



2.29 Numerical Fluid Mechanics

Spring 2015 – Lecture 3

REVIEW Lecture 2

- **Approximation and round-off errors**

- Significant digits, true/absolute and relative errors
 - Iterative schemes and stop criterion, Machine precision
- Number representations
 - Floating-Point representation
- Quantizing errors and their consequences
- Arithmetic operations (e.g. radd.m)
- Errors of arithmetic/numerical operations
 - Large number of addition/subtractions, large & small numbers, inner products, etc
- Recursion algorithms (Heron, Horner's scheme, Bessel functions, etc)
 - Order of computations matter, Round-off error growth and (in)-stability

- **Truncation Errors, Taylor Series and Error Analysis**

- Taylor series:

$$f(x_{i+1}) = f(x_i) + \Delta x f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + \frac{\Delta x^3}{3!} f'''(x_i) + \dots + \frac{\Delta x^n}{n!} f^n(x_i) + R_n$$

- Use of Taylor series to derive finite difference schemes (first-order Euler scheme and forward, backward and centered differences)



2.29 Numerical Fluid Mechanics

Spring 2015 – Lecture 3

REVIEW Lecture 2, Cont'd

- Truncation Errors, Taylor Series and Error Analysis
 - General error propagation formulas and error estimation, with examples

Consider $y = f(x_1, x_2, x_3, \dots, x_n)$.

If ε_i 's are magnitudes of errors on x_i 's, what is the error on y ?

- The Differential Formula:
$$\varepsilon_y \leq \sum_{i=1}^n \left| \frac{\partial f(x_1, \dots, x_n)}{\partial x_i} \right| \varepsilon_i$$
- The Standard Error (statistical formula):
$$E(\Delta_s y) \approx \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \right)^2 \varepsilon_i^2}$$

– Error cancellation (e.g. subtraction of errors of the same sign)

– Condition number:
$$K_p = \frac{\bar{x} f'(\bar{x})}{f(\bar{x})}$$

- Well-conditioned problems vs. well-conditioned algorithms
- Numerical stability

Reference: Chapra and Canale,
Chapters 3 and 4



Numerical Fluid Mechanics: Lectures 3 and 4 Outline

- Roots of nonlinear equations

Reference: Chapra and Canale,
Chapter 5

- Bracketing Methods

- Example: Heron's formula
- Bisection
- False Position

- “Open” Methods

- Open-point Iteration (General method or Picard Iteration)
 - Examples
 - Convergence Criteria
 - Order of Convergence
- Newton-Raphson
 - Convergence speed and examples
- Secant Method
 - Examples
 - Convergence and efficiency
- Extension of Newton-Raphson to systems of nonlinear equations

- Roots of Polynomial (all real/complex roots)

- Open methods (applications of the above for complex numbers)
- Special Methods (e.g. Muller's and Bairstow's methods)



Error Propagation Example with Differential Approach: Multiplications

Multiplication

Error Propagation Formula

$$y = x_1 x_2$$

$$\Rightarrow \log y = \log x_1 + \log x_2$$

$$\Rightarrow \frac{1}{y} \frac{\partial y}{\partial x_i} = \frac{1}{x_i}$$

$$\Rightarrow \frac{\partial y}{\partial x_i} = \frac{y}{x_i}$$

\Rightarrow

$$\left| \frac{\Delta y}{y} \right| \leq \sum_{i=1}^2 \left| \frac{\Delta x_i}{x_i} \right|$$

$$\mathcal{E}_y^r \leq \sum_{i=1}^2 \mathcal{E}_i^r$$

Relative Errors Add for Multiplication

Another example, more general case:

$$y = x_1^{m_1} x_2^{m_2} \cdots x_n^{m_n}$$

$$\mathcal{E}_y^r \leq \sum_{i=1}^n |m_i| \mathcal{E}_i^r$$



Statistical Error Propagation Estimate: Error Expectations and the “Standard Error”

Example: Addition

$$y = x_1 + x_2 + \cdots + x_n$$

a. Chopping

$$\Delta x_i = \bar{x}_i - x_i \leq 0$$

Error Expectation:

on average, half the interval is chopped, hence

$$E(\Delta x_i) = -b^{-t}/2$$

$$E(\Delta y) = -nb^{-t}/2$$

b. Rounding

$$E(\Delta x_i) = 0$$

$$E(\Delta y) = nE(\Delta x_i) = 0$$



Statistical Error Propagation Estimate: Error Expectations and the “Standard Error”

$$y = y(x_1, x_2, x_3, \dots, x_n)$$

Standard Error

(assuming indep. Gaussian errors on x_i ,
compute error standard deviation on y)

$$E(\Delta_s y) \simeq \sqrt{\sum_{i=1}^n \left(\frac{\partial y}{\partial x_i}\right)^2 \epsilon_i^2}$$

Example: Addition

$$y = x_1 + x_2 + \dots + x_n$$

$$E(\Delta_s y) \simeq \sqrt{\sum_{i=1}^n \epsilon_i^2} = \sqrt{n}\epsilon$$

Standard Error is a more accurate measure of expected errors:
It is likely that the error will be of that magnitude.

