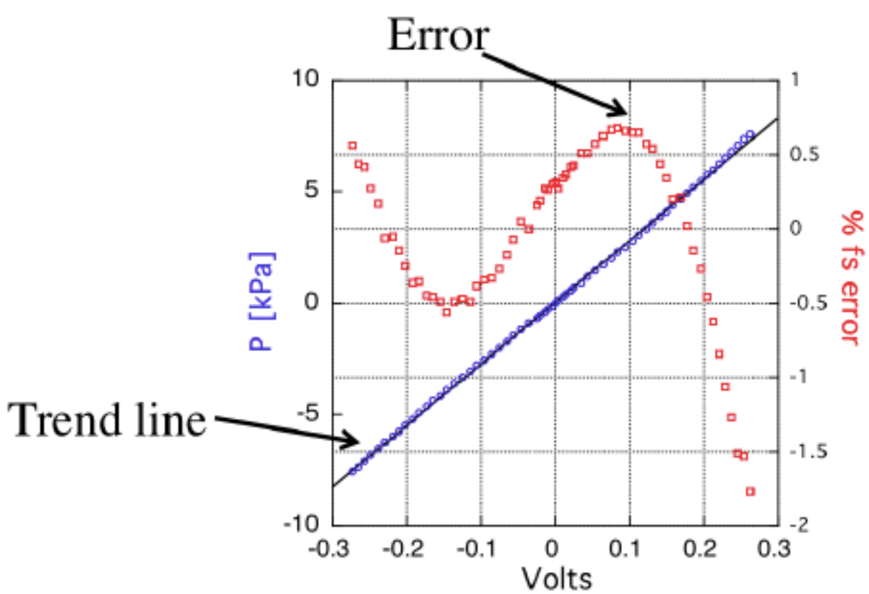
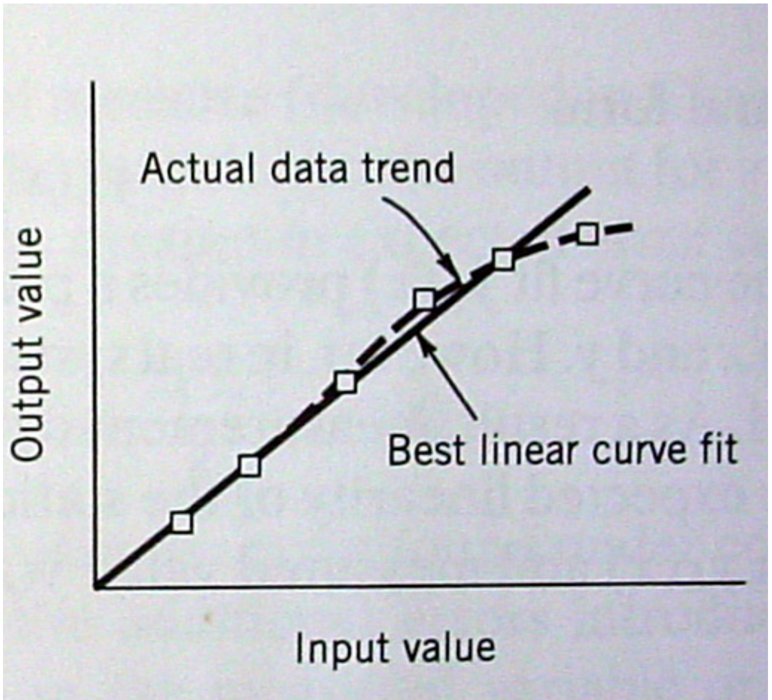
Section 3.2: Introduction Method of Least Squares (B.M.L. Chapter 3, pp. 34-43, 73-97)

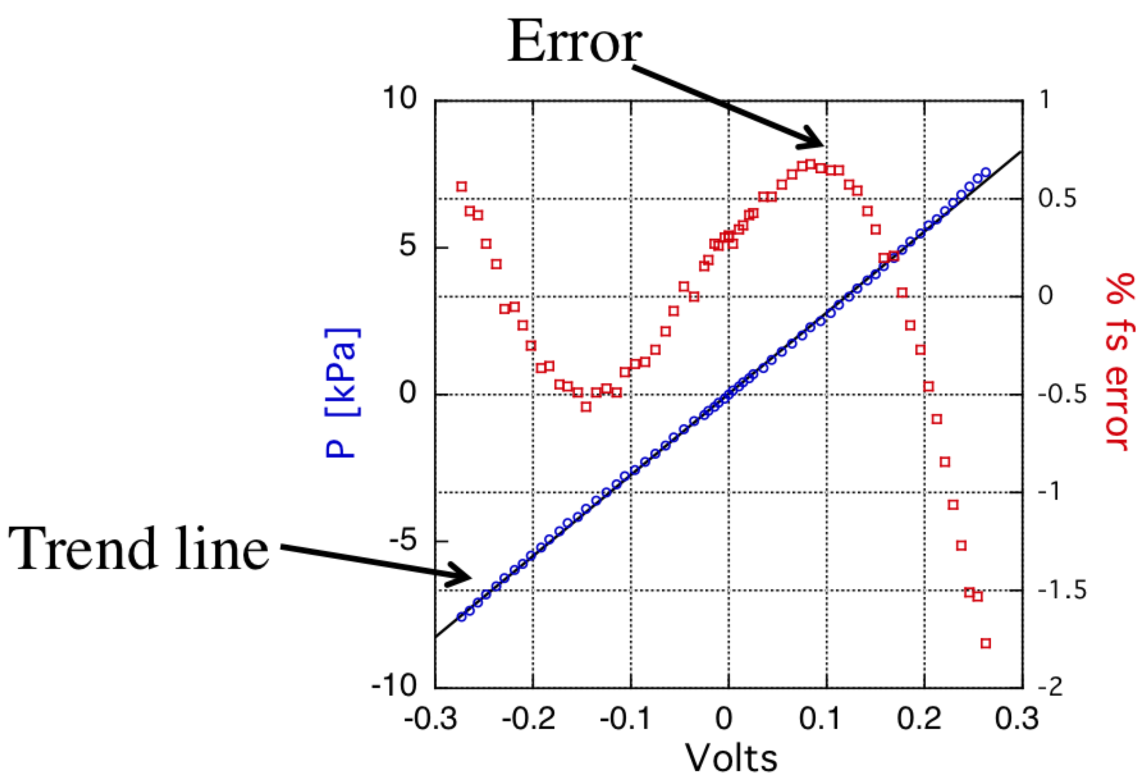




(b) Linearity error

Section 3.2 : "Curve Fitting" Trend Lines in Data

• Typically a measurement will have a mean "trend line"
With a variability about that trend line

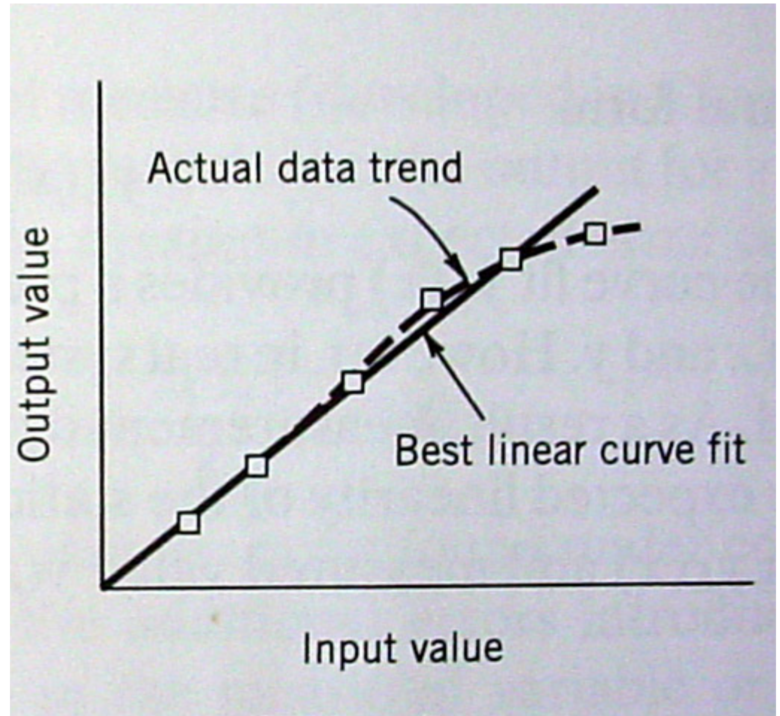


Linearity

Many types of sensors have linear input/output behavior, along a defined range of inputs. The sensor thus follows an input/output relation like

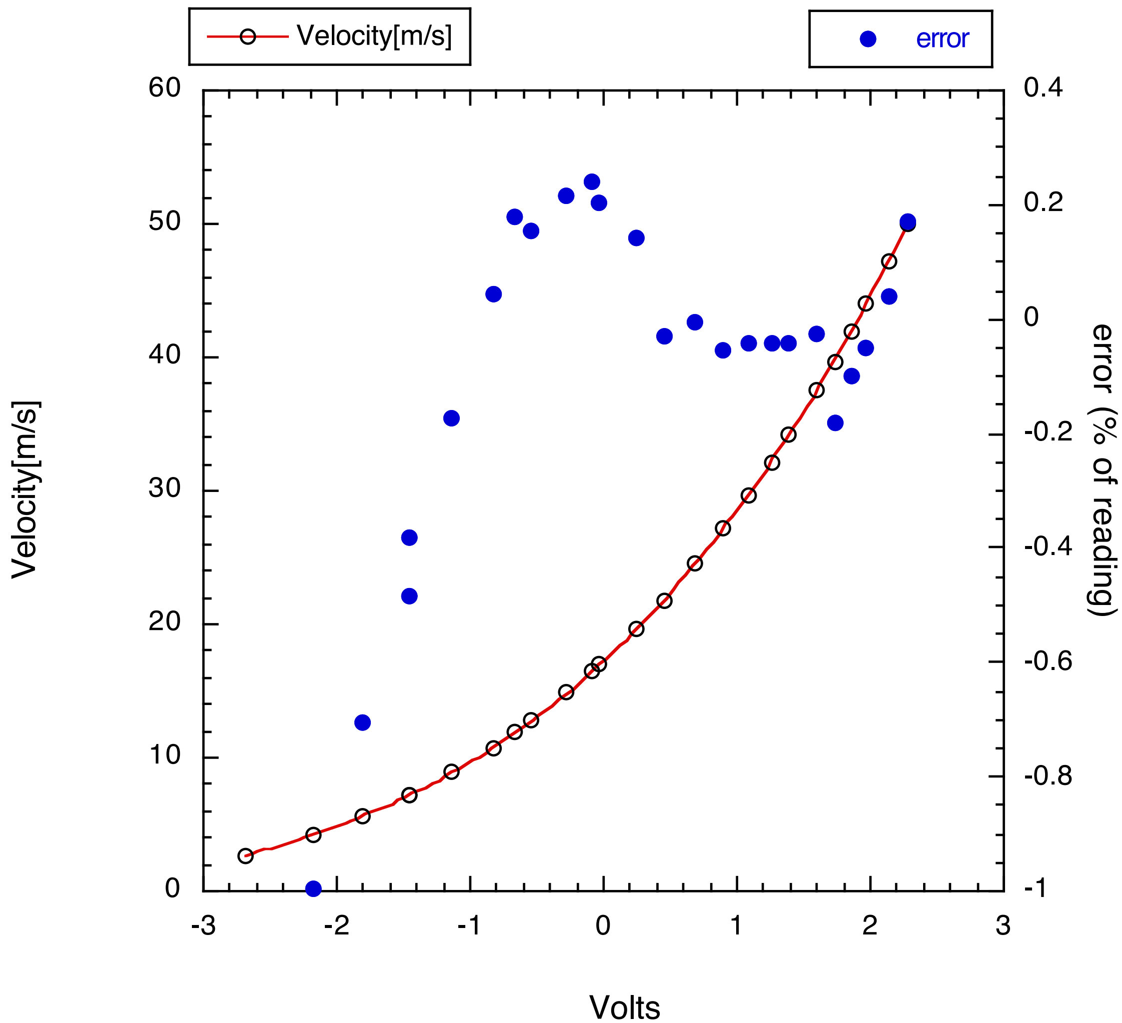
$$y_L(x) = a_0 + a_1 x.$$

These will often be marketed as linear, and the only calibration data you get is the slope of the input/output relation *(a_1)* and the zero input value *(a_0)*. For these types of sensors, the deviation from linear behavior is reported in the specifications. This deviation can be calculated: $e_L(x) = y(x) - y_L(x)$. The spec is usually the percentage error relative to full scale, or



