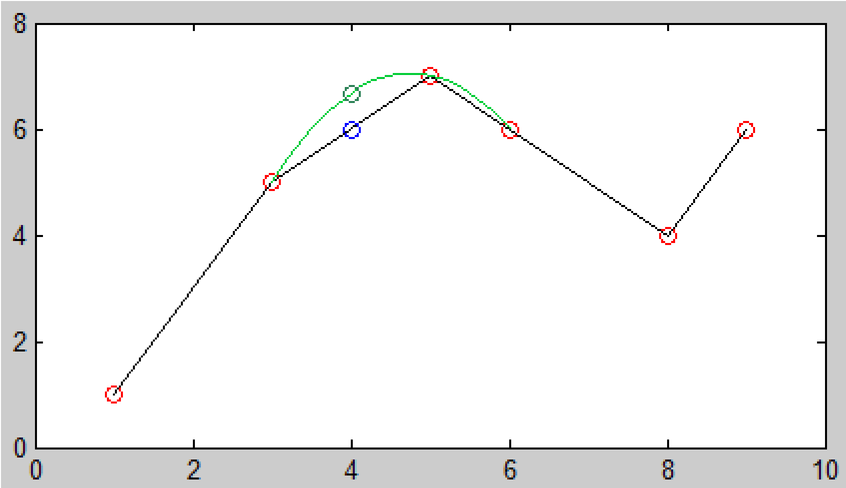
CÁLCULO NUMÉRICO E COMPUTACIONAL

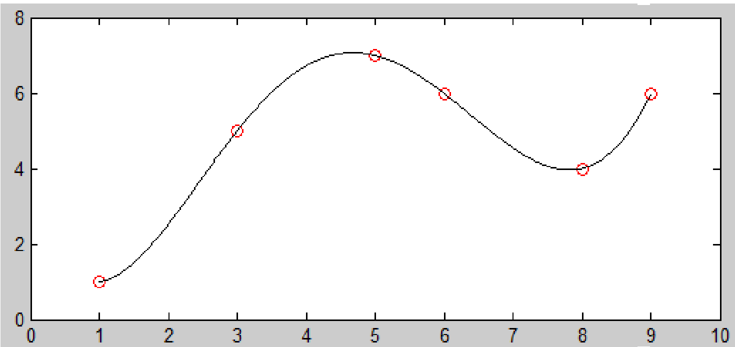
AULA de 19 de maio 2014

TÓPICOS EM CÁLCULO NUMÉRICO E MATLAB

INTERPOLAÇÃO

```
% EXEMPLO 1
clear all

%DATA POINTS
X=[1 3 5 6 8 9]
Y=[1 5 7 6 4 6]
plot(X,Y,'ro')
hold
plot(X,Y,'k-')

% INTERPOLAÇÃO LINEAR
x=4
y=interp1(X,Y,x)
plot(x,y,'bo')

% INTERPOLAÇÃO QUADRÁTICA
X1=[3 5 6]
Y1=[5 7 6]
p2=lagrangepoly(X1,Y1)
>> p2 =  -0.6667    6.3333   -8.0000
xp=3:0.1:6
yp=polyval(p2,xp)
plot(xp,yp,'g-')
x
y
y=polyval(p2,x)
plot(x,y,'go')

% O POLINÔMIO DE LAGRANGE
hold off
plot(X,Y,'ro')
hold
p5=lagrangepoly(X,Y)
>> p5 =  -0.0014    0.0640   -0.8310
            3.9895   -5.5426    3.3214
xp=1:0.01:9;
yp=polyval(p5,xp);
plot(xp,yp,'k-')
```





PLOT(Y) plots the columns of Y versus their index.

Various line types, plot symbols and colors may be obtained with PLOT(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

```
b     blue          .     point              -     solid
g     green         o     circle             :     dotted
r     red           x     x-mark             -.    dashdot
c     cyan          +     plus               --    dashed
m     magenta       *     star             (none)  no line
y     yellow        s     square
k     black         d     diamond
w     white         v     triangle (down)
                    ^     triangle (up)
                    <     triangle (left)
                    >     triangle (right)
                    p     pentagram
                    h     hexagram
```

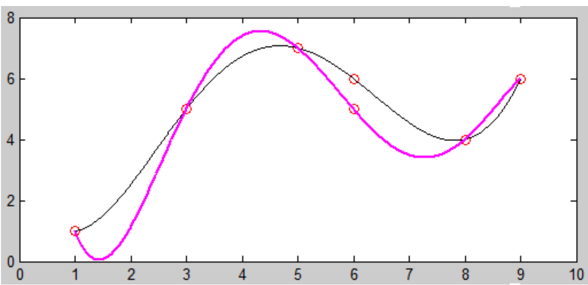
For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus at each data point; PLOT(X,Y,'bd') plots blue diamond at each data point but does not draw any line.

