Polynomial Interpolation with $n+1$ nodes

- Given $n+1$ distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$

- Interpolating polynomial of degree $\leq n$

$$\begin{aligned} P(x) &= f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x), \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \quad \cdots, \quad P(x_n) = f(x_n), \end{aligned}$$

  and Lagrangian polynomials

$$L_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

- For all i, j,

$$L_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

Polynomial Interpolation with $n + 1$ nodes

- **Analysis**: Estimate errors with polynomial interpolation

- **Computation**: How to compute $P(x)$ numerically.

Polynomial Interpolation: Analysis

Theorem: Assume that the nodal points $x_0, \cdots, x_n$ are mutually distinct, then the interpolating polynomial $P(x)$ of degree $\leq n$ exists and is unique.

Polynomial Interpolation: Analysis

Lemma: Let $f(x) \in C^{n+1}[a, b]$ be such that

$$f(x_1) = 0, \quad f(x_2) = 0, \quad \cdots, \quad f(x_n) = 0,$$

where $x_1, x_2, \cdots, x_n \in [a, b]$ are mutually distinct. Then there exists a $\xi \in [\mathbf{min}(x_1, x_2, \cdots, x_n), \quad \mathbf{max}(x_1, x_2, \cdots, x_n)]$ such that

$$f^{(n-1)}(\xi) = 0.$$

Polynomial Interpolation Error

Theorem: Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{n+1}[a, b]$. Then, for each $x \in [a, b]$, a number $\xi(x)$ between $, x_0, x_1, \cdots, x_n$ (hence $\in (a, b)$) exists with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1) \cdots (x - x_n),$$

where $P(x)$ is the interpolating polynomial.

Polynomial Interpolation Error: Proof

If $x = x_0, x_1, \cdots, \ x_n$, then error $= 0$ and theorem is true. Now let x be not equal to any node. Define function g for $t \in [a, b]$

$$g(t) \stackrel{def}{=} (f(t) - P(t)) - (f(x) - P(x)) \frac{(t - x_0)(t - x_1) \cdots (t - x_n)}{(x - x_0)(x - x_1) \cdots (x - x_n)}$$

$$= (f(t) - P(t)) - (f(x) - P(x)) \prod_{j=0}^{n} \frac{(t - x_j)}{(x - x_j)} \in C^{n+1}[a, b].$$

Then $g(t)$ vanishes at $n + 2$ distinct points:

$$g(x) = 0, \quad g(x_k) = 0, \quad , \text{for} \quad k = 0, 1, \cdots, n.$$

There must be a ξ between x and nodal points such that

$$g^{(n+1)}(\xi) = 0.$$

Polynomial Interpolation Error: Proof

Since

$$
\begin{aligned}
g^{(n+1)}(\xi) &= (f(t) - P(t))^{(n+1)}\,|_{t=\xi} - (f(x) - P(x))\left(\prod_{j=0}^{n} \frac{(t - x_j)}{(x - x_j)}\right)^{(n+1)}|_{t=\xi} \\
&= f^{(n+1)}(\xi) - (f(x) - P(x))\,\frac{(n+1)!}{\prod_{j=0}^{n}(x - x_j)} \\
&= 0
\end{aligned}
$$

Therefore

$$
f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n),
$$

Polynomial Interpolation: Corollary

Corollary: Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in the interval $[a, b]$ and f is polynomial of degree at most n, then

$$P(x) = f(x).$$

Secant Method: Order of Convergence

Assume that the secant method

$$p_{k+1} = p_k - \frac{f(p_k)(p_k - p_{k-1})}{f(p_k) - f(p_{k-1})}, \quad k = 1, 2, \cdots,$$

converges to the root p ($f(p) = 0$.) It follows that

$$\begin{aligned} p_{k+1} - p &= p_k - p - \frac{(f(p_k) - f(p))(p_k - p_{k-1})}{f(p_k) - f(p_{k-1})} \\ &= \frac{(p_k - p)(f(p_k) - f(p_{k-1})) - (f(p_k) - f(p))(p_k - p_{k-1})}{f(p_k) - f(p_{k-1})} \\ &= \frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})} \left(f(p) - f(p_k) - \frac{(p_k - p)(f(p_k) - f(p_{k-1}))}{(p_k - p_{k-1})} \right) \end{aligned}$$

