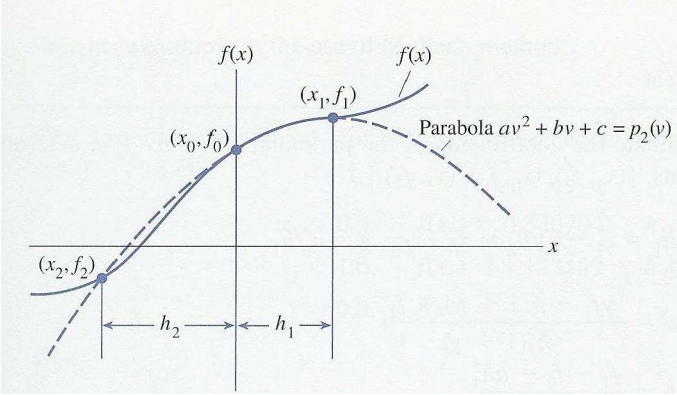


Figure 1: Parabola $a\nu^2 + b\nu + c = p_2(\nu)$

1 Muller's Method

- Most of the root-finding methods that we have considered so far have approximated the function in the neighborhood of the root by a straight line.
- *Muller's method* is based on approximating the function in the neighborhood of the root by a quadratic polynomial.
- A second-degree polynomial is made to fit three points near a root, at x_0, x_1, x_2 with x_0 between x_1, and x_2.
- The proper zero of this quadratic, using the quadratic formula, is used as the improved estimate of the root.
- A quadratic equation that fits through three points in the vicinity of a root, in the form $a\nu^2 + b\nu + c$. (See Fig. 1)
- Transform axes to pass through the middle point, by letting $\nu = x - x_0$. Let $h_1 = x_1 - x_0$ and $h_2 = x_0 - x_2$. We evaluate the coefficients by evaluating $p_2(\nu)$ at the three points:

  $\nu = 0: \ a(0)^2 + b(0) + c = f_0$
  $\nu = h_1: \ ah_1^2 + bh_1 + c = f_1$
  $\nu = -h_2: \ ah_2^2 - bh_2 + c = f_2$

- From the first equation, $c = f_0$. Letting $h_2/h_1 = \gamma$, we can solve the

other two equations for a, and b.

$$a = \frac{\gamma f_1 - f_0(1+\gamma) + f_2}{\gamma h_1^2(1+\gamma)},\ b = \frac{f_1 - f_0 - a h_1^2}{h_1}$$

After computing a, b, and c, we solve for the root of $a\nu^2 + b\nu + c$ by the quadratic formula

$$\nu_{1,2} = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}},\ \nu = x - x_0,\ root = x_0 - \frac{2c}{b \pm \sqrt{b^2 - 4ac}}$$

An algorithm for Muller's method :

Given the points x_2, x_0, x_1 in increasing value,
Evaluate the corresponding function values: f_2, f_0, f_1.
Repeat
(Evaluate the coefficients of the parabola, $ax^2 + bx + c$, determined by the three points.
$(x_2, f_2), (x_0, f_0), (x_l, f_1)$.)
Set $h_l = x_1 - x_0; h_2 = x_0 - x_2; \gamma = h_2/h_1$.
Set $c = f_0$
Set $a = \frac{\gamma f_1 - f_0(1+\gamma) + f_2}{\gamma h_1^2(1+\gamma)}$
Set $b = \frac{f_1 - f_0 - a h_1^2}{h_1}$
(Next, compute the roots of the polynomial.)
Set $root = x_0 - \frac{2c}{b \pm \sqrt{b^2 - 4ac}}$
Choose root, x_r, closest to x_0 by making the denominator as large as possible; i.e. if
$b > 0$, choose plus; otherwise, choose minus.
If $x_r > x_0$,
Then rearrange to:x_0, x_1, and the root
Else rearrange to: x_0, x_2, and the root
End If.
(In either case, reset subscripts so that x_0, is in the middle.)
Until $|f(x_r)| < Ftol$

- Muller's method, like Newton's, will find a complex root if given complex starting values. Of course, the computations must use complex arithmetic.

- Experience shows that Muller's method converges at a rate that is similar to that for Newton's method. It does not require the evaluation of derivatives, however, and (after we have obtained the starting values) needs only one function evaluation per iteration.

