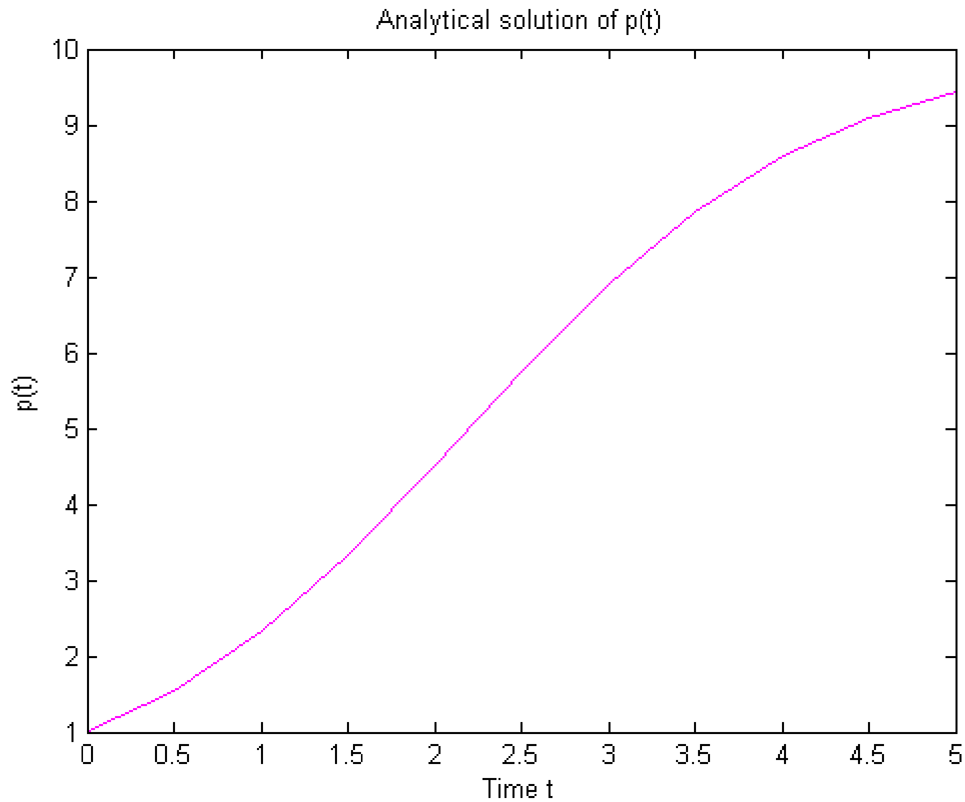
Table of Contents

Worksheet 1

a) Use matlab to plot the function p(t) in a graph. Instantiate time vector

```
dt = [1,1/2,1/4,1/8];
t_end = 5;
t = 0:dt(2):t_end;
p = 10./(1+9.*exp(-t));

plot(t,p,'m-')
xlabel('Time t')
ylabel('p(t)')
title('Analytical solution of p(t)')

fprintf('Program paused. Proceed with Task c). Press enter to continue.\n\n');
pause;
```

```
Program paused. Proceed with Task c). Press enter to continue.
```



b)

see functions Euler.m, Heun.m, RungeKutta.m, (RKEuler.m)

c)

Derivative function of p(t)

