CS 71 Lab 1

Jessie Li
September 20, 2023

Part A

Calculate the exact Bessel functions for n = 0 to 30 and evaluate at x = 1, 15, and 30:

```
x = [1 15 30];

n = 30;

J = zeros(n+1, length(x));

for i = 0:n

    J(i+1, :) = besselj(i, x);

end
```

Approximate Bessel functions in the forward direction for n = 0 to 30, then evaluate at x = 1, 15, and 30:

```
Jhat = bessel_forward(n, x)


function J = bessel_forward(n, x)

% Returns Bessel functions computed in the forward direction from 0

% up to n evaluated at a given value of x

    J = zeros(n+1, length(x));

    J(1, :) = besselj(0, x); % J_0(x)

    J(2, :) = besselj(1, x); % J_1(x)

    for i = 2:n

          J(i+1, :) = (2*i ./ x) .* J(i, :) - J(i-1, :);

    end

end
```

Computing Bessel functions in the forward direction produces the following values:

n	Jhat_n(1)	Jhat_n(15)	Jhat_n(30)
0	0.7652	-0.0142	-0.0864
1	0.4401	0.2051	-0.1188
2	0.995	0.0689	0.0705

3	5.53	-0.1775	0.1329
4	43.2448	-0.1636	-0.0351
5	426.9182	0.0685	-0.1446
…	...	...	...
25	5.56E+31	943.1362	-0.1012
26	2.89E+33	2.95E+03	-0.0054
27	1.56E+35	9.69E+03	0.0916
28	8.73E+36	3.32E+04	0.1764
29	5.06E+38	1.19E+05	0.2494
30	3.04E+40	4.42E+05	0.3223

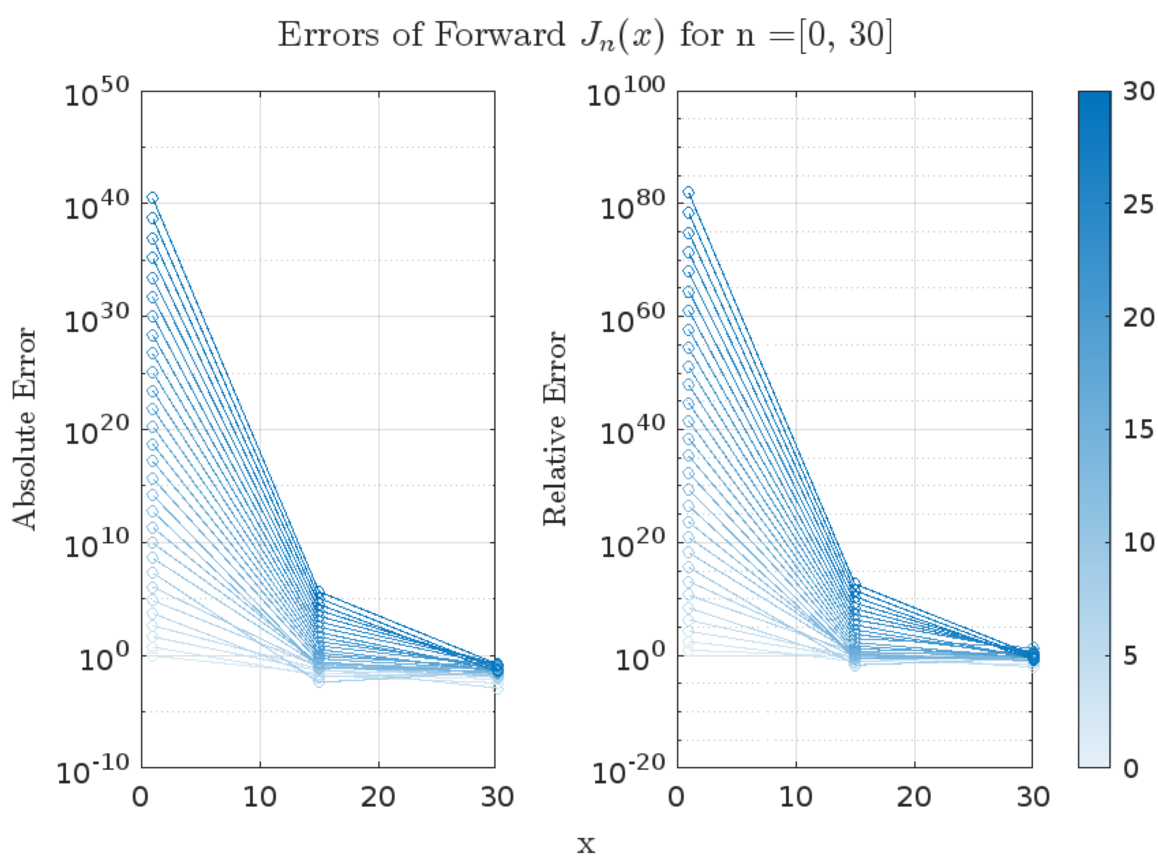
Calculate the absolute and relative errors of the computed Bessel functions:

```
% Calculating absolute and relative errors

absolute_errors_forward = absolute_error(J, Jhat);

relative_errors_forward = relative_error(J, Jhat);


function e = absolute_error(x, xhat)

% Returns an array of absolute errors for each element in x

    e = abs(x - xhat);

end

function e = relative_error(x, xhat)

% Returns an array of relative errors for each element in x

    e = abs((x - xhat) ./ x);

end
```

