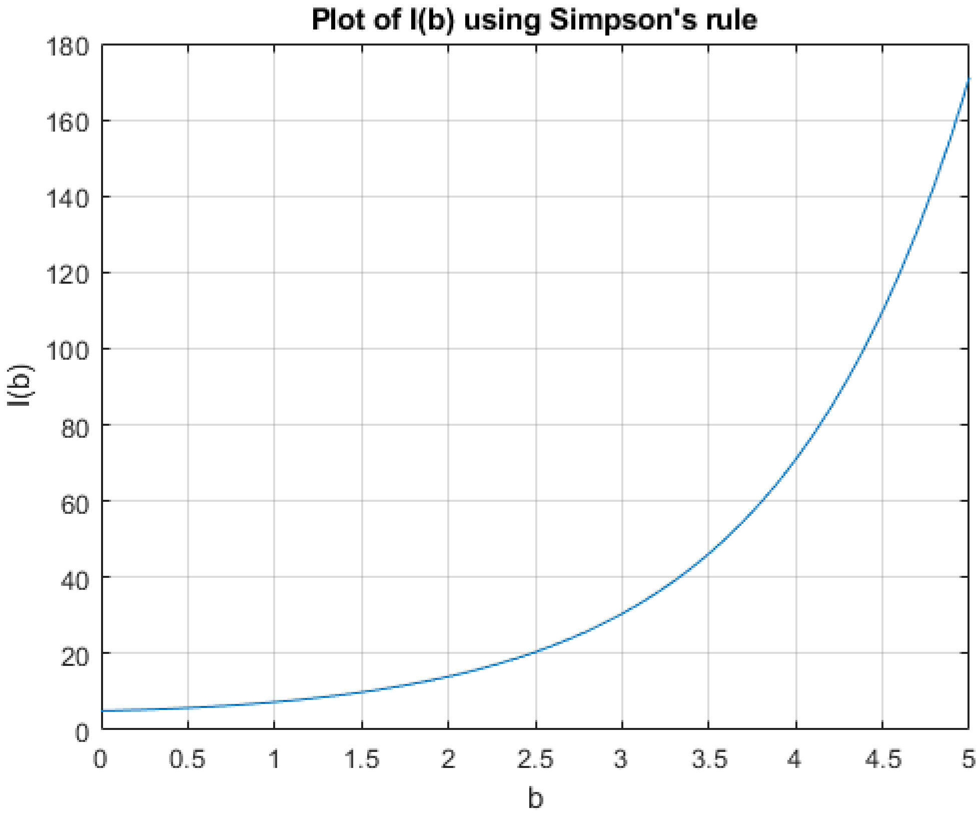
```
function b1()

    % Calculate I(4.6)
    num_intervals = 100; % Using 100 intervals for a good approximation
    b_value_to_shade = 4.6;
    x_values = linspace(0, 5, num_intervals + 1);
    h = x_values(2) - x_values(1);
    y_values = integrand(x_values, b_value_to_shade);
    I_4_6 = (h / 3) * (y_values(1) + 4 * sum(y_values(2:2:end-1)) + 2 *
 sum(y_values(3:2:end-1)) + y_values(end));
    fprintf('I(4.6) = %.4f\n', I_4_6);

    % Calculate I(b) for each b using Simpson's rule
    b_values = 0:0.1:5;
    I_values = zeros(1, length(b_values));

    for idx = 1:length(b_values)
        b = b_values(idx);
        y_values = integrand(linspace(0, 5, num_intervals + 1), b);
        I = (h / 3) * (y_values(1) + 4 * sum(y_values(2:2:end-1)) + 2 *
 sum(y_values(3:2:end-1)) + y_values(end));
        I_values(idx) = I;
    end

    % Plot the results
    plot(b_values, I_values)
    xlabel('b')
    ylabel('I(b)')
    title("Plot of I(b) using Simpson's rule")
    grid on

    function y = integrand(t, b)
        y = exp(-b .* cos(t));
    end
end
```

```
I(4.6) = 119.8920
```



Published with MATLAB® R2023a