Introduction to successive iteration and open the door to numerical methods with the square root example. This worksheet includes several animation clips on successive iteration.
Instructor: Nam Sun Wang

Successive iteration is extremely common in numerical computations whether we are trying to find a solution to a (set of) linear or nonlinear algebraic equations, matrix inverse, or ordinary and partial differential equations. You also see it in probability transition matrix, chaos, and the generation of fractal patterns. The iterative scheme takes the general form of

$$x = g(x)$$

The above form allows us to start with an initial value of x, plug it into function g, and the function g will provide us with the next value of x. We insert the new value of x back into f(x) again, which will crank out yet another x for us. We can repeat the process for as many times as we wish. This process is called **successive iteration** or **successive approximation** (in cases where we resort to iteration to compensate for approximation). Note that the above successive iteration scheme contains a purely x term on the LHS. Let us illustrate the successive methods with the old-fashioned square root problem where the objective is to find a number x such that

$$x^2 = x \cdot x = a \qquad (1)$$

Note that algebra tells us that there are two roots. In the following discussion, let us take as an example:

$$a := 10$$

Of course, we symbolically denote such a number as:

$\sqrt{a}$ or $\sqrt{10}$ (And the other number as $-\sqrt{a}$ or $-\sqrt{10}$)

But what is its value? Here, we want to find the square root without resorting to the square root key on our calculator or without calling the built-in square root operator/function in a numerical package. We recall that 3*3=9 and 4*4=16, thus the answer should lie between 3 and 4. Let's start with:

$$x_0 := 3$$

We have to add a little bit of correction to x_0. Let this correction be ε_0. The updated root after making the correction is:

$$x_1 = x_0 + \varepsilon_0 = \sqrt{a}$$

To find ε_0, we square both sides of the last equation. Remember, we are after a number whose square is *a*.

$$\left(x_1\right)^2 = \left(x_0 + \varepsilon_0\right)^2 = a$$

Expanding (mark the last equation and choose |Symbolic|Expand Expression| from Mathcad menu) yields:

$$\left(x_1\right)^2 = \left(x_0\right)^2 + 2 \cdot x_0 \cdot \varepsilon_0 + \left(\varepsilon_0\right)^2 = a \longrightarrow \left(\varepsilon_0\right)^2 + 2 \cdot x_0 \cdot \varepsilon_0 + \left(x_0\right)^2 - a = 0 \qquad (2)$$

Substituting the values of x_0=3 and a=10, we get:

$$\left(\varepsilon_0\right)^2 + 2 \cdot 3 \cdot \varepsilon_0 + 3^2 - 10 = 0 \longrightarrow \left(\varepsilon_0\right)^2 + 6 \cdot \varepsilon_0 - 1 = 0$$

Recall the well known quadratic formula to solve for ε_0:

$$\varepsilon_0 = \frac{-6 - \sqrt{6^2+4}}{2} = -3 - \sqrt{10} \quad \text{and} \quad \varepsilon_0 = \frac{-6 + \sqrt{6^2+4}}{2} = -3 + \sqrt{10}$$

Ooops. We end up with an expression that contains the square root of 10, which we originally set out to solve in the first place. The problem arises because we are trying to solve the quadratic equation (2) in ε *exactly* with the quadratic formula. Since this leads us going in circles, we propose to **make an approximation --** the key word in numerical method! Let us drop the ε_0^2 term from Equation (2).

$$2 \cdot x_0 \cdot \varepsilon_0 + (x_0)^2 - a = 0$$

(The notation is a bit sloppy here, because I cannot find the approximation sign in Mathcad.)

With this approximation, we have a much more manageable equation that does not require the quadratic formula to find ε_0:

$$\varepsilon_0 = \frac{a - (x_0)^2}{2 \cdot x_0}$$

Let us actually plug some numbers into this formula.

$$\varepsilon_0 := \frac{a - (x_0)^2}{2 \cdot x_0} \qquad \varepsilon_0 = 0.167$$

The next value of x is:

$$x_1 := x_0 + \varepsilon_0 \qquad x_1 = 3.167$$

Check: $(x_1)^2 = 10.028$ ← not quite 10.

