Math 170A HW1

Neo Lee

04/07/2023

Problem 1. Done.

Problem 2. Done.

Problem 3. Figure 1 is the function that implements $AB\vec{x}$ through $(AB)\vec{x}$:

```
function flops = A_times_B(A, B, x)

flops = 0;
C = A;
for i = 1:size(A,1)
    for j = 1:size(A,2)
        foo = 0;
        for k = 1:size(A,2)
            foo = foo + A(i,k) * B(k,j);
            flops = flops + 2;
        end
        C(i,j) = foo;
    end
end

y = zeros(size(x));
for i = 1:size(C,1)
    for j = 1:size(x,1)
        y(i) = y(i) + C(i,j) * x(j);
        flops = flops + 2;
    end
end
end
```

Figure 1: $(AB)\vec{x}$ function

Figure 2 shows the flop count for the function:

```
Command Window
>> flops = zeros(1,4); j = 1;
for i = [100, 200, 400, 800]
A = rand(i); B = rand(i); x = rand(i,1);
flops(j) = A_times_B(A,B,x);
j = j + 1;
end
flops
flops =
   1.0e+09 *
    0.0020    0.0161    0.1283    1.0253
```

Figure 2: Flop count for $(AB)\vec{x}$ function

Figure 3 is the function that implements $AB\vec{x}$ through $A(B\vec{x})$:

```
function flops = B_times_x(A, B, x)

flops = 0;
y = zeros(size(x));
for i = 1:size(B,1)
    for j = 1:size(x,1)
        y(i) = y(i) + B(i,j) * x(j);
        flops = flops + 2;
    end
end

z = zeros(size(y));
for i = 1:size(A,1)
    for j = 1:size(y,1)
        z(i) = z(i) + A(i,j) * y(j);
        flops = flops + 2;
    end
end
end
```

Figure 3: $A(B\vec{x})$ function

Figure 4 shows the flop count for the function:

```
Command Window
>> flops = zeros(1,4); j = 1;
for i = [100, 200, 400, 800]
A = rand(i); B = rand(i); x = rand(i,1);
flops(j) = B_times_x(A,B,x);
j = j + 1;
end
flops
flops =
       40000      160000      640000     2560000
```

Figure 4: Flop count for $A(B\vec{x})$ function

We can see apparently that the implementation of $A(B\vec{x})$ uses significantly fewer flops.

It is because assuming $n \times n$ matrix A and B, the total number flops for AB is $2n^3$. After computing $C = AB$, we need to multiply C with $\vec{x}$, which requires another $2n^2$ flops. Hence, $(AB)\vec{x}$ requires $2n^3 + 2n^2$ flops in total with time complexity $O(n^3)$.

On the other hand, $\vec{z} = A(B\vec{x})$ requires $2n^2$ flops to compute $\vec{y} = B\vec{x}$, and $2n^2$ flops to compute $\vec{z} = A\vec{y}$. Hence, in total $A(B\vec{x})$ requires only $4n^2$ flops with time complexity $O(n^2)$.

Problem 4.

(a)
- 5-th line initializes matrix Z of dimension $m \times m$ with all entries equal 0.
- 6-th line initializes diagonal matrix D of dimension $m \times m$ with diagonal entries equal to m.
- 7-th line initializes subdiagonal matrix $subD$ of dimension $m \times m$ with subdiagonal entries equal to m.
- 8-th line creates a matrix A of dimension $m \times m$ with diagonal entries equal to m and subdiagonal entries equal to $-m$. (Note: $A = D - subD$; would do the same work.)
- 10-th line initializes a row vector $\vec{v}$ with m elements, which represents the equally separated points for $x \in [0, 1]$.
- 11-th line creates a vector $\vec{b}$ by squaring each entry of $\vec{v}$, followed by transposing it to a column vector. It represents right hand side of the equation x^2 at each step interval.
- 13-th line solve the linear system $A\vec{u} = \vec{b}$ to find $\vec{u}$, which is a vector of all the approximations of $u(x)$ at each step interval.
- 15-th to 18-th lines plot the analytic solution to the ODE, $u(x) = \frac{x^3}{3}$, by plotting the value of $\frac{x^3}{3}$ at 100 equally separated points between 0 and 1.

- 20-th to 24-th lines plot the approximated solution to the ODE at m equally separated points between 0 and 1.

(b) Figure 5 shows the plot of *solve_ODE* for $m = 3, 10, 50$:

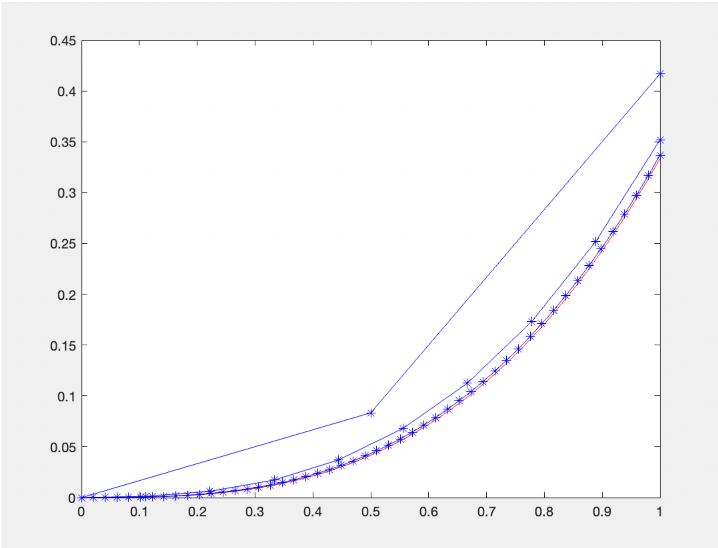

Figure 5: Plot of *solve_ODE*

We can see the blue curve gets closer and closer to the red curve as m increases, which means the approximation is getting closer and closer to the analytic solution. It is because as the interval m gets smaller, the approximation gets more accurate. This agrees with the prediction from finite forward difference approximation where $u'(x) \approx \frac{u(x+\frac{1}{m}-u(x))}{\frac{1}{m}}$, and when m gets bigger, the approximation gets closer to the actual value.

Problem 5. Let $u_i = u(x_i)$ for which $x_i - x_{i-1} = h = \frac{1}{5}$. Since $u''(x) - u(x) = 2$ is true for all $x \in (0,1)$, we can say the following for $x_i \in (0,1)$

$$\begin{aligned} u_i'' - 3u_i &= 2 \\ \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} - 3u_i &\approx 2 \\ 25u_{i+1} - 50u_i + 25u_{i-1} - 3u_i &\approx 2 \\ -\frac{25}{53}u_{i+1} + u_i - \frac{25}{53}u_{i-1} &\approx -\frac{2}{53}. \end{aligned}$$

Then, given $u_0 = u_5 = 0$, we can construct the system of linear equations:

$$\begin{bmatrix} 1 & \frac{-25}{53} & 0 & 0 \\ \frac{-25}{53} & 1 & \frac{-25}{53} & 0 \\ 0 & \frac{-25}{53} & 1 & \frac{-25}{53} \\ 0 & 0 & \frac{-25}{53} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \frac{-2}{53} \\ \frac{-2}{53} \\ \frac{-2}{53} \\ \frac{-2}{53} \end{bmatrix}.$$