Linear Regression

Group Index: 38

Group members: Bârzan Emma, Cozac Antonio, Mureșan Mihai

Problem at hand

Our goal was to build a model that accurately predicts the output based on the given input, such that there is a very low error between the approximation and the output. We used polynomials of selectable degree as basis functions.

We have portrayed the implementation of multiple linear regression with two input variables.

The approximator

The approximator must have the following shape:

$$\hat{y}(k) = \phi^{\mathsf{T}}(k)\theta$$

In a system form, it will be:

$$\begin{bmatrix} y(1) \\ y(2) \\ \dots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_n(1) \\ \phi_1(2) & \phi_2(2) & \dots & \phi_n(2) \\ \dots & \dots & \dots & \dots \\ \phi_1(N) & \phi_2(N) & \dots & \phi_n(N) \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dots \\ \theta_n \end{bmatrix}$$

The regressors $\phi(k)$ will be determined individually for each dataset, but the parameters θ will be based on the identification dataset by matrix division between the approximator and the regressor.

Key Feature 1: Regressor implementation

We noticed that the powers of any two “pair” elements in each polynomial add up to the current degree, thus leading to a combinations list that must be trimmed:

DEGREE	POLYNOMIAL	POWERS
$m = 1,$	$\hat{g}(x) = [1,\ x_1,\ x_2] \cdot \theta$	$\implies \begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$m = 2,$	$\hat{g}(x) = [1,\ x_1,\ x_2,\ x_1^2,\ x_2^2,\ x_1x_2] \cdot \theta$	$\implies \begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{bmatrix}$
$m = 3,$	$\hat{g}(x) = [1,\ x_1,\ x_2,\ x_1^2,\ x_2^2,\ x_1^3,\ x_2^3,\ x_1x_2,\ x_1^2x_2,\ x_1x_2^2] \cdot \theta$	$\implies \begin{bmatrix} \text{pow}_{x_1} \\ \text{pow}_{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 & 3 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 2 & 0 & 3 & 1 & 1 & 2 \end{bmatrix}$

Key Feature 2: Reshaping the dataset output

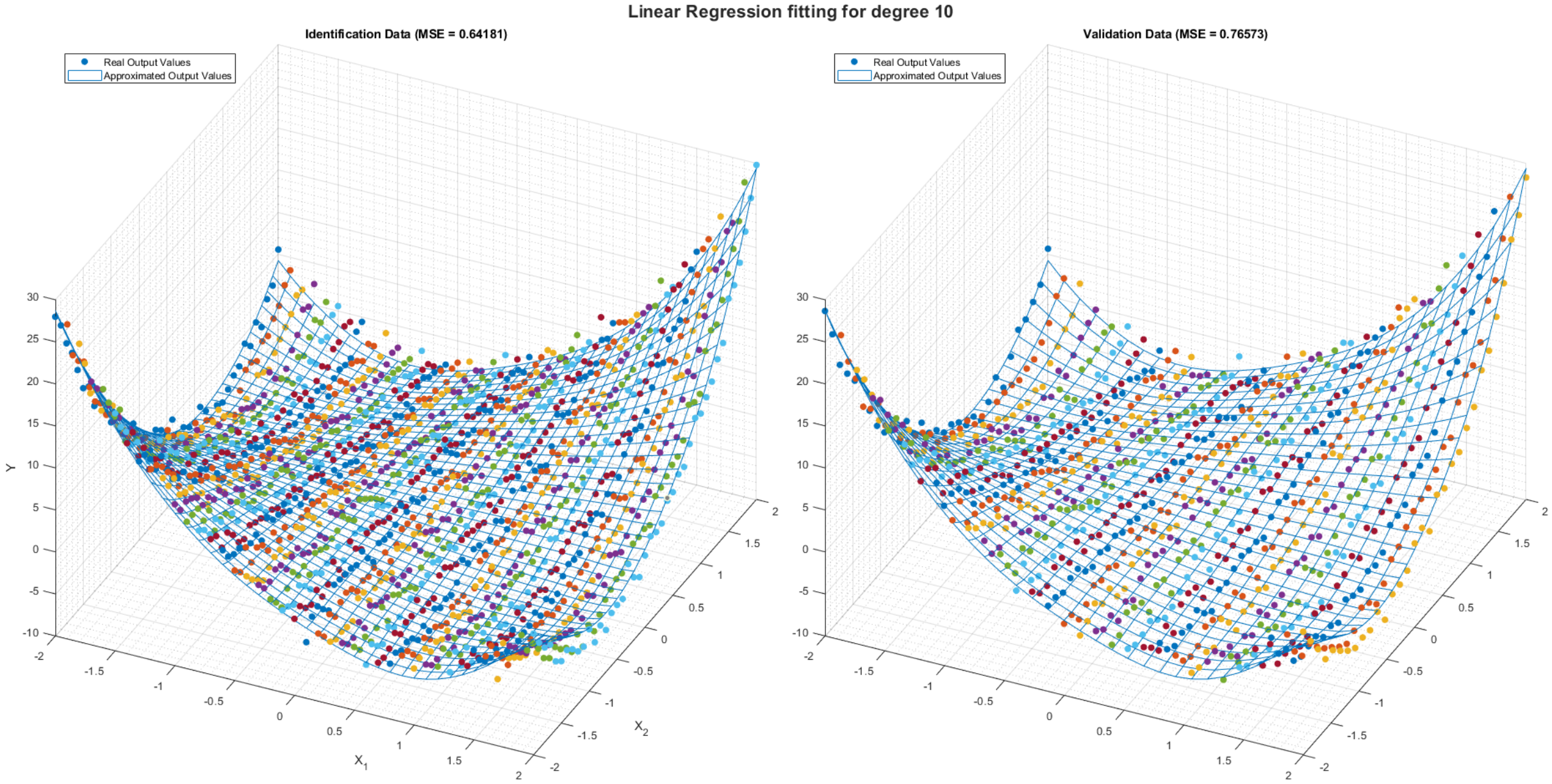
The datasets' outputs are shaped in an $N \times N$ matrix, but our system requires the output to be a column vector.