Plot the errors computed above:

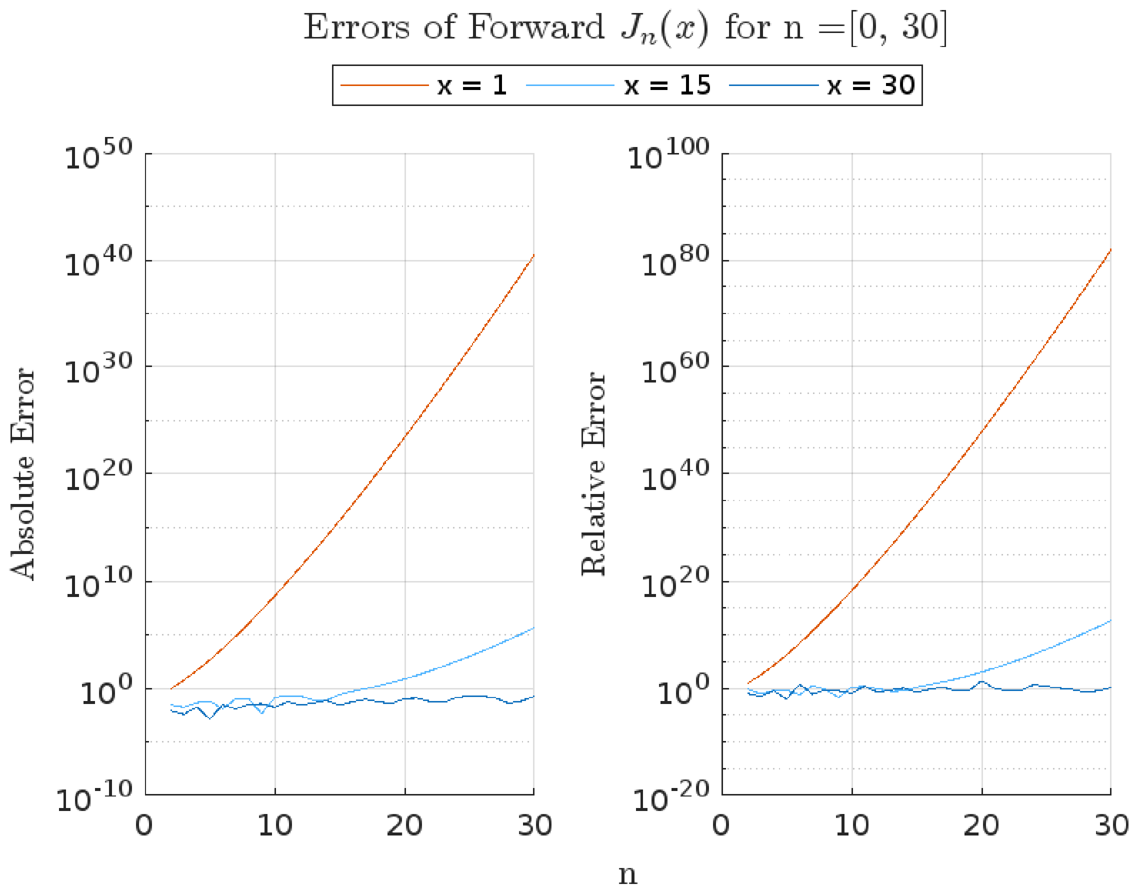
```
% Graphing errors as a function of x

t = tiledlayout(1, 2);

c = sky(31); % colormap

colormap sky

colororder(c);

% Graphing absolute errors

ax1 = nexttile;

plot(x, absolute_errors, '-o', 'MarkerSize', 4);

ylabel(ax1, 'Log Absolute Error', 'interpreter', 'latex');
```

```
grid on
% Relative errors
ax2 = nexttile;
plot(x, relative_errors, '-o', 'MarkerSize', 4);
ylabel(ax2, 'Log Relative Error', 'interpreter', 'latex');
grid on
% Add colorbar
clim([0, 30]);
cb = colorbar();
cb.Layout.Tile = 'east';
% Add title, shared x label, and set y scale to log
title(t, 'Log Errors of $J_n(x)$ for n =[0, 30]', 'interpreter','latex');
xlabel(t, 'x', 'interpreter', 'latex');
set([ax1, ax2], 'YScale', 'log');
```



