Lecture `root2`: Nonlinear Equations: Fixed-Point Iteration

September 9, 2022

Summary: Discussion of fixed-point iteration and the Contraction Mapping Theorem.

References: Section 1.2, pages 32-43 of Sauer's textbook.

Last time we compared a system of linear equations

$$A\mathbf{x} = \mathbf{b},$$

where $\mathbf{x}$ is the unknown, to a system of *nonlinear* equations

$$\mathbf{f}(\mathbf{x}) = \mathbf{0},$$

where $\mathbf{f}$, $\mathbf{x}$, and $\mathbf{0}$ are vectors. We mentioned the characterization of n-by-n linear systems:

$$\det A \neq 0 \Longrightarrow A\mathbf{x} = \mathbf{b} \text{ has exactly one solution } \mathbf{x}. \text{ (What is the formal expression for } \mathbf{x}\text{?)}$$

Furthermore, if $\det A = 0$, then either the system has no solution or infinitely many solutions (either loss of existence or dramatic loss of uniqueness). As mentioned last time, this characterization applies to the 1-by-1 case $ax = b$, where $\det a = a$.

However, this characterization does not have a counterpart for nonlinear equations (scalar or system). Nonlinear equations may have 0, 1, 2, 13, 264, ∞, or whatever number of solutions, as may be seen with the following *scalar* examples:

$$\sin x = 0 \tag{1}$$

$$e^x = -3 \tag{2}$$

$$x^2 - 1 = 0. \tag{3}$$

On the real line, Eq. (1) has infinitely many solutions, (2) has no solutions, and (3) has two solutions. The upshot is that solving nonlinear equations is more tricky than solving linear equations. Last time we considered solving

$$f(x) = 0 \tag{4}$$

via bisection, and here consider the root-finding algorithm known as fixed point iteration.

We'll begin by assuming that our function takes a slightly different form than in Eq. (4):

$$x = f(x). \tag{5}$$

Note that Eqs. (4) and (5) are interchangeable: we need only redefine $f(x)$ to go from one form to the other. In any case, the form of Eq. (5) suggests that we try to solve the equation via the following simple method. Let x_0 be a given starting guess for the solution to (5). Then define the x_i successively as follows:

$$
\begin{aligned}
x_1 &= f(x_0) \\
x_2 &= f(x_1) \\
x_3 &= f(x_2) \\
x_4 &= f(x_3) \\
&\vdots
\end{aligned}
$$

The above sequence of x_i represents a *fixed-point iteration* scheme. That is, we are searching for a *fixed-point* $x^\star$ of the function $f(x)$:

$$f(x^\star) = x^\star.$$

At present, we've no reason to expect that the algorithm will work, but it might.

Example 1
Consider the nonlinear equation

$$x + \frac{1}{\pi}\sqrt{|x+3|} = \frac{1}{2}\sin x \tag{6}$$

We cast this equation into the form (5):

$$x = \frac{1}{2}\sin x - \frac{1}{\pi}\sqrt{|x+3|}. \tag{7}$$

Defining $f(x) := \frac{1}{2}\sin x - \pi^{-1}\sqrt{|x+3|}$, we are now looking for a fixed-point of $f(x)$.

Figure 1 plots $y = f(x)$ and $y = x$ around their single point of intersection. The intersection point is evidently around $x = -1$. Let us now consider how we might go about solving this equation via fixed-point iteration. We write a MATLAB function `iteration.m` to implement our algorithm. It's easy to perform iteration $x_{i+1} = f(x_i)$ indefinitely, but we need to tell MATLAB to stop at some point. To do this, we consider[1]

$$|x_{i+1} - x_i| = |f(x_i) - x_i| := e_i,$$

that is the error measure $|x_{i+1} - x_i|$, the difference between successive iterates. We then suggest a criterion that requires $|x_{i+1} - x_i|$ to be less than 10^{-8}. Given this criterion, an initial guess `x0`, and a function handle using the @ symbol which evaluates $f(x)$, a bare-bones iteration setup might look like the following.

```
tol = 1e-8;
x = x0;
err=1;

while err>tol;
```

[1]On page 37, Sauer defines $e_i = |x^\star - x_i|$, that is the error of the iterate x_i measured relative to the root $x^\star$ (but Sauer uses r for the root). Here our e_i is between successive iterates.