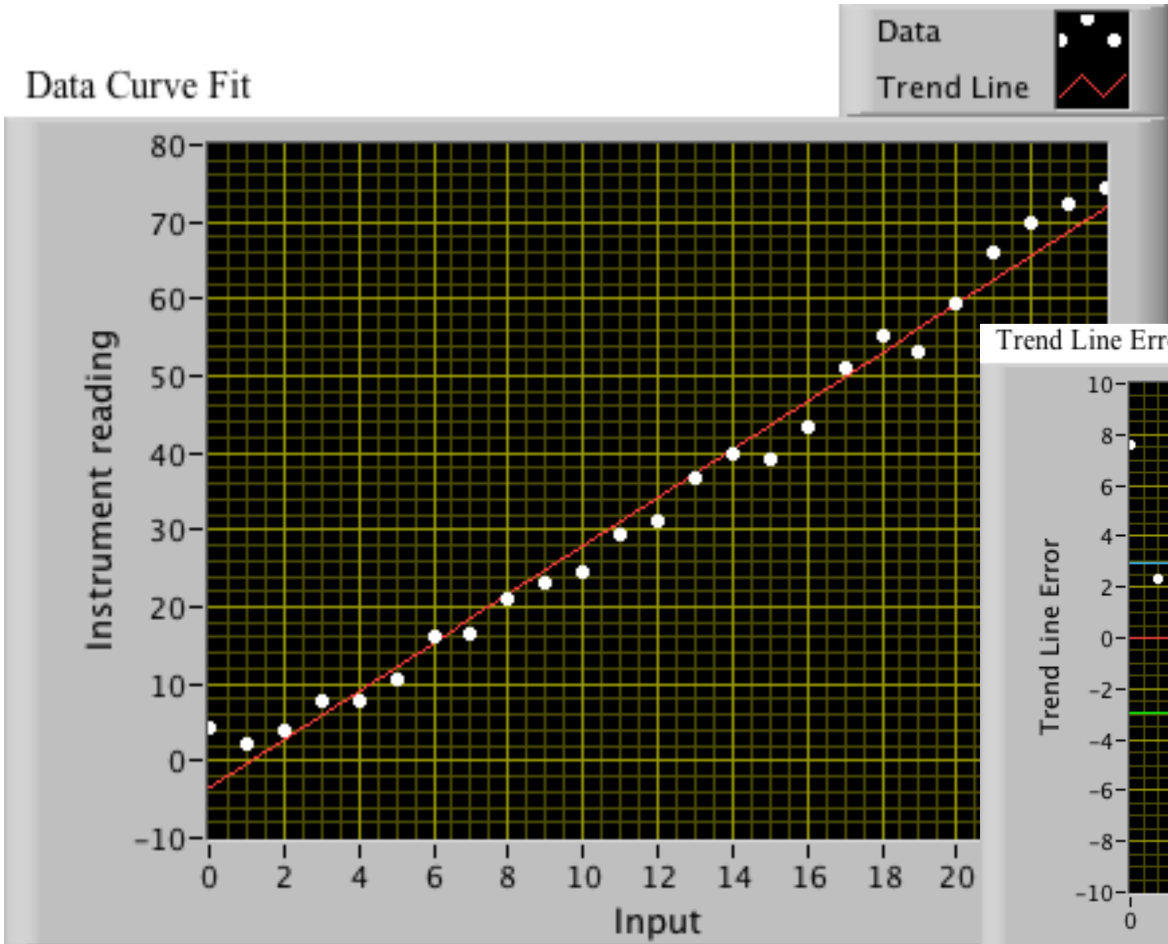
(b) Linearity error

Non-linear Calibration Example



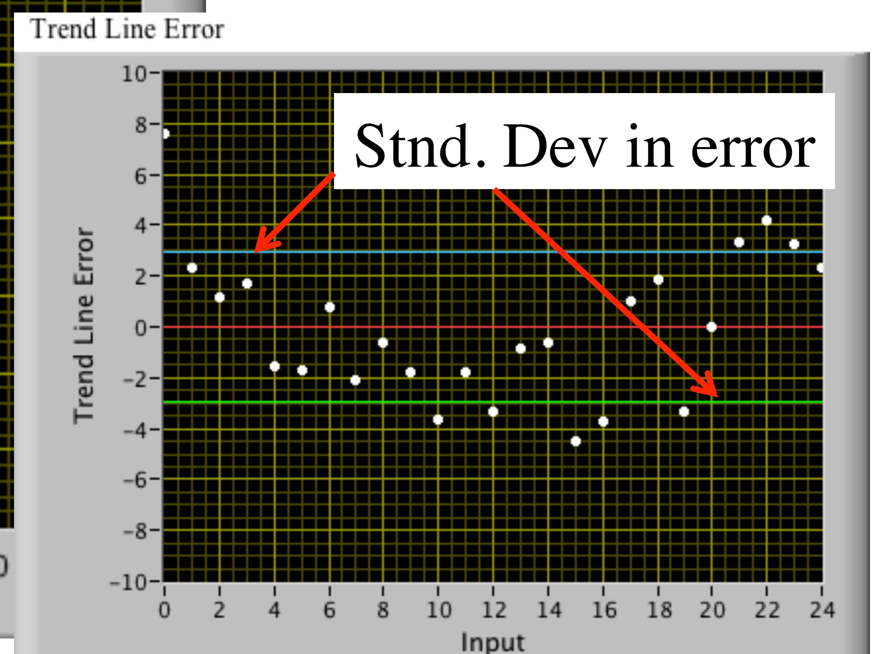
- Many times
The trend is
Not a line but a
"curve" .. And
We describe the
Trend as a
"calibration" curve

The Trend Line: Linear least Squares (1)



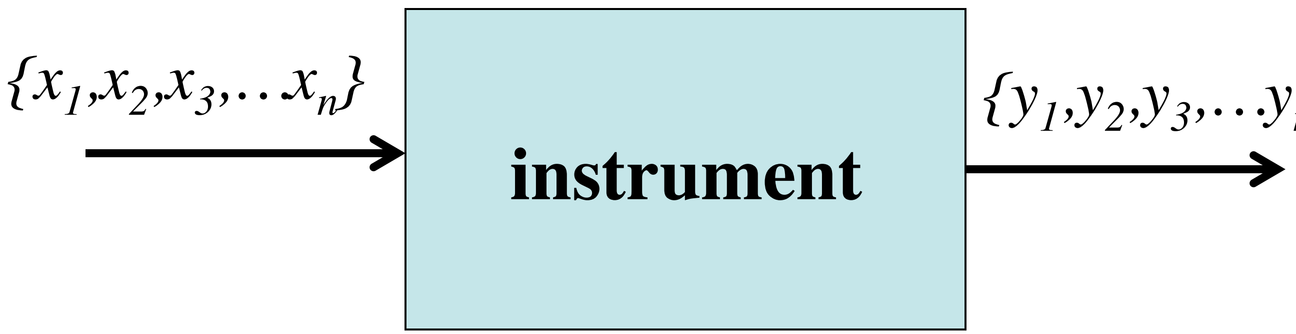
- Linear trend line
  $y = a_1 x + a_0$

- ***How de we Calculate this line?***



The Trend Line: Linear least Squares (2)

- Consider a set of calibration data for an instrument



We want to model the input/output relationship
By a straight line of the form

$$y(x) = a_1 x + a_0$$

- Given the calibration Data set $\{x_i, y_i\}$ we want To compute a_0, a_1 so that we Get the best overall ***"fit"*** to data

UtahState UNIVERSITY — Mechanical & Aerospace Engineering

The Trend Line: Linear least Squares (3)

- We want to minimize the *fit variance* …. The “squared error” or … ***“Least squares”*** of the Collected data set

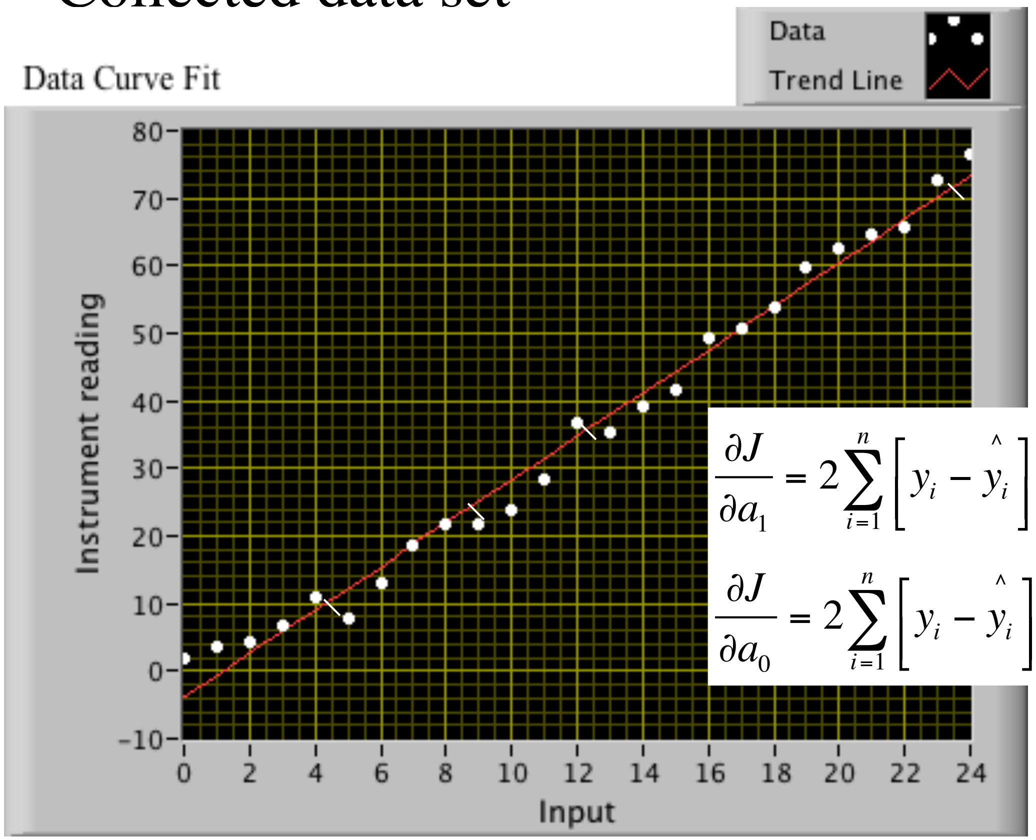
Minimize sum of Squares of the fit error

$$J = \sum_{i=1}^{n}\left[y_i - \hat{y}_i\right]^2 \rightarrow \hat{y}_i = a_1 x_i + a_0$$

from calculus

$$J_{\min} \rightarrow \left[\frac{\partial J}{\partial a_1} = 0, \frac{\partial J}{\partial a_0} = 0\right]$$

We can solve as



$$\frac{\partial J}{\partial a_1} = 2\sum_{i=1}^{n}\left[y_i - \hat{y}_i\right]\left[\frac{\partial\left(a_1 x_i + a_0\right)}{\partial a_1}\right] = 2\sum_{i=1}^{n}\left[y_i - \left(a_1 x_i + a_0\right)\right]\left[x_i\right] = 0$$

$$\frac{\partial J}{\partial a_0} = 2\sum_{i=1}^{n}\left[y_i - \hat{y}_i\right]\left[\frac{\partial\left(a_1 x_i + a_0\right)}{\partial a_0}\right] = 2\sum_{i=1}^{n}\left[y_i - \left(a_1 x_i + a_0\right)\right]\left[1\right] = 0$$

2 equations in two unknowns (a_1, a_0)

“not very convenient”

The Trend Line: Linear least Squares (4)

- An easier and more general way is to write system in Matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} - \begin{bmatrix} (\hat{y}_2) \\ (\hat{y}_2) \\ \cdots \\ (\hat{y}_n) \end{bmatrix} = \begin{bmatrix} y_1 - (a_1 x_1 + a0) \\ y_2 - (a_1 x_2 + a0) \\ \cdots \\ y_n - (a_1 x_n + a0) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & 1 \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} \rightarrow \boxed{Y - \hat{Y} = Y - XA}$$

$$Y \equiv \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} \rightarrow \hat{Y} = \begin{bmatrix} (\hat{y}_2) \\ (\hat{y}_2) \\ \cdots \\ (\hat{y}_n) \end{bmatrix} \rightarrow X \equiv \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & 1 \\ x_n & 1 \end{bmatrix} \rightarrow A = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}$$

The Trend Line: Linear least Squares (5)

- Write the sum of the squared error in matrix form

$$J = \sum_{i=1}^{n}\left[y_i - \hat{y}_i\right]^2 = \left[y_1 - \hat{y}_1 \quad y_2 - \hat{y}_2 \quad \dots \quad y_n - \hat{y}_n\right]\begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \dots \\ y_n - \hat{y}_n \end{bmatrix} = \left[Y - \hat{Y}\right]^T\left[Y - \hat{Y}\right]$$

… and …

$$J = \sum_{i=1}^{n}\left[y_i - \hat{y}_i\right]^2 = \left[Y - \hat{Y}\right]^T\left[Y - \hat{Y}\right] = \left[Y - XA\right]^T\left[Y - XA\right]$$

The Trend Line: Linear least Squares (6)

- Now perform minimization … using the matrix operation rules

$$i)\left[XY\right]^T = Y^T X^T$$

$$ii)\nabla_X\left(YZ\right) = \nabla_X\left(Y\right)Z + Y\nabla_X\left(Z\right)$$

$$iii)\nabla_A\left[A^T X^T XA\right] = \left(\nabla_A A^T\right)\left(X^T XA\right) + A^T\nabla_A\left(X^T XA\right) =$$
$$\left(\nabla_A A^T\right)\left(X^T XA\right) + \left(\nabla_A A^T\right)\left(X^T X\right)A = 2X^T XA$$

$$\nabla_A J = \nabla_A\left(\left[Y - XA\right]^T\left[Y - XA\right]\right) = \nabla_A\left(Y^T Y - 2A^T X^T Y + A^T X^T XA\right) = -2X^T Y + 2X^T XA$$

$$\nabla_A J = 0 \rightarrow \boxed{X^T Y = X^T XA} \rightarrow \boxed{A = \left[X^T X\right]^{-1} X^T Y}$$

The Trend Line: Linear least Squares (7)

- Look at Matrix Solution

$$A = \left[X^T X\right]^{-1} X^T Y \rightarrow \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \left(\begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{pmatrix} \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & 1 \\ x_n & 1 \end{bmatrix} \right)^{-1} \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{pmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{pmatrix} \sum_{i=1}^{n} (x_i)^2 & \sum_{i=1}^{n} (x_i) \\ \sum_{i=1}^{n} (x_i) & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} (x_i y_i) \\ \sum_{i=1}^{n} (y_i) \end{pmatrix}$$

2 x 1 ***2 x 2*** ***2 x 1***

Reduces to a 2-by element system

The Trend Line: Linear least Squares (8)

$$\begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{pmatrix} \sum_{i=1}^{n}(x_i)^2 & \sum_{i=1}^{n}(x_i) \\ \sum_{i=1}^{n}(x_i) & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n}(x_i y_i) \\ \sum_{i=1}^{n}(y_i) \end{pmatrix} \rightarrow \textit{use Cramer's rule for inverse}$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{-1} = \frac{\begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}}{a_{11}a_{22} - a_{12}a_{21}} \rightarrow \begin{pmatrix} \sum_{i=1}^{n}(x_i)^2 & \sum_{i=1}^{n}(x_i) \\ \sum_{i=1}^{n}(x_i) & n \end{pmatrix}^{-1} = \frac{\begin{pmatrix} n & -\sum_{i=1}^{n}(x_i) \\ -\sum_{i=1}^{n}(x_i) & \sum_{i=1}^{n}(x_i)^2 \end{pmatrix}}{\left(n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2 \right)}$$

The Trend Line: Linear least Squares (9)

- Solve for slope and intercept

$$\begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \frac{\begin{pmatrix} n & -\sum_{i-1}^{n}(x_i) \\ -\sum_{i=1}^{n}(x_i) & \sum_{i=1}^{n}(x_i)^2 \end{pmatrix}}{\left(n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2 \right)} \begin{pmatrix} \sum_{i=1}^{n}(x_i y_i) \\ \sum_{i=1}^{n}(y_i) \end{pmatrix}$$

$$a_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2} \rightarrow \text{"}slope\text{"} \rightarrow$$

$$a_0 = \frac{\sum_{i=1}^{n}(x_i)^2\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2} \rightarrow \text{"}intercept\text{"}$$

The Trend Line: Linear least Squares (10)

- Long winded answer but a nice result

- Given the noisy data set …

$$\{input, output\} \rightarrow \left\{ \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \right\} \rightarrow best\ fit \rightarrow y_i = a_1 x_i + a_0$$

$$a_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2} \quad \dots \quad a_0 = \frac{\sum_{i=1}^{n}(x_i)^2\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2}$$