Since we have made an approximation by dropping the ε_0^2 term, the correction we have just made is not an exact one, as we can see from the check Nevertheless, we are now a step closer to the actual root. **The price to pay for making an approximation is iteration.** Now, x_1 is a better estimate of sqrt(10), let us make another correction ε_1 on top of x_1. We hope the resulting estimate x_2 will be an even better one.

$$x_2 = x_1 + \varepsilon_1$$

To find ε_1, we follow the same set of steps as before: square both sides of the last equation, equate it to *a*, and expand the square term.

$$(x_2)^2 = (x_1 + \varepsilon_1)^2 = a$$

$$(x_1)^2 + 2 \cdot x_1 \cdot \varepsilon_1 + (\varepsilon_1)^2 = a$$

$$(\varepsilon_1)^2 + 2 \cdot x_1 \cdot \varepsilon_1 + (x_1)^2 - a = 0$$

As before, we make an approximating by dropping the ε_1^2 term.

$$2 \cdot x_1 \cdot \varepsilon_1 + (x_1)^2 - a = 0$$

$$\varepsilon_1 = \frac{a - (x_1)^2}{2 \cdot x_1}$$

Plug in some numbers.

$$\varepsilon_1 := \frac{a - (x_1)^2}{2\cdot x_1} \qquad \varepsilon_1 = -0.004$$

The next value of x is:

$$x_2 := x_1 + \varepsilon_1 \qquad x_2 = 3.16228$$ Check: $(x_2)^2 = 10.00002 \leftarrow$ almost 10.

If we are satisfied with the results, we stop here. Otherwise, repeat until we are satisfied with the accuracy. We shall do it just one more time, this time without derivation.

$$\varepsilon_2 := \frac{a - (x_2)^2}{2\cdot x_2} \qquad \varepsilon_2 = -3.042\cdot 10^{-6}$$

The next value of x is:

$$x_3 := x_2 + \varepsilon_2 \qquad x_3 = 3.162277660170$$ Check: $(x_3)^2 = 10.000000000009$

Mathcad's build-in function gives: $\sqrt{10} = 3.162277660168$ which agrees with our value to more than 10 digits.

The **general iteration scheme for square root of *a*** is:

For $N := 5$ $\quad i := 0 .. N$ $\quad$ and provide x_0,

$$x_{i+1} := x_i + \frac{a - (x_i)^2}{2\cdot x_i}$$

Although we have started with a reasonably close initial guess of 3, this does not have to be so. We can see that any nonzero initial guess will eventually lead to a root with this scheme. (We may need more number of iterations though.) Zero is not a good initial guess because of the problem with division by 0.

For $N := 15$ $\quad i := 0 .. N$ $\quad x_0 := 1000$ $\leftarrow$ A lousy initial guess.

$$x_{i+1} := x_i + \frac{a - (x_i)^2}{2\cdot x_i}$$

$x_3 = 125.0262489500585$ $\leftarrow$ Not quite there yet.

$x_{last(x)} = 3.16227766016838$ $\leftarrow$ Good!

Furthermore, a different initial guess may lead to a different root when multiple roots exist. Of course, we know that x*x=a has two solutions, a positive one and a negative one, although sqrt(a) refers to the positive root and -sqrt(a) refers to the negative one.

$x_0 := -1000$ $\leftarrow$ Another lousy initial guess from the negative side.

$$x_{i+1} := x_i + \frac{a - (x_i)^2}{2\cdot x_i}$$

$x_3 = -125.0262489500585$ $\leftarrow$ Not quite there yet.

$x_{last(x)} = -3.16227766016838$ $\leftarrow$ Good, but a negative one.

The same formula is also applicable to different values of *a*.

$$a := 100$$

$$x_0 := 1$$

$$x_{i+1} := x_i + \frac{a - \left(x_i\right)^2}{2 \cdot x_i} \qquad x_{last(x)} = 10.000000000000000$$ ← Good!

In terms of the general iteration form of x=g(x) mentioned at the beginning of this worksheet, the iteration fomula is:

$$g(x) := x + \frac{a - x^2}{2 \cdot x} \qquad (3)$$

$$x_{i+1} := g\left(x_i\right)$$

In Mathcad version 6 (not valid in version 5), we can iterate elegantly by defining a function recursively. The following function says if x=g(x) is satisfied, return x and stop; otherwise call g(x) to update x and iterate until convergence. The intended usage is to issue the initial guess as the argument to the "iterate" function.

$$\text{iterate}(x) := (x = g(x)) \cdot x + (x \neq g(x)) \cdot \text{iterate}(g(x))$$

$\text{iterate}(3) =$ ... display the value returned with an initial guess of 3

Another way of saying the same thing:

$$\text{iterate}(x) := \text{if}(x = g(x), x, \text{iterate}(g(x)))$$

$\text{iterate}(3) =$ ... display the value returned with an initial guess of 3

Visualization of Successive Iteration x=g(x). The solution is where the iteration function g(x) intersects with the diagonal line, which is a straight function of x.

