Lab #6 (not in book)

Adam Jaso
10/11/2011

Finding Zeros with Errors Newton-Raphson
And
Integrating *sin(x)* with Errors with the Composite Trapezoidal Method

Objective

This exercise uses the Newton-Raphson approximation method and composite trapezoidal integration method to find the solutions to $x = \cos(x)$ and *INTEGRAL* of $\sin(x)$ over an interval from a to b, respectively. The respective errors analyses for the approximations mentioned above will be supplied.

Procedure

To solve the equation $x = \cos(x)$, the equation may be rewritten as $x - \cos(x) = 0$. The Newton-Raphson method uses a recursive correction equation, $x_{i+1} = x_i - f(x_i)/f'(x_i)$, to find the zero of the function $f(x) = x - \cos(x)$. The error estimate for this method has been shown to be $\varepsilon_{i+1} = -1/2 * \varepsilon_i\hat{}2 * f''(x_i)/f'(x_i)$ where $\varepsilon_i = f(x_i)/f'(x_i)$. These formulae were used in the following procedure to calculate the solution.

IMPLEMENTATION of NEWTON-RAPHSON APPROXIMATION METHOD

1. Set the initial conditions.
   a. The initial x value was chosen at slightly greater than 0 in order to allow the derivative to effectively correct the initial x value toward the function's zero.
   b. Calculate the first correction using the correction equation given above.
   c. Find the first and second derivatives of the function (x – cos(x) = 0) to be used in calculating the approximation error.
   d. Calculate the error from the error equation given above.
   e. Set the iteration counter variable to zero.
   f. Define the error tolerance for this approximation.
2. Successively evaluate the function at the corrected x value until the evaluation returns less than the above defined tolerance.
   a. Calculate the next x value and the next correction.
   b. Calculate the first and second derivatives in order to calculate the error correction.
   c. Report the iteration value and its associated error.
   d. Count the iteration.
3. Display the approximated zero with its associated error.

To evaluate the integral of *sin(x)*, the trapezoidal method was used with successive *sets* of iterations from 2 to 2^n divisions where 0 < n <= 20. The correction formula for this calculation is ERROR = $\Delta x\hat{}2/12 * (f_o' - f_N') - \Delta x\hat{}4/720 * (f_o''' - f_N''')$ where Δx is the division width.

IMPLEMENTATION of COMPOSITE TRAPEZOIDAL INTEGRATION of sin(x)

1. Calculate the x values for every increment.
2. Calculate the y values from sin(x) at every increment.
3. Calculate the initial increment.
4. Calculate the first integral evaluation.
5. Calculate the first and third derivatives at the interval start and interval end respectively.
6. Calculate the error coefficients
7. Set the initial conditions of the iterative loop.
   a. Set the initial number of interval divisions.
   b. Set a number of evenly spaced x values to be evaluated.
   c. Calculate the first and third derivatives of sin(x) evaluated at the endpoints.

d. Calculate the coefficients of the first and third derivatives used in the error calculation.

8. Start a loop which starts at *n = 1* and runs until *n > 2^20*, successively approximating the integral for *n* number of divisions.
   a. Find the sum of areas of the *n* trapezoids.
   b. Calculate the error of this approximation of *n* trapezoids.
   c. Display the approximation with its associated error.
   d. Increment *n* by a factor of 2 and set the other variables accordingly for the next iteration over *n*.
   e.

Code / Results

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

%Set the initial conditions

%Correction
x = .1*pi;
dx = -(xcosx(x))/(1+sin(x));
%Derivatives
dfdx=1+sin(x);
d2fdx= cos(x);
%Error Calculation
error=-1/2*dx^2*d2fdx/dfdx; %=x-(x+dx)
%Tolerance
err = 0.000000001;
iter = 0;


%Loop through the function value
while abs(xcosx(x)) > err
    %Correction
    x = x + dx;
    dx = -(xcosx(x))/(1+sin(x));
    %Derivatives
    dfdx=1+sin(x);
    d2fdx= cos(x);
    %Error Calculation
    error=-1/2*dx^2*d2fdx/dfdx;
    fprintf('n = %d\tError = %1.40f\n',iter,error);
    %Iteration
    iter = iter+1;
end

