

MATLAB assignment 8

Introduction to Linear Algebra (Week 8)

Fall, 2019

1. A common problem in experimental work is to find a curve $y = f(x)$ of a specified form corresponding to experimentally determined values of x and y , say

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n).$$

The followings are the three important models in applications.

- **Linear line model** ($y = ax + b$)
- **Exponential model** ($y = ae^{bx}$)
- **Logarithmic model** ($y = a + b \ln x$)

A function file `LS_solver.m` fits given experimental data to the proper mathematical model using least squares method.

Problem.

- (a) Download the function file `LS_solver_guide.m` on KLMS and complete the missing parts. Save completed file as `LS_solver.m`. (10 points)
solution.

```
1 function [a, b]=LS_solver(x, y, opt)
2 % ----- function file LS_Solver ----- %
3 % input data: x, y and opt
4 %   if opt=1, linear model (y=a*x+b)
5 %   if opt=2, exponential model (y=a*exp(b*x))
6 %   if opt=3, logarithmic model (y=a+b*ln(x))
7
8 [m1, n1]=size(x); [m2, n2]=size(y); % Size of the input data.
9 xx=linspace(min(x), max(x), 100); % xx will be used to plot
   the fitting curve.
10
11 if (m1~=1) || (m2~=1) || (n1~=n2) % If the input data size is
   not proper,
12     fprintf('Error: Improper input data.\n'); % error message.
13 else
14     if (opt==1) || (opt==2) || (opt==3) % option = 1, 2, 3.
15         figure; plot(x, y, 'o'); % Plot the given data
           points.
16         hold on; % Ready to draw the
           next graph.
17         switch opt
18             case 1 % Linear model
19                 fprintf('Linear model\n'); % Print out the 'Linear
           model'.
20                 A=[x' ones(n1,1)]; Y=y'; % Set the least squares
           problem.
21                 sol=A\Y; % Find the least
           squares solution.
22                 a=sol(1); b=sol(2); % Fitting constants a
           and b.
```

```

23         plot(xx, a*xx+b);           % Plot the fitting
           curve with a and b.
24         title('Linear model (y=a*x+b)');
25     case 2                               % Exponential model
26         fprintf('Exponential model\n'); % Print out the '
           Exponential model'.
27         A=[ones(n1,1) x']; Y=log(y'); % Set the least
           squares problem.
28         sol=A\Y;                       % Find the least
           squares solution.
29         a=exp(sol(1)); b=sol(2);       % Fitting constants a
           and b.
30         plot(xx, a*exp(b*xx));        % Plot the fitting
           curve with a and b.
31         title('Exponential model (y=a*exp(b*x))');
32     case 3                               % Logarithmic model
33         fprintf('Logarithmic model\n'); % Print out the '
           Logarithmic model'.
34         A=[ones(n1,1) log(x)']; Y=y'; % Set the least
           squares problem.
35         sol=A\Y;                       % Find the least
           squares solution.
36         a=sol(1); b=sol(2);           % Fitting constants a
           and b.
37         plot(xx, a+b*log(xx));        % Plot the fitting
           curve with a and b.
38         title('Logarithmic model (y=a+b*ln(x))');
39     end
40     hold off;                          % No more graph.
41
42     else                               % for invalid [opt]
43         fprintf('Error: Improper option value.\n'); % error message
44
45         return;                        % Return the process.
46     end
end

```

- (b) Replace the switch-case command of the LS_solver.m with the if-else if-else command. (5 points)

solution.

```

1  function [a, b]=LS_solver_sol(x, y, opt)
2  % ----- function file LS_Solver ----- %
3  % input data: x, y and opt
4  %   if opt=1, linear model (y=a*x+b)
5  %   if opt=2, exponential model (y=a*exp(b*x))
6  %   if opt=3, logarithmic model (y=a+b*ln(x))
7
8  [m1, n1]=size(x); [m2, n2]=size(y); % Sizes of the input datas.
9  xx=linspace(min(x), max(x), 100); % xx will be used to plot
   the fitting curve.
10
11 if (m1~=1)|| (m2~=1)|| (n1~=n2) % If the input data size is
   not proper,
12     fprintf('Error: Improper input data.\n'); % print error

```

```

        message.
13 else
14     figure; plot(x, y, 'o');           % Plot the given data
        points.
15     hold on;                         % Ready to draw the next
        graph.
16     if opt == 1                       % Linear model
17         fprintf('Linear model\n');    % Print out the 'Linear
        model'.
18         A=[x' ones(n1,1)]; Y=y';    % Set the least squares
        problem.
19         sol=A\Y;                     % Find the least squares
        solution.
20         a=sol(1); b=sol(2);          % Fitting constants a and b
        .
21         plot(xx, a*xx+b);            % Plot the fitting curve
        with a and b.
22         title('Linear model (y=a*x+b)');
23     else
24         if opt == 2                  % Exponential model
25             fprintf('Exponential model\n'); % Print out the '
        Exponential model'.
26             A=[ones(n1,1) x']; Y=log(y'); % Set the least squares
        problem.
27             sol=A\Y;                 % Find the least squares
        solution.
28             a=exp(sol(1)); b=sol(2); % Fitting constants a and b
        .
29             plot(xx, a*exp(b*xx));   % Plot the fitting curve
        with a and b.
30             title('Exponential model (y=a*exp(b*x))');
31     else
32         if opt == 3                  % Logarithmic model
33             fprintf('Logarithmic model\n'); % Print out the '
        Logarithmic model'.
34             A=[ones(n1,1) log(x)']; Y=y'; % Set the least
        squares problem.
35             sol=A\Y;                 % Find the least
        squares solution.
36             a=sol(1); b=sol(2);      % Fitting constants a
        and b.
37             plot(xx, a+b*log(xx));   % Plot the fitting
        curve with a and b.
38             title('Logarithmic model (y=a+b*ln(x))');
39         else                          % for invalid [opt]
40             fprintf('Error: Improper option value.\n'); % error
        message.
41         return                       % Return the process.
42     end
43 end
44 end
45 hold off;                           % No more graph.
46 end

```