David Lim

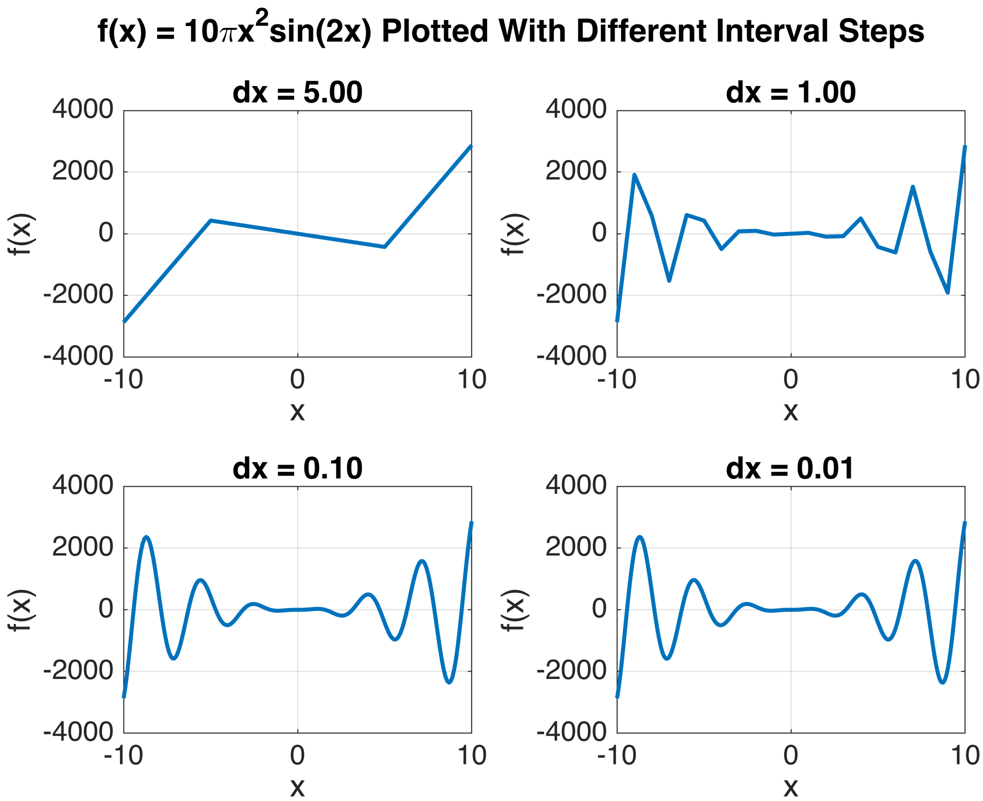
A16398479

MAE 150

07/07/23

Homework 1

Problem 1



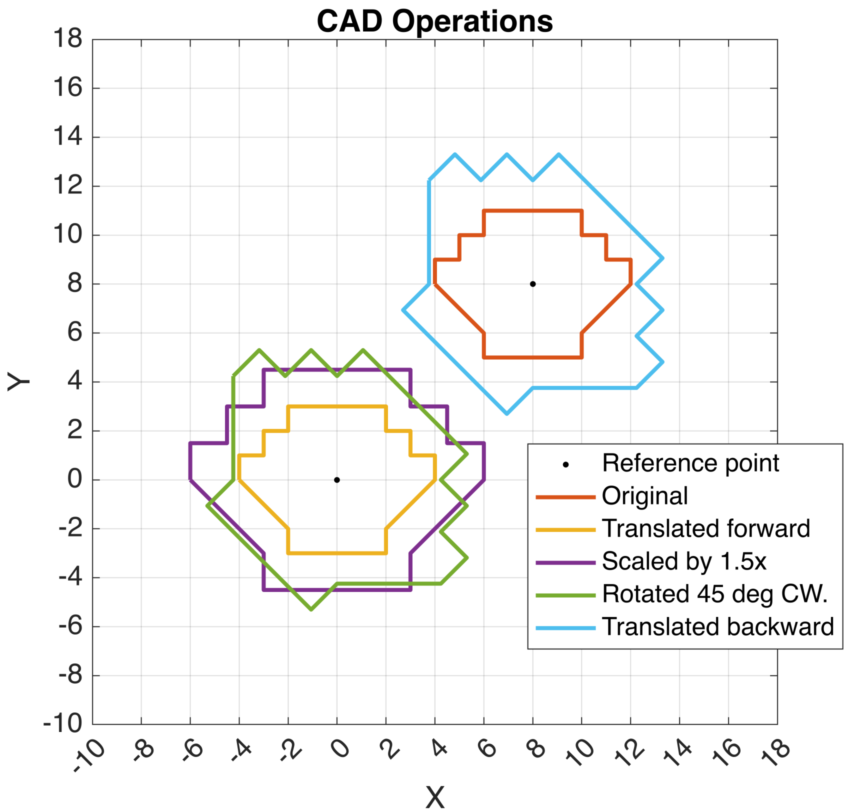
The first subplot with interval step dx = 5 only evaluates the function at 5 points in the range. As a result, the plotted function appears to be piecewise when the acutal function is smooth. Much of the details of the actual function, such as local extrema, slope, concavity, and inflection points, are not represented in this subplot.