Secant Method: Order of Convergence

But the last expression

$$P(p) \stackrel{def}{=} f(p_k) + \frac{(p_k - p)(f(p_k) - f(p_{k-1}))}{(p_k - p_{k-1})}$$

is linear interpolation on nodes p_k and p_{k-1}, so there exists ξ_k between p, p_k and p_{k-1} with

$$f(p) - P(p) = \frac{f^{(2)}(\xi_k)}{2}(p - p_k)(p - p_{k-1}),$$

and therefore

$$p_{k+1} - p = \frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})} \cdot \frac{f^{(2)}(\xi_k)}{2} \cdot (p - p_k)(p - p_{k-1}).$$

Secant Method: Order of Convergence $= \alpha$

Let

$$\alpha = \frac{\sqrt{5}+1}{2} > 1$$

be the *golden ratio*. Then $\alpha(\alpha-1) = 1$, and

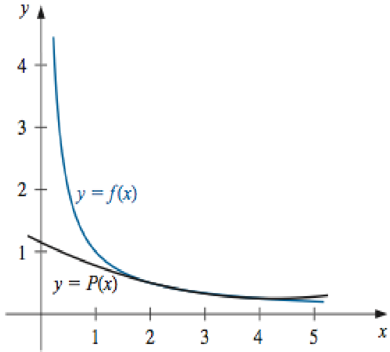
$$\frac{|p_{k+1}-p|}{|p_k-p|^\alpha}\left(\frac{|p_k-p|}{|p_{k-1}-p|^\alpha}\right)^{\alpha-1} = \left|\frac{p_k-p_{k-1}}{f(p_k)-f(p_{k-1})}\cdot\frac{f^{(2)}(\xi_k)}{2}\right|$$

$$\longrightarrow \left|\frac{f^{(2)}(p)}{2f'(p)}\right|, \quad \text{provided} \quad f'(p) \neq 0,$$

or $$\lim_{k\to\infty}\frac{|p_{k+1}-p|}{|p_k-p|^\alpha} = \left|\frac{f^{(2)}(p)}{2f'(p)}\right|^{\frac{1}{\alpha}}. \quad \text{(the limit does exist.)}$$

Polynomial Interpolation, Error Bounds:

Estimate maximum error in second order polynomial interpolation of function $f(x) = \frac{1}{x}$ over interval $[2, 4]$ using nodes $x_0 = 2, x_1 = 2.75$, and $x_2 = 4$.



Polynomial Interpolation, Error Bounds:

$$f(x) = P(x) + \frac{f^{(3)}(\xi(x))}{3!}(x-2)(x-2.75)(x-4),$$

where $P(x)$ is the interpolating polynomial on $[2,4]$.

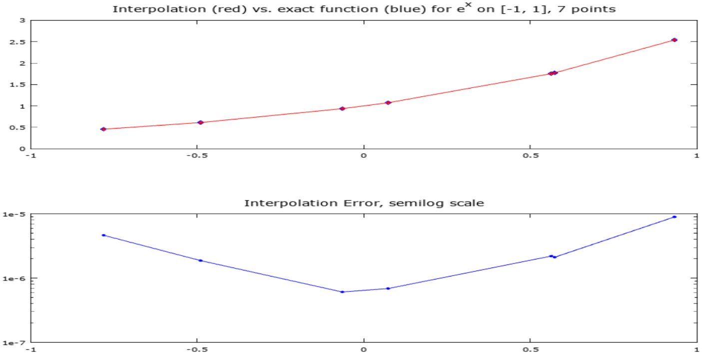
$$\begin{aligned}
f^{(3)}(x) &= -6x^{-4}, \quad |f^{(3)}(\xi(x))| \leq 6 \cdot 2^{-4}, \\
g(x) &\stackrel{def}{=} (x-2)(x-2.75)(x-4) = x^3 - 35x^2 + \frac{49}{2}x - 22. \\
\frac{d\,g(x)}{d\,x} &= 3x^2 - \frac{35}{2}x + \frac{49}{2} = \frac{1}{2}(3x-7)(2x-7). \\
|g(\frac{7}{2})| &= \frac{9}{16} = \mathbf{max}_{x \in [2,4]}|g(x)|. \quad (|g(\frac{7}{3})| = \frac{25}{108} < \frac{9}{16}) \\
|f(x) - P(x)| &\leq \frac{1}{3!} \cdot 6 \cdot 2^{-4} \cdot \frac{9}{16} = \frac{9}{256}.
\end{aligned}$$

Polynomial Interpolation: Experiments:

- **Easy function**: $f(x) = e^x$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| \leq e \quad \text{for all} \quad \xi \in (-1, 1).$$

- 7 nodal points ($n = 6$); random x points.

