```
% ProjectB solution written by Alex Klein

% PROGRAM OVERVIEW
    % Inputs
        % V0 = viral load
        % Q = drug's effectiveness
        % maxtime = maxiumum time span to run the simulation
        % minT = value to stop simulation when the number of total
 T...
            % cells (uninfected and infected) is too low
    % Outputs
        % plot of T and I vs. Time and V vs. Time
        % final output day, uninfected/infected T cell counts, viral
 load

% MAIN LOOP FUNCTION
function ProjectB(V0, Q, maxtime, minT)

% DEFINE AND SET VARIABLES / MATRICES

% set T-cell count to 1
T = 0.9991 ;

% set infected T-cell count to 0
I = 0 ;

% set viral load to V0
V = V0 ;

% reset start day
day = 0 ;

% create matricies to record values over time
dayMatrix = zeros(1, day + 1) ;
TMatrix = zeros(1, day + 1) ;
IMatrix = zeros(1, day + 1) ;
VMatrix = zeros(1, day + 1) ;

% FOR LOOP TO CALL INCREMENT FUNCTION

for i = day : maxtime

    % test that total T cell count is greater than minT
    if (T + I) < minT

        % tell user
        fprintf ('\nThe simulation was stopped after %.0f days because
 \n', day - 1) ;
        fprintf ('the T cell count is less than the minT value of %.4f
 \n ',  minT) ;

        % call end functions
```

```
            graph(dayMatrix, TMatrix, IMatrix, VMatrix) ;
            output(T, I, V, Q, day) ;

            % quit program
            return ;

        end

        % increment T, I, V values - calls procedure subfunctions
        [Tnext, Inext, Vnext] = Increment(T, I, V, Q) ;

        % save incremented values to matrices
        dayMatrix(i + 1) = day ;
        TMatrix(i + 1) = T ;
        IMatrix(i + 1) = I ;
        VMatrix(i + 1) = V ;

        % reset current values
        T = Tnext ;
        I = Inext ;
        V = Vnext ;

        % increment day
        day = day + 1 ;

    end

    % call end functions
    graph(dayMatrix, TMatrix, IMatrix, VMatrix) ;
    output(T, I, V, Q, day) ;

    end

    % END SUBFUNCTIONS

    % subfunction for plots
    function graph(dayMatrix, TMatrix, IMatrix, VMatrix)

    % PLOT T and I VS. TIME

    % plot uninfected T cells
    figure(1) ;
    plot(dayMatrix, TMatrix, '-r') ;

    % plot infected T cells
    hold on ;
    plot(dayMatrix, IMatrix, '-b') ;
    hold off ;

    % edit title
    clear title ;
    title('Uninfected and Infected T-Cells Over Time') ;
    ax = gca ;
    ax.TitleFontSizeMultiplier = 1.5 ;
```

```
% edit axis
xlabel('Time (days)') ;
ylabel('Proportion of Original T-Cells') ;

% legend
legend({'Uninfected T-Cells','Infected T-
Cells'},'Location','northeast') ;

% PLOT V VS. TIME

% plot viral load
figure(2) ;
plot (dayMatrix, VMatrix,'-g') ;

% edit title
clear title ;
title('Viral Load Over Time') ;
ax = gca ;
ax.TitleFontSizeMultiplier = 1.5 ;

% edit axis
xlabel('Time (days)') ;
ylabel('Viral Load') ;

% legend
legend({'Viral Load'},'Location','southeast') ;

% DISPLAY MATRICES (optional, for testing)
%dayMatrix
%TMatrix
%IMatrix
%VMatrix

end

% subfunction for display outputs
function output(T, I, V, Q, day)

% display final time (day) of output and drug effectivness
fprintf(' \nAfter %.0f days with a %.0f percent effective drug: \n',
 day - 1, Q * 100) ;

% display final uninfected and infected T-cell counts
fprintf('The uninfected T-Cell count is %.4f \n', T) ;
fprintf('The infected T-Cell count is %.4f \n', I) ;

% display final viral load
fprintf('The viral load is %.4f \n \n', V) ;

end

% PROCEDURE SUBFUNCTIONS
```

```
% subfunction that increments T-cell count, the viral load and...
    % the infected T-cell count by one day
function [Tnext, Inext, Vnext]= Increment(T, I, V, Q)

    % call subfunction for next day uninfected T cell count
    Tnext = NewT(T, V) ;

    % call subfunction for next day infected T cell count
    Inext = NewI(T, I, V) ;

    % call subfunction for next day viral load
    Vnext = NewV(T, I, V, Q) ;

end

% subfunction for next day uninfected T cell count
function Tnext = NewT(T, V)

Tnext = ( T ) - ( 0.001 * T * V ) ;
    % first term = uninfected T cell count of previous day
    % second term = rate of T cell infection

end

% subfunction for next day infected T cell count
function Inext = NewI(T, I, V)

Inext = ( I ) + ( 0.001 * T * V ) - ( 0.005 * I ) ;
    % first term = infected T cell count of previous day
    % second term = rate of T cell infection
    % third term = rate of infected T cell death

end

% subfunction for next day viral load
function Vnext= NewV(T, I, V, Q)

Vnext = ( V ) + ( 10 * ( 1 - Q ) * I ) - ( 0.05 * V * T) ;
    % first term = viral load of previous day
    % second term = rate of infected T cell death
    % third term = rate of viral clearance

end
```

