- صمم الكود المناسب بلغة **Matlab** لحل جملة المعادلات الخطية المعطاة , بالطريقتين التاليتين:

1- معكوس مصفوفة
2- طريقة **Cramer**

$$\begin{array}{cccccc} 3x & +2y & -z & = & 10 \\ -x & +3y & +2z & = & 5 \\ x & -y & -z & = & -1 \end{array}$$

-1

الحل:

$$A = \begin{bmatrix} 3 & 2 & -1 \\ -1 & 3 & 2 \\ 1 & -1 & -1 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad B = \begin{bmatrix} 10 \\ 5 \\ -1 \end{bmatrix}$$

$$A^{-1}AX = A^{-1}B$$
$$X = A^{-1}B$$

```
A=[3 2 -1; -1 3 2; 1 -1 -1];
B=[10 5 -1]';
x=inv(A)*B
```

```
x=
-2.0000
 5.0000
-6.0000
```

-2

الحل:

$$a_{11}x + a_{12}y + a_{13}z = A$$
$$a_{21}x + a_{22}y + a_{23}z = B$$
$$a_{31}x + a_{32}y + a_{33}z = C$$

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad D_1 = \begin{vmatrix} A & a_{12} & a_{13} \\ B & a_{22} & a_{23} \\ C & a_{32} & a_{33} \end{vmatrix} \quad D_2 = \begin{vmatrix} a_{11} & A & a_{13} \\ a_{21} & B & a_{23} \\ a_{31} & C & a_{33} \end{vmatrix} \quad D_3 = \begin{vmatrix} a_{11} & a_{12} & A \\ a_{21} & a_{22} & B \\ a_{31} & a_{32} & C \end{vmatrix}$$

$$x = \frac{D_1}{\Delta} \quad y = \frac{D_2}{\Delta} \quad z = \frac{D_3}{\Delta}$$

```
A=[3 2 -1; -1 3 2; 1 -1 -1];
delta=det(A);
d1=[10 2 -1; 5 3 2; -1 -1 -1];
D1=det(d1);
d2=[3 10 -1; -1 5 2; 1 -1 -1];
D2=det(d2);
d3=[3 2 10; -1 3 5; 1 -1 -1];
D3=det(d3);
x=D1/delta;
y=D2/delta;
```

```
z=D3/delta;
disp('x=');disp(x);
disp('y=');disp(y);
disp('z=');disp(z);

x=
   -2
y=
    5.0000
z=
   -6.0000
```

أو الكود التالي:

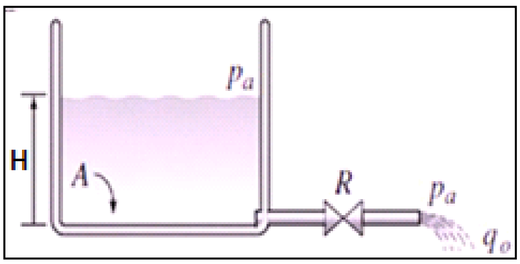
```
a=[3 2 -1 10; -1 3 2 5; 1 -1 -1 -1];
b=a(:,end);
a(:,end)=[];
delta=det(a);
for i=1:size(a,1)
    N=a;
    N(:,i)=b;
    D=det(N);
    x=D/delta;
    disp(['variable',num2str(i),'=']);
    disp(x)
end

variable1=
   -2
variable2=
    5.0000
variable3=
   -6.0000
```

- إذا كانت المعادلة التفاضلية لارتفاع السائل في الخزان الهيدروليكي المبين في الشكل تعطى بالعلاقة:

$$H' + 0.00613H = 0$$

حيث **H** ارتفاع السائل في الخزان **[m]**



و المطلوب:

أوجد حسابياً ارتفاع السائل في الخزان بعد **10 [s]** باستخدام طريقة **Euler** العددية بخطوة مقدارها **5 [s]** علماً أن الارتفاع الابتدائي للسائل في الخزان **1.5[m]**

