

Lab Exercise No. 4 : Fixed Points and Iteration

OUT: Wed 18 Oct 2006

IN : Wed 25 Oct 2006

1 PURPOSE

The purpose of this exercise is to examine, analyse and determine various fixed-point iteration functions, using a combination of analysis (MAPLE) and experiment (MATLAB).

We saw in Chapters 3 and 4 that a fixed point of the function $T()$ is any x such that $x = T(x)$. We also saw that the sequence $x_0, T(x_0), T(T(x_0)), T(T(T(x_0))), \dots$ converges to a fixed point under certain conditions. A simple demonstration of these ideas can be done on any scientific calculator as follows : Enter any number $x > 0$ and then repeatedly press the $\sqrt{\quad}$ button. This will generate a sequence of numbers that converges to 1. This is because $T(x) = \sqrt{x}$ has a fixed point $x = \sqrt{x}$ whose solution is 1 (0 is also a fixed point).

Try the same steps for the $\cos(x)$ button and see does the sequence converge.

In Chapter 4 we saw that the equation $f(x) = 0$ could be solved by transforming it into fixed-point form $x = T(x)$ and then generating the sequence $x_0, T(x_0), T(T(x_0)), T(T(T(x_0))), \dots$ which, we hoped, would converge to a fixed point x of $T(x)$, and which, we hoped again, would be a solution of $f(x) = 0$.

2 TRANSFORMING $f(x) = 0$ into $x = T(x)$.

There are many ways to transform $f(x) = 0$ into $x = T(x)$. The simplest is $x = f(x) + x = T(x)$, but this rarely converges. Newton's method, $x = x - f(x)/f'(x)$, is often used and converges very quickly under the right conditions.

1. **Newton's Method for Square Root.** We wish to calculate \sqrt{a} , $a > 0$. Of necessity, any algorithm to do this must be iterative because \sqrt{a} is irrational in general. The first and crucial step in developing Newton's method for \sqrt{a} is to find a function $f(x)$ that has a zero at $x = \sqrt{a}$, i.e., $f(\sqrt{a}) = 0$. Alternatively, we wish to find $T(x)$ that has a fixed point at $x = \sqrt{a}$, i.e., $\sqrt{a} = T(\sqrt{a})$.

In this case it is fairly easy to see that the function $f(x) = x^2 - a$ has two zeros at $x = \sqrt{a}$ and $x = -\sqrt{a}$. Applying the general Newton formula to this function gives

$$x = x - \frac{x^2 - a}{2x} = \frac{2x^2 - x^2 + a}{2x} = \frac{1}{2} \left(x + \frac{a}{x} \right) = T(x).$$

This gives the iteration formula $x_{k+1} := (x_k + a/x_k)/2$. This uses addition and division only. We can replace the division with multiplication by using the reciprocal algorithm below.

2. **Newton's Method for Reciprocal.** We wish to calculate $1/a$, $a > 0$. This is a very primitive operation which is available 'in silicon' on all floating point processors¹ Nonetheless these processors must have a hardware algorithm to perform this operation. Most use a combination of methods and Newton is one of these.

In this case we have to be careful in finding a function $f(x)$ that has a zero at $x = 1/a$. The function $f(x) = x - 1/a$ obviously has a zero at $x = 1/a$, but is silly because it assumes we have $1/a$. Instead we use $f(x) = 1/x - a$ which has a zero at $x = 1/a$. Applying the general Newton formula to this function gives

$$x = x - \frac{(1/x - a)}{(-1/x^2)} = x + (x - ax^2) = x(2 - ax) = T(x).$$

This gives the iteration formula $x_{k+1} := x_k(2 - ax_k)$. This uses addition and multiplication only.

3 EXERCISE

3.1 Part 1

Write MATLAB functions for each of the methods above and determine experimentally over what range of starting point x_0 they converge.

Analyse mathematically each method to determine the conditions under which they converge. Plotting the functions $f(x)$ or $T(x)$ and the Newton iterations may help here. The results of your experiments should help also.

