אנליזה נומרית / תרגיל בית 4#

אריאל סטולרמן

(1)

$f(x)=\sin\left(\frac{\pi}{2}x\right), x_j\in\{-3,-2,-1,-0,1,2,3\}$

(a) Newton's interpolation polynomial:

$$\begin{cases} f[-3]=\sin\left(-\frac{3\pi}{2}\right)=1\\ f[-2]=\sin(-\pi)=0\\ f[-1]=\sin\left(-\frac{\pi}{2}\right)=-1\\ f[0]=\sin(0)=0\\ f[1]=\sin\left(\frac{\pi}{2}\right)=1\\ f[2]=\sin(\pi)=0\\ f[3]=\sin\left(\frac{3\pi}{2}\right)=-1 \end{cases} \Rightarrow \begin{cases} f[-3,-2]=\frac{1-0}{-3+2}=-1\\ f[-2,-1]=\frac{0+1}{-2+1}=-1\\ f[-1,0]=\frac{-1-0}{-1-0}=1\\ f[0,1]=\frac{0-1}{0-1}=1\\ f[1,2]=\frac{1-0}{1-2}=-1\\ f[2,3]=\frac{0+1}{2-3}=-1 \end{cases} \Rightarrow \begin{cases} f[-3,-2,-1]=\frac{-1+1}{-1+3}=0\\ f[-2,-1,0]=\frac{-1-1}{0+2}=-1\\ f[-1,0,1]=\frac{1-1}{1+1}=0\\ f[0,1,2]=\frac{1+1}{2-0}=1\\ f[1,2,3]=\frac{-1+1}{3-1}=0 \end{cases} \Rightarrow$$

$$\begin{cases} f[-3,-2,-1,0]=\frac{0+1}{0+3}=\frac{1}{3}\\ f[-2,-1,-0,1]=\frac{-1-0}{1+2}=-\frac{1}{3}\\ f[-1,0,1,2]=\frac{0-1}{2+1}=-\frac{1}{3}\\ f[0,1,2,3]=\frac{1-0}{3-0}=\frac{1}{3} \end{cases} \Rightarrow \begin{cases} f[-3,-2,-1,0,1]=\frac{-\frac{1}{3}-\frac{1}{3}}{1+3}=-\frac{1}{6}\\ f[-2,-1,0,1,2]=0\\ f[-1,0,1,2,3]=\frac{\frac{1}{3}+\frac{1}{3}}{3+1}=\frac{1}{6} \end{cases} \Rightarrow \begin{cases} f[-3,-2,-1,0,1,2]=\frac{\frac{1}{6}}{2+3}=\frac{1}{30}\\ f[-2,-1,0,1,2,3]=\frac{\frac{1}{6}}{3+2}=\frac{1}{30} \end{cases} \Rightarrow$$

$f[-3,-2,-1,0,1,2,3]=0$

$\Rightarrow$

$$P_5(x)=1-1(x+3)+\frac{1}{3}(x+3)(x+2)(x+1)-\frac{1}{6}x(x+3)(x+2)(x+1)+\frac{1}{30}x(x+3)(x+2)(x+1)(x-1)=$$

$$=\frac{1}{30}x^5-\frac{1}{2}x^3+\frac{22}{15}x$$

(b) Taylor's polynomial:

- $f(x)=\sin\left(\frac{\pi}{2}x\right), f(0)=0$
- $f'(x)=\frac{\pi}{2}\cos\left(\frac{\pi}{2}x\right), f'(0)=\frac{\pi}{2}$
- $f''(x)=-\frac{\pi^2}{4}\sin\left(\frac{\pi}{2}x\right), f''(0)=0$
- $f^{(3)}(x)=-\frac{\pi^3}{8}\cos\left(\frac{\pi}{2}x\right), f^{(3)}(0)=-\frac{\pi^3}{8}$
- $f^{(4)}(x)=\frac{\pi^4}{16}\sin\left(\frac{\pi}{2}x\right), f^{(4)}(0)=0$
- $f^{(5)}(x)=\frac{\pi^5}{32}\cos\left(\frac{\pi}{2}x\right), f^{(5)}(0)=\frac{\pi^5}{32}$

$$\Rightarrow T_5(x)=\sum_{n=0}^{5}\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n=\frac{\pi}{2}x-\frac{\pi^3}{48}x^3+\frac{\pi^5}{3840}x^5\cong 1.5707x-0.6459x^3+0.0796x^5$$

