

---

```
clear

% Initial Values (J_0,1 then J_49,50 then next x value)
X = [7.6520e-1, 4.4005e-1;
      -1.4224e-2, 2.0510e-1;
      7.3669e-3, 1.2604e-1;
      2.9057e-78, 2.9060e-80;
      3.9789e-21, 6.1061e-22;
      1.3775e-3, 6.8185e-4
];
% Create ground truth for bessel function

x_arr = [1.0, 15.0, 40.0];
for j = 1:3
    for i = 0:50
        R(j, i+1) = besselj(i, x_arr(j));
    end
end

J = zeros(6, 51);
absolute_error = zeros(6, 51);
relative_error = zeros(6, 51);

for i = 1:6
    if i <= 3
        k = 0;
        idx = 1;
    else
        k = 3;
        idx = 50;
    end
    % Compute our approximate from recurrence
    J(i, :) = bessel_recurrence(x_arr(i - k), idx, X(i, 1), X(i, 2));
    absolute_error(i, :) = abs(J(i, :) - R(i - k, :));
    relative_error(i, :) = absolute_error(i, :) ./ abs(R(i - k, :));
end

plot_recurrence(J(1, :), R(1, :), ...
    absolute_error(1, :), relative_error(1, :), ...
    "J_n(" + num2str(1.0) + ")", forward direction")
plot_recurrence(J(2, :), R(2, :), ...
    absolute_error(2, :), relative_error(2, :), ...
    "J_n(" + num2str(15.0) + ")", forward direction")
plot_recurrence(J(3, :), R(3, :), ...
    absolute_error(3, :), relative_error(3, :), ...
    "J_n(" + num2str(40.0) + ")", forward direction)
