System of linear equations:

$$
\begin{aligned}
a_{00}x_0 + a_{01}x_1 + a_{02}x_2 + \cdots + a_{0,N-1}x_{N-1} &= b_0 \\
a_{10}x_0 + a_{11}x_1 + a_{12}x_2 + \cdots + a_{1,N-1}x_{N-1} &= b_1 \\
a_{20}x_0 + a_{21}x_1 + a_{22}x_2 + \cdots + a_{2,N-1}x_{N-1} &= b_2 \\
\cdots \qquad\qquad\qquad\qquad &\cdots \\
a_{M-1,0}x_0 + a_{M-1,1}x_1 + \cdots + a_{M-1,N-1}x_{N-1} &= b_{M-1}
\end{aligned}
$$

Here the N unknowns x_j, $j = 0, 1, \ldots, N-1$ are related by M equations. The coefficients a_{ij} with $i = 0, 1, \ldots, M-1$ and $j = 0, 1, \ldots, N-1$ are known numbers, as are the *right-hand side* quantities b_i, $i = 0, 1, \ldots, M-1$.

Matrix-vector notation:

$$
\begin{aligned}
a_{00}x_0 + a_{01}x_1 + a_{02}x_2 + \cdots + a_{0,N-1}x_{N-1} &= b_0 \\
a_{10}x_0 + a_{11}x_1 + a_{12}x_2 + \cdots + a_{1,N-1}x_{N-1} &= b_1 \\
a_{20}x_0 + a_{21}x_1 + a_{22}x_2 + \cdots + a_{2,N-1}x_{N-1} &= b_2 \\
\cdots \qquad\qquad &\quad \cdots \\
a_{M-1,0}x_0 + a_{M-1,1}x_1 + \cdots + a_{M-1,N-1}x_{N-1} &= b_{M-1}
\end{aligned}
$$

$$
\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & \ldots & a_{0,N-1} \\ a_{10} & a_{11} & \ldots & a_{1,N-1} \\ & \cdots & & \\ a_{M-1,0} & a_{M-1,1} & \ldots & a_{M-1,N-1} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \cdots \\ b_{M-1} \end{bmatrix}
$$

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

We start with: $N = M$

LU decomposition:

Suppose we are able to write the matrix $\mathbf{A}$ as a product of two matrices,

$$\mathbf{L} \cdot \mathbf{U} = \mathbf{A} \tag{2.3.1}$$

where $\mathbf{L}$ is *lower triangular* (has elements only on the diagonal and below) and $\mathbf{U}$ is *upper triangular* (has elements only on the diagonal and above). For the case of a 4×4 matrix $\mathbf{A}$, for example, equation (2.3.1) would look like this:

$$\begin{bmatrix} \alpha_{00} & 0 & 0 & 0 \\ \alpha_{10} & \alpha_{11} & 0 & 0 \\ \alpha_{20} & \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{30} & \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \cdot \begin{bmatrix} \beta_{00} & \beta_{01} & \beta_{02} & \beta_{03} \\ 0 & \beta_{11} & \beta_{12} & \beta_{13} \\ 0 & 0 & \beta_{22} & \beta_{23} \\ 0 & 0 & 0 & \beta_{33} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$\mathbf{L} \cdot \mathbf{y} = \mathbf{b}$$

$$\mathbf{A} \cdot \mathbf{x} = (\mathbf{L} \cdot \mathbf{U}) \cdot \mathbf{x} = \mathbf{L} \cdot (\mathbf{U} \cdot \mathbf{x}) = \mathbf{b}$$

$$\mathbf{U} \cdot \mathbf{x} = \mathbf{y}$$

Linear least squares problems:

N data points $(x_i, y_i), i = 0, \ldots, N-1$

Model: $$y(x) = \sum_{k=0}^{M-1} a_k X_k(x)$$

$X_0(x), \ldots, X_{M-1}(x)$ are arbitrary fixed functions of x, called the *basis functions*.