The second subplot with interval step dx = 1 evaluates the function at 21 points in the range. As a result, the plotted funtion still appears to be piecewise rather than smooth. However, some of the details of the acutal function begin to emerge, such as the relative slopes and relative locations of the local extrema.

The third subplot with interval step dx = 0.1 evaulates the function at 201 points in the range. As a result, the plotted function appears smooth. The "roughness" of the plotted function is only apparent when zoomed in. The details of the actual function are accurately represented in this subplot.

The fourth subplot with interval step dx = 0.01 evaulates the function at 2001 points in the range. As a result, the plotted function appears smooth. The plotted funtion almost appears to be smooth even when zoomed in. The details of the actual function are accurately represented in this subplot.

Problem 2

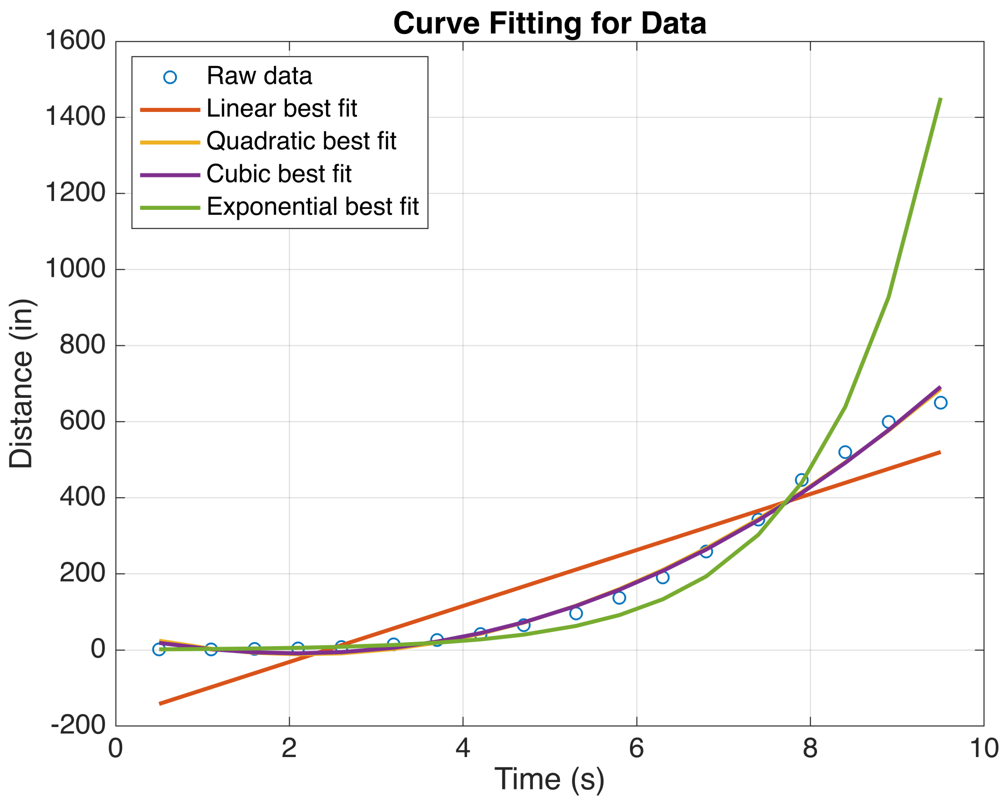



```
Forward translation matrix:
T_f = 3x3
     1     0    -8
     0     1    -8
     0     0     1
Scaling matrix:
S = 3x3
    1.5000         0         0
         0    1.5000         0
         0         0    1.0000
Rotation matrix:
R = 3x3
    0.7071    0.7071         0
   -0.7071    0.7071         0
         0         0    1.0000
Backward translation matrix:
T_b = 3x3
     1     0     8
     0     1     8
     0     0     1
Overall transformation matrix:
M = 3x3
    1.0607    1.0607   -8.9706
   -1.0607    1.0607    8.0000
         0         0    1.0000
```

