2.4 Error Analysis for Iterative Methods

Definition 2.7. Order of Convergence

Suppose $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges to p with $p_n \neq p$ for all n. If positive constants λ and α exist with

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

then $\{p_n\}_{n=0}^{\infty}$ is said to **converges to p of order α with asymptotic error constant λ.**

A higher order convergence means that the sequence converges more rapidly.

- Special cases
  1. If $\alpha = 1$ (and $\lambda < 1$), the sequence is **linearly convergent**
  2. If $\alpha = 2$, the sequence is **quadratically convergent**

Linear vs. Quadratic Convergence

Suppose we have two sequences converging to 0 with:

$$\lim_{n\to\infty} \frac{|p_{n+1}|}{|p_n\ |} = 0.9, \qquad \lim_{n\to\infty} \frac{|q_{n+1}|}{|q_n\ |^2} = 0.9$$

Roughly we have:

$$|p_n| \approx 0.9|p_{n-1}| \approx \cdots \approx 0.9^n|p_0|,$$

$$|q_n| \approx 0.9|q_{n-1}|^2 \approx \cdots \approx 0.9^{2^n-1}|q_0|,$$

Assume $p_0 = q_0 = 1$

n	p_n	q_n
0	1	1
1	0.9	0.9
2	0.81	0.729
3	0.729	0.4782969
4	0.6561	0.205891132094649
5	0.59049	0.0381520424476946
6	0.531441	0.00131002050863762
7	0.4782969	0.00000154453835975
8	0.43046721	0.00000000000021470

Convergence of Fixed-Point Iteration

- **Theorem 2.8**

Let $g \in C[a,b]$ be such that $g(x) \in [a,b]$ for all $x \in [a,b]$. Suppose g' is continuous on (a,b) and that $0 < k < 1$ exists with $|g'(x)| \leq k$ for all $x \in (a,b)$.

If $g'(p) \neq 0$, then for all number p_0 in $[a,b]$, the sequence $p_n = g(p_{n-1})$ converges only **linearly** to the **unique fixed point** p in $[a,b]$.

Proof:

$$p_{n+1} - p = g(p_n) - g(p) = g'(\xi_n)(p_n - p), \xi_n \in (p_n, p)$$

Since $\{p_n\}_{n=0}^{\infty}$ converges to p, $\{\xi_n\}_{n=0}^{\infty}$ converges to p.

Since g' is continuous, $\lim_{n\to\infty} g'(\xi_n) = g'(p)$

$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|} = \lim_{n\to\infty} \left|g'(\xi_n)\right| = |g'(p)| < 1 \Rightarrow$ linear convergence

Comparison of fixed-point iteration and Newton's method

Revisit **MATLAB DEMO:EX1** in Lec 2.3

Consider the function $f(x) = \cos(x) - x$. Solve $f(x) = 0$ using

(a) fixed-point method with $g(x) = \cos(x)$,

(b) Newton's method.

Recall: to reach an accuracy of 10^{-10}, we need 53 iterations for fixed-pt algorithm, but only 4 for Newton's method

The slow convergence of fixed-pt algorithm can be explained by **Theorem 2.8.** Why Newton's method converges much faster?

Speed up Convergence of Fixed Point Iteration

- If we look for faster convergence methods, we must have $g'(p) = 0$ where p is the fixed-point.

Theorem 2.9

Let p be a solution of $x = g(x)$. Suppose $g'(p) = 0$ and g'' is continuous with $|g''(x)| < M$ on an open interval I containing p. Then there exists a $\delta > 0$ such that for $p_0 \in [p - \delta, p + \delta]$, the sequence defined by $p_{n+1} = g(p_n)$, when $n \geq 0$, converges ***at least quadratically*** to p. For sufficiently large n

$$|p_{n+1} - p| < \frac{M}{2} |p_n - p|^2$$

Newton's Method as Fixed-Point Problem

Consider to solve $f(x) = 0$ by Newton's method:

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

Let's define function $g(x)$ by $g(x) = x - \frac{f(x)}{f'(x)}$. The zero p of $f(x) = 0$ is also the fixed-point of $g(x)$ (assuming $f'(p) \neq 0$).

Compute $g'(x)$ to see: $g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2}$

Thus $g'(p) = 1 - \frac{\left(f'(p)\right)^2 - (0)f''(p)}{\left(f'(p)\right)^2} = 0.$

Note: Newton's method will converge at least quadratically if $f(p) = 0$ and $f'(p) \neq 0$.

