ESO 208A

Computational Method in Engineering

Lecture 09

Office Hours

- Monday – 5:00 to 7:00 pm
- Friday – 12:00 to 1:00 pm

Müller Method

Müller's method obtains a root estimate by projecting a parabola to the *x* axis through three function values.

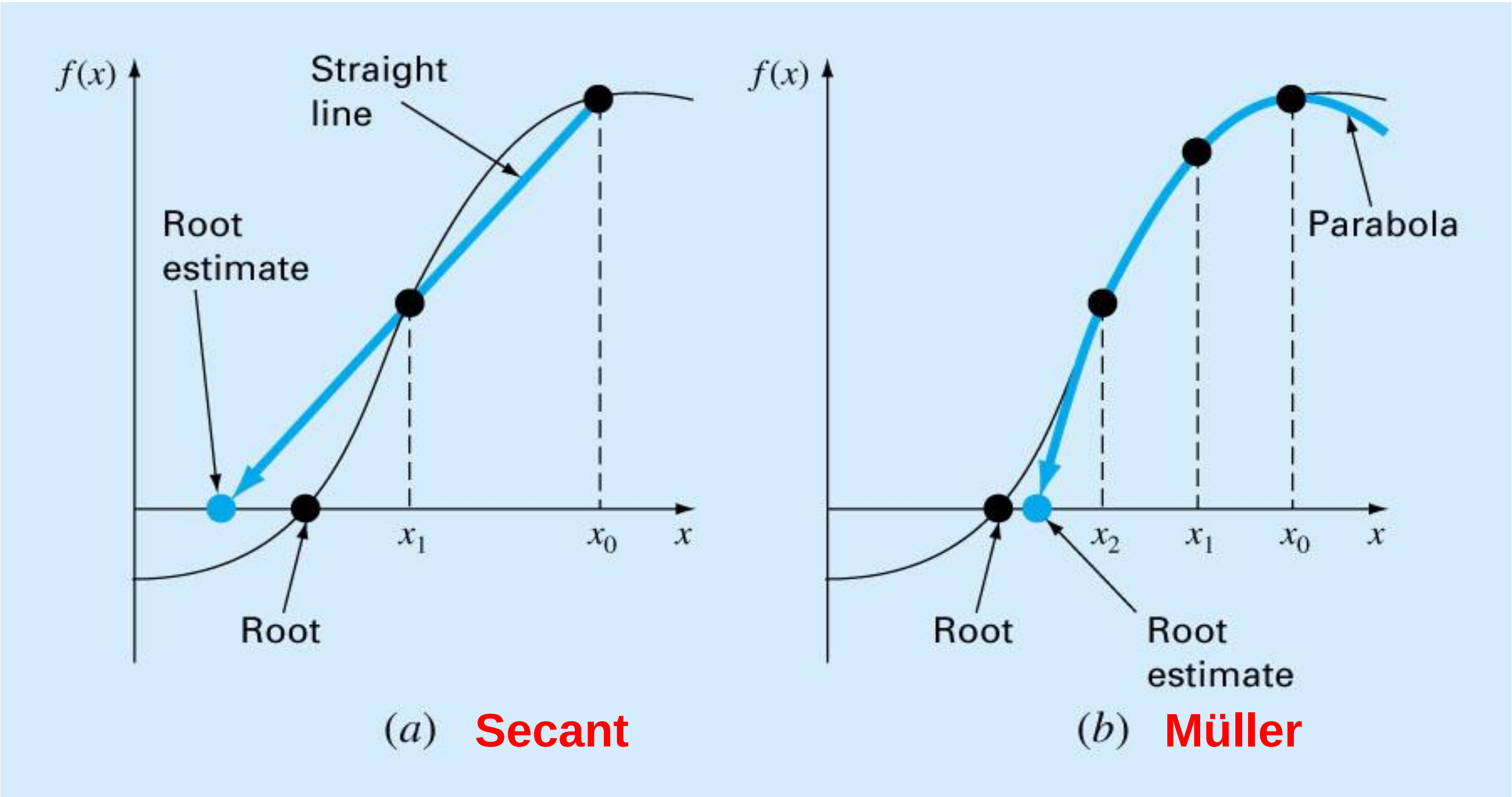

Figure 7.3 of C&C

Müller Method

1. Write the equation of a parabola in a convenient form:

$$f_2(x) = a(x - x_2)^2 + b(x - x_2) + c$$

2. The parabola should intersect the three points [x_o, $f(x_o)$], [x_1, $f(x_1)$], [x_2, $f(x_2)$].

$$f(x_o) = a(x_o - x_2)^2 + b(x_o - x_2) + c$$

$$f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c$$

$$f(x_2) = a(x_2 - x_2)^2 + b(x_2 - x_2) + c$$

Müller Method

3. The three equations can be solved to estimate *a*, *b*, and *c*

Define

$$h_o = x_1 - x_o \qquad h_1 = x_2 - x_1$$

$$\delta_o = \frac{f(x_1) - f(x_o)}{x_1 - x_o} \qquad \delta_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

then,

$$a = \frac{\delta_1 - \delta_o}{h_1 + h_o} \qquad b = ah_1 + \delta_1 \qquad c = f(x_2)$$

Müller Method

4. Roots can be found by applying quadratic formula:

$$x_3 - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

5. $\pm$term yields two roots; the sign is chosen to agree with b. This will result in a largest denominator, and will give root estimate that is closest to x_2.

Müller Method

6. Once x_3 is determined, the process is repeated by employing a sequential approach just like in secant method, x_1, x_2, and x_3 to replace x_0, x_1, and x_2.