However, the general error propagation formula is an upper bound
(if higher order terms can be neglected)



Note on Error Propagation: Error Cancellation (occurs when errors of the same sign are subtracted)

Example: Consider this function of one variable, to be evaluated at $x = 100$

$$y = f(x) = \sqrt{x^2 + 1} - x + 200; \quad x = \underline{100 \pm 4}$$

$$\frac{\partial z_1}{\partial x} = \frac{x}{\sqrt{x^2 + 1}}, \quad \frac{\partial z_2}{\partial x} = -1$$

Define: $\left\{ \begin{array}{l} z_1 = \sqrt{x^2 + 1} \quad \epsilon_1 = 4 \\ z_2 = 200 - x \quad \epsilon_2 = 4 \\ y = z_1 + z_2 \end{array} \right.$

$$\Delta y \simeq \frac{dz_1}{dx} \Delta x + \frac{dz_2}{dx} \Delta x$$

For 2 terms (z_1 and z_2):
 Gen. error (max): $E(\Delta y) = 8$
 Stand. Error : $E(\Delta_{s y}) = 4\sqrt{2} = 5.6$

Direct error
(single y term):

$$= \left(\frac{x}{\sqrt{x^2 + 1}} - 1 \right) \Delta x \simeq_{x \gg 1} \frac{-1}{2x^2} \Delta x$$

Error cancellation

$$x = 100 \pm 4 \Rightarrow |\Delta y| \leq 2 \cdot 10^{-4} \leq \underbrace{0.5 \cdot 10^{-3}}$$

$$y = 200.005 \pm 0.5 \cdot 10^{-3}$$

error if 4 digits



The Condition Number

- The *condition* of a mathematical problem relates to its sensitivity to changes in its input values
- A computation is *numerically unstable* if the uncertainty of the input values are magnified by the numerical method
- Considering x and $f(x)$, the *condition number* is the ratio of the relative error in $f(x)$ to that in x .
- Using first-order Taylor series $f(x) = f(\bar{x}) + f'(\bar{x})(x - \bar{x})$

- Relative error in $f(x)$: $\frac{f(x) - f(\bar{x})}{f(\bar{x})} \cong \frac{f'(\bar{x})(x - \bar{x})}{f(\bar{x})}$

- Relative error in x : $\frac{(x - \bar{x})}{\bar{x}}$

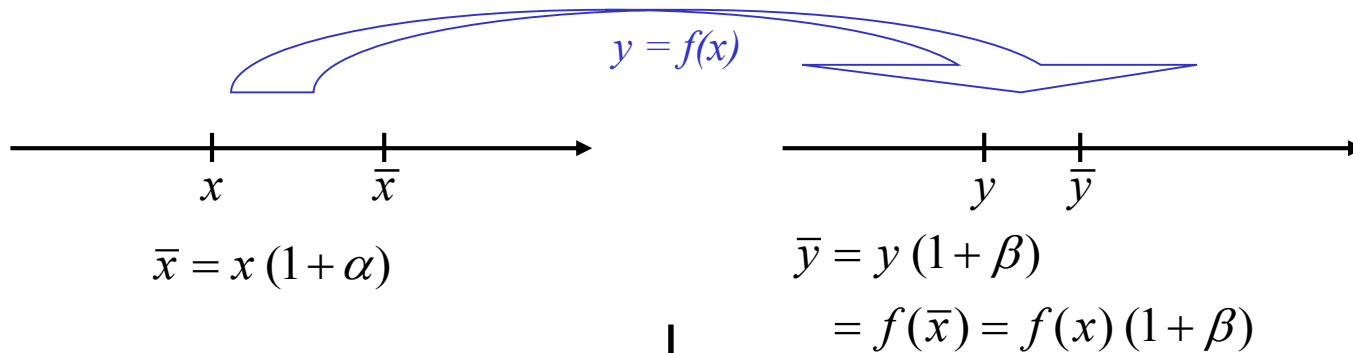
- Condition Nb = Ratio of relative errors:

$$K_p = \left| \frac{\bar{x} f'(\bar{x})}{f(\bar{x})} \right|$$



Error Propagation

Condition Number: another derivation and an example



Problem Condition Number

$$\begin{aligned}
 K_P &= \frac{|\beta|}{|\alpha|} \\
 &= \left| \frac{f(\bar{x}) - f(x)}{f(x)} \right| / \left| \frac{\bar{x} - x}{x} \right| \\
 &= \left| \frac{f(\bar{x}) - f(x)}{\bar{x} - x} \right| \times \left| \frac{x}{f(x)} \right| \\
 &\simeq \left| x \frac{f'(x)}{f(x)} \right|
 \end{aligned}$$

With first-order Taylor series \Rightarrow

$K_P \gg 1$
Problem ill-conditioned

Error cancellation example:

$$y = f(x) = \sqrt{x^2 + 1} - x + 200; \quad x = 100 \pm 4$$

$$K_P = \left| 100 \frac{-0.5 \cdot 10^{-4}}{200.005} \right| = 0.2 \cdot 10^{-4}$$

