Non-Linear Model Fitting

Project2 in Numerical Methods

Vasily Arzhanov, Reactor Physics, KTH

Abstract

Any more or less serious computational research involves a good portion of analytical work. This numerical project guides the students how to successfully combine analytical thinking, numerical calculations, and programming skill. Upon finishing this project, the students will acquire a better insight into an important area of numerical analysis – nonlinear least squares fitting. Mathematically, it may be classified as an unconstrained optimization problem. Nonlinear least-squares fitting (method, analysis, regression) generalizes linear least-squares methodology in the sense it is based on a much broader and more general class of functions.

The current project suggests that the student applies nonlinear least-squares for analysing a real accidental event occurred in Oscarshamn-2 at a boiling water reactor in 1999. While performing this project, the student is supposed to answer the questions marked with **bold face** and submit a written report organized as a scientific manuscript. The layout of the present guidelines may serve as a template for writing the final report.

Key words: Nonlinear least squares, fitting, regression, optimization.

Introduction

Boiling Water Reactors (BWRs) are prone to instability due to the nature of the two-phase thermal-hydraulics in the reactor core facilitated further by the power feedback through core neutron kinetics, plant system thermal-hydraulics and the plant control system. In general, BWR may be subject to coupled thermal-hydraulic – neutronic instabilities while operating at a relatively low flow rate (relative to power e.g., 30% of nominal flow and 50% of nominal power). The instability originates in the core and leads to oscillations in flow rate, pressure, and void fraction.

An example of an instability event is the transient that occurred in Oskarshamn-2 in 1999. A loss of feedwater preheaters and control system failure resulted in a situation with a high feedwater flow and low feedwater temperature without reactor scram. In addition to initiating event, interaction of automatic power and flow control system caused plant to move into low flow – high power regime. Combination of the above mentioned events culminated in diverging power oscillations which triggered automatic scram at high power. The power evolution for the event is shown in Figure 1.

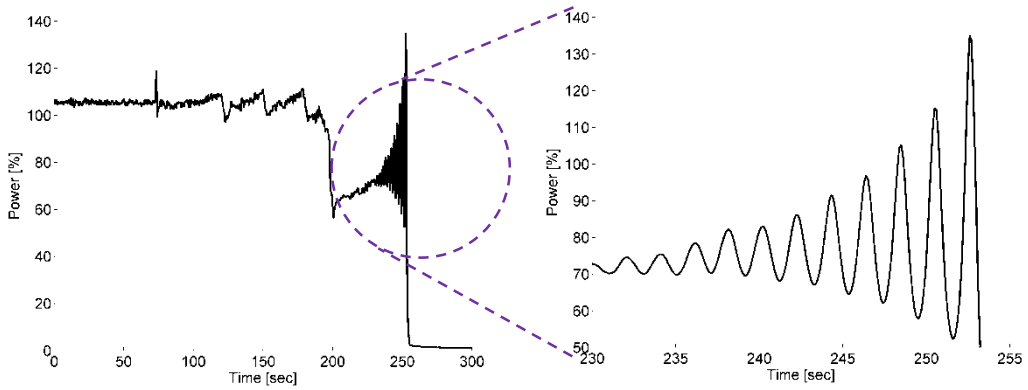

Figure 1. Oskarshamn-2 stability event in 1999.

The overall objective of the current project is to perform a regression analysis of this reach-feature process i.e., to find a simpler (regressed, less developed) time representation. More in detail, the purpose is to develop a few-parameter mathematical model of this oscillatory event and find "optimal" parameter values that best fit to the experimental data in some predefined sense. Once the parameters are found, it is very easy to evaluate the so-called Decay Ratio, DR, which may serve as an indicator of how dangerous the current situation is.

The least-squares method, whether linear or nonlinear, is probably the best mathematical framework to formulate an optimisation problem because of its exceptional clarity. It is natural and pedagogical to first define conceptual ideas in the least-squares method then the project gently introduces students to the linear least-squares topic. Finally, students learn basic ideas in the nonlinear least-squares method and develop a computer code, preferably in Python or Matlab, that numerically solves the suggested problem. The nonlinear problem is typically solved by iterative refinement; at each iterative step, the system is approximated by linear least-squares equations thus making the core calculations very similar in both cases.

1. Regression Analysis

Regression analysis is a branch of mathematical statistics (or more specifically, statistical modelling). It is represented by a (big) number of statistical methods for determining the underlying relationships between a dependent variable (also called outcome, response etc.) and one or several independent variables (predictors, covariates, features etc.).

We focus ourselves on a typical and probably the simplest situation of studying the relationship between an independent variable, say x, and a dependent variable, $y = y(x)$. Suppose we observe or measure N data pairs

$$(x_i, y_i),\ i = 1, 2, \ldots, N$$

It should be stressed here, the symbol x (and y) has a multiple meaning depending on the context. It may denote an independent variable x or a sample of the given values such as in Appendix V.

$$x \equiv \{x_1, x_2, \ldots, x_N\}$$

Examples of data pairs are: (voltage, current), (time, neutron density), (mass number, fission yield). Typically, a researcher selects a functional dependency (model) and estimates how good it fits the given/observed data set. As a rule, the chosen model involves one or several unknown parameters. Let the parameters be represented by a vector (note bold face)

$$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_M)$$

In other words, we assume the model is characterized by $(M+1)$ parameters. The difference between the number of data points and the number of parameters (in our case)

$$\mathrm{DoF} \equiv N - (M+1)$$

often appears in regression analysis and is referred to as the **degrees of freedom**. The functional dependency may be written symbolically as

$$y(x) = f(x; \boldsymbol{\alpha})$$

f = model, α = param, x = input, y = output

For instance, a polynomial model assumes the functional dependency to be a polynomial of some degree M

$$f(x; \boldsymbol{\alpha}) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_M x^M$$

The simplest polynomial model arises when we fix $M = 0$; it is known as the **baseline model**, which always predicts the mean value $(\alpha_0 = \bar{y})$

1 parameter

$$y = f(x; \boldsymbol{\alpha}) = \bar{y} \qquad \bar{y} \equiv (y_1 + y_2 + \ldots + y_N)/N$$

A particular choice of the model may stem from:

- Physical model such as Ohm's law;
- Numerical simulation such as solving numerically Neutron Diffusion Equation, NDE;
- Previous experience e.g., the current conditions are similar to the previous ones;
- Educated guess;
- Visual inspection;
- Assumption subject to confirm or reject.

