Lecture 6

Iterations and Root Finding

L. Olson

Department of Computer Science
University of Illinois at Urbana-Champaign

2006.02.01

Recall

For a really smooth function $f(x)$ we can expand the function around a point $x_0 \in [a, b]$ with a Taylor Series as

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)\frac{df}{dx}|_{x=x_0} + \frac{(x-x_0)^2}{2}\frac{d^2f}{dx^2}|_{x=x_0} \\ &\quad + \cdots + \frac{(x-x_0)^n}{n!}\frac{d^nf}{dx^n}|_{x=x_0} + \ldots \\ &= f(x_0) + (x - x_0)\frac{df}{dx}|_{x=x_0} + \frac{(x-x_0)^2}{2}\frac{d^2f}{dx^2}|_{x=x_0} \\ &\quad + \cdots + \frac{(x-x_0)^n}{n!}\frac{d^nf}{dx^n}|_{x=x_0} + \frac{(x-x_0)^{n+1}}{(n+1)!}\frac{d^{n+1}f}{dx^{n+1}}|_{x=\xi} \end{aligned}$$

Recall

Then the function can be expressed as the n^{th} Taylor Series approximation and a truncation error:

$$\begin{aligned} f(x) &= f_n(x) + R_n(x) \\ &= \underbrace{f(x_0) + (x - x_0)\frac{df}{dx}|_{x=x_0} + \frac{(x-x_0)^2}{2}\frac{d^2f}{dx^2}|_{x=x_0} + \cdots + \frac{(x-x_0)^n}{n!}\frac{d^nf}{dx^n}|_{x=x_0}}_{f_n(x)} \\ &\quad + \underbrace{\frac{(x-x_0)^{n+1}}{(n+1)!}\frac{d^{n+1}f}{dx^{n+1}}|_{x=\xi}}_{R_n(x)} \end{aligned}$$

Convergence

Since $\frac{d^{n+1}f}{dx^{n+1}}$ is continuous on $[a, b]$ it is bounded, so

$$R_n(x) = \mathcal{O}(\frac{(x - x_0)^{(n+1)}}{(n+1)!}) = \mathcal{O}(h^{n+1})$$

We say that the truncation error is of ”order $n + 1$”

Questions:

1. how *fast* does the approximation converge?
2. how *expensive* is it to compute the approximation? (later)

Iterations

- With a Taylor Series, we are free to keep adding terms until we reach a certain tolerance in the approximate error

Start with f_0:

$$\begin{aligned}
f_0 &= f(x_0) \\
f_1 &= f_0 + (x - x_0)\frac{df}{dx}|_{x=x_0} \\
f_2 &= f_1 + (x - x_0)^2\frac{d^2f}{dx^2}|_{x=x_0} \\
&\vdots \\
f_k &= f_{k-1} + (x - x_0)^k\frac{d^kf}{dx^k}|_{x=x_0}
\end{aligned}$$

Better questions:

1. with approximations $f_k(x)$ and $f_{k+1}(x)$ of $f(x)$, how much better is f_{k+1}?
2. how fast is $f_k(x) \to f(x)$ with respect to k?

More General

Convergence rate

- Let $\{y_n\}_{n=1}^{\infty}$ be a sequence:

$$y_0, y_1, y_2, \ldots, y_j, \ldots$$

- If α and K exist so that

$$|y - y_n| \leqslant \frac{K}{n^{\alpha}}$$

  Then we say that $\{y_n\}$ **converges to** y **with rate of convergence** $\frac{1}{n^{\alpha}}$

- So we expect $|e_n| = |y - y_n| \approx c\frac{1}{n^{\alpha}}$

Convergence

- Then

$$\frac{|e_{n+1}|}{|e_n|} \approx \text{constant}$$

- In general, a sequence is said to converge with rate r if

$$\lim_{k\to\infty} \frac{|e_{n+1}|}{|e_n|^r} = C$$

Special Cases:

- If $r = 1$ and $C < 1$, then the rate is *linear*
- If $r > 1$, then the rate is *superlinear*
- If $r = 2$ and $C < 1$, then the rate is *quadratic*
- If $r = 3$ and $C < 1$, then the rate is *cubic*

Example

Convergence Rate

1. $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} \ldots$
2. $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8} \ldots$
3. $10^{-2}, 10^{-3}, 10^{-5}, 10^{-8} \ldots$
4. $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16} \ldots$

Example

Convergence Rate

1. $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$... (linear with $C = 10^{-1}$)
2. $10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$... (linear with $C = 10^{-2}$)
3. $10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$... (superlinear, not quadratic)
4. $10^{-2}, 10^{-4}, 10^{-8}, 10^{-16}$...(quadratic)

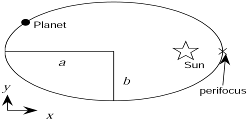
What Next?

We need to study some iterations.

- iteratively finding a root to an equation
- iteratively finding the solution to an algebraic system
- iteratively finding solutions to Ordinary Differential Equations (ODEs)
- ...

Planetary Motion

Courtesy of G. Wright



Consider a planet in orbit around the sun

- ω is the frequency of orbit
- t is the time since the planet was last closest to the sun (perifocus)
- $e = \sqrt{1 - \frac{b^2}{a^2}}$ is the eccentricity of the orbit
- Kepler's Law say the location of the planet at time is

$$x = a(\cos E - e)$$
$$y = a\sqrt{1 - e^2}\sin E$$

- $E = \omega t + e \sin E$ is the eccentric anomaly
- cannot explicitly express E

Planetary Motion

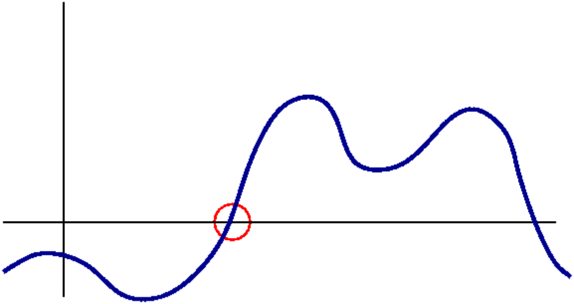
But we can write the eccentric anomaly as a root problem: Find the root of $F(E) = 0$ where

$$F(E) = \omega t + e \sin E - E$$

How?

Root Finding

Given a function $f(x)$, find x so that $f(x) = 0$



Topic Outline

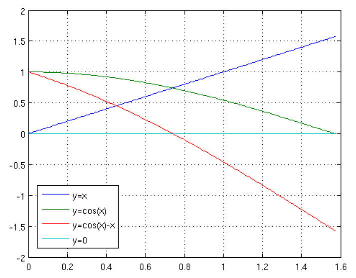
1. bracketing methods
2. fixed point iterations
3. Bisection Method
4. Newton's Method
5. Secant Method
6. Special Case: Roots of Polynomials

Roots of $f(x)$

- Any single valued function can be written as $f(x) = 0$

Example

- Find x so that $\cos(x) = x$
- That is, find where $f(x) = \cos(x) - x = 0$



Analyze your Application

- Is the function complicated to evaluate?
    - lots of expresions?
    - singularities?
    - simplify? polynomial?
- How accuracte does our root need to be?
- How fast/robust should our method be?

!

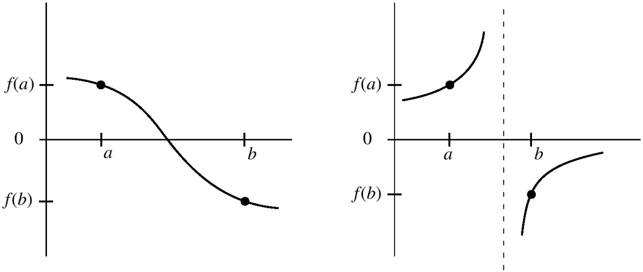
From this, you can pick the right method...