\Rightarrow Well-conditioned problem



Error Propagation

Problem vs. Algorithm Condition Numbers

a. Problem Condition Number

$$y = \sqrt{x^2 + 1} - x$$

$$x = 100 \Rightarrow y = 0.5 \cdot 10^{-2}$$

$$y' = \frac{x}{\sqrt{x^2 + 1}} - 1 \simeq -\frac{1}{2x^2} = -0.5 \cdot 10^{-4}$$

$$K_P = \left| \frac{100 \cdot \frac{-0.5 \cdot 10^{-4}}{0.5 \cdot 10^{-2}}}{0.5 \cdot 10^{-2}} \right| = 1.0$$

K_A is the algorithm condition number, which may be much larger than K_P due to digitized number representation.

Solution:

- Higher precision
- Rewrite algorithm

b. Algorithm Condition Nb. (4 significant digits, rounding)

$$\bar{y} = \sqrt{0.1 \cdot 10^5 + 1} - 0.1 \cdot 10^3$$

$$= \sqrt{(0.1000 + 0.00001) \cdot 10^5} - 0.1 \cdot 10^3 = 0$$

(base 10)

$$|\beta| = \left| \frac{\bar{y} - y}{y} \right| = \frac{0.5 \cdot 10^{-2}}{0.5 \cdot 10^{-2}} = 1$$

$$|\alpha| = \left| \frac{\bar{x} - x}{x} \right| \leq \frac{1}{2} 10^{1-t} \simeq \frac{1}{2} 10^{-3}$$

Algorithm Condition Number

$$K_A = \frac{|\beta|}{|\alpha|} \simeq 2000$$

Well-conditioned Algorithm

$$y = \frac{1}{\sqrt{x^2 + 1} + x}$$

$$\bar{y} = \frac{1}{0.1 \cdot 10^3 + 0.1 \cdot 10^3} = 0.5 \cdot 10^{-2}$$

$$|\beta| \simeq 0 \Rightarrow K_A \simeq 0 \ll 1$$



Roots of Nonlinear Equations

- Finding roots of equations ($x \mid f(x) = 0$) ubiquitous in engineering problems, including numerical fluid dynamics applications
 - In general: finding roots of functions (implicit in unknown parameters/roots)
 - Fluid Application Examples
 - Implicit integration schemes
 - Stationary points in fluid equations, stability characterizations, etc
 - Fluid dynamical system properties
 - Fluid engineering system design
- Most realistic root finding problems require numerical methods
- Two types of root problems/methods:
 1. Determine the real roots of algebraic/transcendental equations (these methods usually need some sort of prior knowledge on a root location)
 2. Determine all real and complex roots of polynomials (all roots at once)



Roots of Nonlinear Equations

- **Bracketing Methods**

- Bracket root, systematically reduce width of bracket, track error for convergence
- Methods: Bisection and False-Position (Regula Falsi)

- **“Open” Methods**

- Systematic “Trial and Error” schemes, don’t require a bracket (can start from a single value)
- Computationally efficient, but don’t always converge
- Methods: Open-point Iteration (General method or Picard Iteration), Newton-Raphson, Secant Method
- Extension of Newton-Raphson to systems of nonlinear equations

- **Roots of Polynomials (all real/complex roots)**

- Open methods (applications of the above for complex numbers)
- Special Methods (e.g. Muller’s and Bairstow’s methods)



Roots of Nonlinear Equations: Bracketing Methods

$$f(x) = 0$$

Example – Square root

$$x^2 - a = 0 \Rightarrow x = \sqrt{a}$$

Heron's Principle

$$x > 0$$

$$x^2 - a = 0 \Leftrightarrow x = \frac{a}{x}$$

Guess root

$$\text{If } x_0 > \sqrt{a} \Leftrightarrow \frac{a}{x_0} < \sqrt{a}$$

$$\text{If } x_0 < \sqrt{a} \Leftrightarrow \frac{a}{x_0} > \sqrt{a}$$

Mean is better guess

$$x_1 = (x_0 + \frac{a}{x_0}) / 2$$

Iteration Formula

$$x_k = (x_{k-1} + \frac{a}{x_{k-1}}) / 2$$

```

a=2;
n=6;
g=2;
% Number of Digits
dig=5;
sq(1)=g;
for i=2:n
    sq(i)= 0.5*radd(sq(i-1),a/sq(i-1),dig);
end
'      i      value      '
[ [1:n]' sq']
hold off
plot([0 n],[sqrt(a) sqrt(a)],'b')
hold on
plot(sq,'r')
plot(a./sq,'r-.')
plot((sq-sqrt(a))/sqrt(a),'g')
legend('sqrt','xn','s/xn','Relative Err')
grid on

```

heron.m

i	value
1.0000	2.0000
2.0000	1.5000
3.0000	1.4167
4.0000	1.4143
5.0000	1.4143
6.0000	1.4143



