Computer Problem #1

By: Adam Headley   20 October 2014

A Reference the MATLAB function **loopa.m** in the appendix for the following analysis. Let n be some large number like 10^6 or 10^9. By running this loop for dt = 10^{-6} and 2 $*10^{-6}$ starting with t = 0 the following table can be made.

Changes per loop dt =	Number of Loops	
	$n = 10^6$	$n = 10^9$
10^{-6}	1.00000000000792	999.999997567259
$2*10^{-6}$	2.00000000001584	1999.99999513452

Error: Absolute: $|t - t^*|$ and Relative: $\frac{|t - t^*|}{|t|}$

For t = 10^6 and dt = 10^{-6}, t = 1, t* = 1.00000000000792

Absolute Error: $|t - t^*|$ = |1 - 1.00000000000792| = 0.792×10^{-11}

Relative Error: $\frac{|t - t^*|}{|t|} = \frac{|1 - 1.00000000000792|}{1}$ = 0.792×10^{-11}

Notice: $t = 1$ ⇒ Absolute Error = Relative Error

For n = 10^6 and dt = $2*10^{-6}$, t = 2, t* = 2.00000000000792

Absolute Error: $|t - t^*|$ = |2 - 2.00000000001584 | = 0.1584×10^{-10}

Relative Error: $\frac{|t - t^*|}{|t|} = \frac{|2 - 2.00000000001584|}{2}$ = 0.792×10^{-11}

For n = 10^9 and dt = 10^{-6}, t = 1000, t* = 999.999997567259

Absolute Error: $|t - t^*|$ = |1000 - 999.999997567259| = 0.2432741×10^{-5}

Relative Error: $\frac{|t - t^*|}{|t|} = \frac{|1000 - 999.999997567259|}{1000}$ = 0.2432741×10^{-8}

For n = 10^9 and dt = $2*10^{-6}$, t = 2000, t* = 1999.99999513452

Absolute Error: $|t - t^*|$ = |2000 - 1999.99999513452| = 0.486548×10^{-5}

Relative Error: $\frac{|t - t^*|}{|t|} = \frac{|2000 - 1999.99999513452|}{2000}$ = 0.2432741×10^{-8}

It is clear that increase in the magnitude of uncertainty of the measured quantity compared to the size of the measurement (relative error: $10^{-11} \rightarrow 10^{-8}$) is directly proportional to the increase in the magnitude of the measured quantity ($10^6 \rightarrow 10^9$). Also, notice that the doubling of absolute error in both cases ($0.792\times10^{-11} \rightarrow 0.1584\times10^{-10}$ & $0.2432741\times10^{-5} \rightarrow 0.486548\times10^{-5}$) is also directly proportional to the doubling of dt ($10^{-6} \rightarrow 2*10^{-6}$).

B The Bessel function of order m, is denoted by $J_m(x)$, is defined as follows:

$$J_m(x) = \left(\frac{x}{2}\right)^m \sum_{k=0}^{\infty} \frac{\left(\frac{-x^2}{4}\right)^k}{k!\,(m+k)!}, \quad m \geq 0.$$

This series converges very rapidly and can be used to evaluate $J_m(x)$ for a given value of x. In addition, the following 3-term recursion relation holds:

$$J^*_{m+1}(x) = \left(\frac{2m}{x}\right) J^*_m(x) - J^*_{m-1}(x), \quad m \geq 1.$$

Using the series definition and the recursion relation above the following table was been constructed. This table includes the values for $J_m(1), m = 0,1,\ldots,8,$ whose absolute errors did not exceed $5.0E-11$ while being *evaluated* to 10 significant digits. For further details involving the evaluation of $J_m(1)$ see the MATLAB function **bessel.m** in the appendix.
Using the given values of $J_0(1) = 0.76519\ 76866$ and $J_1(1) = 0.44005\ 05857$ (the first two values obtained from bessel.m) in the recursion relation, the following *computed* values of $J^*_m(1),\ m = 2,\ldots,8$ are found (reference besrec.m in the appendix to see how $J^*_m(1)$ was computed).

m	$J_m(1)$	$J^*_m(1)$	ϵ_m
0	0.76519 76866	0.76519 76866	0
1	0.44005 05857	0.44005 05857	0
2	0.11490 34849	0.11490 34848	-1 E -10
3	0.01956 33540	0.01956 33535	-5 E -10
4	0.00247 66390	0.00247 66362	-2.8 E -09
5	0.00024 97577	0.00024 97361	-2.16 E -08
6	0.00002 09383	0.00002 07248 0002	-2.1350002 E -07
7	0.00000 15023	-0.00000 10384 99751	-2.540799751 E -06
8	0.00000 00942	-0.00003 52637 9654	-3.535799654 E -05

This table shows the error ϵ_m by computing the differences of the two sets such that $\epsilon_m = J^*_m(1) - J_m(1)$. The absolute error is $|\epsilon_m|$. Notice the first two values of ϵ_m are 0 because they were obtained from $J_m(1)$ and the errors start at E -10. It is clear that errors increase and correlate with the increasing *m*. The error ϵ_m is greater than the value of $J_m(1)$ in some cases for larger *m*. The Bessel Function with larger *m* uses more recursions, and notice in the recursion computation, only the first two values for (which were given) are accurate with ten significant digits. The errors are collected from the time when the first operation takes place starting at E -10 and growing to E -5. Therefore, ϵ_m increases as *m* increases.

Appendix

```
function [ t ] = loopa( dt,n )
%   loopa.m
        %   (a) Let t be initialized to 0
        %       and let the increment dt be a given input.
        %   (b) Inside the loop, let t = t + dt.
        %   (c) Run this loop n times.
        %
        %    Used in A – 1

                t = 0;

                for j = 1:n;
                      t = t + dt;
                end

                %accuracy up to 10 sigfigs
                digits(10)
                t = vpa(t);

                end

function [ J,r,k ] = bessel( m,x )
 %bessel.m
        %This definition of the Bessel function J_m(x) contains a series that converges very
        %rapidly and can be used to evaluate J_m(x) for a given value of m and x.
        %This method is used to attain the results of J_m.
        %
        %  note: m >= 0
        %     J = sum of k terms in bessel function
        %     r = k+1 value of the bessel function
        %        = error bound
        %     k = counter

        J = 0;
        r = 1;
        k = 0;

        while r > (5 * 10^(-11))
              J = J + (0.5*x)^m * (-0.25*x^2)^k / (factorial(k)*factorial(m + k));
              r = abs((0.5*x)^m * (-0.25*x^2)^(k+1) / (factorial(k+1)*factorial(m+(k+1)));
              k = k+1;
        end
        end
```

```
%besrec.m
    % 1st 2 values of Bessel function @ x = 1:
        J0 = 0.7651976866;
        J1 = 0.4400505857;

    % J as an array to store the computed terms from the loop
        J = [9];

    % The first 2 terms need to be stored in the first 2 elements of J
        J(1) = J0;
        J(2) = J1;

    % Store the next 6 terms in J at the ith element starting with J(3).
        for i=3:9;
           J(i) = 2*(i-2)*J(i-1) - J(i-2);
        end

    % Accuracy up to 10 sigfigs
        digits(10)
        J = vpa(J);
```