```
function lab(arg)
clc
if nargin==0, arg='1'; end

switch (arg)

   case '0'
     ti=[-1 0 1 2 4]';  yi=[0 -2 0 3 3]';

     H1 = [ti.^0 ti ti.^2];
     H2 = [ti.^0 cos(ti) sin(ti)];
     c1 = H1\yi, c2 = H2\yi

     r1 = yi-H1*c1,   r2 = yi-H2*c2

     E1 = r1'*r1; E2 = r2'*r2;
     fprintf(' E1 = %.2f\n E2 = %.3f\n',E1,E2);

     tt=(-1.5:0.01:4.5);
     pp = polyval(c1(end:-1:1),tt);
     uu = c2(1) + c2(2)*cos(tt) + c2(3)*sin(tt);

     plot(ti,yi,'ko',tt,pp,'b',tt,uu,'r','LineWidth',2);

     % Usando ecuaciones normales
     c=inv(H1'*H1)*(H1'*yi)


   case '1'  % Linearizacion

       Tk = [25 100 150 300]';
       Vk = [0.7 0.06 0.02 0.0003]';

       b = log(Vk); H = [Tk.^0 -Tk];

       c = H\b; A = exp(c(1)); B = c(2); fprintf('A=%.3f B=%.3f\n',A,B);

       T=50;
       fprintf('Viscosidad a %.0fº = %.3f\n',T,A*exp(-B*T));

       T=(0:300);
       V = A*exp(-B*T);
       figure(1); plot(Tk,Vk,'ro',T,V,'b');


       res1 = (Vk-A*exp(-B*Tk)); res1',
       res2 = (b - H*c); res2'


       % Opcional
       r=(Vk-A*exp(-B*Tk));
```

```
    fprintf('Linearizando (A=%.3f B=%.3f)  -> ',A,B);
    fprintf('Error  %.4f\n',norm(r));

    X0=[A B];
    X=fminsearch(@f_coste,X0);
    A=X(1); B=X(2);  r = (Vk-A*exp(-B*Tk));
    fprintf('Solucion OK  (A=%.3f B=%.3f)  -> ',A,B);
    fprintf('Error  %.4f\n',norm(r));

    V = A*exp(-B*T); hold on; plot(Tk,Vk,'ro',T,V,'k'); hold off


case '2'  % Ajuste normal + ajuste con restricciones
   ti = (0.5:0.5:3)';
   hi = [6.9 11.3 12.5 12.4 9.3 4.2]';
   fprintf('%.1f ',hi); fprintf('\n');

   H = [ti -0.5*ti.^2]; v = hi; c = H\v;

   v0=c(1), g=c(2),

   Q = (H'*H); bb = H'*v;  c = Q\bb; % Equivalente

   res = v - H*c, norm(res)

   t = (0:0.01:3.5);
   h = v0*t-0.5*g*t.^2;
   plot(ti,hi,'ro',t,h,'b','LineWidth',2);

   g=9.8; H = [ti]; v = hi+0.5*g*ti.^2;
   v0=H\v,
   res = v - H*v0, norm(res)

case '3'   % ajuste libre / restricciones / pesos
  ti=[0 1 3 4 5]';  yi =[0 1 2.5 0.5 0]';  % DATOS TABLA

  % Ajuste libre
  H=[ti.^0 ti ti.^2 ti.^3]; v=yi;  c = H\v;

  tt=(0:0.01:5);

  pp=c(1)+c(2)*tt+c(3)*tt.^2+c(4)*tt.^3;

  figure(1);
  plot(ti,yi,'ro',tt,pp,'b')
  res1 = v - H*c
  fprintf('Norma residuos en ajuste libre %.2f\n',norm(res1))

  % Ajuste con condiciones p(0)=0 -> A=0  p(5)=0 -> B=-5C-25D
  H=[(ti.^2-5*ti) (ti.^3-25*ti)];
```

```
    c=H\v;
    C=c(1); D=c(2);  A=0; B=-5*C-25*D;
    pp2=A+B*tt+C*tt.^2+D*tt.^3;
    hold on; plot(tt,pp2,'r'); hold off

    res2 = v - H*c
    fprintf('Norma residuos en ajuste con restricciones %.2f\n',norm↙
(res2))

    % Usar pesos para acercarnos a 2º pueblo
    W=diag([1 1 4 1 1]);
