Notes from book by C. van Loan - Intro. to Scientific computing

Alternatives to Vandermond interp.

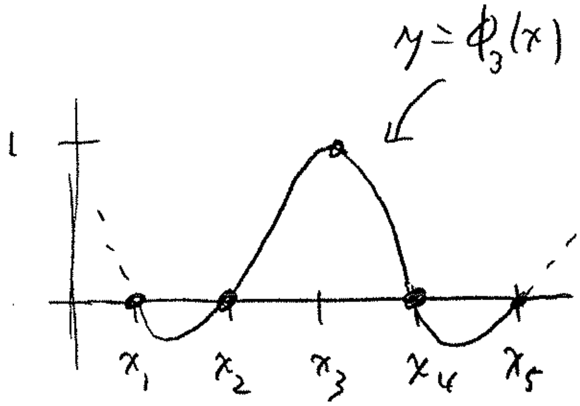
1.) Lagrange interpolation

$$\phi_i^{(m)}(x) = \prod_{\substack{j=1 \\ j\neq 1}}^{m} \frac{x - x_j}{x_i - x_j} = \text{poly. of degree } m-1$$

$$\phi_i(x_k) = \begin{cases} 0, & k \neq i \\ 1, & k = i \end{cases}$$

e.g. $m = 5$, $i = 3$

$$\phi_3^{(5)}(x) = \frac{(x - x_1)(x - x_2)(x - x_4)(x - x_5)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)(x_3 - x_5)}$$



Interpolating polynomial for $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ $(x_i \neq x_j, i \neq j)$ is just given by

$$\boxed{\phi(x) = \sum_{k=1}^{m} a_k \phi_k^{(m)}(x)} = (m-1)\text{st degree poly.}$$

Note: $\boxed{a_k = \phi(x_k) = y_k}$, $k = 1, 2, \ldots, m$

so coefficients a_k are easy to find.

Problem: $\phi(x)$ is inefficient to evaluate since each $\phi_k^{(m)}(x)$ must be evaluated.

2.) Newton representation - section 2.2

- More useful than Vandermonde representation
- More efficient than Lagrange representation

ex $m=3$ data $(x_1, y_1), (x_2, y_2), (x_3, y_3)$

Instead of basis $\phi_1(x)=1, \phi_2(x)=x, \phi_3(x)=x^2$

Use $\phi_1(x)=1, \phi_2(x)=x-x_1, \phi_3(x)=(x-x_1)(x-x_2)$

$(x_i \neq x_j, i \neq j)$

Newton representation

$$p_2(x) = \phi(x) = c_1\phi_1(x) + c_2\phi_2(x) + c_3\phi_3(x)$$

$$= c_1 + c_2(x-x_1) + c_3(x-x_1)(x-x_2)$$

Then we want

$$(*)\quad \begin{cases} y_1 = p_2(x_1) = c_1 \\ y_2 = p_2(x_2) = c_1 + c_2(x_2-x_1) \\ y_3 = p_3(x_3) = c_1 + c_2(x_3-x_1) + c_3(x_3-x_1)(x_3-x_2 \end{cases}$$

This is easy to solve for c_i's.

Just "forward-solve":

$$c_1 = y_1$$

$$c_2 = \frac{y_2 - c_1}{x_2 - x_1}$$

$$c_3 = \frac{y_3 - (c_1 + c_2(x_3 - x_1))}{(x_3 - x_1)(x_3 - x_2)}$$

Think matrix-vector! !!

(*) can be written as

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & x_2 - x_1 & 0 \\ 1 & x_3 - x_1 & (x_3 - x_1)(x_3 - x_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$\downarrow$ Subtract row 1 from rows 2 + 3 to zero the 1's.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & x_2 - x_1 & 0 \\ 0 & x_3 - x_1 & (x_3 - x_1)(x_3 - x_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 - y_1 \\ y_3 - y_1 \end{bmatrix}$$

$\downarrow$ Get 1's on diagonal

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & x_3 - x_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ \frac{y_2 - y_1}{x_2 - x_1} \\ \frac{y_3 - y_1}{x_3 - x_1} \end{bmatrix} =: \begin{bmatrix} y_1 \\ y_{21} \\ y_{31} \end{bmatrix}$$

