Converted from "**Gauss- Newton Nonlinear Regression**" created by **Faouizi Amar**

Global variables

THE GOODNESS OF FIT

$$S_{ry}(X,Y,Model) := \left(\left|Y - \overrightarrow{Model(X)}\right|\right)^2$$

$$S_{ty}(Y) := \left(|Y - \text{mean}(Y)|\right)^2$$

$$ry(X,Y,Model) := \sqrt{\frac{S_{ty}(Y) - S_{ry}(X,Y,Model)}{S_{ty}(Y)}}$$

GAUSS–NEWTON NONLINEAR REGRESSION

INTRODUCTION

Curve fitting is a technique used by engineers and scientists to aid them in the analysis and interpretation of experimental data. The data is typically a "large" collection of (x,y) points. The goal is to find a mathematical model function, ***f()***, that fits the ***trend*** and the ***level*** of the data. In Mathcad, we will write the desired model as ***f(x,A)***, where ***x*** is the independent variable and ***A*** is a vector containing the parameters of the model. For example, assume that we would like to fit a set of (x,y) points with a model of the form:

$$f(x) = a\left(1 - exp(-b \cdot x)\right)$$

Then, in Mathcad, we would write the model equation as:

$$f(x,A) := A_0 \cdot \left(1 - \exp\left(-A_1 \cdot x\right)\right)$$

When performing curve fitting, one must understand two things.First, the choice of the model to be used to fit the data is just that: a choice. The form of the mathematical function is typically based on the trend of the data, some physical interpretation of the phenomena involved in an experiment, experience, or some other source of input. Second, the unknowns of the problem are the parameters of the model –and not the independent variable(s). In other words, once a model, such as the one mentioned above, is picked, the goal becomes to find ***a*** and ***b*** in order to make the curve of ***f()*** go through the (x,y) points.

In previous lectures, we have set the theoretical foundation for ***curve fitting in the least squares sense*** (LS). For the cases where it is desired to fit the data with a polynomial of

degree n, we indicated that the solution can be written in a simple recursive matrix format. We have also studied many models that are amenable to a linear format through simple transformations and change of variables.

In many cases, it is not possible to transform the nonlinear model into a linear one. In other cases, the transformation is possible but results in the loss of sensitivity. Under such circumstances, direct nonlinear regression should be used. There are many algorithms that have been developed specifically for this purpose. One such a model is known as the Gauss–Newton algorithm. The theoretical formulation of the algorithm and its applicability can be found in many references [1-3] and will be discussed thoroughly in class.

FLOWCHART

The following flowchart provides the basic structure of the Gauss–Newton algorithm. Given a set of (x,y) data points and a model ***f(x,A).*** The aim is the find the vector of parameters, A, such that the mean square error between data and model is minimum. The steps involved in finding the solution using the Gauss–Newton algorithm are:

(1) Pick an initial guess for each parameter.

(2) Compute the matrix **Z**

(3) Compute the difference vector **D**

(4) Compute the perturbation in the parameters

$$\Delta A = \left[\left[((Z))^{\mathrm{T}} \cdot Z \right]^{-1} \cdot [((Z))]^{\mathrm{T}} \right] \cdot D$$

(5) Update the new parameters

$$A = A + \Delta A$$

(6) Keep updating until you meet your convergence criteria.

The Gauss–Newton algorithm was implemented using the new programming capability of Mathcad PLUS 6.0. The function ***GN(X, Y, F, A, Max)*** works as follows:

(1) The user provides several inputs:

(a) X and Y are two arrays containing the experimental data. The data need not be sorted.

(b) F is a vector function containing the model as its first element. Subsequent elements contain the derivatives of the model with respect to each of the parameters (the Mathcad symbolic processor may be used to find these derivatives).

(c) **a** is the vector containing the initial guesses of the parameters of the model.

(d) Max represent the maximum number of iterations that the user allows.

(2) Mathcad returns the last estimate of the parameters.

$$GN(X,Y,F,a,Max) := \left\| \begin{array}{l} \text{for } Iteration \in 0..Max \\ \quad \text{for } j \in 0..\text{last}(a) \\ \quad\quad \text{for } i \in 0..\text{last}(X) \\ \quad\quad\quad Z_{i,j} \leftarrow F\left(X_i, a\right)_{j+1} \\ \quad \text{for } i \in 0..\text{last}(X) \\ \quad\quad Y1_i \leftarrow F\left(X_i, a\right)_0 \\ \quad \Delta a \leftarrow \left((Z)^{\mathrm{T}} \cdot Z\right)^{-1} \cdot (Z)^{\mathrm{T}} \cdot (Y - Y1) \\ \quad b \leftarrow a \\ \quad a \leftarrow b + \Delta a \\ a \end{array} \right.$$

EXAMPLE

Suppose that the outcome of an experiment is as follows:

$$Experimental\_Data := \begin{bmatrix} 0.25 & 0.28 \\ 0.75 & 0.57 \\ 1.25 & 0.68 \\ 1.75 & 0.74 \\ 2.25 & 0.79 \end{bmatrix}$$

From experience, it is known that a suitable model for the above data has the form: **a*(1-exp(-b*x)).** Use the newly built Mathcad function GN() to regress the data. Pick an initial guess of 0.75 and 0.5 for ***a*** and ***b*** respectively. Compute the percent correlation coefficient after (a) 5 iterations, and (b) 10 iterations.

