Numerical Computing

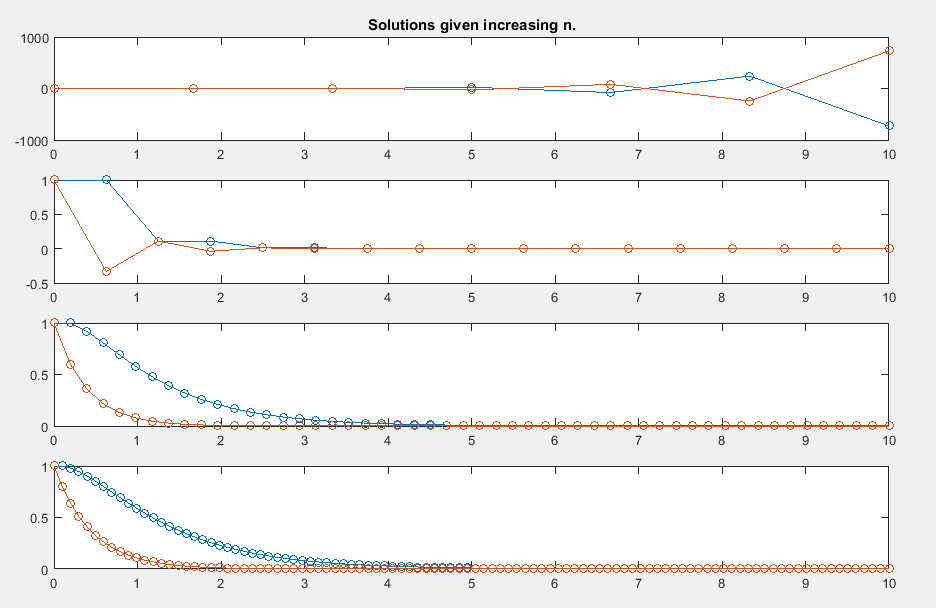
Homework 7

Samuel V

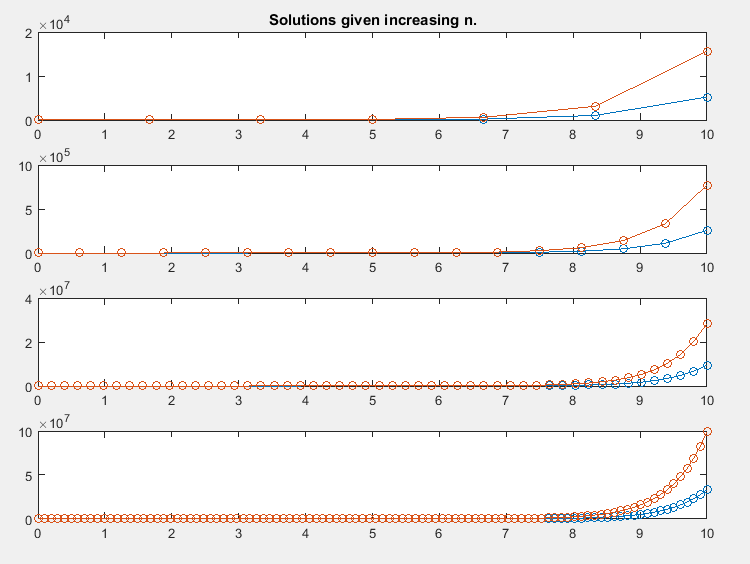
Increasing the size of n causes the approximation to become better. Our step size becomes smaller, so the approximation is more accurate! It takes a lot to get something that looks right – easily more than n=20.

Some visuals:

For n =5, 15, 50, 100, given the first matrix set.



For the second matrix set:



The code:

% Call the forward euler function several times for different values

% of n. Plotting them together, we'll see how the plots increase in

% accuracy.

figure(1)

subplot(4,1,1);

forwardEuler(5);

title('Solutions given increasing n.');

hold on;

subplot(4,1,2);

forwardEuler(15);

subplot(4,1,3);

forwardEuler(50);

subplot(4,1,4);

forwardEuler(100);

hold off;

% Solve a differential equation using forward Euler's method.

function forwardEuler(n)

% Initial Vector.

y0 = [1; 1];

% Choose when to start,

t0 = 0;

% Choose when to end.

T = 10;

% Get the step size.

h = (T - t0)/n;

t = linspace(0, T, n+2);

y(:, 1) = y0;

for i = 1:1:n+1

y(:,i + 1) = y(:, i) + (h \* (feval('myrhs', y(:, i))));

end

% Plot approximation.

plot(t, y, 'o-')

end

% Create the matrix differential equation, and save

% it to dy.

function dy = myrhs(yin)

% Put in whatever A you like. We are in two dimensions.

A = [-1 1; 0 2];

% The differential Equation.

dy = A\*yin;

end