Bracketing Methods: Graphical Approach

A function typically changes sign at the vicinity of the root \Rightarrow “bracket” the root

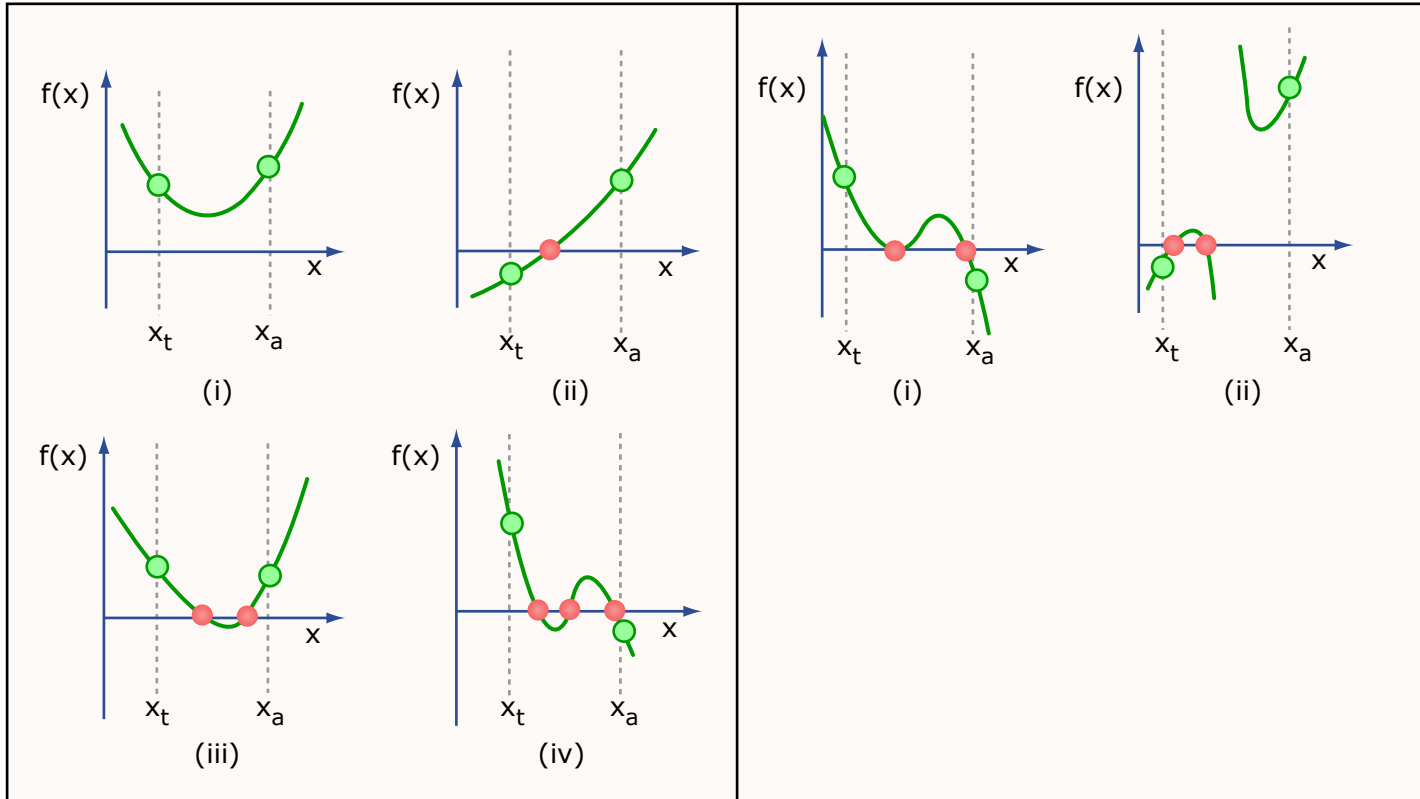


Image by MIT OpenCourseWare.

These graphs show general ways in which roots may occur on a bounded interval. In (i) and (ii), if the function is positive at both boundaries, there will be either an even number of roots or no roots; in the other two graphs, if the function has opposite signs at the boundary points, there will be an odd number of roots within the interval.

These graphs show some exceptions to the general cases illustrated on the left. In (i), a multiple root occurs where the function is tangential to the x axis. There are an even number of intersections for the interval. In (ii) is depicted a discontinuous function with an even number of roots and end points of opposite signs. Finding the roots in such cases requires special strategies.