Once the parameters, **α**, are found, for example by the least-squares method, we can then use the fitted values often denoted with the hat sign

$$\hat{y}_i = f(x_i; \boldsymbol{\alpha})$$

to assess the accuracy of the model in predicting the data. To this end, we define individual residuals and the total residual (also known as the Sum of Squared Residuals, SSR) by

individual residual

$$r_i \equiv \hat{y}_i - y_i \quad \text{and} \quad \mathrm{SSR} \equiv R^2 \equiv \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} \left[\hat{y}_i - y_i\right]^2 = \sum_{i=1}^{N} \left[f(x_i; \boldsymbol{\alpha}) - y_i\right]^2$$

The total residual strongly depends on the number of observations. Because of this, a somewhat better representation of the prediction error is given by the Root Mean Square Error, RMSE, which gives the error in units of the variable y

$$\mathrm{RMSE} \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\hat{y}_i - y_i\right]^2}$$

RMS error

It is always advantageous to use and work with dimensionless/normalized quantities. One way of normalizing RMSE is to divide it by the sample standard deviation defined as

standard deviation of the samples

$$S_y \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2}$$

Thus, we introduce the Root Mean Squares Error Normalized, RMSN, by

RMSN

$$\mathrm{RMSN} \equiv \mathrm{RMSE}/S_y$$

The coefficient of determination, CoD, is also a popular choice to characterize the quality of the regression model in consideration. It is denoted as R^2 or r^2 and pronounced as "R squared" (do not confuse with total residual, R).

$$\mathrm{CoD} \equiv R^2 \equiv 1 - \mathrm{RMSN}^2 = 1 - \frac{\sum_{i=1}^{N}\left[\hat{y}_i - y_i\right]^2}{\sum_{i=1}^{N}\left[\bar{y} - y_i\right]^2}$$

COD

In the best case, the modelled values exactly match the observed values, $\hat{y}_i = y_i$, which results in RMSE = 0 and $R^2 = 1$. A baseline model, which always predicts $\bar{y}$, will have RMSE = S_y and $R^2 = 0$. The model that gives even a worse prediction than that of the baseline will have a negative value of R^2.

2. Linear-Least Squares Method

Least-squares problems fall into two categories: linear or ordinary least-squares setting (LLS/OLS), and nonlinear least-squares (NLS), depending on whether or not the residuals are linear in all unknowns. The linear least-squares problems often occur in statistical regression analysis; they have closed-form solutions. The nonlinear problems are usually solved by iterative refinement i.e., at each iterative step, the system of equations is approximated by a linear problem, and thus the core calculation is similar in both cases.

Linear regression analysis begins with assuming a model function, for instance a polynomial of degree one ($M = 1$) as in the (simple) example below

$$y = f(x;\alpha_0,\alpha_1) = \alpha_0 + \alpha_1 x \tag{1}$$

Here, the parameters of the model function are represented by the vector (α_0,α_1).

The least-squares method stems from a very simple and intuitively clear idea to find the parameter values, (α_0,α_1), that "best" fit the observation in question, (x_i,y_i). The goodness of fit of the model in a particular data point is measured by its residual

$$r_i(\alpha_0,\alpha_1) \equiv f(x_i;\alpha_0,\alpha_1) - y_i = \alpha_0 + \alpha_1 x_i - y_i$$

The optimal parameters are found by minimizing the square of the total residual defined as

$$R^2(\alpha_0,\alpha_1) \equiv \sum_{i=1}^{N} r_i^2(\alpha_0,\alpha_1) \tag{2}$$

The situation is illustrated in Figure 2.

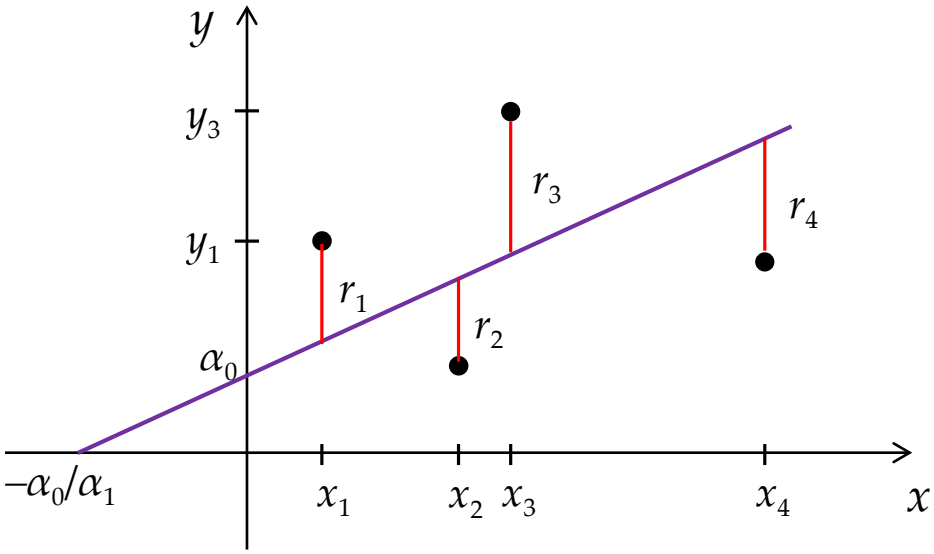

Figure 2. Individual residuals.

Assignment 1

Read Appendix I about Cramer's rule. Find the optimal parameters (α_0, α_1) by minimizing the square of the total residual $R^2(\alpha_0, \alpha_1)$. Write down your system of two linear equations with two unknowns α_1 then α_0 in the matrix form then use the Cramer rule to find first α_1 and then α_0 through α_1.

Assignment 2

It is always useful and fruitful to change the standpoint. To this end, read attentively Appendix V (Sample Statistics) then using a linear model, $f(x;\alpha_0,\alpha_1)=\alpha_0+\alpha_1 x$, do the following:

a) Write down an overdetermined system of algebraic linear equations, $f(x_i;\alpha_0,\alpha_1)=y_i$, in the matrix form, $\mathbf{A}\boldsymbol{\alpha}=\mathbf{b}$, where the unknown vector is $\boldsymbol{\alpha}=\left(\alpha_0,\alpha_1\right)^{\mathrm{T}}$;

b) Show that the normal equation, $\mathbf{A}^{\mathrm{T}}\mathbf{A}\boldsymbol{\alpha}=\mathbf{A}^{\mathrm{T}}\mathbf{b}$, is equivalent to the matrix equation found explicitly in Assignment 1;

c) Divide each line in the normal equations, $\mathbf{A}^{\mathrm{T}}\mathbf{A}\boldsymbol{\alpha}=\mathbf{A}^{\mathrm{T}}\mathbf{b}$, by N and rewrite them in terms of the mean values, $\overline{x}$, $\overline{y}$, $\overline{x^2}$, $\overline{xy}$;

d) Find α_1 using Cramer's rule and express it in terms of S_x, S_y and $\rho(x,y)$;

e) Find α_0 using α_1 and the first line in the normal equations.