3.2 Part 2

Transform, in any way you wish, the following equations into fixed point form and find their fixed points.

1. $f_1(x) = x - e^{-x} = 0$. (1 fixed point)
2. $f_2(x) = x^2 - \cos(x) = 0$. (2 fixed points)
3. $f_3(x) = x^3 - x - 1 = 0$. (1 fixed point)
4. $f_4(x) = x^2 - 2^x = 0$. (3 fixed points)
5. $f_5(x) = x^{10} - 2^x = 0$. (3 fixed points)

4 SOLUTION

4.1 Part 1: Newton's Method for Reciprocal & Square Root

.

See *Calculating the Reciprocal and Square Root* on the class webpage

<http://www.derekconnor.net/NA/Notes/RecSqRoot.pdf>

¹Many embedded processors used in washing machines, cars, etc., may not have this operation. Most do not have floating point arithmetic because it is too expensive in terms of 'silicon'.

4.2 Part 2: Fixed Points

The simplest way to find the zeros of a function is to plot it roughly and see where it changes sign. Even this crude method requires the user to supply some knowledge — the plot range. Notice that we have two plots for $f_5(x) = 10^x - 2^x$ because the first plot would have missed the zero out at about $x \approx 60$, while the second plot would have missed the zeros at $x \approx -1, 1$