(c) For $x=\frac{5}{3}$ $\left(f(x)=\sin\left(\frac{5\pi}{6}\right)=\frac{1}{2}\right)$:

```
>> p5=[1/30 0 1/2 0 22/15 0]
>> polyval(p5,5/3)
ans =
   5.1879
```

```
>> t5=[pi.^5/3840 0 -pi.^3/48 0 pi/2 0]
>> polyval(t5,5/3)
ans =
  0.652273085089210
```

(d)

השגיאה של אינטרפולציית ניוטון היא 0.518~ לעומת שגיאת פולינום טיילור שהיא רק 0.15~, ולכן השגיאה של פולינום טיילור יותר קטנה.

(2)

$$f(x)=\sin(3x)\,,x\in[0,5]\Rightarrow h=\frac{b-a}{n}=\frac{5}{n}$$

$$f(x)=\sin(3x)\,,f'(x)=3\cos(3x)\,,f''(x)=-9\sin(3x)$$

$$\max_{x\in[0,5]}|\sin(3x)-P(x)|\le\frac{h^2}{8}\max_{x\in[0,5]}|-9\sin(3x)|=\frac{h^2}{8}\cdot 9\cdot\underbrace{\max_{x\in[0,5]}|\sin(3x)|}_{=1}=\frac{9}{8}h^2$$

נחסום את השגיאה ונמצא את n :

$$\frac{9}{8}h^2\le 10^{-6}\Leftrightarrow\frac{9}{8}\cdot\frac{25}{n^2}\le 10^{-6}\Leftrightarrow n^2\ge\frac{9\cdot 25}{8\cdot 10^{-6}}\Leftrightarrow n\ge\frac{3\cdot 5}{2\sqrt{2}\cdot 10^{-3}}\cong 5303.3\Rightarrow n\ge 5304$$

(3)

(a) הנוסחה הכללית לשגיאה עבור אינטרפולציה על נקודות במרווחים קבועים היא :

$$\max_{x\in[a,b]}|f(x)-P_n(x)|\le\max_{x\in[a,b]}\left|f^{(n+1)}(x)\right|\cdot\frac{1}{4(n+1)}h^{n+1}$$

ובמקרה זה :

$$\max_{x\in[a,b]}|f(x)-P_2(x)|\le\max_{x\in[a,b]}\left|f^{(3)}(x)\right|\cdot\frac{1}{4\cdot 3}h^3=\frac{1}{12}\max_{x\in[a,b]}\left|f^{(3)}(x)\right|h^3$$

(b)

$$f^{(3)}(x)=-27\cos(3x)\Rightarrow\max_{x\in[0,5]}|\sin(3x)-P_2(x)|\le\frac{1}{12}\cdot\max_{x\in[0,5]}|-27\cos(3x)|\cdot\left(\frac{5}{n}\right)^3=\frac{125\cdot 27}{12\cdot n^3}\le 10^{-6}\Leftrightarrow$$