Assignment 3

Thermal expansion is a very common property of matter to change its shape (length, area volume and density) in response to a change in temperature, T. The material's coefficient of linear thermal expansion, α_{L}, abbreviated as CLTE or simply CTE, characterises the relative expansion, $\Delta L/L$, per unit temperature change. The formal definition reads as

$$\alpha_{\mathrm{L}}(T) = \frac{1}{L}\frac{\mathrm{d}L}{\mathrm{d}T}$$

In practice, we start with measuring the initial length, L_0, at temperature T_0, and then we proceed to measure L_1 at T_1. When the fractional change in length is small, $(L_1 - L_0)/L_0 \ll 1$, we finally approximate

$$\alpha_{\mathrm{L}}(T_0) \approx \frac{1}{L_0}\frac{L_1 - L_0}{T_1 - T_0}$$

Experimental data were collected in file `CTE_Steel.txt` when measuring the coefficient of linear thermal expansion of steel at various temperatures. Write a computer code that reads the experimental data. Then do the following;

a) Find and report the coefficients of the regression line, $f(T;\alpha_0,\alpha_1) \equiv \alpha_0 + \alpha_1 T$;
b) In the same window, plot the regression line together with the experimental data;
c) Determine how good the regression line predicts the CTE coefficient by evaluating the (Pearson) correlation coefficient, $\rho(x,y)$, RMSE, RMSN and CoD.

Assignment 4

Often, linear regression analysis can be applied to nonlinear problems. To this end, consider neutron motion in water, which is of paramount importance in nuclear reactor design. Reactor physics predicts that the density of fast neutrons in water from a point neutron source behaves with distance r from the source as

$$n(r) = \frac{A}{r^2}e^{-r/\lambda}$$

Here λ is the relaxation length subject for determination. Students measure neutron counts per second from a point-like neutron source in water, which is known to be proportional to the neutron density in question. The experimental results are accumulated in a text file, which accompanies the current project. The purpose is to fit the results to the given functional dependency. Clearly, the problem is nonlinear. However, with a sharp eye, one notices that

$$\ln\left[r^2 n(r)\right] = \ln A - r/\lambda$$

It becomes clear now that a quantity, $y \equiv \ln\left[r^2 n(r)\right]$, linearly depends on distance r. Write a computer code that reads the experimental data in `Relaxation.txt` and do the following.

a) Calculate the quantity, y, and plot the scattered data, y versus r.
b) By visual inspection, find outliers i.e., the measurements that do not fit to a linear dependency and exclude them from consideration.
c) Find and report the coefficients of the regression line, $y = f(r;\alpha_0,\alpha_1)$.
d) Report the relaxation length.
e) In the same window, plot the regression line together with the filtered data.
f) Determine how good the regression line represents the quantity y by evaluating the (Pearson) correlation coefficient, $\rho(x,y)$, RMSE, RMSN and CoD.

3. Non-Linear Least Squares Method

Nonlinear least-squares method (nonlinear regression analysis) is very similar to the linear one. Assume, we study a relationship between two variables x and y (time and power; investments and sales etc.). The variable x will be treated as independent whereas the variable y will be regarded as dependent. In a simple case, the starting point in nonlinear least-squares method is a measurement (experiment, observation, simulation) which results in N data pairs

$$\left(x_i, y_i\right),\ i = 1, 2, \ldots, N\,.$$

Here, the x_i and y_i values are instances of the x variable and y variable respectively.

A researcher selects (deduces, assumes, conjectures) a mathematical model (functional dependency) that relates the x and y variables. The model typically involves one or several initially unknown parameters represented by the vector (note bold face)

$$\boldsymbol{\alpha} = \left(\alpha_1, \alpha_2, \ldots, \alpha_M\right)$$

As contrast to Section 1 "Regression Analysis", it is now more convenient to enumerate the model parameters from 1 to M. The reason for doing so will become clear very soon. Thus, the relation between x and y is written symbolically in the functional form

$$y = f(x; \boldsymbol{\alpha})$$

When the number of unknown parameters, M, is the same as the number of measurements, N, we may hope to find a unique set of parameters, $\boldsymbol{\alpha}$, that exactly fits the given data

$$y_i = f(x_i; \boldsymbol{\alpha}) \qquad i = 1, 2, \ldots, N$$

A typical problem is to build a unique polynomial that passes through the given points. A much more common and general situation is when we have few parameters and a big number of observations, $M \ll N$. Because of inevitable experimental errors, we only expect approximate equalities

$$y_i \approx f(x_i; \boldsymbol{\alpha}) \qquad i = 1, 2, \ldots, N\,.$$

Then a question arises to find parameters $\boldsymbol{\alpha}$ that best fit the given data. To this end, we define individual residuals (difference between the fitted and given values)

$$r_i(\boldsymbol{\alpha}) \equiv r_i \equiv f(x_i; \boldsymbol{\alpha}) - y_i$$

Next, we define the total residual (also known as the Sum of Squared Residuals, SSR) by

$$R^2(\boldsymbol{\alpha}) \equiv R^2 \equiv \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} \left[f(x_i; \boldsymbol{\alpha}) - y_i \right]^2$$

It gives a solid mathematical foundation for stating an unconstrained minimisation problem

$$\min_{\boldsymbol{\alpha}} R^2(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \left[f(x_i; \boldsymbol{\alpha}) - y_i \right]^2$$

The answer(s) to this problem may be found among solutions of a (generally nonlinear) system of algebraic equations

$$\begin{cases} \dfrac{\partial R^2(\boldsymbol{\alpha})}{\partial \alpha_m} = 0 \qquad m = 1, 2, \ldots, M \end{cases} \tag{3}$$

