Course Instructor
Dr. Raymond C. Rumpf
Office: A-337
Phone: (915) 747-6958
E-Mail: rcrumpf@utep.edu

Computational Methods in EE

Topic 5b – Nonlinear Regression

EE 4386/5301 Computational Methods in EE

Outline

Computational Methods in EE

- Statement of the Problem
- Multiple-Parameter Taylor Series
- Formulation
- Algorithm
- Example – Fit to a Gaussian

Statement of Problem

Computational Methods in EE

We wish to fit a set of M measured data points to a nonlinear function $f(x)$.

$$y = f\left(x; a_0, a_1, \ldots, a_N\right)$$

$$y \equiv \text{measured value}$$
$$x \equiv \text{parameter from which } f \text{ is evaluated}$$
$$a_n \equiv \text{coefficients for the function fit}$$

The function can be anything like sine's, logarithms, exponentials…

$$f(x) = A + B\sin(Cx) \qquad f(x) = A + Be^{-Cx^2} \qquad f(x) = A + B\ln(Cx)$$

Multiple-Parameter Taylor Series

Computational Methods in EE

Recall the Taylor series for a single parameter…

$$f(x) = f(\tilde{x}) + \frac{f'(\tilde{x})}{1!}(x-\tilde{x}) + \frac{f''(\tilde{x})}{2!}(x-\tilde{x})^2 + \frac{f'''(\tilde{x})}{3!}(x-\tilde{x})^3 + \cdots$$

The two-parameter Taylor series is

$$f(x,y) = f(\tilde{x},\tilde{y}) + \frac{1}{1!}\left[\frac{\partial f(\tilde{x},\tilde{y})}{\partial x}(x-\tilde{x}) + \frac{\partial f(\tilde{x},\tilde{y})}{\partial y}(y-\tilde{y})\right]$$
$$+ \frac{1}{2!}\left[\frac{\partial^2 f(\tilde{x},\tilde{y})}{\partial x^2}(x-\tilde{x})^2 + 2\frac{\partial^2 f(\tilde{x},\tilde{y})}{\partial x \partial y}(x-\tilde{x})(y-\tilde{y}) + \frac{\partial^2 f(\tilde{x},\tilde{y})}{\partial y^2}(y-\tilde{y})^2\right]$$
$$\vdots$$

We will ignore the higher-order terms.

The N-parameter Taylor series using only first-order derivatives is

$$f(x_1, x_2, \ldots, x_N) \approx f(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N) + \sum_{n=1}^{N} \frac{\partial f}{\partial x_n}\Delta x_n \qquad \Delta x_n = x_n - \tilde{x}_n$$

Computational Methods in EE

Formulation

Solution Using Gauss-Newton Method (1 of 5)

Computational Methods in EE

We write our equation for each measured sample.

$$\begin{aligned} y_1 &= f\left(x_1; a_0, a_1, \ldots, a_N\right) + e_1 \\ y_2 &= f\left(x_2; a_0, a_1, \ldots, a_N\right) + e_2 \\ &\vdots \\ y_M &= f\left(x_M; a_0, a_1, \ldots, a_N\right) + e_M \end{aligned} \quad \rightarrow \quad \begin{aligned} y_1 &= f\left(x_1\right) + e_1 \\ y_2 &= f\left(x_2\right) + e_2 \\ &\vdots \\ \underbrace{y_M}&\underbrace{= f\left(x_M\right) + e_M}_{\text{Shorthand notation}} \end{aligned}$$

Solution Using Gauss-Newton Method (2 of 5)

Computational Methods in EE

We can convert the nonlinear equations into linear equations by expanding them into multi-parameter Taylor series.