Errors of forward Bessel as a function of x.

```
% Graphing errors as a function of n
t = tiledlayout(1, 2);
colororder('reef');
% Absolute errors
ax1 = nexttile;
ylabel(ax1, 'Log Absolute Error', 'interpreter', 'latex');
grid on
hold on
for i = 1:3
    plot(ax1, 0:30, absolute_errors(:, i));
end
hold off
% Relative errors
ax2 = nexttile;
ylabel(ax2, 'Log Relative Error', 'interpreter', 'latex');
grid on
hold on
for i = 1:3
    plot(ax2, 0:30, relative_errors(:, i));
end
hold off
% Add title, legend, shared n label, and set y scale to log
title(t, 'Log Errors of $J_n(x)$ for n =[0, 30]', 'interpreter','latex');
xlabel(t, 'n', 'interpreter', 'latex');
set([ax1, ax2], 'YScale', 'log');
leg = legend('x = 1', 'x = 15', 'x = 30', 'Orientation', 'horizontal');
leg.Layout.Tile = 'north';
```



Errors of forward Bessel a function of n.

Approximations lose accuracy as n increases, diverging faster for smaller x. The absolute and relative errors clearly grow exponentially for $x = 1$ when $n > 1$, although the relative errors are generally much larger (which makes sense, because the absolute error is increasing while the exact values rapidly decrease as n increases). For $x = 15$, the error growth starts linear, but becomes visibly exponential around $n = 15$. For $x = 30$, both types of error appear to grow approximately linearly in this interval, $n = 0$ to 30. Both the absolute and relative errors are greatest for smaller x and large n.

Part B

The backwards recurrence can be obtained by rearranging the forward recurrence:

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

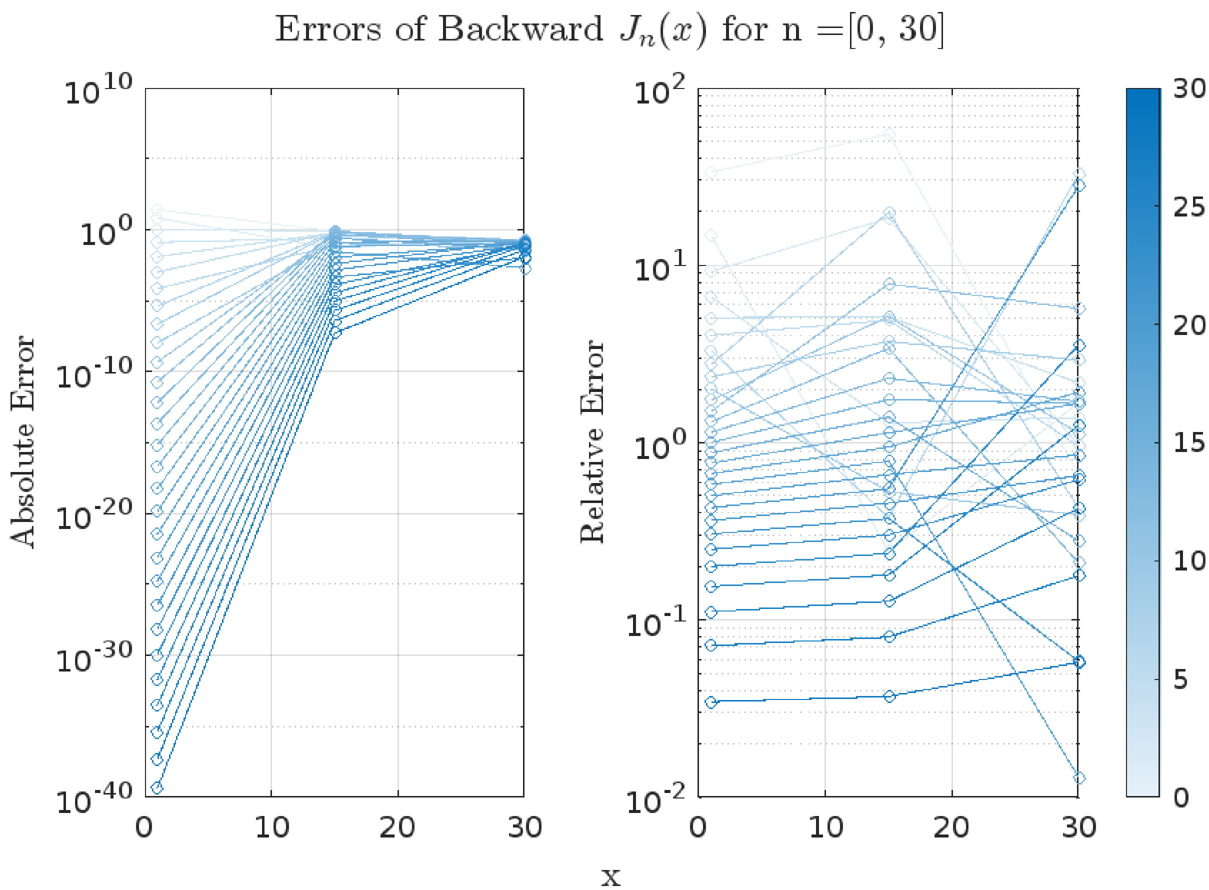
```
Jback = bessel_backward(n, x)


function J = bessel_backward(n, x)

% Returns Bessel functions computed in the backward direction from n
```

```
% down to 0 evaluated at a given value of x
    J = zeros(n+1, length(x));
    J(n+1, :) = besselj(n, x); % J_n(x)
    J(n, :) = besselj(n-1, x); % J_(n-1)(x)
    for i = n:-1:2
          J(i-1, :) = (2*i ./ x) .* J(i, :) - J(i+1, :);
    end
end
```