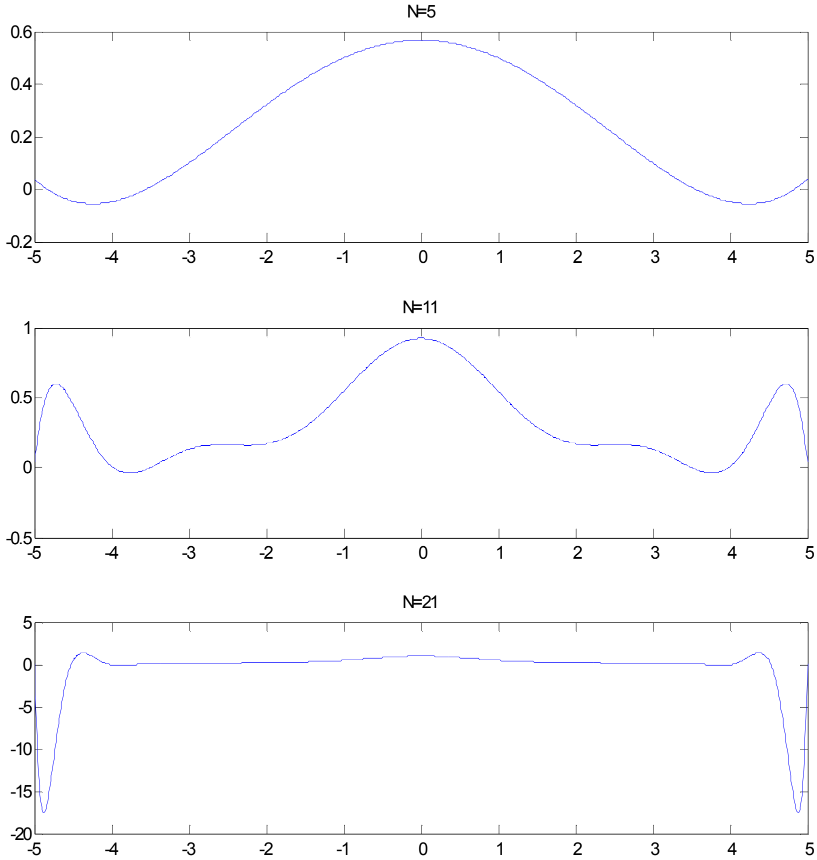
$$n^3\ge\frac{125\cdot 27}{12\cdot 10^{-6}}\Leftrightarrow n\ge\frac{5\cdot 3}{\sqrt[3]{12}\cdot 10^{-2}}=655.18\Rightarrow n\ge 656$$

במקרה זה נזדקק לפחות דגימות כדי להשיג את חסם הטעות הנדרש.

(4)

$$f(x)=\frac{1}{1+x^2},x\in[-5,5]:$$

```
>> x_Nis5 = -5:(10/5):5
>> x_Nis11 = -5:(10/11):5
>> x_Nis21 = -5:(10/21):5
>> y_Nis5 = 1./(1+x_Nis5.^2)
>> y_Nis11 = 1./(1+x_Nis11.^2)
>> y_Nis21 = 1./(1+x_Nis21.^2)
>> p_Nis5 = polyfit(x_Nis5,y_Nis5,5)
>> p_Nis11 = polyfit(x_Nis11,y_Nis11,11)
>> p_Nis21 = polyfit(x_Nis21,y_Nis21,21)
// plotting:
>> x=-5:0.01:5
>> subplot(3,1,1)
>> title('N=5')
>> plot(x,polyval(p_Nis5,x))
>> subplot(3,1,2)
>> title('N=11')
>> plot(x,polyval(p_Nis11,x))
>> subplot(3,1,3)
>> title('N=21')
>> plot(x,polyval(p_Nis21,x))
```



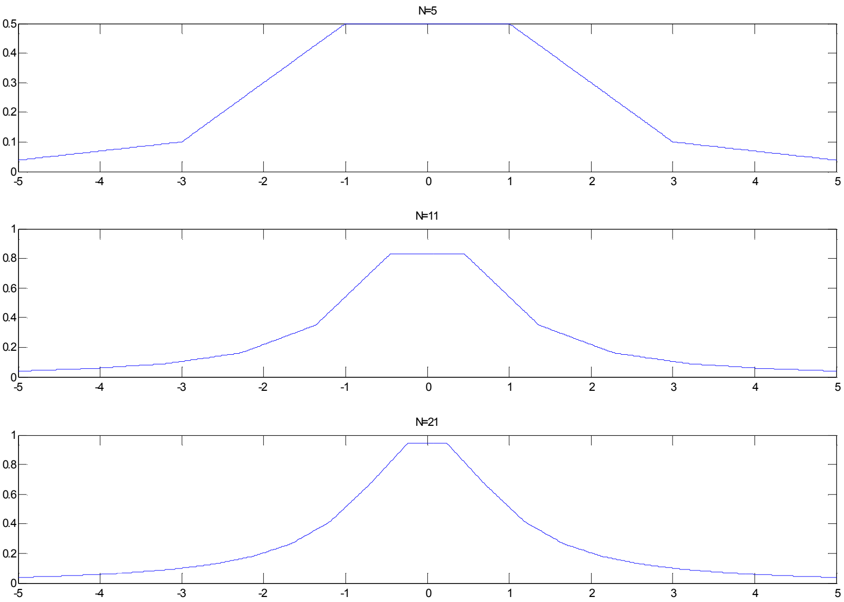
(c) חישוב שגיאת אינטרפולציה מקסימלית :

```
>> f = 1./(1+x.^2)
>> max(abs(f-polyval(p_Nis5,x)))
ans =
   0.4327
>> max(abs(f-polyval(p_Nis11,x)))
ans =
   0.5567
>> max(abs(f-polyval(p_Nis21,x)))
ans =
  17.6020
```

