הפקולטה להנדסה אזרחית וסביבתית

הטכניון - מכון טכנולוגי לישראל

שיטות נומריות למהנדסים (019003)

Preparation Homework #7: Interpolation

Recap question: Interpolation

1. In a few sentences, explain what is the difference and also the pros and cons of the two methods: Newton polynomial, Lagrange polynomial for interpolation.
2. Use either Newton or Lagrange to find the polynomial for $f(x) = (1 + x^2)^{-1}$ at ten points equally spaced from -5 to 5. Plot the original function and the interpolation polynomial.
3. Is this a good polynomial for interpolation? If not, why and what should be used?

Alon Spinner

305184335

1.

Langrange Polynomial:

$$L(x) = \sum_{j=0}^{k} y_j l_j(x)$$

$$l_j(x_i) = \prod_{\substack{m=0 \\ m\neq j}}^{k} \frac{x_i - x_m}{x_j - x_m} \Rightarrow l_j(x_j) \equiv 1$$

$$L(x_j) = y_j$$

(+) linear time to compute coeff (from the looks)

(−) numerically unstable because of division

Netwon:

$$f(x) = \sum_{i=0}^{n} a_i n_i(x)$$

$$n_i(x) = \prod_{j=0}^{i-1} (x - x_j)$$

(+) incremental way for computing a_i

$\Rightarrow$ adding points is easy

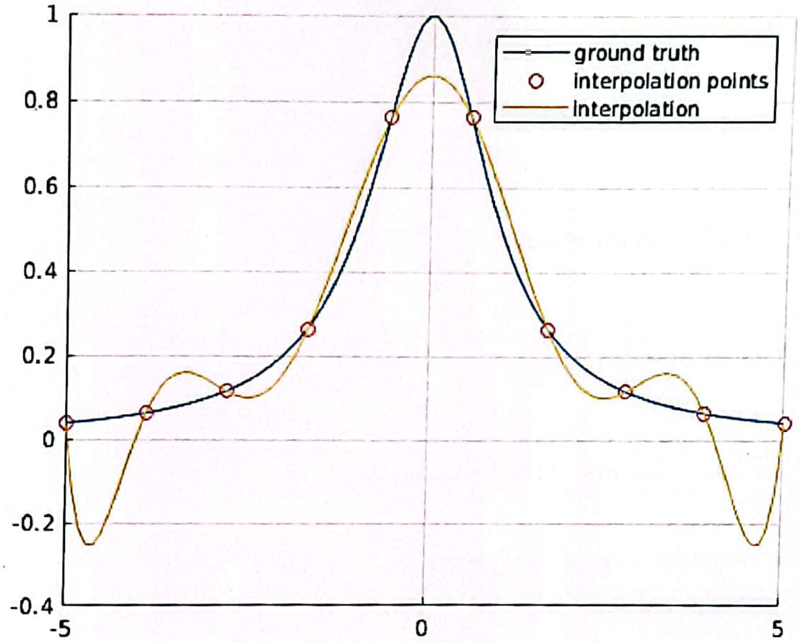
(+) quadratic time for coefficient calculation

Scanned with CamScanner

Q2

```
N = 10;
f = @(x) (1+x.^2).^-1;

xgt = linspace(-5,5,10*N);
ygt = f(xgt);


xSamp = linspace(-5,5,N);
ySamp = f(xSamp);

xIntrp = xgt;
yIntrp = zeros(size(ygt));
for kk = 1:length(xIntrp)
    yIntrp(kk) = lagrange(xIntrp(kk),xSamp,ySamp);
end

fig = figure;
ax = axes(fig); hold(ax,'on'); grid(ax,'on');
plot(ax,xgt,ygt);
scatter(ax,xSamp,ySamp,'marker','o');
plot(ax,xIntrp,yIntrp);
legend('ground truth','interpolation points','interpolation');


function y=lagrange(x,pointx,pointy)        <-- Lagrange
%
%LAGRANGE   approx a point-defined function using the Lagrange polynomial interpolation
%
%      LAGRANGE(X,POINTX,POINTY) approx the function definited by the points:
%      P1=(POINTX(1),POINTY(1)), P2=(POINTX(2),POINTY(2)), ..., PN(POINTX(N),POINTY(N))
%      and calculate it in each elements of X
%
%      If POINTX and POINTY have different number of elements the function will return the NaN value
%
%      function wrote by: Calzino
%      7-oct-2001
%
n=size(pointx,2);
L=ones(n,size(x,2));
if (size(pointx,2)~=size(pointy,2))
   fprintf(1,'\nERROR!\nPOINTX and POINTY must have the same number of elements\n');
   y=NaN;
else
   for i=1:n
      for j=1:n
         if (i~=j)
            L(i,:)=L(i,:).*(x-pointx(j))/(pointx(i)-pointx(j));
         end
      end
   end
   y=0;
   for i=1:n
      y=y+pointy(i)*L(i,:);
   end
```

```
end
end
```



Published with MATLAB® R2021b

this is not a good interpolation. 3
using less points for L' interpolation
would have been better... or
a stabler method like spline