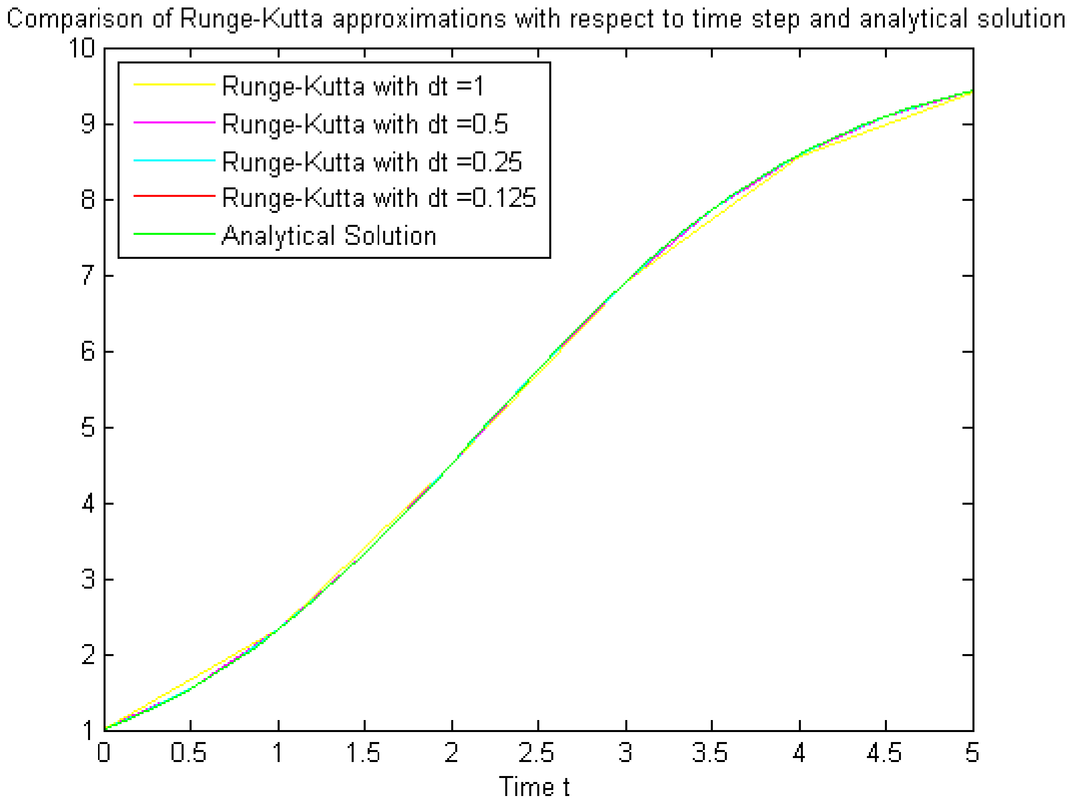
```
eq1 = @(p)(1-(p/10))*p;
y0 = 1;
% Instantiate error vectors
E_eul = zeros(1,length(dt));
E_heu = zeros(1,length(dt));
E_RK = zeros(1,length(dt));

% Color definition matrix for graphs
Color = {[1 1 0], [1 0 1], [0 1 1], [1 0 0], [0 1 0]};

% Euler
% all Euler aproximation values are calculated along with the error E_eul
for i=1:length(dt)
t = 0:dt(i):t_end;
p = 10./(1+9.*exp(-t));

y = Euler(eq1,y0,dt(i),t_end);
```

```
E_eul(i) = sqrt((dt(i)/5)*sum((y-p).^2));
plot(t,y,'Color',Color{i},'LineStyle','-')
hold on
end
fprintf(['The following vector shows the errors obtained for comparing '...
         'the euler approximation to the analytical sol. on different steps.\n']);
E_eul
plot(t,p,'Color',Color{length(dt)+1},'LineStyle','-')
xlabel('Time t')
legend(strcat('Euler with dt =',' ',num2str(dt(1))),...
       strcat('Euler with dt =',' ',num2str(dt(2))),...
       strcat('Euler with dt =',' ',num2str(dt(3))),...
       strcat('Euler with dt =',' ',num2str(dt(4))),...
       'Analytical Solution', 'Location','northwest')
title(['Comparison of Euler approximations with respect to time step ' ...
       'and analytical solution.'])
hold off
fprintf('Program paused. Press enter to continue.\n\n');
pause;

% Heun
% All Heun function approximations are calculated and plotted as well as
% approximation values E_heu
for i=1:length(dt)
t = 0:dt(i):t_end;
p = 10./(1+9.*exp(-t));

y = Heun(eq1,y0,dt(i),t_end);
E_heu(i) = sqrt((dt(i)/5)*sum((y-p).^2));
plot(t,y,'Color',Color{i},'LineStyle','-')
hold on
end
fprintf(['The following vector shows the errors obtained for comparing '...
         'the heun approximation to the analytical sol. on different steps.\n']);
E_heu
% Plot Configuration
plot(t,p,'Color',Color{length(dt)+1},'LineStyle','-')
xlabel('Time t')
legend(strcat('Heun with dt =',' ',num2str(dt(1))),...
       strcat('Heun with dt =',' ',num2str(dt(2))),...
       strcat('Heun with dt =',' ',num2str(dt(3))),...
       strcat('Heun with dt =',' ',num2str(dt(4))),...
       'Analytical Solution', 'Location','northwest')
