Matt McDade

Applied Numerical Methods HW 2

**Problem 1:**

For the bisection method, the number of iterations to guarantee finding a root is equal to:

Where is the number if iterations, and are the bounds, and is the tolerance. So we would have

This means a minimum of **28** iterations is needed to guarantee finding a root between 1 and 3 for a tolerance of 10-8

**Problem 2:**

|  |  |
| --- | --- |
| CODE  f = @(x) cos(x) + (1 / (x^3 + 200));  a = -5;  b = 5;  tol = 1 \* 10 .^ -10  % 4 roots can be seen in the plot  figure(1)  fplot(f)  [r1, i1] = bisect(f, -5, -3, tol);  [r2, i2] = bisect(f, -3, -1, tol);  [r3, i3] = bisect(f, 1, 3, tol);  [r4, i4] = bisect(f, 4, 5, tol);  fprintf('Root 1: %.8f in %d iterations\n', r1, i1)  fprintf('Root 2: %.8f in %d iterations\n', r2, i2)  fprintf('Root 3: %.8f in %d iterations\n', r3, i3)  fprintf('Root 4: %.8f in %d iterations\n', r4, i4) | OUTPUT  Figure 1:    Root 1: -4.70197711 in 35 iterations  Root 2: -1.57589614 in 35 iterations  Root 3: 1.57570042 in 35 iterations  Root 4: 4.70910412 in 34 iterations |

**Problem 3:**

I plugged both equations into a MATLAB fixed-point function and set the max iterations to 10000. The first equation did not diverge and reached the root in 16 iterations. The second equation was not able to find the root after those 10000 iterations, meaning it diverges with the fixed-point method. This is because the second equation uses an exponential with x in it, which will bounce back and forth using the fixed-point method but never actually converge on the root.

**Problem 4:**

In the bisection method, to avoid c not being set, I set it to ‘0’ in this example if not root was found in the function. So it diverging/not finding a root in the number of max iterations is represented as a 0 in these figures, and the real root can be seen on the first function.

|  |  |
| --- | --- |
| CODE  g1 = @(x) (5 - log(x)) / 3;  g2 = @(x) exp(-3 \* x + 5);  x0 = 1;  tol = 10^(-9);  c1 = zeros(1, 20);  c2 = zeros(1, 20);    for max1 = 1:20  c1(max1) = fixedpoint(g1, x0, tol, max1);  c2(max1) = fixedpoint(g2, x0, tol, max1);  end  fprintf('Root: %.9f\n', c1(18))  figure(1); plot(c1)  figure(2); plot(c2) | OUTPUT  Root: 1.525822197  Figure 1:    Figure 2: |

**Problem 5:**

|  |  |
| --- | --- |
| CODE  f = @(x) x^3 - (0.001 \* x^2) + x - 0.001;  fp = @(x) 3\*x^2 - 0.002 \* x + 1;  x0 = 50;  tol = 10^(-10);  maxiter = 20;  [root, iter] = newton(f, fp, x0, tol, maxiter);  fprintf('Root: %.8f found in %d iterations\n', root(maxiter), iter) | OUTPUT  Root: 0.00100000 found in 15 iterations |

**Problem 6:**

|  |  |
| --- | --- |
| CODE  f = @(x) x^3 - (0.001 \* x^2) + x - 0.001;  x0 = 50;  x1 = 49;  tol = 10^(-10);  maxiter = 20;  [root, iter] = secant(f, x0, x1, tol, maxiter);  fprintf('Root: %.8f found in %d iterations\n', root(maxiter), iter) | OUTPUT  Root: 0.00100000 found in 15 iterations |