(d)

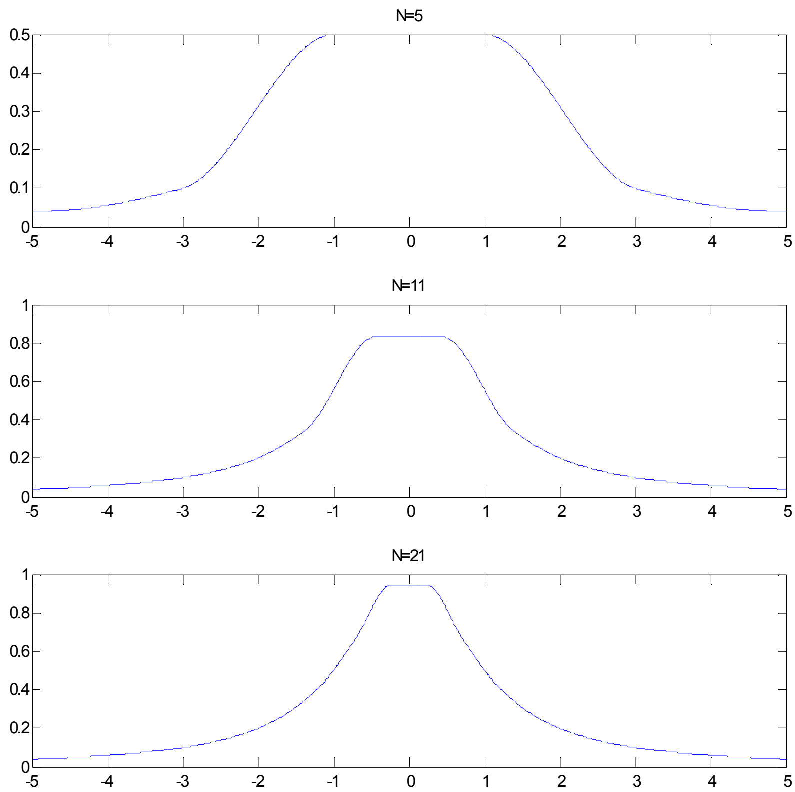
```
>>linear_Nis5 = interp1(x_Nis5,y_Nis5,x,'linear')
>>linear_Nis11 = interp1(x_Nis11,y_Nis11,x,'linear')
>>linear_Nis21 = interp1(x_Nis21,y_Nis21,x,'linear')
>> #plot…
>> max(abs(f-interp1(x_Nis5,y_Nis5,x,'linear')))
ans =
   0.5000
>> max(abs(f-interp1(x_Nis11,y_Nis11,x,'linear')))
ans =
   0.1712
>> max(abs(f-interp1(x_Nis21,y_Nis21,x,'linear')))
ans =
   0.0536
```

שרטוט בעמוד הבא



(e)

```
>> cubic_Nis5 = interp1(x_Nis5,y_Nis5,x,'pchip')
>> cubic_Nis11 = interp1(x_Nis11,y_Nis11,x,'pchip')
>> cubic_Nis21 = interp1(x_Nis21,y_Nis21,x,'pchip')
>> #plot...
>> max(abs(f-interp1(x_Nis5,y_Nis5,x,'pchip')))        # answer: 0.5000
>> max(abs(f-interp1(x_Nis11,y_Nis11,x,'pchip')))      # answer: 0.1712
>> max(abs(f-interp1(x_Nis21,y_Nis21,x,'pchip')))      # answer: 0.0536
```



(5)

```
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/LagrangeInterpolatingPo
lynomial.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
```