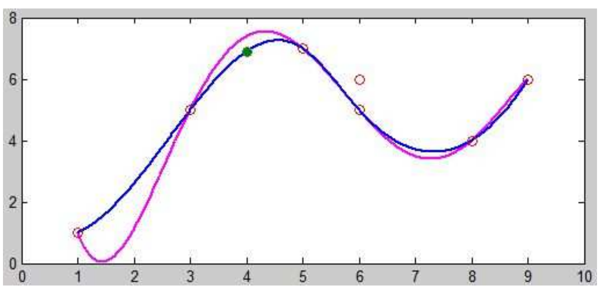
http://www.mathworks.in/matlabcentral/fileexchange/13151-lagrange-interpolator-polynomial/content/lagrangepoly/lagrangepoly.m

```
function [P,R,S] = lagrangepoly(X,Y,XX)
%LAGRANGEPOLY  Lagrange interpolation polynomial fitting a set of points
%   [P,R,S] = LAGRANGEPOLY(X,Y)  where X and Y are row vectors
%   defining a set of N points uses Lagrange's method to find
%   the N-1th order polynomial in X that passes through these
%   points.  P returns the N coefficients defining the polynomial,
%   in the same order as used by POLY and POLYVAL (highest order first).
%   Then, polyval(P,X) = Y.  R returns the x-coordinates of the N-1
%   extrema of the resulting polynomial (roots of its derivative),
%   and S returns the y-values  at those extrema.
%
%   YY = LAGRANGEPOLY(X,Y,XX) returns the values of the polynomial
%   sampled at the points specified in XX -- the same as
%   YY = POLYVAL(LAGRANGEPOLY(X,Y)).
%
%   Example:
%   To find the 4th-degree polynomial that oscillates between
%   1 and 0 across 5 points around zero, then plot the interpolation
%   on a denser grid inbetween:
%     X = -2:2;  Y = [1 0 1 0 1];
%     P = lagrangepoly(X,Y);
%     xx = -2.5:.01:2.5;
%     plot(xx,polyval(P,xx),X,Y,'or');
%     grid;
%   Or simply:
%     plot(xx,lagrangepoly(X,Y,xx));
%
%   Note: if you are just looking for a smooth curve passing through
%   a set of points, you can get a better fit with SPLINE, which
%   fits piecewise polynomials rather than a single polynomial.
%
%   See also: POLY, POLYVAL, SPLINE

% 2006-11-20 Dan Ellis dpwe@ee.columbia.edu
% $Header: $

%  For more info on Lagrange interpolation, see Mathworld:
%  http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html

% Make sure that X and Y are row vectors
if size(X,1) > 1;  X = X'; end
if size(Y,1) > 1;  Y = Y'; end
if size(X,1) > 1 || size(Y,1) > 1 || size(X,2) ~= size(Y,2)
  error('both inputs must be equal-length vectors')
end
```

```
N = length(X);

pvals = zeros(N,N);

% Calculate the polynomial weights for each order
for i = 1:N
  % the polynomial whose roots are all the values of X except this one
  pp = poly(X( (1:N) ~= i));
  % scale so its value is exactly 1 at this X point (and zero
  % at others, of course)
  pvals(i,:) = pp ./ polyval(pp, X(i));
end

% Each row gives the polynomial that is 1 at the corresponding X
% point and zero everywhere else, so weighting each row by the
% desired row and summing (in this case the polycoeffs) gives
% the final polynomial
P = Y*pvals;

if nargin==3
  % output is YY corresponding to input XX
  YY = polyval(P,XX);
  % assign to output
  P = YY;
end

if nargout > 1
  % Extra return arguments are values where dy/dx is zero
  % Solve for x s.t. dy/dx is zero i.e. roots of derivative polynomial
  % derivative of polynomial P scales each power by its power, downshifts
  R = roots( ((N-1):-1:1) .* P(1:(N-1)) );
  if nargout > 2
    % calculate the actual values at the points of zero derivative
    S = polyval(P,R);
  end
end
```

```
% O POLINÔMIO DE LAGRANGE
% EM GERAL NÃO É O MELHOR
% MÉTODO DE INTERPOLAÇÃO
Y(4)=5
plot(6,5,'ro')
p5=lagrangepoly(X,Y)
yp=polyval(p5,xp);
p5 =  -0.0125    0.3529    -3.5866
        15.7006  -26.7759   15.3214
plot(xp,yp,'m-')

% CUBIC SPLINE (bem melhor)
ys=spline(X,Y,xp);
plot(xp,ys,'b-')

y=spline(X,Y,4)
>> y = 6.8934
plot(x,y,'go')
```