Bairstow Method

1. Bairstow's method is an iterative approach loosely related to both Müller and Newton Raphson methods

2. It is based on dividing the given polynomial by a quadratic polynomial x^2-rx-s:

$$f_n(x) = a_o + a_1 x + a_2 x^2 + \ldots + a_n x^n$$
$$= \left(x^2 - rx - s\right) f_{n-2}(x) + R$$

where

$$f_{n-2}(x) = b_2 + b_3 x + \ldots + b_{n-1} x^{n-3} + b_n x^{n-2}$$
$$R = b_1(x - r) + b_o$$

Bairstow Method

3. The coefficients *b*'s are obtained very easily by using recursive relation

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + rb_n$$

$$b_i = a_i + rb_{i+1} + sb_{i+2} \qquad i = n - 2 \text{ to } 0$$

4. Using Newton Raphson approach, *r* and *s* are adjusted so as to make both b_o *and* b_1 approach zero

$$b_1 = a_1 + rb_2 + sb_3 \Rightarrow u(r,s)$$

$$b_0 = a_0 + rb_1 + sb_2 \Rightarrow v(r,s)$$

Bairstow Method

5. Obtain corrections in r and s by Newton-Raphson method

Changes Δs and Δr needed to improve guesses will be estimated by

$$\frac{\partial b_1}{\partial r}\Delta r+\frac{\partial b_1}{\partial s}\Delta s=-b_1$$

$$\frac{\partial b_o}{\partial r}\Delta r+\frac{\partial b_o}{\partial s}\Delta s=-b_o$$

Bairstow Method

6. Bairstow (1920) showed that the partial derivatives of b_1 and b_2 are obtained by the recursive relation

$$c_n = b_n$$

$$c_{n-1} = b_{n-1} + rc_n$$

$$c_i = b_i + rc_{i+1} + sc_{i+2} \qquad i = n-2 \; to \; 2$$

where

$$\frac{\partial b_o}{\partial r} = c_1 \qquad \frac{\partial b_o}{\partial s} = \frac{\partial b_1}{\partial r} = c_2 \qquad \frac{\partial b_1}{\partial s} = c_3$$

7. Iterate the steps untill ($\Delta r/r$) and ($\Delta s/s$) drops below a specified threshold

Bairstow Method

8. The roots quadratic polynomial $x^2\text{-}rx\text{-}s$ are obtained as

$$x = \frac{r \pm \sqrt{r^2 + 4s}}{2}$$

9. At this point three possibilities exist:

   1. *The quotient is a third-order polynomial or greater.* The previous values of *r* and *s* serve as initial guesses and Bairstow's method is applied to the quotient to evaluate new r and s values.

   2. *The quotient is quadratic.* The remaining two roots are evaluated directly, using the above eqn.

   3. *The quotient is a 1st order polynomial*. The remaining single root can be evaluated simply as $x=-s/r$.

Revision of Solution of Non-linear Equations

1. Graphical Method – Provide insights but tedious/subjective

2. Bracketing methods

   1. Bisection method

   2. False position method

   3. Modified false position method

   Guaranteed convergence
   Linear or better convergence

3. Open methods

   1. Fixed-point iteration

   2. Newton-Raphson

   3. Secant & Modified Secant

   May diverge
   FP - linear convergence
   NR – quadratic convergence
   Secant – between linear & quadratic

   NR – problems near zero gradient

Revision of Solution of Non-linear Equations

Hybrid Methods

1. *Dekker method*
2. *Brent method*

Combination
- Bracketing method at the beginning
- Open method near convergence

Multiple roots

1. *Bracketing method* – Only for odd number of roots
2. *Newton-Raphson* - Linear convergence
3. *Modified Newton Raphson* – Quadratic convergence
   a. Known multiplicity
   b. Derivative function

Revision of Solution of Non-linear Equations

Roots of polynomials

1. Evaluation of polynomials

2. Division of polynomials

3. Deflation of polynomials

4. Effective degree of polynomials

Method of finding roots

1. Müller method

2. Bairstow method

Real and complex rooots

Revision of Solution of Non-linear Equations

1. Except for rare cases, computers will provide approximate solution.
2. No method is “universally” better than others.
3. Domain knowledge should guide the selection of algorithm and guess value(s).

Comparison of different algorithms

Method	Type	Guesses	Convergence	Stability	Programming	Comments
Direct	Analytical	—	—	—		
Graphical	Visual	—	—	—	—	Imprecise
Bisection	Bracketing	2	Slow	Always	Easy	
False-position	Bracketing	2	Slow/medium	Always	Easy	
Modified FP	Bracketing	2	Medium	Always	Easy	
Fixed-point iteration	Open	1	Slow	Possibly divergent	Easy	
Newton-Raphson	Open	1	Fast	Possibly divergent	Easy	Requires evaluation of $f'(x)$
Modified Newton-Raphson	Open	1	Fast (multiple), medium (single)	Possibly divergent	Easy	Requires evaluation of $f'(x)$ and $f''(x)$
Secant	Open	2	Medium/fast	Possibly divergent	Easy	Initial guesses do not have to bracket the root
Modified secant	Open	1	Medium/fast	Possibly divergent	Easy	
Brent	Hybrid	1 or 2	Medium	Always (for 2 guesses)	Moderate	Robust
Müller	Polynomials	2	Medium/fast	Possibly divergent	Moderate	
Bairstow	Polynomials	2	Fast	Possibly divergent	Moderate	

Announcement

1. The due date for the first computer assignment will be Sunday, August 21, 11:59 pm.
2. No option of late or second submission.
3. The computer programs should be written so that they can be easily evaluated on independent test data.
    a. GUI
    b. Text file
    c. Screen input