Performing differentiation allows us to rewrite this system of equation more in detail as

$$\begin{cases} \displaystyle\sum_{i=1}^{N} 2 f_i(\boldsymbol{\alpha}) \frac{\partial f_i(\boldsymbol{\alpha})}{\partial \alpha_m} = \sum_{i=1}^{N} 2 \left[f(x_i; \boldsymbol{\alpha}) - y_i \right] \frac{\partial f(x_i; \boldsymbol{\alpha})}{\partial \alpha_m} = 0 \qquad m = 1, 2, \ldots, M \end{cases}$$

It is a system of (typically highly) nonlinear equations and can be solved analytically only in very special situations. Sometimes a nonlinear problem can be reduced to a linear one, for example, by change of variables such as, $y \equiv \ln\left[r^2 n(r) \right]$ or similar, as it was

demonstrated in Assignment 4. In the vast majority of cases, such systems are solved numerically by iterations. One of the most efficient iterative algorithms for solving nonlinear equations is the Newton method also known as the Newton-Raphson method. It will be applied to the original (overdetermined) system of equations

$$\{f_i(\alpha_1,\ldots,\alpha_M)=0 \quad i=1,\ldots,N\gg M\,. \tag{4}$$

4. Newton's Method in 1-D

We start with one nonlinear equation in one dimension. For the sake of simplicity, we return back to the traditional notation for the unknown x as contrast to α as it was used previously. Thus, the equation in question is written symbolically as

$$f(x)=0$$

If the variable x is fixed at some value and $f(x)\neq 0$, we try to find Δx such that

$$f(x+\Delta x)=0$$

This equation is no simpler than the original one but if the displacement Δx is small, we can use an approximate (Taylor) expansion (linearisation of function f at x)

$$f(x+\Delta x)\approx f(x)+f'(x)\Delta x=0\longrightarrow \Delta x=-\frac{f(x)}{f'(x)}$$

It inspires us to define an iterative algorithm. Starting with an initial guess $x^{(0)}$, we generate a sequence, $x^{(k)}$, presumably converging to the root of the original equation.

$$x^{(k+1)}=x^{(k)}-\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)} \qquad k=0,1,2,\ldots \tag{5}$$

A corresponding theorem states that Newton's method generates a fast-converging sequence for most non-pathological functions provided that the initial guess, x_0, is sufficiently close to the root. Equivalently, algorithm (5) may be written as

$$\Delta x^{(k)}=-\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)} \qquad x^{(k+1)}=x^{(k)}+\Delta x^{(k)} \qquad k=0,1,2\ldots \tag{6}$$

We iterate until the moment the relative displacement becomes less than a predefined tolerance

$$\left|\Delta x^{(k)}\right|\Big/\left|x^{(k+1)}\right|\leq\varepsilon$$

5. Newton-Raphson Method in 2-D

Next, we turn to two-dimensional space also known as bi-dimensional. Two coordinates, for example (x_1,x_2), are needed to specify an element (point, vector) in the space. Often, we think about such elements as column vectors, for instance, (note bold face)

$$\mathbf{x}\equiv\left[x_1,x_2\right]^{\mathrm{T}}=\begin{bmatrix}x_1\\x_2\end{bmatrix} \text{ and } \Delta\mathbf{x}\equiv\begin{bmatrix}\Delta x_1\\\Delta x_2\end{bmatrix}$$

Many mathematical models involve functions of two (or more variables), $f(x_1,x_2)$, equivalently written as $f(\mathbf{x})$. It is convenient to use the compact (and modern) notation for partial derivatives

$$\partial_1 f \equiv \partial_1 f(x_1, x_2) \equiv \frac{\partial f(x_1, x_2)}{\partial x_1} \qquad \partial_2 f \equiv \partial_2 f(x_1, x_2) \equiv \frac{\partial f(x_1, x_2)}{\partial x_2}$$

The Taylor expansion reads now as

$$f(x_1 + \Delta x_1, x_2 + \Delta x_2) \approx f(x_1, x_2) + \partial_1 f(x_1, x_2) \cdot \Delta x_1 + \partial_2 f(x_1, x_2) \cdot \Delta x_2$$

Introducing the gradient operator as a row vector, $\nabla = [\partial_1, \partial_2\,]$, the linear Taylor expansion i.e., the linearisation of function *f* about point **x** can be written as (note bold face)

$$\nabla f \equiv \nabla f(\mathbf{x}) \equiv \left[\partial_1 f(\mathbf{x}),\, \partial_2 f(\mathbf{x})\right] \longrightarrow f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \Delta\mathbf{x}$$

Finally, we define vector-valued functions by

$$\mathbf{f}(\mathbf{x}) \equiv \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix}$$

Armed with these concepts, we try now to solve numerically a system of (nonlinear) algebraic equations in two dimensions.

$$\begin{cases} f_1(x_1, x_2) = 0 \\ f_2(x_1, x_2) = 0 \end{cases} \longleftrightarrow \mathbf{f}(\mathbf{x}) = \mathbf{0}$$

Again, if we start with an (arbitrary) fixed point (x_1, x_2), it is highly unlikely that we exactly guess the solution. So, we assume $f_1(x_1, x_2) \neq 0$ and $f_2(x_1, x_2) \neq 0$. Next, we try to find displacements, Δx_1 and Δx_2, such that

$$\begin{cases} f_1(x_1 + \Delta x_1, x_2 + \Delta x_2) = 0 \\ f_2(x_1 + \Delta x_1, x_2 + \Delta x_2) = 0 \end{cases}$$

Unfortunately, we cannot solve it exactly, but the linear Taylor expansion gives an idea

$$\begin{cases} f_1(x_1 + \Delta x_1, x_2 + \Delta x_2) \approx f_1(x_1, x_2) + \partial_1 f_1(x_1, x_2)\Delta x_1 + \partial_2 f_1(x_1, x_2)\Delta x_2 = 0 \\ f_2(x_1 + \Delta x_1, x_2 + \Delta x_2) \approx f_2(x_1, x_2) + \partial_1 f_2(x_1, x_2)\Delta x_1 + \partial_2 f_2(x_1, x_2)\Delta x_2 = 0 \end{cases}$$

to find a pair of displacement $(\Delta x_1, \Delta x_2)$ that obeys the (linear) system of equations, (note that x_1 and x_2 are fixed here):

$$\begin{bmatrix} \partial_1 f_1(x_1, x_2) & \partial_2 f_1(x_1, x_2) \\ \partial_1 f_2(x_1, x_2) & \partial_2 f_2(x_1, x_2) \end{bmatrix} \cdot \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = -\begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix}$$

