**Practical 1:**

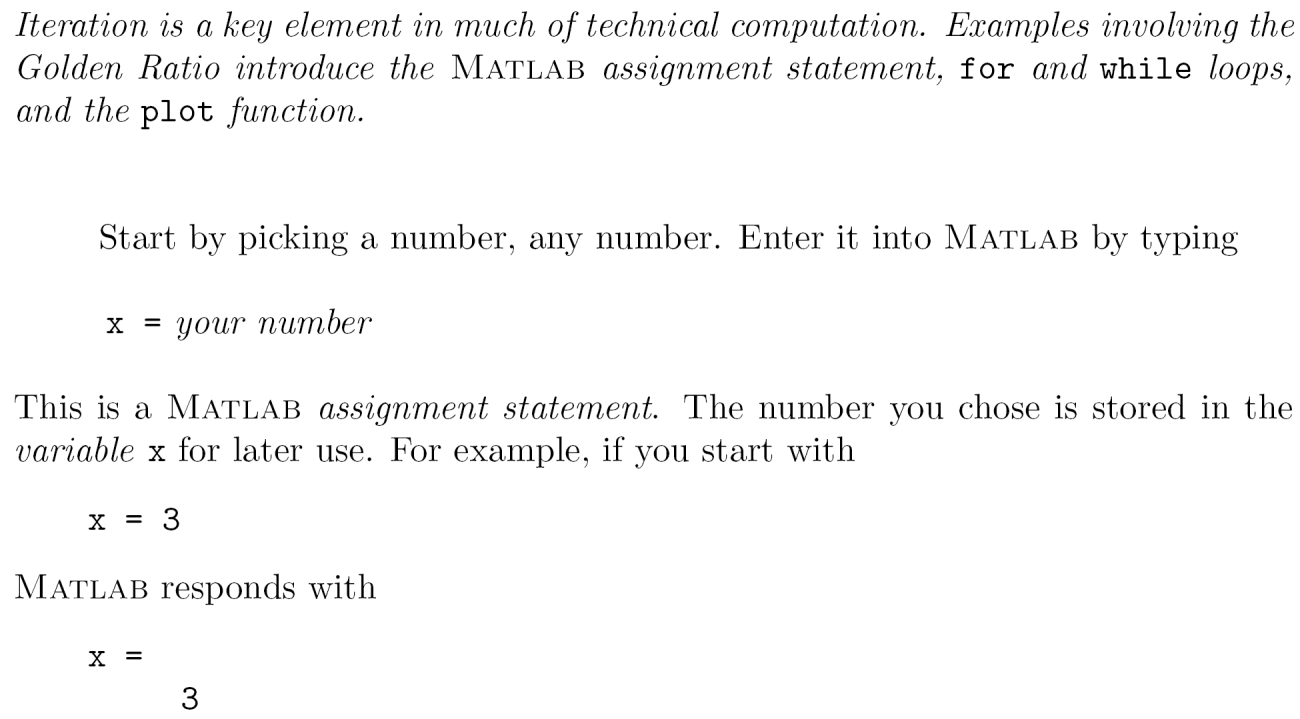
**Objective**: Computer Algorithm for Iterative Calculations.

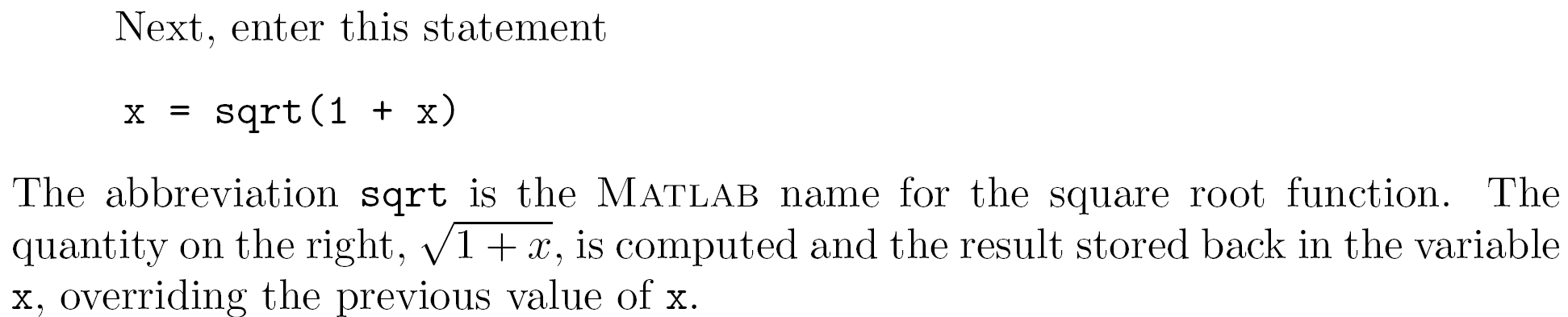
In [computational mathematics](https://en.wikipedia.org/wiki/Computational_mathematics), an iterative method is a mathematical procedure that generates a sequence of improving approximate solutions for a class of problems. A specific implementation of an iterative method, including the [termination](https://en.wikipedia.org/wiki/Algorithm#Termination) criteria, is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) of the iterative method. An iterative method is called convergent if the corresponding sequence converges for given initial approximations

