector (fila ahora) z'

(l) `w = [x;y]` -Genera una matriz con los valores de x en la primer fila, y con los valores de y en la segunda fila

Ejercicio 5: En Octave los polinomios se representan por un vector de coeficientes ordenados de la mayor potencia hacia la menor. Por ejemplo, el polinomio $2x^3 - 3x^2 + 2$ se representa por el vector `[2, -3, 0, 2]`.
Sea `p = [2 3 5 7]` un vector que define los coeficientes de un polinomio. Investigue qué es lo que realizan las siguientes instrucciones:

(a) `polyout(p, 'x')` -Desarrolla el polinomio con la variable simbólica 'x'

(b) `polyval(p, x)` (Defina antes la variable x asignándole algún valor escalar y también como vector). -X es una variable o vector con valores. Polyval evalúa el polinomio p con los valores de x.

(c) `roots(p)` -Retorna las raíces del polinomio p

Ejercicio 6 (Aula): Los siguientes ejemplos definen diferentes tipos de matrices. Pruebe y saque conclusiones:

(a) `A = [1 2 3;4 5 6;7 8 9]` -Matriz de 3x3. El ";" separa las filas.

(b) `B = A'` - B ahora es la Transpuesta de A. Las filas pasan a ser columnas

(c) `C = [-3.2,5,7.4,6;4,17,-1.3,2.1;5.9,-6,0,4.5]` -Matriz de 3x4. Las "," separan las columnas (en este caso no es necesario colocarlas)

(d) `mat=C'` -mat es la Transpuesta de C. Matriz de 4x3

(e) `C(1:2,2:4)` -matriz con las filas 1y2 y columnas de 2 a 4

(f) `C(:,3)` -Matriz con todas las filas pero solo la columna 3. (o vector columna)

(g) `C(2,:)` -Matriz (o vector fila) con la fila 2 y todas las columnas

(h) `zeros(5,2)` -Matriz de 5x2 con todos los valores en 0.

(i) `ones(2,3)` -Matriz de 2x3 con todos los valores en 1.

(j) `v = diag(A)` -Vector columna con los valores de la diagonal de la matriz A. Vector de 3x1

(k) `D = diag(v,1)` Matriz diagonal con los valores de v en la diagonal 1. Matriz de 4x4 (con ceros en columna 1, Fila 4)

(l) `E = diag(v,-1)` -Matriz Diagonal con los valores de v en la diagonal -1. Matriz de 4x4 (con ceros en COLUMNA 4, FILA 1)

Matriz de 3x3, Diagonal principal con valores en 5, diagonal -1 (inferior) con valores en 1, diagonal 1 (superior) con valores -3. Ceros en posición 1,3 y 3,1

(m) `F = diag(5*ones(3,1),0)+diag(ones(2,1),-1)+diag(-3*ones(2,1),1)`

Ejercicio 7: Los siguientes comandos ejemplifican algunas de las posibilidades de manipulación de vectores que ofrece Octave. Trate de deducir que realiza en cada paso.

```
m = 5;
n = 4*m+1;
x = linspace(0,1,n);
y = zeros(1,n);
a = x(1:m+1);
y(1:m+1) = sin(2*pi*a);
y(2*m+1:-1:m+2) = y(1:m);
y(2*m+2:n) = -y(2:2*m);
```

-Asigna el valor 5 a la variable m.

-Asigna el valor 4*(5)+1 a la variable n. --> n=21

-Asigna 21 valores a x, desde el 0 hasta el 1, (paso 0.5)

-y es un vector fila de ceros. Vector de tamaño 1xn

-Asigna a 'a' los valores desde 1 hasta m+1 (6) de la serie de valores de x

-Asigna o reemplaza desde la posición 1, a la posición m+1 (6) de 'y' los valores del seno de 2*pi*a. 'a' debe ser de tamaño m+1

-Modifica con los valores de 'y' desde 1 a m(5) los valores de 'y' desde la posición 2*m+1 (11) en retroceso (-1) hasta la posición m+2 (7)

Ejercicio 8: Utilice los vectores x e y del punto anterior y grafíquelos con el comando `plot(x,y)`. A continuación grafique la siguiente función en el intervalo $[0, 2]$

$$f(x) = \left(\frac{x+1}{x^2+x+1}\right)^6 \cdot (\cos(x)+3)$$

Ejercicio 9 (Aula): Si A es una matriz cuadrada e invertible, el sistema $Ax = b$ tiene, teóricamente, una única solución. Investigue para un mismo sistema las siguientes instrucciones:

(a) `x1 = inv(A)*b`

(b) `x2 = A\b`

Ejercicio 10 (Aula): En Octave hay dos tipos de programas: los scripts y las funciones. Un script es simplemente una secuencia de órdenes. No tiene argumentos de entrada ni de salida. En cambio una función sí los tiene. Por otro lado, las variables definidas en un script son globales mientras que en una función, las variables definidas en la misma son locales.

