Práctica 1: Introducción a Matlab.

15 de noviembre de 2022

1. Introducción

En esta práctica se van introducir las bases de Matlab para resolver numéricamente y representar gráficamente los resultados de distintos problemas gobernados por ecuaciones diferenciales ordinarias. Asimismo, se realizarán interpolaciones y aproximaciones lineales. Todo ello se hará con problemas reales de Mecánica de Fluidos.

2. MatLab: Matrix Laboratory

Matlab es un lenguaje de programación interpretado que basa su potencia en el manejo de matrices. De esta forma, se pueden representar los vectores en filas o columnas.

```
A = [1,1,1] % Vector fila
A = [1;1;1] % Vector columna
A = [1,2,3;4,5,6;7,8,9] % Matriz
```

Asimismo se pueden definir vectores de forma automática indicando el comienzo del vector, el final y el espaciado entre ellos,

```
A = 0:0.1:10 % Vector fila
```

o mediante la función *linspace*, indicando el comienzo del vector, el final y el número de elementos,

```
A = linspace(0,10,11) % Vector fila
```

Existen funciones externas que permiten conocer el tamaño de una matriz o un vector, como son *size* o *length*,

```
A = [1,1,1]; size(A) % Devuelve: 1 3
A = [1;1;1]; size(A) %  Devuelve: 3 1
A = [1,2,3;4,5,6]; size(A) % Devuelve: 2 3
```

y el símbolo de punto y coma hace que no devuelva por pantalla lo ejecutado anteriormente en la misma línea.

Se puede acceder a los elementos de un vector o matriz indicándolo entre paréntesis,

```
A = [1,2,3;4,5,6];
A(2,2) % Devuelve 5
A(2,:) % Devuelve el vector de la segunda fila de la matriz A
```

También se tiene acceso a la traspuesta utilizando la siguiente expresión

```
A = [1,2]; % Vector fila
A' % Devuelve la traspuesta de A, es decir un vector columna
```

3. Expresiones algebraicas

A cada escalar que tengamos definido se le pueden realizar operaciones algebraicas y asignar el resultado a una nueva variable. Por ejemplo, se puede sustituir en un polinomio

```
x = 3
y = 1+2*x+x^2 % Que devuelve 16 después de evaluar 1+2*3+3^2
```

De igual manera, se pueden usar funciones trigonométricas y exponenciales, entre otras funciones especiales,

```
x = cos(pi/2) % Devuelve 0
y = exp(1) % Devuelve 2.7183
```

4. Bucles *for*

Se pueden realizar bucles for en los que una variable recorre los valores que toman los elementos de un vector. De esta forma, se puede realizar el producto entre una matriz y un vector,

```
A=rand(3,3)
x=rand(3,1)
for i=1:3
```

$T(°C)$	ρ(kg/m^3)	μ(Pa·s)
0	958.4	12.0700
10	981.8	3.9000
20	1003.9	1.4100
30	1027.3	0.6120
40	1052.1	0.2840
50	1077.3	0.1420
60	1103.9	0.0813
70	1121.8	0.0506
80	1156.0	0.0319
90	1182.7	0.0213

Cuadro 1: Medidas de densidad y viscosidad dinámica en una muestra de Glicerina 100 %.

```
for j=1:3
y(i,1)=A(i,j)*x(j);
end
end
```

que se puede realizar de forma compacta en Matlab como

```
y=A*x
```

5. Dibujar datos

Una de las características más valiosas de programas científicos interpretados, como pueden ser Matlab o Python-Scipy, es la posibilidad de dibujar gráficas de gran calidad. Supongamos que partimos de los datos densidad y viscosidad dinámica en función de la temperatura descritos en la tabla 1.

Para dibujar una gráfica con cada uno de ellos, esta tarea se puede realizar de forma sencilla declarando los distintos datos en una matriz A, y tomando cada una de las columnas como datos para la función *plot*. Se ha usado como marcador el círculo, y en el segundo caso se le indica que pinte también una línea entre dos puntos:

```
%%
% Carga los datos en la matriz A
A = [0    958.4  12.0700
   10   981.8    3.9000
```

```
   20   1003.9   1.4100
   30   1027.3   0.6120
   40   1052.1   0.2840
   50   1077.3   0.1420
   60   1103.9   0.0813
   70   1121.8   0.0506
   80   1156.0   0.0319
   90   1182.7   0.0213];
T = A(:,1);
rho = A(:,2);
mu = A(:,3);
% Pintar en dos figuras distintas, una con marcador y otra con marcador y
% líneas
figure(1)
plot(T,rho,'o')
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\rho$(kg/m$^3$)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
figure(2)
plot(T,mu,'o-')
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\mu$(Pa$\cdot$s)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
```