$$y_1 = f(x_1) + \frac{\partial f(x_1)}{\partial a_0}\Delta a_0 + \frac{\partial f(x_1)}{\partial a_1}\Delta a_1 + \cdots + \frac{\partial f(x_1)}{\partial a_N}\Delta a_N + e_1$$

$$y_2 = f(x_2) + \frac{\partial f(x_2)}{\partial a_0}\Delta a_0 + \frac{\partial f(x_2)}{\partial a_1}\Delta a_1 + \cdots + \frac{\partial f(x_2)}{\partial a_N}\Delta a_N + e_2$$

$$\vdots$$

$$y_M = f(x_M) + \frac{\partial f(x_M)}{\partial a_0}\Delta a_0 + \frac{\partial f(x_M)}{\partial a_1}\Delta a_1 + \cdots + \frac{\partial f(x_M)}{\partial a_N}\Delta a_N + e_M$$

Solution Using Gauss-Newton Method (3 of 5)

Computational Methods in EE

Now that our equations are linear, we can put them in matrix form.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{M-1} \\ y_M \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{M-1}) \\ f(x_M) \end{bmatrix} + \begin{bmatrix} \frac{\partial f(x_1)}{\partial a_0} & \frac{\partial f(x_1)}{\partial a_1} & \cdots & \frac{\partial f(x_1)}{\partial a_N} \\ \frac{\partial f(x_2)}{\partial a_0} & \frac{\partial f(x_2)}{\partial a_1} & \cdots & \frac{\partial f(x_2)}{\partial a_N} \\ \frac{\partial f(x_3)}{\partial a_0} & \frac{\partial f(x_3)}{\partial a_1} & \cdots & \frac{\partial f(x_3)}{\partial a_N} \\ \vdots & \vdots & & \\ \frac{\partial f(x_{M-1})}{\partial a_0} & \frac{\partial f(x_{M-1})}{\partial a_1} & \cdots & \frac{\partial f(x_{M-1})}{\partial a_N} \\ \frac{\partial f(x_M)}{\partial a_0} & \frac{\partial f(x_M)}{\partial a_1} & \cdots & \frac{\partial f(x_M)}{\partial a_N} \end{bmatrix} \begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \vdots \\ \Delta a_N \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_{M-1} \\ e_M \end{bmatrix}$$

$$[y] = [f] + [Z][\Delta a] + [e]$$

Solution Using Gauss-Newton Method (4 of 5)

Computational Methods in EE

We solve for $[\Delta a]$ using least-squares.

$$[y]=[f]+[Z][\Delta a]+\cancel{[e]}$$

$$[y]-[f]=[Z][\Delta a]$$

$$[d]=[Z][\Delta a]$$

$$[Z]^T[d]=[Z]^T[Z][\Delta a]$$

$$[\Delta a]=\left([Z]^T[Z]\right)^{-1}[Z]^T[d]$$

Let $[d]=[y]-[f]$

$[d]$ is a new error function.
$[y]$ contains the measured values and $[f]$ contains the calculated values from the curve fit $f(x)$.

Solution Using Gauss-Newton Method (5 of 5)

Computational Methods in EE

We now have

$$[\Delta a]=\left([Z]^T[Z]\right)^{-1}[Z]^T[d]$$

This will only tell us how to change the coefficients $[a]$ given an initial guess $[a]_0$.

We will have to iterate to find the actual coefficients of $[a]$.

Computational Methods in EE

Algorithm

Algorithm for Nonlinear Regression (1 of 2)

Computational Methods in EE

Step 0 – Derive analytical expressions for partial derivatives of $f(x)$.

Step 1 – Make an initial guess at your coefficients.

$$[a]_0 = \begin{bmatrix} a_{0,0} & a_{1,0} & \cdots & a_{N,0} \end{bmatrix}^T$$

Make an intelligent guess!

Step 2 – Calculate the function $[f]_i$ at all measured points given the current value of the coefficients $[a]_i$.

$$[f]_i = \begin{bmatrix} f_i(x_1) & f_i(x_2) & f_i(x_3) & \cdots & f_i(x_{M-1}) & f_i(x_M) \end{bmatrix}^T$$

Step 3 – Calculate the error $[d]_i$ in the estimate of $[f]_i$.

$$[d]_i = [y] - [f]_i$$

```
% CALCULATE d
d = fm(:) - f;
```

Algorithm for Nonlinear Regression (2 of 2)