الحل:

طريقة **Euler** تعطى بالمعادلات التالية:**H**

$$y_{i+1} = y(x_i) + f(x_i, y_i)h$$
$$x_{i+1} = x_i + h$$

نقوم بوضع المعادلة التفاضلية على شكل معادلة **Euler**

$$H' = -0.00613H$$
$$f(x, y) = -0.00613H$$

نضع i=0

$$y_{t=5} = y_{t=0} + f(0{,}1.5)h$$
$$y_{t=5} = 1.5 + (-0.00613(1.5))5 = 1.4540[\mathrm{m}]$$

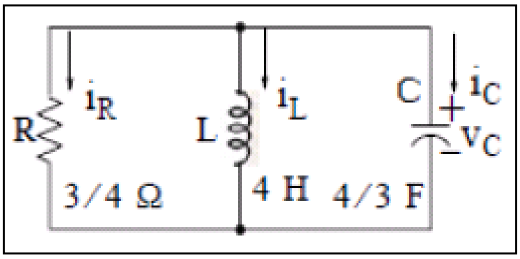
نضع i=1

$$y_{t=10} = y_{t=5} + f(5{,}1.4540)h$$
$$y_{t=10} = 1.4540 + (-0.00613(1.4540))5 = 1.4094[\mathrm{m}]$$

و هو ارتفاع السائل في الخزان بعد **10 [s]**

- في الدارة الكهربائية المبينة بالشكل تعطى معادلة تغير الجهد على المكثف كما يلي

$$v_c = -0.5e^{-0.25t} + 1.5e^{-0.75t}$$



و المطلوب :

صمم الكود المناسب بلغة **Matlab** لحساب اللحظة الزمنية التي يصبح فيها جهد المكثف يساوي **0.4[v]** باستخدام طريقة **Newton Raphson** العددية. (اجعل نقطة البدء **a=2** و دقة الحل **ε=0.001**)

الحل:

```
clc
clear
syms t
vct=-0.5*exp(-0.25*t)+1.5*exp(-0.75*t)-0.4;
u=diff (vct);
a=2
n=input('n=');
for i=1:n
    w=a-subs(vct,a)/subs(u,a);
    vc=double(w);
    disp(vc)
    if abs(w-a)<= 0.001
        break
```

```
    else
        a=w;
    end
end
```

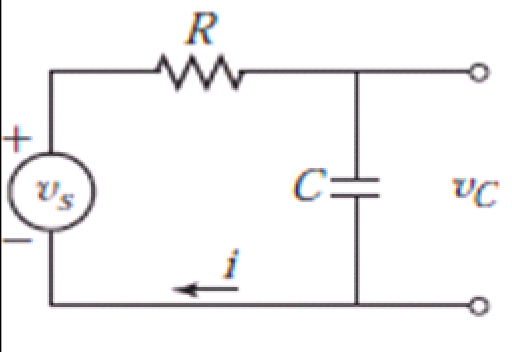
- تعبر المعادلة التفاضلية التالية عن تغير كمية عنصر مشع خلال الزمن (بالأيام) :

$$\frac{dy}{dt} = -ky$$

و المطلوب :
صمم الكود المناسب بلغة **Matlab** باستخدام طريقة **Runge Kutta** العددية لرسم علاقة تغير كمية المادة المشعة خلال **30 يوم** بخطوة زمنية **1 يوم** بفرض كمية المادة المشعة الابتدائية **20 وحدة كمية** حيث معدل التلاشي **k=0.0866**
الحل :

```
clc
clear
syms f x y
h = 1;
k=0.0866;
f =-k*y;
X(1) = 0;
Y(1) = 20;
xf = 30;
for i=1:(xf-X(1))/h
    y=Y(i);
    x=X(i);
    k1=subs(f);
    y=Y(i)+0.5*k1*h;
    x=X(i)+0.5*h;
    k2=subs(f);
    y=Y(i)+0.5*k2*h;
    x=X(i)+0.5*h;
    k3=subs(f);
    y=Y(i)+k3*h;
    x=X(i)+h;
    k4=subs(f);
    Y(i+1)=Y(i)+(1/6)*(k1+2*k2+2*k3+k4)*h;
    X(i+1)=X(i)+h;
end
plot(X,Y)
grid
```