MATLAB $\hookrightarrow$ $\begin{bmatrix} y_{21} \\ y_{31} \end{bmatrix} = \left(\begin{bmatrix} y_2 \\ y_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_1 \end{bmatrix}\right) ./ \left(\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_1 \end{bmatrix}\right) = (y(2:3) - y(1))./x(2:))

2.2.2 The General n case

Recursion process

$$p_{n-1}(x) = c_1 + c_2(x - x_1) + c_3(x-x_1)(x-x_2) + \dots + c_n(x-x_1)\cdots(x-x_{n-1})$$

$$y_1 = p_{n-1}(x_1) = c_1$$

$$y_i = p_{n-1}(x_i) = y_1 + c_2(x_i - x_1) + \dots + c_n(x_i - x_1)\cdots(x_i - x_{n-1})$$

$\downarrow$ $i = 2:n$

$$\frac{y_i - y_1}{x_i - x_1} = c_2(x_i - x_2) + \dots + c_n(x_i - x_2)\cdots(x_i - x_{n-1})$$

i.e. $q(x) := c_2 + c_3(x - x_2) + \dots + c_n(x-x_2)\cdots(x - x_{n-1})$

interpolates $\left(x_i, \dfrac{y_i - y_1}{x_i - x_1}\right)$ for $i = 2:n$

Note that then

$$p(x) = c_1 + (x - x_1)\, q(x)$$

Check $p(x_j) = c_1 + (x_j - x_1)\, q(x_j)$

$$= y_1 + (x_j - x_1)\frac{y_j - y_1}{x_j - x_1}$$

$$= y_j \quad \text{for } j = 2:n$$

and

$$p(x_1) = c_1 + (x_1 - x_1)\, q(x_1)$$

$$= y_1 \quad (j=1)$$

InterpNRecur is uses this observation. It is a recursive routine which calls itself – Not allowed in FORT.

(from p. 3)

Nonrecursive implementation

see sec 2.4.1

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & x_3-x_2 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_{21} \\ y_{31} \end{bmatrix}, \quad \text{where } y_{21} := \frac{y_2-y_1}{x_2-x_1} = f[x_1,x_2]$$

$$y_{31} := \frac{y_3-y_1}{x_3-x_1} = f[x_1,x_3]$$

Note $c_2 = y_{21}$

subtract 2nd eq from 3rd eq

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & x_3-x_2 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_{21} \\ y_{31}-y_{21} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_{21} \\ y_{321} \end{bmatrix} \quad \text{where } y_{321} := \frac{y_{31}-y_{21}}{x_3-x_2}$$

$$= f[x_1,x_2,x_3]$$

$$(= c_3,\ p.82)$$

Note now $\boxed{c_1 = y_1,\ c_2 = y_{21},\ c_3 = y_{321}}$

Compare with InterpN(x,y), $n = \text{length}(x) = 3$

$k=1$

$$y(k+1:n) = y(2:3) = \begin{pmatrix} y(2) \\ y(3) \end{pmatrix} \leftarrow \big(y(k+1:n) - y(k)\big)\ ./\ \big(x(k+1:n) - x(k)\big)$$

$$= \big(y(2:3) - y(1)\big)./\big(x(2:3) - x(1)\big)$$

$$= \begin{pmatrix} y(2)-y(1) \\ y(3)-y(1) \end{pmatrix} ./ \begin{pmatrix} x(2)-x(1) \\ x(3)-x(1) \end{pmatrix} = \begin{pmatrix} \frac{y(2)-y(1)}{x(2)-x(1)} \\ \frac{y(3)-y(1)}{x(3)-x(1)} \end{pmatrix} = \begin{bmatrix} f[x_1,x_2] \\ f[x_1,x_3] \end{bmatrix}$$

$k=2$

$$y(k+1:n) = y(3:3) = y(3) \leftarrow \big(y(k+1:n) - y(k)\big)./\big(x(k+1:n) - x(k)\big)$$

$$= \big(y(3) - y(2)\big)./\big(x(3) - x(2)\big)$$

$$= \frac{y(3)-y(2)}{x(3)-x(2)} = \frac{f[x_1,x_3]-f[x_1,x_2]}{x_3-x_2} = f[x_1,x_2,x_3] \text{ above}$$

end

$c = y$ then gives $c_1 = y_1$, $c_2 = y_{21}$, $c_3 = y_{321}$ again.