Output prior to reshape	Output post reshape
$\begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & \cdots & y_{1,N} \\ y_{2,1} & y_{2,2} & y_{2,3} & \cdots & y_{2,N} \\ y_{3,1} & y_{3,2} & y_{3,3} & \cdots & y_{3,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{N,1} & y_{N,2} & y_{N,3} & \cdots & y_{N,N} \end{bmatrix}$	$\begin{bmatrix} y_{1,1} \\ y_{1,2} \\ y_{1,3} \\ \cdots \\ y_{N,N} \end{bmatrix}$

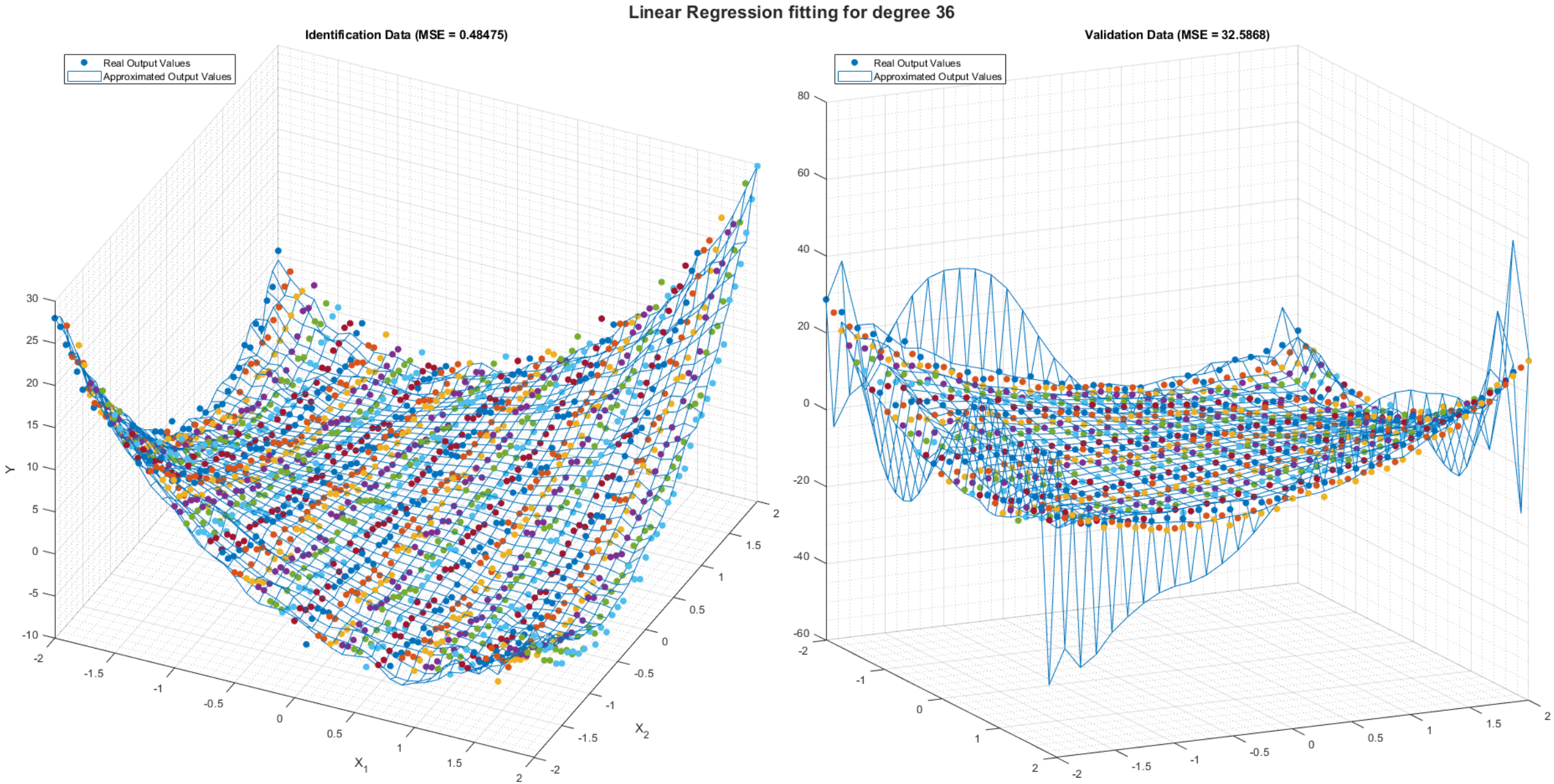
Key Feature 3: Runtime Optimization

- Given dataset is massive and creates even larger regressors
- Utilizing Matlab's computation speed for vectorization
- Possible use of multithreading (via parallel pools); if it is already active, it may increase performance for small datasets, but almost halves the performance time on larger sets. If inactive, greatly increases the computation time