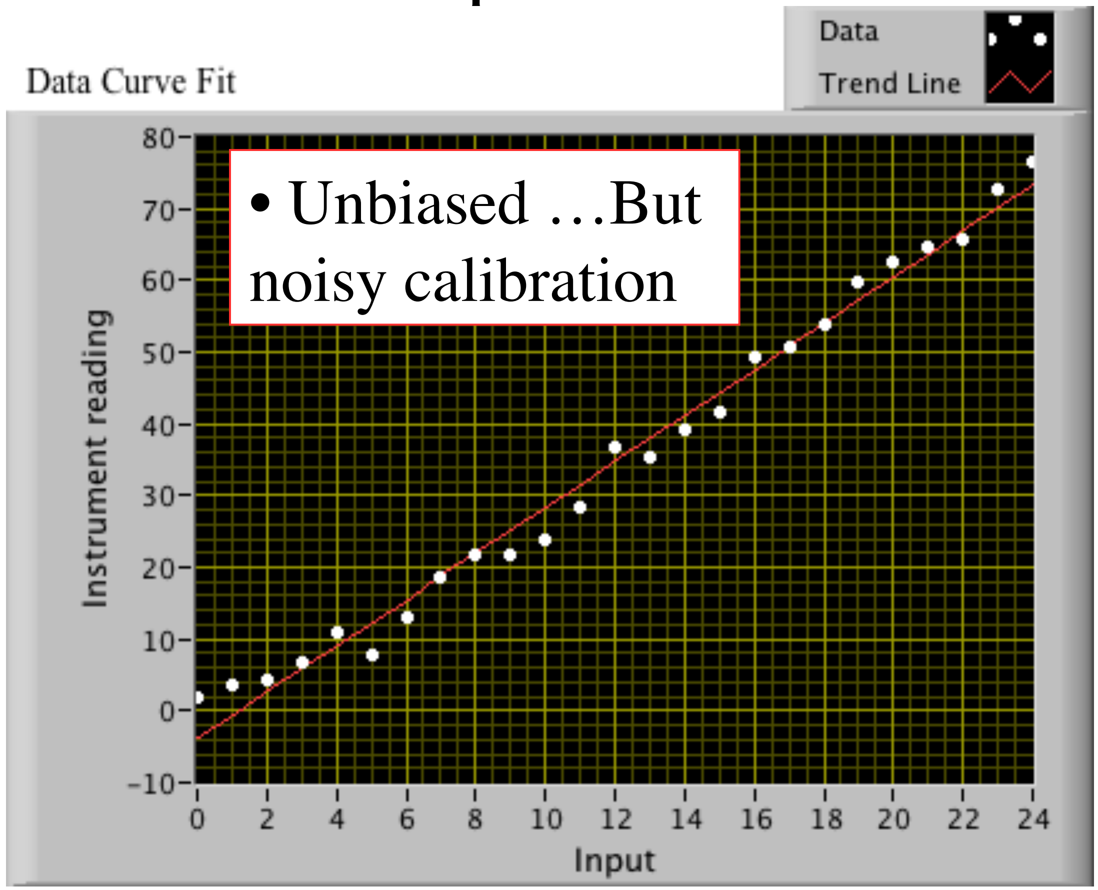
"careful with book keeping indices"

UtahState UNIVERSITY

Mechanical & Aerospace Engineering

Linear Fit Example

Y (inst. reading)	X (Calibration input)	Y (Best fit)
-2.20616	0	-3.87386
6.59695	1	-0.710946
8.59318	2	2.45197
3.52106	3	5.61488
7.81478	4	8.7778
9.67202	5	11.9407
17.9149	6	15.1036
20.3216	7	18.2665
18.3478	8	21.4295
20.5511	9	24.5924
24.8597	10	27.7553
26.8673	11	30.9182
29.1865	12	34.0811
35.5831	13	37.244
37.8743	14	40.4069
40.1035	15	43.5699
45.8147	16	46.7328
50.7996	17	49.8957
52.4604	18	53.0586
53.2805	19	56.2215
60.0723	20	59.3844
69.5095	21	62.5473
69.9549	22	65.7103
70.2957	23	68.8732
74.2384	24	72.0361
0	0	0



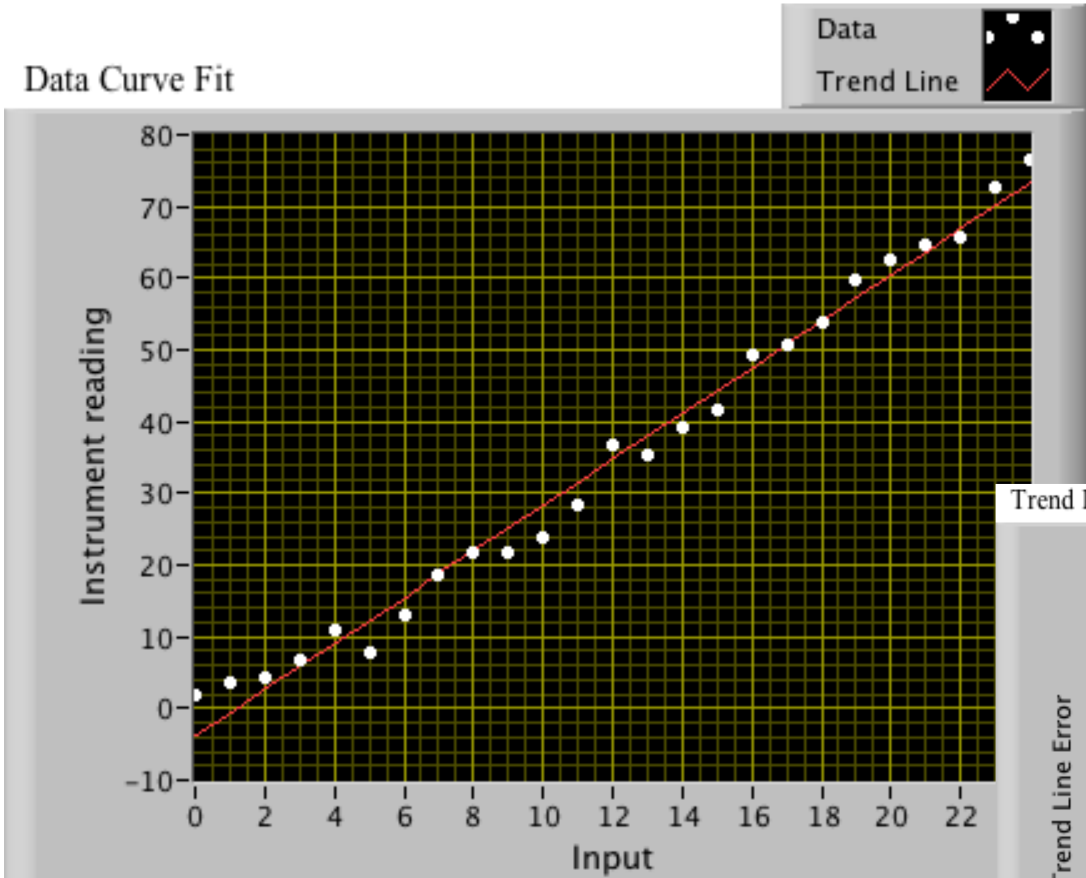
Fit Coefficients

-3.873	a_0
3.1629	a_1

Mean Fit Error (μ): 1.46372E-14 ⟶ bias

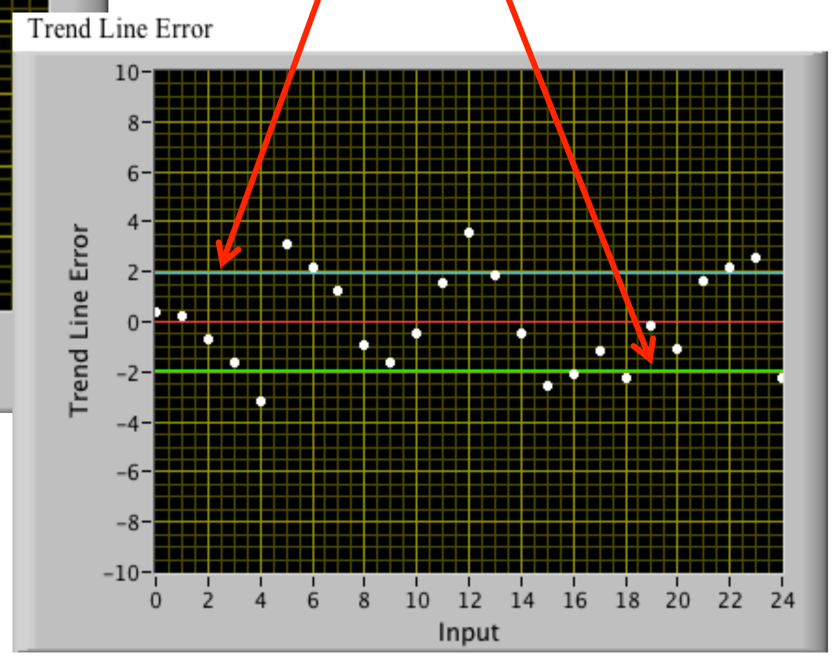
Standard Fit Error (σ): 3.67287 ⟶ RMSE (precision)

Linear Fit Example (2)



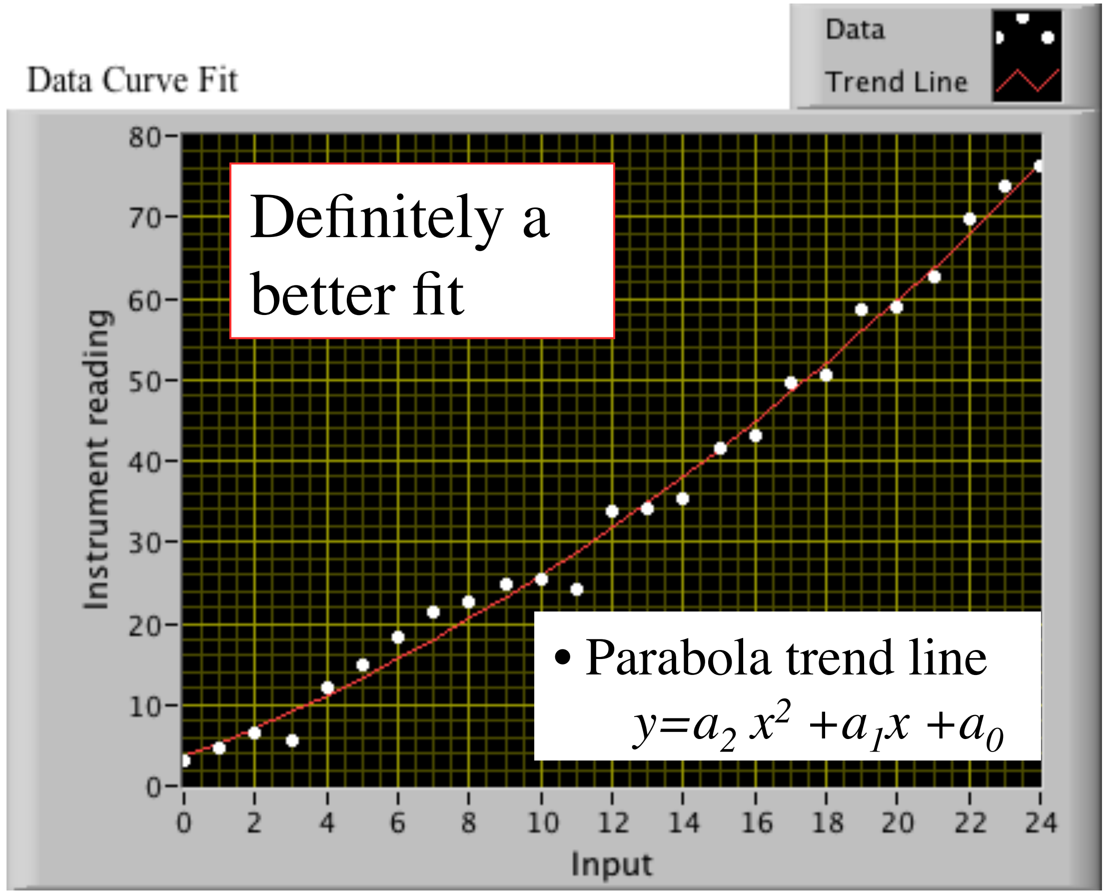
What appears to be Random error was Actually systematic

RMS error



Second Order Fit…

Y (inst. reading)	X (Calibration input)	Y (Best fit)
-2.20616	0	-3.87386
6.59695	1	-0.710946
8.59318	2	2.45197
3.52106	3	5.61488
7.81478	4	8.7778
9.67202	5	11.9407
17.9149	6	15.1036
20.3216	7	18.2665
18.3478	8	21.4295
20.5511	9	24.5924
24.8597	10	27.7553
26.8673	11	30.9182
29.1865	12	34.0811
35.5831	13	37.244
37.8743	14	40.4069
40.1035	15	43.5699
45.8147	16	46.7328
50.7996	17	49.8957
52.4604	18	53.0586
53.2805	19	56.2215
60.0723	20	59.3844
69.5095	21	62.5473
69.9549	22	65.7103
70.2957	23	68.8732
74.2384	24	72.0361
0	0	0



Definitely a better fit

- Parabola trend line $y = a_2 x^2 + a_1 x + a_0$

Fit Coefficients

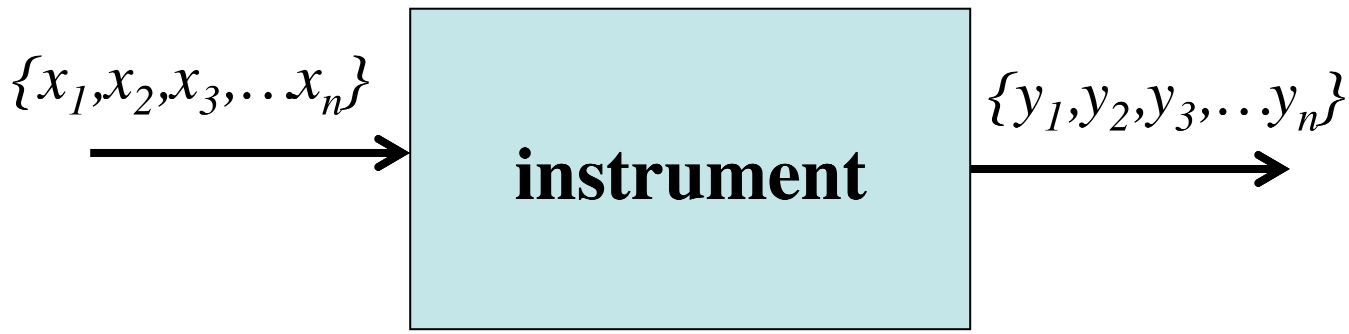
a_0	3.6341
a_1	1.6657
a_2	0.0573

Mean Fit Error (μ): 4.21352E-14 → bias

Standard Fit Error (σ): 1.99551 → RMSE (precision)

