Rahul Wankhede
MTech (Res), CDS

Homework-3

Problem 1

Recursion in the forward direction is given by:

$$J_n(x) = \frac{2(n-1)}{x} J_{n-1}(x) - J_{n-2}(x)$$

Code 1: Forward code in C

```c
#include <stdio.h>
#include <stdlib.h>

#define SIZE 11

double * getBesselArray(int x, double * J){
/* function to get J_i(x) for i = 2 to 10; returns filled array
x = 1, 5, 50; J = pointer to first element of array */

  for (int i = 2; i < SIZE; i++){
    J[i] = (2 * (i-1) * J[i-1] / x) - J[i-2];
  }
  return J;
}


int main(){

/* initialising first two elements with J_0 and J_1 */

  double * J1 = malloc(sizeof(double) * SIZE);
  J1[0] = 0.76519, J1[1] = 0.44005;

  double * J5 = malloc(sizeof(double) * SIZE);
  J5[0] = -0.17759, J5[1] = -0.32757;

  double * J50 = malloc(sizeof(double) * SIZE);
  J50[0] = 0.055812, J50[1] = -0.097511;

  J1 = getBesselArray(1, J1);
  J5 = getBesselArray(5, J5);
  J50 = getBesselArray(50, J50);

  printf("J_10(1) = %f\n", J1[10]);
  printf("J_10(5) = %f\n", J5[10]);
  printf("J_10(50) = %f\n", J50[10]);

  free(J1); free(J5); free(J50);

  return 0;
}
```

The output of the following code is:

$$J_{10}(1) = 560.553310 \tag{1}$$
$$J_{10}(5) = 0.001585 \tag{2}$$
$$J_{10}(50) = -0.113847 \tag{3}$$

Values	x = 1	x = 5	x = 50
Absolute value (p)	2.53×10^{-10}	1.47×10^{-3}	-0.113847849
Calculated value (p^*)	560.5533	1.59×10^{-3}	-0.113847000
Absolute error ($\mid p - p^* \mid$)	560.5533	1.17×10^{-4}	8.49×10^{-7}
Relative error ($\frac{\lvert p-p^*\rvert}{\lvert p\rvert}$)	2.13×10^{8}	0.079	7.46×10^{-6}

We can see that the relative errors are unacceptably large for $x = 1$, just acceptable for $x = 5$, and quite good for $x = 50$.

Problem 2

Recursion in the backward direction is given by:

$$J_n(x) = \frac{2(n+1)}{x} J_{n+1}(x) - J_{n+2}(x)$$

Code 2: Backward code in C

```
double * getBesselArray(int x, double * J){
/* function to get J_i(x) for i = 8 to 0 */

  for (int i = 8; i >= 0; i--){
    J[i] = (2 * (i+1) * J[i+1] / x) - J[i+2];
  }
  return J;
}
```

The output is:

$$J_0(1) = 0.765190 \tag{4}$$
$$J_0(5) = -0.177594 \tag{5}$$
$$J_0(50) = 0.055807 \tag{6}$$

Values	x = 1	x = 5	x = 50
Absolute value (p)	$7.65197686 \times 10^{-1}$	$-1.7759677131 \times 10^{-1}$	5.5812×10^{-2}
Calculated value (p^*)	$7.65190000 \times 10^{-1}$	$-1.7759400000 \times 10^{-1}$	5.5807×10^{-2}
Absolute error ($\mid p - p^* \mid$)	7.686×10^{-6}	2.77131×10^{-6}	5×10^{-6}
Relative error ($\frac{\mid p-p^* \mid}{\mid p \mid}$)	1.0044×10^{-5}	1.56045×10^{-5}	8.9586×10^{-5}

Clearly, the backward recursion has much less error than the forward one for all values of x (around $\frac{1}{1000}^{th}$ to $\frac{1}{100}^{th}$ percentage relative error).

Problem 3

Can the error propagation be formally analyzed using the difference equation analysis we performed in class?

I think so. I'll try:

Say there is an error in the recording of the observations $J_0(x)$ and $J_1(x), = \Delta x_0$ and Δx_1 respectively. Then, what we are actually calculating is:

$$
\begin{aligned}
J_n^*(x) &= \frac{2(n-1)}{x}[J_{n-1}(x) + \Delta x_1] - [J_{n-2}(x) + \Delta x_0] \\
\epsilon_n &= \mid J_n(x) - J_n^*(x) \mid \\
&= \frac{2(n-1)}{x}\Delta x_1 - \Delta x_0 \\
\epsilon_{n+1} &= \frac{2n}{x}\epsilon_n - \epsilon_{n-1} \\
\frac{\epsilon_{n+1}}{\epsilon_n} &= \frac{2n}{x} - \frac{\epsilon_{n-1}}{\epsilon_n}
\end{aligned}
$$

i.e., Current error rate = $\frac{2n}{x}$ - Previous error rate. Let's call them R_{n+1} and R_n respectively,

$$
\begin{aligned}
R_{n+1} &= \frac{2n}{x} - R_n \\
\frac{R_{n+1}}{R_n} &= \frac{2n}{x} - 1 \\
\frac{R_{n+1}}{R_n} &\propto \frac{2n}{x}
\end{aligned}
$$

So, the rate of change of error rate is *directly* proportional to n, and *inversely* proportional to x.

Can the error behavior be understood by this analysis?

Yes. Since the rate of growth of error rate is *directly* proportional to n, we can write:
(if we were working with continuous data, we would write in terms of derivatives, but since we have discrete data)

$$\frac{\Delta^2 \epsilon}{\Delta^2 n} = k_1$$
$$\frac{\Delta \epsilon}{\Delta n} = k_1 n + k_2$$
$$\epsilon = k_1 n^2 + k_2 n + k_3$$

So, the error grows quadratically with increasing n; and that is why we see unconditional error growth in the case of forward recursion (since we are increasing n), but not in the case of backward recursion (decreasing n).

Also, we see most error when x = 1, less when x = 5, and even less so when x = 50; since the rate of growth of error rate is *inversely* proportional to x.