Compare InterpN with InterpN2

InterpN2(x, y) for $n = \text{length}(x) = 3$

$k=1$

$$y(k+1:n) = y(2:3) \leftarrow \big(y(k+1:n) - y(k:n-1)\big)./\big(x(k+1:n) - x(1:n-k)\big)$$

$$= \big(y(2:3) - y(1:2)\big)./\big(x(2:3) - x(1:2)\big)$$

$$= \begin{pmatrix} \dfrac{y(2)-y(1)}{x(2)-x(1)} \\[2ex] \dfrac{y(3)-y(2)}{x(3)-x(2)} \end{pmatrix} = \begin{pmatrix} f[x_1, x_2] \\[2ex] f[x_2, x_3] \end{pmatrix}$$

$k=2$

$$y(3:3) \leftarrow \big(y(3:3) - y(2:2)\big)./\big(x(3:3) - x(1:1)\big)$$

$$= \frac{y(3)-y(2)}{x(3)-x(1)} = \frac{f[x_2,x_3] - f[x_1,x_2]}{x_3 - x_1} = f[x_1,x_2,x_3]$$

(eq. 2.5)

end

Again $c = y$ gives $c_1 = f[x_1]$, $c_2 = f[x_1,x_2]$, $c_3 = f[x_1,x_2,x_3]$

InterpN2(x, y) is just computing the divided difference table by columns:

$$\begin{array}{cccc}
 & k=1 & k=3 & k=4 \\
 & \downarrow & \downarrow & \downarrow \\
f[x_1] & & & \\
f[x_2] & f[x_1,x_2] & & \\
f[x_3] & f[x_2,x_3] & f[x_1,x_2,x_3] & \\
f[x_4] & f[x_3,x_4] & f[x_2,x_3,x_4] & f[x_1,x_2,x_3,x_4]
\end{array}$$

if $n=4$ add this

~ 9/26/99 – 9/21/99

```
  function a = InterpV(x,y)
%
% Pre:
%   x:   column n-vector with distinct components.
%   y:   column n-vector.
%
% Post:
%   a:   column n-vector with the property that
%        if p(x) = a(1) + a(2)x + ... a(n)x^(n-1) then
%        p(x(i)) = y(i), i=1:n
%
  n = length(x);
  V = ones(n,n);
  for j=2:n
     % Set up column j.
     V(:,j) = x.*V(:,j-1);
  end
  a = V\y;
```

```
  function pval = HornerV(a,z)
%
% Pre:
%    a:   a column n-vector.
%    z:   a column m-vector.
%
% Post:
%    pval: a column m-vector with the property that if
%          p(x) = a(1) + .. +a(n)x^(n-1), then
%          pval(i) = p(z(i)) for i=1:m.
%
    n = length(a);
   m = length(z);
    pval = a(n)*ones(m,1);
    for k=n-1:-1:1
        pval = z.*pval + a(k);
    end
```

```
  function pval = HornerN(c,x,z)
%
% Pre:
%    c:   a vector.
%    x:   a vector with at least length(c)-1 components
%    z:   a vector.
%
% Post:
% pval:   a vector the same size as z with the property that if
%         p(x) = c(1) +  c(2)(x-x(1))+ ... + c(n)(x-x(1))...(x-x(n-1))
%         then pval(i) = p(z(i)) for i=1:m.
%

  n = length(c);
  pval = c(n)*ones(size(z));
  for k=n-1:-1:1
     pval = (z-x(k)).*pval + c(k);
  end
```

```
function c = InterpN(x,y)
%
% Pre:
%   x:  column n-vector with distinct components.
%   y:  column n-vector.
%
% Post:
%   c:  a column n-vector with the property that if
%       p(x) = c(1) + c(2)(x-x(1))+...+ c(n)(x-x(1))...(x-x(n-1))
%       p(x(i)) = y(i), i=1:n.
%
%
  n = length(x);
  for k = 1:n-1
     y(k+1:n) = (y(k+1:n)-y(k)) ./ (x(k+1:n) - x(k));
  end
  c = y;
```

```
function c = InterpNRecur(x,y)
%
% Pre:
%   x:  column n-vector with distinct components.
%   y:  column n-vector.
%
% Post:
%   c:  a column n-vector with the property that if
%       p(x) = c(1) + c(2)(x-x(1))+...+ c(n)(x-x(1))...(x-x(n-1))
%       p(x(i)) = y(i), i=1:n.
%
%
  n = length(x);
  c = zeros(n,1);
  c(1) = y(1);
  if n > 1
     c(2:n) = InterpNRecur(x(2:n),(y(2:n)-y(1))./(x(2:n)-x(1)));
  end
```

```
function c = InterpN2(x,y)
%
% Pre:
%   x:  column n-vector with distinct components.
%   y:  column n-vector.
%
% Post:
%   c:  a column n-vector with the property that if
%       p(x) = c(1) + c(2)(x-x(1))+...+ c(n)(x-x(1))...(x-x(n-1))
%       p(x(i)) = y(i), i=1:n.
%
%
  n = length(x);
  for k = 1:n-1
     y(k+1:n) = (y(k+1:n)-y(k:n-1)) ./ (x(k+1:n) - x(1:n-k));
  end
  c = y;
```

