Note Title 2/23/2015

$$\begin{array}{cccc}
y[x_0] = y_0 & & & \\
 & y[x_0, x_1] & & \\
y[x_1] = y_1 & & y[x_0, x_1, x_2] & \\
 & y[x_1, x_2] & & y[x_0, x_1, x_2, x_3] \\
y[x_2] = y_2 & & y[x_1, x_2, x_3] & \\
 & y[x_2, x_3] & & \\
y[x_3] = y_3 & & &
\end{array}$$

Algorithm (to compute $y[x_0, \dots, x_i]$)

Input: x_i, y_i

Ouput: $y[x_0, \dots, x_i]$ in y_i

for $j = 1:n$

for $i = n : j$

$$y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-j}}$$

end

end

Complexity: $n-1 + (n-2) + \dots + 1 = \frac{n(n-1)}{2} = O(n^2)$

Obs: $P(x) = y[x_0] + y[x_0, x_1](x - x_0) + \dots + y[x_0, \dots, x_n](x - x_0)\dots(x - x_{n-1})$

$$P(x) = \left(\left(y[x_0, \dots, x_n](x - x_{n-1}) + y[x_0, \dots, x_{n-1}]\right)(x - x_{n-2}) + y[x_0, \dots, x_{n-2}]\right)(x - x_{n-3})$$

Algorithm (to evaluate $P(x)$)

Input: $y[x_0, \ldots, x_i]$ in y_i and x_i and x

Output: $P(x)$ in p

$p = y_n$

for $i = n-1 : 1$

$\quad p = p(x - x_i) + y_i$

end

Complexity $= n-1 = O(n)$

<u>Example</u> $f(x) = \dfrac{1}{1+25x^2}$ $\quad -5 \le x \le 5$

$x_i = -5+i \qquad 0 \le i \le 10 \qquad y_i = f(x_i)$

What went wrong?

1) Maybe chossing equidistant points is not that good

2) Maybe it is better to approximate the function by many different low degree polynomials. One in each little interval.

First devide the whole interval in many small intervals.

Notation $y_i = f(x_i) \qquad y[x_0, \dots, x_k] = f[x_0, \dots, x_k]$

P = polynomial of degree n that satisfies $P(x_i) = y_i \quad 0 \le i \le n$

Goal: How big is $f(x) - P(x)$ for x in $\text{int}(x_0, \dots, x_n)$

$\text{int}(x_0, \dots, x_n) = (a, b)$ where $a = \min\{x_i\}$ $b = \max\{x_i\}$

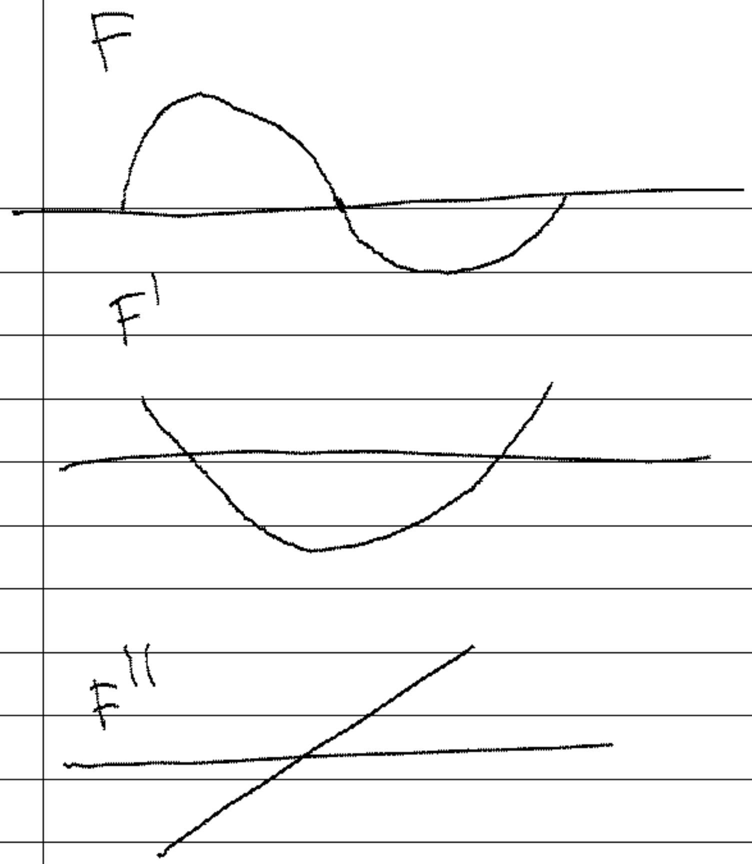
<u>Obs</u>: There exists $\xi \in \text{int}(x_0, \dots, x_n)$ such that