plot_recurrence(J(4, :), R(1, :), ...
    absolute_error(4, :), relative_error(4, :), ...
    "J_n(" + num2str(1.0) + ")", backward direction")
plot_recurrence(J(5, :), R(2, :), ...
    absolute_error(5, :), relative_error(5, :), ...
```

---

```

"J_n(" + num2str(15.0) + "), backward direction")
plot_recurrence(J(6, :), R(3, :), ...
    absolute_error(6, :), relative_error(6, :), ...
    "J_n(" + num2str(40.0) + "), backward direction")

function [] = plot_recurrence( ...
    J_slice, truth, error_abs, error_rel, label)

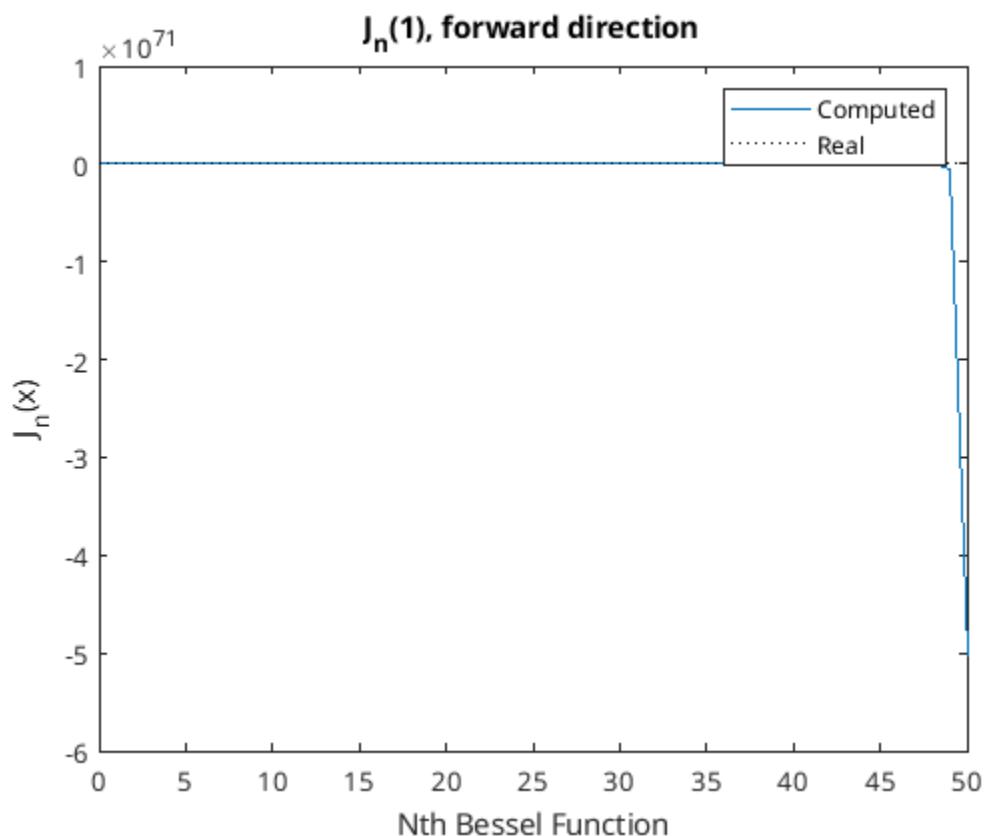
    figure
    n = 0:50;
    plot(n, J_slice)
    hold on
    plot(n, truth, "k:", LineWidth=0.5)
    xlabel("Nth Bessel Function")
    ylabel("J_n(x)")
    legend("Computed", "Real")
    title(label)
    hold off

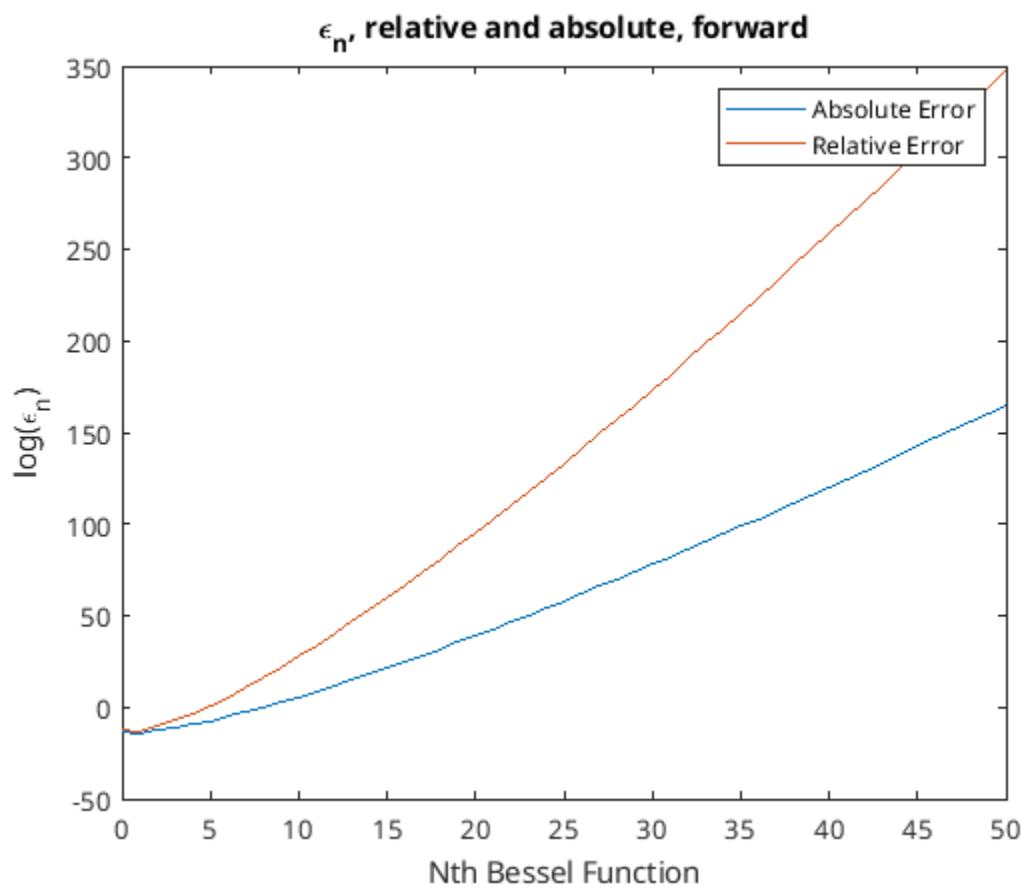
    figure