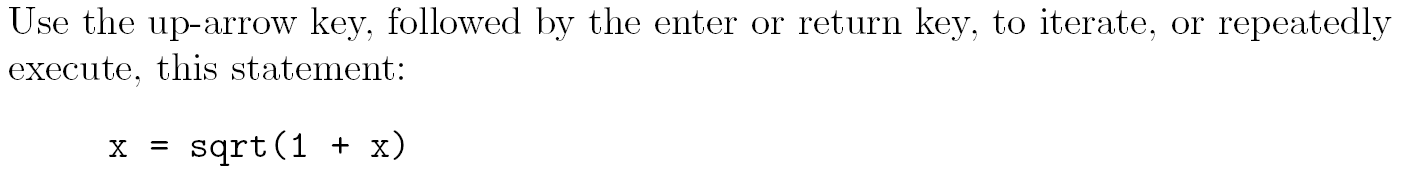
**Part A:  Golden Ratio Computation**

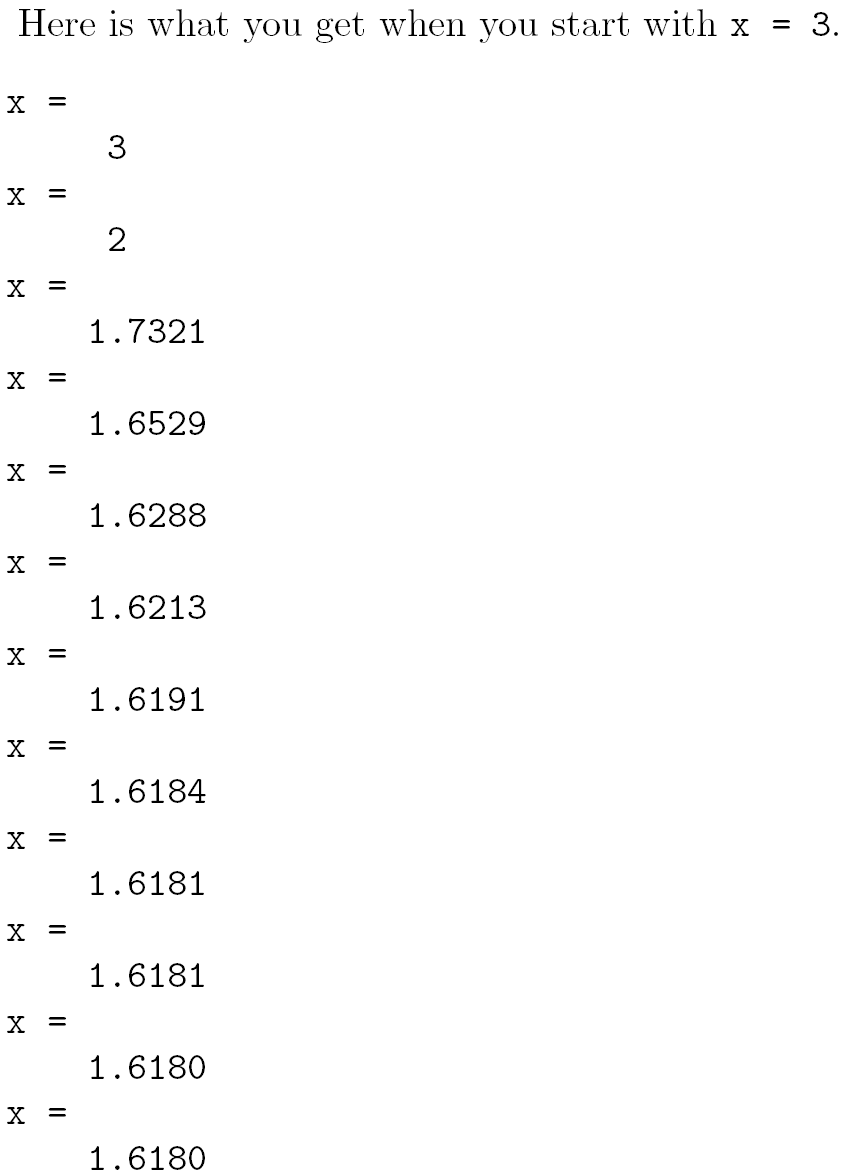
**Instruction:**

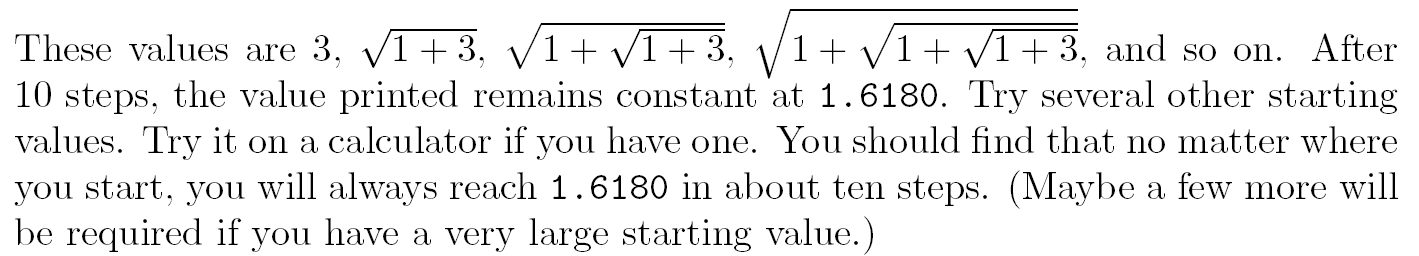
1. Open <http://octave-online.net/> and create an account.