Problem 3

```
(a) Best Fit Coefficients
Linear Fit:
a_0 = -178.5232
a_1 = 73.5824
Quadratic Fit:
a_0 = 49.1043
a_1 = -55.9558
a_2 = 12.9538
Cubic Fit:
a_0 = 38.6522
a_1 = -44.8905
a_2 = 10.2614
a_3 = 0.1795
Exponential Fit:
a_0 = 1.2131
a_1 = 0.7460
```

```
(b)
```




```
(c) Best Fit Coefficients Using Polyfit
Linear Fit:
a_0 = -178.5232
a_1 = 73.5824
Quadratic Fit:
a_0 = 49.1043
a_1 = -55.9558
a_2 = 12.9538
Cubic Fit:
a_0 = 38.6522
a_1 = -44.8905
a_2 = 10.2614
```

```
a_3 = 0.1795
Exponential Fit:
a_0 = 1.2131
a_1 = 0.7460

(d) RMS Error of Each Fit
Linear Fit:
RMS_error = 87.9899
Quadratic Fit:
RMS_error = 18.8620
Cubic Fit:
RMS_error = 18.6452
Exponential Fit:
RMS_error = 207.8426
```

(c) The coefficients calculated in part (a) are equal to the corresponding coefficients in part (c).

(e) The cubic best fit is the best fit line for this data set because it has the least RMS error and the shape of the function matches the overall pattern of the data. A higher order approximation may have less RMS error but the shape of higher order best fit lines may not match the true physical relationship due to overfitting to the data.

Problem 1 Code

```
clear
close all

intervals = [5 1 0.1 0.01]; % define interval step sizes

for i = 1:4 % loop 4 times for each interval size
    % Evaluating
    dx = intervals(i);
    x = -10:dx:10; % inputs
    f = 10*pi*x.^2.*sin(2*x); % outputs
    % Plotting
    subplot(2,2,i)
    plot(x,f,'LineWidth',2)
    title(sprintf('dx = %0.2f',dx))
    xlabel('x')
    ylabel('f(x)')
    set(gca,'fontsize',14)
    grid on
end
sgtitle('f(x) = 10\pix^2sin(2x) Plotted With Different Interval
 Steps','fontsize',16,'fontweight','bold')
```

Published with MATLAB® R2023a

Problem 2 Code

```
clear
close all

% Define points as matrix of vectors in homogeneous coordinates
shape0 = [4 6 6 10 10 12 12 11 11 10 10  6  6  5 5 4 4;
          8 6 5  5  6  8  9  9 10 10 11 11 10 10 9 9 8;
          ones(1,17)];
x_0 = 8;
y_0 = 8;

T_f = [1 0 -x_0;
       0 1 -y_0;
       0 0    1]; % homogeneous forward translation matrix
shape1 = T_f*shape0;

s_x = 1.5;
s_y = 1.5;
S = [s_x   0 0;
       0 s_y 0;
       0   0 1]; % homogenous scaling matrix
shape2 = S*shape1;

theta = -45;
R = [cosd(theta) -sind(theta) 0;
     sind(theta)  cosd(theta) 0;
               0            0 1]; % homogeneous rotation matrix
shape3 = R*shape2;

T_b = [1 0 x_0;
       0 1 y_0;
       0 0 1]; % homogeneous beackward translation matrix
shape4 = T_b*shape3;

% Plotting
linewidth = 2;
plot([x_0 0],[y_0 0],'k.','MarkerSize',10)
hold on
plot(shape0(1,:),shape0(2,:),'LineWidth',linewidth)
plot(shape1(1,:),shape1(2,:),'LineWidth',linewidth)
plot(shape2(1,:),shape2(2,:),'LineWidth',linewidth)
plot(shape3(1,:),shape3(2,:),'LineWidth',linewidth)
plot(shape4(1,:),shape4(2,:),'LineWidth',linewidth)
hold off
title('CAD Operations')
xlabel('X')
ylabel('Y')
legend('Reference point','Original','Translated forward','Scaled
 by 1.5x','Rotated 45 deg CW.','Translated backward','Position',
[0.6,0.2,0.28214,0.24405])
grid on
```

```
axis equal
axis([-10 18 -10 18])
xticks(-10:2:18)
yticks(-10:2:18)
set(gca,'FontSize',14)

% Print matrices
fprintf('Forward translation matrix:\n')
T_f
fprintf('Scaling matrix:\n')
S
fprintf('Rotation matrix:\n')
R
fprintf('Backward translation matrix:\n')
T_b
fprintf('Overall transformation matrix:\n')
M = T_b*R*S*T_f
```

