Table of Contents

Question 1 1
Question 2 1
Question 3 1
Question 4 1
Modified template code for Q3 2

Question 1

```
syms u(t)
eqn = (t+1)^2 * diff(u,t,2) - 3*(t+1)*diff(u,t) + t == 1;
Du = diff(u,t);
conds = [u(0) == 1,Du(0) == 1];
result = dsolve(eqn, conds);
pretty(result)
```

```
                              4
t   log(t + 1)   7 (t + 1)    17
- - ---------- + ---------- + --
3        2           24       24
```

Question 2

Substituting v = du/dt, the equation becomes

```
syms v(t)
eqn = (t+1)^2 * diff(v,t) - 3*(t+1)*v + t == 1;
```

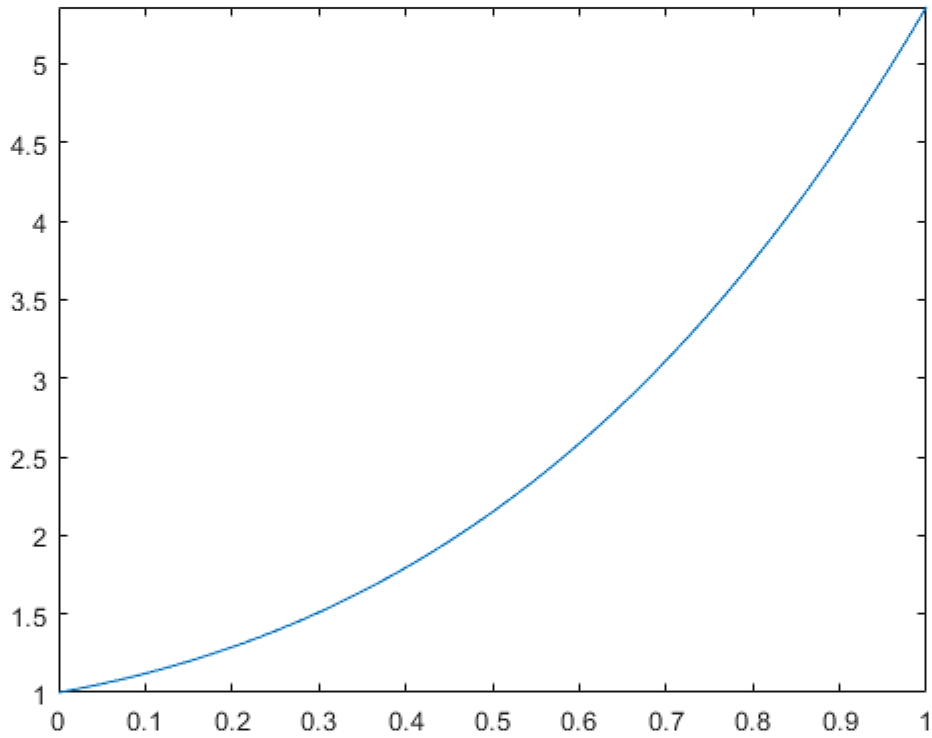
So we have (t+1)^2 dv/dt - 3(t+1)v + t = 1, or equivalently we have dv/dt = (1-t+3v(t+1))/(t+1)^2

Question 3

See end of document

Question 4

```
clf;
fplot(result, [0,1]);
hold on;
[x, sol] = ode();
plot(x,sol(:, 1))
legend("Explicit Solution", "Numeric Solution")
title("Numeric vs Explicit ODE Solutions, Evan Quiney, St Catz")
xlabel("Time")
ylabel("Solution")
```



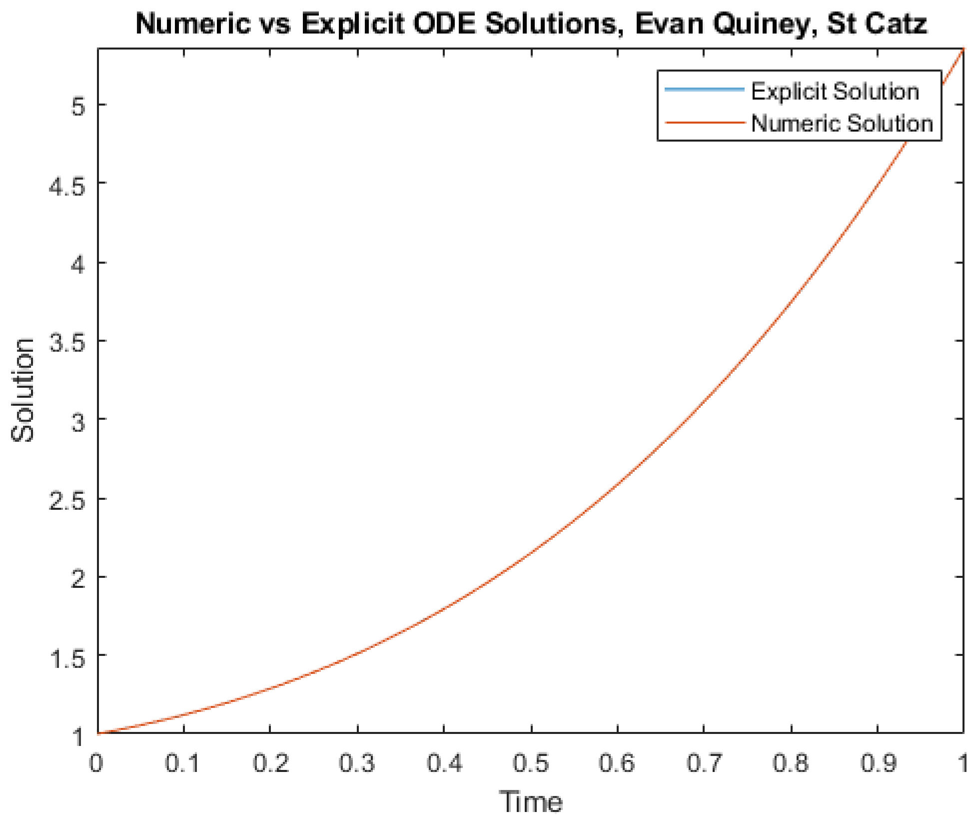
Note: You can't actually see the difference! Also I had some trouble getting the figure to output correctly so it is at the bottom of the pdf.

Modified template code for Q3

```
function [x,sol]=ode() % Function name must be the same as filename.
    y0= [1,1];                       % Specify initial conditions.
    limits=[0,1];                  % Input limits of integration.
    [x,sol] = ode45(@my_system,limits,y0); % Command to numerically
 solve the system 'my_system'
end

function dU=my_system(t,U) % Name of the system we wish to solve
    % dU is an nx1 vector, where n is the number of equations in the
 system.
    dU=zeros(2,1);

    % Input equations below
    dU(1) = U(2);
    dU(2) = (1-t+3*U(2)*(t+1))/(t+1)^2;
end
```



Published with MATLAB® R2019b