

We review methods for finding roots of nonlinear functions such as fixed-point method and Newton's method. We will do a mix of coding and theory exercise.

This worksheet, and the demonstration code, can be found at https://github.com/Empyreal092/NA_Worksheet.

1 Fixed point methods

We want to find roots of the nonlinear equation

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

using the fixed point method.

(1.a) Plot $f(x)$ in your favorite programming language.

1. Remember to add a title and axis labels to your plot.
2. Try to “generalize” your code. For example, if we want to plot another function, is it straightforward to edit your code to adapt to the new task?

(1.b) The problem has two roots. What are they?

We will try to find the positive root (x^*) through the fixed point iteration of the form $x_{k+1} = g(x_k)$. We investigate two choices:

- $g_1(x) = x^2 - 2$
- $g_2(x) = \sqrt{x + 2}$

(1.c) Verify that x^* is indeed fixed points for the two functions. That is, $x^* = g(x^*)$.

1. You could verify this with a plot. (If there are two lines in a plot, put legends on them.)

(1.d) Will both choices work in the fixed point algorithm to find the root x^* ? (Hint: consider the stability of the fixed point.)

1. Could you determine the stability using the plot you made in [\(1.c\)](#)?

(1.e) Implement the fixed point method for both choices of $g(x)$ above.

1. Set $x_0 = 5$.

2. What would be some good termination criteria for the fixed point algorithm?
3. Try to “generalize” your code. For example, what if we want to pick a new function $g(x)$?

Now consider the convergence behavior of the stable fixed point algorithm.

(1.f) From theory alone, how fast do you expect the convergence? (linear/superlinear/quadratic)? What is the expected convergence rate?

(1.g) Is this what your numerical results show? How do you verify the convergence behavior numerically? Try to show this with a plot.

1. Plot a representation of the error $|x_k - x^*|$ against iteration numbers. Which methods of plotting should we use? `plot`, `semilogy`, or `loglog`?
2. In more complicated examples we do not know x^* . What should we plot on the y -axis?
3. Did you remember to add a title and axis label to your plot?

(1.h) Could you diagnose the convergence rate numerically? Does it match the theoretical expectation?

(1.i) Is it possible to use $g_2(x)$ in the fixed-point iteration to find the negative root? Could we modify it so that it would work? In your spare time, you could verify the convergence behavior to the negative root as an exercise.

2 Newton's method and roots with higher multiplicity

In the HW we have an example showing that Newton's method no longer converges quadratically if the root has higher than one multiplicity. We explore some more in this direction.

We first define multiplicity: $r \in \mathbb{R}$ is a root of multiplicity m for the equation $f(x) = 0$ if there is a function $h(x)$ such that $h(r) \neq 0$ and $f(x) = (x - r)^m h(x)$.

(2.a) Suppose that a function f has m continuous derivative on the interval (a, b) containing c (i.e. $f(x) \in C_{(a,b)}^m$). Show that f has a zero of multiplicity m at c if and only if

$$0 = f(r) = f'(r) = \dots = f^{(m-1)}(r)$$

and

$$f^m(r) \neq 0.$$

(2.b) Suppose r is a zero of multiplicity m of f , and $f(x) \in C_{(a,b)}^m$, $r \in (a, b)$. Show that the following fixed-point method has $g'(r) = 0$:

$$g(x) = x - m \frac{f(x)}{f'(x)}.$$

What can you say about the convergence behavior of this fixed-point method?

(2.c) Code this modified Newton's method to solve $f(x) = x^2 = 0$. Check the convergence numerically. How do you plan to show this?

3 Some more tips about plotting

(3.a) Size your figure so that they fit neatly onto a page without having to scale them in \LaTeX .

(3.b) Exporting as vector images.

(3.c) Set desired behaviors as default.

My bag of MATLAB tools: https://github.com/Empyreal092/MATLAB_Tool.