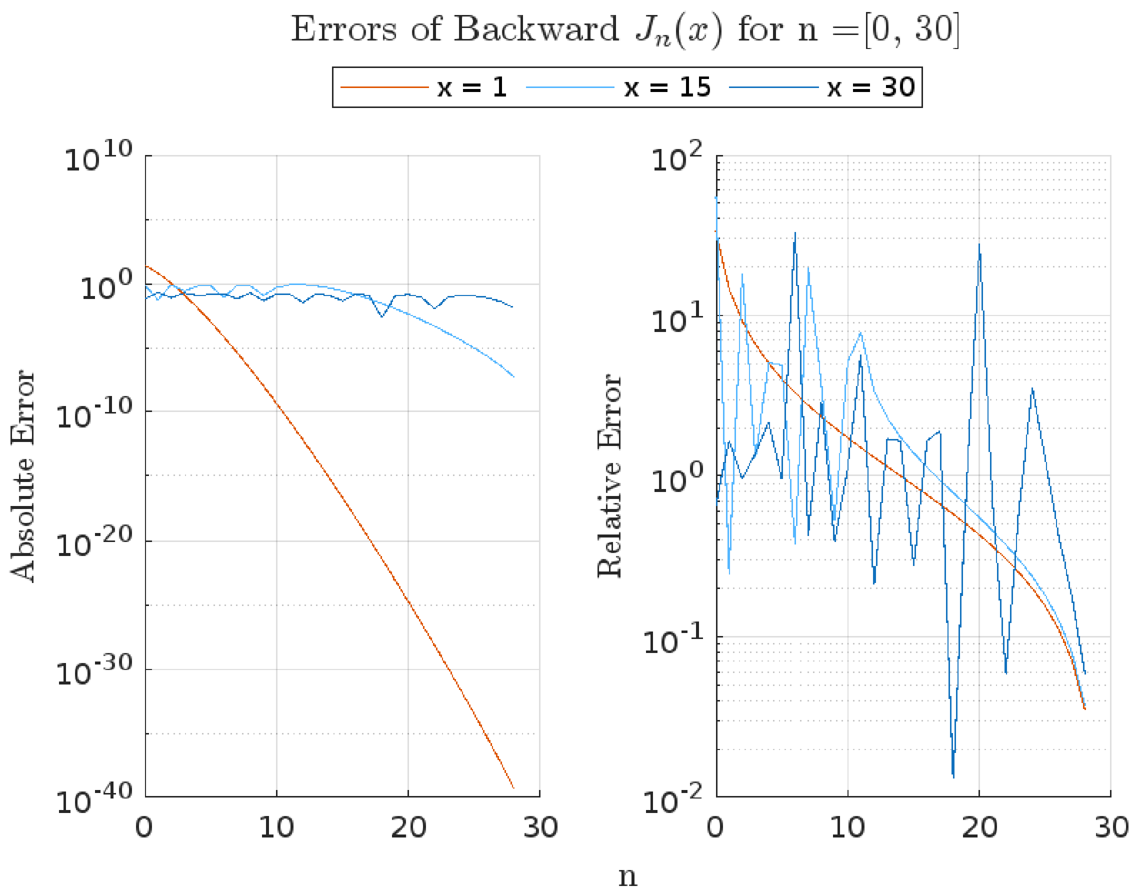
Backwards recursion produces the following values:

n	J_n(1)	J_n(15)	J_n(30)
0	2.64E+01	7.65E-01	-1.43E-01
1	6.89E+00	1.55E-01	8.00E-02
2	1.17E+00	-7.24E-01	1.54E-01
3	1.49E-01	-4.45E-01	-4.92E-02
4	1.50E-02	4.87E-01	-1.67E-01
5	1.26E-03	7.69E-01	-6.41E-03
…	…	…	…
25	2.20E-33	5.97E-05	1.90E-01
26	4.07E-35	1.80E-05	2.46E-01
27	7.27E-37	5.22E-06	2.54E-01
28	1.25E-38	1.45E-06	2.27E-01
29	2.09E-40	3.88E-07	1.86E-01
30	3.48E-42	1.04E-07	1.44E-01

Error graphs (same code as forward Bessel, see code attached):



Errors of backward Bessel as a function of x.



Errors of forward Bessel as a function of n.

Compared to the forward recurrence, the backward recurrence leads to smaller absolute and relative errors overall. The absolute error is the largest for $x = 30$ and smallest for $x = 1$ at the beginning ($n = 30$), but the opposite is true at the end ($n = 0$). When n is large, both errors grow exponentially, faster for small x. The rate of error growth seems to decrease (eventually becoming linear on average) as $n \to 0$, first for large x. Specifically, the errors grows exponentially throughout the interval $n = 30$ to 0 for $x = 1$; for $x = 15$, the absolute error growth starts to plateau, becoming linear around $n = 15$, where changes in the relative error go from strictly increasing to approximately linear, though sporadic; for $x = 30$, the absolute error and relative error growth are overall linear throughout this range of n. It's important to note that the range of relative errors is significantly smaller than the range of absolute errors, so while the relative errors (particularly for $x = 15$ and $x = 30$) may appear to fluctuate dramatically in the figure above, closer inspection reveals that the absolute errors actually follow a similar pattern.

