Environment: Matlab 2019

1.

* Run `problem_1.m` and see the result.

```
>> problem_1
By linear regression, assume y=mx+b,
can find the equation y=-0.113845x+32.065392, where m = -0.113845, b = 32.065392
By quadratic regression, assume y=a0x^2+a1x+a2,
can find the equation y=-0.025579x^2+0.391100x+30.494211, where a0 = -0.025579, a1 = 0.391100, a2 = 30.494211
```

Use the formula to find the coefficients.

Linear regression:

```
n = length(x);
sigma_x = sum(x);
sigma_y = sum(y);
sigma_xx = sum(x(:).^2);
sigma_xy = sum(x(:).*y(:));
m = (sigma_x*sigma_y - n*sigma_xy)/(sigma_x*sigma_x - n*sigma_xx);
b = (sigma_y - m* sigma_x)/n;
```

Quadratic regression:

```
X = [ones(1,n).' x(:), x(:).^2];
A = inv(transpose(X)*X)*transpose(X)*transpose(y);
```

2.

* Run `problem_2.m` and see the result.

```
>> problem_2
Use Fletcher-Reeves algorithm,
By the linear regression model, assume y=mx+b,
can find the equation y=-0.113845x+32.065392, where m = -0.113845, b = 32.065392
By the quadratic regression model, assume y=a0x^2+a1x+a2,
can find the equation y=-0.025579x^2+0.391100x+30.494211, where a0 = -0.025579, a1 = 0.391100, a2 = 30.494211

Use DFP method,
By the linear regression model, assume y=mx+b,
can find the equation y=-0.113845x+32.065392, where m = -0.113845, b = 32.065392
By the quadratic regression model, assume y=a0x^2+a1x+a2,
can find the equation y=-0.025579x^2+0.391100x+30.494211, where a0 = -0.025579, a1 = 0.391100, a2 = 30.494211
```

* Set tolerance = 10^(-6).
* Use DFP as the quasi-Newton method.

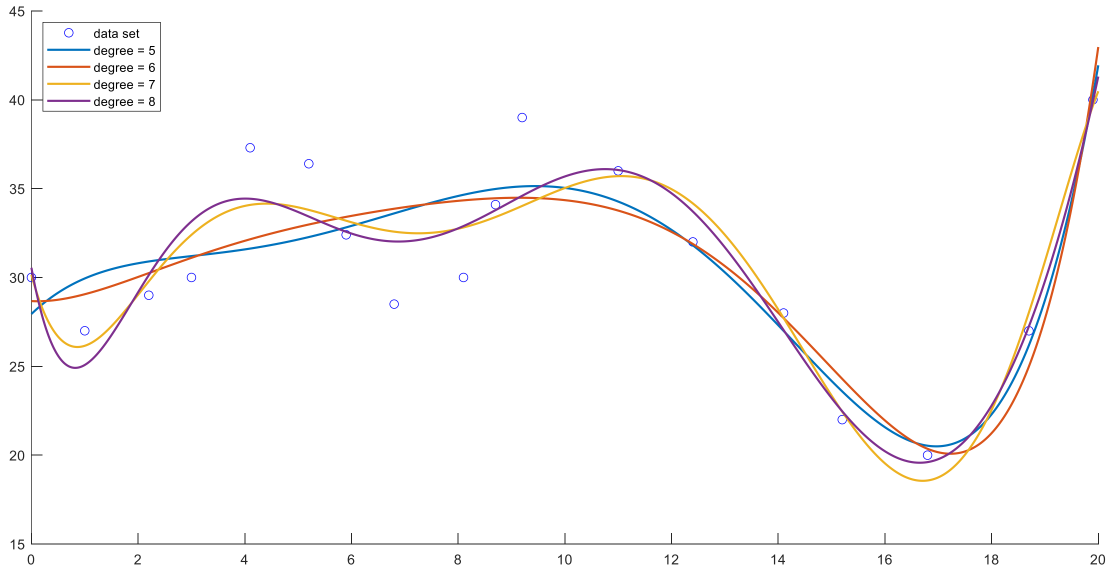
3.

When <u>order = 11</u>, the polynomial reasonably fits the data.

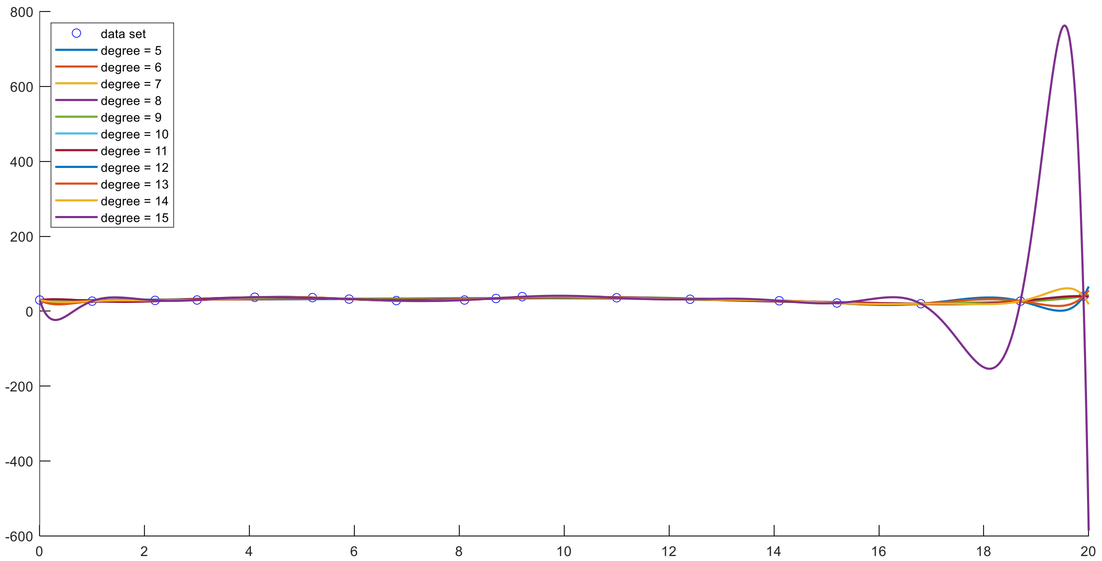
* Run `problem_3.m` will show the order from 10 to 14.

<Explain>

Initially set show the order from 5 to 8.

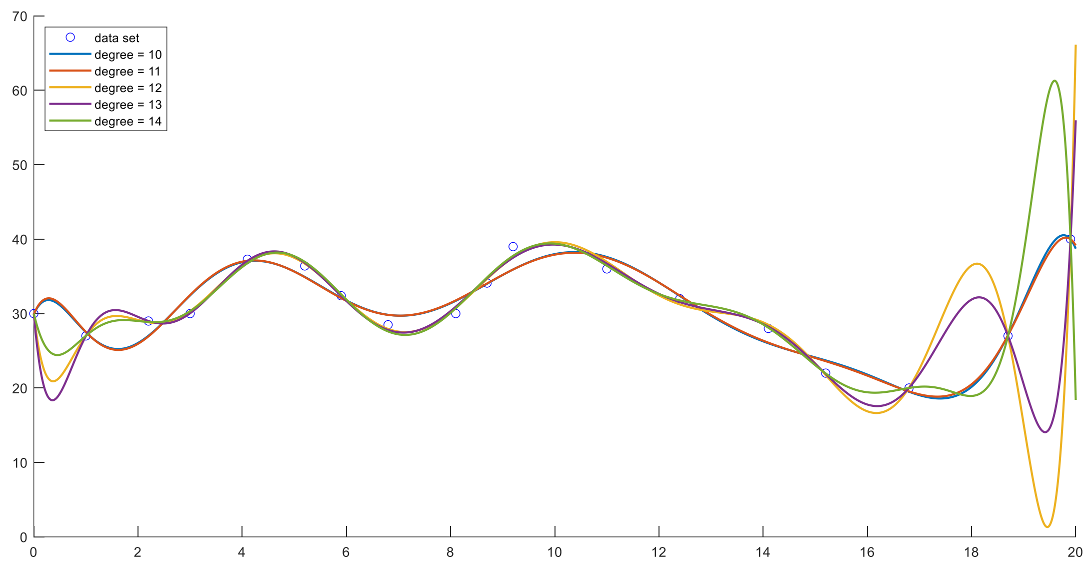


The figure shows that the higher degree can get better-fitting data. For instance, degree=5 can't fit the two high peaks in [4,10]. However, degree=8 can fit the data. Therefore, I try to increase the degree to find the best-fitting data. Secondly, choose the maximum degree = 15, and show the picture with degree from 5 to 15.

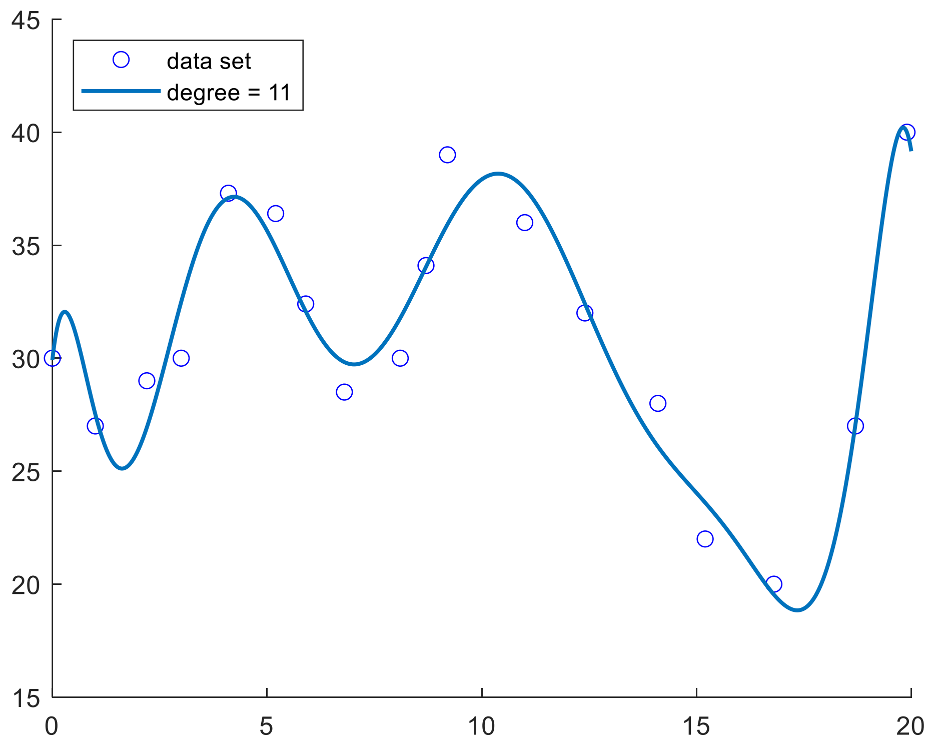


From the figure above, when degree=15, although the regression model fits the data, it become over-fitting in [16,20].

Eliminate degree = 15 and limit the degree from 10 to 14. Plot the figure again.



The figure indicates that when order=12(yellow line), the function starts to distort. Therefore, when order=11, the polynomial best fits the data without distortion, and show the following figure.



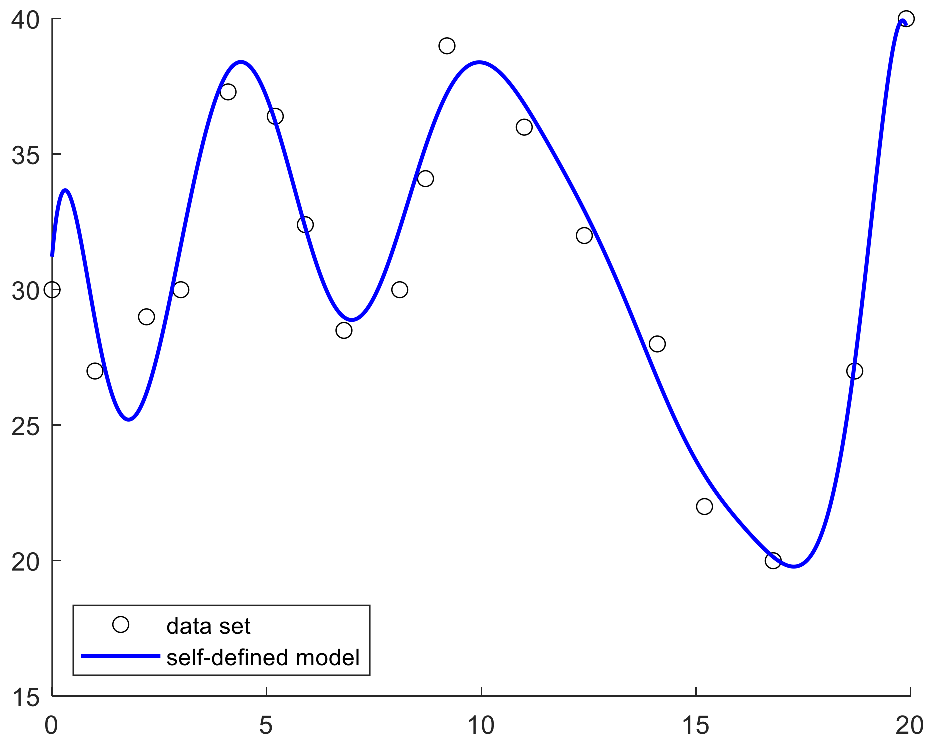
4.

Find the regression model:

F(x) =

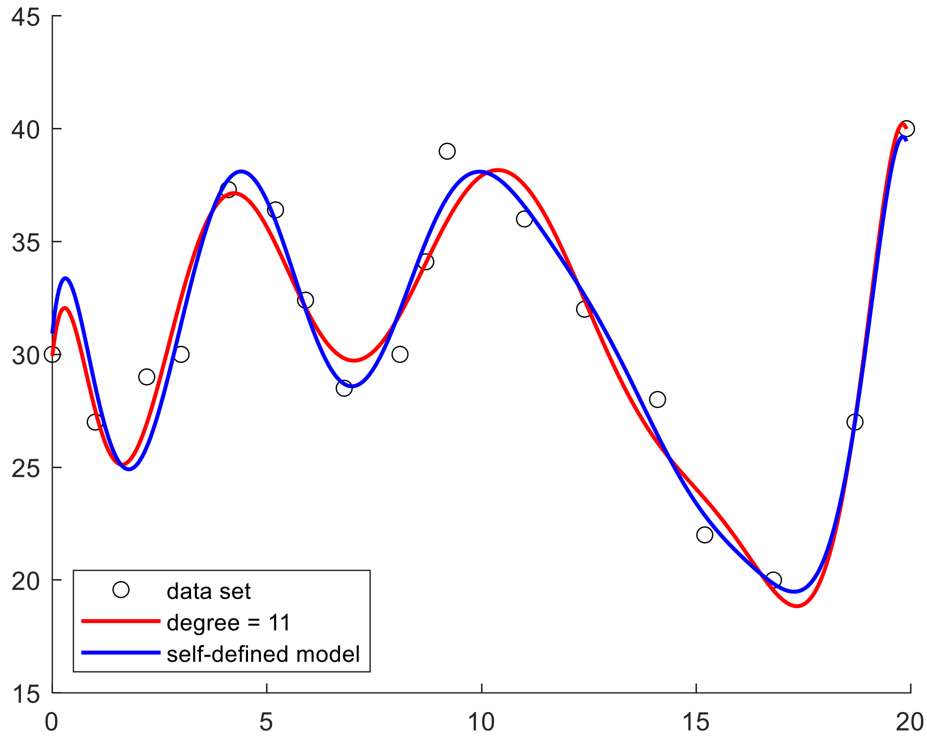
16.836177210758275890611912473105*x + cos((3*x)/2) + sin((29*x)/20) -
40.578444470932673482366226380691*x^2 +
29.956409607170890296856669010594*x^3 -
10.414812343524964433072454994544*x^4 +
2.0285399620751145555175298795803*x^5 -
0.23965967336217394723085760688264*x^6 +
0.017716791014445953944544953628792*x^7 -
0.00081479420058734152842311315723123*x^8 +
0.000022195548828431577707441177405023*x^9 -
0.00000031527897727677510574713641740285*x^10 +
0.0000000016361746466989936768189979591115*x^11 +
30.220808962382057671902657602914

* Run `problem_4.m` will show the figure.



Pick degree =11 from polynomial regression first. To let data fit better, let the function move left and enlarge the amplitude.

Therefore, add sin(alpha*x)+cos(beta*x). After several experiments, choose sin(1.45*x) + cos(1.5*x). And the result will be:



Finally, add a constant to move the function up. Set constant as 0.3