The matrix involved here is known as the Jacobian matrix; it generalizes the ordinary derivative for the multidimensional case

$$\mathbf{J} \equiv \mathbf{J}(\mathbf{x}) \equiv \mathbf{J}(x_1, x_2) \equiv \begin{bmatrix} \partial_1 f_1(x_1, x_2) & \partial_2 f_1(x_1, x_2) \\ \partial_1 f_2(x_1, x_2) & \partial_2 f_2(x_1, x_2) \end{bmatrix}$$

In vector-matrix notation, the displacement equation is equivalently rewritten as

$$\mathbf{J}\Delta\mathbf{x} = -\mathbf{f}$$

We conclude that the new values, $x_1 + \Delta x_1$ and $x_2 + \Delta x_2$, are expected to be closer to the exact solution. A corresponding theorem proves this observation provided the starting point $\mathbf{x}^{(0)}$ is sufficiently close the solution (root) in question. The above considerations lead to an iterative procedure starting with an arbitrary guess $\mathbf{x}^{(0)}$:

$$\mathbf{J}\left(\mathbf{x}^{(k)}\right)\Delta\mathbf{x}^{(k)} = -\mathbf{f}\left(\mathbf{x}^{(k)}\right) \qquad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \qquad k = 0, 1, 2\ldots \tag{7}$$

This iterative scheme naturally generalizes (scalar) Newton's algorithm. It is known also as the Newton-Raphson method. The iterations run until the relative displacement becomes less than a predefined tolerance

$$\left\|\Delta\mathbf{x}^{(k)}\right\| / \left\|\mathbf{x}^{(k+1)}\right\| \tag{8}$$

Here, $\|\mathbf{x}\|$ is any vector norm, for example (Euclidean) 2-norm.

Assignment 5

Consider a (simple) system of nonlinear equations

$$\begin{cases} y = x^2 - 2 \\ x = y^2 - 2 \end{cases}$$

a) Plot the two curves determined by each equation in the region, $-2.5 \le x \le 2.5$, and $-2.5 \le y \le 2.5$ and make the axes be equal each other.

b) By visual inspection, determine the number of solutions; make a conjecture about two exact solutions and show that your guess is correct.

c) Write a computer code that implements the Newton-Raphson method in two dimensions; check that your code generates a sequence converging to one of the exact solutions if you start sufficiently close to the root in question.

d) Find numerically two more solutions and report them.

e) Start your code at the initial guess, $x_0 = 0.5$ and $y_0 = 0.5$, and explain the outcome.

Not compulsory (for fanatics):

f) Find all points that fail your code; and g) Find analytically all the solutions.

6. Newton-Raphson Method in Multidimensional Space

A system of nonlinear equations is generally written as

$$\begin{cases} f_1(x_1, x_2, \ldots, x_M) = 0 \\ f_2(x_1, x_2, \ldots, x_M) = 0 \\ \vdots \\ f_M(x_1, x_2, \ldots, x_M) = 0 \end{cases} \tag{9}$$

The Newton-Raphson method easily generalizes for multidimensional space if we define a multidimensional variable, **x**, and a vector-valued function, **f**, as

$$\mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \qquad \mathbf{f}(\mathbf{x}) \equiv \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_M(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_M) \\ f_2(x_1, x_2, \ldots, x_M) \\ \vdots \\ f_M(x_1, x_2, \ldots, x_M) \end{bmatrix}$$

System (9) is then compactly written as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

The Jacobian matrix is now $M \times M$ and reads as

$$\mathbf{J}(\mathbf{x}) \equiv \begin{bmatrix} \partial_1 f_1(\mathbf{x}) & \partial_2 f_1(\mathbf{x}) & \cdots & \partial_M f_1(\mathbf{x}) \\ \partial_1 f_2(\mathbf{x}) & \partial_2 f_2(\mathbf{x}) & \cdots & \partial_M f_2(\mathbf{x}) \\ \vdots & \vdots & \cdots & \vdots \\ \partial_1 f_M(\mathbf{x}) & \partial_2 f_M(\mathbf{x}) & \cdots & \partial_M f_M(\mathbf{x}) \end{bmatrix}$$

It is easy to show the linear Taylor expansion may be written as

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x}$$

Equating the above expansion to zero leads to the equation

$$\mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x} = \mathbf{0} \longrightarrow \mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})$$

It suggests the following iterative algorithm that starts with an arbitrary guess $\mathbf{x}^{(0)}$

$$\mathbf{J}\left(\mathbf{x}^{(k)}\right)\Delta\mathbf{x}^{(k)} = -\mathbf{f}\left(\mathbf{x}^{(k)}\right) \qquad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \qquad k = 0,1,2\ldots$$

This algorithm is essentially the same as in (7).

7. Newton-Raphson Method for Overdetermined Systems

This time, we have more equations than unknowns such as our original problem (4), namely

$$\begin{cases} f_1(x_1, x_2, \ldots, x_M) = 0 \\ f_2(x_1, x_2, \ldots, x_M) = 0 \\ \vdots \\ f_N(x_1, x_2, \ldots, x_M) = 0 \end{cases} \tag{10}$$

One way of attacking this challenge is to convert this problem to minimising the total residual as it was done in (3). Intuitively, it is easier to apply the Newton-Raphson technique directly. We begin with rewriting (10) in a compact (vector) form (note bold face)

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

Here, the vector of unknowns, **x**, and a vector-valued function, **f**, are defined by

$$\mathbf{x} \equiv \left[x_1, x_2, \ldots, x_M\right]^{\mathrm{T}} \qquad \mathbf{f} \equiv \left[f_1, f_2, \ldots, f_N\right]^{\mathrm{T}}$$

Clearly, the Jacobian matrix is rectangular now, $N \times M$, $(N > M)$

$$\mathbf{J}(\mathbf{x}) \equiv \begin{bmatrix} \partial_1 f_1(\mathbf{x}) & \partial_2 f_1(\mathbf{x}) & \cdots & \partial_M f_1(\mathbf{x}) \\ \partial_1 f_2(\mathbf{x}) & \partial_2 f_2(\mathbf{x}) & \cdots & \partial_M f_2(\mathbf{x}) \\ \vdots & \vdots & \cdots & \vdots \\ \partial_1 f_N(\mathbf{x}) & \partial_2 f_N(\mathbf{x}) & \cdots & \partial_M f_N(\mathbf{x}) \end{bmatrix}$$