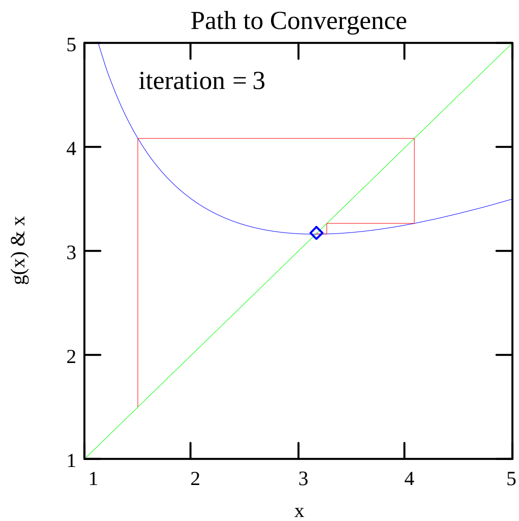
Converging Case -- Equation (3)

$a := 10 \qquad g(x) := \dfrac{x^2 + a}{2 \cdot x} \qquad N := 3 \qquad i := 0 .. N$

$x_0 := 1.5 \qquad x_{i+1} := g\left(x_i\right)$

Generate the steps for plotting:

$u_j := x_{\mathrm{floor}(0.5 \cdot j)} \qquad v_j := x_{\mathrm{floor}(0.5 \cdot (j+1))} \qquad xx := 1, 1.1 .. 5$



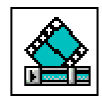
Animation section: toggle off the next equation and set FRAME=0..2N

$FRAME := 2 \cdot N \qquad FRAME = 6$

$j := 0 .. FRAME$

$iteration := \mathrm{floor}(0.5 \cdot FRAME)$

Click on the following icon to play an animation clip.



iterate1.avi

Let us now follow similar steps to find a **cubic root** of a given number *a*. The objective is to find x such that x*x*x=a. Approximate x first, then make a correction.

$a := 10$ ← A given number whose cubic root is to be estimated.

$x_0 := 2$ ← Initial guess.

$x_1 = x_0 + \varepsilon_0$

We calculate the correction term ε_0 from (|Symbolic|Expand Expression| with Mathcad):

$$(x_1)^3 = (x_0 + \varepsilon_0)^3 = a$$

$$(x_0)^3 + 3\cdot(x_0)^2\cdot\varepsilon_0 + 3\cdot x_0\cdot(\varepsilon_0)^2 + (\varepsilon_0)^3 = a$$

Ignore the quadratic term ($\varepsilon_0 2$) and the cubic term ($\varepsilon_0 3$):

$$(x_0)^3 + 3\cdot(x_0)^2\cdot\varepsilon_0 = a$$

$$\varepsilon_0 = \frac{a - (x_0)^3}{3\cdot(x_0)^2}$$

Plug in some numbers.

$$\varepsilon_0 := \frac{a - (x_0)^3}{3\cdot(x_0)^2} \qquad \varepsilon_0 = 0.167$$

Update x:

$x_1 := x_0 + \varepsilon_0 \qquad x_1 = 2.167 \qquad$ Check: $(x_1)^3 = 10.171$

Repeat the correction a couple more times:

$\varepsilon_1 := \frac{a - (x_1)^3}{3\cdot(x_1)^2} \qquad x_2 := x_1 + \varepsilon_1 \qquad x_2 = 2.1545 \qquad$ Check: $(x_2)^3 = 10.001$

$\varepsilon_2 := \frac{a - (x_2)^3}{3\cdot(x_2)^2} \qquad x_3 := x_2 + \varepsilon_2 \qquad x_3 = 2.15443469 \qquad$ Check: $(x_3)^3 = 10.00000003$

Mathcad's built-in routine gives: $a^{\frac{1}{3}} = 2.15443469$

It is clear that the scheme is converging rapidly to an actual cubic root. Thus, the **general iteration scheme for a cubic root of *a*** is:

For $N := 15 \quad i := 0..N$ and provide an initial guess x_0,

$$x_{i+1} := x_i + \frac{a - (x_i)^3}{3\cdot(x_i)^2}$$

We know from mathematical theories that there are three roots to a cubic equation. For $x^3 = a$, where *a* is a real number, there is a real root and two complex roots. To find a complex root, we must start with a complex guess:

$a := 1 \quad x_0 := 1 + -0.7i$

← Initial guess. The "i" here denotes the imaginary part, which is entered by typing "1i", which is not to be confused with the running index "i" below. The resulting answer is very sensitive to the initial guess. In fact, we can generate an interesting fractal pattern based on whether the resulting answer is the real number "1" or a complex one.

$$x_{i+1} := x_i + \frac{a - (x_i)^3}{3 \cdot (x_i)^2}$$

$x_{last(x)} = 1$

The **general iteration formula for the n-root of a given number a** is:

$$g(x) := x + \frac{a - x^n}{n \cdot x^{n-1}}$$

$$x_{i+1} := g(x_i)$$

Example: $n := 4 \quad a := 10$ with an initial guess of $x_0 := 2$ yields the following answer:

$x_{last(x)} = 1.77827941$

Mathcad's built-in routine gives: $a^{\frac{1}{n}} = 1.77827941$

Below, we present a different way of deriving an iteration formula for finding the square root of a given number *a*. We start with what we are trying to achieve.

$$x^2 = a$$

Let us add x^2 to both sides of the equation.

$$x^2 + x^2 = x^2 + a$$

$$2 \cdot x^2 = x^2 + a$$

Divide the above equation by 2x so that we obtain an equation in the form of x=g(x) suitable for iteration

$$x = \frac{x^2 + a}{2 \cdot x}$$

Thus, the iteration scheme is:

$$g(x) := \frac{x^2 + a}{2 \cdot x} \qquad (4)$$

Now, let us apply this iteration scheme to $a := 10$

Start with: $x_0 := 3$

$x_1 := g(x_0) \qquad x_1 = 3.167$

$x_2 := g(x_1) \qquad x_2 = 3.162280701754$

:

$x_{i+1} := g(x_i) \qquad x_{last(x)} = 3.162277660168$

A close inspection shows that Equation (4) is identical to Equation (3), which incidentally is identical to the Newton's method.

The algebraic equation f(x) to be solved for is: $x^2 - a = 0$

$$f(x) := x^2 - a \qquad f'(x) := 2\cdot x$$

$$x_{i+1} := x_i - \frac{f(x_i)}{f'(x_i)}$$

Many numerical methods exist for solving nonlinear equation of the form f(x)=0. Each one of these iterate based on the same general form of x=g(x) but a different specific functional form of g(x). Thus, deriving a good numerical algorithm is to find an expression of x=g(x). Who is to say that we cannot derive a different formula? There are infinite possibilities. For example, if we add a $2x^2$ term, instead of x^2, to both sides of the equation at the beginning of this section, the resulting iteration formula would be:

$$g(x) := \frac{2\cdot x^2 + a}{3\cdot x}$$

Now, let's apply this new iteration scheme to $a := 10$

Start with: $x_0 := 3$

$x_1 := g(x_0)$ $\qquad x_1 = 3.111$

$x_2 := g(x_1)$ $\qquad x_2 = 3.145502645503$

:

$x_{i+1} := g(x_i)$ $\qquad x_{last(x)} = 3.162277656689$

which is also eventually converging to the value of $\sqrt{10} = 3.162277660168$

Let us try another scheme by simply adding a term x to the both sides of equation (1). This is a favorite trick for many because it quickly converts an algebraic equation of the form 0=f(x) into a successive iteration form x=g(x), which is simply x+f(x).

$$x + x^2 = x + a$$

$$x = x + a - x^2$$

$$g(x) := x + a - x^2 \qquad (5)$$

Now, let us apply this new iteration scheme to $a := 10$

Start with: $x_0 := 3$

$x_1 := g(x_0)$ $\quad x_1 = 4$

$x_2 := g(x_1)$ $\quad x_2 = -2$

$x_3 := g(x_2)$ $\quad x_3 = 4$

:

$x_{i+1} := g(x_i)$ $\quad x_{last(x)} = -2$

Hmm... The value of x is just switching back and forth, not at all converging.