- See Fig. 2 that an example is given

Find a root between 0 and 1 of the same transcendental function as before: $f(x) = 3x + \sin(x) - e^x$. Let

$$\begin{aligned} x_0 &= 0.5, & f(x_0) &= 0.330704 & h_1 &= 0.5, \\ x_1 &= 1.0, & f(x_1) &= 1.123489 & h_2 &= 0.5, \\ x &= 0.0, & f(x_2) &= -1 & \gamma &= 1.0. \end{aligned}$$

Then

$$a = \frac{(1.0)(1.123189) - 0.330704(2.0) + (-1)}{1.0(0.5)^2(2.0)} = -1.07644,$$

$$b = \frac{1.123189 - 0.330704 - (-1.07644)(0.5)^2}{0.5} = 2.12319,$$

$$c = 0.330704,$$

and

$$\begin{aligned} \text{root} &= 0.5 - \frac{2(0.330704)}{2.12319 + \sqrt{(2.12319)^2 - 4(-1.07644)(0.330704)}}. \\ &= 0.354914. \end{aligned}$$

For the next iteration, we have

$$\begin{aligned} x_0 &= 0.354914, & f(x_0) &= -0.0138066 & h_1 &= 0.145086, \\ x_1 &= 0.5, & f(x_1) &= 0.330704 & h_2 &= 0.354914, \\ x_2 &= 0, & f(x_2) &= -1 & \gamma &= 2.44623. \end{aligned}$$

Then

$$a = \frac{(2.44623)(0.330704) - (-0.0138066)(3.44623) + (-1)}{2.44623(0.145086)^2(3.44623)} = -0.808314,$$

$$b = \frac{0.330704 - (-0.0138066) - (-0.808314)(0.145086)^2}{0.145086} = 2.49180,$$

$$c = -0.0138066,$$

$$\begin{aligned} \text{root} &= 0.354914 - \frac{2(-0.0138066)}{2.49180 + \sqrt{(2.49180)^2 - 4(-0.808314)(-0.0138066)}} \\ &= 0.360465. \end{aligned}$$

After a third iteration, we get 0.3604217 as the value for the root, which is identical to that from Newton's method after three iterations.

Figure 2: An example of the use of Muller's method.

2 Fixed-point Iteration; $x = g(x)$ Method

- we rearrange $f(x)$ into an equivalent form $x = g(x)$, which usually can be done in several ways.
- Observe that if $f(r) = 0$, where r is a root of $f(x)$, it follows that $r = g(r)$.
- Whenever we have $r = g(r)$, r is said to be a *fixed* point for the function g.
- The iterative form:

  $$x_{n+1} = g(x_n) \quad n = 0, 1, 2, 3, \ldots$$

  converges to the fixed point r, a root of $f(x)$.
- Example

  $$f(x) = x^2 - 2x - 3 = 0$$

  Suppose we rearrange to give this equivalent form:

  $$x = g_1(x) = \sqrt{2x + 3}$$

- If we start with $x = 4$ and iterate with the fixed-point algorithm, successive values of x are
  $x_0 = 4$
  $x_1 = \sqrt{11} = 3.31662$
  $x_2 = \sqrt{9.63325} = 3.10375$
  $x_3 = 3.03439$
  $x_4 = 3.01144$
  $x_5 = 3.00381$
  and it appears that the values are converging on the root at $x = 3$.

2.1 Other Rearrangements

- Another rearrangement of $f(x)$ is

  $$x = g_2(x) = \frac{3}{(x - 2)}$$

  Let us start the iterations again with $x_0 = 4$. Successive values then are:

$x_0 = 4$
$x_1 = 1.5$
$x_2 = -6$
$x_3 = -0.375$
$x_4 = -1.263158$
$x_5 = -0.919355$
$x_6 = -1.02762$
$x_7 = -0.990876$
$x_8 = -1.00305$
and it seems that we now converge to the other root, at $x = -1$.

- Consider a third rearrangement:

$$x = g_3(x) = \frac{(x^2 - 3)}{2}$$

starting again with $x_0 = 4$, we get
$x_0 = 4$
$x_1 = 6.5$
$x_2 = 19.625$
$x_3 = 191.070$
and the iterates are obviously diverging.

- The fixed point of $x = g(x)$ is the intersection of the line $y = x$ and the curve $y = g(x)$ plotted against x. Figure 3 shows the three cases.

