Chpt 1

Goal: Avoid errors due to finite precision

- underflow / overflow
- loss of significant digits
- truncation error (Taylor's Thm)

1.3 Convergence Rate of

- Sequences: $\alpha_n = \alpha + O(n^{-p})$
- Functions: $F(h) = L + O(h^p)$

generally interested in largest p

$|\alpha_n - \alpha| \le K n^{-p} \qquad n \ge n_0$

$|F(h) - L| \le K h^p \qquad h \le h_0$

Chp 2: Root finding / Fixed points.

Goal: find p s.t. $f(p) = 0$

or p s.t. $g(p) = p$

Tools: IVT, MVT

Algorithms:

- Bisection
  - requirements: $f \in C[a,b]$, $f(a) \cdot f(b) < 0$
  - linearly convergent
  - limited to 1D (generally)
- Fixed Point Iteration: $p_n = g(p_{n-1})$
  - require:
    - $g \in C[a,b]$, $g(x) \in [a,b]$ $\forall x \in [a,b]$ (existence)
    - $|g'(x)| \leq k < 1$ for $k \in (0,1)$ $\forall x \in [a,b]$ (uniqueness)

- Error bounds

$$|p_n - p| \le k^n \max\{|p_0 - a|, |p_0 - b|\}$$

$$|p_n - p| \le \frac{k^n}{1-k} |p_1 - p_0|$$

- Newton's Method

  - $$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

  - Converges quadratically if simple root ($f'(p) \neq 0$) and p_0 close enough to p

- Secant Method

  - $$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

  - superlinear convergence

- Method False Position (secant with root bracketing)

- # Convergence Rate/Order

<u>linear</u> if

$$\lim_{n\to\infty} \underbrace{\frac{|p_{n+1}-p|}{|p_n-p|}}_{\lambda_n} = \lambda \qquad 0 \le \lambda \le 1$$

Superlinear if $\lambda = 0$

linear if $0 < \lambda < 1$

sublinear if $\lambda = 1$

quadratic if

$$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|^2} = \lambda$$

- Modified Newton:
  - apply Newton's to $\mu(x) = \frac{f(x)}{f'(x)}$

- $p_n = p_{n-1} - \dfrac{\mu(p_{n-1})}{\mu'(p_{n-1})}$
- root always simple $\Rightarrow$ quadratic convergence.

Accelerating linearly-convergent sequences $\{p_n\}$.

<u>Aitken's Δ^2 method:</u>

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} = \{\Delta^2\}(p_n)$$

(superlinear)

<u>Steffensen's</u>

$k=0 \quad p_0^{(0)}, \quad p_1^{(0)} = g(p_0^{(0)}), \quad p_2^{(0)} = g(p_1^{(0)})$

$k=1 \quad p_0^{(1)} = \{\Delta^2\} p_0^{(0)}, \quad p_1^{(1)} = g(p_0^{(1)}), \quad p_2^{(1)} = g(p_1^{(1)}$

⋮

- Quadratic convergence if p_0 close

Horners Method: fast evaluation of poly.

Deflation: estimate all roots of poly. by successively applying Newton's

Chpt 3: Interpolation.

Goal: given $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$.

Find function (polynomial) $p_n(x)$ s.t.

$$p_n(x_i) = f_i \qquad i = 0, 1, \dots, n$$

Polynomial interpolation

Power series: $p_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$

Newton Form: $p_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)\cdots(x-x_{n-1})$

Langrange Poly:

$$L_{n,j}(x) = \frac{(x-x_0)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}$$

$$P_n(x) = \sum_{j=0}^{n} f_j L_{n,j}(x)$$

Linear Splines: linear interpolation on each subinterval

Cubic Splines: cubic interpolation on each subinterval

Cubic B-Splines: use basis $B_j(x)$ for cubic splines

smooth and accurate!

Num. Diff.

Fwd Diff: $f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$

Bwd Diff: $f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$

Centered Diff : $$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$ 140

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

Num Integration

NC Rules : Equidistant Nodes

Open Rules : exclude boundary pts

Closed Rules : include boundary pts

Common rules (Midpt., Trap., Simpson's)

Composite NC Rules

- Use low order NC rule on partitioned subinterval

Gauss Rules

- optimized nodes/weights for DOP $2n+1$
- nodes are the roots of Legendre Polynomials.