Plot for Switching Case -- Equation (5)

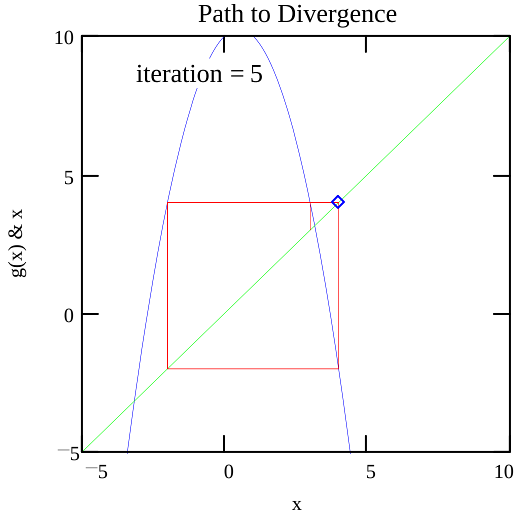
$N := 5 \qquad i := 0 .. N$

$x_0 := 3 \qquad x_{i+1} := g(x_i)$

Generate the steps for plotting:

$u_j := x_{floor(0.5 \cdot j)} \qquad v_j := x_{floor(0.5 \cdot (j+1))} \qquad xx := -5, -4.9 .. 10$

Path to Divergence



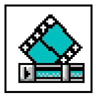
Animation section: toggle off the next equation and set FRAME=0..2N

$FRAME := 2 \cdot N \qquad FRAME = 10$

$j := 0 .. FRAME$

$iteration := floor(0.5 \cdot FRAME)$

Click on the following icon to play an animation clip.



iterate2.avi

Let us try another scheme. If we add 0.95*x to both sides of equation (1), we get:

$$0.95\cdot x + x^2 = 0.95\cdot x + a$$

$$x = x + \frac{a - x^2}{0.95}$$

$$g(x) := x + \frac{a - x^2}{0.95} \qquad (6)$$

Start with: $x_0 := 3$

$x_1 := g(x_0)$ $\quad x_1 = 4.053$

$x_2 := g(x_1)$ $\quad x_2 = -2.709$

$x_3 := g(x_2)$ $\quad x_3 = 0.090$

$x_4 := g(x_3)$ $\quad x_4 = 10.608$ $\quad$ ← Oscillating away from the solution.

Hmm... The value of x is getting larger and larger and not converging into a single number. This is called **divergence**.

Plot for Diverging Case -- Equation (6)

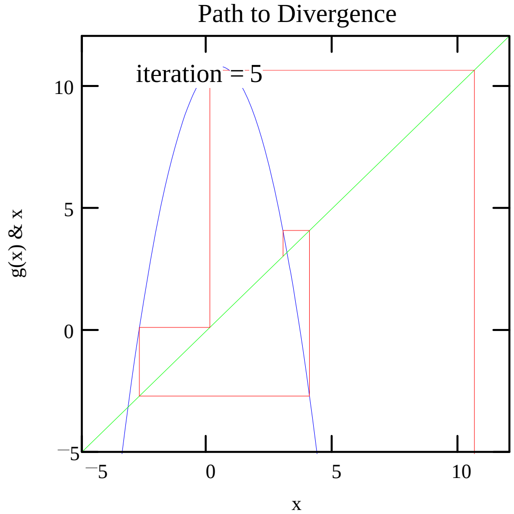
$N := 5 \qquad i := 0 .. N$

$x_0 := 3 \qquad x_{i+1} := g(x_i)$

Generate the steps for plotting:

$u_j := x_{\mathrm{floor}(0.5\cdot j)} \qquad v_j := x_{\mathrm{floor}(0.5\cdot(j+1))} \qquad xx := -5, -4.9 .. 12$

Path to Divergence



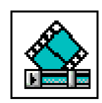
Animation section: toggle off the next equation and set FRAME=0..2N

$FRAME := 2\cdot N \qquad FRAME = 10$

$j := 0 .. FRAME$

$iteration := \mathrm{floor}(0.5\cdot FRAME)$

Click on the following icon to play an animation clip.



iterate3.avi

Now, if we add 1.05*x to both sides of equation (1), we get:

$$g(x) := x + \frac{a - x^2}{1.05} \qquad (7)$$

← Try different values for the denominator, in place of 1.05.

Start with: $x_0 := 3$

$x_1 := g(x_0)$ $\quad x_1 = 3.952$

$x_2 := g(x_1)$ $\quad x_2 = -1.401$

$x_3 := g(x_2)$ $\quad x_3 = 6.253$

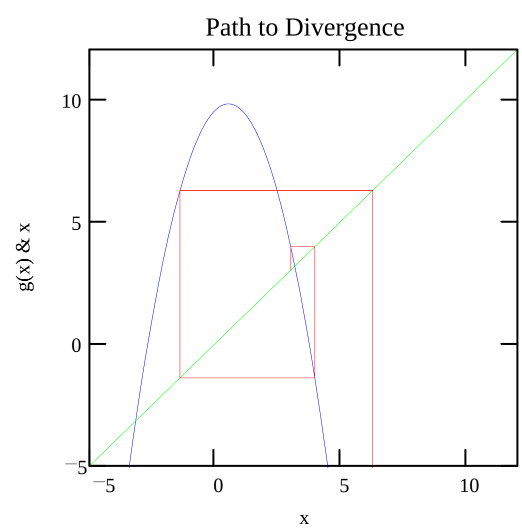
$x_4 := g(x_3)$ $\quad x_4 = -21.456$ ← Getting away from the solution.

:

$x_{i+1} := g(x_i)$ $\quad x_{last(x)} = -2$

Plot for Diverging Case -- Equation (7)

$u_j := x_{floor(0.5\cdot j)}$ $\quad v_j := x_{floor(0.5\cdot(j+1))}$ $\quad xx := -5, -4.9 .. 12$



Another easy way of deriving an iteration expression is to reduce the given algebraic equation first into the form of f(x)=0. This is also a favorite step for many. Since f(x)=0, we can multiply, divide or do almost anything to it and the resulting expression is still 0. Then, we can add a term x to both side of the equation to reach an iteration form of x=g(x). Let us demonstrate this approach where we divide f(x) by 2x first, then add x.

Step 1. Reduce x^2=a to the form f(x)=0

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

Step 2. Multiply by -1/2x (or some other expression of x)

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

Step 3. Add x to both sides (which is an easy way of generating the iteration form x=g(x)).

$$x = x - \frac{x^2 - a}{2\cdot x}$$

Step 4. Find g(x), which is the RHS of the last equation

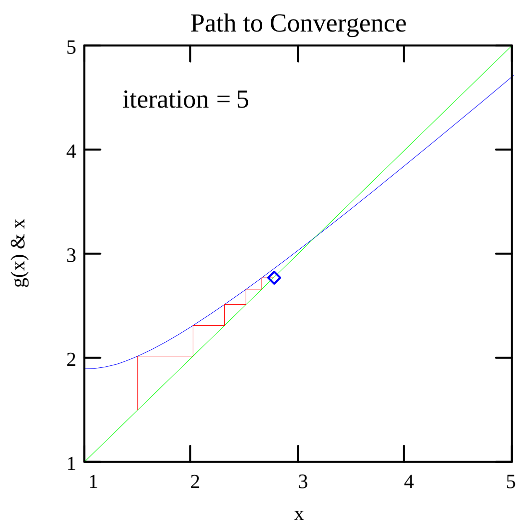
$$g(x) := x - \frac{x^2 - a}{2\cdot x}$$

The resulting formula is the same as that from the Newton's method. Different numerical algorithms mainly differ in the expression mutiplied to f(x)=0 in Step 2. If we choose to multiply by -0.1/x instead, we get an expression that has a slower convergence property near the root. A slower convergence is not necessarily bad. We sometimes prefer a slower convergence when we want to approach the root gingerly and conservatively.

$$g(x) := x - \frac{0.1}{x}\cdot\left(x^2 - a\right)$$

Let us generate some numbers based on the above iteration formula and visualize convergence graphically.

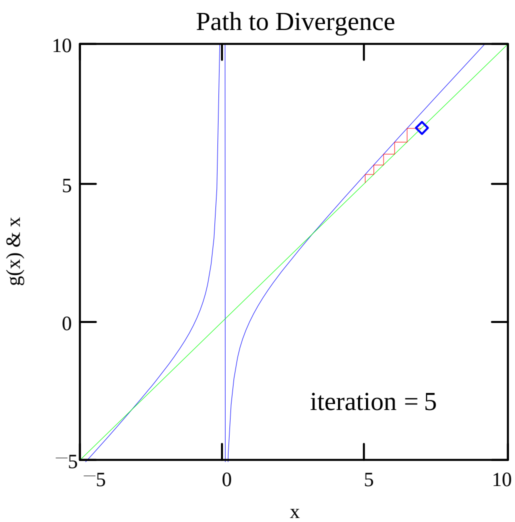
$x_0 := 1.5$ $\quad x_{i+1} := g\left(x_i\right)$ $\quad u_j := x_{floor(0.5\cdot j)}$ $\quad v_j := x_{floor(0.5\cdot (j+1))}$



Of course, not all iteration formula lead to a root. On the other hand, if we choose to multiply by +0.1/x, we get divergence.

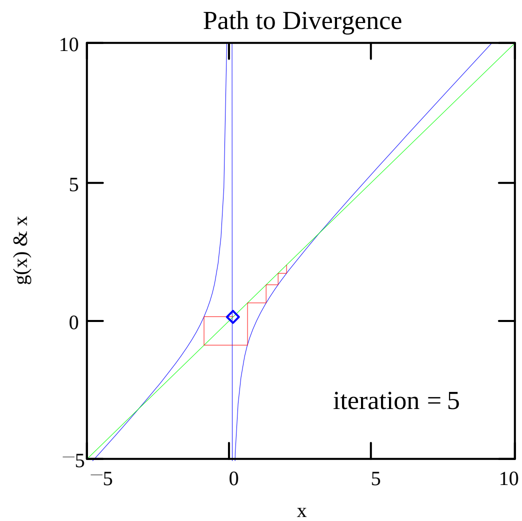
$$g(x) := x + \frac{0.1}{x}\cdot\left(x^2 - a\right)$$

$x_0 := 5$ $\quad x_{i+1} := g\left(x_i\right)$ $\quad u_j := x_{floor(0.5\cdot j)}$ $\quad v_j := x_{floor(0.5\cdot(j+1))}$



$x_0 := 2$ $\quad x_{i+1} := g\left(x_i\right)$

$u_j := x_{floor(0.5\cdot j)}$ $\quad v_j := x_{floor(0.5\cdot(j+1))}$

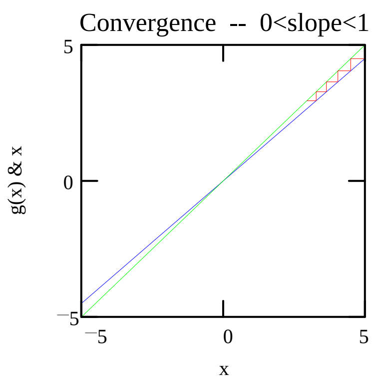


When do we achieve convergence and when do we face divergence? **In general, we reach convergence when the |slope of g(x)| at the point of intersection with x is within -1 to 1**. Furthermore, as shown below, we achieve very fast convergence when the slope of g(x) is close to 0. On the other hand, convergence is slow when the slope of g(x) is close to 1 or -1. Hence, the trick is to come up with an iteration scheme such that the slope of g(x) lies within -1 and 1, preferably close to 0. If we did not know anything about the convergence property, we have about 50% chance on the average of coming up with a converging formula (and 50% chance of a diverging one). So if the first try does not work, keep trying different manipulations to get x=g(x). The chances are you will hit one converging formula if you tried often enough. Below, we graphically demonstrate these two cases.

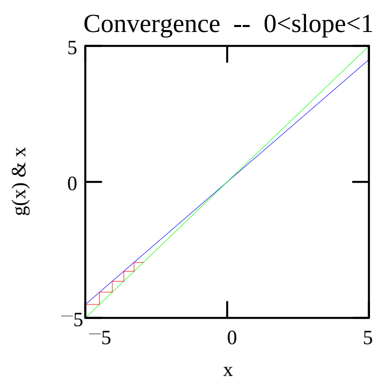
Case 1a. *0<slope<1* -- Monotonic convergence to the intersection of x and g(x) for all initial guesses