- Observe that we always get the successive iterates by this construction. Start on the x-axis at the initial x_0, go vertically to the curve, then horizontally to the line $y = x$, then vertically to the curve, and again horizontally to the line.

- Repeat this process until the points on the curve converge to a fixed point or else diverge

Iteration algorithm with the form $x = g(x)$

To determine a root of $f(x) = 0$, given a value x_1 reasonably close to the root
Rearrange the equation to an equivalent form $x = g(x)$
Repeat
Set $x_2 = x_l$
Set $x_l = g(x_1)$
Until $|x_1 - x_2| < tolerance\ value$

The method may converge to a root different from the expected one, or it may diverge. Different rearrangements will converge at different rates.

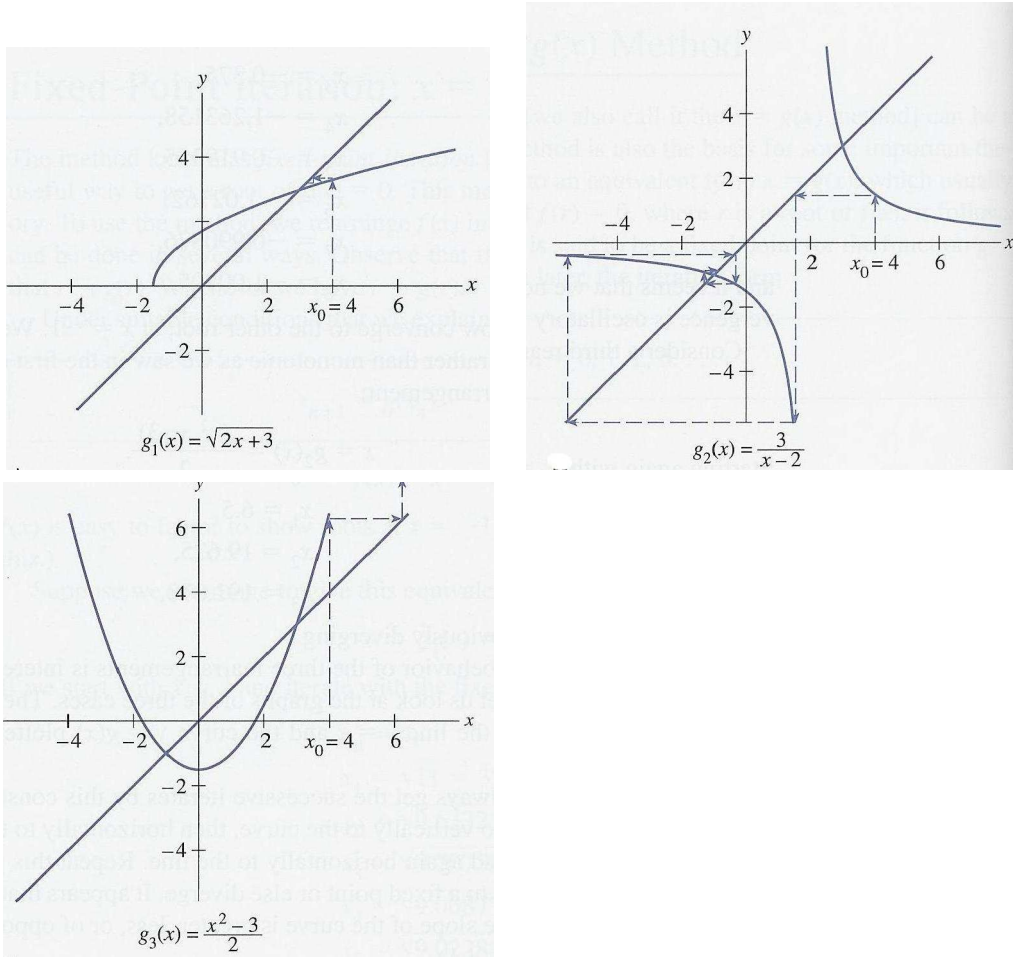


Figure 3: The fixed point of $x = g(x)$ is the intersection of the line $y = x$ and the curve $y = g(x)$ plotted against x.

2.2 Order of Convergence

- The fixed-point method converges at a linear rate; it is said to be *linearly convergent*, meaning that the error at each successive iteration is a constant fraction of the previous error. (Actually, this is true only as the errors approach zero.)