    plot(n, log(error_abs))
    hold on
    plot(n, log(error_rel))
    title("\epsilon_n, relative and absolute, forward")
    xlabel("Nth Bessel Function")
    legend("Absolute Error", "Relative Error")
    ylabel("log(\epsilon_n)")
    hold off
end

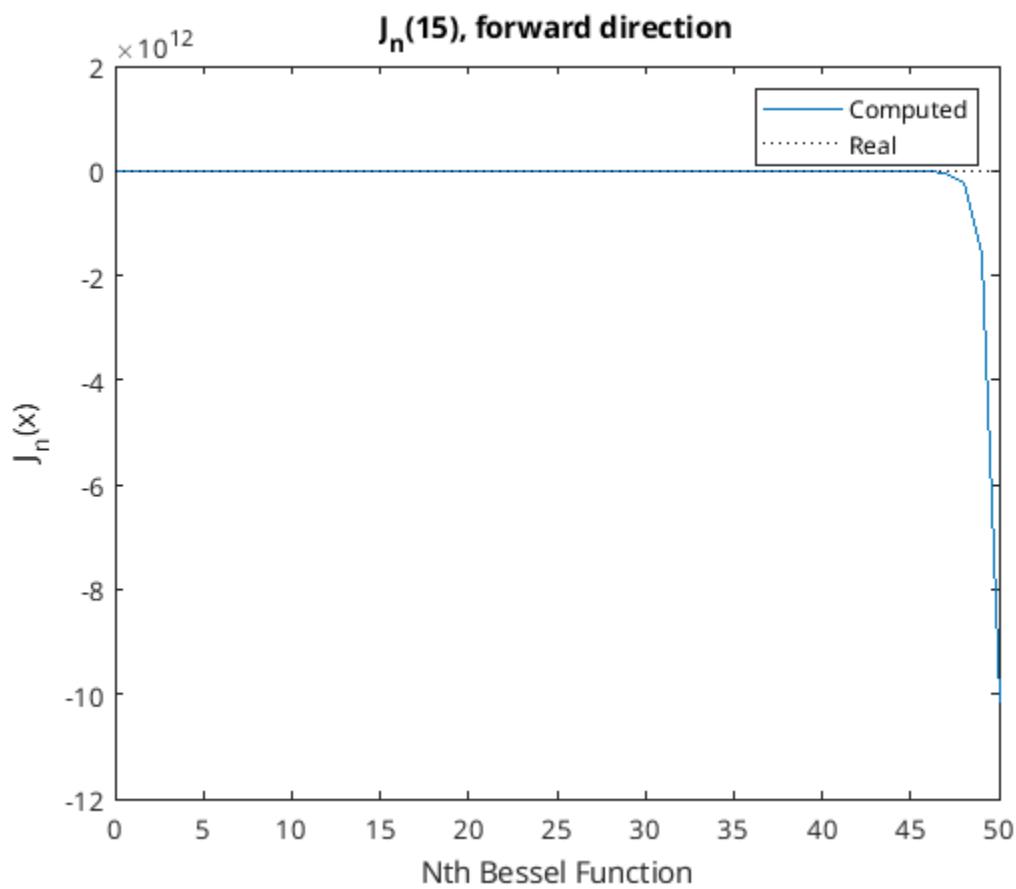
% Recurrence relation J_{n+1} = 2n/x J_n - J_{n-1}
function [J] = bessel_recurrence(x, idx, j0, j1)
    % Specify start index (if 50 go backwards)
