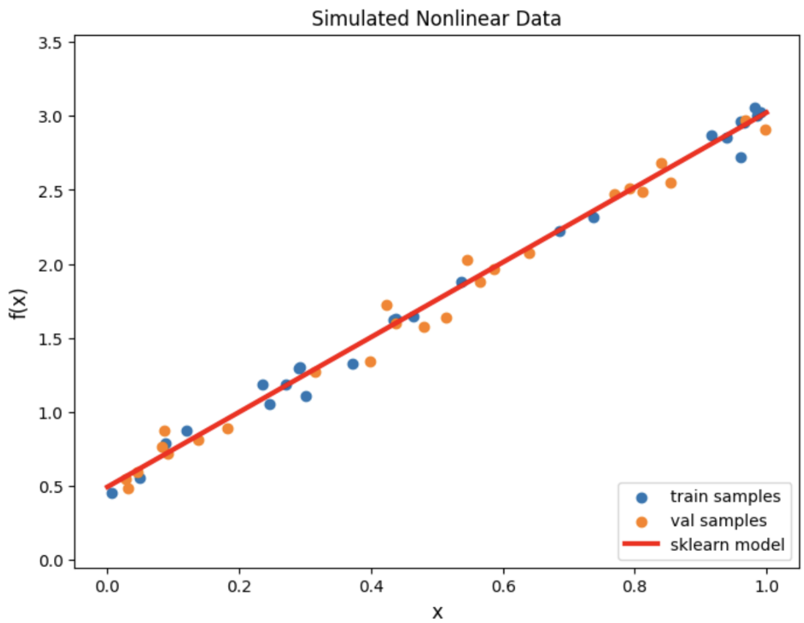
https://colab.research.google.com/drive/1Da5gNfUWho-v5dG7XQnyIXhtqg3FIEJe?usp=sharing

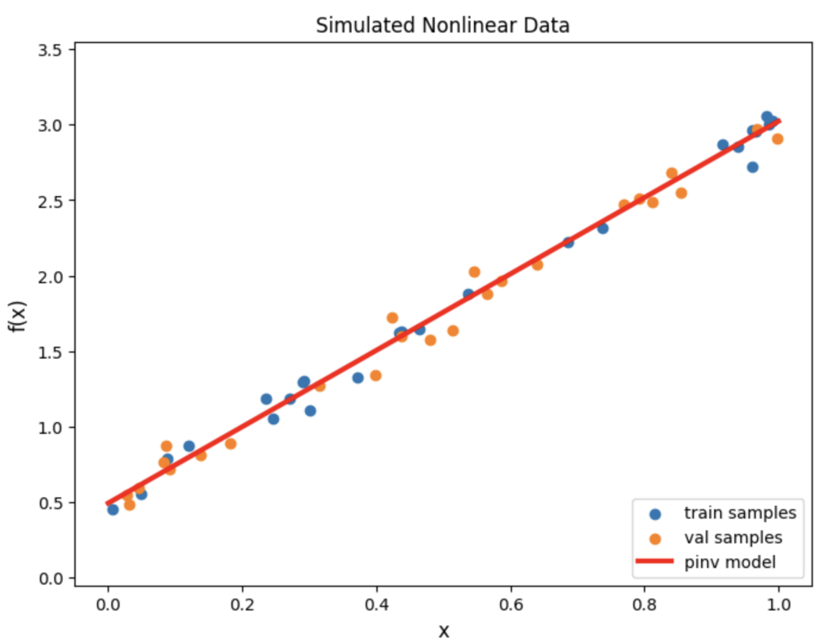
	Part 1.a	Part 1.b	Part 1.c
MSE	0.00795462682779033	0.007954626827790339	0.0065
Coefficients	[2.52838262] [0.49447669]	[[2.52838262] [0.49447669]]	[2.51545484 0.49356359]