Part C

For a given value of x, the forward recurrence is

$$J_{n+1} = C * J_n - J_{n-1} \qquad \text{(exact)}$$

$$\hat{J}_{n+1} = C * \hat{J}_n - \hat{J}_{n-1} \qquad \text{(computed)}$$

where $C = 2n/x \geq 0$ for all $n \geq 0$ and $x > 0$. Subtracting (computed) from (exact):

$$E_{n+1} = C * E_n - E_{n-1} \qquad (*)$$

where $E_{n+1} = J_{n+1} - \hat{J}_{n+1}$ represents the absolute error of term $n + 1$. If we assume $E_n \propto \lambda^n$, then the general form of E_n will be:

$$E_n = c_1\lambda_1^n + c_2\lambda_2^n$$

(*) becomes:

$$\lambda^{n+1} = C * \lambda^n - \lambda^{n-1}$$
$$\lambda^2 - C * \lambda^n + 1 = 0$$

Solving for λ:

$$\lambda_{1,2} = \frac{1}{2}(C \pm \sqrt{C^2 - 4})$$

Analyzing the Forward Recurrence (Part A)

For $x > 0$, $C > 2$ when $n > x$ (for small x, this happens after just a few iterations). When $C > 2$, the characteristic equation is guaranteed to have two distinct roots $\lambda_1 > \lambda_2$ and at least one root λ_1 must be positive and greater than 1. Even if the other root λ_2 is negative, the positive root must be greater in magnitude because C is positive ($|\lambda_2| < |\lambda_1|$). Consistent with the graphs from Part A, this implies that for positive x, the absolute error E_n starts linear but grows faster with each iteration, and is guaranteed to grow exponentially when $n > x$, with $c_1\lambda_1^n$ dominating as $n \to \infty$. Since $C = 2n/x$ grows faster for small x, E_n grows fastest (becoming exponential first) for $x = 1$, followed by $x = 15$, and is slowest (becoming exponential last) for $x = 30$.

Analyzing the Backward Recurrence (Part B)

A similar analysis of the backward recurrence leads to the same characteristic equation. Although the error growth starts exponential since n is large relative to x (particularly for $x = 1, n = 30$), C is instead decreasing with each iteration because n is decreasing, causing the growth of E_n to decline as $n \to 0$ and become linear once $n < x$, first for $x = 30$, then $x = 15$, and finally $x = 1$.

Absolute v. Relative Error

The graphs of errors as function of n show that absolute and relative error tend to grow with each iteration (forwards and backwards) for each value of x, but faster for smaller x. Both errors also tend to be larger for smaller x. Here, graphs of the absolute and relative errors have similar shapes (even in Part B), but their scales in both Part A and Part B differ. In Part A, the range of relative errors appears to be larger compared to the range of absolute errors, but in Part B, the range of absolute errors is much larger. In general, relative error seems like a more meaningful measure of accuracy than absolute error because relative error accounts for the size of the value.

Helpful MATLAB Resources

Important sources I used for this assignment:

- From MATLAB:
  - BesselJ (https://www.mathworks.com/help/matlab/ref/besselj.html)
  - TiledLayout (https://www.mathworks.com/help/matlab/ref/tiledlayout.html)
  - Colormap (https://www.mathworks.com/help/matlab/ref/colormap.html)
  - Colorbar (https://www.mathworks.com/help/matlab/ref/colorbar.html)
  - Colororder (https://www.mathworks.com/help/matlab/ref/colororder.html)
- From MathWorks Central:
  - How to create and position a legend (https://www.mathworks.com/matlabcentral/answers/783273-how-and-where-to-place-the-legend-box-in-a-tiledlayout)
  - How to share a colorbar across multiple subplots (https://www.mathworks.com/matlabcentral/answers/1444554-how-to-set-common-colorbar-for-multiplots)