- إذا علمت أن المقاومة في الدارة المبينة تبلغ **10^5 [Ω]** , و سعة المكثف **$C= 6.17*10^{-6}$[F]** , و بأن الجهد طبق عندما **t<0** ثم تم إيقافه فجأة عندما **t=0** , حيث تم عند تلك اللحظة قياس الجهد على المكثف خلال فترات زمنية لحظية كما هو مبين في الجدول :



Time t (s)	Voltage v_C (V)
0	5
0.5	2.2
1	0.9
1.5	0.4

و المطلوب :

تصميم الكود البرمجي المناسب بلغة **Matlab** لحساب جهد المكثف عند اللحظة **0.75 [s]** باستخدام طريقة **Lagrange Inetrpolation Polynomial** العددية

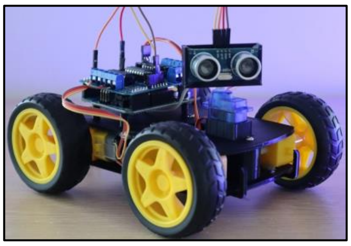
الحل:

باستيفاء النقاط المعطاة بمنحني من الدرجة الثالثة باستخدام طريقة **Lagrange Inetrpolation Polynomial**

$$l_N(x) = \sum_{m=0}^{N} y_m L_{N,m}(x) \text{ with } L_{N,m}(x) = \frac{\prod_{k\neq m}^{N}(x-x_k)}{\prod_{k\neq m}^{N}(x_m-x_k)} = \prod_{k\neq m}^{N} \frac{x-x_k}{x_m-x_k}$$

```
clc
clear
x = [0 0.5 1 1.5];
y = [5 2.2 0.9 0.4];
N = length(x)-1;
l = 0;
for m = 1:N + 1
    P = 1;
    for k = 1:N + 1
        if k ~= m
            P = conv(P,[1 -x(k)])/(x(m)-x(k));
        end
    end
    l = l + y(m)*P;
end
v_d=polyval(l,0.75)
```

- الجدول المبين في الشكل يعطي قياسات السرعة مع اللحظات الزمنية المقابلة لها لروبوت يتحرك بمسار مستقيم



Time [s]	0	0.2	0.6	1.2	1.4	2	2.5	3
Speed [cm/s]	0	1.26	1.9	1.99	1.99	1.99	2	2

و المطلوب :

صمم الكود المناسب بلغة **Matlab** للخوارزمية العددية المناسبة لحساب المسافة التي قطعها الروبوت بعد **3 ثانية**

الحل:

المسافة هي تكامل السرعة بالنسبة للزمن في الفترة من 0 إلى 3 , و التي يمكن حسابها من خلال تطبيق طريقة شبه المنحرف المركب **Composite Trapezoid** مع خطوة متغيرة في كل مقطع وفقا للجدول المعطى , كما يلي :

```
clc
clear
t=[0 0.2 0.6 1.2 1.4 2 2.5 3];
v=[0 1.26 1.9 1.99 1.99 1.99 2 2];
h=diff(t);
S=0.5*h(1)*(v(1)+v(2));
for i = 2:length(t)-1
    S = S+0.5*h(i)*(v(i)+v(i+1));
end
disp('The value of integral:')
disp(S)
```

- الجدول التالي يبين العلاقة بين الزمن و السرعة لروبوت يتحرك بمسار مستقيم و المبين في الشكل :

الزمن [s]	0	0.2	0.4
السرعة [m/s]	0	0.35	0.45

و المطلوب :