The Trend Curve: Second Order Least Squares (2)

- Use same approach as linear fit derivation



We want to model the input/output relationship
By a now use polynomial of the form

$$y(x) = a_2 x^2 + a_1 x + a_0$$

- Given the calibration
Data set $\{x_i, y_i\}$ we want
To compute a_0, a_1, a_2 so that we
Get the best overall ***"fit"*** to data

The Trend Curve: Second Order Least Squares (3)

$$J = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \begin{bmatrix} (y_1 - \hat{y}_1) & (y_2 - \hat{y}_2) & . & . & (y_n - \hat{y}_n) \end{bmatrix} \cdot \begin{bmatrix} (y_1 - \hat{y}_1) \\ (y_2 - \hat{y}_2) \\ . \\ . \\ (y_n - \hat{y}_n) \end{bmatrix} =$$

$$\left(Y - \hat{Y}\right)^T \left(Y - \hat{Y}\right) = (Y - X \cdot A)^T (Y - X \cdot A) \rightarrow Squared\ Error\ Index...$$

$$J = (Y - X \cdot A)^T (Y - X \cdot A) = Y^T \cdot Y - (X \cdot A)^T \cdot Y - Y^T (X \cdot A) + (X \cdot A)^T \cdot (X \cdot A) =$$

$$Y^T \cdot Y - A^T \cdot X^T \cdot Y - Y^T (X \cdot A) + A^T X^T \cdot X \cdot A$$

$$\boxed{J = Y^T \cdot Y - A^T \cdot X^T \cdot Y - Y^T (X \cdot A) + A^T X^T \cdot X \cdot A}$$

$$want....\nabla_A J = 0 \rightarrow \nabla_A \left\{ Y^T \cdot Y - A^T \cdot X^T \cdot Y - Y^T (X \cdot A) + A^T X^T \cdot X \cdot A \right\} =$$

$$\left\{ -\nabla_A A^T \cdot X^T \cdot Y - \nabla_A Y^T (X \cdot A) + \nabla_A A^T X^T \cdot X \cdot A \right\} = \left\{ -X^T \cdot Y - X^T Y^T + 2 \cdot X^T \cdot X \cdot A \right\} = 0$$

$$Solve \rightarrow X^T \cdot Y = X^T \cdot X \cdot A \rightarrow \boxed{\boxed{A = \left(X^T \cdot X\right)^{-1} X^T \cdot Y}}$$

The Trend Curve: Second Order Least Squares (3)

$$or..easy\ way \rightarrow Y = XA$$

$$\rightarrow solve\ for\ A \rightarrow X^T Y = X^T XA \rightarrow \boxed{A = \left(X^T X\right)^{-1} \cdot X^T Y}$$

$$"pseudo\ Inverse" \rightarrow \left(X^T X\right)^{-1} \cdot X^T$$

$$\boxed{A = \left[X^T X\right]^{-1} X^T Y}$$

Can be solved in Closed form but easier To use Numerical Methods

The Trend Curve: Polynomial Least Squares (1)

- In general for an "mth" order fit

$$Y \equiv \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix} \rightarrow \hat{Y} = \begin{bmatrix} (\hat{y}_2) \\ (\hat{y}_2) \\ \ldots \\ (\hat{y}_n) \end{bmatrix} \rightarrow X \equiv \begin{bmatrix} x_1{}^m & \ldots & x_1{}^2 & x_1 & 1 \\ x_2{}^m & \ldots & x_2{}^2 & x_2 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_n{}^m & .. & x_n{}^2 & x_n & 1 \end{bmatrix} \rightarrow A = \begin{bmatrix} a_m \\ \ldots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \rightarrow$$

$$A = \left[X^T X\right]^{-1} X^T Y$$

- Numerical Methods required for Solution to this system

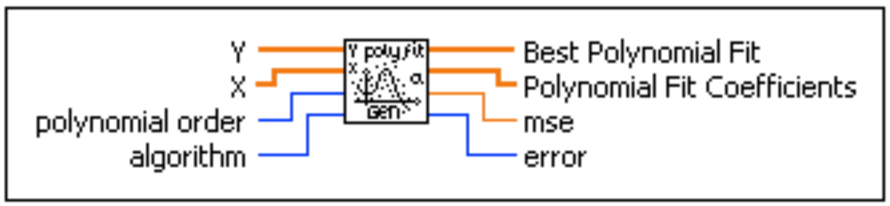
The Trend Curve: Polynomial Least Squares (2)

- In general for an “mth” order fit

$$A = \left[X^T X\right]^{-1} X^T Y$$

- Numerical Methods required for Solution to this system

- **Labview fit VI**



- Solution Algorithms

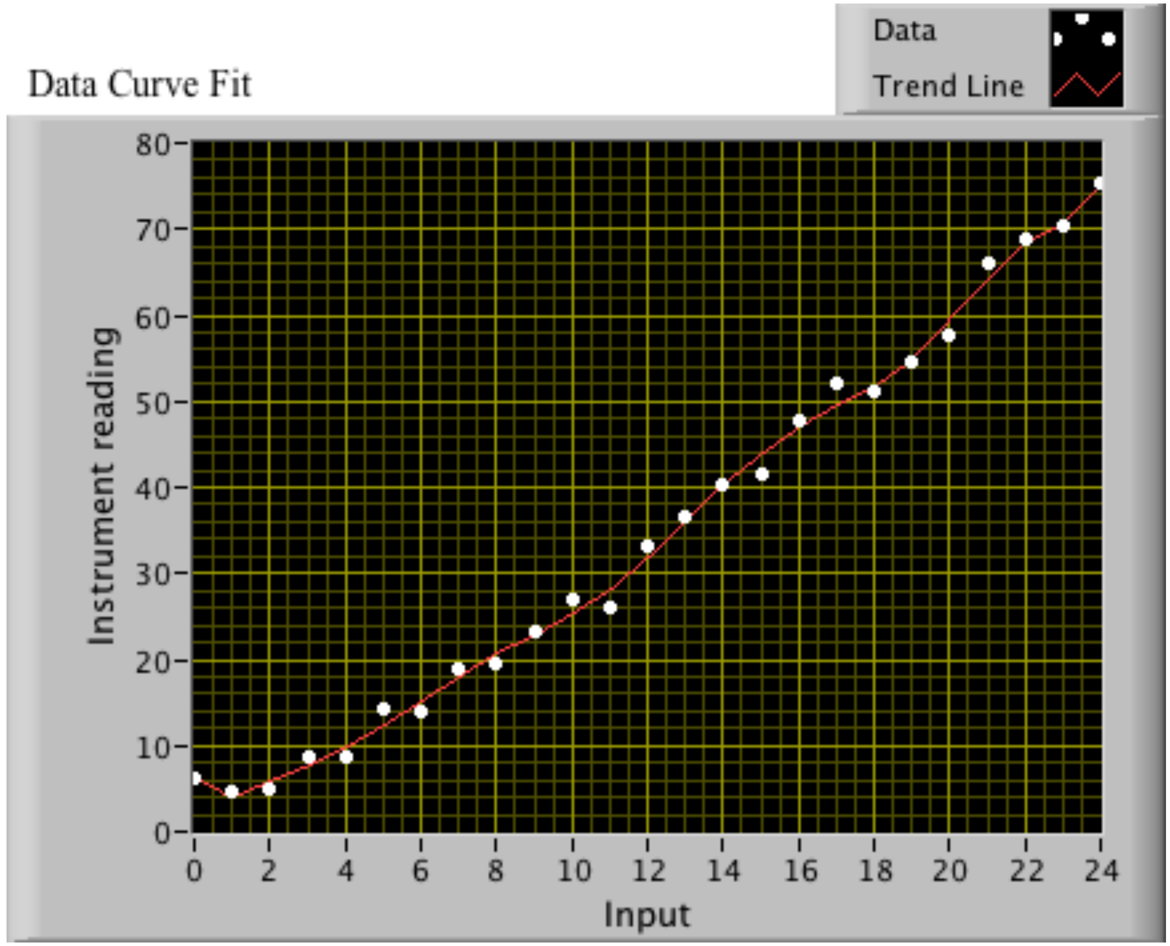
0	SVD (default)
1	Givens
2	Givens2
3	Householder
4	LU decomposition
5	Cholesky

The Trend Curve: Polynomial Least Squares (4)

- 10th order curve fit

- **Are we starting to over fit the data here?**

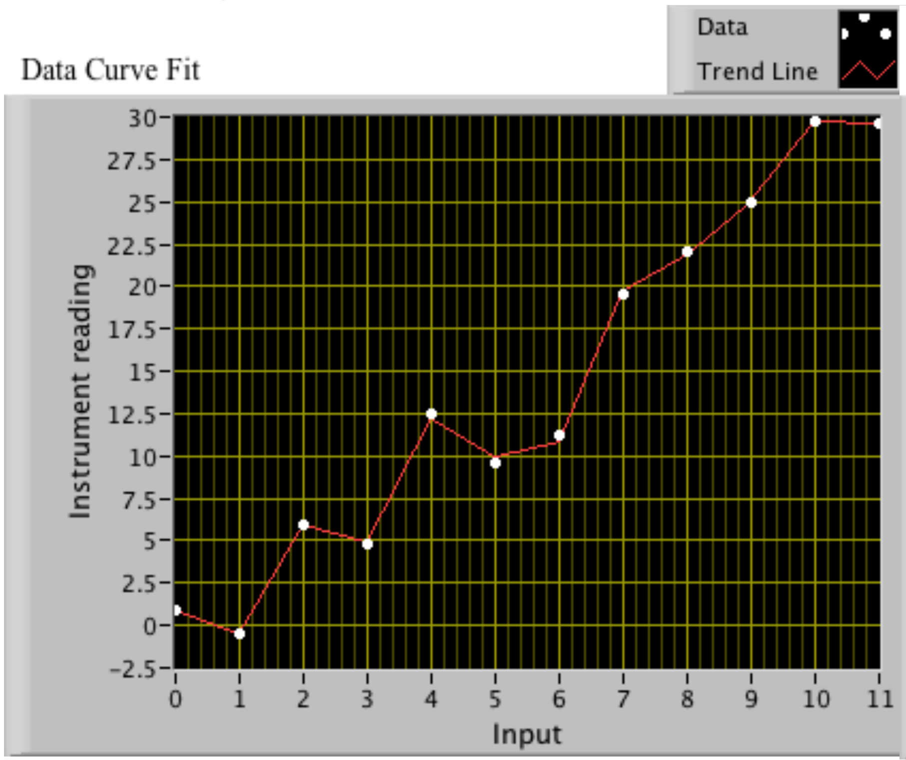
- **The curve inflections Are matching random Components in the data And not systematic trends**



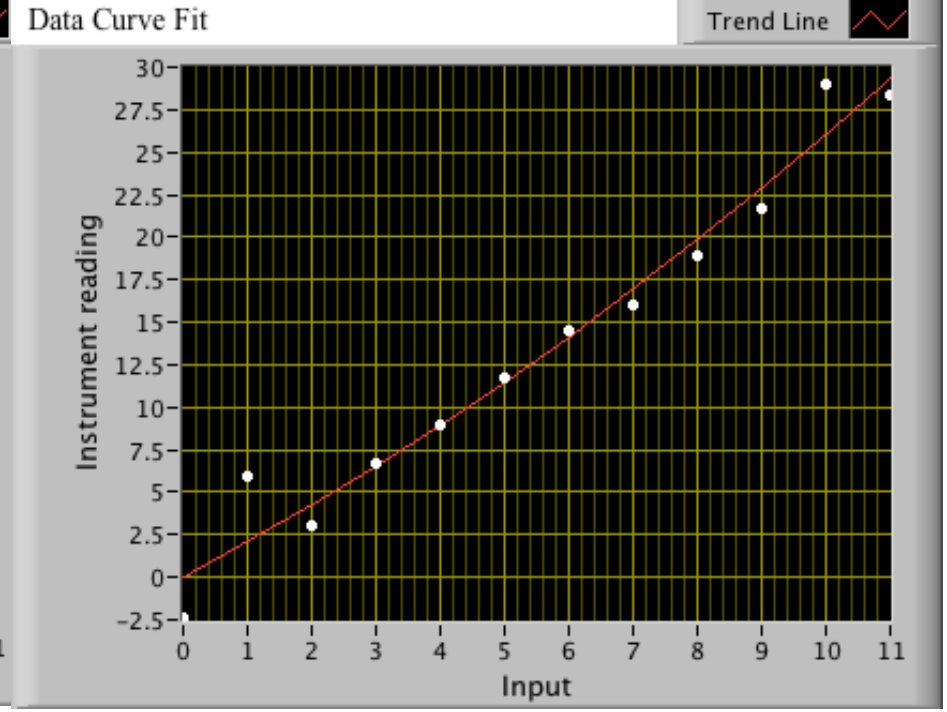
The Trend Curve: Polynomial Least Squares (5)