Published with MATLAB® R2023a

Table of Contents

Problem 3 Code

```
clear
close all

t = [0.5 1.1 1.6 2.1 2.6 3.2 3.7 4.2 4.7 5.3 5.8 6.3 6.8 7.4 7.9 8.4 8.9 9.5];
d = [1.3 1.8 2.9 4 8 15 26 42 65 96 137 191 259 343 447 520 600 650];
M = length(t);

fprintf('(a) Best Fit Coefficients\n')

% Linear fit
A = [     M    sum(t);
     sum(t) sum(t.^2)];
b = [sum(d);
     sum(d.*t)];
a = A\b;
fprintf('Linear Fit:\n')
a_0 = a(1)
a_1 = a(2)
d_linear = a_0 + a_1*t;

% Quadratic fit
A = [        M    sum(t) sum(t.^2);
        sum(t) sum(t.^2) sum(t.^3);
     sum(t.^2) sum(t.^3) sum(t.^4)];
b = [      sum(d);
        sum(d.*t);
     sum(d.*t.^2)];
a = A\b;
fprintf('Quadratic Fit:\n')
a_0 = a(1)
a_1 = a(2)
a_2 = a(3)
d_quad = a_0 + a_1*t + a_2*t.^2;

% Cubic fit
A = [        M    sum(t) sum(t.^2) sum(t.^3);
        sum(t) sum(t.^2) sum(t.^3) sum(t.^4);
     sum(t.^2) sum(t.^3) sum(t.^4) sum(t.^5);
     sum(t.^3) sum(t.^4) sum(t.^5) sum(t.^6)];
b = [sum(d);
     sum(d.*t);
     sum(d.*t.^2);
```

```
     sum(d.*t.^3)];
a = A\b;
fprintf('Cubic Fit:\n')
a_0 = a(1)
a_1 = a(2)
a_2 = a(3)
a_3 = a(4)
d_cube = a_0 + a_1*t + a_2*t.^2 + a_3*t.^3;

% Exponential Fit
A = [     M    sum(t);
     sum(t) sum(t.^2)];
b = [sum(log(d));
     sum(log(d).*t)];
a = A\b;
fprintf('Exponential Fit:\n')
a_0 = exp(a(1))
a_1 = a(2)
d_exp = a_0*exp(a_1*t);
```

(b) Plotting

```
fprintf(' \n')
fprintf('(b)')
linewidth = 2;
plot(t,d,'o','LineWidth',linewidth)
hold on
plot(t,d_linear,'LineWidth',linewidth)
plot(t,d_quad,'LineWidth',linewidth)
plot(t,d_cube,'LineWidth',linewidth)
plot(t,d_exp,'LineWidth',linewidth)
hold off
title('Curve Fitting for Data')
xlabel('Time (s)')
ylabel('Distance (in)')
legend('Raw data','Linear best fit','Quadratic best fit','Cubic best
 fit','Exponential best fit','Location','northwest')
grid on
set(gca,'FontSize',14)
```

(c) Using Polyfit

```
fprintf(' \n')
fprintf('(c) Best Fit Coefficients Using Polyfit\n')
% Linear fit
a = polyfit(t,d,1);
fprintf('Linear Fit:\n')
a_0 = a(2)
a_1 = a(1)
% Quadratic fit
a = polyfit(t,d,2);
fprintf('Quadratic Fit:\n')
a_0 = a(3)
```

```
a_1 = a(2)
a_2 = a(1)
% Cubic fit
a = polyfit(t,d,3);
fprintf('Cubic Fit:\n')
a_0 = a(4)
a_1 = a(3)
a_2 = a(2)
a_3 = a(1)
% Exponential fit
a = polyfit(t,log(d),1);
fprintf('Exponential Fit:\n')
a_0 = exp(a(2))
a_1 = a(1)
```

(d) RMS Error

```
fprintf(' \n')
fprintf('(d) RMS Error of Each Fit\n')
% Linear fit
fprintf('Linear Fit:\n')
RMS_error = sqrt(sum((d-d_linear).^2)/M)
% Quadratic fit
fprintf('Quadratic Fit:\n')
RMS_error = sqrt(sum((d-d_quad).^2)/M)
% Cubic fit
fprintf('Cubic Fit:\n')
RMS_error = sqrt(sum((d-d_cube).^2)/M)
% Exponential fit
fprintf('Exponential Fit:\n')
RMS_error = sqrt(sum((d-d_exp).^2)/M)
```

Published with MATLAB® R2023a