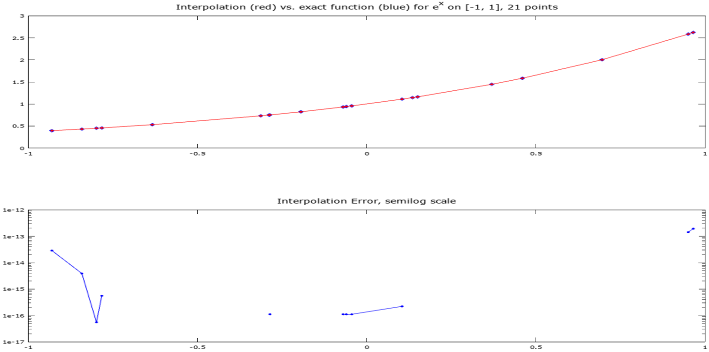


Polynomial Interpolation: Experiments:

- **Easy function**: $f(x) = e^x$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| \leq e \quad \text{for all} \quad \xi \in (-1, 1).$$

- 21 nodal points ($n = 20$); random x points.

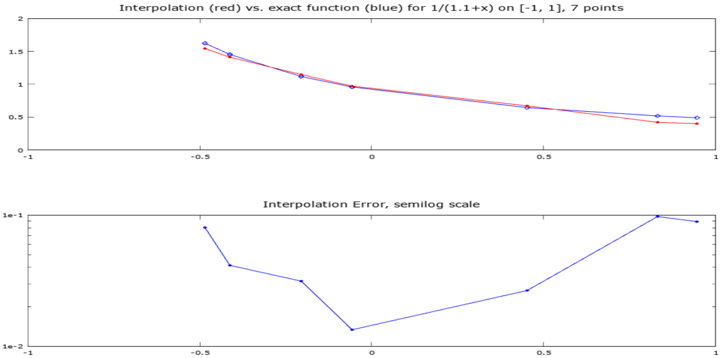


Polynomial Interpolation: Experiments:

- **Hard function**: $f(x) = \frac{1}{1.1+x}$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| = \frac{n!}{(1.1+\xi)^{n+1}} \leq 10^{n+1} \cdot n! \quad \text{for all} \quad \xi \in (-1, 1).$$

- 7 nodal points ($n = 6$); random x points.

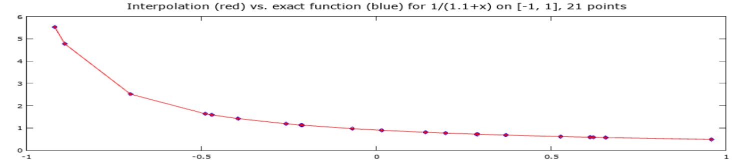


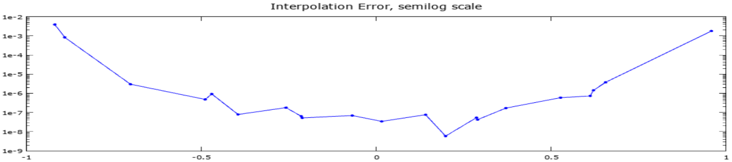
Polynomial Interpolation: Experiments:

- **Hard function**: $f(x) = \frac{1}{1.1+x}$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| = \frac{n!}{(1.1+\xi)^{n+1}} \leq 10^{n+1} \cdot n! \quad \text{for all} \quad \xi \in (-1, 1).$$

- 21 nodal points ($n = 20$); random x points.



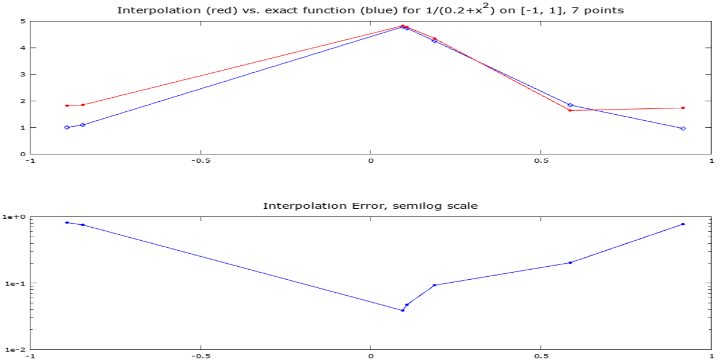


Polynomial Interpolation: Experiments:

- **Hardest function**: $f(x) = \frac{1}{0.2+x^2}$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| \quad \text{can be very large for some} \quad \xi \in (-1, 1).$$

- 7 nodal points ($n = 6$); random x points.