2.3.2 Accuracy

Q: How well does polynomial $\underbrace{\text{interpolant}}_{}^{p_{n-1}(x)}$ approximate $f(x)$?

Thm 2 Suppose $p_{n-1}(x)$ interpolates $f(x)$ at the distinct points $x_1, x_2, \ldots, x_n$. If f is n times continuously differentiable (that is, the nth derivative $f^{(n)}$ is continuous) on an interval I containing the x_i, the for any $x \in I$

$$\boxed{f(x) = p_{n-1}(x) + \frac{f^{(n)}(\eta)}{n!}(x-x_1)(x-x_2)\cdots(x-x_n)}$$

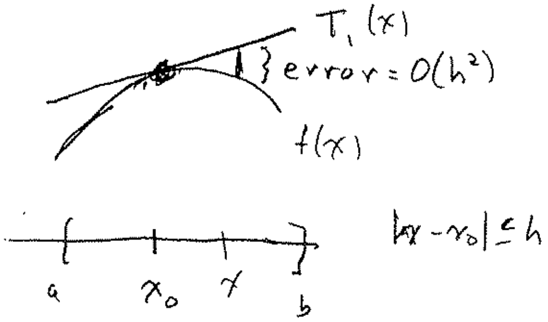
for some η, $a \le \eta \le b$.

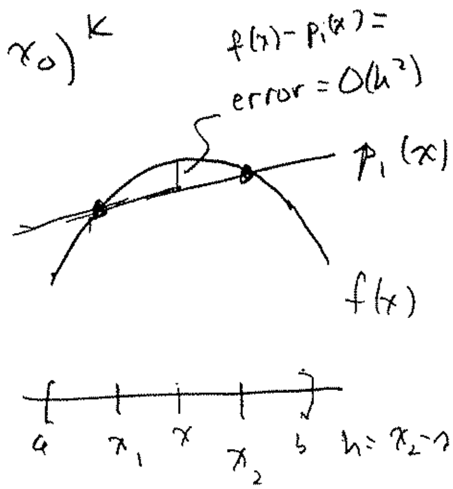
Remark: Learn to read and interpret theorems. In this case the error depends on the smoothness of f. Compare this with the error term for the n-1 st degree Taylor polynomial $T_{n-1}(x)$ near $x = x_0$:

$$f(x) = T_{n-1}(x) + \frac{f^{(n)}(\eta)}{n!}(x-x_0)^n, \quad a \le \eta \le b$$

where
$$T_{n-1}(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$$

picture $n = 2$





$I = [a, b]$

Pf: For $m=2$, see text for $m=4$. General case is the same

$p_{m-1}(x) = p_1(x) = 1^{st}$ degree (linear) polynomial,

If $x = x_1$ or x_2 Theorem is trivial.

$\therefore$ fix $x \neq x_1, x_2$ e.g. $x_1 < x < x_2$.