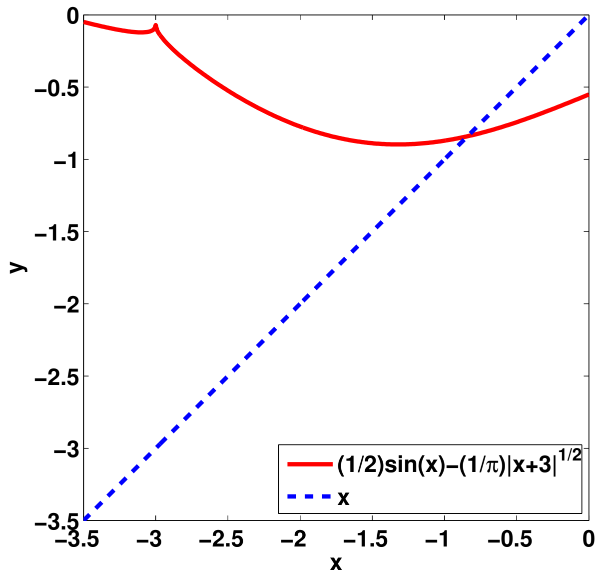

Figure 1: The functions x (the identity) and $\frac{1}{2}\sin x - \pi^{-1}\sqrt{|x+3|}$ near their common point of intersection, the location $x^\star$ we are trying to find.

```
  x0 = x;
  x = f(x0);
  err = abs(x-x0);
end
```

This implementation has one serious problem: if the x_i do *not* get closer to the solution $x^\star$, then we'll continue to iterate forever: the condition `err>tol` which controls the while loop will never be false! To avoid this possibility, we add a counter to record the number of iterations. When a maximum number `kmax` of iterations is reached, we'll just give up:

```
tol = 1e-8;
x = x0;
err=1;
kmax=1e5;
k = 0;

while err>tol;
  x0 = x;
  x = f(x0);
  err = abs(x-x0);
  k = k+1;

  if k>kmax;
    disp(['Could not converge to solution after ' ...
    num2str(kmax) ' iterations']);
```

```
    return;
  end
end
```

The MATLAB command `return` exits from the current `m`-file immediately, without executing the remaining lines of code. We add some more minor features and save the routine as `iteration.m`. We then define the function `lecroot2_fun1.m` (first function of lecture **root2**).

```
function[y] = lecroot2_fun1(x);
y = 1/2*sin(x) - 1/pi*sqrt(abs(x+3));
```

The only thing we are missing now is an initial guess x_0. For no particularly good reason, we choose $x_0 = 0$, go to the command line, and run the following.

```
>> format long g; format compact
>> x0 = 0; tol = 1e-8; kmax = 1e5;
>> iteration(@lecroot2_fun1,x0,tol,kmax)
Converged in 14 iterations with tol 1e-08
ans =
        -0.840180274139402
```

Thus, the answer above is good to about eight decimal places, and the iterate returned was x_{14}. Note the very quick convergence to the solution: only 14 iterations were required. This example was designed so that our algorithm would work. In many cases the algorithm will not work, but at the very least, we have the following theorem.

Theorem 1. *Let $f(x)$ be a scalar real-valued function which is continuous on $[a, b]$ and continuously differentiable on (a, b), with*

$$\left|f'(x)\right| < 1, \qquad \text{for all } \ x \in (a, b). \tag{8}$$

Suppose further that $x \in [a, b] \implies f(x) \in [a, b]$. Then on the interval $[a, b]$, there exists a unique fixed-point $x^\star$ satisfying the equation

$$x^\star = f(x^\star),$$

and the sequence of iterates defined by

$$x_{i+1} = f(x_i)$$

will converge to $x^\star$ if $x_0 \in [a, b]$.

This is a specific realization of the **Contraction Mapping Theorem** (aka the **Banach fixed point theorem**), whose proof is beyond the scope of the course. It provides conditions under which fixed-point iteration will converge to a unique solution. However, we have the restrictive requirement (8), and that $f(x)$ must map points in $[a, b]$ to points in $[a, b]$. If we *already know* there is a fixed-point $x^\star = f(x^\star)$ and simply want to know if we can approximate