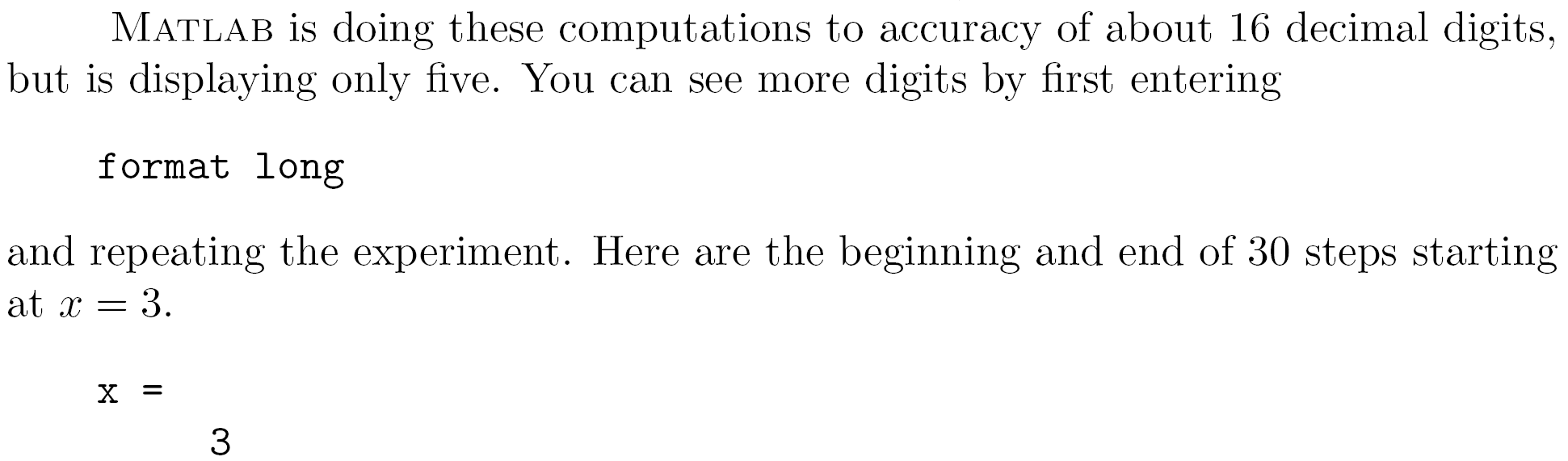


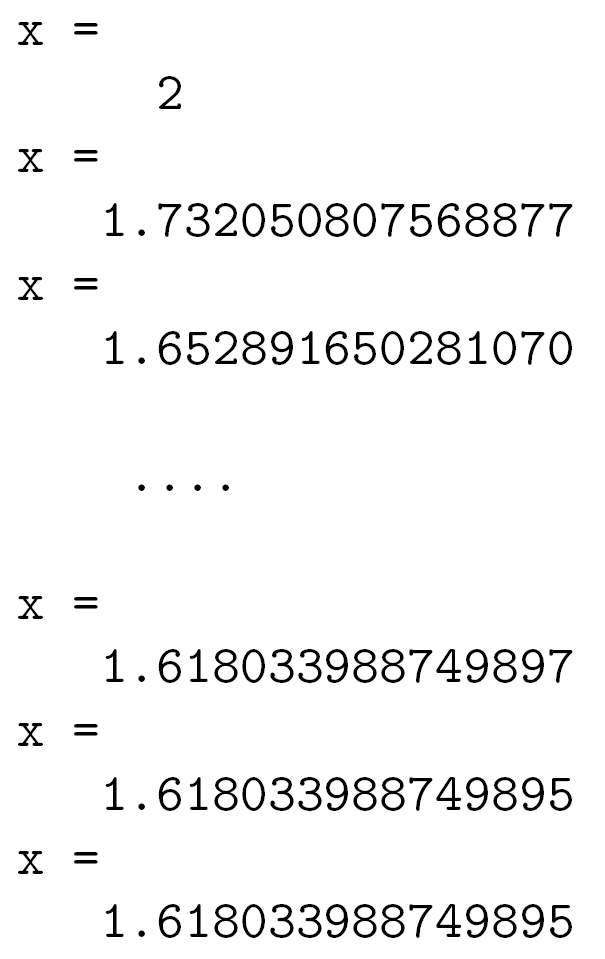




