Lecture 25
Solutions of Equations in One Variable

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Root-finding Problem

This process involves finding a root, or solution, of an equation of the form $f(x) = 0$, for a given function f. A root of this equation is also called a zero of the function f.

- Bisection Method
- Fixed-Point Iteration
- Newton's Method
- Müller's Method

The Bisection Method

The Bisection (Binary-search) Method

- Suppose f continuous on $[a,\ b]$, and $f(a), f(b)$ opposite signs
- By the IVT, there exists an x in $(a,\ b)$ with $f(x) = 0$.
- Divide the interval $[a,\ b]$ by computing the midpoint

$$p = (a + b)/2$$

- If $f(p)$ has same sign as $f(a)$, consider new interval $[p,\ b]$
- if $f(p)$ has same sign as $f(b)$, consider new interval $[a,\ p]$
- Repeat until interval small enough to approximate x well.

The Bisection Method – Implementation

MATLAB Code

```
function p = bisection(f, a, b, tol, MaxIter)
iter = 0;
while 1
   iter = iter + 1;
   if iter > Maxter, break; end
   p = (a + b)/2;
   if p - a < tol, break; end
   if f(a) * f(p) > 0
     a = p;
   else
     b = p;
   end
end
```

Bisection Method –Termination Criteria

Termination Criteria

We select a tolerance $\epsilon > 0$ and generate $p_1, \ldots, p_N$ until one the following conditions is met:

- Many ways to decide when to stop:

$$|p_N - p_{N-1}| < \epsilon$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon$$

$$|f(p_N)| < \epsilon$$

- None is perfect, use a combination in real software.

Bisection Method – Convergence

Theorem

Suppose that $f \in C[a,\ b]$ *and* $f(a) \cdot f(b) < 0$. *The Bisection method generates a sequence* $\{p_n\}_{n=1}^{\infty}$ *approximating a zero* p *of* f *with*

$$|p_n - p| \leqslant \frac{b-a}{2^n}, \quad \textit{when } n \geqslant 1.$$

Convergence Rate

The sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence $O(1/2^n)$:

$$p_n = p + O\left(\frac{1}{2^n}\right).$$

Fixed Points

Fixed Points and Root-Finding

- A number p is a fixed point for a given function g if $g(p) = p$.
- Given a root-finding problem $f(p) = 0$, there are many g with fixed points at p:

$$g(x) = x - f(x)$$

$$g(x) = x + 3f(x)$$

$$\cdots$$

- If g has fixed point at p, then $f(x) = x - g(x)$ has a zero at p.

Existence and Uniqueness of Fixed Points

Theorem

- *If* $g \in C[a,\ b]$ *and* $g(x) \in [a,\ b]$ *for all* $x \in [a,\ b]$*, then* g *has a fixed point in* $[a,\ b]$*.*
- *If, in addition,* $g'(x)$ *exits on* $(a,\ b)$ *and a positive constant* $k < 1$ *exists with*

$$|g'(x)| \leqslant k,\ \forall x \in (a,\ b),$$

then the fixed point in $[a,\ b]$ *is unique.*

Fixed-Point Iteration

Fixed-Point Iteration

- For initial p_0, generate sequence $\{p_n\}_{n=0}^{\infty}$ by $p_n = g(p_{n-1})$.
- If the sequence converges to p, then

$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g(\lim_{n\to\infty} p_{n-1}) = g(p).$$

MATLAB Code

```
function p = fixedpoint(g, p0, tol, MaxIter)
iter = 0;
while 1
   iter = iter + 1; if iter > MaxIter, break; end
   p = g(p0);
   if abs(p - p0) < tol, break; end
   p0=p;
end
```

Convergence of Fixed-Point Iteration

Theorem (Fixed-Point Theorem)

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

$$|g'(x) \leqslant k|, \text{ for all } x \in (a,\ b).$$

Then, for any number p_0 in $[a,\ b]$, the sequence defined by $p_n = g(p_{n-1})$ converges to the unique fixed point p in $[a,\ b]$.

Corollary

If g satisfies the hypotheses above, then bounds for the error are given by

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

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

Newton's Method

Taylor Polynomial Derivation

Suppose $f \in C^2[a,\ b]$ and $p_0 \in [a,\ b]$ approximates solution p of $f(x) = 0$ with $f'(p_0) \neq 0$. Expand $f(x)$ about p_0:

$$f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p))$$

Set $f(p) = 0$, assume $(p - p_0)^2$ neglibible:

$$p \approx p_1 = p_0 - \frac{f(p_0)}{f'(p_0)}$$

This gives the sequence $\{p_n\}_{n=0}^{\infty}$:

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

Newton's Method – Implementation

MATLAB Code

```
function p = newton(f, df, p0, tol, MaxIter)
iter = 0;
while 1
  iter = iter + 1;
   if iter > MaxIter, break; end
   p = p0 - f(p0)/df(p0);
   if abs(p - p0) < tol, break; end
   p0 = p;
end
```

Newton's Method – Convergence

Fixed Point Formulation

Newton's method is fixed point iteration $p_n = g(p_{n-1})$ with

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

Theorem

Let $f \in C^2[a,\ b]$. If $p \in [a,\ b]$ is such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p - \delta,\ p + \delta]$.