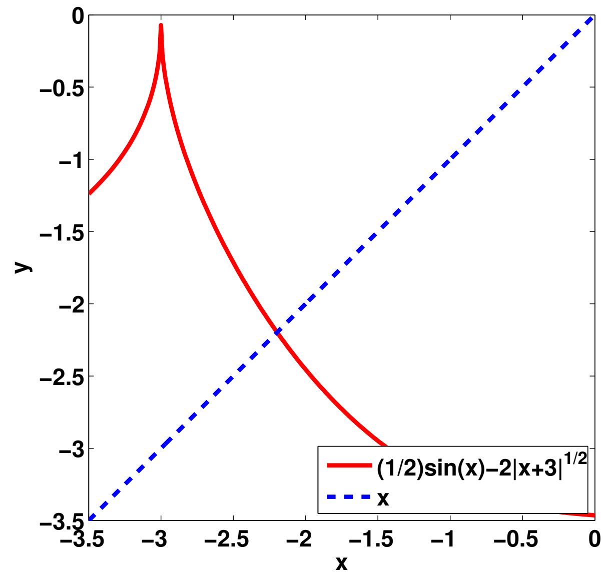

Figure 2: The functions x (the identity) and $\frac{1}{2}\sin x - 2\sqrt{|x+3|}$ near their common point of intersection, the location $x^\star$ we are trying to find.

it via fix-point iteration, then Sauer's textbook gives an easier-to-use result on page 37:

Theorem 2. *Assume $f(x)$ is continuously differentiable on a neighborhood of $x^\star$, that $x^\star = f(x^\star)$, and that $|f'(x^\star)| < 1$. Then fixed-point iteration converges to the point $x^\star$ for initial guesses sufficiently close to $x^\star$.* (We'll talk about the rate of convergence in class.)

Let us return to **Example 1**, where our function $f(x)$ is defined in Eq. (7). Let us now assume $x > -2$, so that $x + 3 > 1$ and $|x + 3| = x + 3$. Moreover, $x + 3 > 1 \implies \sqrt{x+3} > 1 \implies 1/\sqrt{x+3} < 1$. With these results, we see that the $x > -2$ assumpition yields

$$\begin{aligned} |f'(x)| &= \left| \frac{1}{2}\cos x - \frac{1}{2\pi\sqrt{x+3}} \right| \\ &\le \left| \frac{1}{2}\cos x \right| + \left| \frac{1}{2\pi\sqrt{x+3}} \right| \\ &\le \frac{1}{2} + \frac{1}{2\pi} \\ &< 1. \end{aligned}$$

In particular, $|f'(x^\star)| < 1$, since from Fig. 1 it's clear that $x^\star > -2$. Moreover, $f(x)$ is continuously differentiable on all open intervals not containing the point $x = -3$. **Theorem 2** above then establishes the convergence we saw earlier.

Example 2

Let's consider an equation that is very similar to that in **Example 1**. Let's try to solve

$$x + 2\sqrt{|x+3|} = \frac{1}{2}\sin x. \tag{9}$$

This equation has the same form as Eq. (6). The only thing we have changed is the constant in front of the $\sqrt{|x+3|}$ term; it was $1/\pi$ and now it's 2. We write our new equation as

$$x = \frac{1}{2}\sin x - 2\sqrt{|x+3|} := f(x), \tag{10}$$

and define `lecroot2_fun2.m` which evaluates $f(x)$ as $\frac{1}{2}\sin x - 2\sqrt{|x+3|}$. The solution $x^\star$ to the equation $x = f(x)$ above is around $x = -2.2$ (see Fig. 9 for a graphical demonstration); however, we see visually that $|f'(x^\star)| > 1$. The key assumption of **Theorem 2** is not satisfied for $f(x) = \frac{1}{2}\sin x - 2\sqrt{x+3}$, and we cannot apply the theorem to guarantee convergence. Nevertheless, let's see if the iteration converges anyway:

```
>> x0 = 0; tol = 1e-8; kmax = 1e5;
>> iteration(@lecroot2_fun2,x0,tol,kmax)
No convergence after 100000 iterations.
Error at this stage is 2.2976
ans =
         -0.96606121638565
>> x0 = -2.2
x0 =
                      -2.2
>> iteration(@lecroot2_fun2,x0,tol,kmax)
No convergence after 100000 iterations.
Error at this stage is 2.2976
ans =
         -3.26365087819059
```

This algorithm can not find the solution even if our initial guess is taken very close to the solution. When we took $x_0 = 0$, the iterates x_i indexed by i were

```
 0         0
 1        -3.46410161513775
 2        -1.20402649566579
 3        -3.14702399935583
 4       -0.764158513604989
 5        -3.33651095487791
 6        -1.06334809374951
 7        -3.22026651589781
 8       -0.899354829897603
 9        -3.29018337594631
10        -1.00335116241542
11        -3.24769495022932
12       -0.942427559304618
13        -3.27334189204985
14       -0.979957969007933
15        -3.25780063223344
16       -0.957508125140178
```

```
17        -3.26719544352513
18        -0.971182625129722
19        -3.26150804559577
```

The iteration gets stuck oscillating back and forth and fails.