M adjustable parameters $a_j,\ j = 0, \ldots, M-1$

From statistics, we know that we must minimize:

$$\chi^2 = \sum_{i=0}^{N-1} \left[\frac{y_i - \sum_{k=0}^{M-1} a_k X_k(x_i)}{\sigma_i} \right]^2$$

σ_i is the measurement error (standard deviation) of the ith data point

First order conditions are

$$\frac{\partial \chi^2}{\partial a_k} = 0 \quad k = 0, \ldots, M-1$$

from which we get

$$0 = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[y_i - \sum_{j=0}^{M-1} a_j X_j(x_i) \right] X_k(x_i) \qquad k = 0, \ldots, M-1 \qquad (15.4.6)$$

$$0 = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[y_i - \sum_{j=0}^{M-1} a_j X_j(x_i) \right] X_k(x_i) \qquad k = 0, \ldots, M-1 \qquad (15.4.6)$$

Interchanging the order of summations, we can write (15.4.6) as the matrix equation

$$\sum_{j=0}^{M-1} \alpha_{kj} a_j = \beta_k \qquad (15.4.7)$$

where $\alpha_{kj} = \sum_{i=0}^{N-1} \frac{X_j(x_i) X_k(x_i)}{\sigma_i^2}$ and $\beta_k = \sum_{i=0}^{N-1} \frac{y_i X_k(x_i)}{\sigma_i^2}$

We can now formulate this with a ”design matrix” A and a right hand side b

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i} \qquad b_i = \frac{y_i}{\sigma_i} \qquad \mathbf{a} \equiv (a_0, \ldots, a_{M-1}) \qquad (\mathbf{A}^T \cdot \mathbf{A}) \cdot \mathbf{a} = \mathbf{A}^T \cdot \mathbf{b}$$

”Normal equations”

Design matrix A (and right hand side b):

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i}$$

$$b_i = \frac{y_i}{\sigma_i}$$

$$\begin{array}{ccc}
 & & \begin{array}{cccc} X_0(\ ) & X_1(\ ) & \cdots & X_{M-1}(\ ) \end{array} \\
\text{data points} & \begin{array}{c} x_0 \\ \\ x_1 \\ \\ \vdots \\ \\ x_{N-1} \end{array} & \begin{pmatrix}
\frac{X_0(x_0)}{\sigma_0} & \frac{X_1(x_0)}{\sigma_0} & \cdots & \frac{X_{M-1}(x_0)}{\sigma_0} \\
\frac{X_0(x_1)}{\sigma_1} & \frac{X_1(x_1)}{\sigma_1} & \cdots & \frac{X_{M-1}(x_1)}{\sigma_1} \\
\vdots & & & \vdots \\
\vdots & & & \vdots \\
\vdots & & & \vdots \\
\frac{X_0(x_{N-1})}{\sigma_{N-1}} & \frac{X_1(x_{N-1})}{\sigma_{N-1}} & \cdots & \frac{X_{M-1}(x_{N-1})}{\sigma_{N-1}}
\end{pmatrix}
\end{array}$$

Algorithm for Linear Least Square Problems (Normal Equations):

N data points $(x_i, y_i), i = 0, \ldots, N-1$

Model: $y(x) = \sum_{k=0}^{M-1} a_k X_k(x)$

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i} \qquad b_i = \frac{y_i}{\sigma_i} \qquad \mathbf{a} \equiv (a_0, \ldots, a_{M-1})$$

$$\begin{aligned} \mathbf{C} &\equiv \mathbf{A}^T \cdot \mathbf{A} \\ \mathbf{c} &\equiv \mathbf{A}^T \cdot \mathbf{b} \end{aligned}$$

Solve $\mathbf{C} \cdot \mathbf{a} = \mathbf{c}$

Notice, that the matrix $\mathbf{C}$ is symmetric and positive (semi)-definite (proof on the blackboard)

Cholesky decomposition (symmetric and pos. def. matrices):

Symmetric means that $a_{ij} = a_{ji}$ for $i, j = 0, \ldots, N-1$

positive-definite means that $\mathbf{v} \cdot \mathbf{A} \cdot \mathbf{v} > 0$ for all nonzero vectors $\mathbf{v}$

Similar to LU decomposition:

$$\mathbf{L} \cdot \mathbf{L}^T = \mathbf{A}$$

$$L_{ii} = \left(a_{ii} - \sum_{k=0}^{i-1} L_{ik}^2 \right)^{1/2}$$

$$L_{ji} = \frac{1}{L_{ii}} \left(a_{ij} - \sum_{k=0}^{i-1} L_{ik} L_{jk} \right) \qquad j = i+1, i+2, \ldots, N-1$$

Cholesky decomposition vs. LU decomposition:

- Cholesky decomposition can only be used for symmetric and positive definite matrices.

- Compared to LU decomposition, Cholesky decomposition is approximately twice as fast and requires only half the storage

- Cholesky decomposition is numerically more stable than LU decomposition (reason is subtle and beyond the scope of NR)