We search a vector $\Delta\mathbf{x}$ such that $\mathbf{x} + \Delta\mathbf{x}$ is closer to the solution than **x**. To this end, we equate the linear Taylor expansion to zero

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x} = \mathbf{0} \longrightarrow \mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x} = -\mathbf{f}(\mathbf{x}) \tag{11}$$

System (11) is linear because **x** is known at this moment and $\Delta\mathbf{x}$ is unknown. However, system (11) is overdetermined so we solve it in the sense of linear least-squares. One way of solving the overdetermined system (11) is to convert it first to the normal equations

$$\mathbf{J}(\mathbf{x})^{\mathrm{T}}\mathbf{J}(\mathbf{x}) \cdot \Delta\mathbf{x} = -\mathbf{J}(\mathbf{x})^{\mathrm{T}}\mathbf{f}(\mathbf{x}) \tag{12}$$

However, one can do it directly in Matlab (as well as in Python) using built-in tools, such as

```
x = A\f
```

Here, the back-slash operator \ solves the system of linear algebraic equations, $\mathbf{A}\mathbf{x} = \mathbf{f}$, in: (a) ordinary sense if **A** is a square matrix; (b) least-squares sense if **A** represents an overdetermined system; and (c) in the sense of basic solution if **A** represents an under-determined system.

These considerations give rise to the following iterative scheme. As any other iterative algorithms, it starts with an arbitrary (educated) guess, $\mathbf{x}^{(0)}$, and then generates a sequence of supposedly converging approximations.

$$\mathbf{J}\left(\mathbf{x}^{(k)}\right) \cdot \Delta\mathbf{x}^{(k)} = -\mathbf{f}\left(\mathbf{x}^{(k)}\right) \qquad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \qquad k = 0,1,2\ldots \tag{13}$$

It is worth stressing again, Eq.(13) is solved in the least-squares sense.

8. Decay ratio

A great variety of physical processes may be modelled by a simple harmonic motion, or more generally, by a simple harmonic oscillator, which is determined by three parameters, namely the amplitude, a, the angular (circular) frequency, ω, and the phase, ϕ. If y is the physical quantity in question (displacement, reactor power, price), the mathematical model is given by

$$y(t) = a\cos\left(\omega t + \phi\right)$$

Figure 3 visually illustrates a harmonic oscillator with amplitude, a, and period, T.

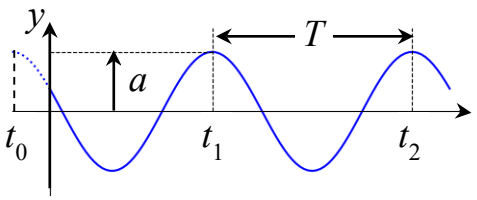

***Figure* 3.** Simple harmonic oscillator.

Maxima occur at times $t_0 = -\phi/\omega$, $t_1 = \left(2\pi - \phi\right)/\omega$, $t_2 = \left(4\pi - \phi\right)/\omega$ etc. The period, T, and thus the frequency, f, are found as

$$T = t_2 - t_1 = 2\pi/\omega; \quad f = 1/T = \omega/2\pi.$$

The vast majority of the oscillatory systems suffer from some sort of irreversible energy loss. Therefore we expect such oscillations to eventually damp away. Many oscillatory systems with energy loss are well approximated by the following mathematical model.

$$Y(t) = ae^{-\gamma t}\cos\left(\omega t + \phi\right) + m$$

In addition to the simple harmonic oscillator, two more parameters are involved here. They are the decay constant, γ, and the mean value, m, giving totally five parameters. Typically, it is very easy to evaluate the mean value. So in what follows, we will analyse a shifted signal

$$y(t) = Y(t) - m = ae^{-\gamma t}\cos\left(\omega t + \phi\right) \tag{14}$$

It is now determined by four parameters, an appreciable reduction in complexity. *Figure* 4 gives a visual picture of decaying oscillations with zero mean value.

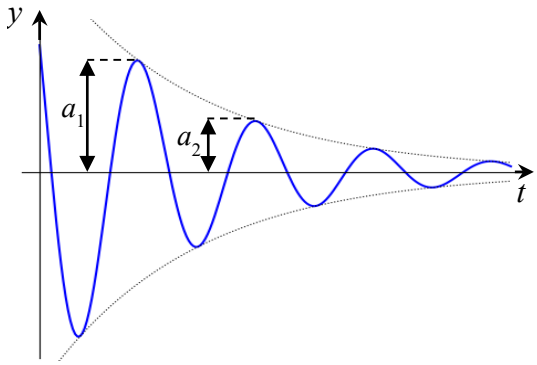

***Figure* 4.** Decaying oscillator.

Any decaying harmonic oscillator may be characterised by the so-called decay ratio that is defined as a ratio between two successive maxima

$$\mathrm{DR} \equiv a_2/a_1 \tag{15}$$

The importance of this concept stems from the fact that when DR < 1, the oscillations die out. On contrary, the oscillations diverge when DR > 1. In practice, an oscillatory physical process with the decay ratio greater than one may end up in an accident. That's why the decay ratio may serve as an emergency indicator.

Assignment 6

Let a physical process obey Eq.(14), do the following:

(a) Find the decay ratio;

(b) If t_{m} is a time moment when a local maximum in Eq. (14) occurs, find the phase ϕ in terms of t_{m} and the period T.

9. Finding Decay Ratio

File `PowerOscillation.txt` accompanies the current project. The data is organised in two columns. The first column gives sampling times in seconds and the second column shows the measured reactor thermal power as percentage of the nominal power. This time series of power level values is also referred to as a signal. A minimal characterisation of any signal (sample, selection, discrete data) is its mean value and standard deviation. It is always advantageous to do some pre-processing on the given data. It may include:

- Setting local time;
- Subtracting the mean value (centring data);
- Removing the linear (or higher) trend (it also removes the mean value);
- Scaling the data for example by the standard deviation, std.

Assignment 7

It is not a bad idea to start data analysis (especially analysis of time series) with simply representing the given information visually. To this end, write a computer code that reads sampling times and corresponding power values. Prepare two subplots (one over another).

a) In the first sub-window, plot the data as it is, i.e. power as a function of time together with the mean value line; report also the mean value and the standard deviation.

b) Remove a possible linear trend (standard in Matlab/Python) from the given signal.

c) Recalculate the standard deviation and report this value; scale the signal by std.

d) In the second sub-window, plot the pre-processed data together with the t-axis.