- If we tabulate the errors after each step in getting the roots of the polynomial and its ratio to the previous error, we find that the magnitudes of the ratios to be leveling out at 0.3333. (See Fig. 1)

Table 1: The order of convergence for the iteration algorithm with the different forms of $x = g(x)$.

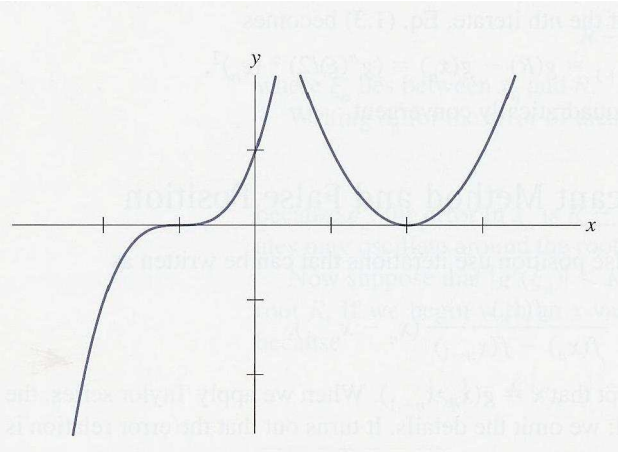
	If $g(x) = \sqrt{2x + 3}$		If $g(x) = 3/(x - 2)$	
Iteration	**Error**	**Ratio**	**Error**	**Ratio**
1	0.31662	0.31662	2.50000	0.50000
2	0.10375	0.32767	−5.00000	−2.00000
3	0.03439	0.33143	0.62500	−0.12500
4	0.01144	0.33270	−0.26316	−0.42105
5	0.00381	0.33312	0.08065	−0.30645
6			−0.02762	−0.34254
7			0.00912	−0.33029
8			−0.00305	−0.33435

3 Multiple Roots

- A function can have more than one root of the same value. See Fig. 4left.
- The methods we have described do not work well for multiple roots. For example, Newton's method is only linearly convergent at a double root. $f(x) = (x - 1)(e^{(x-1)} - 1)$ has a double root at $x = 1$, as seen in Fig. 4right.
- Table 2left gives the errors of successive iterates and the convergence is clearly linear.
- When Newton's method is applied to a triple root, convergence is still linear, as seen in Table 2right. With a triple root, the ratio of errors is larger, about $\frac{2}{3}$, compared to $\frac{1}{2}$ for the double root of Table 2left.

4 Nonlinear Systems

- A pair of equations:
  $x^2 + y^2 = 4$
  $e^x + y = 1$
  Graphically, the solution to this system is represented by the intersections of the circle $x^2 + y^2 = 4$ with the curve $y = 1 - e^x$ (see Fig. 5)



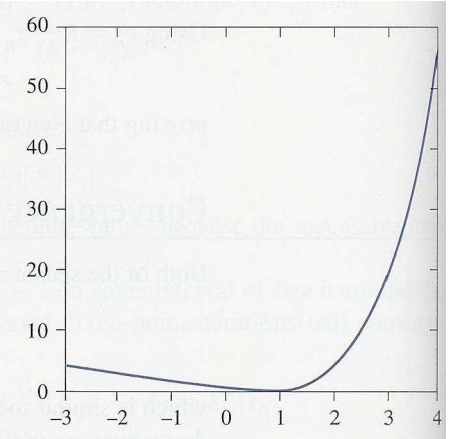


Figure 4: Left: The curve on the left has a triple root at $x = -1$ [the function is $(x+1)^3$]. The curve on the right has a double root at $x = 2$ [the function is $(x-2)^2$].Right: Plot of $(x-1)(e^{(x-1)}-1)$.

Table 2: Right: Errors when finding a double root. Left: Successive errors with Newton's method, for $f(x) = (x+1)^3 = 0$.