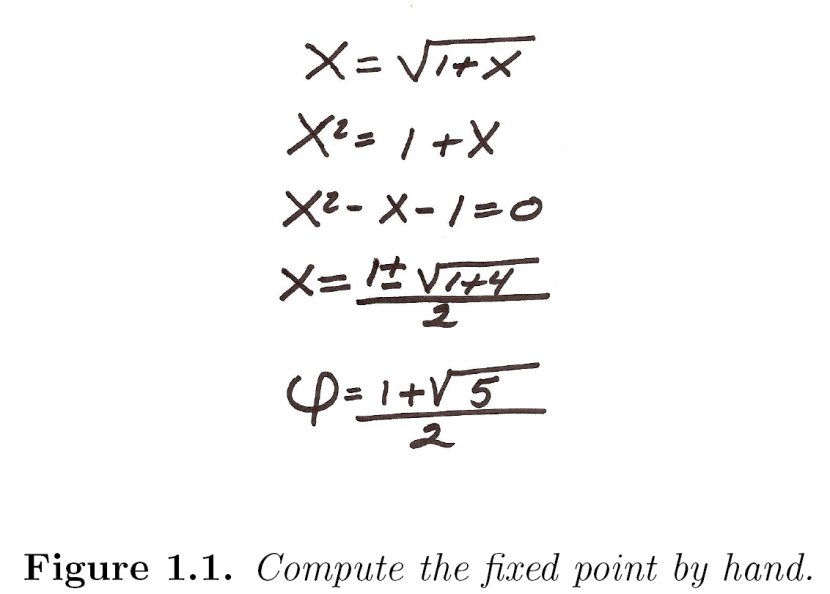


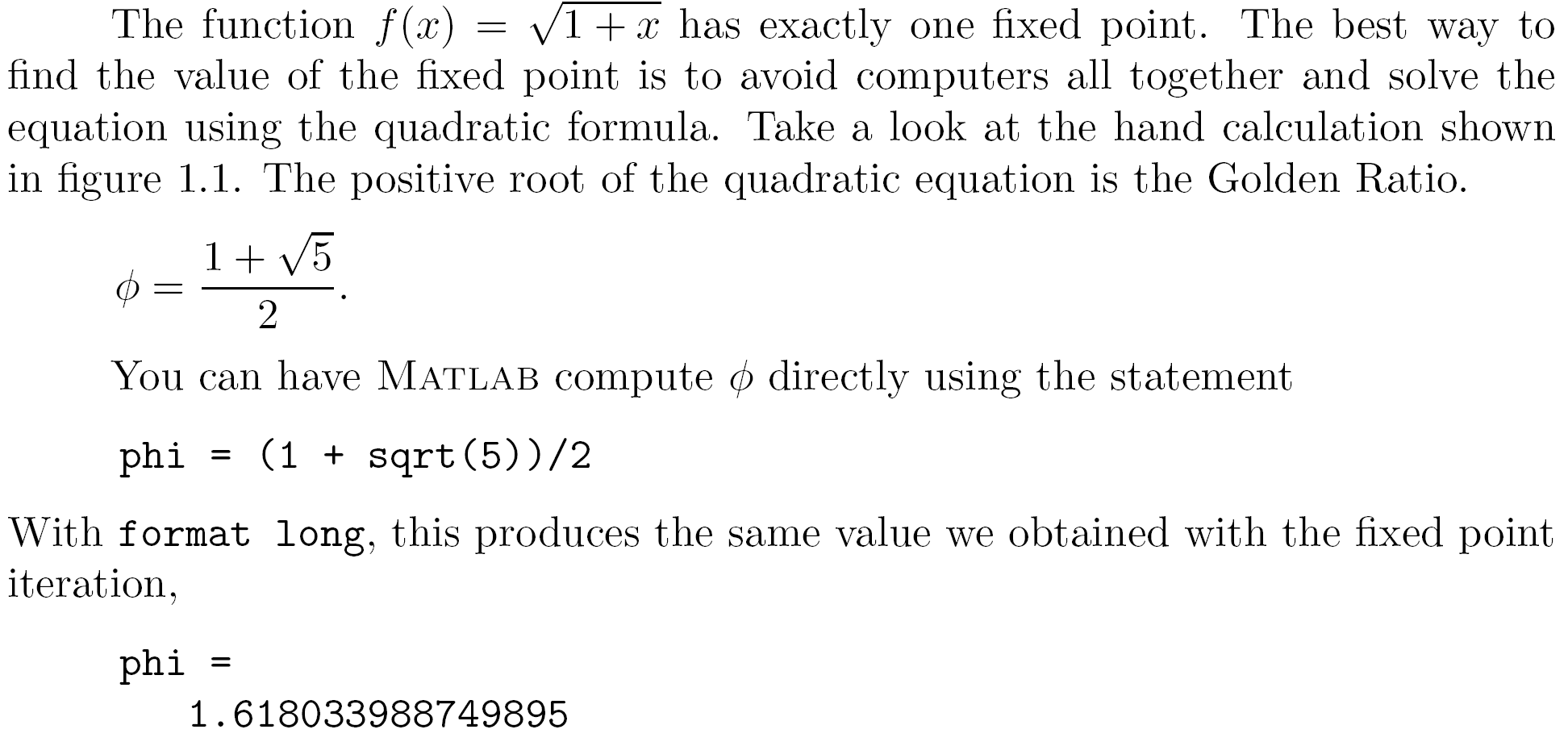


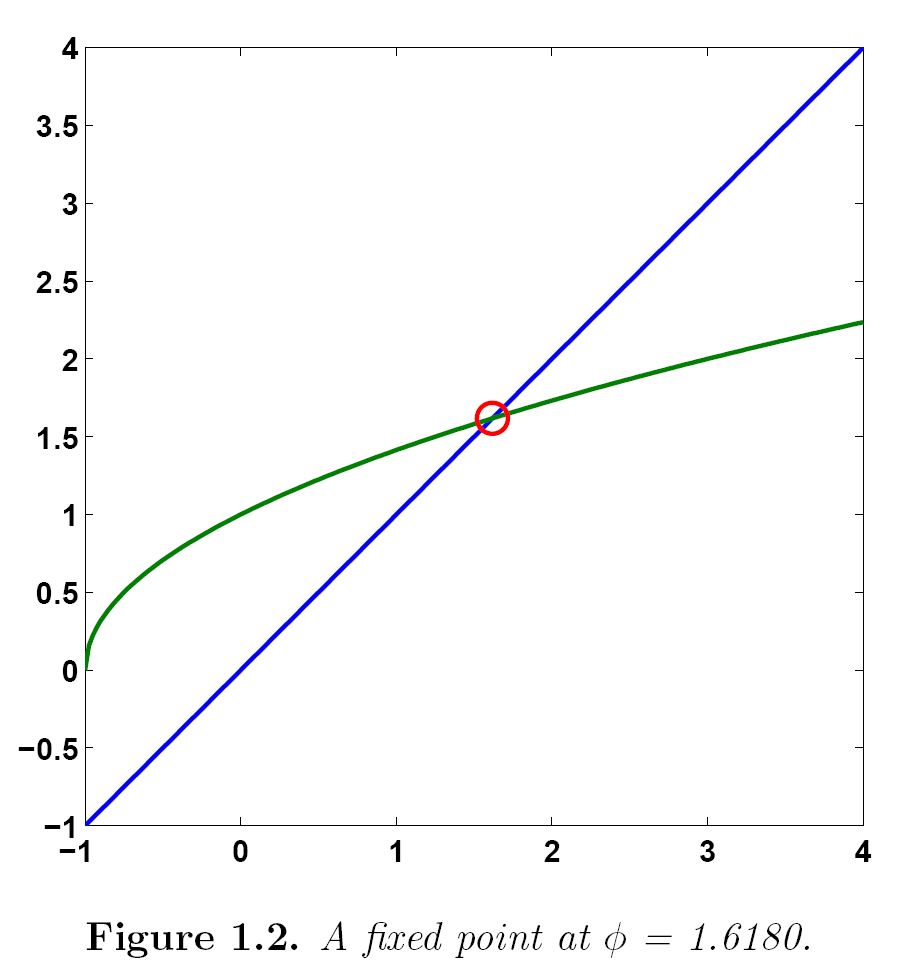




**Golden Ratio，**