Next, we turn to implementing algorithm (13). We begin with setting a vector of unknown parameters as is defined in (14)

$$\boldsymbol{\alpha} = \left[a, \gamma, \omega, \phi \right]^{\mathrm{T}}$$

The accompanying file together with pre-processing provides N data pairs

$$\left(t_i, p_i \right) \quad i = 1, 2, \ldots, N$$

The suggested model function is

$$p(t; \boldsymbol{\alpha}) = a e^{-\gamma t} \cos(\omega t + \phi)$$

We are trying to solve the following overdetermined system of N nonlinear equations

$$f_i(\boldsymbol{\alpha}) \equiv p(t_i; \boldsymbol{\alpha}) = p_i \quad i = 1, 2, \ldots, N$$

Assignment 8

Build the Jacobian matrix **J** by calculating analytically the elements

$$J_{i,j} = \frac{\partial f_i(\boldsymbol{\alpha})}{\partial \alpha_j} = \frac{\partial p(t_i; \boldsymbol{\alpha})}{\partial \alpha_j} \quad j = 1, 2, 3, 4$$

It is enough to calculate one row in the Jacobian matrix for a fixed i.

Assignment 9

Nonlinear iterative methods are very sensitive to the initial guess (starting point). Because of this, it is of paramount importance to select an initial approximation that is sufficiently close to the solution in question. There is no unique technique for doing so. We are lucky since we can use the visual inspection method. To this end, do the following.

a) Plot the pre-processed signal in a separate window and activate the grid option to facilitate visual inspection.

b) Analyse the last three peaks. Visual inspection shows the first peak is squeezed between roughly $13 < t_1 < 15$; the second one between $15 < t_2 < 17$; and the third one occurs between $17 < t_3 < 19$. Write a script that finds the last three successive peaks in this plot. You will probably need to use a build-in function, `argmax`, together with the above constrains. Evaluate then the period, *T*. One way of doing so is $T_1 = t_2 - t_1$; another option is, $T_2 = t_3 - t_2$, but a better one is $T = \left(T_1 + T_2\right)/2 = \left(t_3 - t_1\right)/2$. Then evaluate the ordinary and angular frequency, $FR = 1/T$ and $\omega = 2\pi/T$.

c) Find also the values of the peaks, i.e. $p_1 = p(t_1)$, $p_2 = p(t_2)$ and $p_3 = p(t_3)$. Then estimate the decay ratio by $DR_1 = p_2/p_1$ and $DR_2 = p_3/p_2$. Next, average these values $DR = \left(DR_1 + DR_2\right)/2$, which gives somewhat better approximation.

d) Recapitulate Assignment 6 and estimate the decay constant, $\gamma = -FR \times \ln\left(DR\right)$, together with the amplitude *a*. Note that, if at time *t* a local maximum occurs, then it holds $\cos\left(\omega t + \phi\right) = 1$. It follows then, $p_i = ae^{-\gamma t_i}$ for *i* = 1,2,3. It gives three evaluations for the amplitude, a_i. The final estimate could be $a = \left(a_1 + a_2 + a_3\right)/3$.

e) Finally, estimate the phase, ϕ, keeping in mind the condition, $0 \le \phi < 2\pi$. From b), two successive maxima at t_1 and t_2 are known. Based on Assignment 6, calculate two estimates for the phase, ϕ_1 and ϕ_2, then average, $\phi = (\phi_1 + \phi_2)/2$.

Assignment 10

Write a computer code for solving the overdetermined system of non-linear equation

$$p(t_i; \boldsymbol{\alpha}) = p_i \quad i = 1, 2, \ldots, N$$

by the Newton-Raphson (or simply Newton's) method. Choose a stopping criterion to be a relative displacement i.e., $\left\|\Delta\boldsymbol{\alpha}^{(k)}\right\| / \left\|\boldsymbol{\alpha}^{(k+1)}\right\| \le \varepsilon_R$. Solve numerically the above system using:

(1) the initial guess found in Assignment 9; and (2) a relative tolerance, $\varepsilon_R = 10^{-6}$.

a) Report the number of iterations needed to achieve the specified relative tolerance;

b) Report the found parameters with 6 decimal places in a table below;

	α	γ	ω	ϕ	DR	RMSE
Guess:						
Found						

c) Plot the found model function as solid line together with the original (de-trended) signal as dots.

Assignment 11

Write a report in the form of a scientific manuscript, which typically follows (somewhat simplified) structure:

- Cover page with necessary details such as: Title, Author, Institution/Course, Year;
- Abstract (not more than half-page) with key words where you state very briefly the main subject, purpose and objectives, methodology, methods, and results;
- Content;
- Introduction;
- Theory;
- Methods (optional);
- Results;
- Discussion;
- References (optional);
- Appendices (Optional)

No plagiarism check is made. You may copy text from various sources but do not overuse this method. It is recommended write in your own words. Try to follow the academic style, which is characterized by

- Grammatically correct sentences with good structure;
- Using appropriate academic vocabulary;
- Creating a good text flow;
- Staying focused;
- Logical flow of ideas and argumentation;
- Expressing ideas in a brief and exact style;
- Avoiding repetitions;
- Good paragraphing.

Appendix I

Cramer's Rule

In 1750, a Genevan mathematician, Gabriel Cramer (1704–1752), published an explicit formula for the solution of a system of linear algebraic equations with a square matrix

$$\mathbf{Ax}=\mathbf{b};\quad \mathbf{x}=\left(x_1,x_2,\ldots,x_N\right)^T.$$

The rule states, in case when $\det\left(\mathbf{A}\right)\neq 0$, there is a unique solution given by

$$x_i=\frac{\det\left(\mathbf{A}_i\right)}{\det\left(\mathbf{A}\right)}$$

Here, $\mathbf{A}_i$ is the matrix formed by replacing the i-th column of $\mathbf{A}$ by the column vector $\mathbf{b}$.

Cramer's rule gives an expensive and very often numerically unstable solution already for low order matrices, $N\geq 10$. However, it is a very useful theoretical tool. In practice, this method is commonly used for low order matrices, especially when $N=2$ or $N=3$.