احسب رياضياً المسافة التي يقطعها الروبوت عند اللحظة **[s] 0.4** باستخدام طريقة **composite trapezoid** العددية

الحل:

في هذه الحالة ستكون المسافة هي تكامل السرعة في الفترة من 0 إلى 0.4 كما يلي:

$$distance = \int_0^{0.4} v\,dt$$

الحل العددي باستخدام قانون شبه المنحرف المركب حيث الفترة الزمنية من 0 إلى 0.4 مقسمة إلى مقطعين

$$D = \frac{b-a}{n}\left\{\frac{1}{2}[f(x_0)+f(x_n)] + f(x_1) + f(x_2) + \cdots + f(x_{n-2}) + f(x_{n-1})\right\}$$

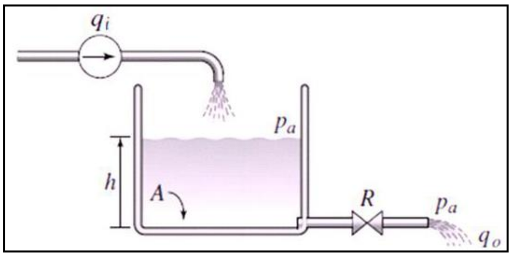
$$D = \frac{0.4-0}{2}\left(\frac{1}{2}f(x_0) + f(x_1) + \frac{1}{2}f(x_2)\right)$$

$$D = 0.2*(\frac{1}{2}0 + 0.35 + \frac{1}{2}0.45) = 0.115[m]$$

- إذا كان ارتفاع السائل في الخزان الهيدروليكي المبين في الشكل يعطى بالعلاقة:

$$h = 2 - 2e^{\frac{-t}{1000}}$$

حيث **h** ارتفاع السائل في الخزان **[m]**



و المطلوب :

تنفيذ خوارزمية الحل الرياضي لحساب الزمن اللازم لكي يصل ارتفاع السائل في الخزان إلى **1[m]** اعتباراً من الشروط الصفرية و ذلك باستخدام طريقة **Bisection** العددية **(طريقة التنصيف)** ضمن المجال **[0 1000]** وبخطأ مقداره **0.01**

(ملاحظة: يكتفى بإجراءعمليتي تنصيف في الحل الرياضي مع تحديد مجال البحث التالي بعد عملية التنصيف الثانية علماً بأن المطلوب حل رياضي و ليس برنامج حاسوبي)

الحل:

بالتعويض في معادلة ارتفاع السائل في الخزان :

$$1 = 2 - 2e^{-\frac{t}{1000}}$$

يصبح المطلوب إيجاد جذر المعادلة التالية:

$$f = 1 - 2e^{-\frac{t}{1000}}$$

نسمي بداية المجال a

و نهاية المجال b

$f(0) = -1$

$f(1000) = 0.2642$

$\varepsilon = b_0 - a_0 = 1000 - 0 = 1000 > \mathbf{0.01}$

$r = \dfrac{a_0 + b_0}{2} = \dfrac{0 + 1000}{2} = 500$

$f(500) = \text{-}0.2131$

$a_1 = r = 500, b_1 = b_0 = 1000$

$\varepsilon = b_1 - a_1 = 1000 - 500 = 500 > \mathbf{0.01}$

$r = \dfrac{a_1 + b_1}{2} = \dfrac{500 + 1000}{2} = 750$

$f(750) = 0.0553$

$a_2 = a_1 = 500, b_2 = r = 750$

و هكذا يتم الاستمرار بتكرار عملية التنصيف إلى أن نحصل على الدقة المطلوبة

- الجدول التالي يبين العلاقة بين الزمن و السرعة لروبوت يتحرك بمسار مستقيم و المبين بالشكل:

الزمن [s]	0	0.2	0.4	0.6
السرعة [m/s]	0	0.35	0.45	0.48

و المطلوب : احسب رياضياً سرعة الروبوت بعد زمن **0.3 [s]** باستخدام طريقة **Lagrange Inetrpolation Polynomial**

