ΠΑΡΕΜΒΟΛΗ- Matlab

```
function yint = Lagrange(x,y,xx)
% input:
% x = independent variable
% y = dependent variable
% xx = value of independent variable at which the
% interpolation is calculated
% output:
% yint = interpolated value of dependent variable
n = length(x);
if length(y)~=n, error('x and y must be same length'); end
s = 0;
for i = 1:n
product = y(i);
for j = 1:n
if i ~= j
product = product*(xx-x(j))/(x(i)-x(j));
end
end
s = s+product;
end
yint = s;
```

Παράδειγμα Lagrange

>> R=[1101.0 911.3 636 451.1]
R =

1101.00 911.30 636.00 451.10

>> T=[25.113 30.131 40.120 50.128]
T =

25.113 30.131 40.120 50.128

>> Lagrange(R,T,754.8)
ans = 35.242

Newton με διαιρεμένες διαφορές

```
function yint = Newtint(x,y,xx)
% x = independent variable
% y = dependent variable
% xx = value of independent variable at which interpolation is calculated
n = length(x);
if length(y)~=n, error('x and y must be same length'); end
b = zeros(n,n);
% assign dependent variables to the first column of b.
b(:,1) = y(:); % the (:) ensures that y is a column vector.
for j = 2:n
for i = 1:n-j+1
b(i,j) = (b(i+1,j-1)-b(i,j-1))/(x(i+j-1)-x(i));
end
end
% use the finite divided differences to interpolate
xt = 1;
yint = b(1,1);
for j = 1:n-1
xt = xt*(xx-x(j));
yint = yint+b(1,j+1)*xt;
end
```

Παράδειγμα

```
>> >> R=[1101.0 911.3 636 451.1]
R =

   1101.00    911.30    636.00    451.10

>> T=[25.113 30.131 40.120 50.128]
T =

   25.113   30.131   40.120   50.128

>> Newtint(R,T,754.8)
ans =  35.242
```

```
function [yi, p, pval] = newtForwint(x, y, xi)
%    NEWTINT(X,Y,XI,C) interpolates to find YI, the value
%    of the underlying function Y at the point XI, using Newton's forward interpolation
%     P, the coefficients f the calculated interpolating polynomial,
%     PVAL, the maximum degree of the interpolating polynomial
pval = length(x)*(0+1) - 1;
n = length(x)-1;
D = zeros(n+1,n+1);
D(:,1) = y(:);
h = x(2)-x(1);
for i = 1:n
   for j = 1:i
      D(i+1,j+1) = D(i+1,j)-D(i,j);
            end
end
% forward difference interpolation
p = diag(D);
q = (xi-x(1))/h;
yi = p(1);
for i = 1:length(p)-1
            term = 1;
            for k = 0:(i-1)
                        term = term*(q-k);
            end
            term = term/factorial(i);
            yi = yi + term*p(i+1);
end
```

Χρήση Newton με προς τα εμπρός διαφορές

```
>> x= [0 1 2 3]
x =

   0   1   2   3

>> y= [1 0 1 10]
y =

    1    0    1   10

>> newtForwint(x, y, 0.5)
ans =  0.62500
```

```
function [yi, p, pval] = newtBackint(x, y, xi)
%    NEWTINT(X,Y,XI,C) interpolates to find YI, the value
%    of the underlying function Y at the point XI, using Newton's backward interpolation formula.
%     P, the coefficients f the calculated interpolating polynomial,
%     PVAL, the maximum degree of the interpolating polynomial
pval = length(x)*(0+1) - 1;
n = length(x)-1;
D = zeros(n+1,n+1);
D(:,1) = y(:);
h = x(2)-x(1);
for i = 1:n
  for j = 1:i
    D(i+1,j+1) = D(i+1,j)-D(i,j);
  end
end
    % backward difference interpolation
    p = D(n+1,:).';
    q = (xi-x(n+1))/h;
    yi = p(1);
    for i = 1:length(p)-1
      term = 1;
      for k = 0:(i-1)
        term = term*(q+k);
      end
      term = term/factorial(i);
      yi = yi + term*p(i+1);
    end
```

Χρήση Newton με προς τα πίσω διαφορές

```
>> x= [0 1 2 3]
x =

   0   1   2   3

>> y= [1 0 1 10]
y =

    1    0    1   10

>> newtBackint(x, y, 0.5)
ans =  0.62500
```

Ενσωματωμένες συναρτήσεις στο Octave

```
xf=0:0.05:10; yf = sin(2*pi*xf/5);
xp=0:10; yp = sin(2*pi*xp/5);
lin=interp1(xp,yp,xf,"linear");
spl=interp1(xp,yp,xf,"spline");
cub=interp1(xp,yp,xf,"pchip");
near=interp1(xp,yp,xf,"nearest");
plot(xf,yf,"r",xf,near,"g",xf,lin,"b",xf,cub,"c",xf,spl,"m
",xp,yp,"r*");
legend ("original","nearest","linear",
"pchip","spline")
```

