Table of Contents

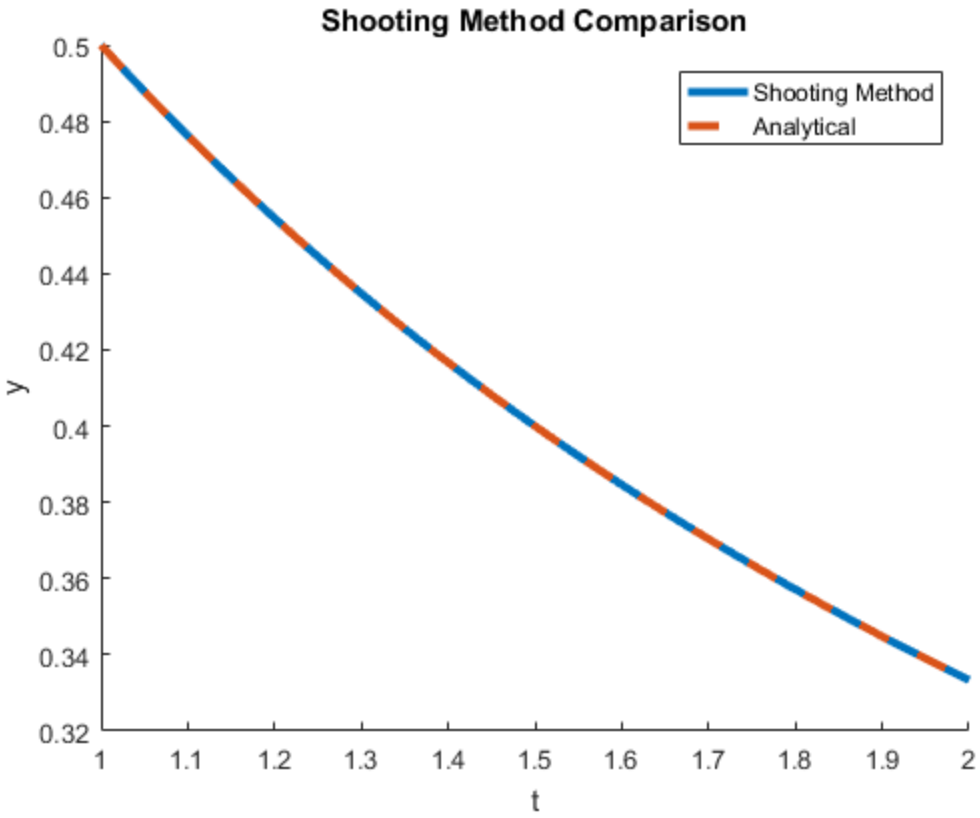
540.305 Problem Set 6:

```
clear
clc
```

1 Solving non-linear boundary value problems with the shooting method

```
% original ode: y''=y^3-y*y'
% as system: let x1=y, x2=y' and x1(1)=1/2, x1(2)=1/3:

%            x1'=y'=x2,
%            x2'=y''=y^3-y*y'=x1^3-x1*x2

x2_init = fsolve(@zero_at_x_initial,1/3);
[t,x] = ode45(@ode_system_1,[1 2],[1/2 x2_init]);
figure
hold on
plot(t,x(:,1),'LineWidth',3)
plot(t,1./(1+t),'--','LineWidth',3)
legend('Shooting Method', 'Analytical')
xlabel('t')
ylabel('y')
title('Shooting Method Comparison')
hold off
```

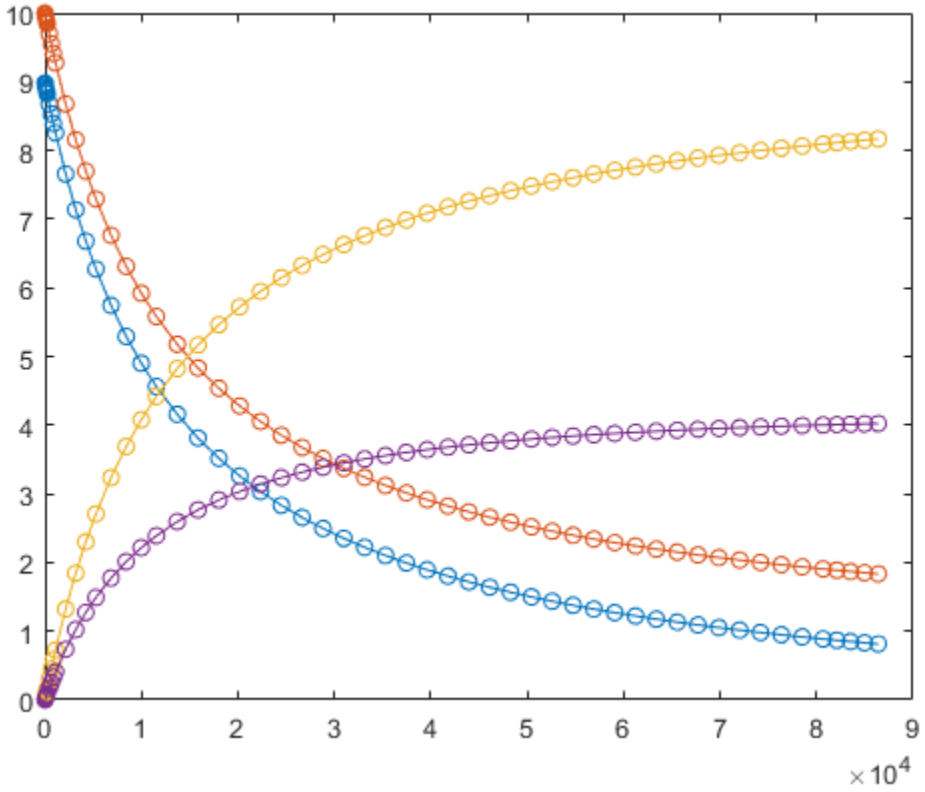
```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.
```