Computational Methods in EE

Step 4 – Construct the $[Z]_i$ matrix given the current coefficients $[a]_i$.

$$[Z]_i = \begin{bmatrix} \dfrac{\partial f_i(x_1)}{\partial a_0} & \dfrac{\partial f_i(x_1)}{\partial a_1} & \cdots & \dfrac{\partial f_i(x_1)}{\partial a_N} \\ \dfrac{\partial f_i(x_2)}{\partial a_0} & \dfrac{\partial f_i(x_2)}{\partial a_1} & \cdots & \dfrac{\partial f_i(x_2)}{\partial a_N} \\ \vdots & \vdots & \ddots & \\ \dfrac{\partial f_i(x_M)}{\partial a_0} & \dfrac{\partial f_i(x_M)}{\partial a_1} & & \dfrac{\partial f_i(x_M)}{\partial a_N} \end{bmatrix}$$

We will need to derive $N+1$ derivatives, one for each coefficient a_n.

We will need to evaluate each of these $N+1$ derivatives at all M points.

This is a lot of work!

Step 5 – Solve for $[\Delta a]_i$ using least squares.

$$[\Delta a]_i = \left([Z]_i^T [Z]_i\right)^{-1} \left([Z]_i^T [d]_i\right)$$

Algorithm for Nonlinear Regression (3 of 3)

Computational Methods in EE

Step 6 – Adjust $[a]_i$ using $[\Delta a]_i$.

$$[a]_{i+1} = [a]_i + [\Delta a]_i$$

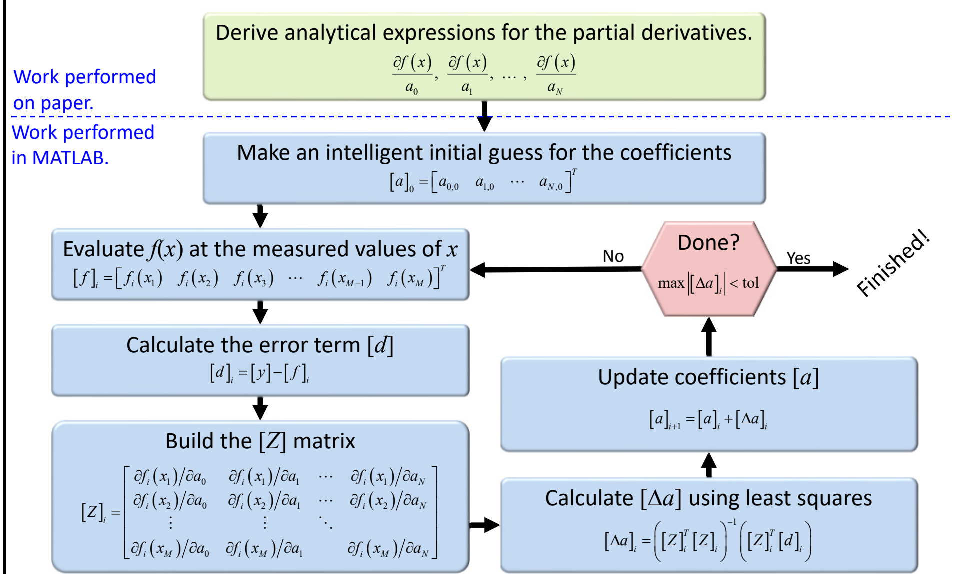
Step 7 – Go back to Step 2 until converged.

Convergence happens when the change in coefficient values $[\Delta a]_i$ falls below some tolerance.

$$\left|\frac{a_{n,i+1} - a_{n,i}}{a_{n,i+1}}\right| \cdot 100\% \le \text{tol} \qquad \text{for all } a_n$$

Block Diagram

Computational Methods in EE



Notes for Nonlinear Regression

Computational Methods in EE

- Method can be used to fit any set of measured data to any function who's first-derivative exists.
- This method does not always converge. Try different initial guesses.
- Use the most intelligent initial guess as possible.
- Linear regression is a special case of nonlinear regression.