Published with MATLAB® R2018b

```
% hivodefun ODE solution written by Alex Klein

% Function Overview
% inputs: maxTime and V0 in hivodefun.m call ; Q in ProjectB2.m code
% outputs: graphs of T and I over time and V over time...
    % and display information of final T, I, V after maxTime days

% main function
function hivodefun(maxTime, V0)

[t,y] = ode45(@ProjectB2, [0, maxTime], [1; 0; V0]);

% set output variables
T = y(end,1);
I = y(end,2);
V = y(end,3);
day = t(end,1);

% call output functions
output(T, I, V, day);
graph(t, y);

end

% subfunction for display outputs
function output(T, I, V, day)

% display final time (day) of output and drug effectivness
fprintf(' \nAfter %.0f days: \n', day) ;

% display final uninfected and infected T-cell counts
fprintf('The uninfected T-Cell count is %.4f \n', T) ;
fprintf('The infected T-Cell count is %.4f \n', I) ;

% display final viral load
fprintf('The viral load is %.4f \n \n', V) ;

end

% subfunction for plots
function graph(t, y)

% PLOT T and I VS. TIME

% plot uninfected T cells
figure(1) ;
plot( t,y(:,1), '-r');

% plot infected T cells
hold on
plot( t,y(:,2), '-b');
hold off
```

```
% edit title
clear title ;
title('Uninfected and Infected T-Cells Over Time') ;
ax = gca ;
ax.TitleFontSizeMultiplier = 1.5 ;

% edit axis
xlabel('Time (days)') ;
ylabel('Proportion of Original T-Cells') ;

% legend
legend({'Uninfected T-Cells','Infected T-
Cells'},'Location','northeast') ;

% PLOT V VS. TIME

% plot viral load
figure(2) ;
plot( t,y(:,3), 'g');

% edit title
clear title ;
title('Viral Load Over Time') ;
ax = gca ;
ax.TitleFontSizeMultiplier = 1.5 ;

% edit axis
xlabel('Time (days)') ;
ylabel('Viral Load') ;

% legend
legend({'Viral Load'},'Location','southeast') ;

end

%{
PARAGRAPH ABOUT OBSERVATIONS

hivodefun.m takes the maxTime and initial viral load inputs and calls
ProjectB2.m through the ode45 function. This solves the differnetial
equations in a column vector and hivodefun.m plots and displays the
 values.
Comparing the output of this function and ProjectB.m, we can see that
 the
ode45 solver's model approximation is slightly different than that of
ProjectB.m. This makes sense as ode45 integrates over time, while
ProjectB.m simply takes one value each day. As maxTime increases,
 however,
the results of the two functions gets closer and closer. Below are are
 two
example cases, at different maxTime values:

Example 1:
```

```
>> hivodefun(maxTime = 200, V0 = 0.01) ; Q = 0.1

After 200 days:
The uninfected T-Cell count is 0.0000
The infected T-Cell count is 0.7177
The viral load is 458.1195

>> ProjectB(0.01, 0.1, 200, 0)

After 200 days with a 10 percent effective drug:
The uninfected T-Cell count is 0.0000
The infected T-Cell count is 0.7305
The viral load is 435.0848

Outcome: infect T-Cell count and viral load is slightly different

Example 2:

>> hivodefun(2000, 0.01) ; Q = 0.3

After 2000 days:
The uninfected T-Cell count is 0.0000
The infected T-Cell count is 0.0001
The viral load is 1349.8714

>> ProjectB(0.01, 0.3, 2000, 0)

After 2000 days with a 30 percent effective drug:
The uninfected T-Cell count is 0.0000
The infected T-Cell count is 0.0001
The viral load is 1349.8721

Outcome: uninfected and infected T-Cell counts are the same,
         final viral load is (practically) the same

As we can see, increasing the maxTime (and Q) causes the ProjectB.m
 and
hivodefun.m values to get closer, which are better approximations.

%}
```

Published with MATLAB® R2018b

```
% subfunction that increments T-cell count, the viral load and...
    % the infected T-cell count by one day
function [Tnext, Inext, Vnext]= Increment(T, I, V, Q)

    % call subfunction for next day uninfected T cell count
    Tnext = NewT(T, V) ;

    % call subfunction for next day infected T cell count
    Inext = NewI(T, I, V) ;

    % call subfunction for next day viral load
    Vnext = NewV(T, I, V, Q) ;

end
```

Published with MATLAB® R2018b

```
% subfunction for next day infected T cell count
function Inext = NewI(T, I, V)

Inext = ( I ) + ( 0.001 * T * V ) - ( 0.005 * I ) ;
    % first term = infected T cell count of previous day
    % second term = rate of T cell infection
    % third term = rate of infected T cell death

end
```

Published with MATLAB® R2018b

```
% subfunction for next day uninfected T cell count
function Tnext = NewT(T, V)

Tnext = ( T ) - ( 0.001 * T * V ) ;
    % first term = uninfected T cell count of previous day
    % second term = rate of T cell infection

end
```

Published with MATLAB® R2018b

```
% subfunction for next day viral load
function Vnext= NewV(T, I, V, Q)

Vnext = ( V ) + ( 10 * ( 1 - Q ) * I ) - ( 0.05 * V * T) ;
    % first term = viral load of previous day
    % second term = rate of infected T cell death
    % third term = rate of viral clearance

end
```

Published with MATLAB® R2018b

```
% ProjectB2 ODE solution written by Alex Klein

function Increment = ProjectB2(t,y)

Q = 0;

Increment = [-0.001*y(1)*y(3); 0.001*y(1)*y(3) - 0.005*y(2); 10*(1-
Q)*y(2) - 0.05*y(1)*y(3)];

end
```

Published with MATLAB® R2018b