Let $\boxed{F(t) := f(t) - p_1(t) - c\,L(t)}$ (trick)

where $L(t) := (t-x_1)(t-x_2) = t^2 - (x_1+x_2)t + x_1x_2$
(quadratic in t)

(Note $L''(t) = 2!$).

and $c = c_x := \dfrac{f(x) - p_1(x)}{(x-x_1)(x-x_2)}$.

i.e. $F(t) = f(t) - p_1(t) - \dfrac{f(x)-p_1(x)}{(x-x_1)(x-x_2)}(t-x_1)(t-x_2)$

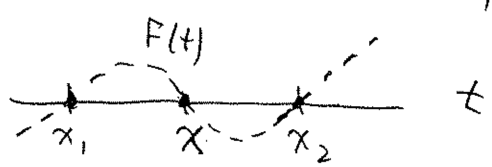
Next note that for $t = x, x_1, x_2$ we have

$$F(x) = f(x) - p_1(x) - \frac{f(x)-p_1(x)}{(x-x_1)(x-x_2)}(x-x_1)(x-x_2)$$
$$= f(x) - p_1(x) - f(x) + p_1(x) = 0$$

$$F(x_i) = f(x_i) - p_1(x_i) - \frac{f(x_i)-p_1(x_i)}{(x_i-x_1)(x_i-x_2)}(x_i-x_1)(x_i-x_2)$$
$$= f(x_i) - p_1(x_i) - f(x_i) + p_1(x_i) = 0, \quad i = 1,2$$

So $F(t)$ has 3 distinct 0's in $[a,b]$, x, x_1, x_2

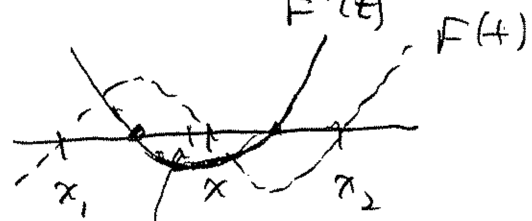
$F(x) = F(x_1) = F(x_2)$



Now note that $F'(t) = f'(t) - p_1'(t) - c L'(t)$ exists

and $F'(t) = 0$, for 2 distinct values between

x_1 and x and between x and x_2



Continuing in this way,

we see that

$$F''(t) = f''(t) - \underset{\overset{|||}{0}}{p_1''(t)} - c\, \underset{\overset{||}{2!}}{L''(t)}$$

$$= f''(t) - c \cdot 2!$$

and $F''(\eta_x) = 0$ for some η_x between the

0's of $F'(t)$, i.e. $F''(\eta_x) = f''(\eta_x) - c \cdot 2! = 0$

$$\Rightarrow \boxed{c = \frac{f''(\eta)}{2!}} \qquad \eta = \eta_x$$

$\therefore$ since for $t = x$ $\quad f(x) - p_1(x) - c L(x) = F(x) = 0$

we get $\quad f(x) = p_1(x) + c L(x)$

or

$$\boxed{f(x) = p_1(x) + \frac{f''(\eta)}{2!}(x - x_1)(x - x_2)}$$

(qed

<u>Remark</u>, what does this say about the error $f(x) - p_1(x)$ for $x_1 \le x \le x_2$?

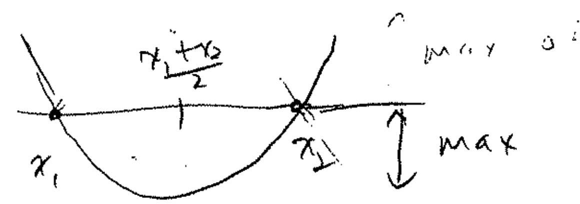
$$\text{error} = |f(x)-p_1(x)| = \left|\frac{f''(\eta_x)}{2!}(x-x_1)(x-x_2)\right| \le \frac{M_2}{2}\left|(x-x_1)(x-x_2)\right|$$

where $|f''(\eta)| \le M_2$ for all $\eta \in [a,b]$.

The error achieves its max at the critical pts of $(x-x_1)(x-x_2)$.

$$\frac{d}{dx}(x-x_1)(x-x_2) = \frac{d}{dx}\left(x^2-(x_1+x_2)x+x_1x_2\right) = 2x-(x_1+x_2) = 0$$

for $x = \frac{x_1+x_2}{2}$ = midpt.