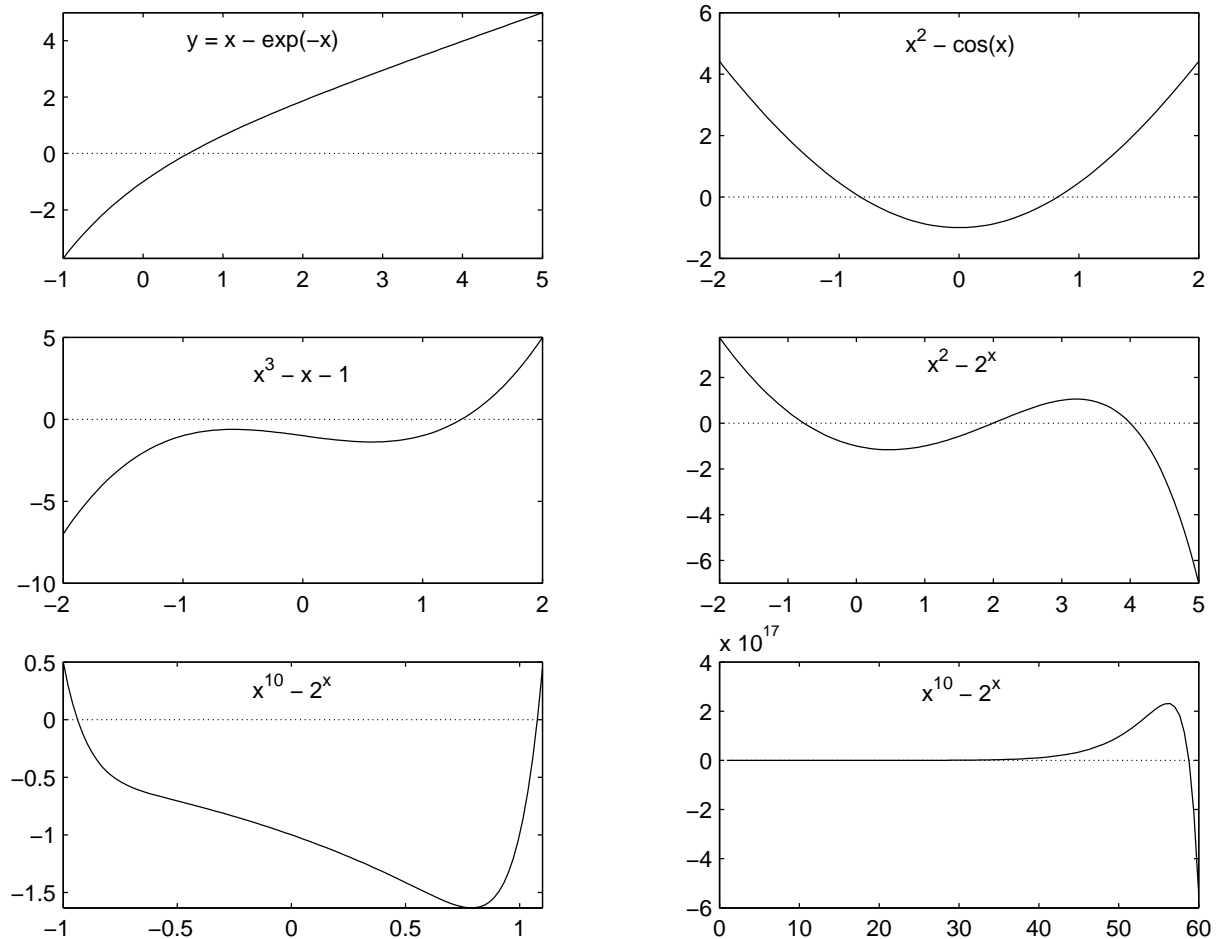


Figure 1: Zeros of Functions

Having plotted the functions, we should analyse each function to obtain more information. There is no theory that tells us how many zeros a general function has, unlike polynomials.

4.2.1 Transforming to Fixed Point Form

There are many ways to transform $f(x) = 0$ into fixed point form $x = T(x)$. The simplest way, $x = x + f(x)$, rarely works, as we see below.

1. The first function $f_1(x) = x - e^x$ has a zero at $x = 0.56714329040978$, calculated by MATLAB's `fzero(f,x0)` function. The fixed point iteration of $T_1(x) = x + x - e^x$ diverges to $\pm\infty$ no matter