Variants without Derivatives

The Secant Method

Replace the derivative in Newton's method by

$$f'(p_{n-1}) \approx \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}$$

to get

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

Method of False Position (Regula Falsi)

Like the Secant method, but with a test to ensure the root is bracketed between iterations.

Secant Method: MATLAB Code

function p = secant(f, p0,p1,tol,MaxIter)

$iter = 0;$

while 1

  $iter = iter + 1;$

  if $iter >$ MaxIter, break; end

  ptmp = p1;

  $p1 = p_1 - f(p1)(p1 - p0)/(f(p1) - f(p0))$

  if $abs(p1 - ptmp) < tol$, break; end

  p0 = ptmp;

end

Order of Convergence

Definition

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 $\{p_n\}_{n=0}^{\infty}$ converges to p of order α, with asymptotic error constant λ. An iterative technique $p_n = g(p_{n-1})$ is said to be of order α if the sequence $\{p_n\}_{n=0}^{\infty}$ converges to the solution $p = g(p)$ of order α.

Special Cases

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

Fixed Point Convergence

Theorem

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

Theorem

Let p be solution of $x = g(x)$. Suppose $g'(p) = 0$ and g'' continuous with $|g''(x)| < M$ on open interval I containing p. Then there exists $\delta > 0$ s.t. for $p_0 \in [p-\delta, p+\delta]$, the sequence defined by $p_n = g(p_{n-1})$ converges at least quadratically to p, and

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

Newton's Method as Fixed-Point Problem

Derivation

Seek g of the form

$$g(x) = x - \phi(x)f(x).$$

Find differentiable ϕ giving $g'(p) = 0$ when $f(p) = 0$:

$$g'(x) = 1 - \phi'(x)f(x) - f'(x)\phi(x)$$

$$g'(p) = 1 - \phi'(p) \cdot 0 - f'(p)\phi(p)$$

and $g'(p) = 0$ if and only if $\phi(p) = 1/f'(p)$. This gives Newton's method

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

Multiplicity of Zeros

Definition

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

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

Theorem

The function $f \in C^m[a,\ b]$ has a zero of multiplicity m at point p in $(a,\ b)$ if and only if

$$0 = f(p) = f'(p) = f''(p) = \cdots = f^{(m-1)}(p), \ \text{but}\ f^{(m)}(p) \neq 0$$

Variants for Multiple Roots

Newton's Method for Multiple Roots

Define $\mu(x) = f(x)/f'(x)$. If p is a zero of f of multiplicity m and $f(x) = (x-p)^m q(x)$, then

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

also has a zero at p. But $q(p) \neq 0$, so

$$\frac{q(p)}{mq(p) + (p-p)q'(p)} = \frac{1}{m} \neq 0,$$

and p is a simple zeros of μ. Newton's method can be applied to μ to give

$$g(x) = x - \frac{f(x)f'(x)}{[f'(x)]^2 0 - f(x)f''(x)}$$

Aitken's Δ^2 Method

Accelerating linearly convergent sequences

- Suppose $\{p_n\}_{n=0}^{\infty}$ linearly convergent with limit p
- Assume that

$$\frac{p_{n+1} - p}{p_n - p} \approx \frac{p_{n+2} - p}{p_{n+1} - p}$$

- Solving for p gives

$$p \approx \frac{p_{n+2}p_n - p_{n+1}^2}{p_{n+2} - 2p_{n+1} + p_n} = \cdots = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}$$

- Use this for new more rapidly converging sequence $\{\hat{p}_n\}_{n=0}^{\infty}$:

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

Delta Notation

Definition

For a given sequence $\{p_n\}_{n=0}^{\infty}$, the forward difference Δp_n is defined by

$$\Delta p_n = p_{n+1} - p_n, \text{ for } n \geqslant 0$$

Higher powers of the operator Δ are defined recursively by

$$\Delta^k p_n = \Delta(\Delta^{k-1} p_n), \text{ for } k \geqslant 2$$

Aitken's Δ^2 method using delta notation

Since $\Delta^2 p_n = p_{n+2} - 2p_{n+1} + p_n$, we can write

$$\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}, \text{ for } n \geqslant 0$$

