SYSTEM IDENTIFICATION

Polynomial Approximation

By Peter Alexia, Vilau Paul and Tivadar Razvan

12/8

Problem statement

GOAL

Develop a polynomial model to approximate an unknown, nonlinear static function.

DATA PROVIDED

Identification data and validation data for I/O

REQUIREMENTS

Finding the best fit for the unknown function

Polynomial Approximator Structure

$$
\begin{aligned}
y(k) &= \theta_1 + \theta_2 x_1(k) + \theta_3 x_2(k) + \theta_4 x_1^2(k) + \theta_5 x_2^2(k) + \theta_6 x_1(k) x_2(k) \\
&= \begin{bmatrix} 1 & x_1(k) & x_2(k) & x_1^2(k) & x_2^2(k) & x_1(k)x_2(k) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} \\
&= \phi^\top(x(k))\theta = \phi^\top(k)\theta
\end{aligned}
$$

How to find θ parameters

1. Calculate regressor matrix Φ

$$\phi = \phi(x_1, x_2)$$

2. Calculate θ column vector

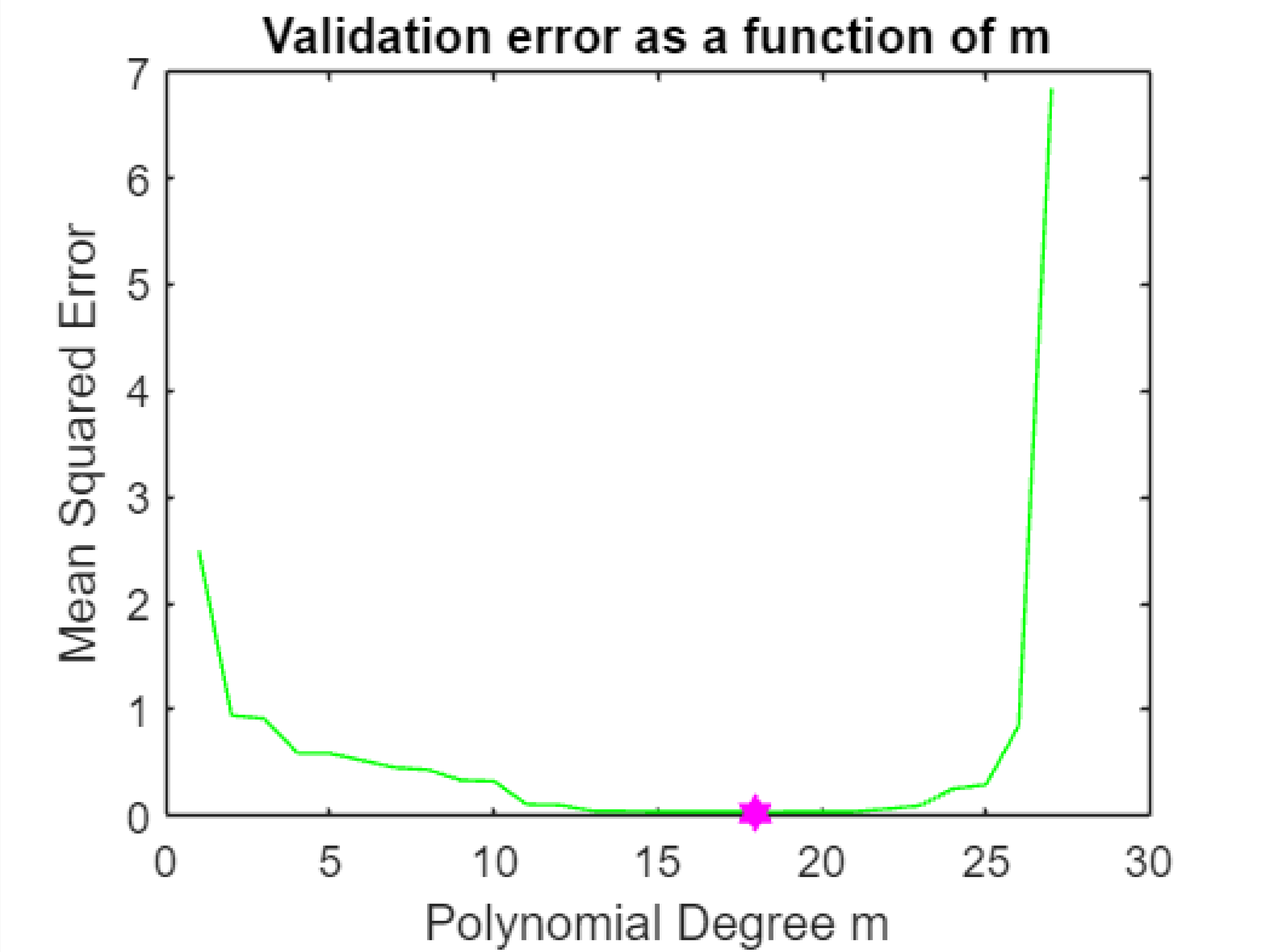
$$\theta = (\phi^\top \phi)^{-1} \phi^\top Y$$