    J = zeros(1,51);
    if idx == 50
        J(50) = j0; % J_49
        J(51) = j1; % J_50
        % J_50 == J(51)
        for i = linspace(50,2,49)
            J(i - 1) = ((2.0 * (i - 1)) / x) * J(i) - J(i + 1);
        end
    elseif idx == 1
        J(1) = j0; % J_0
        J(2) = j1; % J_1
        for i = 2:1:50
            J(i + 1) = ((2.0 * (i - 1)) / x) * J(i) - J(i - 1);
        end
    end
end

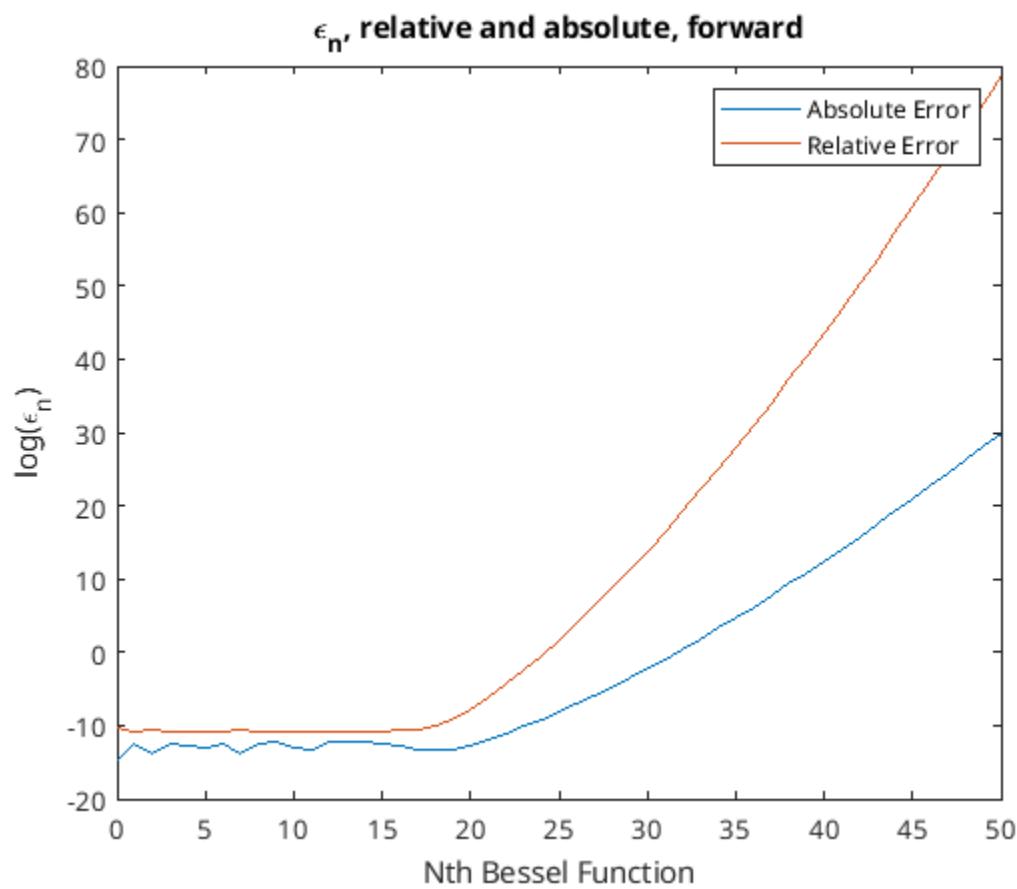
```

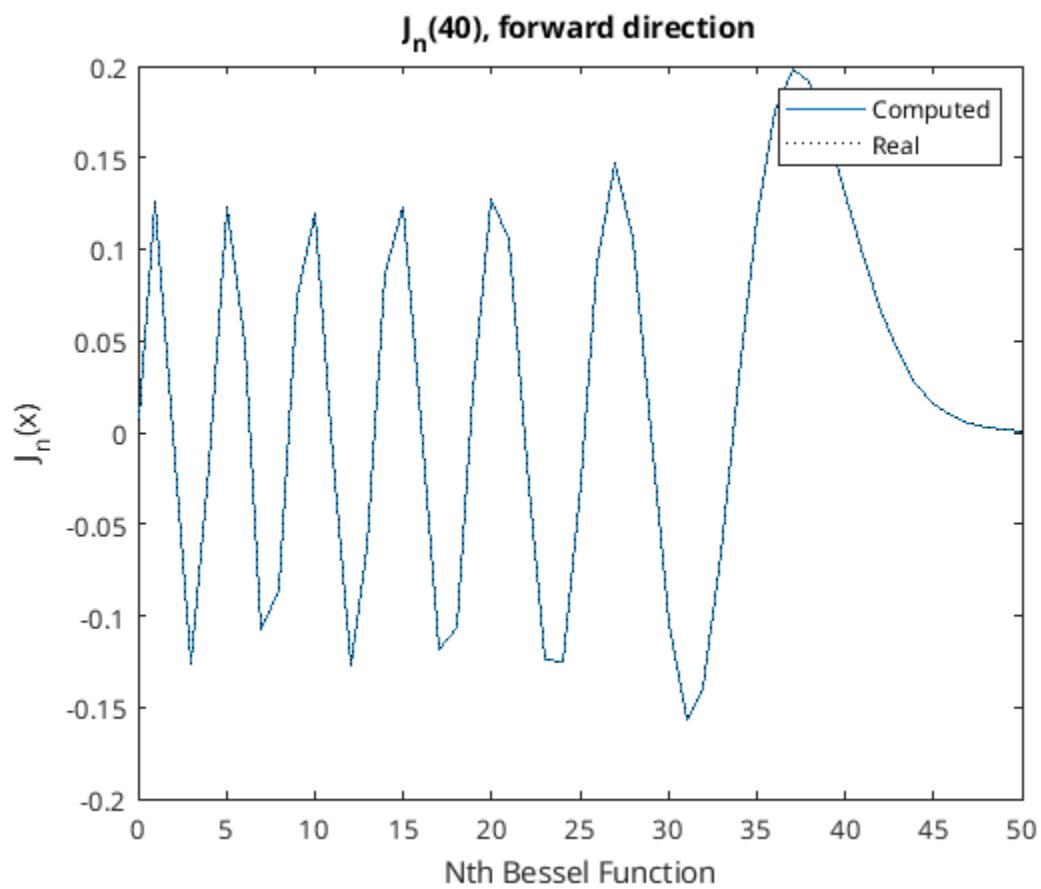
---

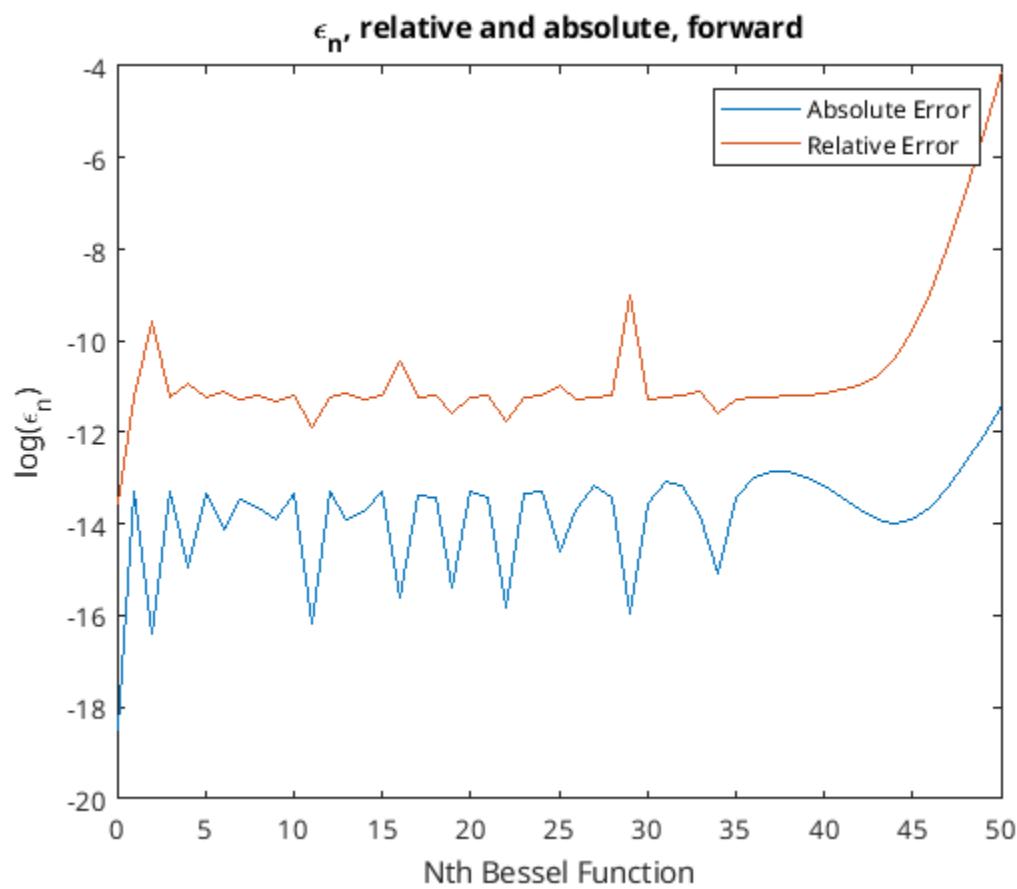


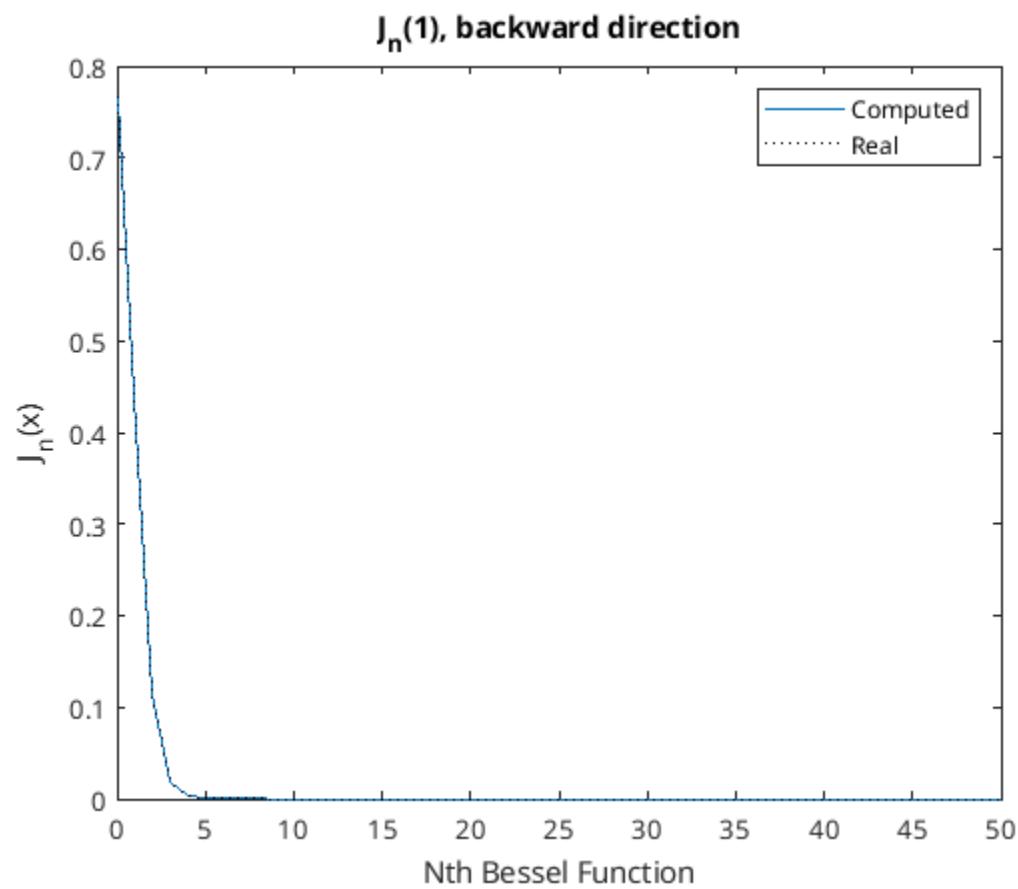


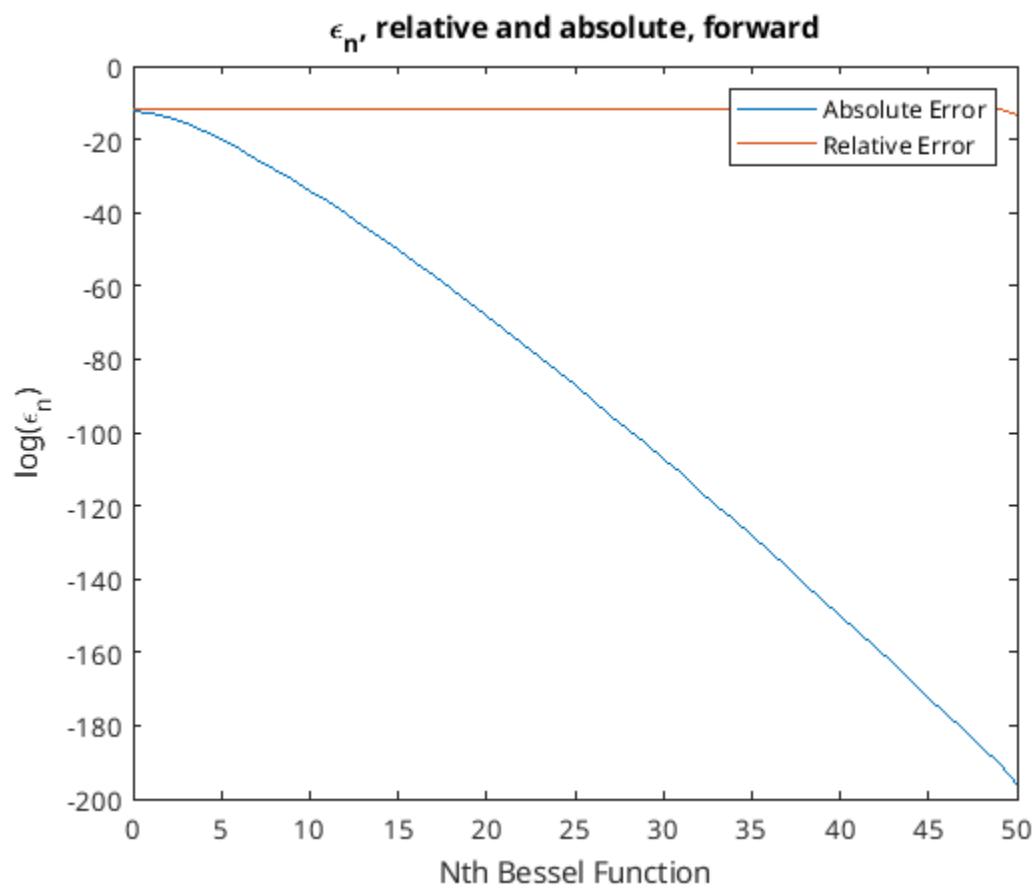


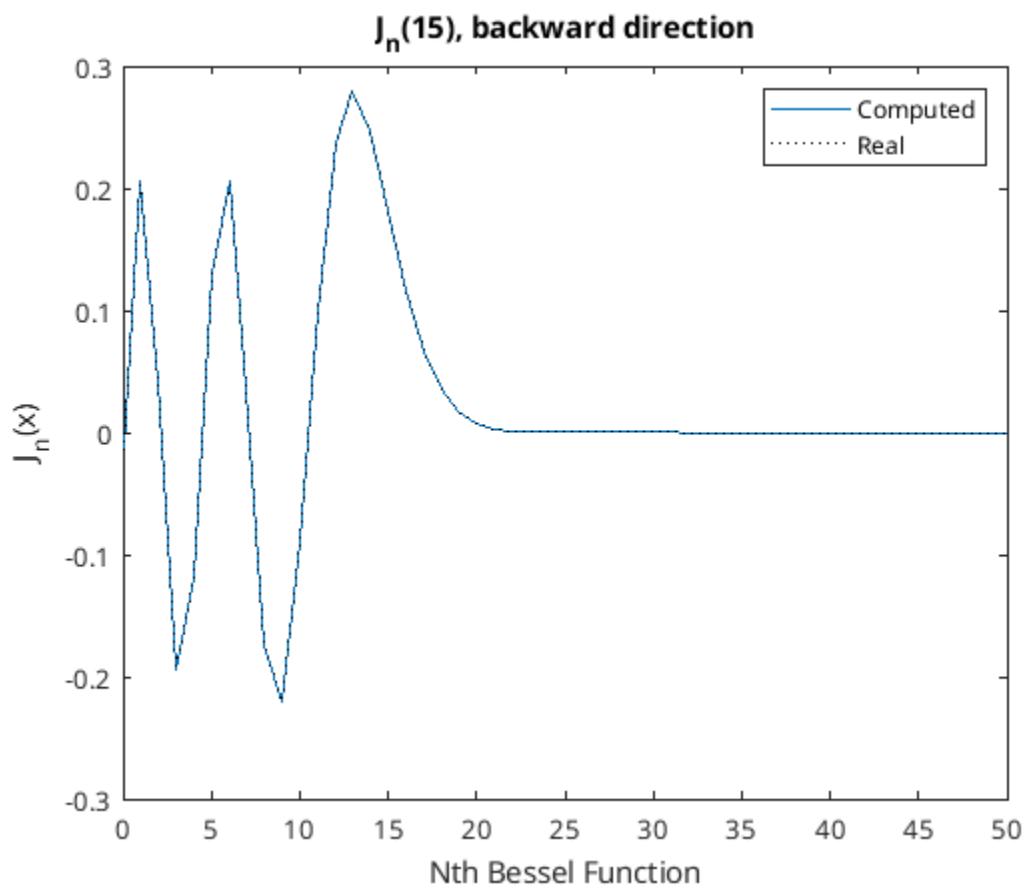


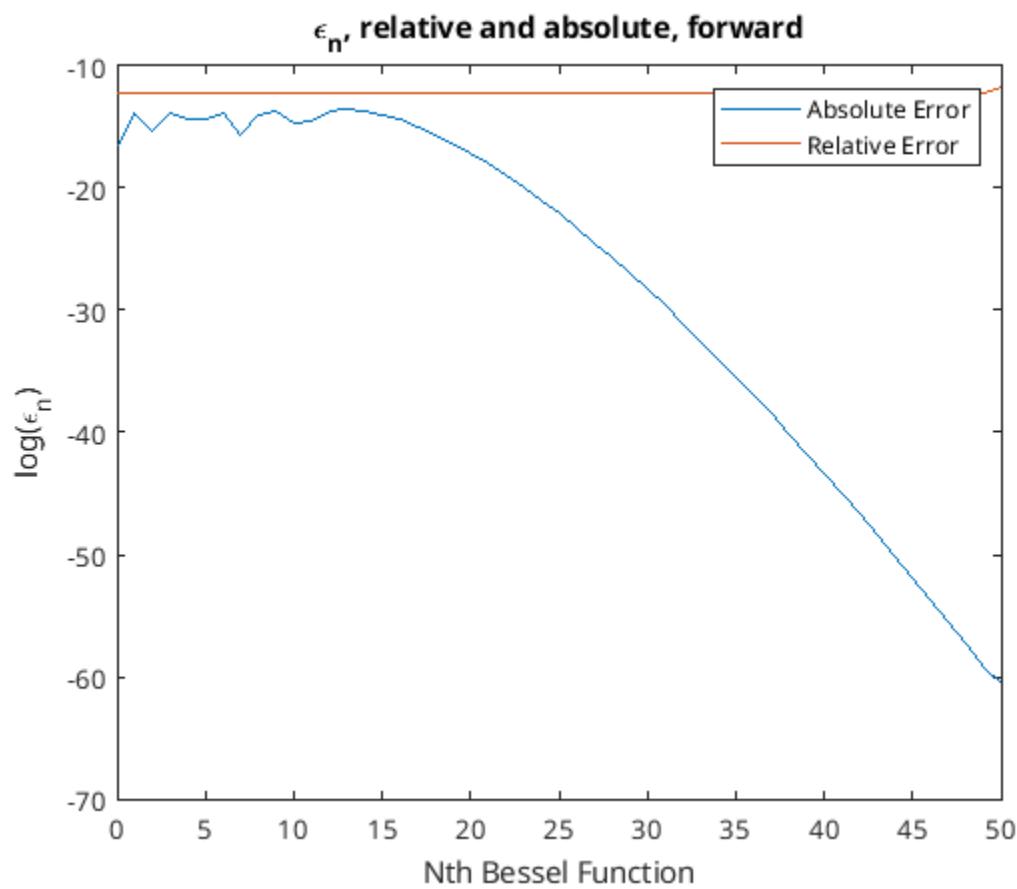


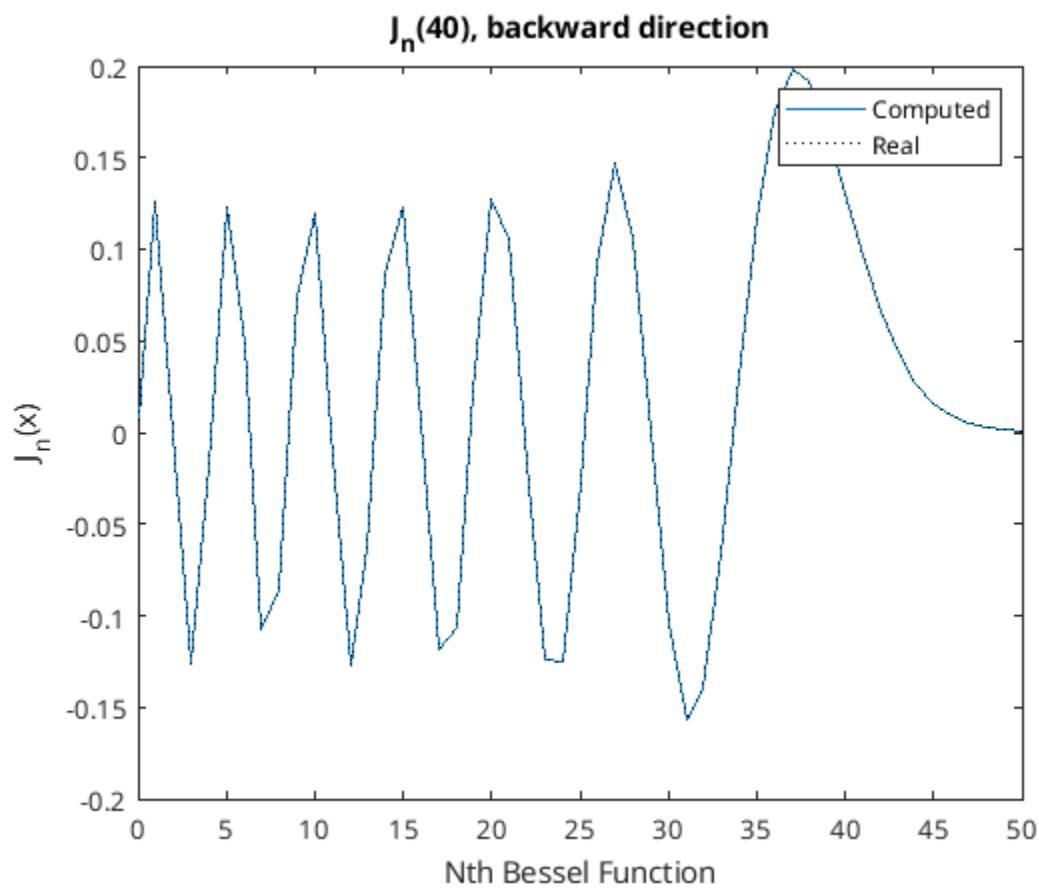


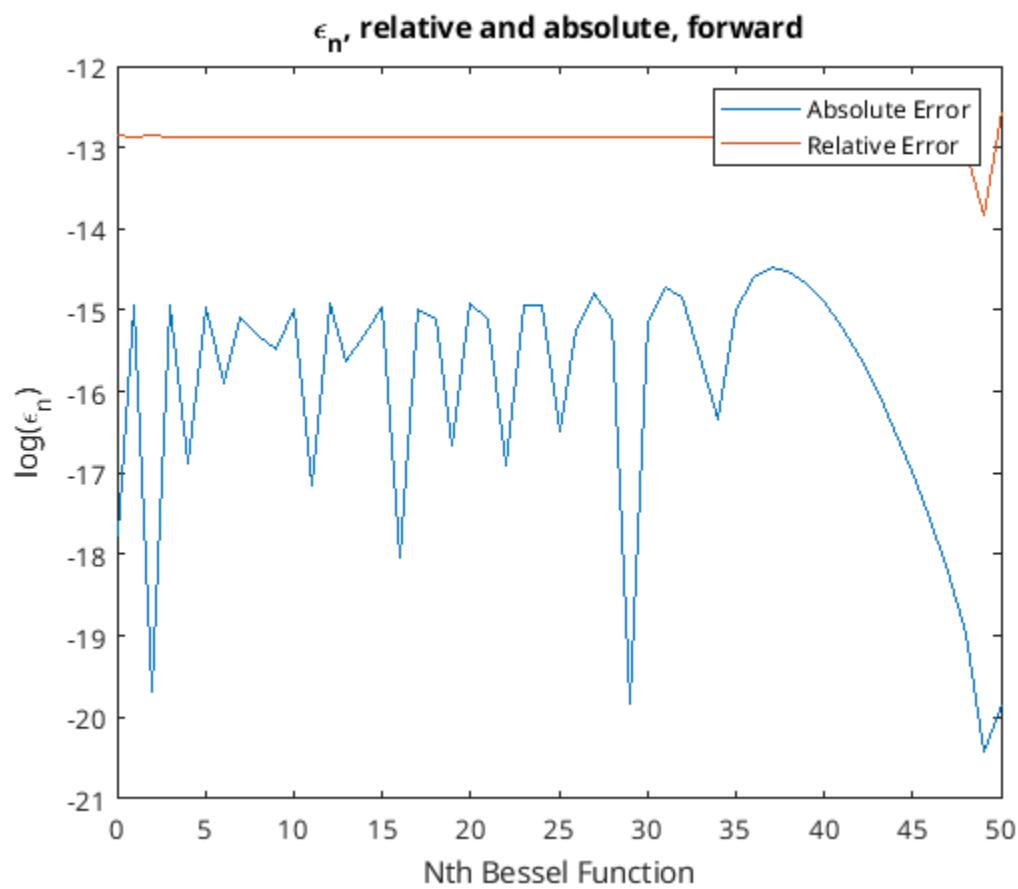












Published with MATLAB® R2024b