Lecture 04: Rootfinding
-or- solving non-linear equations \( f(x) = 0 \)

Outline

0) Last point on Truncation and Floating point errors
1) Introduction to rootfinding problems \( f(x) = 0 \)
   Examples: FTV Retirement problem
2) Method 1: Fixed Point iteration \( x = g(x) \)
   Example problems: (some work, some don't)
   A) Retirement problem
   B) \( x = \exp(-x) \)
   C) \( x = -\ln(x) \)
3) Convergence and Analysis of fixed point iteration
4) More robust algorithms
   Bisection
   Newton's method
   Secant Method
   Brent's method and fsolve()

Floating Point and Truncation Error

In many numerical schemes both truncation error and floating point error can contribute to the overall error

Example #1: Finite Difference approximation to \( f'(x) \)

let \( f(x) = \exp(x), \ f(1) = e, \ f'(1) = e \)

\[
\begin{align*}
    f'(x) & = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \\
    f'(1) & \approx \frac{\exp(1+h) - \exp(1)}{h} + O(h)
\end{align*}
\]

Onward to Rootfinding or solving non-linear equations of 1 variable \( f(x) = 0 \)

A useful example: The Future Time Value Annuity problem (or when can I retire)

\[
A = \frac{P}{r/m} \left[ \left( 1 + \frac{r}{m} \right)^{mn} - 1 \right]
\]

where

\( P \): payment amount per compounding period
\( m \): number of compounding periods per year
\( r \): Annual interest rate
\( n \): number of years to retirement
\( A \): total value after \( n \) years
Question: What interest rate \( r \) will give me a Million dollars in 20 years (compounded monthly)?

i.e. set \( P=18,000/12 = \$1500 \), \( m=12 \), \( n=20 \)

\[
A(r) = \frac{P \cdot r(1+r)^{mn}}{r-(1+r)^{mn}}
\]

Clearly a solution, but how to find?

First approach: fixed point iteration

rewrite the problem as

\[
A = \frac{P}{r}((1+r)^{mn}-1)
\]

\[
r = \frac{P}{A}((1+r)^{mn}-1)
\]

Fixed Point iteration example #2

solve \( f(x) = x - \exp(-x) = 0 \)

or \( x = \exp(-x) \)

or \( x = g(x) \) with \( g(x) = \exp(-x) \)

Fixed Point iteration example #3

solve \( f(x) = \log(x)+x=0 \)

or \( x = -\log(x) \)

or \( x = g(x) \) with \( g(x) = -\log(x) \)
Lecture 04
January 31, 2008

What's going on here?

Converges

Doesn't Converge

Analysis of Fixed Point Iteration

Theorem 1: Existence and uniqueness of fixed-points

Assume \( g \in C[a,b] \), if the range of mapping \( y = g(x) \) satisfies \( y \in [a,b] \) for all \( x \in [a,b] \) the \( g \) has a fixed point in \([a,b]\).

Furthermore: suppose \( g'(x) \) is defined over \((a,b)\) and that a positive constant \( K < 1 \) exists with \( |g'(x)| \leq K < 1 \) for all \( x \in (a,b) \); then \( g \) has a unique fixed point \( P \) in \([a,b]\).

What's going on here?

Converges

Doesn't Converge

Asymptotic convergence behavior of fixed point iterations

\[ x_{k+1} = g(x_k) \]

1) assume \( g \) has a fixed point at \( x^* \) such that \( x^* = g(x^*) \)

\[
\begin{align*}
x_{k+1} & = x^* + e_{k+1} \\
x_k & = x^* + e_k \\
e_{k+1} & = g(x_k) + g'(x^*)e_k + \frac{g''(x^*)}{2}e_k^2 \\
e_{k+1} & \rightarrow 0 \quad \text{as} \quad |g'(x^*)| < 1 \\
e_{k+1} & \rightarrow C |e_k| \
\end{align*}
\]
Convergence of iterative schemes:

Given any iterative scheme where

\[ |e_{k+1}| = C |e_k|^n \]

Define:

- **Linear Convergence**: \( n=1 \) \( C<1 \) (\( C>1 \) divergent)
- **Quadratic Convergence**: \( n=2 \)
- **Superlinear Convergence**: \( n > 1 \)

Back to our examples:

1) \( g(x) = \exp(-x) \) (\( x^* \approx 0.56 \))
   - \( g'(x^*) \approx 2.14 \)

2) \( g(x) = -\log(x) \) (\( x^* \approx 0.56 \))

3) \( g(x) = \frac{mP}{A} \left( \frac{1+x/m}{mn} \right) \) \( (x^* \approx .09) \)
   - \( g'(x^*) \approx 0.36 \)

Better ways of rootfinding: solve \( f(x)=0 \)

Point: If \( x \) is a fixed-point of \( g(x) \) then \( x \) is a "root" of \( f(x)=x-g(x) \) such that \( f(x)=0 \).

Example: for Annuity problem solve

\[ f(r) = r - \frac{mP}{A} \left( \frac{1-r/m}{mn} \right) = 0 \]

or alternatively just

\[ f(r) = A - \frac{mP}{r} \left( \frac{1-r/m}{mn} \right) = 0 \]

Better ways of rootfinding: solve \( f(x)=0 \)

\[ f(r) = A - \frac{mP}{r} \left( \frac{1-r/m}{mn} \right) = 0 \]

**Classical Methods:**
- Bisection (linear convergence)
- Newton’s Method (Quadratic)
- Secant Method (super-linear)

**Combined Methods**
- RootSafe (Newton + Bisection)
- Brent’s method (fzero in matlab)
Bracketing and Bisection:

**Definition**: a bracket is an interval \([a, b]\) such that \(\text{sgn}(f(a)) \neq \text{sgn}(f(b))\)

**Theorem**: if \(f(x)\) is \(C[a, b]\) and \(\text{sgn}(f(a)) \neq \text{sgn}(f(b))\) then there exists a number \(c \in (a, b)\) such that \(f(c) = 0\). (proof: Intermediate value theorem)

Bisection: the algorithm

Given a bracket \([a, b]\) and a function \(f(x)\)

**Initialize**: \(fa = f(a)\), \(fb = f(b)\), \(h = b - a\)

**While** \((h > \text{tol})\)

\[
C = a + \frac{h}{2}
\]

\[
f_c = f(r)
\]

if \(\text{sgn}(f_a) \neq \text{sgn}(f_c)\)

\[
e_{a1} = a1 - \frac{h}{2}
\]

\[
e_{b1} = b1 + \frac{h}{2}
\]

\[
e_{c1} = C - \frac{h}{2}
\]

\[
e_{c1} = C + \frac{h}{2}
\]

\[
e_{a1} = a1 - \frac{h}{2}
\]

\[
e_{b1} = b1 + \frac{h}{2}
\]

**End**

**Bisection**: Given a bracket \([a, b]\): cut the interval in half and find the new bracket. Repeat until the bracket width is < \(\text{tol}\). Given an initial bracket, bisection is guaranteed to converge.