Linear Least Squares Curve Fitting (Regression) Simulations

The Excel file **line_data** contains four sets of y measurements ($\boldsymbol{y_0}$, $\boldsymbol{y_1}$, $\boldsymbol{y_2}$, and $\boldsymbol{y_3}$) obtained in the range $-10 \leq x \leq +10$. $\boldsymbol{y_0}$ contains y measurements without any error. $\boldsymbol{y_1}$ contains a small amount of error, while $\boldsymbol{y_2}$ and $\boldsymbol{y_3}$ contain moderate amounts of error, about twice as much as in $\boldsymbol{y_1}$.

Use the linear least squares method to estimate the best fit line for each data set. You should have functions available to multiply, transpose matrices and vectors and to invert matrices (or Gaussian elimination to solve equations). Do not use any least squares function or curve fitting function that your programming environment may offer. Don't hard code input parameters – use variables and input/initialize them conveniently. Use double precision for computations. Indent your code, use comments.

1) Verify your program by fitting a line to $\boldsymbol{x}$ vs $\boldsymbol{y_0}$. This should be a perfect fit, and $\widehat{\boldsymbol{x}}$ should have the true parameter values of the line. Plot the data and fitted line $\widehat{\boldsymbol{y}}_{\boldsymbol{0}}$. Is $\boldsymbol{b} = \boldsymbol{y_0}$ in $C(\boldsymbol{A})$? How can your verify it?

2) Fit lines to $\boldsymbol{y_1}, \boldsymbol{y_2}$, and $\boldsymbol{y_3}$ and plot them. Suggest making three plots, each containing $\boldsymbol{y_0}, \boldsymbol{y_i}$, and $\widehat{\boldsymbol{y}}_{\boldsymbol{i}}$ for comparison. In each case,
   a) calculate the TLSE (***E***),
   b) calculate the projection matrices $\boldsymbol{P}$ and $\boldsymbol{I} - \boldsymbol{P}$ and verify that they satisfy $\boldsymbol{P} = \boldsymbol{P}^T = \boldsymbol{P}^2$ and $(\boldsymbol{I} - \boldsymbol{P}) = (\boldsymbol{I} - \boldsymbol{P})^T = (\boldsymbol{I} - \boldsymbol{P})^2$; compute $\boldsymbol{e} = (\boldsymbol{I} - \boldsymbol{P}\ )\boldsymbol{b}$. Does this $\boldsymbol{e}$ computed using the projection matrix match with $\boldsymbol{e}$ computed as $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{A}\widehat{\boldsymbol{x}}$?
   c) Verify that ***e*** has an average value of 0 and lies in the left null space of ***A.***
   d) How do the estimated line parameters for each case compare with the true values?

3) Check how well the model trained on the $\boldsymbol{y_2}$ data set fits the $\boldsymbol{y_3}$ data set (unseen/test data) and vice-versa. Calculate ***E*** in each case (cross-TLSE), and compare with the ***E*** values in Part 2). Are they very different? Would you say that the models do about as well on the test data as on the training data?

4) Fit higher order polynomials, *n* = 2, 3, 4, 8, 12, 16 to $\boldsymbol{y_0}$, $\boldsymbol{y_1}$, $\boldsymbol{y_2}$, and $\boldsymbol{y_3}$ and observe how ***E*** changes as the polynomial order increases. In the case of $\boldsymbol{y_0}$, what are the coefficients of the polynomial for each *n*? Why? Make a table for ***E*** with rows corresponding to the polynomial model orders and columns to $\boldsymbol{y_1}, \boldsymbol{y_2}$, and $\boldsymbol{y_3}$.