Χρήση ενσωματωμένων συναρτήσεων

```
>> x=[0:5]
x =

   0   1   2   3   4   5

>> y=[15 10 9 6 2 0]
y =

   15   10    9    6    2    0

>> lin=interp1(x,y,3.5,"linear")
lin =  4
>> spl=interp1(x,y,3.5,"spline")
spl =  3.9417
>> cub=interp1(x,y,3.5,"pchip")
cub =  3.9048
```

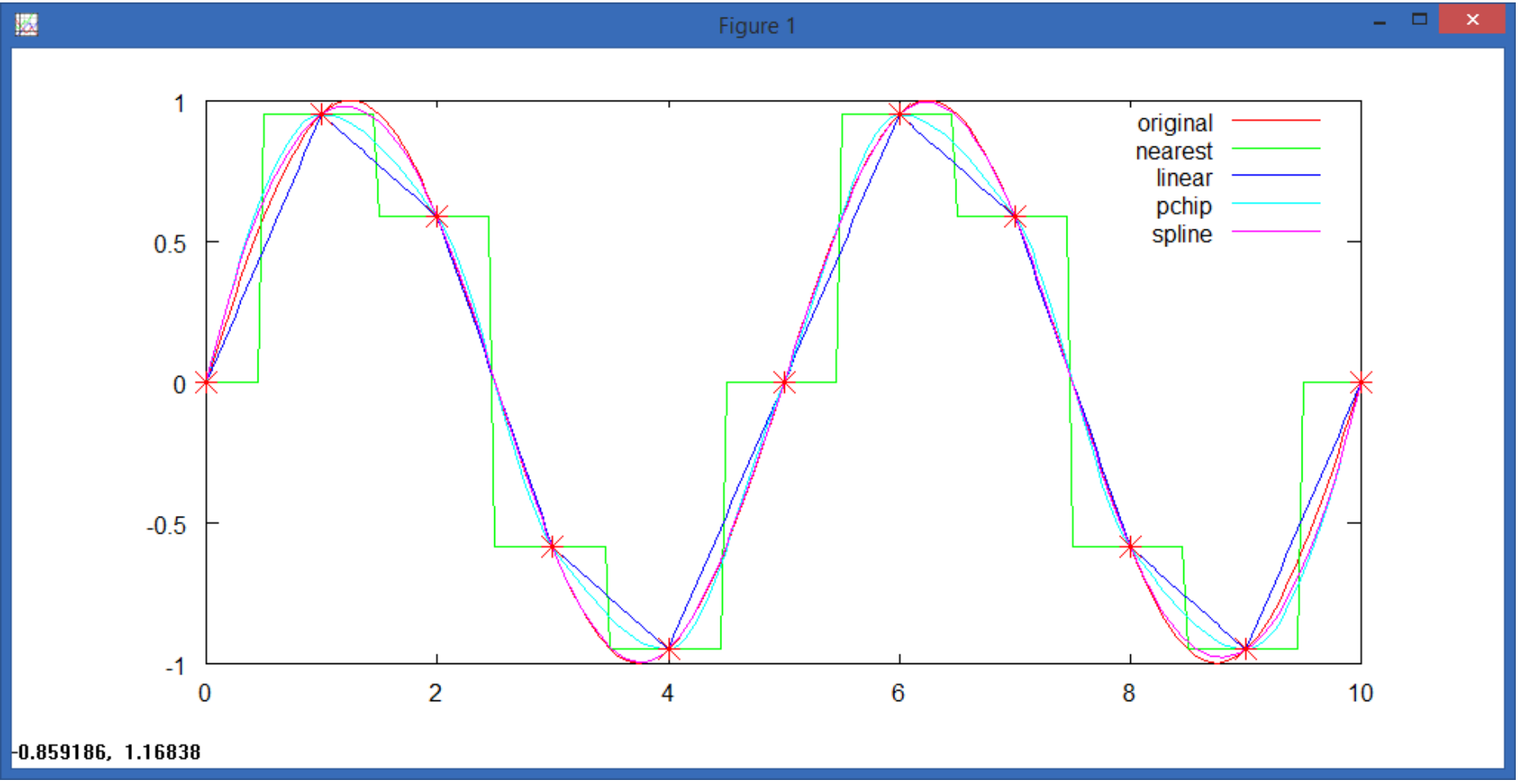
Figure 1
original
nearest
linear
pchip
spline
1
0.5
0
-0.5
-1
0
2
4
6
8
10
-0.859186, 1.16838