Computational Methods in EE

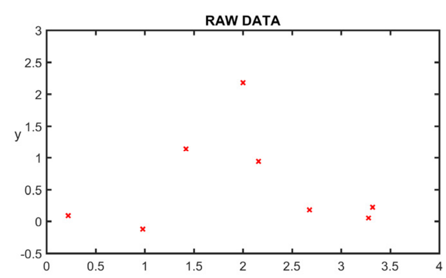
Example – Fit to a Gaussian

Example 1 – Fit to a Gaussian

Computational Methods in EE

We wish to fit a set of measured data to a standard Gaussian function.

$$f(x) = A\exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right]$$




```
% MEASURED DATA
xm = [ -0.14 ; 0.22 ;  0.98 ; 1.42 ; 2.00 ; 2.16 ; 2.68 ; 3.28 ; 3.32  ];
fm = [  0.01 ; 0.09 ; -0.12 ; 1.14 ; 2.18 ; 0.94 ; 0.18 ; 0.05 ; 0.22  ];
```

Example 1 – Fit to a Gaussian

Computational Methods in EE

Formulation Step 1 – Identify the unknown parameters.

$$[a] = \begin{bmatrix} A \\ x_0 \\ \sigma \end{bmatrix}$$

In this case, we have 3 unknown parameters: A, x_0, and σ.

Example 1 – Fit to a Gaussian

Computational Methods in EE

Formulation Step 2 – Derive terms in $[Z]$ matrix.

$$\frac{\partial f(x)}{\partial A} \qquad \frac{\partial f(x)}{\partial x_0} \qquad \frac{\partial f(x)}{\partial \sigma}$$

$$[Z] = \begin{bmatrix} \frac{\partial f(x_1)}{\partial A} & \frac{\partial f(x_1)}{\partial x_0} & \frac{\partial f(x_1)}{\partial \sigma} \\ \frac{\partial f(x_2)}{\partial A} & \frac{\partial f(x_2)}{\partial x_0} & \frac{\partial f(x_2)}{\partial \sigma} \\ & \vdots & \\ \frac{\partial f(x_M)}{\partial A} & \frac{\partial f(x_M)}{\partial x_0} & \frac{\partial f(x_M)}{\partial \sigma} \end{bmatrix}$$

$$\frac{\partial f(x)}{\partial A} = \exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right] = \frac{f(x)}{A}$$

$$\frac{\partial f(x)}{\partial x_0} = A\exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right]\cdot\left\{-\frac{2}{\sigma^2}(x-x_0)\cdot(-1)\right\} = \frac{2(x-x_0)}{\sigma^2}f(x)$$

$$\frac{\partial f(x)}{\partial \sigma} = A\exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right]\cdot\left\{-(x-x_0)^2\cdot\left(-\frac{2}{\sigma^3}\right)\right\} = \frac{2(x-x_0)^2}{\sigma^3}f(x)$$

Example 1 – Fit to a Gaussian

Computational Methods in EE

The $[Z]$ matrix is

$$[Z] = \begin{bmatrix} \frac{f(x_1)}{A} & \frac{2(x_1 - x_0)}{\sigma^2} f(x_1) & \frac{2(x_1 - x_0)^2}{\sigma^3} f(x_1) \\ \frac{f(x_2)}{A} & \frac{2(x_2 - x_0)}{\sigma^2} f(x_2) & \frac{2(x_2 - x_0)^2}{\sigma^3} f(x_2) \\ & \vdots & \\ \frac{f(x_M)}{A} & \frac{2(x_M - x_0)}{\sigma^2} f(x_M) & \frac{2(x_M - x_0)^2}{\sigma^3} f(x_M) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{A} f(\mathbf{x}) & \frac{2(\mathbf{x} - x_0)}{\sigma^2} f(\mathbf{x}) & \frac{2(\mathbf{x} - x_0)^2}{\sigma^3} f(\mathbf{x}) \end{bmatrix}$$

← We will calculate $[Z]$ this way.