Solution

$$x := Experimental\_Data^{\langle 0 \rangle} \qquad y := Experimental\_Data^{\langle 1 \rangle}$$

$$f(x, a, b) := a \cdot (1 - \exp(-b \cdot x))$$

First, we need to give initial guesses to the unknown parameters ***a*** and ***b***. Note that these initial guesses must be chosen carefully chosen since they control to a great degree the success and rate of convergence of the Gauss–Newton algorithm.

$$A := \begin{bmatrix} 0.75 \\ 0.50 \end{bmatrix}$$

Second, we need to build a vector function **F(x,A)** that contains the model and its derivatives with respect to ***a*** and ***b***.

$$F(x, A) := \begin{bmatrix} A_0 \cdot \left(1 - \exp\left(-A_1 \cdot x\right)\right) \\ 1 - \exp\left(-A_1 \cdot x\right) \\ A_0 \cdot x \cdot \exp\left(-A_1 \cdot x\right) \end{bmatrix}$$

After three iteration: $\begin{bmatrix} a \\ b \end{bmatrix} := GN(x, y, F, A, 5)$

$a = 0.744$ $b = 1.202$

$Model(v) := f(v, a, b)$ $ry(x, y, Model) = 83.102\ 1\%$

After ten iterations: $\begin{bmatrix} a \\ b \end{bmatrix} := GN(x, y, F, A, 10)$

$a := 0.792$ $b := 1.67$

$Model(v) := f(v, a, b)$ $ry(x, y, Model) = 99.799\ 1\%$

Therefore, after five iterations, we still do not have convergence. After ten iterations, convergence is reached since increasing the number of iteration hardly changes the values of ***a*** and ***b***. The chosen model, therefore, accounts for 99.80% of the data.

$t := 0, 0.01 .. 2.5$ $i := 0 .. last(x)$

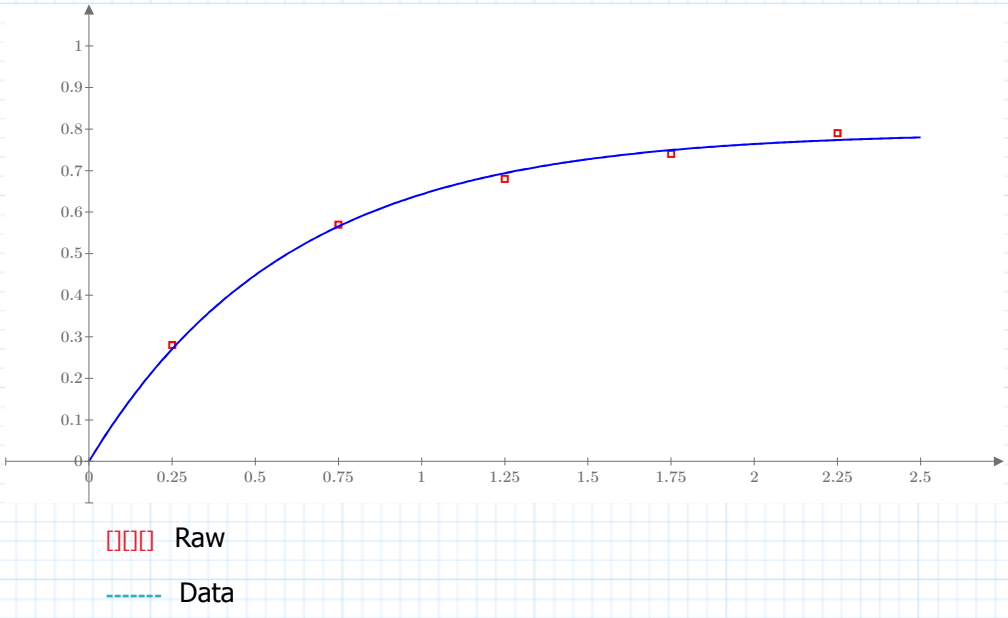

Figure 1: Final fit of the data using the Gauss-Newton algorithm

(1) Try changing the initial guess. For instance, pick a = 1 and b = 1. You will notice that convergence is reached after two or three iterations only. Other choices may make the algorithm fail (pick for instance a = 1 and b = 0.5).

(2) For a fixed initial guess, use GN() to plot ***a***, ***b***, and the correlation coefficient as a function of the number of iterations.

$$S_{ry}(X,Y,Model) := \left(\left|Y - \overrightarrow{Model(X)}\right|\right)^2$$ Model is any model at hand.

$$S_{ty}(Y) := \left(|Y - \text{mean}(Y)|\right)^2$$

$$ry(X,Y,Model) := \sqrt{\frac{S_{ty}(Y) - S_{ry}(X,Y,Model)}{S_{ty}(Y)}}$$

Faouzi Amar
amar@uta.edu
The Wave Scattering Research Center
University of Texas at Arlington
Electrical Engineering Department
PO Box 19016
Arlington, Texas 76019