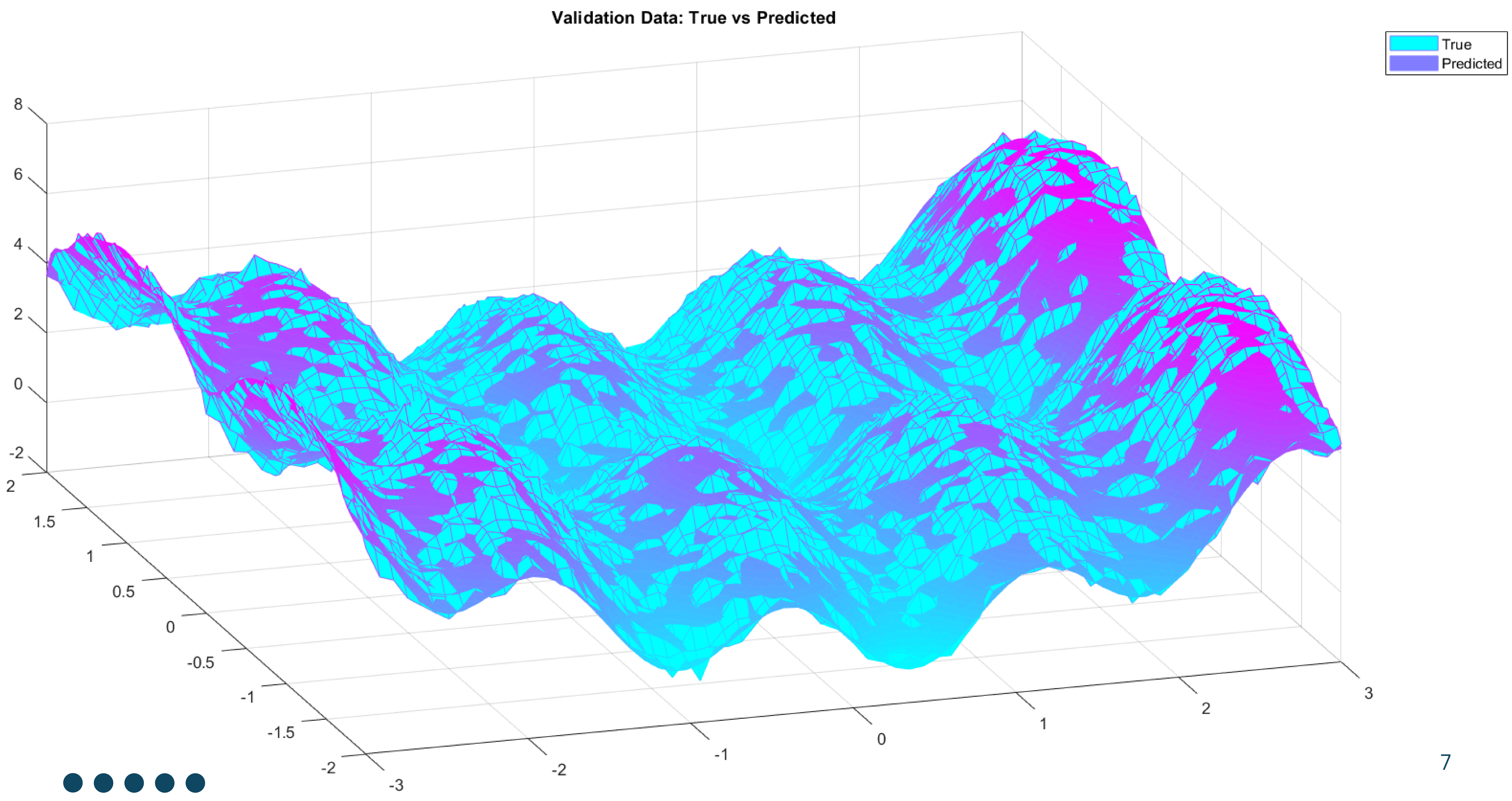
or

$$\theta = \phi \backslash Y$$

Key features

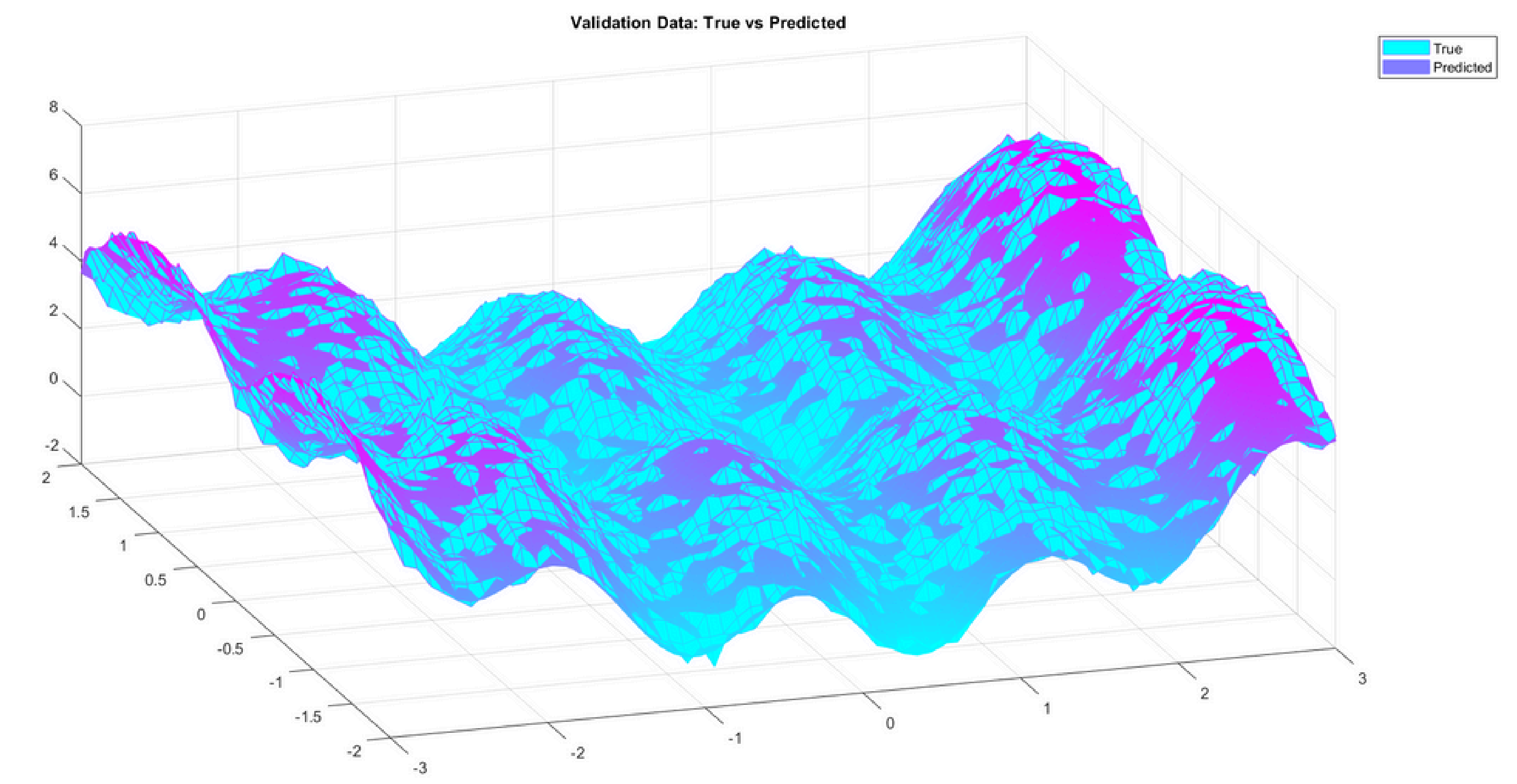
- Custom polynomial basis expansion
- Adaptive degree selection
- Efficient polynomial generating function