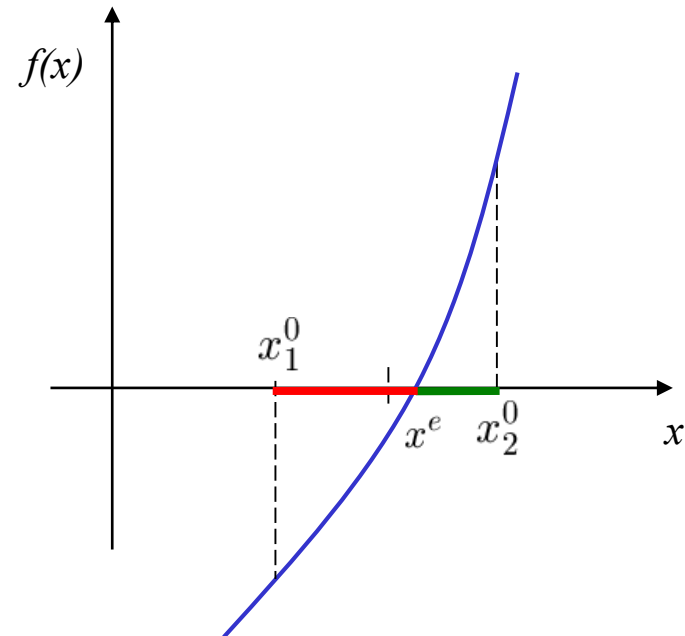
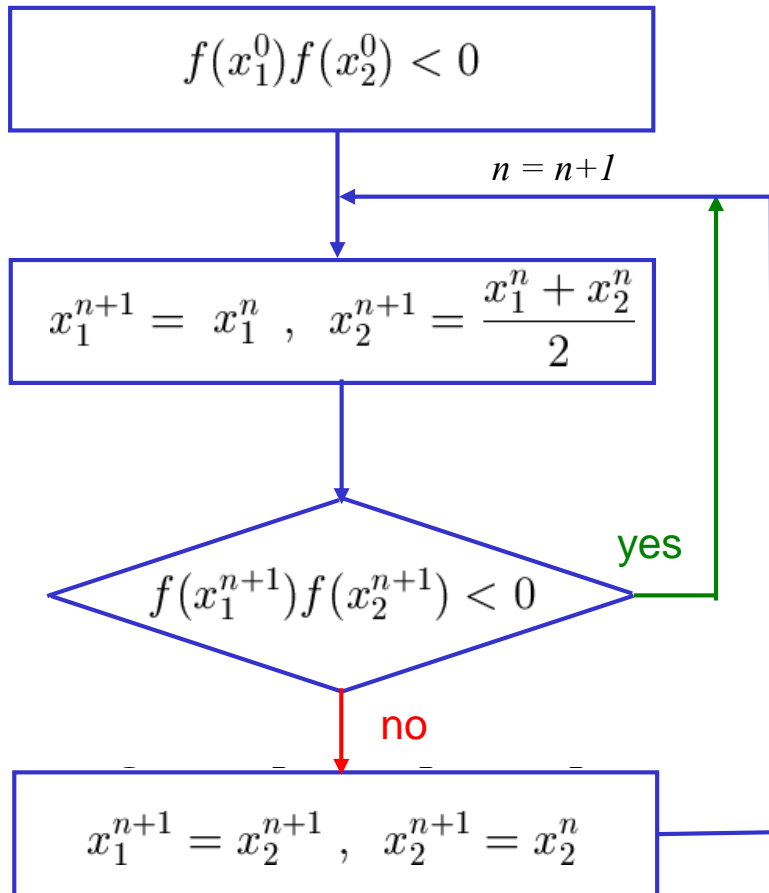
Roots of Nonlinear Equations: Bracketing Methods

Bisection

Incremental search method defined as:

“Keep dividing in two the interval within which at least one root lies”

Algorithm

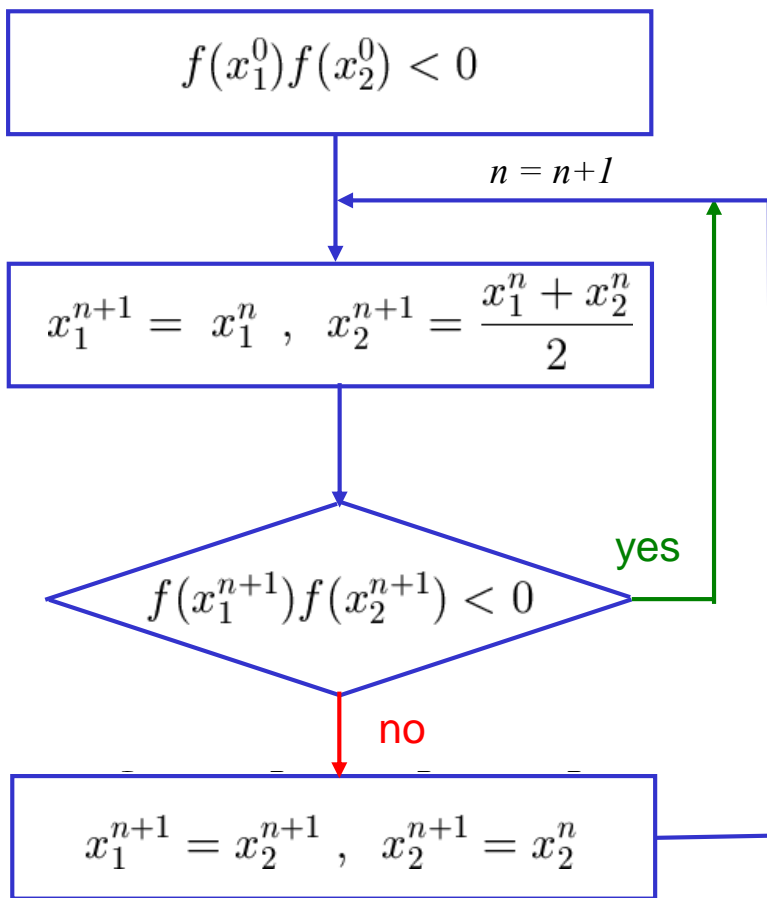




Roots of Nonlinear Equations: Bracketing Methods

Bisection

Algorithm

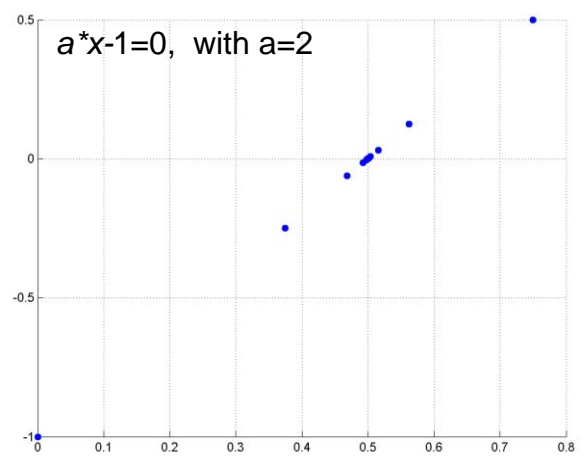


```

% Root finding by bi-section
f=inline(' a*x -1','x','a');
a=2
figure(1); clf; hold on
x=[0 1.5]; eps=1e-3;
err=max(abs(x(1)-x(2)),abs(f(x(1),a)-f(x(2),a)));
while (err>eps & f(x(1),a)*f(x(2),a) <= 0)
    xo=x; x=[xo(1) 0.5*(xo(1)+xo(2))];
    if ( f(x(1),a)*f(x(2),a) > 0 )
        x=[0.5*(xo(1)+xo(2)) xo(2) ]
    end
    x
    err=max(abs(x(1)-x(2)),abs(f(x(1),a)-f(x(2),a)));
    b=plot(x,f(x,a),'.b'); set(b,'MarkerSize',20);
    grid on;
end
  
```

bisection.m

Stop criterion above based on $\Delta x \sim 0$ and $\Delta f(x) \sim 0$





Roots of Nonlinear Equations: Bracketing Methods

Bisection

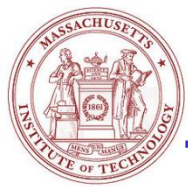
- Stop criteria:
 - absolute added values go to zero: $\Delta x \sim 0$ and $\Delta f(x) \sim 0$
 - better to stop when relative estimated error, i.e. relative added value, is negligible:

$$|\mathcal{E}_a| = \left| \frac{\hat{x}_r^n - \hat{x}_r^{n-1}}{\hat{x}_r^n} \right| \leq \mathcal{E}_s$$

- Each successive iteration halves the maximum error, a general formula thus relates the error and the number of iterations:

$$n = \log_2 \left(\frac{\Delta x^0}{E_{a,d}} \right)$$

where Δx^0 is an initial error and $E_{a,d}$ the desired error



Another Bracketing Method: The False Position Method (Regula Falsi)

- Bisection is a brute force scheme
 - Somewhat inefficient approach
 - Doesn't account for magnitudes of $f(x_L)$ and $f(x_U)$

- False Position

$$x_r = x_U - \frac{f(x_U)(x_L - x_U)}{f(x_L) - f(x_U)}$$

$$f(x_r) = f(x_U) + \frac{f(x_U) - f(x_L)}{x_U - x_L}(x_r - x_U) \cong 0$$

$$\frac{f(x_L)}{(x_r - x_L)} = \frac{f(x_U)}{(x_r - x_U)}$$

$$x_r = x_U - \frac{f(x_U)(x_L - x_U)}{f(x_L) - f(x_U)}$$

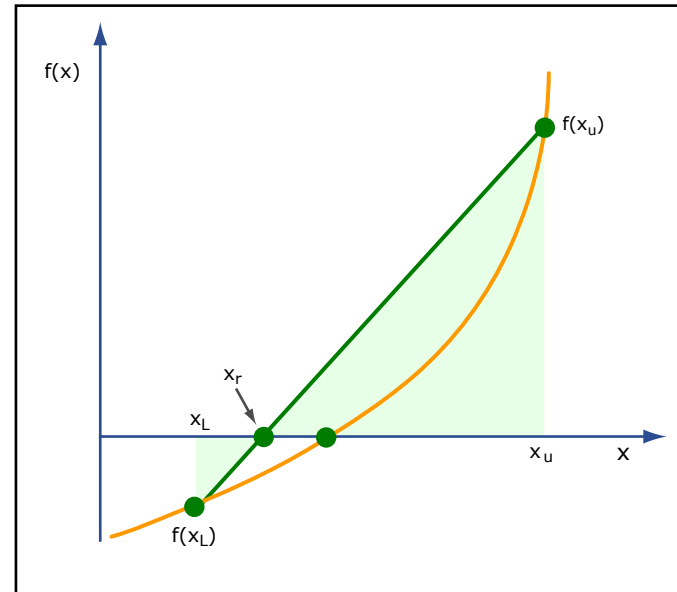


Image by MIT OpenCourseWare.