Calibration data set

- 10th order curve fit



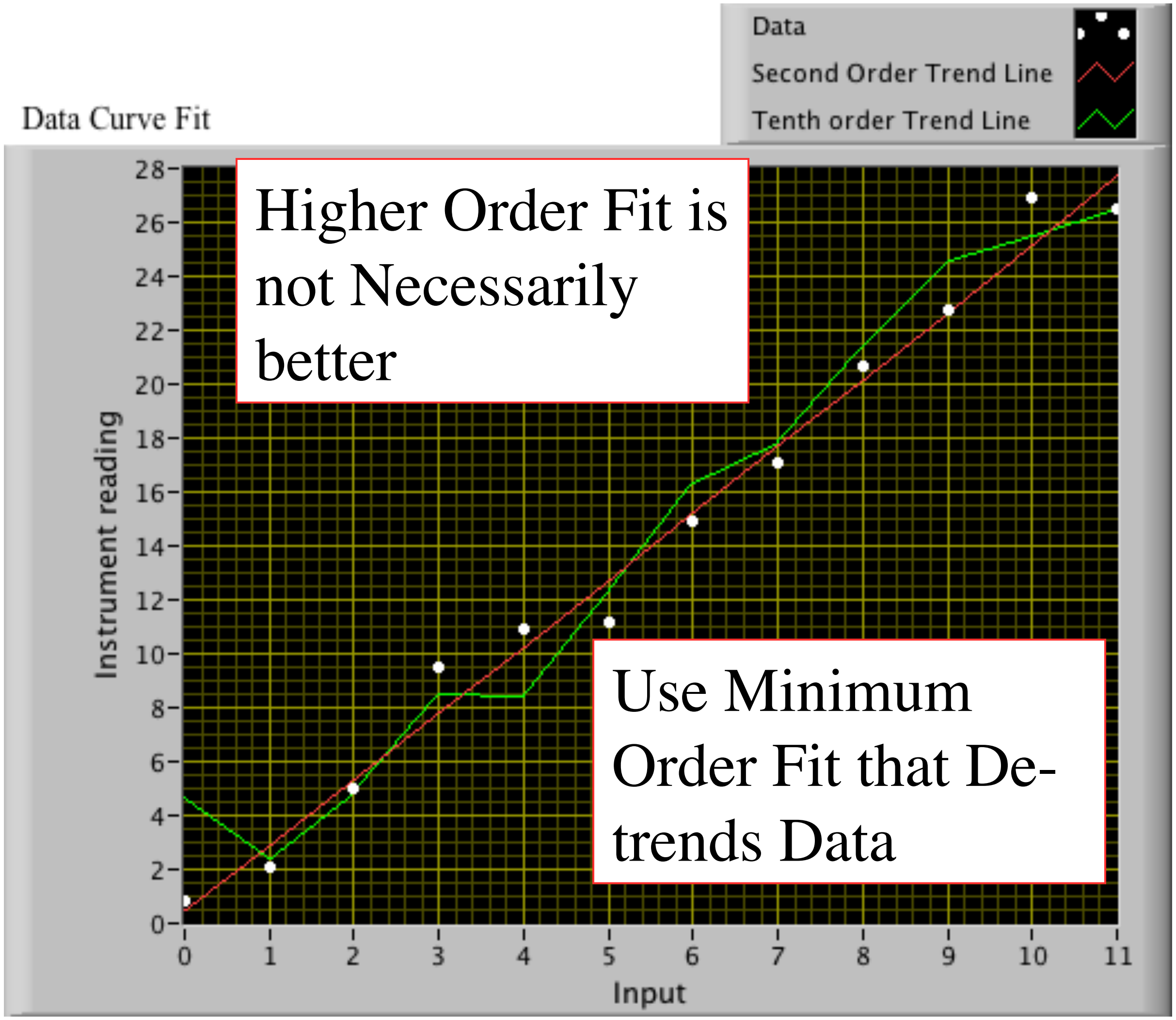
- 2nd order curve fit



- Which fit de-trends the data best?

The Trend Curve: Polynomial Least Squares (6)

New Data Set Old Calibration Coefficients



Second Order Fit

Mean Fit Error (μ)

-1.36928E-14

Standard Fit Error (σ)

1.04767 RMSE

Tenth Order Fit

Mean Fit Error (μ)

0.418559

Standard Fit Error (σ)

1.64355 RMSE

UtahState UNIVERSITY

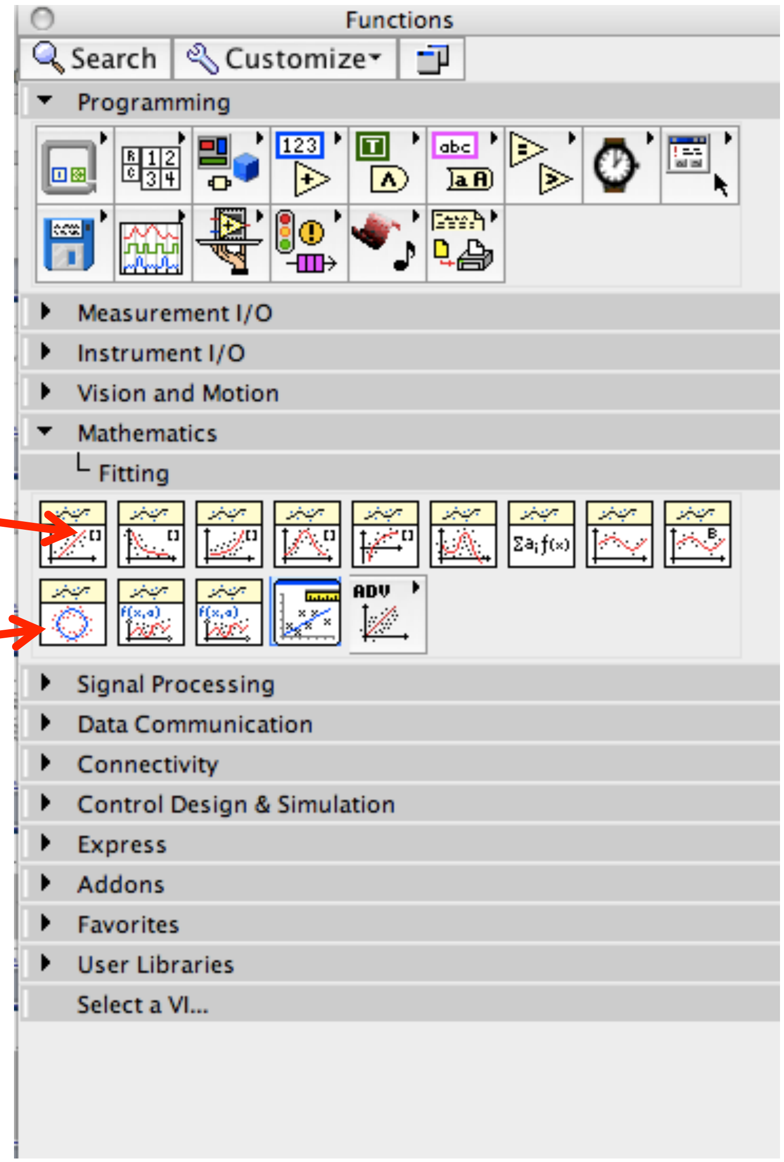
Mechanical & Aerospace Engineering

Labview Curve Fitting VI's

- "Right Click" on Block Diagram For "Function" Pallet

Simple Linear Fit

Polynominal Fit

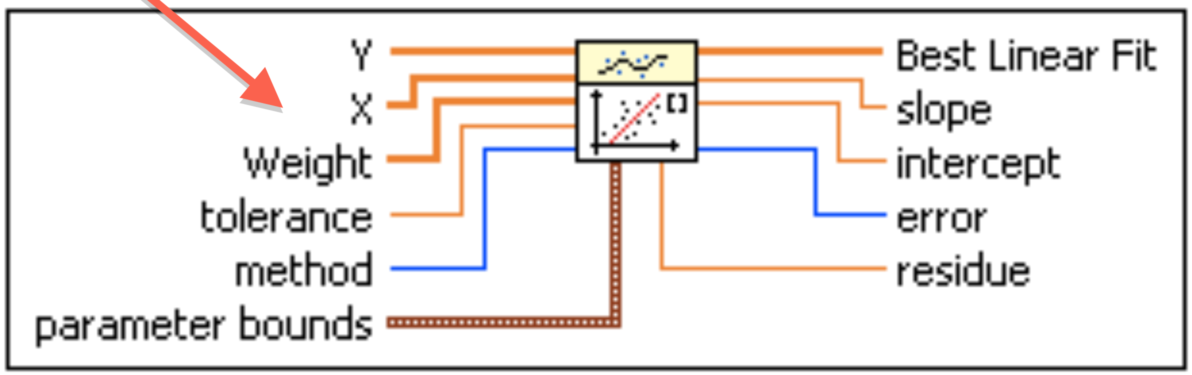


LabVIEW Linear Least Squares FIT VI

$$a_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2} \rightarrow \text{"slope"} \rightarrow$$

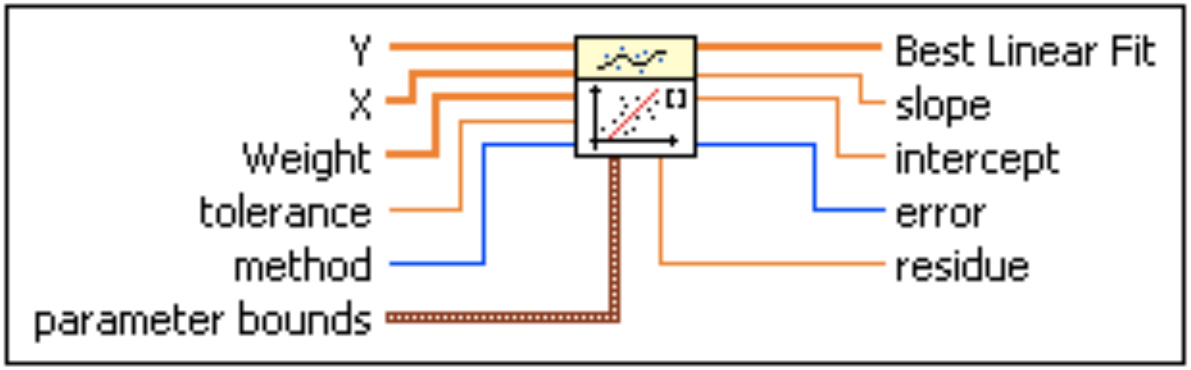
$$a_0 = \frac{\sum_{i=1}^{n}(x_i)^2\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)}{n\sum_{i=1}^{n}(x_i)^2 - \left(\sum_{i=1}^{n}(x_i)\right)^2} \rightarrow \text{"intercept"}$$

$$\widehat{y} = a_1 \cdot x + a_0$$



$$residue \equiv MSE = \frac{\sum_{i=0}^{n}\left[\widehat{y}_{(i)cal} - y_{(i)}\right]^2}{n-1}$$

LabVIEW Linear Least Squares FIT VI (2)



[DBL] **Y** is the array of dependent values. The length of **Y** must be greater than or equal to the number of unknown parameters.

[DBL] **X** is the array of independent values. **X** must be the same size as **Y**.

[DBL] **Weight** is the array of weights for the observations (**X**, **Y**). **Weight** must be the same size as **Y**. If you do not wire an input to **Weight**, the VI sets all elements of **Weight** to 1.

[DBL] **Best Linear Fit** returns the y-values of the fitted model.

DBL **slope** returns the slope of the fitted model.

DBL **intercept** returns the intercept of the fitted model.

LabVIEW Polynomial Least Squares FIT VI

- In general for an "mth" order fit

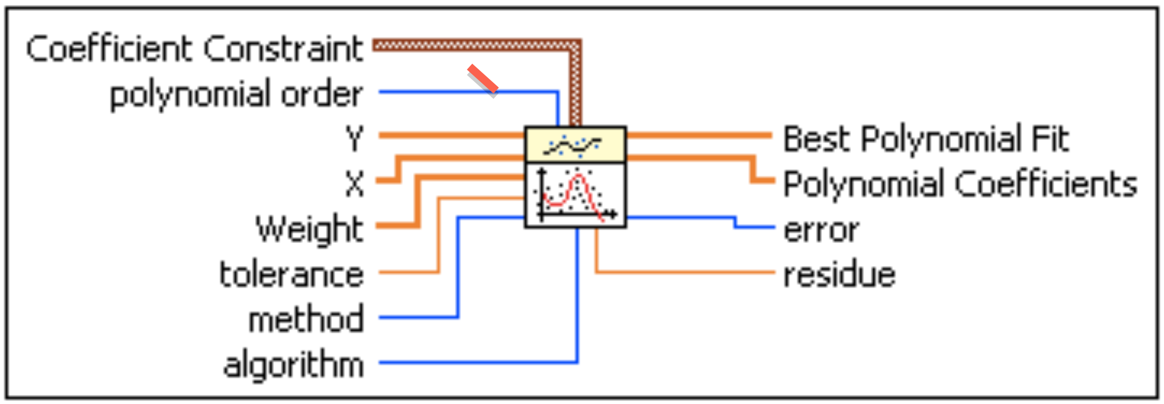
$$A = \left[X^T X\right]^{-1} X^T Y$$

- Numerical Methods required for Solution to this system

- Solution Algorithms

0	SVD (default)
1	Givens
2	Givens2
3	Householder
4	LU Decomposition
5	Cholesky
6	SVD for Rank Deficient H

- **LabVIEW polynomial fit VI**



$$\hat{y}_i = \sum_{j=1}^{m}\left(a_j \cdot x_i^{(j)}\right) + a_0 \rightarrow \begin{array}{l} m = fit\ order \\ i = data\ point \end{array}$$

LabVIEW Polynomial Least Squares VI (2)

I32 **polynomial order** specifies the order of the polynomial that fits to the data set. **polynomial order** must be greater than or equal to 0. If **polynomial order** is less than zero, this VI sets **Polynomial Coefficients** to an empty array and returns an error. In real applications, **polynomial order** is less than 10. If **polynomial order** is greater than 25, the VI sets the coefficients in **Polynomial Coefficients** to zero and returns a warning. The default is 2.

[DBL] **Y** is the array of dependent values. The number of sample points in **Y** must be greater than **polynomial order**. If the number of sample points is less than or equal to **polynomial order**, this VI sets **Polynomial Coefficients** to an empty array and returns an error.

[DBL] **X** is the array of independent values. The number of sample points in **X** must be greater than **polynomial order**. If the number of sample points is less than or equal to **polynomial order**, this VI sets **Polynomial Coefficients** to an empty array and returns an error. **X** must be the same size as **Y**.

[DBL] **Weight** is the array of weights for the observations (**X**, **Y**). **Weight** must be the same size as **Y**. If you do not wire an input to **Weight**, the VI sets all elements of **Weight** to 1. If an element of **Weight** is less than 0, the VI uses the absolute value of the element.

LabVIEW Polynomial Least Squares VI (3)

DBL **Best Polynomial Fit** returns the y-values of the polynomial curve that best fits the input values.

DBL **Polynomial Coefficients** returns the coefficients of the fitted model in ascending order of power. The total number of elements in **Polynomial Coefficients** is *m* + 1, where *m* is the polynomial order.

DBL **residue** returns the weighted mean error of the fitted model. If **method** is `Least Absolute Residual`, **residue** is the weighted mean absolute error. Otherwise, **residue** is the weighted mean square error.

