HW5

Mohammad Rahimi

Problem solved

#6

#18

#21

#23

6

$Sin(x)$ $\quad 0 \le x \le \pi/2$ $\quad x_n - x_{n-1} = h$ $\quad$ total error $= 10^{-6}$ $\quad h = ?$

howmany Significant digits ?

total interpolation error (linear) $\le \underbrace{\frac{h^2}{8} Sup\,|f''(\xi)|}_{\text{truncation}} + \underbrace{\varepsilon}_{\text{rounding}}$

$\rightarrow f''(x) = -Sin(x)$

if $0 \le x \le \pi/2 \rightsquigarrow Sup\,|-Sin(x)| = 1$ $\Big\}$ $\frac{h^2}{8} + \varepsilon = 10^{-6}$

best case is when we have same level of contribution from both rounding & truncation error : $\varepsilon_{truncation} \simeq \varepsilon_{rounding} \rightarrow 2\frac{h^2}{8} = 10^{-6}$

$\frac{h^2}{4} = 10^{-6} \rightsquigarrow \frac{h}{2} = 10^{-3} \rightarrow \boxed{h = 2 \times 10^{-3}}$

ε (Error rounding) $= \frac{10^{-6}}{2}$ $\quad \varepsilon = 5 \times 10^{-7} \rightarrow \varepsilon = \pm\, 0.0000005$

$\rightarrow$ which makes sense to keep 6 decimals

18

inverse interpolation → replacing dependent & independent variables

```
% since we are doing inverse interpolattion i replaced the
dependent and
% independent variables
x=[2,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9];
j0=[0.223891,0.166607,0.110362,0.05554,0.002508,-0.04838,-
0.0968,-0.14245,-0.18504,-0.22431];
xxx=Newtint(j0,x,0)
function yint = Newtint(x,y,xx)
% Newtint: Newton interpolating polynomial
% yint = Newtint(x,y,xx): Uses an (n - 1)-order Newton
interpolating polynomial based on n
% data points (x, y) to determine a value of the dependent
variable (yint)@ a given value of
% the independent variable, xx.
% input:
% x = independent variable; % y = dependent variable
% xx = value of independent variable at which
interpolation is calculated
% output: yint = interpolated value of dependent variable
% compute the finite divided differences in the form of a
difference table
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
xt = 1;
yint = b(1,1);
for j = 1:n-1
 xt = xt*(xx-x(j));
 yint = yint+b(1,j+1)*xt;
end
end
```

using the Matlab code from Lecture 17

inputting values from the table

$$\bar{J}_0^{-1}(0) = 2.4048$$

note on accuracy: Since increasing the order of polynomial increase the error towards the ends of x interval, it is safe to say using all the data points yeilds the most accurate result in the middle of the interval

$x = 2.4048255577.$

relative error: $\dfrac{|x_{actual} - x_{est}|}{x_{actual}} = 1.063 \times 10^{-5}$

absolute error = 0.0000255577

21

x	-2	-1	0	1	2	3
$p(x)$	-5	1	1	1	7	25

as mentioned in the question since the data comes from a polynomial $P_n(x) \equiv f(x)$ & it is unique which means the coefficient of higher order terms will cancel each other out and become zero using Newton interpolation scheme

			First	Second	Third	Fourth
i	x_i	$f(x_i)$	$f[x_{i+1}, x_i]$	$f[x_{i+2}, x_{i+1}, x_i]$	$f[x_{i+3}, \dots x_i]$	$f[x_{i+4} \dots x_i]$
0	-2	-5	6	-3	1	0
1	-1	1	0	0	1	0
2	0	1	0	3	1	0
3	1	1	6	6	0	0
4	2	7	18	0	0	0
5	3	25	0	0	0	0

Since all the fourth column became zero the 5th column is also zero

So it's a third-order polynomial

23

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

to solve this I used the newton method code from the Lecture 17 to construct the $P_{10}(x)$ interpolation with 11 data point between [-5, 5] Then used the interpolation function & actual function to plot the values also plotted the absolute error value.

As can be seen the error is maximized toward the ends of interval [-5, 5]

```
x=linspace(-5,5,11)
for i= 1:length(x)
 y(i)=f(x(i));
end
x1=linspace(-5,5,1000)
for i=1:length(x1)
yap(i)=Newtint(x,y,x1(i));
yact(i)=f(x1(i));
y1(i)= abs(Newtint(x,y,x1(i))-f(x1(i)));
end
ff= figure()
plot (x1,yap)
hold on
box on
plot(x1,yact)
xlabel('x','FontWeight','b')
ylabel('y','FontWeight','b')
legend('interpolation', 'actual function')
f2= figure()
plot(x1,y1)
xlabel('x','FontWeight','b')
ylabel('absolute error','FontWeight','b')
function yint = Newtint(x,y,xx)
% Newtint: Newton interpolating polynomial
% yint = Newtint(x,y,xx): Uses an (n - 1)-order Newton
interpolating polynomial based on n
% data points (x, y) to determine a value of the dependent
variable (yint)@ a given value of
% the independent variable, xx.
% input:
% x = independent variable; % y = dependent variable
% xx = value of independent variable at which
interpolation is calculated
% output: yint = interpolated value of dependent variable
% compute the finite divided differences in the form of a
difference table
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
xt = 1;
yint = b(1,1);
for j = 1:n-1
 xt = xt*(xx-x(j));
 yint = yint+b(1,j+1)*xt;
end
end
function y= f(x)
y=1/(1+x^2);
end
```