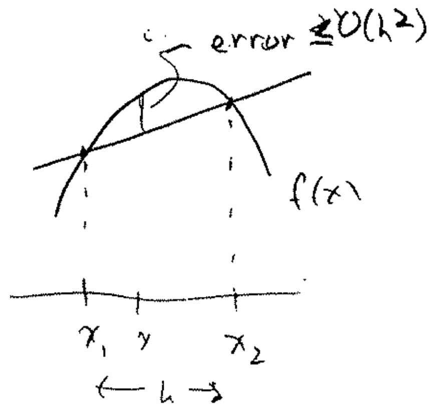


$$\therefore \text{max} = \left|\left(\frac{x_1+x_2}{2}-x_1\right)\left(\frac{x_1+x_2}{2}-x_2\right)\right| = \left|\left(\frac{x_2-x_1}{2}\right)\left(\frac{x_1-x_2}{2}\right)\right|$$

$$= \frac{(x_2-x_1)^2}{4} \le \frac{(b-a)^2}{4}$$

$$\therefore \text{the error} = |f(x)-p_1(x)| \le \frac{M_2}{2\cdot 4}(b-a)^2 = \frac{M_2}{8}h^2 = O(h^2)$$

if $a = x_1$, $b = x_2$, $h = b-a$



Error is as expected!

General case

$$\text{error} = |f(x) - p_{n-1}(x)| \le \frac{M_n}{n!} \, |(x-x_1)(x-x_2)\cdots(x-x_n)|$$

where $|f^{(n)}(\eta)| \le M_n \quad \forall\, \eta \in [a,b]$

Using equally spaced points $x_i = a + \frac{b-a}{n-1}(i-1)$, $i=1:n$

and for $x \in [a,b]$ using $x = a + \frac{b-a}{n-1} s$, $0 \le s \le n-1$

we have

$$\begin{aligned}(x-x_1)(x-x_2)\cdots(x-x_n) &= \prod_{i=1}^{n} (x - x_i) \\ &= \prod_{i=1}^{n} \left(\frac{b-a}{n-1}\right)(s-i+1) \\ &= \left(\frac{b-a}{n-1}\right)^n \prod_{i=1}^{n} (s-i+1) \\ &= \left(\frac{b-a}{n-1}\right)^n s(s-1)(s-2)\cdots(s-n+1)\end{aligned}$$

$$\therefore \text{error} = |f(x) - p_{n-1}(x)| \le M_n \left(\frac{b-a}{n-1}\right)^n \underbrace{\max_{0 \le s \le n-1} \left| \frac{s(s-1)\cdots(s-n+1)}{n!} \right|}_{\le \frac{1}{4n}}$$

C.B.S

$$\therefore |f(x) - p_{n-1}(x)| \le \frac{M_n}{4n} \left(\frac{b-a}{n-1}\right)^n = \frac{M_n}{4n} h^n = O(h^n)$$

where $h := \frac{b-a}{n-1}$

---

ex (Runge) $f(x) = \frac{1}{1+25x^2}$, $x \in [-1,1]$. Equally space x_i not good.

C.B.S $\frac{M_n}{4n} \to \infty$, $n \to \infty$ since $f^{(n)}(x)$ badly behaved.

(at $x = \pm i/5$, $f^{(n)}(x) = \infty$)