    Q=H'*W*H;
    c=Q\(H'*W*v);
    C=c(1); D=c(2);  A=0; B=-5*C-25*D;

    pp3=A+B*tt+C*tt.^2+D*tt.^3;
    hold on; plot(tt,pp3,'g'); hold off

    res3 = v - H*c

    % Resolver con pesos el apartado b)
    w=[10000 1 1 1 10000]'; W=diag(w);
    H=sqrt(W)*[ti.^0 ti ti.^2 ti.^3]; v=sqrt(W)*yi;
    c=H\v;
    A=c(1); B=c(2);  C=c(3); D=c(4);

    pp4=A+B*tt+C*tt.^2+D*tt.^3;
    figure(2); plot(ti,yi,'ro',tt,pp2,'b',tt,pp4,'r')

    res4 = v - H*c


  case '4'  % Linearizacion

    % TABLA
    r_k = [19.25 12.84 9.38 6.97 3.77 1.77]';
    th_k= [ 3.12  3.00 2.89 2.79 2.51 1.99]';

    H = [th_k.^0  cos(th_k)  sin(th_k) ],  v = 1./r_k;
    % Descomentar para usar solo 3 datos (3 primeras filas de H y v)
    %H = H(1:3,:); v=v(1:3);

    cc = H\v  % Resolucion sistema

    A=cc(1); B=cc(2); C=cc(3);
    fprintf('A= %.3f   B = %.3f  C = %.3f\n',A,B,C);

    L = 1/A; E=L*sqrt(B*B+C*C); I=atan(C/B)*180/pi;
    fprintf('L= %.3f   E = %.3f  I = %.1f\n',L,E,I);

    % Preparación grafica
```

```
    th=(0:0.001:2*pi);
    r = 1./(A+B*cos(th)+C*sin(th));  % Orbita asteroide

    xk=r_k.*cos(th_k); yk=r_k.*sin(th_k);  % Datos de tabla
    plot(xk,yk,'rs'); hold on

    x_orb=r.*cos(th); y_orb=r.*sin(th); % Orbita asteroide
    plot(x_orb,y_orb,'b:');

    x_t = cos(th); y_t=sin(th);  % Orbita Tierra
    plot(x_t, y_t,'k');

    % Sol en centro.
    plot(0,0,'yo','MarkerFaceColor','y','MarkerSize',5);
    axis equal


  case '5'  % Ajuste con polinomios + linealización
     ti=  [0.5  1.5  2.5  3.5  4.5]';
     yi = [0.70 1.25 0.78 0.24 0.06]';

     plot(ti,yi,'bs');

     xx=(0:0.001:5);  % Malla fina para gráficos

     % SOLUCION CON POLINOMIO GRADO 3
     H = [ti.^0 ti ti.^2 ti.^3], c = H\yi; c'
     pp = polyval(c(end:-1:1),xx);      % Curva para graf

     r = yi -H*c; % residuos
     e1=sum(abs(r)); e2 = dot(r,r); e3=max(abs(r));  % Error global
     fprintf('Res p(t): '); fprintf('%6.3f ',r);
     fprintf(' => E1=%.4f  E2=%.4f  E3=%.4f\n',e1,e2,e3);

     % Solucion CON a · t · exp(b·t^2)
     H = [ti.^0 ti.^2], v = log(yi./ti); v', cc = H\v; cc'
     a=exp(cc(1)); b=cc(2);  % Parametros a y b originales
     uu = a*(xx).*exp(b*(xx).^2);   % Curva para grafica
     plot(ti,yi,'ks',xx,uu,'b',xx,pp,'r');  % Grafica

     r = yi-a*(ti).*exp(b*(ti).^2); % Residuos
     e1=sum(abs(r)); e2 = r'*r; e3=max(abs(r));  % Error global

     fprintf('Res u(t): '); fprintf('%6.3f ',r);
     fprintf(' => E1=%.4f  E2=%.4f  E3=%.4f\n',e1,e2,e3);
  end

end


function r=f_coste(X)
```

```
 tk=[25 100 150 300]';
 Vk=[0.7 0.06 0.02 0.0003]';
 A=X(1); B=X(2);
 res=Vk-A*exp(-B*tk);
 r=norm(res);
end
```