Example 1 – Fit to a Gaussian

Computational Methods in EE

An example of a more intelligent initial guess…

$$f(x) = A \exp\left[-\left(\frac{x - x_0}{\sigma} \right)^2 \right]$$

$$A_1 \leftarrow \max[f_i]$$

Consider using the maximum value of f_i in your measured points as your initial guess for A.

$$x_{0,1} \leftarrow \text{average}[x_i]$$

Consider using the average value of x_i as your initial guess for x_0.

$$\sigma \leftarrow s\left(\max[x_i] - \min[x_i]\right)$$

The standard deviation will likely be on the same order of magnitude as the range of values of x_i. Maybe choose $s = 0.5$?

```
% SMART INITIAL GUESS
A  = max(fm);
x0 = mean(xm);
s  = 0.5*(max(xm) - min(xm));
```

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step a – Calculate $[f_i]$

$$[f]_i = \left[f_i(x_1) \quad f_i(x_2) \quad f_i(x_3) \quad \cdots \quad f_i(x_{M-1}) \quad f_i(x_M) \right]^T$$

```
% COMPUTE COLUMN VECTOR F
f = A*exp(-((xm - x0)/s).^2);
```

Here is $[f]$ over ten iterations…

0.6452	0.0393	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.9780	0.1218	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.7707	0.6366	0.0160	0.2244	0.0088	0.0234	0.0131	0.0141	0.0141	0.0141
2.0931	1.0530	0.4043	1.7341	0.7211	1.2323	1.1202	1.1358	1.1357	1.1357
2.1414	1.2289	1.5594	1.3029	2.3184	2.0714	2.1838	2.1829	2.1833	2.1833
2.0714	1.1582	1.2653	0.6630	1.2660	0.9323	0.9277	0.9363	0.9361	0.9362
1.6520	0.7047	0.1130	0.0124	0.0111	0.0042	0.0026	0.0028	0.0028	0.0028
1.0165	0.2228	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.9757	0.2018	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step b – Calculate $[d_i]$

$$[d]_i = [y] - [f]_i$$

```
% CALCULATE d
d = fm(:) - f;
```

Here is $[d]$ over ten iterations…

-0.6352	-0.0293	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
-0.8880	-0.0318	0.0900	0.0899	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
-1.8907	-0.7566	-0.1360	-0.3444	-0.1288	-0.1434	-0.1331	-0.1341	-0.1341	-0.1341
-0.9531	0.0870	0.7357	-0.5941	0.4189	-0.0923	0.0198	0.0042	0.0043	0.0043
0.0386	0.9511	0.6206	0.8771	-0.1384	0.1086	-0.0038	-0.0029	-0.0033	-0.0033
-1.1314	-0.2182	-0.3253	0.2770	-0.3260	0.0077	0.0123	0.0037	0.0039	0.0038
-1.4720	-0.5247	0.0670	0.1676	0.1689	0.1758	0.1774	0.1772	0.1772	0.1772
-0.9665	-0.1728	0.0497	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500	0.0500
-0.7557	0.0182	0.2198	0.2200	0.2200	0.2200	0.2200	0.2200	0.2200	0.2200

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step c – Build $[Z_i]$

$$\mathbf{z}_1 = \frac{1}{A} f(\mathbf{x}) \qquad \mathbf{z}_2 = \frac{2(\mathbf{x}-x_0)}{\sigma^2} f(\mathbf{x})$$

$$\mathbf{z}_3 = \frac{2(\mathbf{x}-x_0)^2}{\sigma^3} f(\mathbf{x})$$

$$\mathbf{Z}_i = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \mathbf{z}_3]$$

```
% FORM Z MATRIX
z1 = f/A;
z2 = 2*f.*(xm - x0)/s^2;
z3 = 2*f.*(xm - x0).^2/s^3;
Z  = [ z1 z2 z3 ];
```