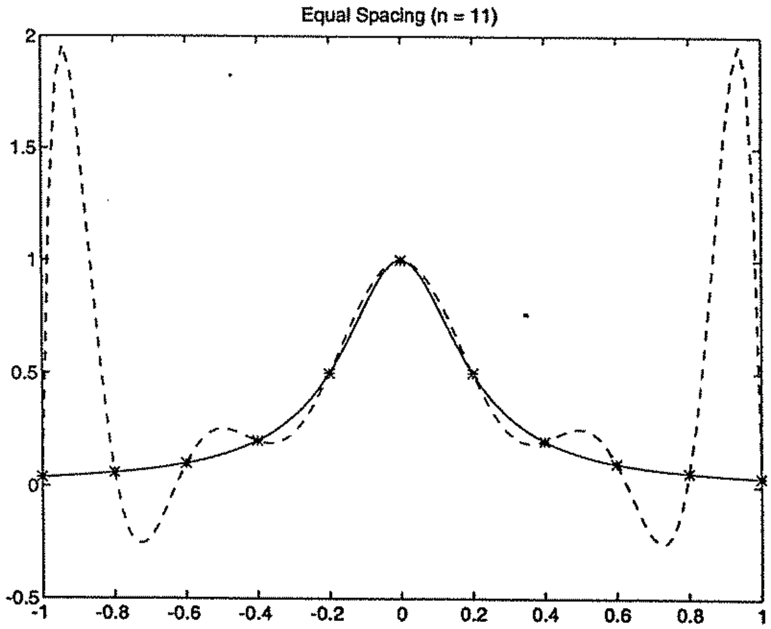


FIGURE 2.4 The Runge phenomenon.

```
% Script File: RungeEg
%
% For n=10:13, interpolants of f(x) = 1/(1+25x^2) on [-1,1]
% are plotted.
%
  close all
  x = linspace(-1,1,100)';
  y = ones(100,1)./(1 + 25*x.^2);
  for n=10:13
     figure
     xEqual = linspace(-1,1,n)';
     yEqual = ones(size(xEqual))./(1+25*xEqual.^2);;
     cEqual=InterpN(xEqual,yEqual);
     pvalsEqual = HornerN(cEqual,xEqual,x);
     plot(x,y,x,pvalsEqual,xEqual,yEqual,'*')
     title(sprintf('Equal Spacing (n = %2.0f)',n))
  end
```

See Fig 2.4 for sample output.) While the interpolant "captures" the trend of the function in he middle part of the interval, it blows up near the endpoints.

roblems

2.3.1 Write a MATLAB script that compares HornerN and HornerV from the flop point of view.

2.4.1 Divided Differences

(some details filled in + rewritten)

ex: $p_2(x) = c_1 + c_2(x-x_1) + c_3(x-x_1)(x-x_2)$

interpolates (x_i, y_i) $i = 1,2,3$ $\quad y_i = f(x_i)$

given function

$$f(x_1) = y_1 = p_2(x_1) = c_1$$

$$f(x_2) = y_2 = p_2(x_2) = c_1 + c_2(x_2 - x_1)$$

$$f(x_3) = y_3 = p_2(x_3) = c_1 + c_2(x_3-x_1) + c_3(x_3-x_1)(x_3-x_2)$$

$\Downarrow$

divided difference | order

$$c_1 = f(x_1) \quad =: f[x_1] \qquad 0$$

$$c_2 = \frac{y_2 - c_1}{x_2 - x_1} = \frac{f[x_2] - f[x_1]}{x_2 - x_1} \quad =: f[x_1, x_2] \qquad 1$$

(more generally $f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$, $i \neq j$)

$$c_3 = \frac{y_3 - (c_1 + c_2(x_3 - x_1))}{(x_3-x_1)(x_3-x_2)}$$

$$= \frac{f[x_3] - f[x_1] - \frac{f(x_2)-f(x_1)}{x_2 - x_1}(x_3 - x_1)}{(x_3-x_1)(x_3-x_2)}$$

$$= \frac{\frac{f(x_3)-f(x_1)}{x_3-x_1} - \frac{f(x_2)-f(x_1)}{x_2-x_1}}{x_3 - x_2}$$

(Not quite what we want)

$$= \frac{f[x_1, x_3] - f[x_1, x_2]}{x_3 - x_1} \quad =: f[x_1, x_2, x_3] \qquad 2$$

In general, if $p_{n-1}(x)$ is $n-1$ st degree poly. interpolating (x_i, y_i) $i = 1, \ldots, n$, $f(x_i) = y_i$

$$p_{n-1}(x) = c_1 + c_2(x-x_1) + c_3(x-x_1)(x-x_2) + \ldots + c_n(x-x_1)\cdots(x-x_{n-1})$$

$$= f[x_1] + f[x_1,x_2](x-x_1) + f[x_1,x_2,x_3](x-x_1)(x-x_2) + \ldots + f[x_1,x_2,\ldots,x_n](x-x_1)\cdots(x-x_{n-1})$$

$$= \sum_{k=1}^{n} f[x_1,\ldots,x_k] \prod_{j=1}^{k-1} (x-x_j)$$

i.e. $c_k = f[x_1, \ldots, x_k]$ = $k-1$ st order divided difference

Can we write $k-1$ st order divided difference <u>in terms of</u> $k-2$ st order div. diff.?

