Here we generate some artificial "data". At four data points \( x=1,2,3,4 \) our \( y \) values are given by the function \( f(x) \) above with some "noise" added to them. We then seek parameters \( c[1], c[2], c[3] \) that give a least squares fit to the model

\[
\]

This yields a matrix problem with 4 equations for 3 unknowns. The rows of \( A \) are \([ 1 \ x \_i \ \log(x\_i) ]\). Then the least squares solution \( c \) solves the normal equations

\[
A^T A \ c = A^T \ y
\]

Below is some code to solve the normal equations. It is "clunky". For large amounts of data, other solution methods are used.

\[
c := \text{inverse} (\text{transpose}(A) \ &* A) \ &* (\text{transpose}(A) \ &* y): \\
cc := \text{evalf} (\text{evalm}(c));
\]

Compare these to the "exact" values \( cc=(2,3,1) \). Below is a plot of \( f(x) \) versus the Least Squares fit of the noisy data: