Numerical Methods: Computer Assignment #2

Hong Chenhui
drredthered@gmail.com

Zhejiang University — March 12, 2024

Info: This piece of work was only intended as a rough exercise for how fixed-point iteration works, not the detailed mechanics. One could follow the hyperlink attached to this page to find more info.

1 Problem

Determinate the positive root of fuction $f(x) = x^2 - 3 = 0$ with fix-point iteration methods in the following form.

1. $g(x) = x^2 - 3 + x$.
2. $g(x) = x - \frac{x^2-3}{4}$.
3. $g(x) = \frac{1}{2}(x + \frac{3}{x})$.

1.1 Theoretical viewpoint

Question

Simple fixed-point iteration (or, as it is also called, one-point iteration or successive substitution) is applied by rearranging the function $f(x) = 0$ so that x is on the left-hand side of the equation:

$$x = g(x) \tag{1}$$

This transformation can be accomplished either by algebraic manipulation or by simply adding x to both sides of the original equation.

The utility of Eq. (1) is that it provides a formula to predict a new value of x as a function of an old value of x. Thus, given an initial guess at the root x_i, Eq. (6.1) can be used to compute a new estimate x_{i+1} as expressed by the iterative formula

$$x_{i+1} = g(x_i) \tag{2}$$

As with other iterative formulas, the approximate error for this equation can be determined using the error estimator

$$\epsilon_a = |\frac{x_{i+1} - x_i}{x_{i+1}}|100\% \tag{3}$$

1.2 Algorithmic issue

Question

The computer algorithm for fixed-point iteration is extremely simple. It consists of a loop to iteratively compute new estimates until the termination criterion has been met.

(a) Specify ϵ_s, max iteration times i_{max} and a starting point x_0.

(b) Iterate with $x_{i+1} = g(x_i)$ until reaching i_{max} or ϵ_s.

(c) Output the result x_r.

2 Implementation

$MATLABR2023a$ codes seen at the codes section as well as attached files. All iteration process was done with $x_0 = 3$, $i_{max} = 1000$ and $\epsilon_s = \frac{1}{2} \times 10^{-6}$ which guarantees six-digit accuracy.

1. Results

	x_r	$iteration$	ϵ_a
$g(x) = x^2 - 3 + x$	Inf	1000	NaN
$g(x) = x - \frac{x^2-3}{4}$	1.732050806888473	11	2.539304021350097e-07
$g(x) = \frac{1}{2}(x + \frac{3}{x})$	1.732050807568877	5	1.412112265044279e-07

2. Figure tracking the iteration process

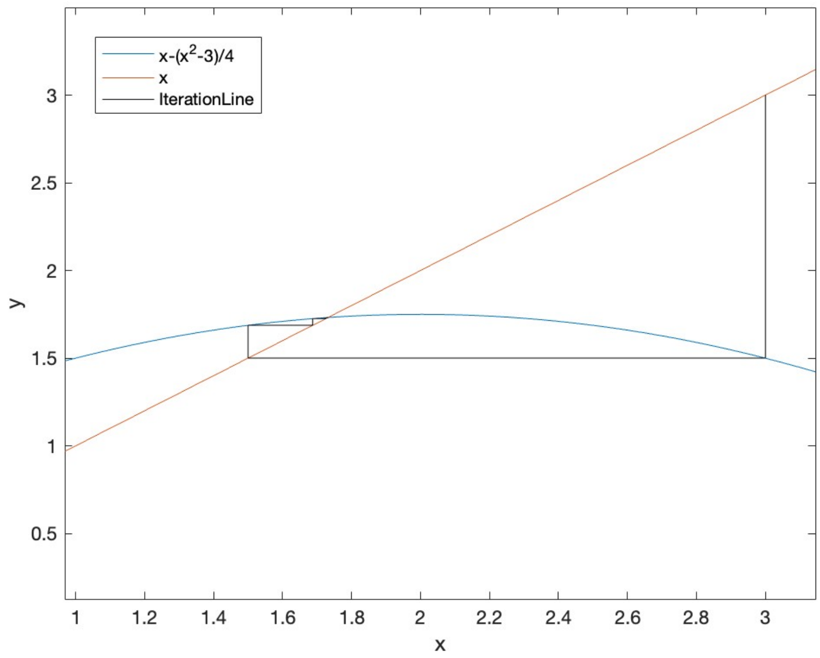

Fig.1 $g(x) = x - \frac{x^2-3}{4}$ with excellent result

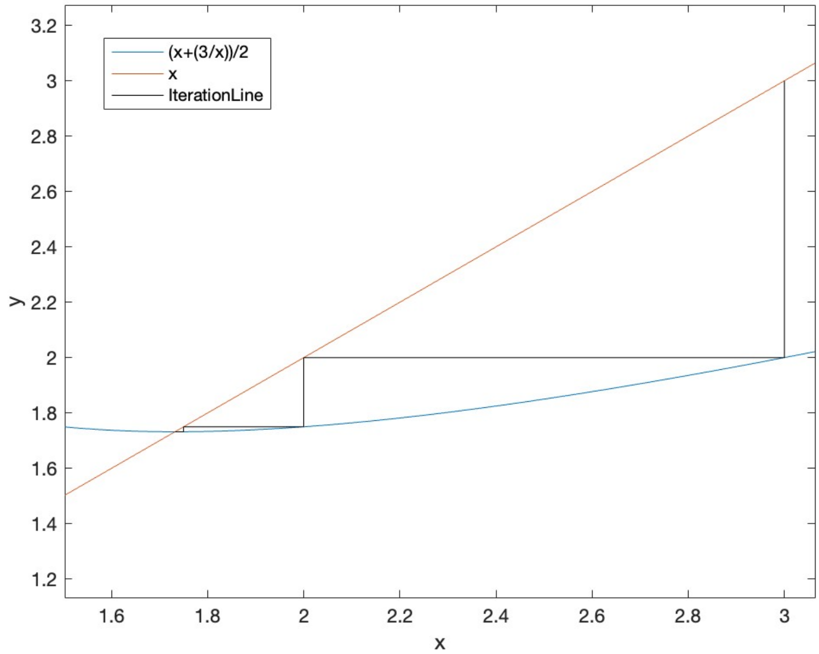

Fig.2 $g(x) = \frac{1}{2}(x + \frac{3}{x})$ with even better result

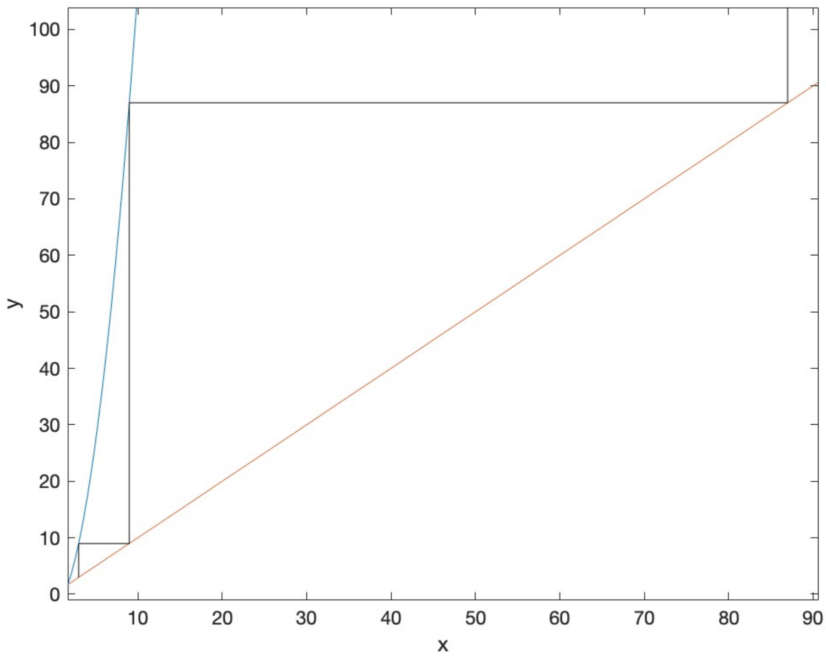

Fig.3 $g(x) = x^2 - 3 + x$ with off-putting diverge result

3 Analysis

3.1 Why is the first method diverging?

It should be clear that fixed-point iteration converges if, in the region of interest, $|g'(x)| < 1$. In other words, convergence occurs if the magnitude of the slope of $g(x)$ is less than the slope of the line $f(x) = x$.

This observation can be demonstrated theoretically at *Numerical Methods in Economics. Cambridge: MIT Press. pp. 165–167.* In general it is the result of

$$E_{t,i+1} = g'(\xi)E_{t,i} \tag{4}$$

Calculating the first method with $g(x) = x^2 - 3 + x$, we have $g'(x) = 2x + 1$, which is greater than 1 when $x > 0$, so the method must diverges.

3.2 What about the other two methods?

While it is clear that the other two approaches were designed to avoid divergence, the means by which they accomplish this differ.

The first one is extremely straight-forward, yet brutal to some extent. Since the original derivativeis no matter what larger than 1, one simply divide it by a constant to make it smaller. This is a common practice in numerical methods, and it is also the most effective way to avoid divergence (And just so to make it converge into the right direction, it added a negative sign to the fraction). In doing so makes

$$g'(x) = -\frac{1}{2}x + 1 \tag{5}$$

whose absolute value is less than 1 at $x \in [0, 4]$, so it converges at $x = 3$.

The second one is tricky to understand. It is something called **Heron's method**, after the first-century Greek mathematician Hero of Alexandria who described the method in his AD 60 work Metrica. This method is also called the Babylonian method (not to be confused with the Babylonian method for approximating hypotenuses), although there is no evidence that the method was known to Babylonians.

This is equivalent to using **Newton's method** to solve $f(x) = x^2 - 3$. This algorithm is quadratically convergent: the number of correct digits roughly doubles with each iteration. If one recalls, it was described as following

$$g(x) = x - \frac{f(x)}{f'(x)} \tag{6}$$

in this case,

$$\begin{aligned} g(x) &= x - \frac{x^2 - 3}{2x} \\ &= \frac{1}{2}(x + \frac{3}{x}) \end{aligned} \tag{7}$$

Which is sort of cheating considering the fact that we only requires linear convergence for fix-point iteration. But still good way to avoid divergence. Details at this wiki page.

4 Codes

Matlab Codes are attached below and in the file. Also at my private blog: drredthered.github.io.

Fixpt.m

```
function [xr,iter,ea] = Fixpt(x0,es,imax,g)

%initialization
xrold = x0;
xr = g(xrold);
```

Fixpt.m

```
iter = 0;
ea = 0;

%plotting
fplot(g);
hold on
fplot(@(x)x);

%recurrence iteration
while(1)
        plot([xrold xrold], [xrold xr], 'k-')
        xlim([0,5])
        ylim([0,5])
        plot([xrold xr], [xr xr], 'k-')
        xlim([0,5])
        ylim([0,5])

        xrold = xr;
        xr = g(xrold);
        iter = iter + 1;
        if xr~= 0
                ea = abs((xrold - xr)/xr)*100;
        end
        if ( iter >= imax || ea < es ),break,end
end
%result
end
```

Command Line

```
>> [xr,iter,ea] = Fixpt(3,0.5e-6,1000,@(x)x.^2-3+x)
>> [xr,iter,ea] = Fixpt(3,0.5e-6,1000,@(x)x-(x.^2-3)./4)
>> [xr,iter,ea] = Fixpt(3,0.5e-6,1000,@(x)(x+3./x)./2)
```

Warning: For some interesting reason, Matlab argues that support for character vector inputs will be removed in a future release, and one shall use function handles instead. However, common expression such as fplot('x') should still be available befor the update took effect. Use wisely if one obtains this document after 2023.

Referrence

1. Heath, Thomas (1921). A History of Greek Mathematics, Vol. 2. Oxford: Clarendon Press. pp. 323–324.

2. Judd, Kenneth L. (1998). "Fixed-Point Iteration". Numerical Methods in Economics. Cambridge: MIT Press. pp. 165–167. ISBN 0-262-10071-1.

3. Canale, R. P., & Chapra, S. C. (2014). Numerical methods for engineers. Mcgraw-hill Education-Europe. pp. 149-153. ISBN 0-072-91873-X.