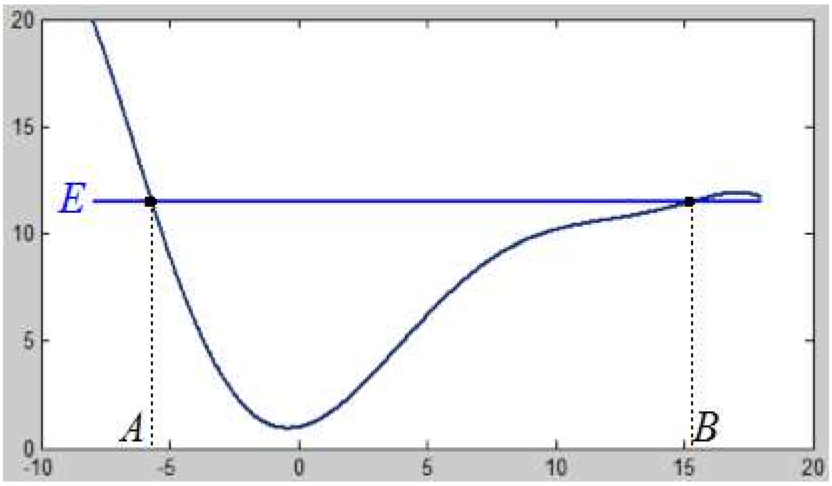
M. Fabbri USF 2014

INTEGRAÇÃO NUMÉRICA

EXEMPLO: Oscilador não-harmônico conservativo

A energia total se conserva: $U(x)+\frac{1}{2}mv^2=E \quad \Rightarrow \quad U(x)+\frac{1}{2}m\left(\frac{dx}{dt}\right)^2=E \quad \Rightarrow \quad T=\sqrt{2m}\int_A^B \frac{dx}{\sqrt{E-U(x)}}$

OBSERVE: Supomos que a velocidade do corpo é sempre positiva, ou seja, estamos calculando o tempo que ele leva para ir de A até B (metade do período).

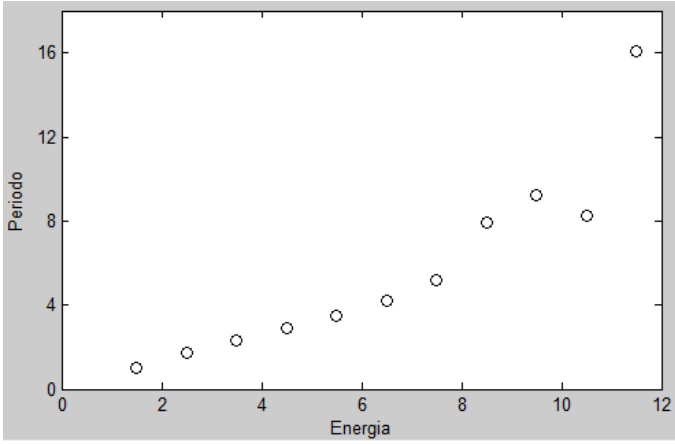



```
% POÇO DE POTENCIAL NÃO-HARMÔNICO
clear all
format short G
U = [-4.34e-6 1.606e-4 -9.75e-4 -0.0243 0.293 0.223 0.972]
x=-8:0.01:18;
y=polyval(U,x);
plot(x,y)
hold
```

$$U(x)=0,972+0,223x+0,293x^2-0,0243x^3-0,000975x^4+0,0001606x^5-0,00000434x^6$$

```
% ESCOLHA A ENERGIA TOTAL DO OSCILADOR
E=11.5;
% ENCONTRE OS EXTREMOS DO MOVIMENTO
U1=U
U1(7)=U1(7)- E
U1 = -4.34e-006  0.0001606  -0.000975  -0.0243   0.293    0.223  -10.528
r=roots(U1)
r =    18.317
       15.335
       10.425 +     4.4694i
       10.425 -     4.4694i
      -11.818
      -5.6797
% OS LIMITES DE OSCILAÇÃO
A=r(6)
  A = -5.6797
B=r(2)
  B = 15.335

% CALCULO DO PERIODO DE OSCILAÇÃO
% SUPONHA m = ½
% OBSERVE A SINTAXE DA FUNÇÃO ANÔNIMA ABAIXO
% O PONTO INDICA QUE A OPERAÇÃO DEVE SER EFETUADA
%    EM CADA ELEMENTO DO VETOR
f2 = (@(x) sqrt(E-polyval(U,x)).^-1);
T=quadgk(f2,A,B)
  T = 16.097
```

```
% CALCULANDO O PERÍODO EM FUNÇÃO DA ENERGIA
ind=1;
% BASTA REPETIR OS PASSOS ABAIXO,
% MUDANDO O VALOR DA ENERGIA
ind=ind+1
  ind=2
E(ind)=10.5
  E =  11.5  10.5
U1=U;
U1(7)=U1(7)-E(ind);
r=roots(U1)
 r =  19.139
     11.939 +     3.8786i
     11.939 -     3.8786i
     11.303
    -11.905
    -5.4099
% CUIDADO PARA TOMAR AS RAIZES CORRETAS
A=[A r(6)]
  A = -5.6797   -5.4099
B = [B r(4)]
 B =   15.335    11.303
f2 = (@(x) sqrt(E(ind)-polyval(U,x)).^-1);
T=[T quadgk(f2,A(ind),B(ind))]
T =   16.097   8.2692

% REPETINDO OS PASSOS ACIMA, OBTEMOS
E =  11.5   10.5   9.5   8.5   7.5   6.5   5.5   4.5   3.5   2.5   1.5
A = -5.6797  -5.4099  -5.1316  -4.8417  -4.5366  -4.2111  -3.8578  -3.4651
    -3.0124  -2.4554  -1.6482
B =  15.335   11.303   8.4776   7.1088   6.0981   5.2409   4.4581   3.7038
     2.9382   2.1048   1.0499
T =  16.097   8.2692   9.1988   7.9123   5.2083   4.2266   3.5042   2.8898
      2.315   1.7224   0.99353
```



SOLUÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS

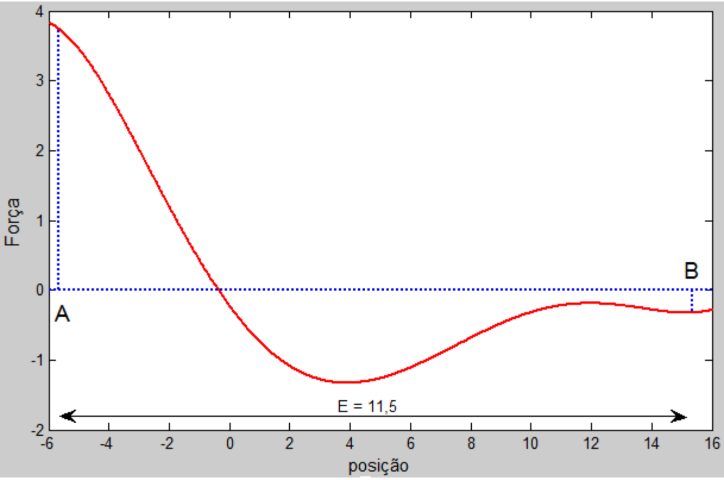
EXEMPLO: Determinar a forma de onda produzida pelo oscilador não-harmônico conservativo

Usando a segunda lei de Newton, devemos ter $F = ma \quad \Rightarrow \quad \dfrac{d^2x}{dt^2} = \dfrac{F(x)}{m}$, onde $F(x) = -\dfrac{dU}{dx}$

Essa equação diferencial de segunda ordem é equivalente a um sistema de duas equações diferenciais de primeira ordem:

$$\begin{cases} x_1' = x_2 \\ x_2' = \dfrac{F(x_1)}{m} \end{cases} \quad \text{com as condições iniciais} \quad \begin{cases} x_1(0) = x_0 = \text{posição inicial} \\ x_2(0) = v_0 = \text{velocidade inicial} \end{cases}$$

Se especificarmos a energia E, as condições iniciais devem satisfazer $v_0 = \pm\sqrt{\dfrac{2}{m}\left(E - U(x_0)\right)}$




```
% ENCONTRANDO A FORÇA EM FUNÇÃO DA POSIÇÃO
clear all
format short G
U = [-4.34e-6 1.606e-4 -9.75e-4 -0.0243 0.293 0.223 0.972]
F=-polyder(U)
x=-8:0.01:18;
z=polyval(F,x);
plot(x,z)
```

```
% INTEGRANDO A EQUAÇÃO DIFERENCIAL
clear all
format short G
% ESCOLHA A ENERGIA
E=11.5
% CONDIÇÕES INICIAIS
% VAMOS ESCOLHER A POSIÇÃO INICIAL
% CUIDADO PARA NÃO ULTRAPASSAR O VALOR DA ENERGIA!
x0=15.0
U = [-4.34e-6 1.606e-4 -9.75e-4 -0.0243 0.293 0.223 0.972]
U0=polyval(U,x0)
v0 = -2*sqrt(E-U0)
IC = [x0 v0]
% USANDO ode45
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4]);
[T,X] = ode45(@oscilador,[0 40],IC,options);
plot(T,X(:,1))
```

```
function dx = oscilador(t,x)
%estamos supondo m = ½
dx = zeros(2,1);    % a column vector
dx(1) = x(2);
dx(2) = 2*force(x(1));
end
```

```
function Fr = force(x)
U = [-4.34e-6 1.606e-4 -9.75e-4 -0.0243 0.293 0.223 0.972];
F = -polyder(U);
Fr = polyval(F,x);
end
```