	Part 2.a Order = 1	Part 2.a Order = 3	Part 2.a Order = 5	Part 2.a Order = 7	Part 2.b
MSE	0.06362852709262727	0.012059223742868292	0.007475463079281102	0.011549227838827407	0.01205922374286855
Coefficients	[[1.04997846 1.4448342]]	[[0.39883796 8.68921502 -18.07027007 12.18401084]]	[[0.63284179 2.30261338 25.94068538 -105.99696315 135.01003494 -55.09169475]]	[[0.87278919 -6.50389643 110.95383686 -452.95566198 836.50779907 -778.76493098 350.74072313 -57.70415648]]	[[0.39883796] [8.68921502] [-18.07027007] [12.18401084]]

Part 1.a Visualization:

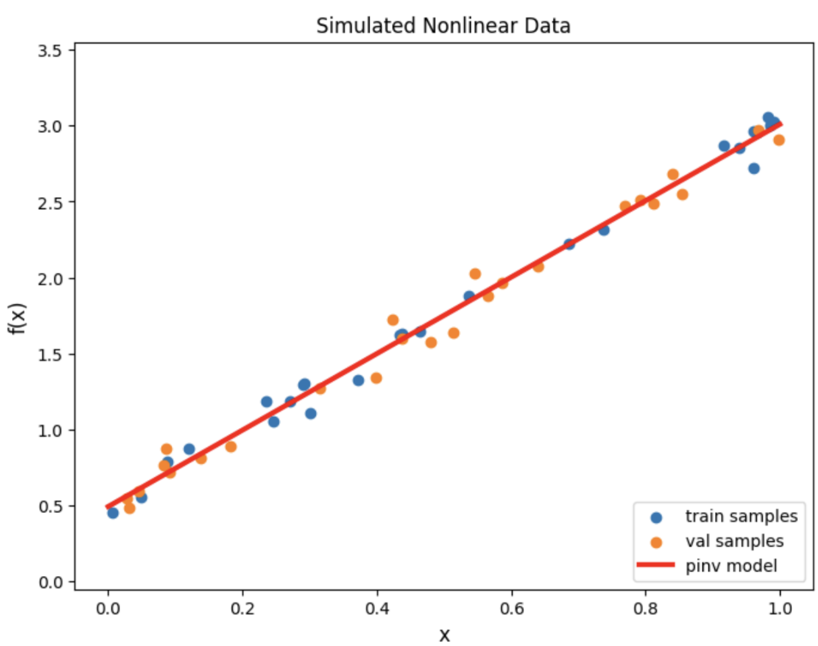


Part 1.b Visualization:

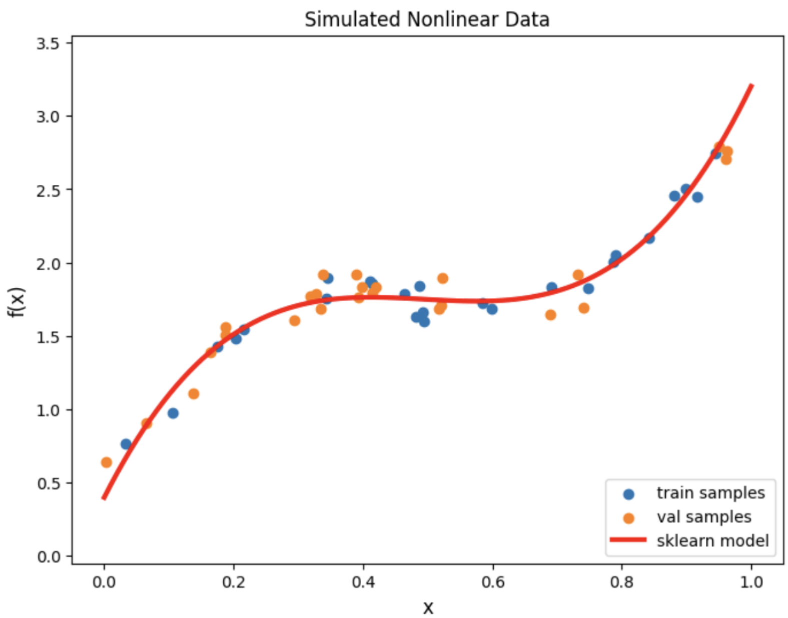


Part 1.c Results and Visualization:

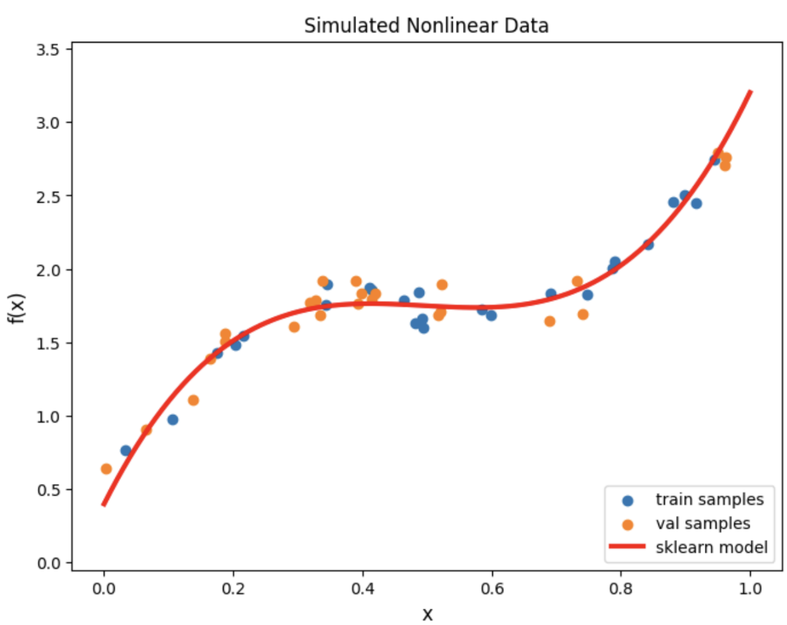
MSE error at step 1: 1.7236
Coefficients: [-1.16148226 1.8064415]
MSE error at step 100: 0.2412
Coefficients: [1.05069603 1.28977019]
MSE error at step 200: 0.0497
Coefficients: [1.88797896 0.83464435]
MSE error at step 300: 0.0145
Coefficients: [2.24688946 0.63954925]
MSE error at step 400: 0.0080
Coefficients: [2.40074046 0.55591956]
MSE error at step 500: 0.0068
Coefficients: [2.4666904 0.52007076]
MSE error at step 600: 0.0066
Coefficients: [2.49496058 0.50470377]
MSE error at step 700: 0.0066
Coefficients: [2.50707891 0.49811654]
MSE error at step 800: 0.0065
Coefficients: [2.51227357 0.49529285]
MSE error at step 900: 0.0065
Coefficients: [2.51450032 0.49408244]
MSE error at step 1000: 0.0065
Coefficients: [2.51545484 0.49356359]



Part 2.a Visualization:



Part 2.b Results:



About Part 1:

Yes, the gradient descent solution is very close to the solutions obtained for Part 1a and b. The reason for this is that all three methods are attempting to find the coefficients that minimize the mean squared error between the predicted values and the actual values of the target variable. In detail, the closed-form solution finds the minimum mean squared error, while the gradient descent method approaches that solution iteratively. Although the methods differ in their approach, the solutions obtained are very similar since the learning rate is correctly chosen.

About Part 2:

I think the optimal degree is 3 since the data visualization seems like polynomial order 3. Therefore, it should generalize the data more correctly. However, it appears that the model with order = 5 has the lowest MSE, indicating the best performance on the validation set. When the degree is too small, as in order = 1, the model may underfit the data, resulting in high bias and poor performance on both the training and validation sets. On the other hand, when the degree is too high, as in order = 7, the model may overfit the data, resulting in low bias but high variance and poor generalization to new data. In general, the optimal degree value will depend on the complexity of the underlying relationship between the input and output variables, as well as the size and noise level of the data. One approach to finding the optimal degree is to use a validation set to evaluate the performance of models with different degrees and choose the degree that minimizes the validation error as we did for this problem.