2 Optimizing initial conditions for a chemical reactor to maximize profit

```
tic
% function [prof]=profit(C)
k1=0.000008;
k2=0.000005;
money = [-100 -250 700 -450]*1000;
sys = @(t,c)[...
    -k1*c(1)*c(2);...
    -k1*c(1)*c(2);...
     k1*c(1)*c(2);...
     k2*c(1)^2;...
    ];
% y=getfield(ode45(sys, [0 24*60*60], [C(1) C(2) 0 0]),'y');
% prof=sum(y(:,end)'.*money);
% end

C=(fmincon(@(C) profit(C),[10 10],[],[],[],[],[0 0],[inf 10]));

ode45(sys, [0 24*60*60], [C(1) C(2) 0 0])
disp(['Profit: $',num2str(-profit(C))])
toc
```

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the default value of the optimality
 tolerance,
and constraints are satisfied to within the default value of the
 constraint tolerance.


Profit: $3366584.2461
Elapsed time is 0.182870 seconds.
```



3a Problem 2.1.14 - Paving Stones

```
stones = 600;
pcracked = 15/stones;
pdiscolored = 27/stones;
pperfect = 562/stones;

disp(['The probability that a random stone is cracked, disoclored, or
 both is: '...
    , num2str(1-pperfect)]);
disp(['The probability that a random stone is cracked and disoclored
 is: '...
```

```
    , num2str(pcracked+pdiscolored-(1-pperfect))]);
disp(['The probability that a random stone is cracked, but not
 disoclored is: '...
    , num2str((1-pperfect)-pdiscolored)]);
```

```
The probability that a random stone is cracked, disoclored, or both
 is: 0.063333
The probability that a random stone is cracked and disoclored is:
 0.0066667
The probability that a random stone is cracked, but not disoclored is:
 0.018333
```

3a Problem 2.1.18 - Human Blood

```
A = 0.35;
B = 0.10;
AB = 0.05;
O = 1-A-B+AB;

disp(['The probability that a random donor is type O is:
 ',num2str(O)]);
disp(['The probability that a random blood doesnt contain B is: '...
    , num2str(O+A-AB)]);
```

```
The probability that a random donor is type O is: 0.6
The probability that a random blood doesnt contain B is: 0.9
```

3b Problem 2.2.4 - Baseball

```
tp = 18; % total players
ppt = 9; % player per team

% since order does not matter:
disp(['Number of differnet teams is: ',num2str(nchoosek(18,9))]);
```

```
Number of differnet teams is: 48620
```

3b Problem 2.2.12 - Socks

```
disp(['Probaility two socks mathc is: '...
,num2str((nchoosek(6,2)+nchoosek(4,2)+nchoosek(2,2))/
nchoosek(12,2))]);
```

```
Probaility two socks mathc is: 0.33333
```

4 Heads in a row

```
trials = 100;
events = 500000;
k = [6 12];

for n = k
```

```
    tosses = zeros(trials, events,n);
    tosses(:,:,1) = round(rand(trials,events));

    for j = 2:n
        tosses(:,:,j) = circshift(tosses(:,:,1),j-1,2);
        tosses(:,1:j-1,2:n) = 0;
    end
    tosses = circshift(sum(tosses,3),-(n-1),2);
    tosses = tosses.*(tosses==n);
    tosses = tosses - circshift(tosses,1,2);
    tosses = tosses.*(tosses==n)/n;

    disp(['The probability that you flip at least ',num2str(n),' heads
 in a row is: ',num2str(mean(sum(tosses,2)/events))]);
end
```

```
The probability that you flip at least 6 heads in a row is: 0.0078346
The probability that you flip at least 12 heads in a row is:
 0.00012374
```

Published with MATLAB® R2016a