Multiple Roots

- Newton's method and Secant method have difficulty to solve $f(x) = 0$ when $f(p) = 0$ and $f'(p) = 0$.
- How to modify Newton's method when $f'(p) = 0$? Here p is the root of $f(x) = 0$.
- **Definition 2.10. Multiplicity of a Root**

  A solution p of $f(x) = 0$ is a zero of multiplicity m of f if for $x \neq p$, we can write $f(x) = (x-p)^m q(x)$, where $\lim_{x\to p} q(x) \neq 0$.
- **Theorem 2.11**

  $f \in C^1[a,b]$ has a **simple zero** at p in (a,b) if and only if $f(p) = 0$, but $f'(p) \neq 0$.
- **Theorem 2.12**

  The function $f \in C^m[a,b]$ has a zero of multiplicity m at point p in (a,b) if and only if $0 = f(p) = f'(p) = f''(p) = \cdots = f^{(m-1)}(p)$, but $f^{(m)}(p) \neq 0$

Example 1.

Let $f(x) = e^x - x - 1$. Show that f has a zero of multiplicity 2 at $x = 0$.

Modified Newton's Method for Zeroes of Higher Multiplicity ($m > 1$)

Define the new function $\mu(x) = \frac{f(x)}{f'(x)}$.

Write $f(x) = (x-p)^m q(x)$, hence

$$\mu(x) = \frac{f(x)}{f'(x)} = (x-p)\frac{q(x)}{mq(x) + (x-p)q'(x)}$$

Note that $f(p) = 0$ and p is a simple zero of $\mu(x)$.

- Apply Newton's method to solve $\mu(x) = 0$ to give:

$$x = g(x) \equiv x - \frac{\mu(x)}{\mu'(x)}$$

$$= x - \frac{f(x)f'(x)}{[f'(x)]^2 - f(x)f''(x)}$$

- Quadratic convergence of the modified Newton's method:

$$p_n = p_{n-1} - \frac{f(p_{n-1})f'(p_{n-1})}{[f'(p_{n-1})]^2 - f(p_{n-1})f''(p_{n-1})}$$

Drawbacks of modified Newton's method:

- Compute $f''(x)$ is expensive
- Iteration formula is more complicated – more expensive to compute
- Roundoff errors in denominator – both $f'(x)$ and $f(x)$ approach zero.

Example 2. (MATLAB)

Let $f(x) = e^x - x - 1$. Use Newton's method and modified Newton's method to solve $f(x) = 0$ with $p_0 = 1$.

Newton

```
% -----------------------inputs------------------------------
f = @(x) exp(x)-x-1;
df = @(x) exp(x)-1; % function derivative
p0 = 1;
TOL = 1e-5; NI = 100;
% -------------------------------------------------------

% STEP 1
i = 1;
converge = false; % convergence flag
% STEP 2
while i<=NI
    % STEP 3: compute p(i)
    p = p0-f(p0)/df(p0);
    err = abs(p-p0);
    % STEP 4: check if meets the stopping criteria
    if (err< TOL)
        converge = true; break
    else
        i = i+1; % STEP 5
        p0 = p;  % STEP 6: update p0
    end
end

if converge
    fprintf('\n\nApproximate solution P = %.8f\n',p)
    fprintf('With F(P) = %.3e\n',f(p))
    fprintf('Number of iterations = %3i\n',i)
    fprintf('Tolerance = %.3e  |p-pold| = %.3e\n',TOL, err)
end
```

```
Approximate solution P = 0.00000542
With F(P) = 1.472e-11
Number of iterations =  18
Tolerance = 1.000e-05  |p-pold| = 5.425e-06
```

Modified Newton

```
% -----------------------inputs------------------------------
f = @(x) exp(x)-x-1;
df = @(x) exp(x)-1; % function derivative
ddf = @(x) exp(x); % second derivative
p0 = 1;
TOL = 1e-5; NI = 100;
% -------------------------------------------------------

% STEP 1
i = 1;
converge = false; % convergence flag
% STEP 2
while i<=NI
    % STEP 3: compute p(i)
    f0 = f(p0);
    df0 = df(p0);
    ddf0 = ddf(p0);
    p = p0-f0*df0/(df0^2-f0*ddf0);
    err = abs(p-p0);
    % STEP 4: check if meets the stopping criteria
    if (err< TOL)
        converge = true; break
    else
        i = i+1;
        p0 = p; % update p0
    end
end

if converge
    fprintf('\n\nApproximate solution P = %.8f\n',p)
    fprintf('With F(P) = %.3e\n',f(p))
    fprintf('Number of iterations = %3i\n',i)
    fprintf('Tolerance = %.3e  |p-pold| = %.3e\n',TOL, err
end
```

```
Approximate solution P = -0.00000000
With F(P) = 0.000e+00
Number of iterations =   5
Tolerance = 1.000e-05  |p-pold| = 0.000e+00
```