Tuning results



Tuning results - overfitting



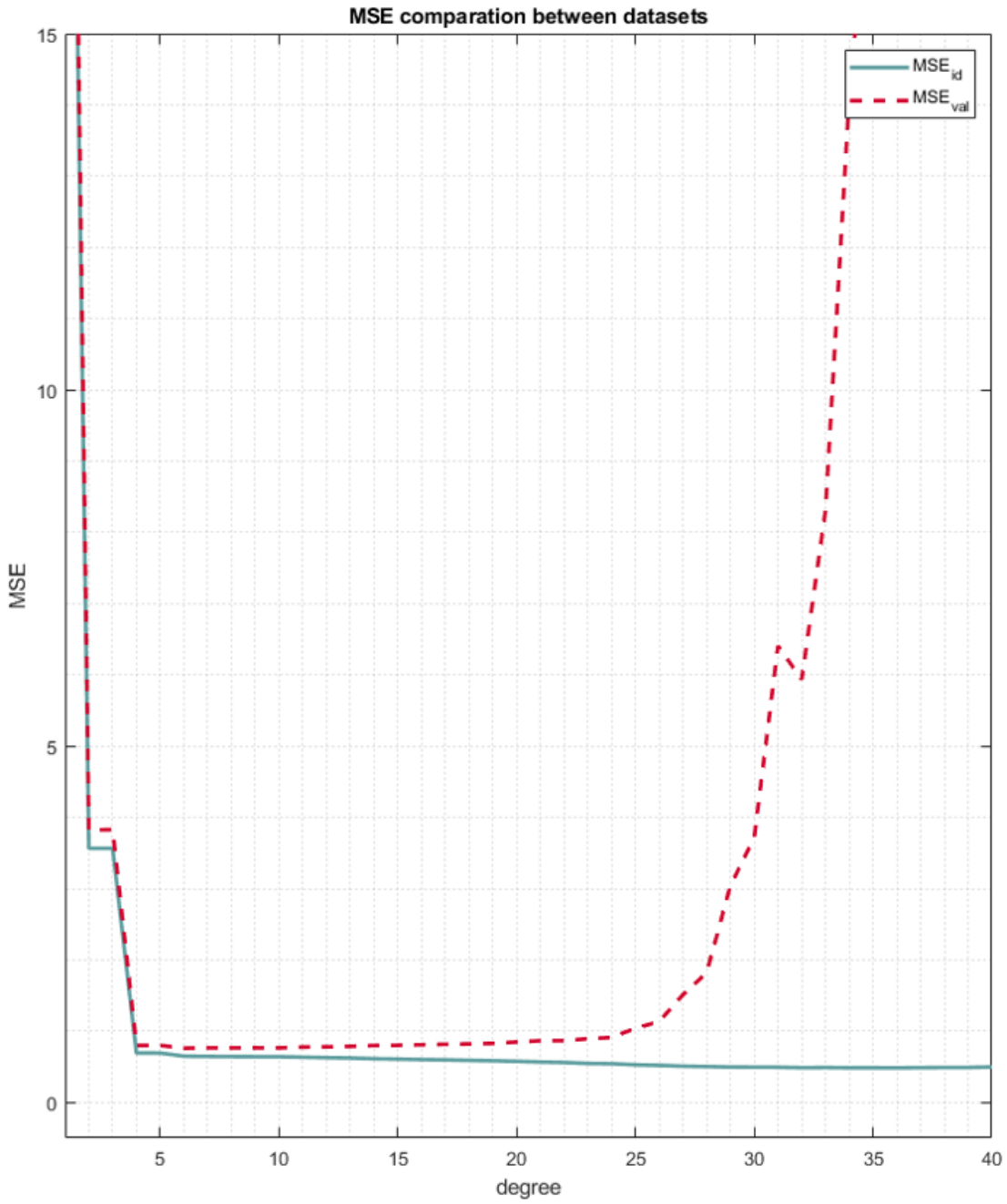
Tuning results

- Degree 1 to 3 are too low, not have enough information to create an accurate model
- Degree 6 to 10 are best cases where both of the datasets' MSEs are low
- Degree 25 shows major increase in the validation dataset's MSE
- Degree 36 is "best" for the identification set, but it overfits and models the possible noise too

Degree	MSE_id	MSE_val
1	27.63	29.09
2	3.57	3.83
3	3.57	3.83
4	0.69	0.80
5	0.69	0.80
6	0.65	0.76
7	0.65	0.77
8	0.64	0.77
9	0.64	0.77
10	0.64	0.77
11	0.63	0.78
12	0.63	0.78
13	0.62	0.79
14	0.61	0.80
15	0.61	0.80
16	0.60	0.81
17	0.60	0.82
18	0.59	0.82
19	0.58	0.83
20	0.58	0.85
21	0.57	0.87
22	0.56	0.87
23	0.55	0.89
24	0.54	0.91
25	0.53	1.04
26	0.52	1.13
27	0.51	1.51
28	0.50	1.82
29	0.50	3.04
30	0.50	3.71
31	0.49	6.41
32	0.49	5.94
33	0.49	8.30
34	0.49	14.11
35	0.49	17.81
36	0.48	32.59
37	0.49	51.71
38	0.49	125.70
39	0.49	118.95
40	0.50	108.07

Plots for optimal m

- Degree range 1 to 40 was considered to show overfitting error increasing
- Degrees from 1 to 25 would have been sufficient to show tendency



Conclusion

- Any degrees between 6 and 10, including those, result in a low approximation error