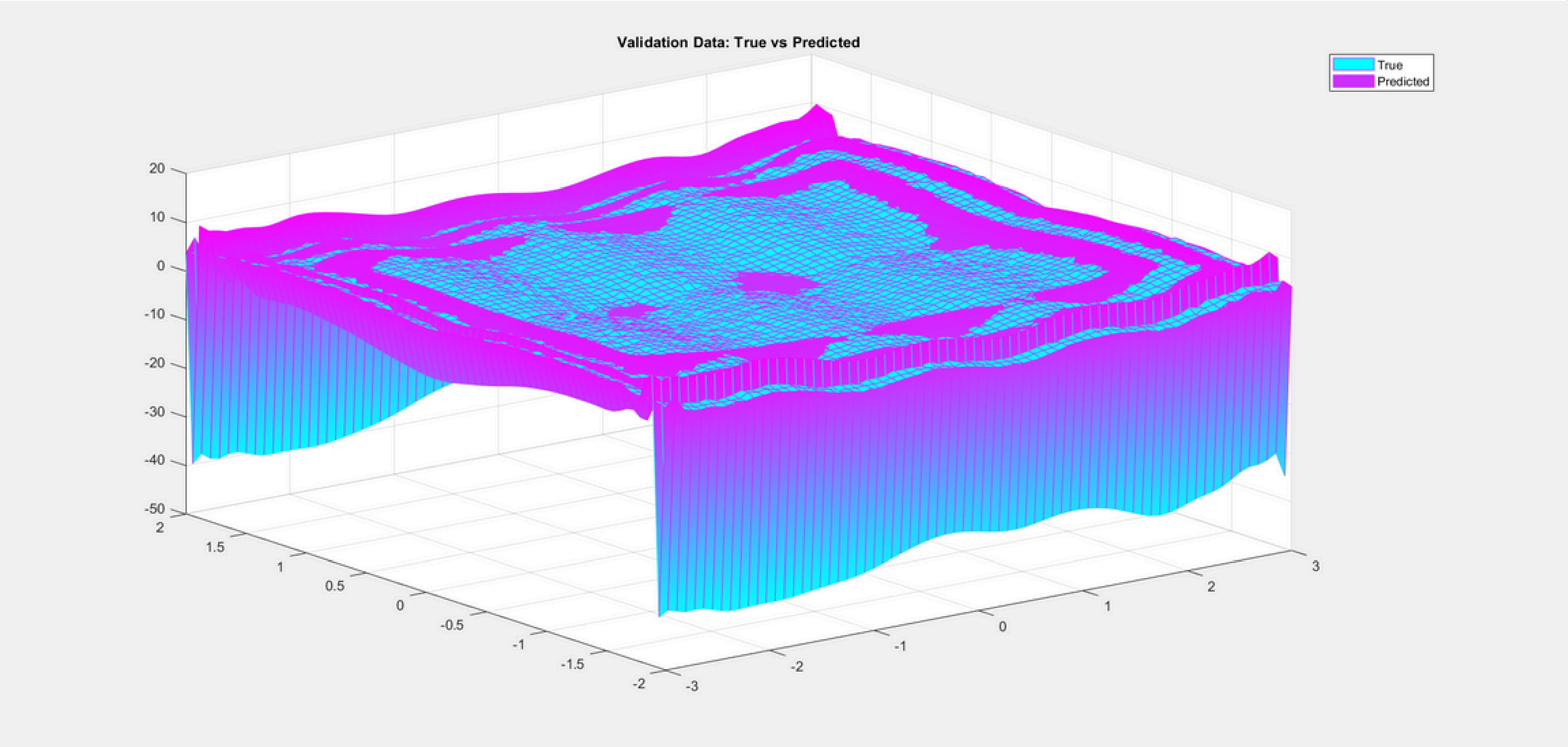

Validation error as a function of m
Mean Squared Error
Polynomial Degree m
0
1
2
3
4
5
6
7
0
5
10
15
20
25
30


Validation Data: True vs Predicted
True
Predicted
8
6
4
2
0
-2
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2
-1
0
1
2
3

CONCLUSIONS

The optimal degree for the approximator was determined by carefully monitoring the validation error, ensuring that the model was neither too simple (underfitting) nor too complex (overfitting).





Thank you

Code

```
function matrix = Polynomial(x1, x2, m, n)
 matrix = [];
 for i = 1:n
   for j = 1:n
     polynomial_terms = 1;
     for degree = 1:m
       for k = 0:degree
          l = degree - k;
          monomial = x2(i).^k * x1(j).^l;
          polynomial_terms = [polynomial_terms, monomial];
       end
    end
    matrix = [matrix; polynomial_terms];
    end
  end
end
```

```
for degree = 1:m
   R_degree = R(:, 1:(degree+1)*(degree+2)/2);
   Rval_degree = R_val(:, 1:(degree+1)*(degree+2)/2);

   theta = R_degree\Y_flat;
   Y_hat = R_degree*theta;
   Y_hat_val = Rval_degree*theta;

   errors(degree) = mean((Y_flat - Y_hat).^2);
   errors_val(degree) = mean((Y_val_flat - Y_hat_val).^2);
 end
```

```
 figure
 mesh(X_val{1}, X_val{2}, Y_val, 'FaceColor', 'c'); hold on;
 colormap(cool)
 mesh(X_val{1}, X_val{2},
Y_hat_val_optimal_reshaped,'FaceColor', 'interp');
 colormap(cool)
 title('Validation Data: True vs Predicted');
 legend('True', 'Predicted');
```