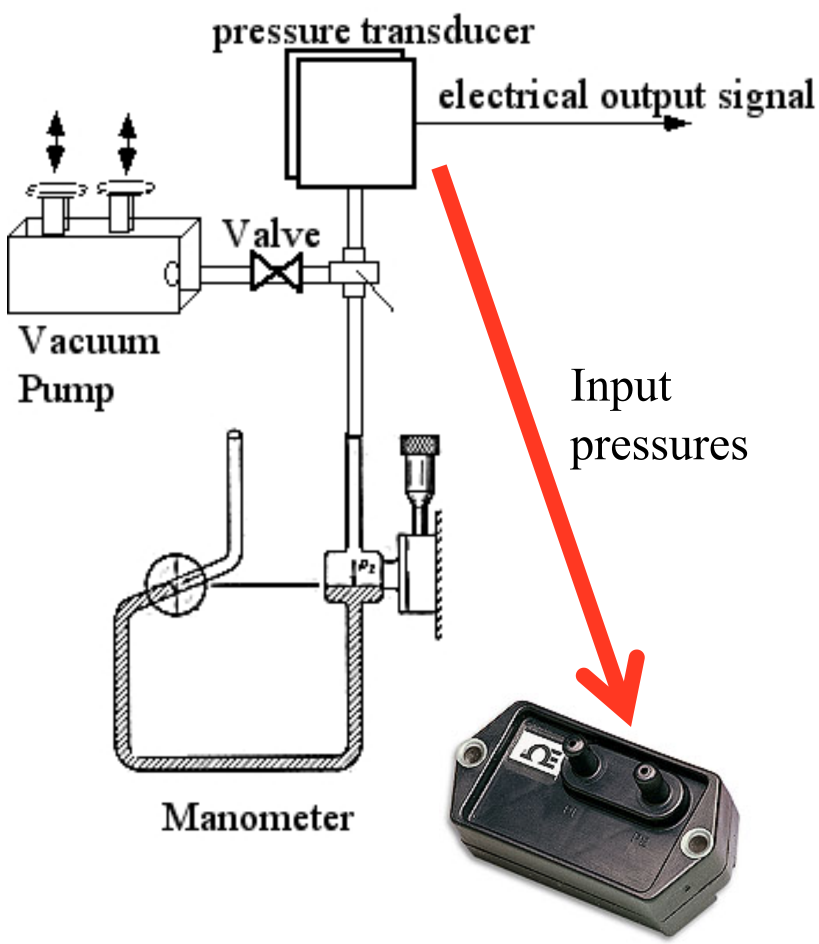
$$A = \left[X^T X\right]^{-1} X^T Y$$

$$A_{m\times 1} = \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_m \end{bmatrix}$$

$$\hat{y}_i = \sum_{j=1}^{m}\left(a_j \cdot x_i^{(j)}\right) + a_0$$

$\rightarrow$ $m = fit\ order$, $i = data\ point$

Example Curve Fitting Problem



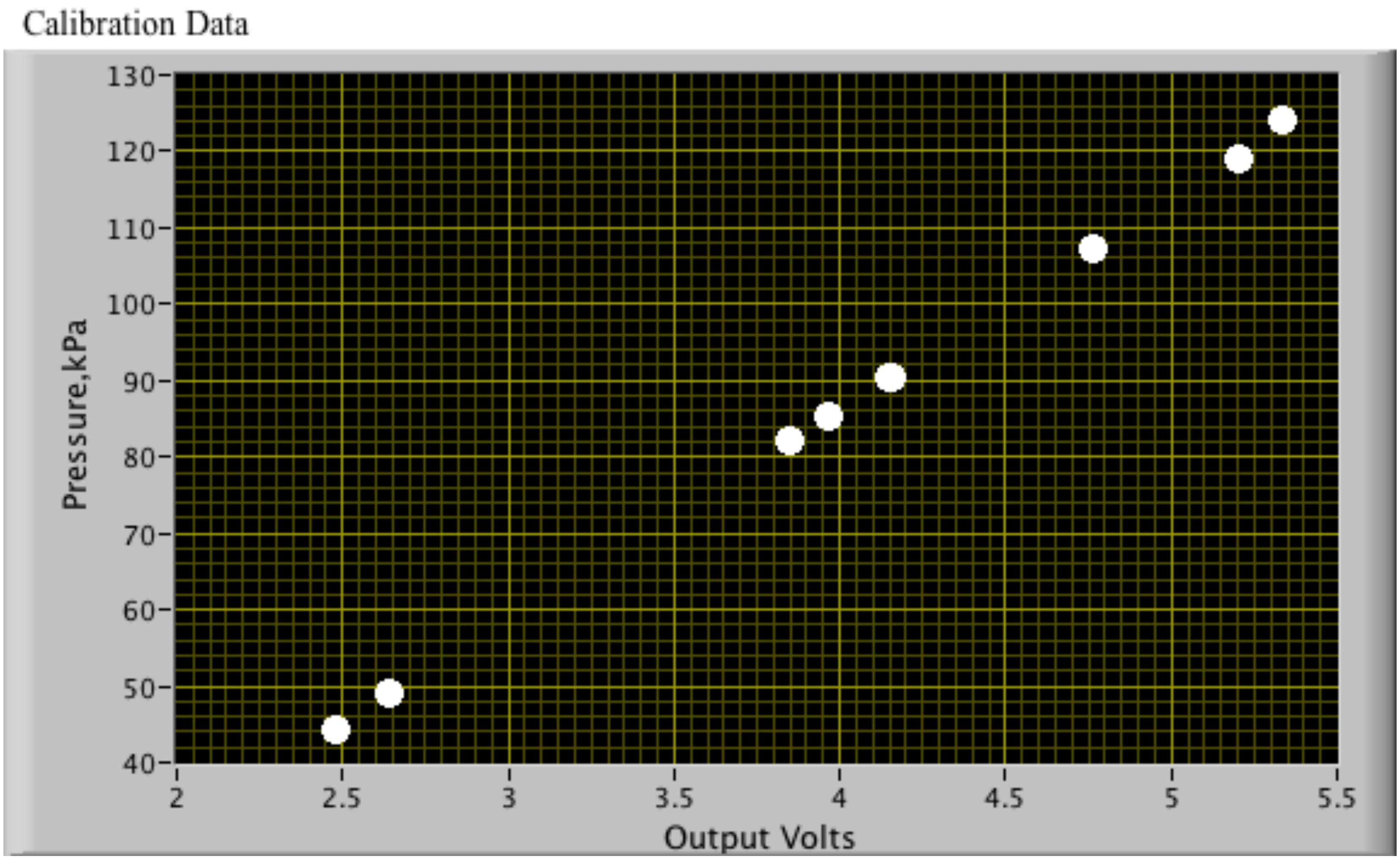
- Pressure Transducer Calibration
- Input Known Pressure Value
- Read Output Voltage From Transducer

Example Curve Fitting Problem (2)

Pressure Transducer Calibration 1

Data Point	Output Volts	Pressure, kPa
1	2.47	44.52
2	2.64	49.12
3	3.85	82.21
4	3.96	85.32
5	4.15	90.44
6	4.15	90.54
7	4.76	107.29
8	5.20	119.22
9	5.33	124.12

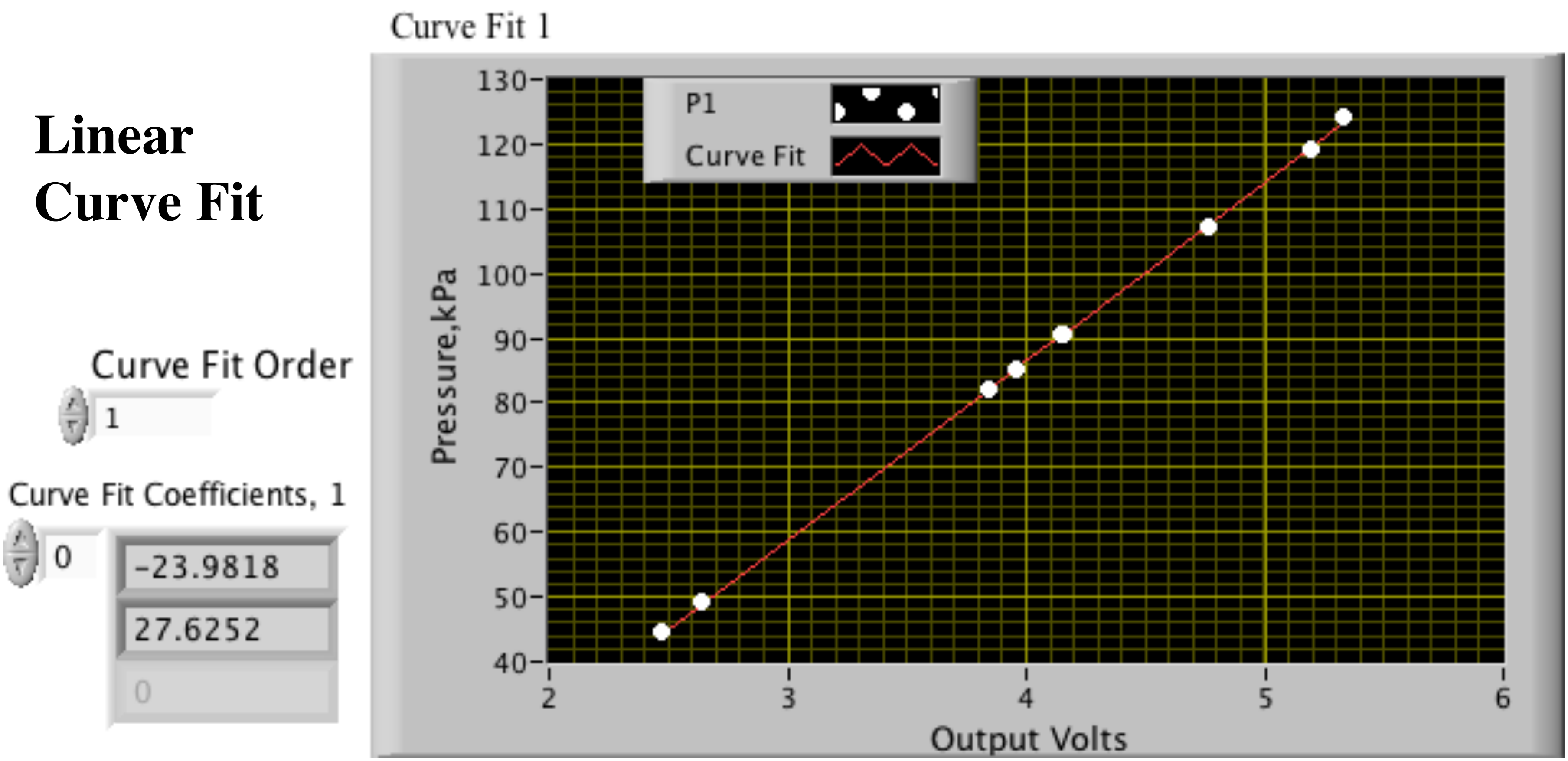
Example Curve Fitting Problem (3)



Example Curve Fitting Problem (4)

$$\widehat{P}_{cal} = 27.6252_{kPa/Volt} \cdot V_{out} - 23.9818_{kPa}$$

Linear Curve Fit



Example Curve Fitting Problem (5)

How Good is the Fit?

$$MSE = \frac{\sum_{i=0}^{n}\left[\widehat{P}_{(i)cal} - P_{(i)}\right]^2}{n-1} =$$ MSE Fit Error 0.110194 kPa^2

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}\left[\widehat{P}_{(i)cal} - P_{(i)}\right]^2}{n-1}} =$$ RMS Fit Error 0.331955 kPa

Example Curve Fitting Problem (6)

How Many Degrees of Freedom in Estimate of MSE?

$$DOF_{MSE} = \left(n - things\ you\ calculated\ from\ data\right) =$$
$$n - \left(a_1 + a_0\right)\text{-}MSE = n - \left(Fit\ Order + 2\right) = 9 - 3 = 6$$

What is the 95% confidence interval of this fit MAE?
… Use $\chi 2$ Test for 6 degrees of freedom

$$S_x^2 \approx MSE = \frac{1}{n-1}\left[\sum_{i=1}^{n}\left(y_i - \hat{y}\right)^2\right] \rightarrow \begin{bmatrix} DOF = n-3 \\ \alpha/2 = 0.025 \\ 1-\alpha/2 = 0.975 \end{bmatrix}$$

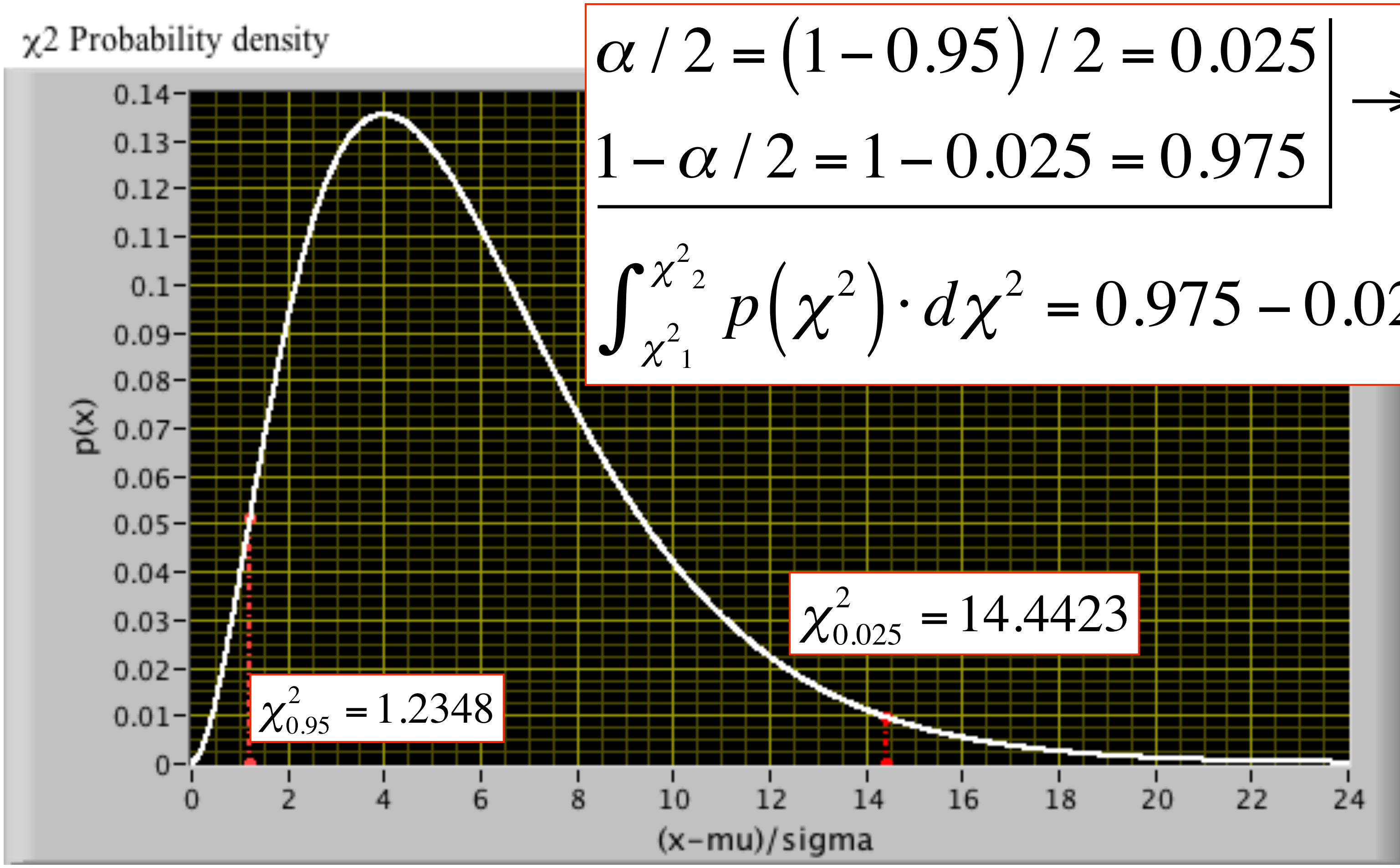
$$P\left[\frac{\nu_{DOF}\cdot S_x^2}{\chi^2_{0.025}} \le MSE \frac{\nu_{DOF}\cdot S_x^2}{\chi^2_{0.95}}\right] = 1 - 0.05 = 0.95$$