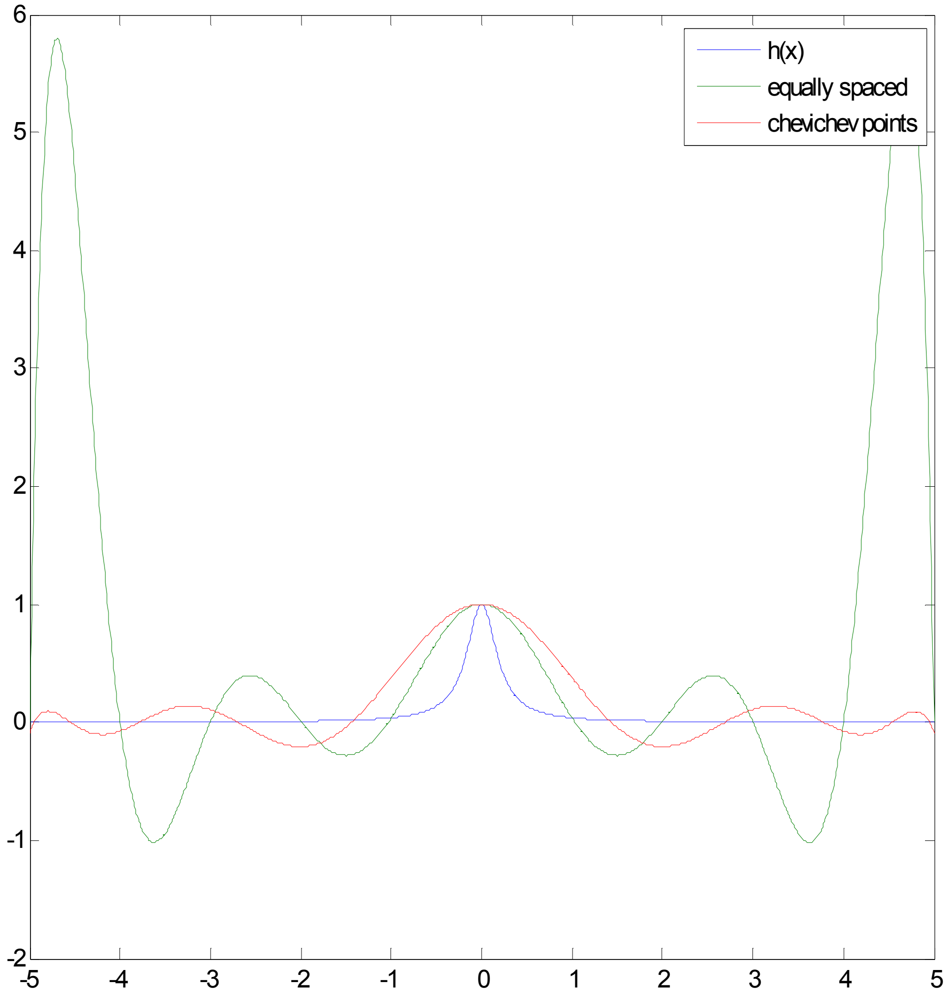
(a)

```
>> x_eq = -5:10/(11-1):5
>> y_eq = 1./(1+25*x_eq.^2)
>> [p_eq,r_eq_s_eq] = lagrangepoly(x_eq,y_eq)
>> p_eq =
  -0.000065550827217  -0.000000000000000   0.003607917530022   0.000000000000000
  -0.067202812944167  -0.000000000000000   0.503824186591548  -0.000000000000000
  -1.401702201888648  -0.000000000000000   1.000000000000000
```

(b)

```
>> for i=1:11
       x_chev(i) = 0.5*(-5+5) + 0.5*(5-(-5))*cos((2*i-1)*pi./(2*11))
end
>> y_chev = 1./(1+25*x_chev.^2)
>> [p_chev,r_chev_s_chev] = lagrangepoly(x_chev,y_chev)
p_chev =
   -0.000009234352441   0.000000000000000   0.000635231104406  -0.000000000000000
   -0.015896952501884   0.000000000000000   0.174230882481253  -0.000000000000000
   -0.781946933429506   0.000000000000001   1.000000000000000
```

```
>> x = -5:0.01:5
>> y1 = 1./(1+25*x.^2)
>> y2 = polyval(p_eq,x)
>> y3 = polyval(p_chev,x)
>> plot(x,y1,x,y2,x,y3)
```



(d) נראה כי קירוב לפי נקודות צ׳ביצ׳ב נותן שגיאה קטנה יותר מאשר נקודות במרווחים שווים (כמצופה).