%Return the zero of the function and the number of iterations completed
fprintf('zero = %1.9f +/- %1.40f\niterations = %d\n',x,abs(error),iter);

*****     XCOSX.M FUNCTION DEFINITION     *****
function xcosx=xcosx(x)
xcosx=x-cos(x);
end

*****     NEWTON.M RESULTS     *****:
>> newton
n = 0 Error = -0.0007499369577627851900000000000000000000
n = 1 Error = -0.0000001360392225721950000000000000000000
n = 2 Error = -0.0000000000000040911599444233222000000000
n = 3 Error = -0.0000000000000000000000000000038565739482
zero = 0.739085133 +/- 0.0000000000000000000000000000038565739482
iterations = 4
```

```
*****     TRAP.M CODE     *****:
%trap.m
%This script determines the integral of sin(x) using the trapezoidal rule

a = input('Enter the interval start: ');
b = input('Enter the interval end:   ');
%n = input('Enter the intervals: ');

n = 1;
xs = linspace(a,b,n);
ys = sin(xs);
dx = (b-a)/n;
I = 1/2*(sin(a)+sin(b));

d1dxa=cos(a);
d1dxb=cos(b);
d3dxa=-sin(a);
d3dxb=-sin(b);

%Calculate error coefficients
c1=1/12*(d1dxa-d1dxb);
c2=1/720*(d3dxa-d3dxb);

while n <= 2^20
    %Composite Trapezoidal Method of Integration: Summation
    for i=2:n
        I = I + .5 * ((xs(i)-xs(i-1))*(ys(i)+ys(i-1)));
    end
    %Calculate error
    err=dx^2*c1-dx^4*c2;
    %Print result
    fprintf('+------------------------------------------------------------\n');
    fprintf('|\tINT(sin(x),x,%f,%f) = %1.10f\n',a,b,I);
    fprintf('|\tn = %d\terr = %1.10f\n',n,err);

    %Prepare for next iteration
    n = n*2;
    xs = linspace(a,b,n);
    ys = sin(xs);
    dx = (b-a)/n;
    I = 0;
end
```

```
*****     TRAP.M RESULTS     *****:
>> trap
Enter the interval start: 0
Enter the interval end:   pi
+-------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 0.0000000000
|     n = 1 err = 1.6449340668
+-------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 0.0000000000
|     n = 2 err = 0.4112335167
+-------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.8137993642
|     n = 4 err = 0.1028083792
+-------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9663166788
|     n = 8 err = 0.0257020948
+-------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9926838315
|     n = 16      err = 0.0064255237
```

```
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9982880170
|     n = 32      err = 0.0016063809
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9995855374
|     n = 64      err = 0.0004015952
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9998980128
|     n = 128     err = 0.0001003988
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999747030
|     n = 256     err = 0.0000250997
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999937005
|     n = 512     err = 0.0000062749
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999984282
|     n = 1024    err = 0.0000015687
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999996074
|     n = 2048    err = 0.0000003922
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999019
|     n = 4096    err = 0.0000000980
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999755
|     n = 8192    err = 0.0000000245
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999939
|     n = 16384   err = 0.0000000061
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999985
|     n = 32768   err = 0.0000000015
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999996
|     n = 65536   err = 0.0000000004
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 1.9999999999
|     n = 131072  err = 0.0000000001
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 2.0000000000
|     n = 262144  err = 0.0000000000
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 2.0000000000
|     n = 524288  err = 0.0000000000
+--------------------------------------------------------------
|     INT(sin(x),x,0.000000,3.141593) = 2.0000000000
|     n = 1048576 err = 0.0000000000
```

Observations:

By the resulting calculated errors in the Newton-Raphson approximation shown above, it is easy to see that this method is extremely efficient. After only 4 iterations, the errors only begin to appear in the 30^{th} decimal place.

For the Composite Trapezoidal approximation of integration, the number of step/divisions of the interval must exceed 130,000 for the sine function to return the true value to 10 decimal places. Although this is a high number of iterations, the method is simple, easy to understand, and most importantly easy to code. Also, observe that the errors in the above calculations almost exactly complement the integration value, summing up to the true value of the integral.