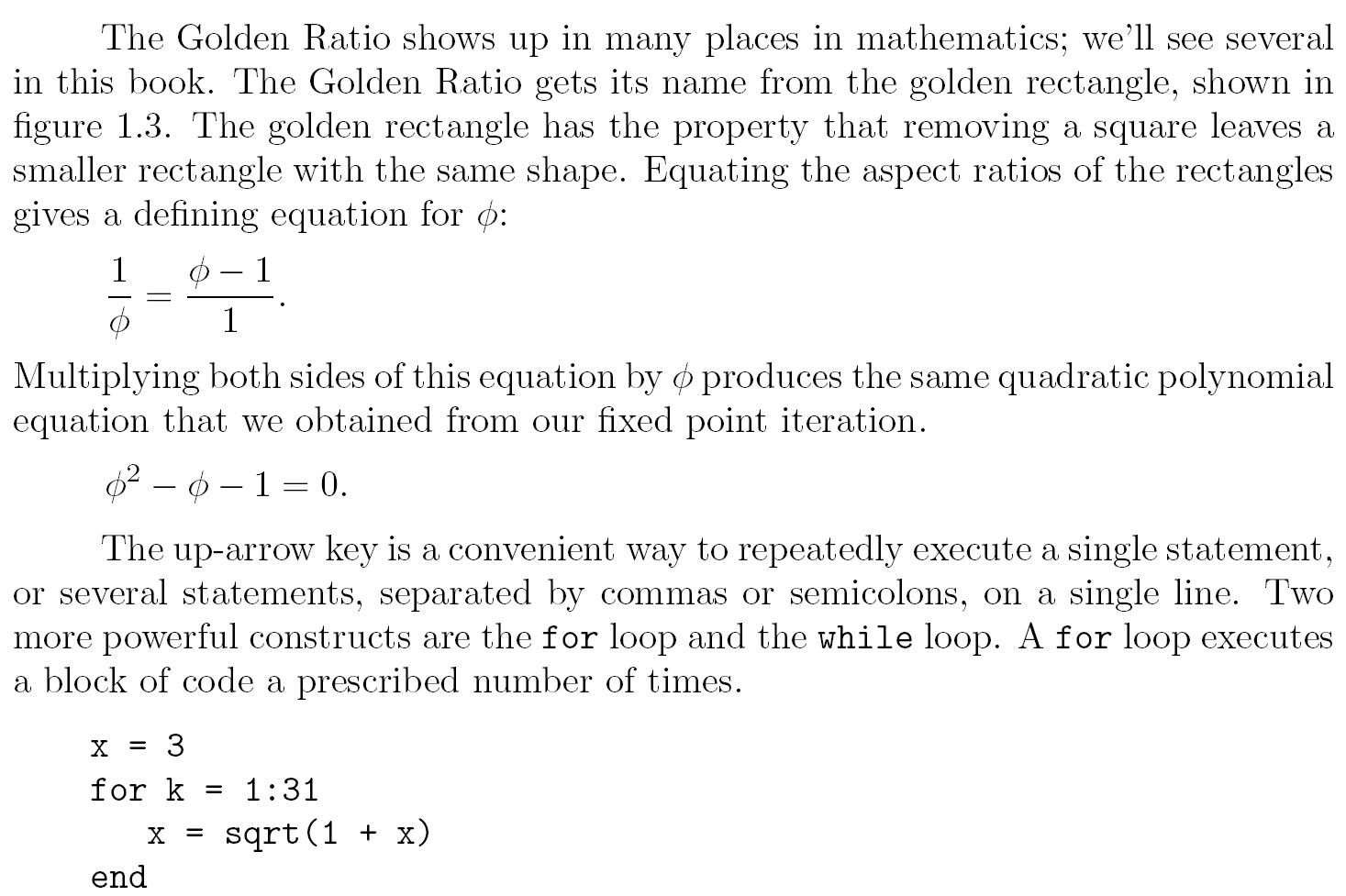
x = -1:.02:4;

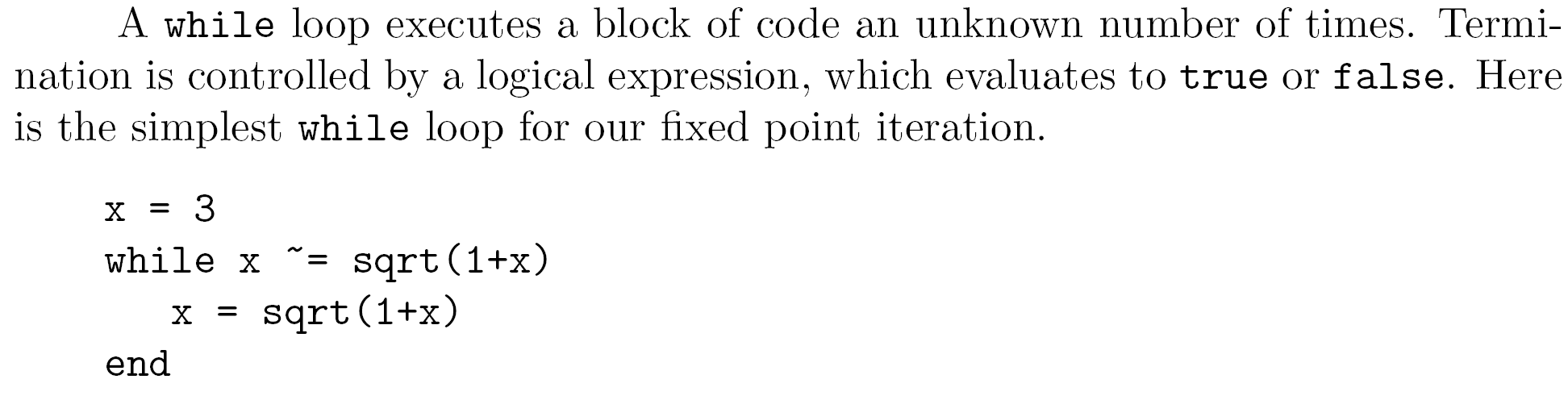
y1 = x;

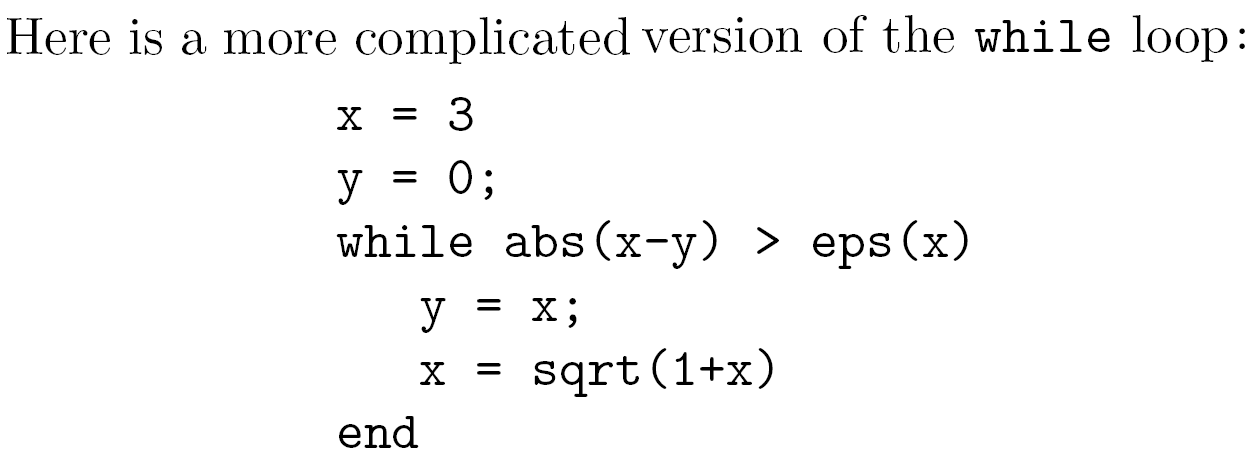
y2 = sqrt(1+x);

plot(x,y1,'-',x,y2,'-',phi,phi,'o')