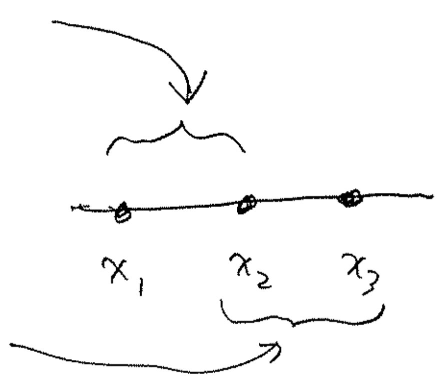
<u>Idea</u> $p_L(x) = f[x_1] + f[x_1,x_2](x-x_1)$

interpolates $(x_1, f(x_1))$, $(x_2, f(x_2))$

and

$p_R(x) = f[x_2] + f[x_2,x_3](x-x_2)$

interpolates $(x_2, f(x_2))$, $(x_3, f(x_3))$



$\therefore$ $$p(x) = \frac{x - x_3}{x_1 - x_3}\, p_L(x) + \frac{x_1 - x}{x_1 - x_3}\, p_R(x)$$

is the 2^{nd} degree poly. interpolant

of $(x_i, f(x_i))$ $i = 1, 2, 3$

Check: $$p(x_1) = \frac{x_1 - x_3}{x_1 - x_3}\, p_L(x_1) = f(x_1)$$

$$p(x_2) = \frac{x_2 - x_3}{x_1 - x_3}\, p_L(x_2) + \frac{x_1 - x_2}{x_1 - x_3}\, p_R(x_2)$$

$$= \frac{x_2 - x_3}{x_1 - x_3}\, f(x_2) + \frac{x_1 - x_2}{x_1 - x_3}\, f(x_2)$$

$$= \frac{x_1 - x_3}{x_1 - x_3}\, f(x_2) = f(x_2)$$

$$p(x_3) = \frac{x_1 - x_3}{x_1 - x_3}\, p_R(x_3) = f(x_3)$$

Note also that $\qquad$ terms of 1^{st} order

$$p(x) = \ldots + f[x_1, x_2, x_3](x - x_1)(x - x_2)$$

$$= \frac{x - x_3}{x_1 - x_3}\, p_L(x) + \frac{x_1 - x}{x_1 - x_3}\, p_R(x)$$

$$= \ldots + \frac{x - x_3}{x_1 - x_3}\, f[x_1, x_2](x - x_1) + \ldots + \frac{x_1 - x}{x_1 - x_3}\, f[x_2, x_3](x - x_2$$

$$= \ldots + \frac{x - x_2 + x_2 - x_3}{x_1 - x_3}\, f[x_1, x_2](x - x_1) + \frac{f[x_2, x_3]}{x_3 - x_1}(x - x_1)(x$$

$$= \ldots 1^{st} \text{ order terms} + \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}(x - x_1)(x - x_2)$$

$\therefore$ since coefficient of x^2 terms must be the same

$$\boxed{f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}}$$

In general we may show (p. 93) that the divided differences satisfy the recursion:

$$\boxed{f[x_1, \ldots, x_k] = \frac{f[x_2, \ldots, x_k] - f[x_1, \ldots, x_{k-1}]}{x_k - x_1}}$$

<u>picture</u>:

$$\begin{tikzcd}
f[x_1] \arrow[dr, dash] & & & \\
f[x_2] \arrow[r, dash] \arrow[dr, dash] & f[x_1, x_2] \arrow[dr, dash] & & \\
f[x_3] \arrow[r, dash] \arrow[dr, dash] & f[x_2, x_3] \arrow[r, dash] \arrow[dr, dash] & f[x_1, x_2, x_3] \arrow[dr, dash] & \\
f[x_4] \arrow[r, dash] & f[x_3, x_4] \arrow[r, dash] & f[x_2, x_3, x_4] \arrow[r, dash] & f[x_1, x_2, x_3, x_4] \\
\vdots & \vdots & \vdots & \ddots
\end{tikzcd}$$

<u>See InterpN2</u>