A graphical depiction of the false position method, Regula Falsi. Shaded in green are the triangles used to derive the method.



Comments on Bracketing Methods (Bisection and False Position)

- Bisection and False Position:
 - Convergent methods, but can be slow and thus can require lots of function evaluations, each of which can be very expensive (full code).
- Error of False Position can decrease much faster than that of Bisection, but if the function is far from a straight line, False Position will be slow (one of the end point remains fixed)
- Remedy: “Modified False Position”
 - e.g. if end point remains fixed for a few iterations, reduce end-point function value at end point, e.g. divide interval by 2
- One can not really generalize behaviors for Root Finding methods (results are function dependent)
- Always substitute estimate of root x_r at the end to check how close estimate is to $f(x_r)=0$



Roots of Nonlinear Eqs.: Open Methods

Fixed Point Iteration (General Method or Picard Iteration)

Non-linear Equation

$$f(x) = 0$$

Goal: Converging series

$$x_0, x_1, \dots, x_n \rightarrow x^e, \quad n \rightarrow \infty$$

Rewrite Problem

$$f(x) = 0 \Leftrightarrow g(x^e) = x^e$$

Example

$$g(x) = x + c \cdot f(x)$$

Iteration

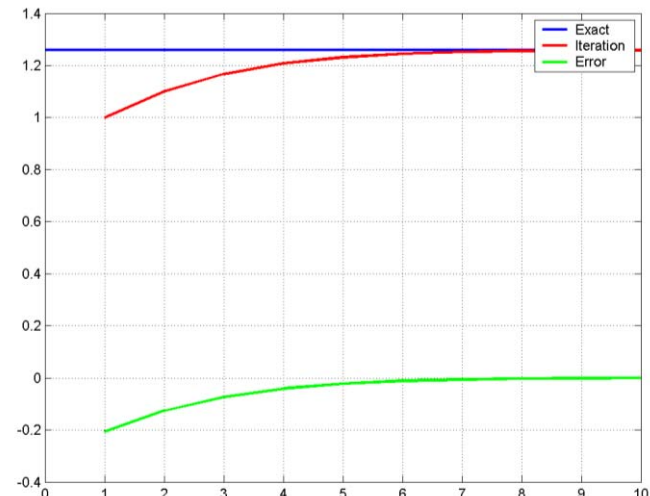
$$x_n = g(x_{n-1})$$

Example: Cube root

```

% f(x) = x^3 - a = 0
% g(x) = x + C*(x^3 - a)
a=2;
n=10;
g=1.0;
C=-0.1;
sq(1)=g;
for i=2:n
    sq(i)= sq(i-1) + C*(sq(i-1)^3 -a);
end
hold off
plot([0 n],[a^(1./3.) a^(1./3.)], 'b')
hold on
plot(sq, 'r')
plot( (sq-a^(1./3.))/(a^(1./3.)), 'g')
grid on
    
```