$$\mathbf{Z}_1 = \begin{bmatrix} 0.2960 & -0.8230 & 0.9081 \\ 0.4486 & -1.0123 & 0.9063 \\ 0.8123 & -0.9335 & 0.4257 \\ 0.9601 & -0.4880 & 0.0984 \\ 0.9823 & 0.3307 & 0.0442 \\ 0.9502 & 0.5414 & 0.1224 \\ 0.7578 & 1.0058 & 0.5297 \\ 0.4663 & 1.0265 & 0.8966 \\ 0.4476 & 1.0114 & 0.9068 \end{bmatrix} \qquad \mathbf{Z}_{10} = \begin{bmatrix} 0.0000 & -0.0000 & 0.0000 \\ 0.0000 & -0.0000 & 0.0000 \\ 0.0042 & -0.1944 & 0.4552 \\ 0.3352 & -6.9938 & 7.3115 \\ 0.6445 & 8.5245 & 5.6503 \\ 0.2763 & 6.2539 & 7.0924 \\ 0.0008 & 0.0436 & 0.1163 \\ 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

Topic 5b -- Nonlinear regression 25

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step d – Solve for $[\Delta a]_i$

$$[\Delta a]_i = \left([Z]_i^T [Z]_i\right)^{-1} \left([Z]_i^T [d]_i\right)$$

```
% CALCULATE [Da]
da = (Z'*Z)\(Z'*d);
```

Here is $[\Delta a]$ over ten iterations…

-0.9316	0.3327	0.7433	0.5188	0.3549	0.2022	-0.0135	0.0010	-0.0000	0.0000
0.0958	0.0823	-0.2859	0.1775	-0.0723	0.0092	-0.0006	0.0000	-0.0000	0.0000
-0.6519	-0.6268	-0.0058	-0.0881	-0.0027	-0.0172	0.0022	-0.0001	0.0000	-0.0000

Topic 5b -- Nonlinear regression 26

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step e – Update coefficients $[a]_i$

$$[a]_{i+1} = [a]_i + [\Delta a]_i$$

```
% UPDATE COEFFICIENTS [a]
A   = A + da(1);
x0  = x0 + da(2);
s   = s + da(3);
```

Here is $[a] = [A \;\; x_0 \;\; \sigma]^T$ over ten iterations…

1.2484	1.5810	2.3244	2.8432	3.1981	3.4003	3.3868	3.3878	3.3878	3.3878
1.8647	1.9470	1.6611	1.8386	1.7663	1.7755	1.7749	1.7750	1.7750	1.7750
1.0781	0.4513	0.4454	0.3574	0.3546	0.3374	0.3396	0.3395	0.3395	0.3395

Example 1 – Fit to a Gaussian

Computational Methods in EE

The main loop…

Step f – Calculate error and check for convergence

$$\varepsilon = \max\left[\left|\frac{a_{n,i+1} - a_{n,i}}{a_{n,i+1}}\right|\right]$$

```
% CALCULATE MAXIMUM ERROR
err = max(abs(da./[A;x0;s]));
```

Here is `err` over ten iterations…

0.7463	1.3889	0.3198	0.2465	0.1110	0.0595	0.0065	0.0003	0.0000	0.0000

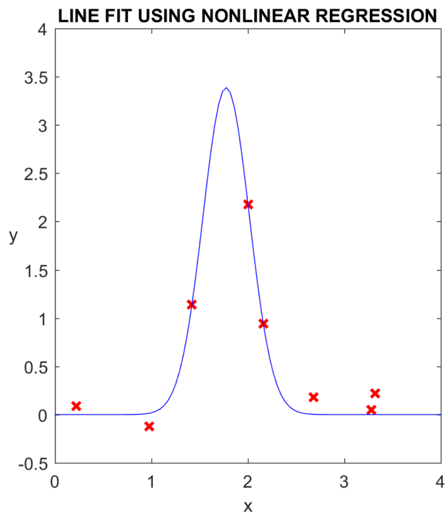
Example 1 – Fit to a Gaussian

Computational Methods in EE

The final answer is…

$$A = 3.3878$$
$$x_0 = 1.7750$$
$$\sigma = 0.3395$$

$$f(x) = A\exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right] = 3.3878\exp\left[-\left(\frac{x-1.7750}{0.3395}\right)^2\right]$$



Example 1 – Fit to a Gaussian

Computational Methods in EE

Here are the first 10 iterations of the algorithm visualized…