title(['Comparison of Heun approximations with respect to time step ' ...
       'and analytical solution.'])
hold off
fprintf('Program paused. Press enter to continue.\n\n');
pause;

% Runge-Kutta
% All Runge-Kutta approximations for given values are calcluated and ploted
% along with error calculation E_RK
for i=1:length(dt)
t = 0:dt(i):t_end;
```

```
p = 10./(1+9.*exp(-t));

y = RungeKutta(eq1,y0,dt(i),t_end);
E_RK(i) = sqrt((dt(i)/5)*sum((y-p).^2));
plot(t,y,'Color',Color{i},'LineStyle','-')
hold on
end
fprintf(['The following vector shows the errors obtained for comparing '...
         'the RK4 approximation to the analytical sol. on different steps.\n']);
E_RK
% Plot Configuration
plot(t,p,'Color',Color{length(dt)+1},'LineStyle','-')
xlabel('Time t')
legend(strcat('Runge-Kutta with dt =',' ',num2str(dt(1))),...
       strcat('Runge-Kutta with dt =',' ',num2str(dt(2))),...
       strcat('Runge-Kutta with dt =',' ',num2str(dt(3))),...
       strcat('Runge-Kutta with dt =',' ',num2str(dt(4))),...
       'Analytical Solution', 'Location','northwest')
title(['Comparison of Runge-Kutta approximations with respect to ' ...
       'time step and analytical solution'])
hold off
fprintf('Program paused. Proceed with Task d). Press enter to continue.\n\n');
pause;
```

```
The following vector shows the errors obtained for comparing the euler app

E_eul =

  Columns 1 through 3

   0.778316794729438   0.382716683345925   0.189844548631259

  Column 4

   0.094511192576505

Program paused. Press enter to continue.

The following vector shows the errors obtained for comparing the heun appr

E_heu =

  Columns 1 through 3

   0.274860456339433   0.071713282342125   0.018884377444475

  Column 4

   0.004882033620056

Program paused. Press enter to continue.

The following vector shows the errors obtained for comparing the RK4 appro
```

```
E_RK =

  Columns 1 through 3

   0.007894373573189   0.000522485844564   0.000034553781296

  Column 4

   0.000002233140232

Program paused. Proceed with Task d). Press enter to continue.
```