cube.m





Roots of Nonlinear Equations: Bracketing vs. Open Methods in Graphics

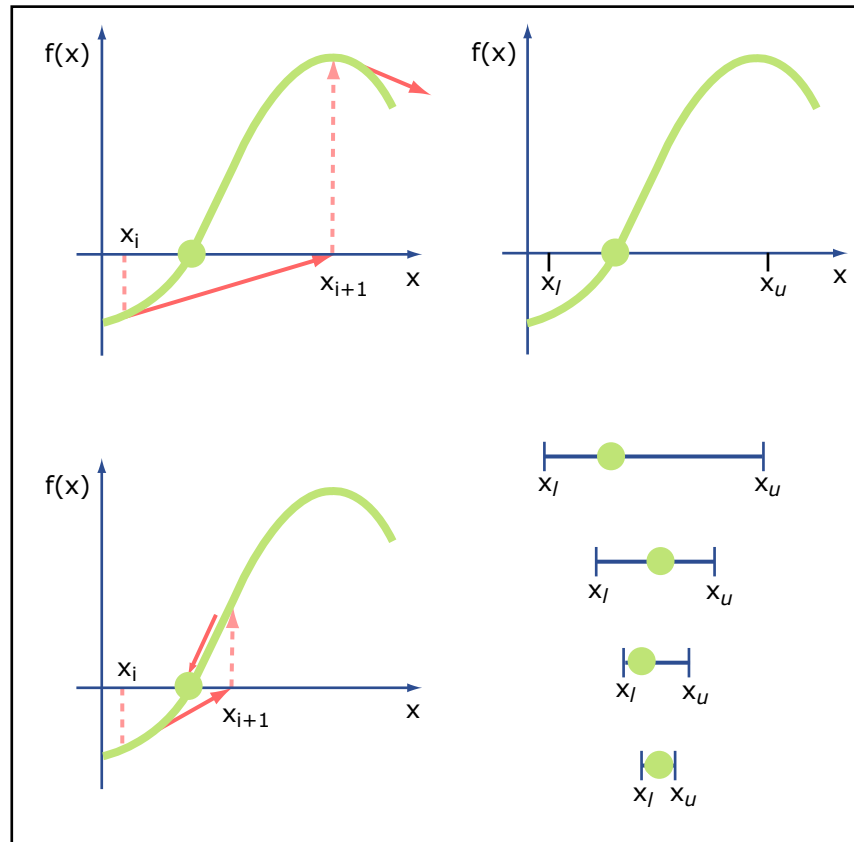


Image by MIT OpenCourseWare.

A graphical depiction of the fundamental differences between the bracketing method (shown on the right) and the open methods (shown on the left). On the right, the root is constrained within the interval; on the left, a formula is used to project iteratively between x_i and x_{i+1} and can diverge or converge rapidly.



Open Methods (Fixed Point Iteration) Stop-criteria

“Unrealistic” stop-criteria: continue while

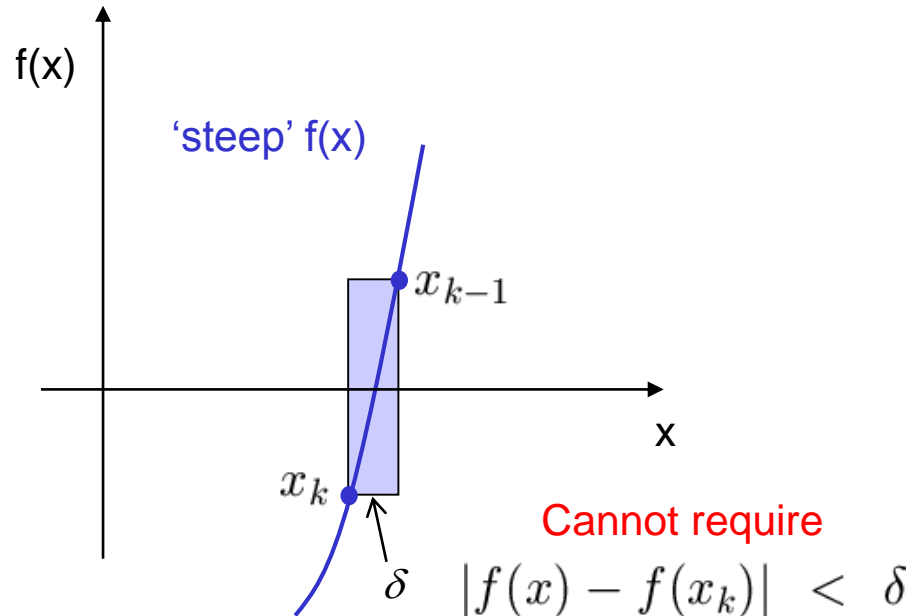
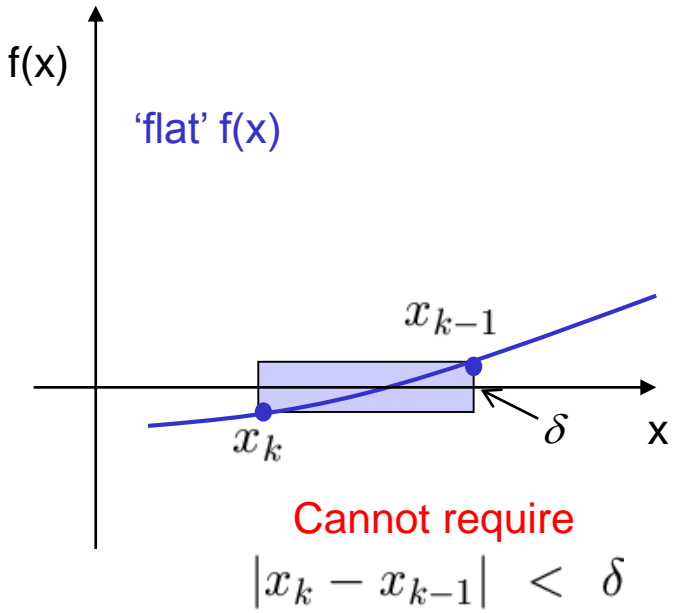
$$x_{k+1} \neq x_k$$

Realistic stop-criteria

$$|x_k - x_{k-1}| < \delta \leftarrow \text{Machine Epsilon}$$

$$|f(x) - f(x_k)| < \delta$$

Use an “or” combination of the two criteria



MIT OpenCourseWare
<http://ocw.mit.edu>

2.29 Numerical Fluid Mechanics

Spring 2015

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.