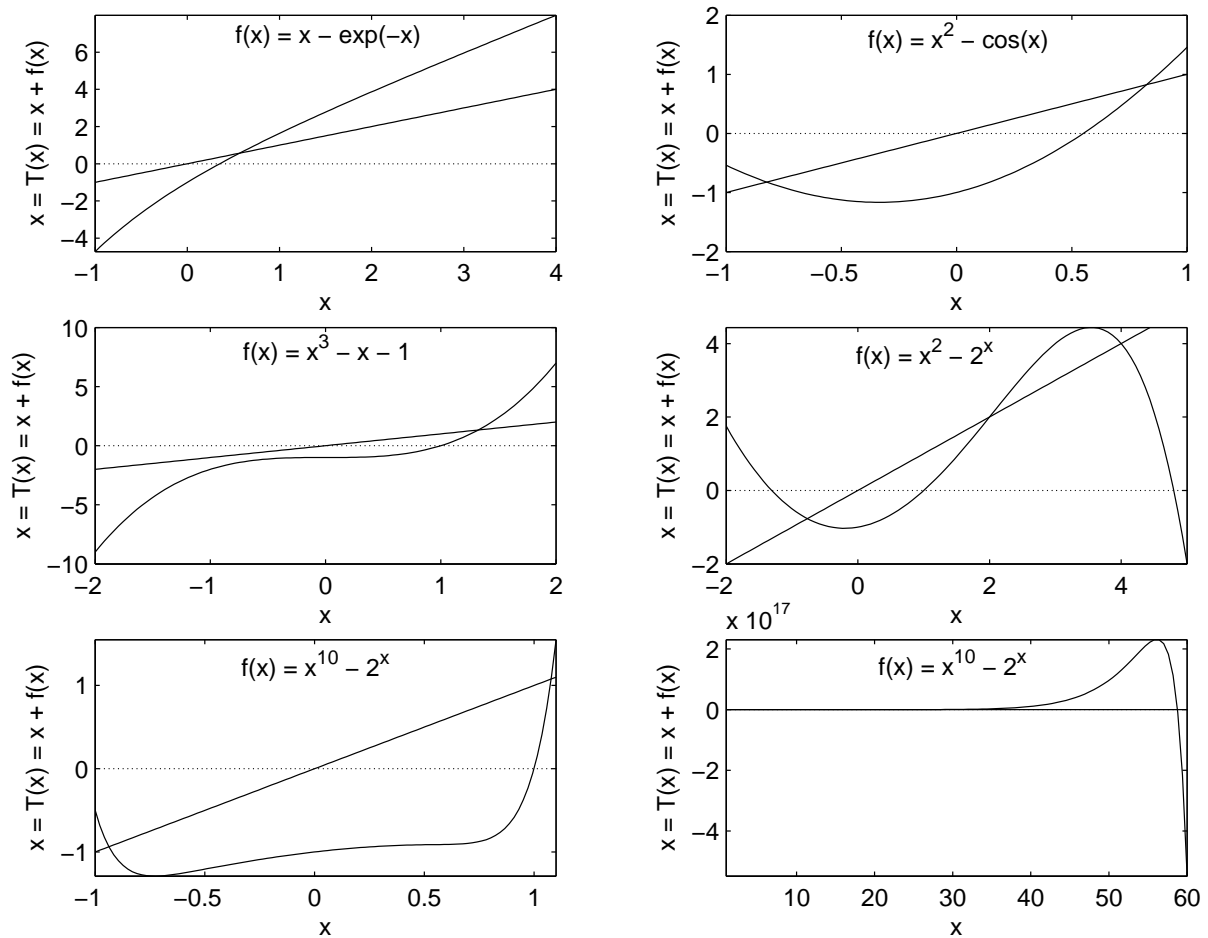


Figure 2: The Fixed Points of Functions

where we start. This is called a *repulsive fixed point*. The only way for $x_{k+1} := T_1(x_k)$ to converge is to start at the fixed point, which is impossible in finite-precision arithmetic. Even starting with MATLAB'S value $x = 0.56714329040978$ causes divergence. Try it.

2. The second function $f_2(x) = x^2 - \cos(x)$ has zeros at $x = \pm 0.82413231230252$, calculated by MATLAB'S `fzero(f,x0)` function. Starting the fixed point iterations at $x_0 = 0.82413231230252$ gives the strange behavior shown in Figure 3
3. The fourth function $f_4(x) = x^2 - 2^x$ has zeros at $x = -0.76666469596212, 2, 4$, calculated by MATLAB'S `fzero(f,x0)` function. The fixed point $x = 2$ is repulsive and $x_{k+1} := T_1(x_k)$ converges for $x = 2$ only because 2 is exactly representable in IEEE arithmetic. This iteration function is very unstable. Starting at $x_0 = 3.0$ it converges immediately to $x = 4$. This is because $T_4(3) = 3 + 3^2 - 2^3 = 4$, which is calculated exactly in IEEE arithmetic. Then $T_4(4) = 4 + 4^2 - 2^4 = 4$, which is exact also. However, if we start at $x_0 = 3.0 + 3.0 * \epsilon_m = 3.0000000000000001$, we get instability after about 40 iterations, as shown in Figure 4.

Can you tell from Figure 5, for what starting values Newton's method converges to the fixed points near -1.0 and 1.0 ?