También podemos pintar los dos datos dentro de la misma figura mediante *subplot*:

```
%%
% Pintar en la misma figura
figure(3)
subplot(2,1,1)
plot(T,rho,'o')
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\rho$(kg/m$^3$)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
subplot(2,1,2)
plot(T,mu,'o-')
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\mu$(Pa$\cdot$s)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
```

Por último, se pueden editar de forma automática los marcadores cambiando el color del mismo, el color del borde, el tamaño, etc. En este caso se ha usado de marcador $'p'$, que se refiere a una estrella pentagonal. Para ver más opciones de marcadores y colores, acceder a la documentación de la función *plot* (escribir doc plot en el espacio de trabajo).

5.1. Interpolación y aproximación de funciones

A veces nos interesan valores de una variable en un punto intermedio que no tenemos. Para calcularlo se puede utilizar la interpolación, que consiste en encontrar una función que pase por todos los puntos de una lista. Según sea la aproximación utilizada (lineal, cúbica o mediante splines), la estimación tendrá un valor más cercano al real. Por ejemplo, para obtener la viscosidad para una temperatura de 25° C, se puede hacer lo siguiente:

```
%%
figure(5)
T0 = 25;
plot(T,mu,'o-')
hold on
mu_lineal = interp1(T,mu,T0);
mu_cubic = interp1(T,mu,T0,'cubic');
mu_spline = interp1(T,mu,T0,'spline');
plot(T0,mu_lineal,'s',T0,mu_cubic,'^',T0,mu_spline','d')
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\mu$(Pa$\cdot$s)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
```

Por otro lado, si en vez de utilizar funciones que pasen por todos los puntos de una lista se busca una función que minimice el error cometido con respecto a los datos de la misma, se habla de aproximación. Esto se puede realizar mediante la función $polyfit$, que ajusta por mínimos cuadrados un polinomio de grado n, definido en un vector como

$$p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \quad p = [a_n a_{n-1} \cdots a_1 a_0]. \qquad (1)$$

Si queremos obtener mediante aproximación el valor de la densidad para $T_0 = 25°$ C, se puede hacer lo siguiente:

```
%%
figure(6)
```

```
T0 = 25;
plot(T,rho,'o')
hold on
% Al ser un polinomio lineal se le indica que el grado del mismo es 1
p = polyfit(T,rho,1);
rho_approx = polyval(p,T);
plot(T,rho_approx)
xlabel('$T(^\circ C)$','Interpreter','LaTeX','FontSize',16)
ylabel('$\rho$(kg/m$^3$)','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
rho_approx_25 = polyval(p,T0);
display(['La densidad a ',num2str(T0),'°C es de ',num2str(rho_approx_25),'kg/m^3'])
```

6. Viscosímetro

En el Capítulo 5 se ha analizado cómo modelar un viscosímetro de embudo como el que se presenta en la figura 1. Éste consta de un depósito superior de sección A uniforme que descarga por un conducto circular de longitud L ($L \gg D$). Utilizando Hagen-Poiseuille podemos modelar el vaciado del depósito, que inicialmente se encuentra a una altura h_0 conocida, de la siguiente forma:

$$Q = \frac{\pi D^4}{128 L\mu}\rho g(L+h), \tag{2}$$

$$A\frac{dh}{dt} = -Q. \tag{3}$$

La ecuación diferencial para $h = h(t)$ a resolver será,

$$\begin{cases} A\frac{dh}{dt} = -\frac{\pi D^4}{128\mu L}\rho g(L+h), \\ h(t=0) = h_0. \end{cases} \tag{4}$$

Adimensionalizando las variables utilizando

$$\eta = \frac{h}{h_0}, \qquad \tau = \frac{t}{t_c}, \tag{5}$$

donde elegimos el tiempo característico t_c dado por

$$t_c = \frac{A128L\mu}{\pi D^4 \rho g}. \tag{6}$$

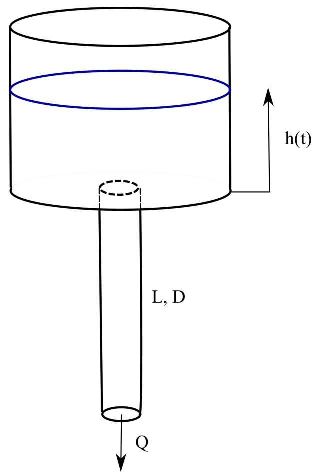

Figura 1: Esquema de un viscosímetro de embudo.

