Lab #4 (not from Lab Manual)

Adam Jaso
09/27/2011

Determination of Zeroes by the Newton-Raphson and Bisection Algorithms

Objective

The objective of this exercise is to use the Newton-Raphson and Bisection methods to find the zeroes of a function. The common element between these methods is their evaluation of the function at strategic values and recursively predicting the next value for evaluation until the evaluation is less than a given error margin. This exercise solves the equation x = cos(x) by searching for the zero of f(x) = x – cos(x) = 0.

Procedure

BISECTION ALGORITHM

1. Define the appropriate initial values for the algorithm.
   a. Define an interval containing the zero of the function by setting appropriate minimum and maximum values.
   b. Find the midpoint value of this interval.
   c. Define the error margin of your desired solution.
2. Recursively evaluate the function until the absolute values of these evaluations are less than the above defined margin.
   a. Check the evaluation of the function at the maximum and mid values for the condition of f(max) > error and f(mid) < error or f(max) < error and f(mid) > error
   b. If the evaluation of the above condition is false, then the other interval from the mid point to the minimum value is the interval containing the zero of the function (provided you have properly defined your interval).
3. Once the above loop exits, return the value of the interval mid point as the zero of the function.

NEWTON-RAPHSON ALGORITHM

1. Define the appropriate initial values for the algorithm.
   a. Set an initial value for evaluation that will be corrected toward the zero of the function.
   b. Calculate the first correction to the evaluation.
   c. Define the error margin of your desired solution.
2. Recursively evaluate the function until the absolute value of the evaluation is less than the above defined error margin.
   a. Correct the current value for evaluation by the Newton correction: $\Delta x_n = -f(x_{(n-1)})/f'(x_{(n-1)})$. And return the initial condition to evaluate this corrected value.
3. Once the above loop exits, return the value of the corrected value.

Code / Results

BISECTION ALGORITHM

```
%bisection.m
%Solution of the equation x - cos(x) = 0 by the bisection method.

%Set the initial conditions
lo = .01*pi;
hi = pi/2;
m = (hi+lo)/2;
err = 0.000000001;
iter = 0;

%Recursively evaluate the function until it returns a value less than the
```

```
%error margin
while abs(xcosx(hi)) > err && abs(xcosx(m)) > err && abs(xcosx(lo)) > err
    if xcosx(hi)*xcosx(m) < 0
        lo = m;
        m = (hi+lo)/2;
    else
        hi = m;
        m = (hi+lo)/2;
    end
    iter = iter+1;
end

fprintf('zero = %f\niterations = %d\n',m,iter);
```

BISECTION RESULT

```
>> bisection
zero = 0.739085133
iterations = 28
```

NEWTON-RAPHSON ALGORITHM

```
%newton.m
%Solution of x - cos(x) = 0 by the Newton-Raphson method.

%Set the initial conditions
x = .1*pi;
dx = -(xcosx(x))/(1+sin(x));
err = 0.000000001;
iter = 0;

%Loop through the function value
while abs(xcosx(x)) > err
    x = x + dx;
    dx = -(xcosx(x))/(1+sin(x));
    iter = iter+1;
end

%Return the zero of the function and the number of iterations completed
fprintf('zero = %f\niterations = %d\n',x,iter);
```

NEWTON-RAPHSON RESULT

```
>> newton
zero = 0.739085133
iterations = 4
```

FUNCTION DEFINITION

```
function xcosx=xcosx(x)
xcosx=x-cos(x);
end
```

Observations

The Newton-Raphson algorithm with an error margin of 1x10^-9 gave the solution as 0.739085133. For an accuracy of 1 in 10,000,000, the Newton-Raphson method used 4 iterations and the Bisection method used 23. Again for an accuracy of 1 in 1,000,000,000, the Newton-Raphson method used 4 iterations and the Bisection method used 28. In summary the Newton-Raphson method is much more efficient than the Bisection method.