Basic Root Finding Strategy

1. Plot the function
    - Get an initial guess
    - Identify problematic parts
2. Start with the initial guess and iterate

Bracket Basics

- A root x is *bracketed* on $[a, b]$ if $f(a)$ and $f(b)$ have opposite sign.
- Changing signs does not mean bracketed, however: singularity



- helps get an initial guess

Bracket Algorithm

Algorithm 6.1   Bracket Roots

```
given: $f(x)$, $x_{\min}$, $x_{\max}$, $n$

$dx = (x_{\max} - x_{\min})/n$
$x_{\text{left}} = x_{\min}$
i = 0

while $i < n$
    $i \leftarrow i + 1$
    $x_{\text{right}} = x_{\text{left}} + dx$
    if $f(x)$ changes sign in $[x_{\text{left}}, x_{\text{right}}]$
        save $[x_{\text{left}}, x_{\text{right}}]$ for further root-finding
    end
    $x_{\text{left}} = x_{\text{right}}$
end
```

Testing Sign

$f(a) \times f(b) < 0$

Should we use?

```
fa = myfunc(a);
fb = myfunc(b);

if(fa*fb<0)
  (save)
end
```

Bestter Sign Test

!

Nope. Underflow...

`sign()`

Use matlab's sign

```
fa = myfunc(a);
fb = myfunc(b);

if(sign(fa) ~= sign(fb))
  (save)
end
```

brackPlot.m (in text)

Let's look at **brackPlot.m**:

- searches for brackets of user-defined $f(x)$
- plots the brackets
- returns the brackets

Usage:

```
Xb = brackPlot(fun,xmin,xmax)
Xb = brackPlot(fun,xmin,xmax,nx)
fun = (string) name of mfile function
```

Moving forward...

Bracketing is fine. But we need to find the actual root:

- Fixed Point Iteration
- Bisection
- Newton’s Method
- Secant Method

Process:

1. use brackPlot.m to get a visual and brackets
2. search brackets with these methods

Fixed Point Iteration

- simple
- not-robust: only works for nice functions

Definition

A **fixpoint** of a function is a number mapped to itself: $x = g(x)$

Goal

Write our equation

$$f(x) = 0$$

as

$$g(x) = x$$

Fixed Point Iteration

Given $x = g(x)$, take the first guess:

$$\begin{aligned} x_1 &= g(x_0) \\ x_2 &= g(x_1) \\ &\vdots \\ x_k &= g(x_{k-1}) \end{aligned}$$

Algorithm

```
pick $x_0$
for $k = 1, 2, \ldots$
   $x_k = g(x_{k-1})$
   stop if converged
end
```

Careful!

Automated root-finding implementations need to monitor progress and stop if necessary

- avoids over computation (convergence to unnecessary accuracy)
- can look at to successive iterates to see if there’s a stall
- can monitor the exact goal: $f(x) = 0$

Example

Consider

$$x - x^{1/3} - 2 = 0$$

How to write $x = g(x)$?

Example

Consider

$$x - x^{1/3} - 2 = 0$$

How to write $x = g(x)$?

rearrange

$$x_{k+1} = g_1(x_k) = x_k^{1/3} + 2$$

Example

Consider

$$x - x^{1/3} - 2 = 0$$

How to write $x = g(x)$?

rearrange

$$x_{k+1} = g_1(x_k) = x_k^{1/3} + 2$$

rearrange

$$x_{k+1} = g_2(x_k) = (x_k - 2)^3$$

Example

Consider

$$x - x^{1/3} - 2 = 0$$

How to write $x = g(x)$?

rearrange

$$x_{k+1} = g_1(x_k) = x_k^{1/3} + 2$$

rearrange

$$x_{k+1} = g_2(x_k) = (x_k - 2)^3$$

more later...

$$x_{k+1} = g_3(x_k) = \frac{6 + 2x^{1/3}}{3 - x^{-2/3}}$$