resultando en la siguiente ecuación diferencial adimensional para $\eta = \eta(\tau)$

$$\begin{cases} \frac{d\eta}{d\tau} = -\left(\alpha + \eta\right), \\ \eta(\tau = 0) = 1, \end{cases} \tag{7}$$

donde $\alpha = L/h_0$

La solución de esta ecuación diferencial es sencilla y se puede escribir como

$$\eta = (\alpha + 1)e^{-\tau} - \alpha. \tag{8}$$

6.1. Ejemplo de integración numérica con la función *ode45*

El siguiente código resuelve numéricamente las ecuaciones (7) y representa el resultado para varios valores del parámetro α (en línea discontinua). También se compara con la solución analítica (8) (representada con círculos),

```
close all % Cierra todas las graficas previas
clear all % Limpia todas las variables previas
clc % Limpia la consola

for alpha=0.2:.2:2; % Abre el bucle for en alpha

% Definicion de la ecuación diferencial
detadtau = @(tau,eta) -(alpha+eta);
```

```
% Tau final de la simulacion
tau_f=8;

% Valor de la condición inicial
eta0=1;
% Resolver la ecuación diferencial
[tau,eta] = ode45(@(tau,eta)detadtau(tau,eta),[0 tau_f],eta0);

% Representar graficamente
figure(1), hold on
plot(tau,eta,'LineWidth',2)  % Solucion numerica
plot(tau,(alpha+1)*exp(-tau)-alpha,'o') % Solucion analitica

end  % Cierra el bucle for en alpha

% Comandos de etiquetas y de estilo de la grafica
xlabel('$\tau$','Interpreter','LaTeX','FontSize',16)
ylabel('$\eta$','Interpreter','LaTeX','FontSize',16)
set(gca,'TickLabelInterpreter','Latex','FontSize',16)
grid on
```

7. Ejercicio

7.1. Apartado 1. (2 puntos)

Modifica el código anterior para representar el valor de τ en el momento final del vaciado τ_v dado por $\eta(\tau_v) = 0$, mediante interpolación y como función de α. Representar dicha función $t_v(\alpha)$ numérica y compárela con la analítica obtenida de (8).

7.2. Apartado 2. (2 puntos)

Modifique el código anterior para obtener la viscosidad de forma numérica como función de las variables del problema. Las propiedades del dispositivo son $A = 0{,}0314\text{m}^2$, $L = 0{,}25\text{m}$, $D = 0{,}01\text{m}$ y se realiza un experimento con un fluido de densidad $\rho = 950\text{kg/m}^3$. Obtenga la viscosidad para un experimento en el que se utiliza una altura inicial de $h_0 = 0{,}05\text{m}$ y se mide un tiempo de vaciado de $t_v = 40\text{s}$. Para ello, utilice la función $t_v(\alpha)$ obtenida anteriormente.

Finalmente, compruebe que la solución numérica es correcta comparando con la solución analítica.

7.3. Apartado 3. (2 puntos)

Modifique el código inicial para calcular la viscosidad integrando las ecuaciones dimensionales (4). Para ello, elimine el barrido paramétrico en α, el cual se puede considerar como medido en el experimento, y por tanto, dato de entrada. Valide el código mediante comparación con la solución analítica.

7.4. Apartado 4. (4 puntos)

Intégrese la ecuación diferencial que gobierna la velocidad de sedimentación de una esfera a bajos números de Reynolds (ecuación 14.49 de los apuntes),

$$\frac{4}{3}\pi R^3 \frac{\mathrm{d}V}{\mathrm{d}t} = \frac{4}{3}\pi R^3 (\rho_p - \rho) g - 6\pi\mu V R\,. \tag{9}$$

con condición inicial $V(0) = V_0$, siendo $V_0 = 0\text{m/s}$ y $V_0 = 1\text{m/s}$. Para ello, considere esferas de diferentes radios entre R=1m y 5m, con valores de propiedades $\rho = 1000\text{kg/m}^3$, $\rho_p = 7000\text{kg/m}^3$ y $\mu = 1\text{mPa·s}$, con gravedad $g = 9{,}81\text{m/s}^2$.

De manera opcional, se puede resolver el problema en forma adimensional, utilizando variables $\eta = V/V_c$ y $\tau = t/t_c$, eligiendo convenientemente valores de la velocidad y tiempo característicos, V_c y t_c, tal que los tres términos sean del mismo orden.