Scripts: Un *script* es una secuencia de comandos que deseamos ejecutar a menudo y que por lo tanto nos gustaría no tener que escribirla cada vez que tenemos que usarla. Para esto podemos guardar la lista de comandos en un archivo de extensión `.m` y así poder ejecutarla tantas veces como queramos. El script puede estar colocado en cualquier carpeta y puede escribirse en un editor de textos cualquiera. Para mayor practicidad, se utiliza el procesador de texto incorporado en las nuevas versiones de `Octave`.

Escriba un script con el siguiente contenido:

```
n = 100;
A = rand(n,n);
x0 = rand(n,1);
b = A*x0;
x = A\b;
```

En `Octave` el script se ejecuta escribiendo el nombre del archivo (sin la extensión `.m`) en la línea de comandos.

Funciones: En `Octave`, la extensión es `.m` (como en los scripts). El esquema general de una función es

```
function [res1, res2, ...] = nombrefuncion(par1, par2, ...)
... ...
endfunction
```

donde `nombrefuncion` es el nombre de la función, que en el caso de `Octave` *debe coincidir con el nombre del archivo donde está escrita*. `par1`, `par2`, etc. son los argumentos de entrada y `res1`, `res2`, etc. son los argumentos de salida de la función. Escriba uno o varios archivos (según corresponda) con las siguientes funciones

```
function [x, y] = polarCart(r, t)
% Conversion de coordenadas polares a cartesianas.
x = r*cos(t) y = r*sin(t)
endfunction
%---------------------------------------------------------
function [x, y] = polarCartGr(r, t)
% Conversion de coordenadas polares a cartesianas,
% el angulo esta dado en grados.
[x, y] = polarCart(r, t*pi/180)
endfunction
%---------------------------------------------------------
function fx = f(x)
fx = x(1)^2 + x(2)^2
endfunction
```

En `Octave`, no hace falta cargar la función en el entorno, basta con colocar el archivo en un lugar donde `Octave` pueda "verla" (ver los comandos `path`, `addpath` y relacionados). Una vez cargada, la función se puede utilizar como las funciones *built-in* de Octave. Pruebe las siguientes instrucciones y saque conclusiones:

```
[x1, y1] = polarCart(2, 0.7854)
[u, v] = polarCartGr(3, 30)

valor = f([3; 4])
x = [5; 6], res = f(x)
```

Una diferencia importante entre función y script, es que las variables definidas en una función sólo existen dentro de ella, mientras que en un script, una vez ejecutado, las variables pasan a formar parte de nuestro espacio de trabajo.

Ejercicio 11 (Aula): Escriba una función de Octave que calcule la fórmula de Baskara, ingresando sólo un vector con los coeficientes del polinomio cuadrático y obteniendo como salida no sólo las raíces sino también una leyenda indicando el tipo de raíz.

Función "disp" para mostrar en pantalla:
disp('No existen raíces reales')

Gráficas: En `Octave`, el comando `plot` dibuja puntos en el plano generados al utilizar las componentes del primer vector introducido *versus* las componentes del segundo vector, como vimos en el Ejercicio 8.

Ejercicio 12 (Aula): Se utiliza el comando plot para representar los puntos $(-1, 3)$, $(0, 2)$, $(1.5, 2)$ y $(2, 0)$:

`x=[-1 0 1.5 2]; y=[3 2 2 0]; plot(x,y)` x e y deben ser del mismo tamaño

Investigue la diferencia que se producen con los siguientes comandos:

(a) `plot(x,y,'o')`

(b) `plot(x,y,'*')`

(c) `plot(x,y,'r')`

(d) `plot(x,y,'g*-')`

-title('....') --> Título a la grilla
-grid on --> para la grilla
-grid minor --> para una grilla mas chica
-hold on --> para poder agregar otra grafica en la misma figura

Podemos ver que cada vez que usamos el comando `plot` se genera un nuevo gráfico que sustituye al anterior. Si queremos que aparezcan varias gráficas conjuntamente, hay varias opciones. Una de ellas es la siguiente:

Ejercicio 13: Comentar al lado de las siguientes instrucciones lo que realiza cada una:

```
x=linspace(0,2*pi,201);  serie de 201 valores equidistantes entre el 0 y el 2pi
y=sin(x);      -aplico la función seno a cada valor de x y se los asigno a y
plot(x,y)      -grafico de los valores de x con los de y
hold on        -para poder superponer graficas en una misma figura
               - aplico la función coseno a cada valor de x y se los asigno a z
z=cos(x);
plot(x,z,'k-.')  - grafica de z en función de x, con líneas y puntos entrecortados
hold off       -Para dejar de poder superponer graficas en una misma figura
```