$i := 0 .. 10$

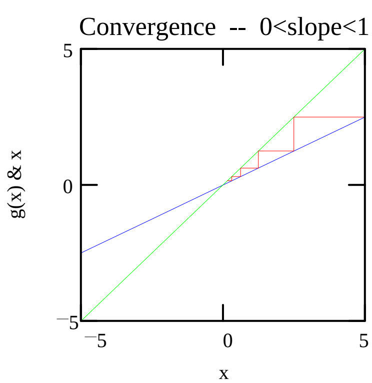
$g(x) := 0.9 \cdot x \qquad x_0 := 5 \qquad x_{i+1} := g(x_i) \qquad u_j := x_{floor(0.5 \cdot j)} \qquad v_j := x_{floor(0.5 \cdot (j+1))} \qquad xx := -5, -4.9 .. 5$



$x_0 := -5 \qquad x_{i+1} := g(x_i) \qquad u_j := x_{floor(0.5 \cdot j)} \qquad v_j := x_{floor(0.5 \cdot (j+1))}$

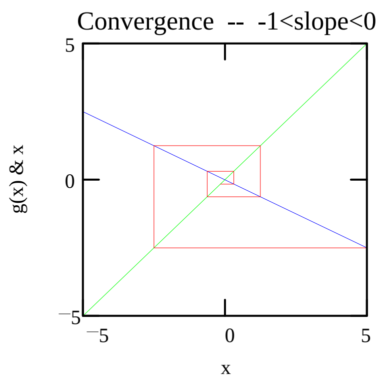


$g(x) := 0.5 \cdot x \qquad x_0 := 5 \qquad x_{i+1} := g(x_i) \qquad u_j := x_{floor(0.5 \cdot j)} \qquad v_j := x_{floor(0.5 \cdot (j+1))}$

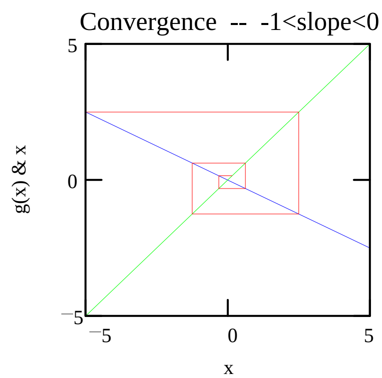


Case 1b. *-1<slope<0* -- Oscillatory convergence to the intersection of x and g(x) for all initial guesses

$g(x) := -0.5 \cdot x \quad x_0 := 5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$

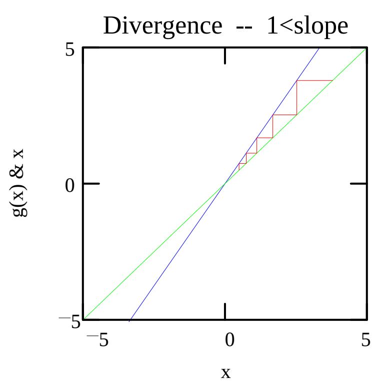


$x_0 := -5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$

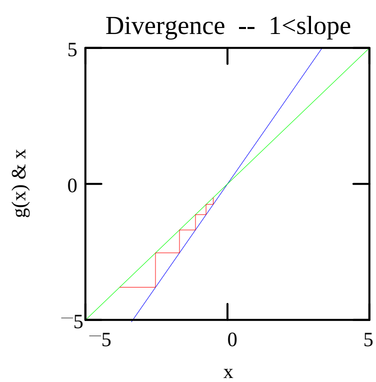


Case 2a. *1<slope* -- Monotonic divergence from the intersection of x and g(x) for all initial guesses

$g(x) := 1.5 \cdot x \quad x_0 := 0.5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$

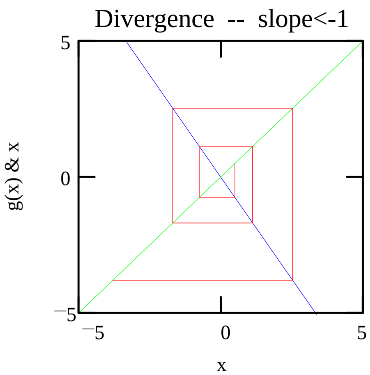


$x_0 := -0.5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$



Case 2b. *slope<-1* -- Oscillatory divergence from the intersection of x and g(x) for all initial guesses

$g(x) := -1.5 \cdot x \quad x_0 := 0.5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$



$x_0 := -0.5 \quad x_{i+1} := g(x_i) \quad u_j := x_{floor(0.5 \cdot j)} \quad v_j := x_{floor(0.5 \cdot (j+1))}$