Example Curve Fitting Problem (7)

$\chi 2$ Density Distribution for 6 D.O.F.

$$\alpha/2 = (1-0.95)/2 = 0.025$$
$$1-\alpha/2 = 1-0.025 = 0.975$$
$$\rightarrow 95\%$$

$$\int_{\chi^2_1}^{\chi^2_2} p(\chi^2)\cdot d\chi^2 = 0.975 - 0.025 = 0.95$$



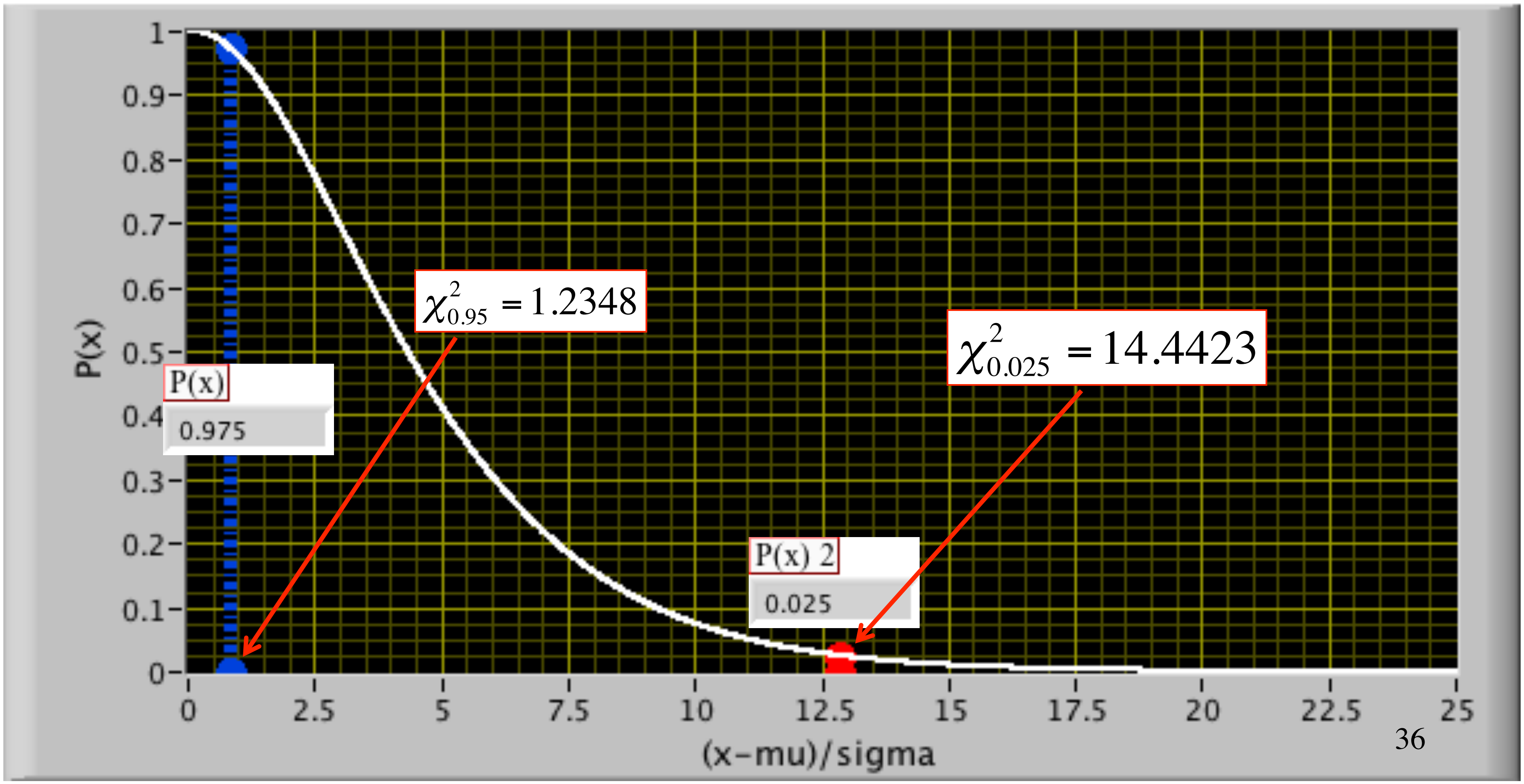
UtahState UNIVERSITY — Mechanical & Aerospace Engineering

Example Curve Fitting Problem (8)

$\chi 2$ Density Distribution for 6 D.O.F.

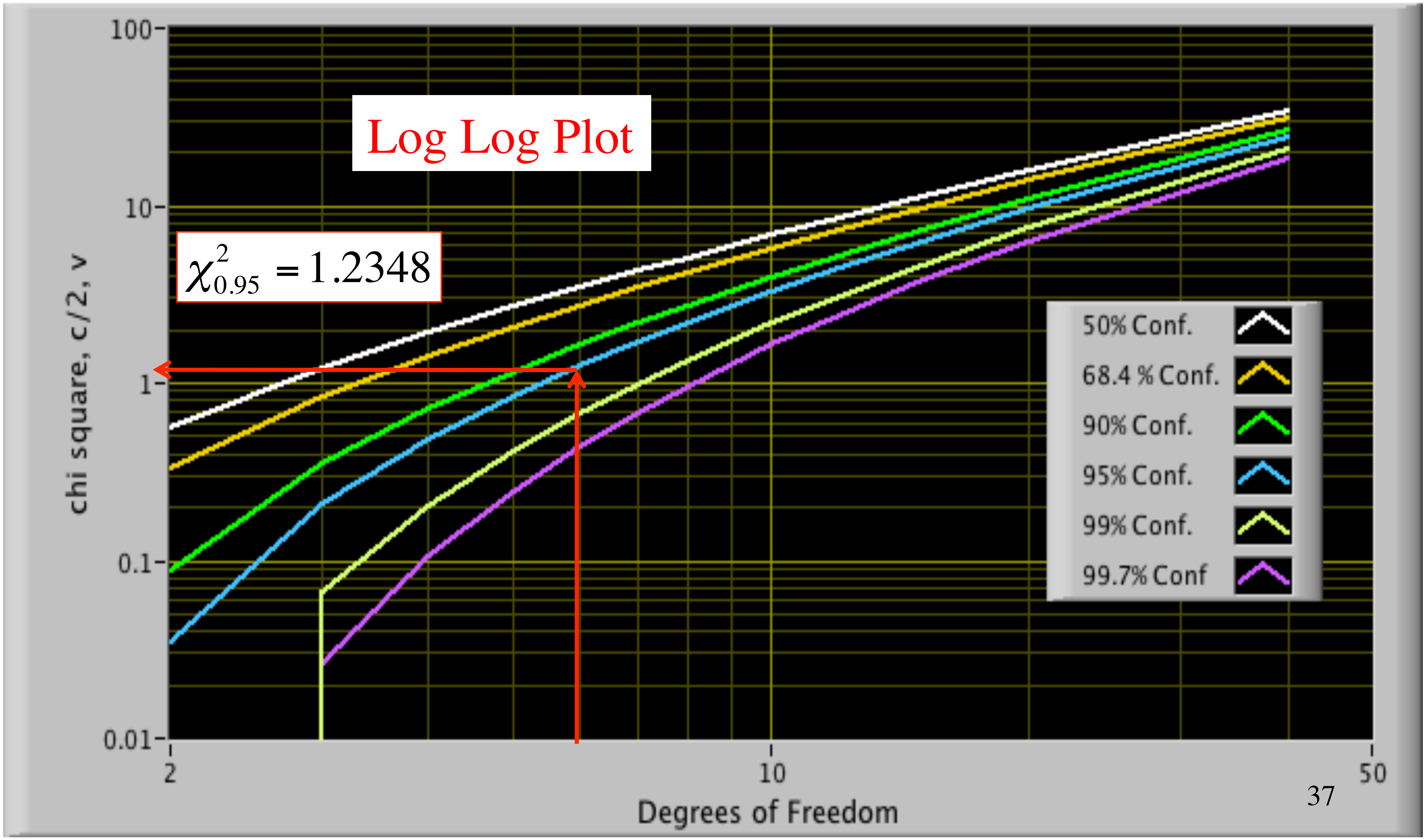
$$\int_{\chi^2_1}^{\chi^2_2} p\left(\chi^2\right)\cdot d\chi^2$$

1–χ2 Probability Integral

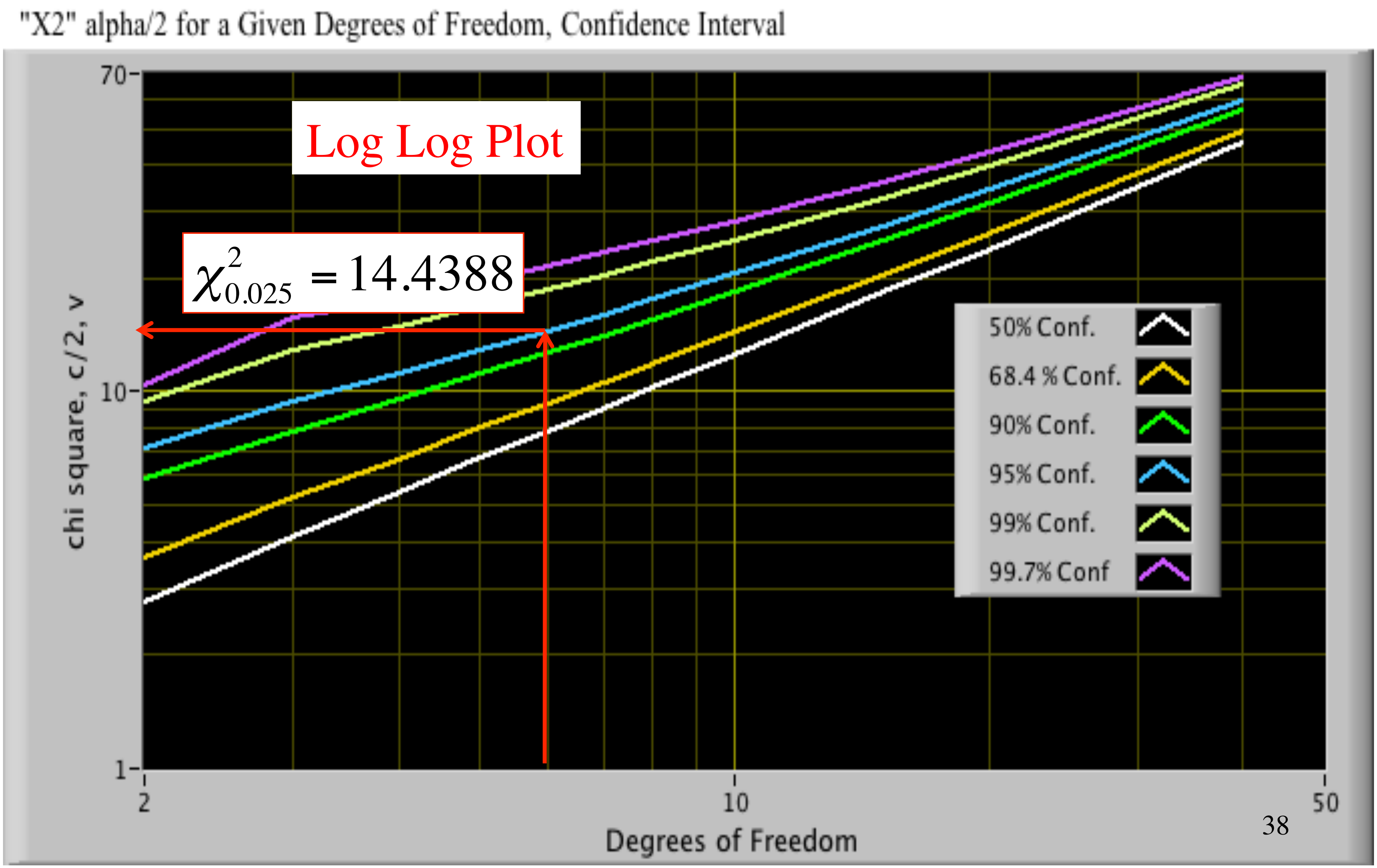


Example Curve Fitting Problem (9)

"X2" 1-alpha/2 for a Given Degrees of Freedom, Confidence Interval



Example Curve Fitting Problem (10)



Example Curve Fitting Problem (11)

Chi Square Statistics

D.O.F. on MSE	Alpha	Chi Alpha
6	0.95	1.23131
Alpha/2	**1- Alpha/2**	**Chi 1-Alpha**
0.025	0.975	14.4388

$$\frac{9-3}{14.4388} MSE_x \le MSE_{true} \le \frac{9-3}{1.231} MSE_x$$

MSE, RMSE Error range, kPa^2, kPa

MSE Min	MSE Fit Error	MSE Max
0.0457911	0.110194	0.536962
RMSE Min	**RMS Fit Error**	**RMSE Max**
0.213988	0.331955	0.732777