**Extra Notes:**

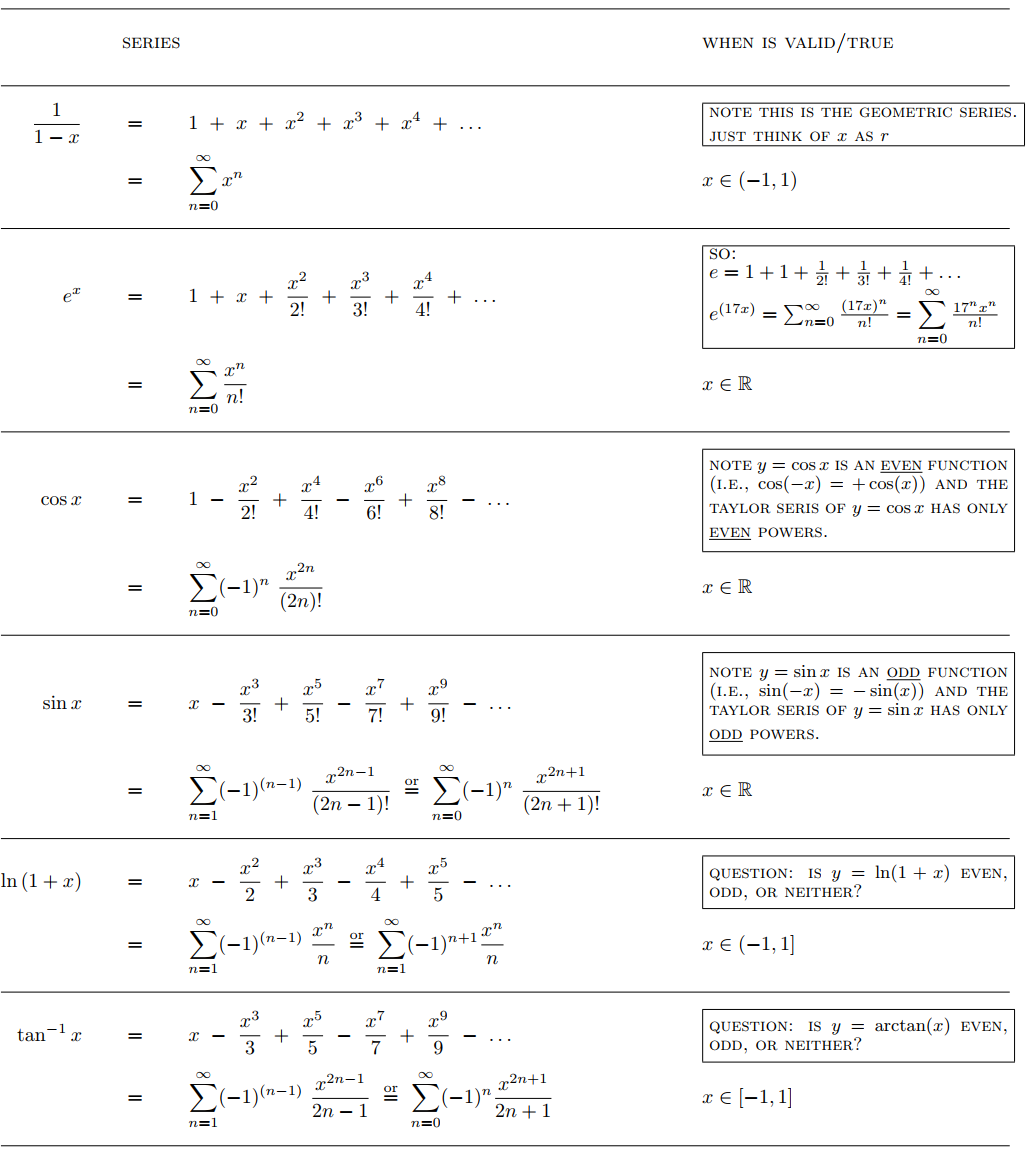




****

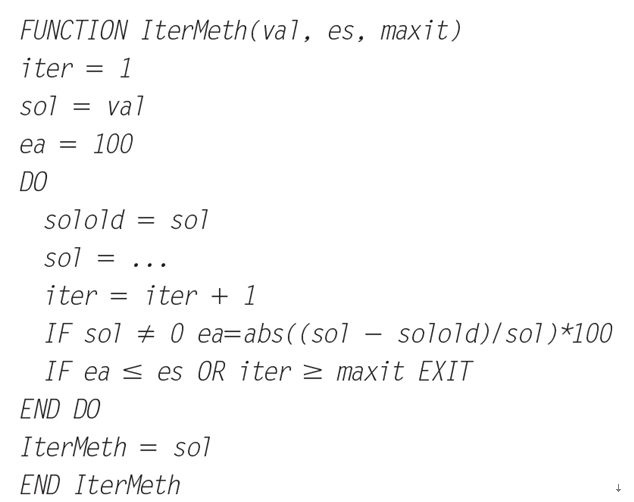
**Part B: Interactive Computations**

**Commonly Used Taylor Series**



**Problem 1: Interactive Method for Exponential Function**

1. There are two basic flavors to implement iterative solution:
   1. count-controlled and
   2. decision loops.
2. Most iterative solutions use decision loops. Thus, rather than employing a pre-specified number of iterations, the process typically is repeated until an approximate error estimate falls below a stopping criterion.
3. Sample pseudocode for a generic iterative calculation:



The function is passed a value (***val***) along with a stopping error criterion (***es***) and a maximum allowable number of iterations (***maxit***). The value is typically either (1) an initial value or (2) the value for which the iterative calculation is to be made. The function first initializes three variables:

1. A variable ***iter*** that keeps track of the number of iterations.
2. A variable ***sol*** that holds the current estimate of the solution.
3. A variable ***ea*** that holds the approximate percent relative error (Note that ***ea*** is initially set to a value of 100 to ensure that the loop executes at least once).

**Instruction:**

1. Open <http://octave-online.net/> and create an account.
2. Create a new file (e.g., IterMeth.m)
3. Develop a computer program based on the pseudocode from ③.

function[v,ea,iter]=IterMeth(x,es,maxit)

% initialization

iter=1;

sol=1;

ea=100;

%iterative calculation

while(1)

solold=sol;

sol=sol+x^iter/factorial(iter);

iter=iter+1;

if sol~=0

ea=abs((sol-solold)/sol)\*100;

%disp(ea);

end

%disp(solold);

%disp(sol)

if ea<=es|iter>=maxit,break,end

end

v=sol;

end

**Sample Output:**

**octave:>** format long

**octave:>** [val,ea,iter]=IterMeth(1,1e-6,100)

val = 2.71828182619849

ea = 9.21615564152297e-07

iter = 12

**Problem 2: Maclaurin Series Approximation**

1. Estimate using . Add terms one at a time until the absolute value of the approximate error estimate falls below an error criterion conforming to two significant figures.
2. Write a program to calculate the approximation using a value of Determine the number of terms necessary to approximate to 8 significant figures.

**Sample Output:**

j= 1 cos(x)= 1.0000000000

j= 2 cos(x)= 0.5558678020 ea = 8.0e-01

j= 3 cos(x)= 0.5887433702 ea = 5.6e-02

j= 4 cos(x)= 0.5877699636 ea = 1.7e-03

j= 5 cos(x)= 0.5877854037 ea = 2.6e-05

j= 6 cos(x)= 0.5877852513 ea = 2.6e-07

j= 7 cos(x)= 0.5877852523 ea = 1.7e-09

**Problem 3: Taylor Series Approximation (based on Homework 1)**

1. **Sample output for Q1**
2. **Sample output for Q2**