Iteration	Error	Ratio
1	0.3679	
2	0.1666	0.453
3	0.0798	0.479
4	0.0391	0.490
5	0.0193	0.494
6	0.0096	0.497
7	0.0048	0.500
8	0.0024	0.500

Iteration	Error	Iteration	Error
0	0.5	6	0.0439
1	0.3333	7	0.0293
2	0.2222	8	0.0195
3	0.1482	9	0.0130
4	0.0988	10	0.00867
5	0.0658		

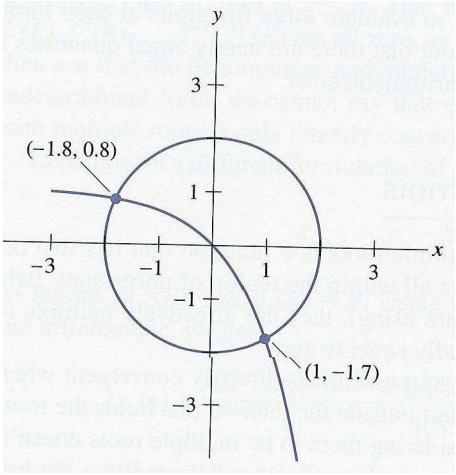


Figure 5: A pair of equations.

- Newton's method can be applied to systems as well as to a single non-linear equation. We begin with the forms

$$f(x, y) = O, g(x, y) = O$$

- Let $x = r, y = s$ be a root, and expand both functions as a Taylor series about the point (x_i, y_i) in terms of $(r - x_i), (s - y_i)$, where (x_i, y_i) is a point near the root:

$$f(r, s) = 0 = f(x_i, y_i) + f_x(x_i, y_i)(r - x_i) + f_y(x_i, y_i)(s - y_i) + \dots$$
$$g(r, s) = 0 = g(x_i, y_i) + g_x(x_i, y_i)(r - x_i) + g_y(x_i, y_i)(s - y_i) + \dots$$

Truncating both series gives

$$0 = f(x_i, y_i) + f_x(x_i, y_i)(r - x_i) + f_y(x_i, y_i)(s - y_i)$$
$$0 = g(x_i, y_i) + g_x(x_i, y_i)(r - x_i) + g_y(x_i, y_i)(s - y_i)$$

which we can rewrite as

$$f_x(x_i, y_i)\Delta x_i + f_y(x_i, y_i)\Delta y_i = -f(x_i, y_i)$$
$$g_x(x_i, y_i)\Delta x_i + g_y(x_i, y_i)\Delta y_i = -g(x_i, y_i)$$

where Δx_i and Δy_i are used as increments to x_i and y_i so that $x_{i+1} = x_i + \Delta x_i$ and $y_{i+1} = y_i + \Delta y_i$ are improved estimates of the (x, y) values. We repeat this until both $f(x, y)$ and $g(x, y)$ are close to zero.

- Example:

$$f(x, y) = 4 - x^2 - y^2 = 0$$

$$g(x, y) = 1 - e^x - y = 0$$

The partial derivatives are

$$f_x = -2x, f_y = -2y, g_x = -e^x, g_y = -1$$

Beginning with $x_0 = 1, y_0 = -1.7$, we solve

$$-2\Delta x_0 + 3.4\Delta y_0 = -0.1100$$

$$-2.7183\Delta x_0 - 1.0\Delta y_0 = 0.0183$$

- This gives $\Delta x_0 = 0.0043, \Delta y_0 = -0.0298$, from which $x_1 = 1.0043, y_1 = -1.7298$. These agree with the true value within 2 in the fourth decimal place.

- Repeating the process once more produces $x_2 = 1.004169, y_2 = -1.729637$. The function values at this second iteration are approximately -0.000000l and -0.00000001.

4.1 Solving a System by Iteration

- There is another way to attack a system of nonlinear equations. Consider this pair of equations:

$$e^x - y = 0$$

$$xy - e^x = 0$$

- We know how to solve a single nonlinear equation by fixed-point iterations –we rearrange it to solve for the variable in a way that successive computations may reach a solution.

$$x = ln(y)$$

$$y = e^x/x$$

To start, we guess at a value for y, say, $y = 2$. See Table 3. which are precisely the correct results.

y-value	x-value
2	0.69315
2.88539	1.05966
2.72294	1.00171
2.71829	1.00000
2.71828	1.00000

Table 3: An example for solving a system by iteration

- Here is another example for the pair of equations whose plot is Fig. 5.

$$x^2 + y^2 = 4$$

$$e^x + y = 1$$

rearrangement;

$$y = -\sqrt{(4 - x^2)}x = ln(1 - y)$$

and begin with $x = 1.0$, the successive values for y and x are: See Table 4. and we are converging to the solution in an oscillatory manner.

y-value	x-value
-1.7291	1.0051
-1.72975	1.00398
-1.72961	1.00421
-1.72964	1.00416
-1.72963	1.00417

Table 4: Another example for solving a system by iteration