Can we do anything about this? Well, we know from the stated theorems that divergence is caused by $|f'(x^\star)| > 1$. So we perform the following operations on the original equation:

$$x = \frac{1}{2}\sin x - 2\sqrt{|x+3|}$$

$$\frac{x}{4} = \frac{1}{8}\sin x - \frac{1}{2}\sqrt{|x+3|}$$

$$x = \frac{1}{8}\sin x - \frac{1}{2}\sqrt{|x+3|} + \frac{3}{4}x.$$

Let us now define $\tilde{f}(x) = \frac{1}{8}\sin x - \frac{1}{2}\sqrt{|x+3|} + \frac{3}{4}x$. The fixed point $x^\star = \tilde{f}(x^\star)$ will be the same, as can be seen from Fig. 3, but in comparing Figs. 3 and 2 we indeed see that the functions $\tilde{f}(x)$ and $f(x)$ are different. From Fig. 3, we see graphically that $x^\star = \tilde{f}(x^\star)$ satisfies $-2.5 < x^\star < -2$. To show that fixed-point iteration will converge when applied to $\tilde{f}(x)$, we establish the assumptions of **Theorem 2**. Now, for $x > -3$ we have

$$\tilde{f}'(x) = \frac{1}{8}\cos x - \frac{1}{4\sqrt{x+3}} + \frac{3}{4},$$

which is continuous on $(-3, \infty)$. For $x > -\frac{5}{2} > -3$ we then find

$$\begin{aligned}
|\tilde{f}'(x)| &\le \left|\frac{1}{8}\cos x\right| + \left|\frac{3}{4} - \frac{1}{4\sqrt{x+3}}\right| \\
&\le \frac{1}{8} + \frac{1}{4}\left|3 - \frac{1}{\sqrt{x+3}}\right| \\
&\le \frac{1}{8} + \frac{3}{4} \\
&= \frac{7}{8} \\
&< 1,
\end{aligned}$$

so that $|\tilde{f}'(x^\star)| < 1$, since $x^\star > -2.5$ (see Fig. 3). One inequality used $|3 - 1/\sqrt{x+3}| < 3$. To get this, note $x > -\frac{5}{2} \implies \sqrt{x+3} > \sqrt{\frac{1}{2}} \implies 0 < 1/\sqrt{|x+3|} < \sqrt{2}$. This implies

$$-\sqrt{2} < -\frac{1}{\sqrt{x+3}} < 0 \implies 3 - \sqrt{2} < 3 - \frac{1}{\sqrt{x+3}} < 3 \implies \left|3 - \frac{1}{\sqrt{x+3}}\right| = 3 - \frac{1}{\sqrt{x+3}} < 3.$$

We now write a function `lecroot2_fun3.m` that evaluates $\tilde{f}(x)$, and then try the following.

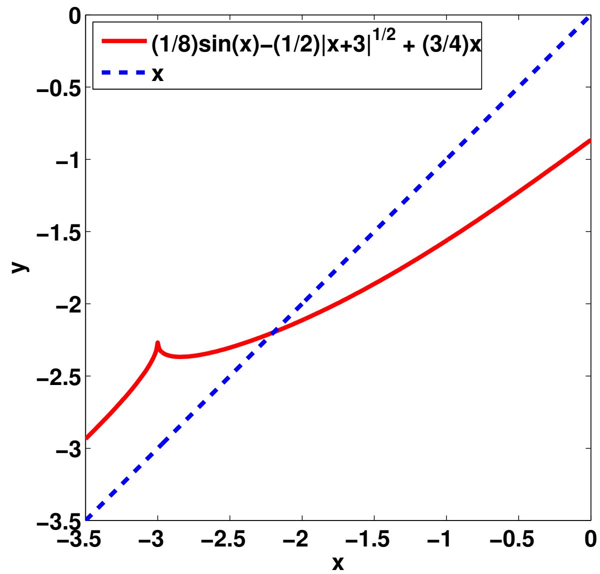

Figure 3: The functions x (the identity) and $\frac{1}{8}\sin x - \frac{1}{2}\sqrt{|x+3|} + \frac{3}{4}x$ near their common point of intersection, the location $x^{\star}$ we are trying to find.

```
>> x0 = 0; tol = 1e-8; kmax = 1e5;
>> iteration(@lecroot2_fun3,x0,tol,kmax)
Converged in 23 iterations with tol 1e-08
ans =
         -2.19713883798641
```

So the iteration converges as **Theorem 2** predicts.