Lab #3

Adam Jaso
09/20/2011

Experiment 3: Newton-Raphson Solutions with Loops and Conditions

Objective:

The objective of this experiment was to use MATLAB looping and conditional statements in conjunction with the Newton-Raphson numerical approximation method to solve two physics problems.

Procedure:

The first problem is to determine the wavelength that maximizes Max Planck's Blackbody radiation equation. This maximization is also known as Wein's law. The equation to be solved is:

$\Delta x = -f(x)/f'(x)$ where $f(x) = x * e \wedge x + 5 - 5 * e \wedge x$ and $f'(x) = (x - 4) * e \wedge x$.

The provided initial guess was that x = 10 and from there, until the value of f(x) is less than a certain error margin, the code iterates through x values correcting the input value by the formula, $x = x + \Delta x$, until a solution is reached. This solution uses a function called "newton.m" which is an implementation of the Δx equation shown above.

The Second problem is to determine the eccentric anomaly, E, at a given time, t, of celestial bodies moving under constraint of Kepler's Equation of motion. The procedure to the solution of this problem is identical to the procedure of the first problem. That is, the correction equation, $\Delta E = -f(E)/f'(E)$, is used where $f(x) = E - e * \sin(E) - \eta * t$ and where $f'(E) = E - e * \cos(E)$ and where η is the average angular speed of the body's motion, e is the eccentricity of the body's orbit, and t is any time in the period of motion of the body. In the same method as the first problem, the value of E is corrected successively by the ΔE equation until the function f produces a value less than a certain specified error. Like the first problem, a function called "knewton.m" is used as an implementation of the above ΔE equation.

Code / Results:

Problem 1:

WEIN.M CODE:

```
%wein.m
%This program determines the wavelength that maximizes Planck's
black body
%equation

h=6.6262e-34;
c=3e8;
k=1.3806e-23;
x=10;
dx = newton(x);
err=0.0000000001;

while abs(dx) > err
    dx = newton(x);
```

```
    x = x + dx;
end

constant=(h*c)/(x*k);
fprintf('x = %2.10f\n',constant);
```

NEWTON.M FUNCTION CODE:

```
function value=newton(x)
value=-((x*exp(x)+5-5*exp(x))/((x-4)*exp(x)));
```

WEIN.M OUTPUT:

```
>> wein
x = 0.0028999378
```

OBSERVATIONS:

I noticed that the value for x that my code calculated was slightly different from the value that I looked up in my Modern Physics textbook 2.898 x 10^-3 m K(Thornton & Rex, p102, 3rd Ed.) This may be the cause of the required use of c = 3 x 10^8 m / s.

Problem 2:

KEPLER.M CODE:

```
%kepler.m
%This program finds the eccentric anomaly, E, of an elliptical
orbit.

P=75.99;
a=17.94;
angSp=2*pi/P;
e=0.967;
t=15;
err=1e-7;

E=angSp*t+e*sin(angSp*t)+0.5*e^2*sin(2*angSp*t);
dE=knewton(E,e,angSp,t);

while abs(dE) > err
    dE=knewton(E,e,angSp,t);
    E=E+dE;
end

r=a*(1-e*cos(E));
th=180/pi*acos((cos(E)-e)/(1-e*cos(E)));
fprintf('E(t = %2.1f yrs) = %5.8f rad\nr(theta = %2.4f deg) =
%5.8f AU\n',t,E,th,r);
```

KNEWTON.M FUNCTION

```
function value=knewton(E,ecc,angSp,t)
value=-(E-ecc*sin(E)-angSp*t)/(1-ecc*cos(E));
```

KEPLER.M OUTPUT

```
>> kepler
E(t = 15.0 yrs) = 2.08311341 rad
r(theta = 171.3352 deg) = 26.44394782 AU
```

OBSERVATIONS:

This method is very applicable to the solution of all kinds of transcendental equations in Physics.