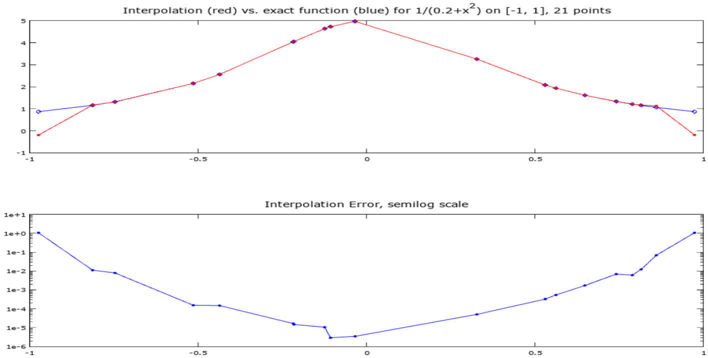


Polynomial Interpolation: Experiments:

- **Hardest function**: $f(x) = \frac{1}{0.2+x^2}$ on $[-1,1]$.

$$|f^{(n)}(\xi)| \quad \text{can be very large for some} \quad \xi \in (-1,1).$$

- 21 nodal points ($n = 20$); random x points.

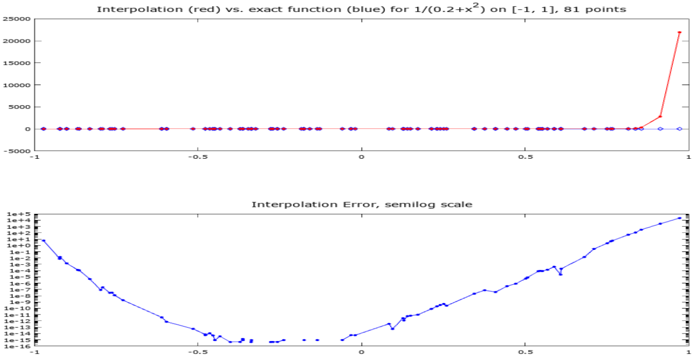


Polynomial Interpolation: Experiments:

- **Hardest function**: $f(x) = \frac{1}{0.2+x^2}$ on $[-1,1]$.

  $$|f^{(n)}(\xi)| \quad \text{can be very large for some} \quad \xi \in (-1,1).$$

- 81 nodal points ($n = 80$); random x points.



Polynomial Interpolation: Analysis and Computation

- **Analysis**: Estimate errors with polynomial interpolation
- **Computation**: How to compute $P(x)$ numerically

Polynomial Interpolation: Neville's Method

- **Let** $Q(x)$ interpolate $f(x)$ at $x_0, x_1, \cdots, x_k$,
- **Let** $\widehat{Q}(x)$ interpolate $f(x)$ at $x_1, \cdots, x_k, x_{k+1}$.
- **Then**

$$P(x) \stackrel{def}{=} \frac{(x - x_{k+1})Q(x) - (x - x_0)\widehat{Q}(x)}{x_0 - x_{k+1}}$$

interpolates $f(x)$ at $x_0, x_1, \cdots, x_k, x_{k+1}$

Build higher degree interpolating polynomial from lower degreed ones.

Neville's Method: Proof

- **Let** $Q(x)$ interpolate $f(x)$ at $x_0, x_1, \cdots, x_k$,
- **Let** $\widehat{Q}(x)$ interpolate $f(x)$ at $x_1, \cdots, x_k, x_{k+1}$.

$$P(x) \stackrel{def}{=} \frac{(x - x_{k+1})Q(x) - (x - x_0)\widehat{Q}(x)}{x_0 - x_{k+1}}$$

- **for** $1 \leq j \leq k$,

$$\begin{aligned} P(x_j) &= \frac{(x_j - x_{k+1})Q(x_j) - (x_j - x_0)\widehat{Q}(x_j)}{x_0 - x_{k+1}} \\ &= \frac{(x_j - x_{k+1})f(x_j) - (x_j - x_0)f(x_j)}{x_0 - x_{k+1}} = f(x_j). \end{aligned}$$

Neville's Method: Proof

- **Let** $Q(x)$ interpolate $f(x)$ at $x_0, x_1, \cdots, x_k$,
- **Let** $\widehat{Q}(x)$ interpolate $f(x)$ at $x_1, \cdots, x_k, x_{k+1}$.

$$P(x) \stackrel{def}{=} \frac{(x - x_{k+1})Q(x) - (x - x_0)\widehat{Q}(x)}{x_0 - x_{k+1}}$$

- **for** $j = 0, k+1$,

$$P(x_0) = \frac{(x_0 - x_{k+1})Q(x_0) - (x_0 - x_0)\widehat{Q}(x_0)}{x_0 - x_{k+1}} = f(x_0),$$

$$P(x_{k+1}) = \frac{(x_{k+1} - x_{k+1})Q(x_{k+1}) - (x_{k+1} - x_0)\widehat{Q}(x_{k+1})}{x_0 - x_{k+1}} = f(x_{k+1}).$$

- **Thus** $P(x)$ interpolates $f(x)$ at $x_0, x_1, \cdots, x_k, x_{k+1}$.