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

<u>proof</u>: $\Phi(z) = (z - x_0)(z - x_1) \cdots (z - x_n)$

$$G(z) = f(z) - P(z) - \frac{(f(x) - P(x))}{\Phi(x)} \times \Phi(z)$$

$G(z) = 0$ for $z = x_0, x_1, \dots, x_n$ and also $z = x$



If F has $n+2$ zeros in (a,b)
then $F^{(n+1)}$ has a zero in (a,b)

then, there exist $\xi \in \text{int}(x_0, \dots, x_n)$ such that $G^{(n+1)}(\xi) = 0$

$$G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \underbrace{P^{(n+1)}(\xi)}_{=0} - \frac{f(x) - P(x)}{\Phi(x)} \underbrace{\Phi^{(n+1)}(\xi)}_{=(n+1)!} = 0$$

$$f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Phi(x)$$

Obs: $\Phi(x) = (x-x_0)(x-x_1)\cdots\cdots(x-x_n)$

Goal: Given (a,b), select $x_0, x_1, \dots, x_n \in [a,b]$ so that

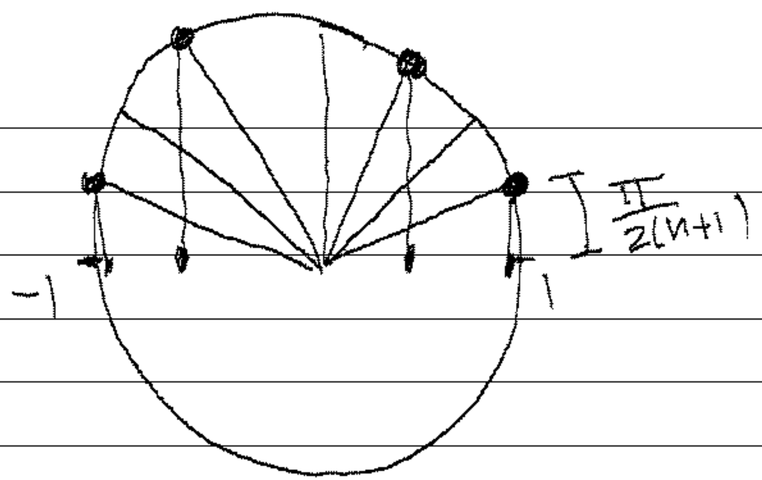
$\max_{x\in[a,b]} |\Phi(x)|$ is as small as possible

Notation $\|\Phi\|_\infty = \max_{x\in[a,b]} |\Phi(x)|$

Obs: The points $x_0, \dots, x_n$ that make $\|\Phi\|_\infty$ as small as possible are the Tchebycheff points

$$x_k = \frac{(b-a)}{2} \cos\left[\left(\frac{2(n-k)+1}{n+1}\right)\frac{\pi}{2}\right] + \frac{(a+b)}{2}$$

$k = 0, \dots, n$



Splines

Obs: Approximating a function f over an interval $[a,b]$ by a polynomial p, does not work very well always. It is better to split the interval $[a,b]$ into n little intervals and approximate f by a different low degree polynomial in each little interval.

Def. Let $a = x_0 < x_1 < \dots < x_n = b$. A spline function of degree d with nodes at the points x_i, $i = 0, 1, \dots, n$ is a function $s(x)$ with the properties

a) On each interval $[x_{i-1}, x_i]$, $s(x)$ is a polynomial of degree d

b) $s(x)$ and its first $(d-1)$ derivatives are continuous in $[a, b]$