d)

error reduction factor calculation for increase in frequency

```
factor_eul = zeros(1,length(dt)-1);
factor_heu = zeros(1,length(dt)-1);
factor_RK = zeros(1,length(dt)-1);


for i=1:length(factor_eul)
    factor_eul(i) = E_eul(i)/E_eul(i+1);
    factor_heu(i) = E_heu(i)/E_heu(i+1);
    factor_RK(i) = E_RK(i)/E_RK(i+1);
```

```
end

factor_eul
factor_heu
factor_RK

fprintf('Program paused. Proceed with Task e). Press enter to continue.\n\n');
pause;
```

```
factor_eul =

   2.033663094916464   2.015947711457801   2.008699112304424


factor_heu =

   3.832769151858727   3.797492533337755   3.868137525086608


factor_RK =

  15.109258280065427  15.120945522170071  15.473180231698869

Program paused. Proceed with Task e). Press enter to continue.
```

e)

error calculation compared to best possible approximation Vector initialization and equation definition

```
eq1 = @(p)(1-(p/10))*p;
y0 = 1;
E_eul = zeros(1,length(dt));
E_heu = zeros(1,length(dt));
E_RK = zeros(1,length(dt));

% Color definition matrix for graphs
Color = {[1 1 0], [1 0 1], [0 1 1], [1 0 0], [0 1 0]};

% Euler
stepsize = 1;
for i=length(dt):-1:1
    ct = 1; %counter for for-loop
    y = Euler(eq1,y0,dt(i),t_end);
    diff_vec = zeros(1,length(y));
    if i == length(dt)
        p = y; % init of the p comparitor vector for first repetition

        % p represents Euler with the highest time resolution
    end

    for j=1:stepsize:t_end/dt(length(dt))+1 % with different resolutions
                                            % come different vector
```

```
                                                % lengths. this for loop
                                                % matches correct values in the
                                                % various vectors

            diff_vec(ct) = p(j)-y(ct);  % subtract from best result current
                                        % result for coresponding value


            ct = ct + 1;  % count on for y

        end

        E_eul(i) = sqrt((dt(i)/5)*sum((diff_vec).^2));
        % Final deviation calculation
        stepsize = stepsize * 2; % increases the step size for next itteration
    end
    fprintf(['The following vector shows the errors obtained for comparing '...
             'the euler approximation to the best Euler approx. on different steps.\n'
    E_eul
    fprintf('Program paused. Press enter to continue with Heun.\n\n');
    pause;

    % Heun - see Euler for comments
    stepsize = 1;
    for i=length(dt):-1:1
        ct = 1;
        y = Heun(eq1,y0,dt(i),t_end);
        diff_vec = zeros(1,length(y));
        if i == length(dt)
            p = y;
        end
        for j=1:stepsize:t_end/dt(length(dt))+1
            diff_vec(ct) = p(j)-y(ct);
            ct = ct + 1;
        end
        E_heu(i) = sqrt((dt(i)/5)*sum((diff_vec).^2));
        stepsize = stepsize * 2;
    end
    fprintf(['The following vector shows the errors obtained for comparing '...
             'the Heun approximation to the best Heun approx. on different steps.\n'])

    E_heu

    fprintf('Program paused. Press enter to continue Runge-Kutta.\n\n');
    pause;

    % Runge-Kutta  - see Euler for comments
    stepsize = 1;
    for i=length(dt):-1:1
        ct = 1;
        y = RungeKutta(eq1,y0,dt(i),t_end);
        diff_vec = zeros(1,length(y));
        if i == length(dt)
            p = y;
```

```
    end
    for j=1:stepsize:t_end/dt(length(dt))+1
        diff_vec(ct) = p(j)-y(ct);
        ct = ct + 1;
    end
    E_RK(i) = sqrt((dt(i)/5)*sum((diff_vec).^2));
    stepsize = stepsize * 2;
end
fprintf(['The following vector shows the errors obtained for comparing '...
         'the RK4 approximation to the best RK4 approx. on different steps.\n']);
E_RK

fprintf('Program paused. Press enter to continue.\n\n');
pause;
```

```
The following vector shows the errors obtained for comparing the euler app

E_eul =

  Columns 1 through 3

   0.686786187104003   0.288783327415189   0.095417074994865

  Column 4

                   0

Program paused. Press enter to continue with Heun.

The following vector shows the errors obtained for comparing the Heun appr

E_heu =

  Columns 1 through 3

   0.270084627204589   0.066815372027406   0.013990380006809

  Column 4

                   0

Program paused. Press enter to continue Runge-Kutta.

The following vector shows the errors obtained for comparing the RK4 appro

E_RK =

  Columns 1 through 3

   0.007892157260324   0.000520236879178   0.000032312975616

  Column 4

                   0
```

```
Program paused. Press enter to continue.
```

Published with MATLAB® R2013a