Neville's Method, general case

- **Let** $Q(x)$ interpolate $f(x)$ at all of $x_0, x_1, \cdots, x_k, x_{k+1}$ but x_j,
- **Let** $\widehat{Q}(x)$ interpolate $f(x)$ at all of $x_0, x_1, \cdots, x_k, x_{k+1}$ but x_i, $i \neq j$.
- **Then**

$$P(x) \stackrel{def}{=} \frac{(x - x_j)Q(x) - (x - x_i)\widehat{Q}(x)}{x_i - x_j}$$

interpolates $f(x)$ at all of $x_0, x_1, \cdots, x_k, x_{k+1}$

Neville's Method, recursive

- **Step 1**: compute $Q_{i,i+1}(x)$, linear polynomial interpolating $f(x)$ at x_i, x_{i+1}, $i = 0, 1, \cdots, n-1$.
- **Step 2**: compute $Q_{i,i+1,\cdots,j+1}(x)$, polynomial of degree $j - i + 1$, interpolating $f(x)$ at $x_i, i + 1, \cdots, x_{j+1}$, from $Q_{i,i+1,\cdots,j}(x)$ and $Q_{i+1,i+2,\cdots,j+1}(x)$,
  for $i = 0, 1, \cdots, n-1$, and $j = i + 1, \cdots n$.

Neville's Method, pseudo-code

```
function [y,out] = nev(xx,x,Q)
% Neville's algorithm as a function (save as "nev.m")
%
% inputs:
%    n = order of interpolation (n+1 = # of points)
%    x(1),...,x(n+1)    x coords
%    Q(1),...,Q(n+1)    function values at x
%    xx=evaluation point for interpolating polynomial p
%
%
% On output
%    y      = p(xx):
%    out.Q = intermediate Q values.
%    out.Q(j): approximation by interpolating at points x(j) ... x(n+1)
%
% Written by Ming Gu for Math 128A, Spring 2015
%

N = length(x);
n = N - 1;

for i = n:-1:1
   for j = 1:i
      Q(j) = (xx-x(j))*Q(j+1) - (xx-x(j+N-i))*Q(j);
      Q(j) = Q(j)/(x(j+N-i)-x(j));
   end
end

y = Q(1);
out.Q = Q;
```

Divided Differences

- Given $n+1$ distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$

- Interpolating polynomial of degree $\leq n$

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \\ &\quad \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \quad \cdots, \quad P(x_n) = f(x_n). \end{aligned}$$

It follows

$$\begin{aligned} a_0 &= P(x_0) = f(x_0), \\ \frac{P(x) - f(x_0)}{x - x_0} &= a_1 + a_2(x - x_1) + \\ &\quad \cdots + a_n(x - x_1)\cdots(x - x_{n-1}). \end{aligned}$$

Divided Differences

$$\text{Define} \quad f[x_i] = f(x_i), \quad f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, \quad \text{for all } i$$

Let $x = x_1$ in

$$\frac{P(x) - f(x_0)}{x - x_0} = a_1 + a_2(x - x_1) + \cdots + a_n(x - x_1) \cdots (x - x_{n-1}).$$

It follows that

$$\begin{aligned} a_1 &= \frac{P(x_1) - f(x_0)}{x_1 - x_0} = f[x_0, x_1] \\ \frac{P(x) - P(x_0)}{x - x_0} &= f[x_0, x_1] + a_2(x - x_1) + \cdots \\ &\quad + a_n(x - x_1) \cdots (x - x_{n-1}) \\ \frac{P[x_0, x] - P[x_0, x_1]}{x - x_1} &= a_2 + \cdots + a_n(x - x_2) \cdots (x - x_{n-1}). \end{aligned}$$

Divided Differences

recursive, $f[x_i, x_{i+1}, \cdots, x_{j+1}] = \dfrac{f[x_{i+1}, \cdots, x_{j+1}] - f[x_i, x_{i+1}, \cdots, x_j]}{x_{j+1} - x_i}$,

then in

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \\ &\quad \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned}$$

we have

$$\begin{aligned} a_0 &= f[x_0] \\ a_1 &= f[x_0, x_1] \\ a_2 &= f[x_0, x_1, x_2] \\ &\vdots \\ a_{n-1} &= f[x_0, x_1, \cdots, x_n] \end{aligned}$$