Convergence of Aitken's Δ^2 Method

Theorem

Suppose that $\{p_n\}_{n=0}^{\infty}$ *converges linearly to* p *and that*

$$\lim_{n\to\infty} \frac{p_{n+1}-p}{p_n-p} < 1$$

Then $\{\hat{p}_n\}_{n=0}^{\infty}$ *converges to* p *faster that* $\{p_n\}_{n=0}^{\infty}$ *in the sense that*

$$\lim_{n\to\infty} \frac{\hat{p}_n-p}{p_n-p} = 0$$

Steffensen's Method

Accelerating fixed-point iteration

Aitken's Δ^2 method for fixed-point iteration gives

$$p_0, p_1 = g(p_0), p_2 = g(p_1), \hat{p}_0 = \{\Delta^2\}(p_0),$$

$$p_3 = g(p_2), \hat{p}_1 = \{\Delta^2\}(p_1), \dots$$

Steffensen's method assumes $\hat{p}_0$ is better than p_2:

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

Theorem

Suppose $x = g(x)$ *has solution* p *with* $g'(p) \neq 1$. *If exists* $\delta > 0$ *s.t.* $g \in C^3[p - \delta,\ p + \delta]$, *then Steffensen's method gives quadratic convergence for* $p_0 \in [p - \delta,\ p + \delta]$.

Steffensen's Method – Implementation

MATLAB Code

```
function p = steffensen(g,p0,tol,MaxIter)
iter = 0;
while 1
    iter = iter + 1;
    if iter > MaxIter, break; end
    p1 = g(p0);
    p2 = g(p1);
    p = p0 - (p1 - p0)^2/(p2 - 2*p1 + p0);
    if abs(p - p0) < tol, break; end
    p0 = p;
end
```

Zeros of Polynomials

Polynomial

A polynomial of degree n has the form
$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + x_0$ with coefficients a_i and $a_n \neq 0$.

Theorem (Fundamental Theorem of Algebra)

If $P(x)$ polynomial of degree $n \geqslant 1$, with real or complex coefficients, $P(x) = 0$ has at least one root.

Corollary

- Exists unique $x_1, \ldots, x_k$ and $m_1, \ldots, m_k$, with $\sum_{i=1}^{k} m_i = n$ and
$$P(x) = a_n (x - x_1)^{m_1} (x - x_2)^{m_2} \cdots (x - x_k)^{m_k}.$$
- $P(x), Q(x)$ polynomials of degree at most n. If $P(x_i) = Q(x_i)$ for $i = 1, 2, \ldots, k$, with $k > n$, then $P(x) = Q(x)$.

Horner's Method

Theorem (Horner's Method)

Let $P(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$. *If* $b_n = a_n$ *and*

$$b_k = a_k + b_{k+1}a_0, \text{ for } k = n-1, n-2, \ldots, 1, 0,$$

then $b_0 = P(x_0)$. *Moreover, if*

$$Q(x) = b_n x^{n-1} + b_{n-1}x^{n-2} + \ldots + b_2 x + b_1,$$

then $P(x) = (x - x_0)Q(x) + b_0$.

Computing Derivatives

Differentiation gives

$$P'(x) = Q(x) + (x - x_0)Q'(x), \text{ and } P'(x_0) = Q(x_0)$$

Horner's Method – Implementation

MATLAB Code

```
function [y, z] = horner(a,x)
n = length(a) - 1;
y = a(1);
z=a(1);
for j=2:n
   y = x*y+a(j);
   z = x*z+y;
end
y = x*y+a(n+1);
```

Deflation

Deflation

- Compute approximate root $\hat{x}_1$ using Newton. Then

$$P(x) \approx (x - \hat{x}_1) Q_1(x).$$

- Apply recursively on $Q_1(x)$ until the quadratic factor $Q_{n-2}(x)$ can be solved directly
- Improve accuracy with Newton's method on original $P(x)$.

Müller's Method

Müller's Method

- Similar to the Secant method, but parabola instead of line
- Fit quadric polynomial $P(x) = a(x-p_2)^2 + b(x-p_2) + c$ that passes through $(p_0, f(p_0)), (p_1, f(p_1)), (p_2, f(p_2))$.
- Solve $P(x) = 0$ for p_3, choose root closest to p_2:

$$p_3 = p_2 - \frac{2c}{b + sgn(b)\sqrt{b^2 - 4ac}}$$

- Repeat until convergence
- Relatively insensitive to initial p_0, p_1, p_2, but e.g., $f(p_i) = f(p_{i+1}) = f(p_{i+2}) \neq 0$ gives problem.