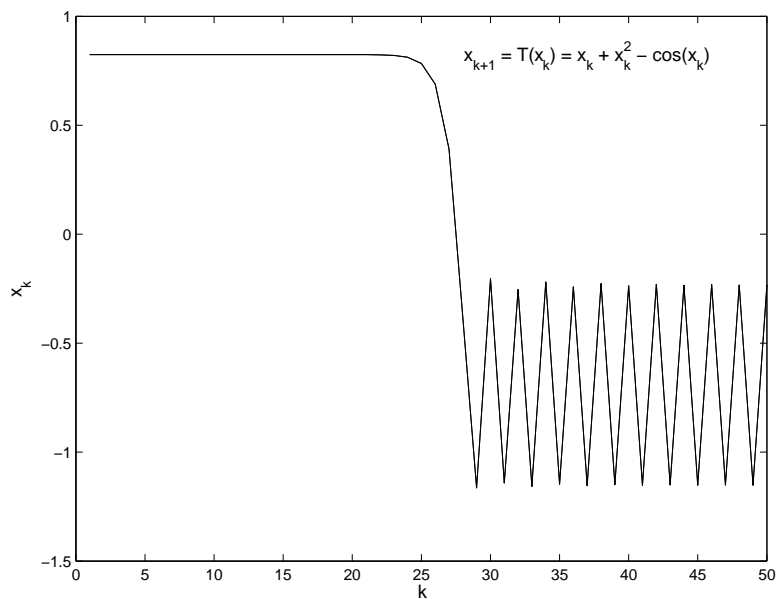


Figure 3: Strange Behavior of $T_2(x) = x + x^2 - \cos(x)$

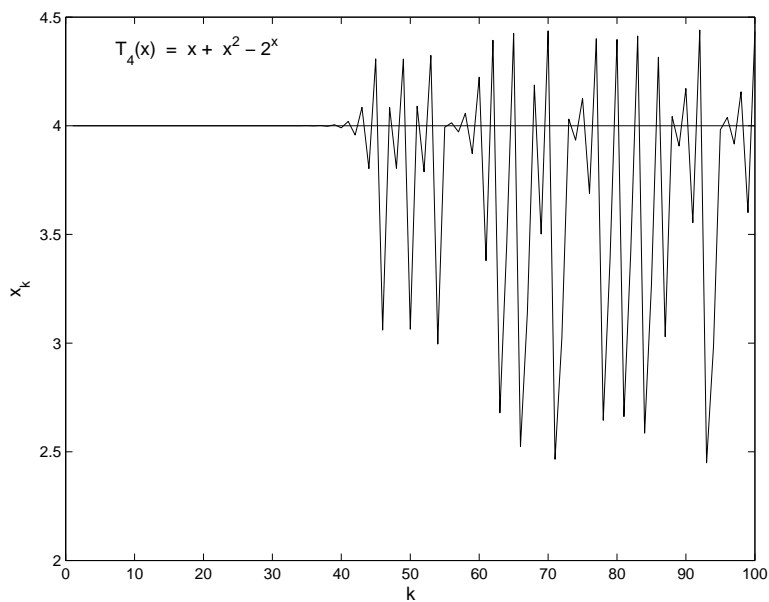


Figure 4: Chaotic Behavior of $T_4(x) = x + x^2 - 2^x$

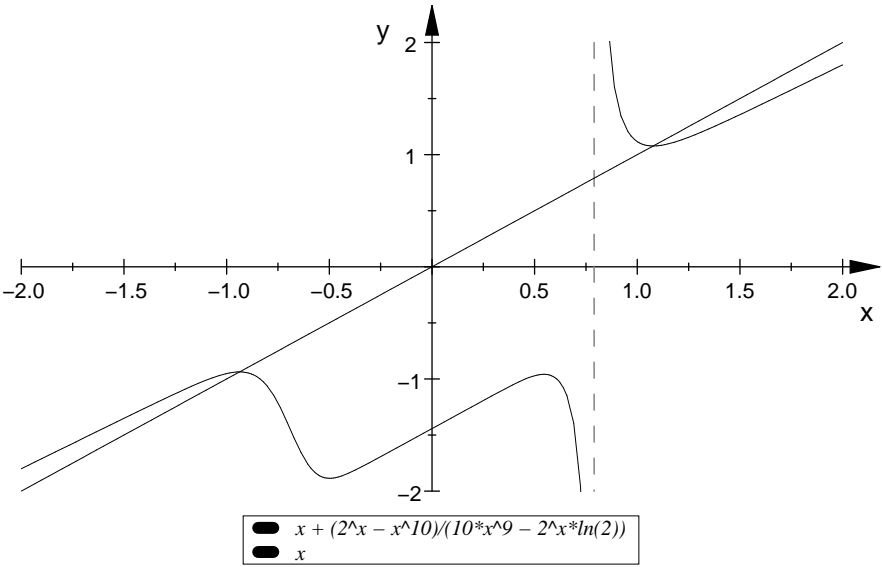


Figure 5: Fixed points of Newton's Method :