Newton Divided Difference Example

Table 3.10

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

Newton Divided Difference Example

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{0.6200860 - 0.7651977}{1.3 - 1.0} = -0.4837057.$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-0.5489460 - (-0.4837057)}{1.6 - 1.0} = -0.1087339.$$

Coefficients are numbers on top of all $f[\cdots]$ columns

i	x_i	$f[x_i]$	$f[x_{i-1},x_i]$	$f[x_{i-2},x_{i-1},x_i]$	$f[x_{i-3},\ldots,x_i]$	$f[x_{i-4},\ldots,x_i]$
0	1.0	0.7651977				
			−0.4837057			
1	1.3	0.6200860		−0.1087339		
			−0.5489460		0.0658784	
2	1.6	0.4554022		−0.0494433		0.0018251
			−0.5786120		0.0680685	
3	1.9	0.2818186		0.0118183		
			−0.5715210			
4	2.2	0.1103623				

Newton Divided Difference

Theorem: Suppose that $f \in C^n[a, b]$ and $x_0, x_1, \cdots, x_n$ are distinct nodes in $[a, b]$. Then a number $\xi \in (a, b)$ exists such that

$$f[x_0, x_1, \cdots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Proof: Let $g(x) = f(x) - P(x)$, where

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \\ &\quad \cdots + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}). \end{aligned}$$

Since $g(x_j) = f(x_j) - P(x_j) = 0, \quad j = 0, 1, \cdots, n$, it follows that a number $\xi \in (a, b)$ exists such that $g^{(n)}(\xi) = 0$. But

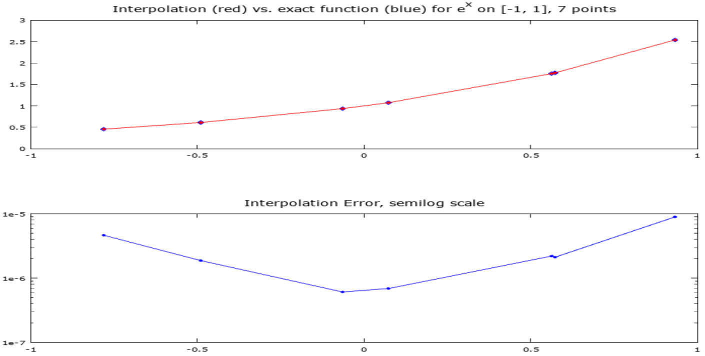
$$\begin{aligned} g^{(n)}(\xi) &= f^{(n)}(\xi) - P^{(n)}(\xi) = f^{(n)}(\xi) - a_n\, n! \\ &= f^{(n)}(\xi) - f[x_0, x_1, \cdots, x_n]\, n! = 0. \end{aligned}$$

Polynomial Interpolation: Experiments:

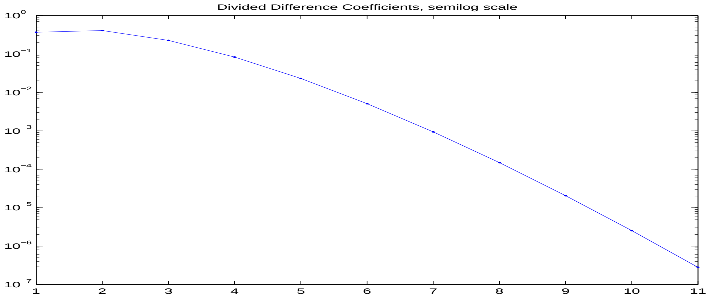
- **Easy function**: $f(x) = e^x$ on $[-1, 1]$.

$$|f^{(n)}(\xi)| \leq e \quad \text{for all} \quad \xi \in (-1, 1).$$

- 7 nodal points ($n = 6$); random x points.