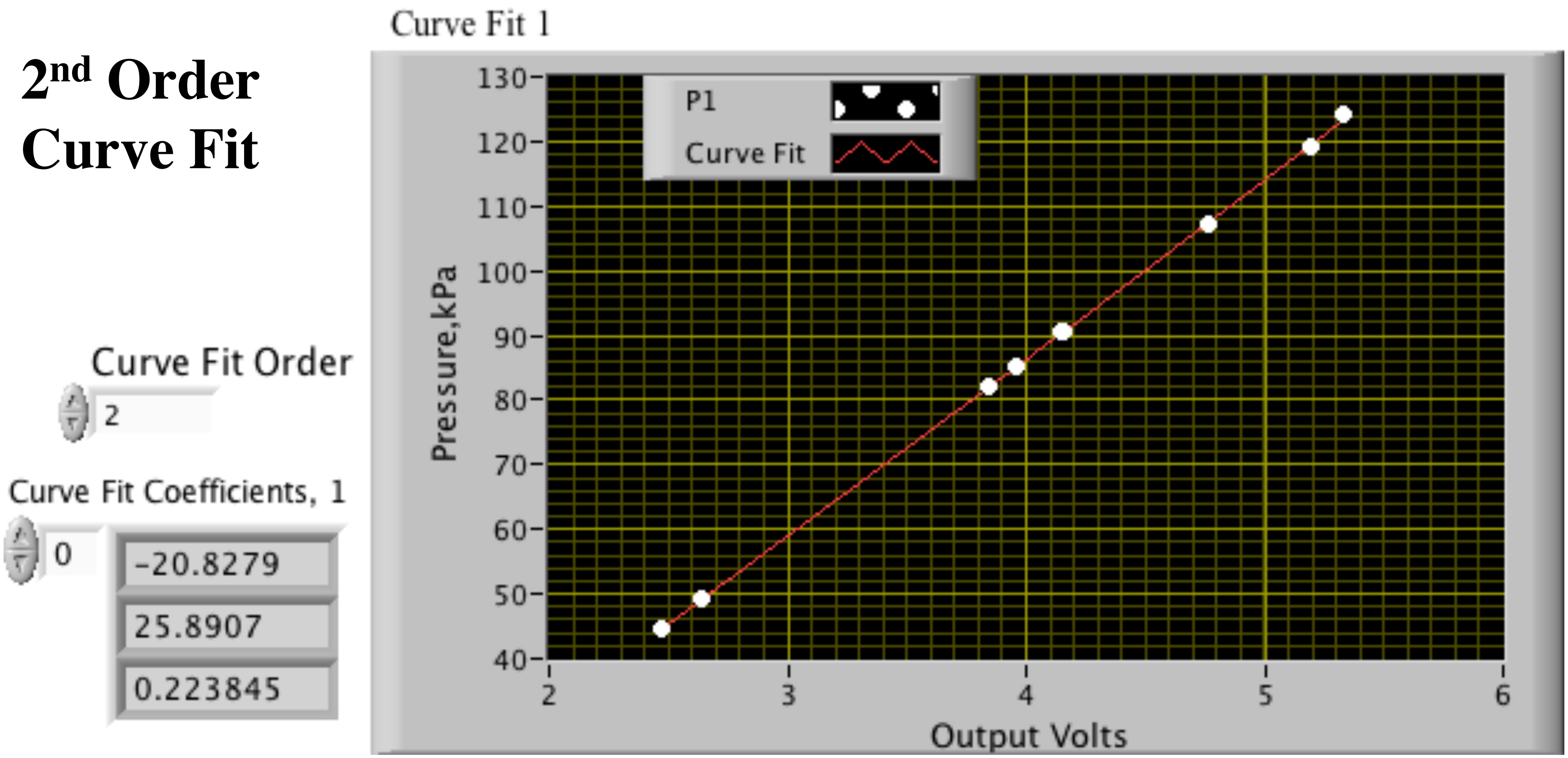
$$0.045791 \le MSE_{true} \le 0.53696$$

$$0.213988 \le RMSE_{true} \le 0.73278$$

Example Curve Fitting Problem (12)

$$\widehat{P}_{cal} = 0.223845_{kPa/V^2olt} \cdot V^2_{out} + 25.8907_{kPa/Volt} \cdot V_{out} - 20.8279_{kPa}$$

2nd Order Curve Fit



Example Curve Fitting Problem (13)

How Many Degrees of Freedom in Estimate of MSE?

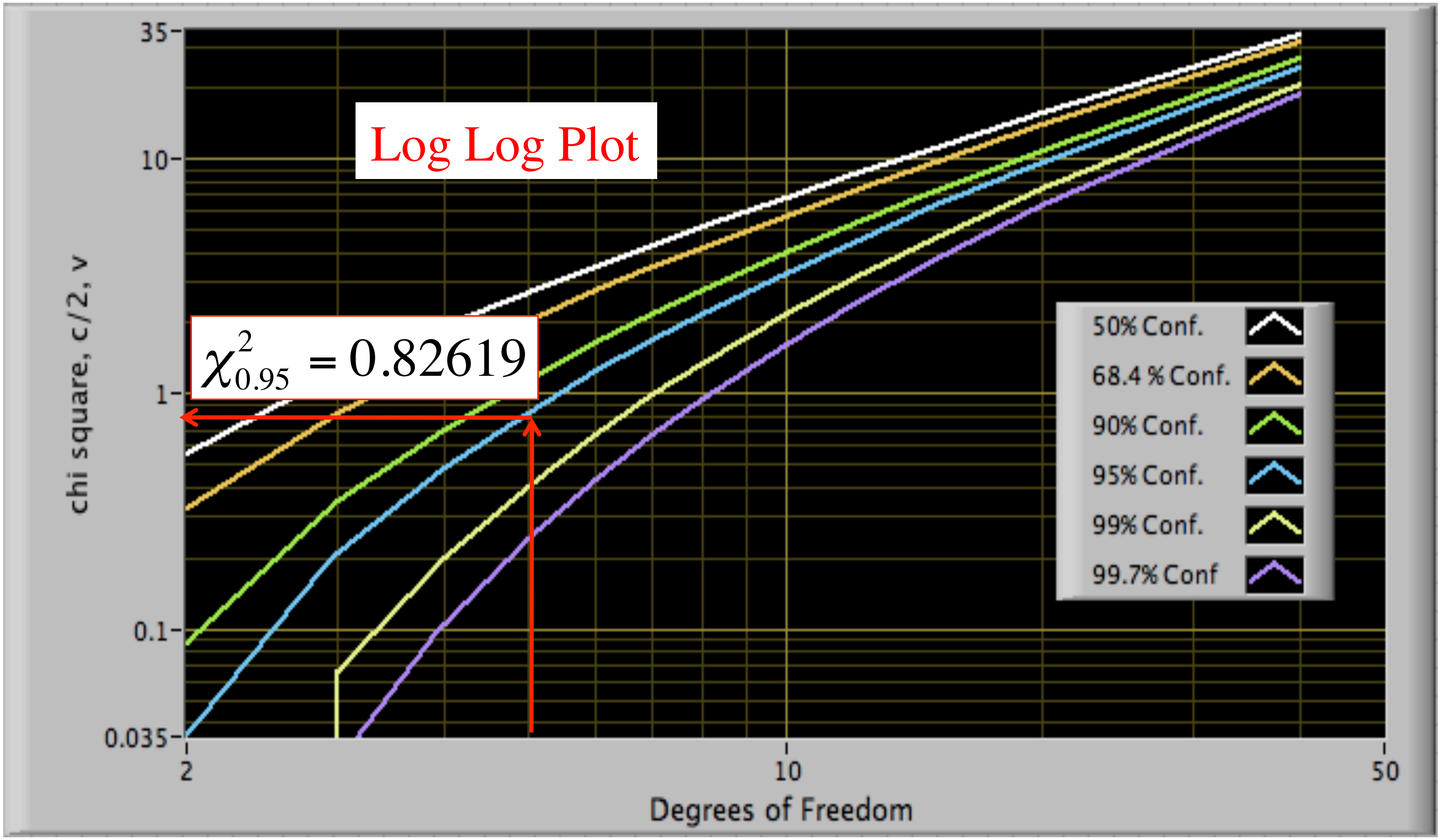
$$n-\left(a_1+a_0\right)\text{-}MSE = n-\left(Fit\ \ Order+2\right) = 9 - 4 = 5$$

What is the 95% confidence interval of this fit MAE?

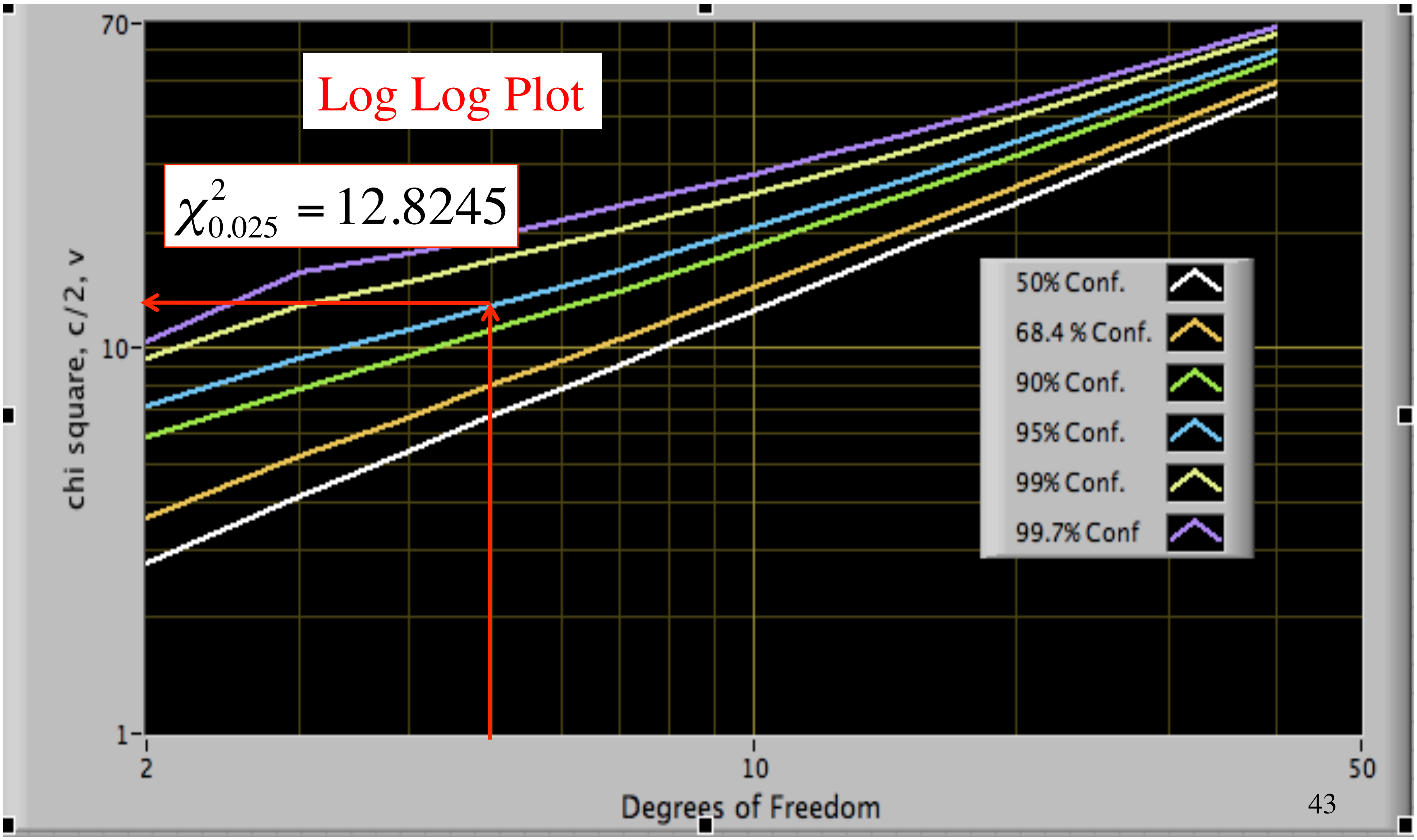
… Use $\chi 2$ Test

$$\frac{(n-1)S_x^2}{\chi^2_{\alpha/2}} < \sigma^2 < \frac{(n-1)S_x^2}{\chi^2_{1-\alpha/2}} \text{.....}(c\%) \qquad c\% = 1 - \alpha$$

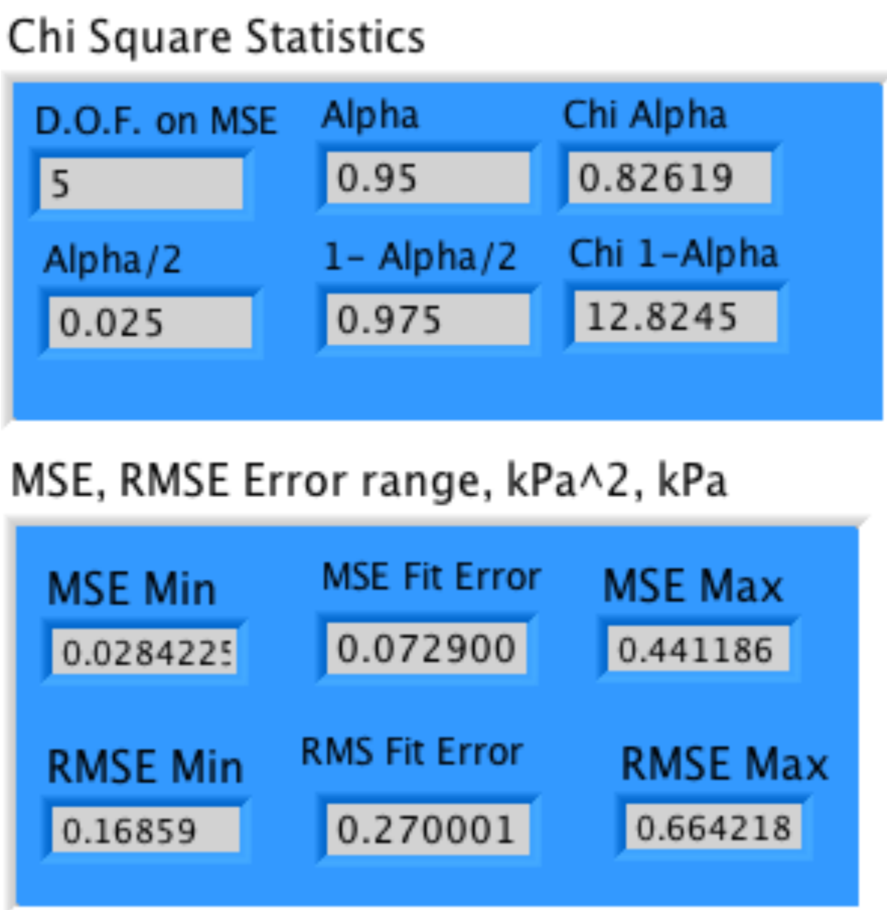
Example Curve Fitting Problem (14)



Example Curve Fitting Problem (15)



Example Curve Fitting Problem (12)



$$\frac{9-4}{12.8245}\cdot MSE_x \le MSE_{true} \le \frac{9-4}{0.82618}$$

$$0.028422 \le MSE_{true} \le 0.44186$$

Compare to First Order Fit…

$$0.045791 \le MSE_{true} \le 0.53696$$

First order fit

Some Improvement …
But not much …might be worth extra coefficient in curve fit