الحل:

باستيفاء النقاط المعطاة بمنحني من الدرجة الثالثة باستخدام طريقة **Lagrange Inetrpolation Polynomial**

$$l_N(x) = \sum_{m=0}^{N} y_m L_{N,m}(x) \text{ with } L_{N,m}(x) = \frac{\prod_{k\neq m}^{N}(x-x_k)}{\prod_{k\neq m}^{N}(x_m-x_k)} = \prod_{k\neq m}^{N}\frac{x-x_k}{x_m-x_k}$$

$$f_3(x) = \frac{(x-0.2)(x-0.4)(x-0.6)}{(0-0.2)(0-0.4)(0-0.6)}0 + \frac{(x-0)(x-0.4)(x-0.6)}{(0.2-0)(0.2-0.4)(0.2-0.6)}0.35 + \frac{(x-0)(x-0.2)(x-0.6)}{(0.4-0)(0.4-0.2)(0.4-0.6)}0.45 + + \frac{(x-0)(x-0.2)(x-0.4)}{(0.6-0)(0.6-0.2)(0.6-0.4)}0.48$$

و تعويض x=0.3 في العلاقة الناتجة :

$$f_3(0.3) = \frac{(0.3-0.2)(0.3-0.4)(0.3-0.6)}{(0-0.2)(0-0.4)(0-0.6)}0 + \frac{(0.3-0)(0.3-0.4)(0.3-0.6)}{(0.2-0)(0.2-0.4)(0.2-0.6)}0.35 + \frac{(0.3-0)(0.3-0.2)(0.3-0.6)}{(0.4-0)(0.4-0.2)(0.4-0.6)}0.45 + \frac{(0.3-0)(0.3-0.2)(0.3-0.4)}{(0.6-0)(0.6-0.2)(0.6-0.4)}0.48$$

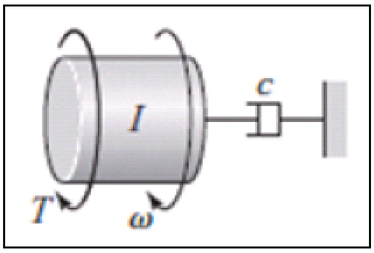
$f_3(0.3) = 0.42[m/s]$ و هي سرعة الروبوت بعد 0.3 ثانية وفق طريقة

Lagrange Inetrpolation Polynomial

- إذا كانت المعادلة التفاضلية لحركة ذراع روبوتية عند إيقاف عزم التدوير تعطى بالعلاقة:

$$w' + 20w = 0$$

حيث **w** السرعة الدورانية للذراع **[rad/s]**



و المطلوب:

تصميم كود برمجي بلغة **Matlab** لحساب السرعة الزاوية للذراع بعد **0.3** ثانية باستخدام طريقة **Runge Kutta** بخطوة مقدارها **0.01[s]** علماً أن السرعة الزاوية الابتدائية **3[rad/s]** حيث يقوم البرنامج برسم جميع نقاط الحل

الحل :

```
clc
clear all
syms f x y
h = input('step size='); % 0.01 [s]
f = input('the function f(x,y)='); % -20*y
X(1) = input('x0='); % 0 [s]
Y(1) = input('y0='); % 3 [rad/s]
xf = input('xf='); % 0.3 [s]
for i=1:(xf-X(1))/h
    y=Y(i);
    x=X(i);
    k1=subs(f);
    y=Y(i)+0.5*k1*h;
    x=X(i)+0.5*h;
    k2=subs(f);
    y=Y(i)+0.5*k2*h;
    x=X(i)+0.5*h;
```

```
    k3=subs(f);
    y=Y(i)+k3*h;
    x=X(i)+h;
    k4=subs(f);
    Y(i+1)=Y(i)+(1/6)*(k1+2*k2+2*k3+k4)*h;
    X(i+1)=X(i)+h;
end
plot (X,Y,'b.') % numerical solution
grid
```