Divided difference coefficients for e^x on $[-1, 1]$

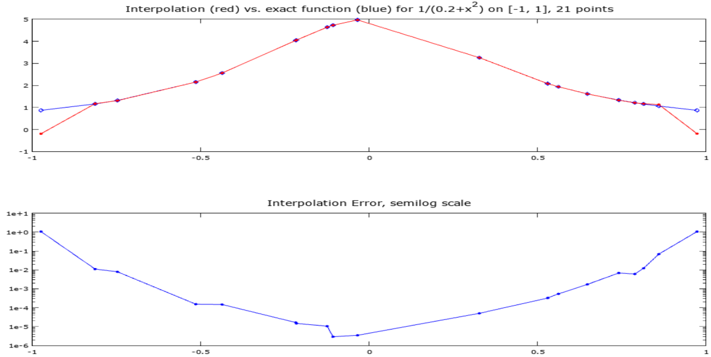


Polynomial Interpolation: Experiments:

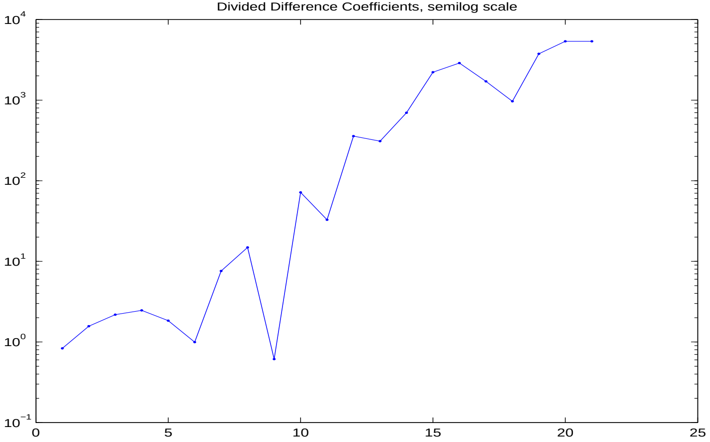
- **Hardest function**: $f(x) = \frac{1}{0.2+x^2}$ on $[-1, 1]$.

  $$|f^{(n)}(\xi)| \quad \text{can be very large for some} \quad \xi \in (-1, 1).$$

- 21 nodal points ($n = 20$); random x points.



Divided difference coefficients for $1/(0.2 + x^2)$ on $[-1, 1]$



Divided difference, book version

```
function F = NewtonDividedDifference(x,f)
%
% This function implements Newton's Divided Difference Algorithm
% F is the vector of coefficients
%
% Written by Ming Gu for Math 128A, Fall 2008
%
n = length(x);
P = diag(f);
for k=2:n
   for j = k-1:-1:1
       P(k,j) = (P(k,j+1)-P(k-1,j))/(x(k)-x(j));
   end
end
F = P(:,1);
```

Divided difference, $O(n)$ memory

```
function F = NDD1(x,f)
%
% This function implements Newton's Divided Difference Algorithm
% F is the vector of coefficients
%
% Updated by Ming Gu for Math 128A, Spring 2015
%
N = length(x);
F = f;
for k=2:N
    for j = N:-1:k
       F(j) = (F(j)-F(j-1))/(x(j)-x(j-k+1));
    end
end
```

Interpolation via NDD

```
function f = EvaluateNDD(xnew,x,F)
%
% This function evaluates the interpolating polynomial given
% point xnew, nodes x, and NDF coefficients F
%
n = length(x);
m = length(xnew);
f = F(n)*ones(m,1);
for k=n-1:-1:1
    f = F(k) + f .* (xnew-x(k));
end
```

Forward Differences

Let $x_1 = x_0 + h$, $x_2 = x_0 + 2h$

$$
\begin{aligned}
f[x_0, x_1] &= f[x_0, x_0 + h] = \frac{f(x_0 + h) - f(x_0)}{h} \quad \text{(first order FD)}, \\
f[x_0, x_1, x_2] &= f[x_0, x_0 + h, x_0 + 2h] = \frac{f[x_1, x_2] - f[x_0, x_1]}{2h} \\
&= \frac{f(x_2) + f(x_0) - 2f(x_1)}{2h^2} \quad \text{(second order FD)}
\end{aligned}
$$

Forward Differences

Let $x_1 = x_0 + h$, $x_2 = x_0 + 2h$

$$\begin{aligned} f[x_0, x_1] &= f[x_0, x_0 + h] = \frac{f(x_0 + h) - f(x_0)}{h} \quad \text{(first order FD)}, \\ f[x_0, x_1, x_2] &= f[x_0, x_0 + h, x_0 + 2h] = \frac{f[x_1, x_2] - f[x_0, x_1]}{2h} \\ &= \frac{f(x_2) + f(x_0) - 2f(x_1)}{2h^2} \quad \text{(second order FD)} \end{aligned}$$

```
>> x=0;h=0.1;x1=x+h;x2=x+2*h;
>> f1=(exp(x1)-exp(x))/h;
>> f2 = (exp(x2)+exp(x)-2*exp(x1))/(2*h^2);
>> f=exp(x);
>> [f f1 f2]
ans =

   1.00000   1.05171   0.55305
```

Backward Differences

Let $x_{n-1} = x_n - h$, $x_{n-2} = x_n - 2h$

$$
\begin{aligned}
f[x_{n-1}, x_n] &= f[x_n - h, x_n] = \frac{f(x_n) - f(x_n - h)}{h} \quad \text{(1st order BD)}, \\
f[x_{n-2}, x_{n-1}, x_n] &= f[x_n - 2h, x_n - h, x_n] \\
&= \frac{f[x_n - h, x_n] - f[x_n - 2h, x_n - h]}{2h} \\
&= \frac{f(x_n) + f(x_{n-2}) - 2f(x_{n-1})}{2h^2} \quad \text{(2nd order BD)}
\end{aligned}
$$

Backward Differences

Let $x_{n-1} = x_n - h$, $x_{n-2} = x_n - 2h$

$$f[x_{n-1}, x_n] = f[x_n - h, x_n] = \frac{f(x_n) - f(x_n - h)}{h} \quad \text{(1st order BD)},$$

$$\begin{aligned} f[x_{n-2}, x_{n-1}, x_n] &= f[x_n - 2h, x_n - h, x_n] \\ &= \frac{f[x_n - h, x_n] - f[x_n - 2h, x_n - h]}{2h} \\ &= \frac{f(x_n) + f(x_{n-2}) - 2f(x_{n-1})}{2h^2} \quad \text{(2nd order BD)} \end{aligned}$$

```
>> x=0;h=0.1;x1=x-h;x2=x-2*h;
>> f1=(exp(x)-exp(x1))/h;
>> f2 = (exp(x2)+exp(x)-2*exp(x1))/(2*h^2);
>> f=exp(x);
>> [f f1 f2]
ans =

   1.00000   0.95163   0.45280
```

Double nodes (I)

- Given 2 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1))$$

- Interpolating polynomial of degree ≤ 1

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \end{aligned}$$

Where

$$a_0 = f(x_0), \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Double nodes (I)

- Given 2 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1))$$

- Interpolating polynomial of degree ≤ 1

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) \\ \text{with} \quad P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \end{aligned}$$

Where

$$a_0 = f(x_0), \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Now let $x_1 \to x_0$, we obtain (Taylor expansion)

$$P(x) = a_0 + a_1(x - x_0),$$

where $a_0 = f(x_0)$, $a_1 = f'(x_0) \stackrel{def}{=} f[x_0, x_0]$.

$$P(x_0) = f(x_0), \quad P'(x_0) = f'(x_0).$$

Double nodes (II)

- Given 3 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)),$$

- Interpolating polynomial of degree ≤ 2

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \quad P(x_2) = f(x_2), \end{aligned}$$

where we have $a_0 = f[x_0], a_1 = f[x_0, x_1]$,

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Double nodes (II)

- Given 3 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)),$$

- Interpolating polynomial of degree ≤ 2

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \quad P(x_2) = f(x_2), \end{aligned}$$

where we have $a_0 = f[x_0], a_1 = f[x_0, x_1]$,

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Now let $x_2 = x_1$, we obtain (mixed interpolation)

$$\begin{aligned} f[x_1, x_2] &= f'(x_1) = f[x_1, x_1], \\ a_2 &= \frac{f[x_1, x_1] - f[x_0, x_1]}{x_1 - x_0} \stackrel{def}{=} f[x_0, x_1, x_1] \\ P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ P(x_0) &= f(x_0), \quad P(x_1) = f(x_1) \quad P'(x_1) = f'(x_1). \end{aligned}$$

Double nodes (III)

- Given 4 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)),$$

- Interpolating polynomial of degree ≤ 3

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2).$$

where $a_0 = f[x_0], a_1 = f[x_0, x_1]$, It follows

$$\begin{aligned} a_2 &= f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ a_3 &= f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}. \end{aligned}$$

Double nodes (IV)

Let $x_1 = x_0$, and $x_3 = x_2$. It follows that

$$\begin{aligned} a_1 &= f[x_0, x_1] = f[x_0, x_0] \\ a_2 &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{f[x_0, x_2] - f[x_0, x_0]}{x_2 - x_0} \\ a_3 &= f[x_0, x_1, x_2, x_3] = \frac{f[x_0, x_2, x_2] - f[x_0, x_0, x_2]}{x_2 - x_0}. \end{aligned}$$

Hermite Interpolation:

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^2(x - x_2) \\ P(x_0) &= f(x_0), \;\; P'(x_0) = f'(x_0), \;\; P(x_2) = f(x_2), \;\; P'(x_2) = f'(x_2). \end{aligned}$$