For example, consider a system of equations

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

Then

$$x=\frac{\begin{vmatrix} f_1 & b_1 \\ f_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}=\frac{f_1b_2-f_2b_1}{a_1b_2-a_2b_1};\quad y=\frac{\begin{vmatrix} a_1 & f_1 \\ a_2 & f_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}=\frac{a_1f_2-a_2f_1}{a_1b_2-a_2b_1}.$$

Consider one more system of equations

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

The Cramer rule gives

$$x=\frac{\begin{vmatrix} f_1 & b_1 & c_1 \\ f_2 & b_2 & c_2 \\ f_3 & b_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}};\quad y=\frac{\begin{vmatrix} a_1 & f_1 & c_1 \\ a_2 & f_2 & c_2 \\ a_3 & f_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}};\quad z=\frac{\begin{vmatrix} a_1 & b_1 & f_1 \\ a_2 & b_2 & f_2 \\ a_3 & b_3 & f_3 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}}.$$

There are several handy formulas to evaluate determinants of 3×3 matrices. Three of them are given below.

1) The Laplace formula for the determinant of a 3×3 matrix is

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}$$

2) The Leibniz formula for the determinant of a 3×3 matrix is

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 \left(b_2 c_3 - c_2 b_3 \right) - b_1 \left(a_2 c_3 - c_2 a_3 \right) + c_1 \left(a_2 b_3 - b_2 a_3 \right)$$

3) The Sarrus scheme for the determinant of a 3×3 matrix is

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} \begin{matrix} + & + & + & - & - & - \\ a_1 & b_1 & c_1 & a_1 & b_1 & \\ a_2 & b_2 & c_2 & a_2 & b_2 & \\ a_3 & b_3 & c_3 & a_3 & b_3 & \end{matrix} = a_1 b_2 c_3 + b_1 c_2 a_3 + c_1 a_2 b_3 - c_1 b_2 a_3 - a_1 c_2 b_3 - b_1 a_2 c_3$$

Appendix II

Binomial Theorem

In elementary algebra, the binomial theorem (binomial expansion) states that for any nonnegative integer n, it possible to expand

$$(x+y)^n = \binom{n}{0}x^n y^0 + \binom{n}{1}x^{n-1}y^1 + \binom{n}{2}x^{n-2}y^2 + \ldots + \binom{n}{n-1}x^1 y^{n-1} + \binom{n}{n}x^0 y^n$$

Using the summation notation, it reads

$$(x+y)^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k$$

The binomial coefficients are given by

$$\binom{n}{k} \equiv \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} = \frac{n!}{k!(n-k)!}$$

These numbers naturally appear in combinatorics where $\binom{n}{k}$ gives the number of different combinations of k elements that can be chosen from an n-element set. That's why they are often pronounced "n choose k". A particular case of the binomial theorem reads

$$(1+x)^n = \sum_{k=0}^{n}\binom{n}{k}x^k$$

Pascal's rule states the identity

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

This rule suggests organizing the binomial coefficients in a convenient form of the so called Pascal triangle, a fragment of which is shown below

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

The binomial theorem was known for the case, $n = 2$, by the Greek mathematician Euclid (of Alexandria) in around 300 BC and stated in its modern form by the French mathematician Blaise Pascal (1623 – 1662). It was published posthumously in 1665. In 1676, the English scientist Isaak Newton (1642 – 1726) showed that the formula also holds for negative integers.

$$(x+a)^{-n} = \sum_{k=0}^{\infty}\binom{-n}{k}x^k a^{-n-k}$$

Appendix III

Inner Product Space

In linear algebra, an inner product space is a linear vector space V over a field of scalars F with an additional structure called an inner product of two vectors, which is commonly denoted by angular brackets, $\langle \mathbf{a}, \mathbf{b} \rangle$. In the most general case, the field F is complex numbers but very often, it is enough to set the scalar field F to be real numbers, $F = \mathbb{R}$.

The inner product is defined as a mapping

$$\langle \bullet, \bullet \rangle : V \times V \to F$$

that obeys the following three axioms for all vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$ and all scalars $\alpha \in F$:

(1). Conjugate symmetry, $\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$; when $F = \mathbb{R}$, it is simply, $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$;

(2). Linearity in the first argument, $\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$ and $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$;

(3). Positive-definite, $\langle \mathbf{x}, \mathbf{x} \rangle > 0 \quad \forall \mathbf{x} \neq \mathbf{0}$.

The following is some elementary properties.

- $\langle \mathbf{x}, \mathbf{0} \rangle = 0$;
- $\langle \mathbf{x}, \mathbf{x} \rangle = 0 \mapsto \mathbf{x} = \mathbf{0}$;
- $\langle \mathbf{x} + \mathbf{y}, \mathbf{x} + \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{x} \rangle + 2\langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle$ when $F = \mathbb{R}$ (binomial expansion).

An inner product induces a norm and distance between two vectors.

- $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ (normed space);
- $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ (metric space).

An inner product allows defining an angle between two vectors:

- $\angle(\mathbf{x}, \mathbf{y}) \equiv \arccos \dfrac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$ when $F = \mathbb{R}$;
- $\mathbf{x} \perp \mathbf{y}$ when $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ (orthogonality).

It holds also

- $\|\alpha \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$ (homogeneity);
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (triangle inequality);
- $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \cdot \|\mathbf{y}\|$ (Cauchy-Swartz iniquality)

The Pythagorean Theorem in a normed space states, whenever $\mathbf{x} \perp \mathbf{y}$, it holds

$$\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 = \|\mathbf{x} + \mathbf{y}\|^2$$

If $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are orthogonal, i.e. $\langle \mathbf{x}_i, \mathbf{y}_j \rangle = 0$ for any $i \neq j$, Parseval's identity reads

$$\sum_{i=1}^{N} \|\mathbf{x}_i\|^2 = \left\| \sum_{i=1}^{N} \mathbf{x}_i \right\|^2$$

The Parallelogram Law reads in an inner product space

$$\|\mathbf{x} + \mathbf{y}\|^2 + \|\mathbf{x} - \mathbf{y}\|^2 = 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$$

Appendix IV

Fundamental Theorem of Algebra

The Fundamental theorem of algebra states that any non-constant polynomial with complex coefficients has at least one complex root. The theorem implies that any polynomial with complex coefficients of degree n has n complex roots, counted with multiplicity. A field F with the property that every non-constant polynomial with coefficients in F has a root in F is called algebraically closed, so the fundamental theorem of algebra states that the field $\mathbb{C}$ of complex numbers is algebraically closed.

Appendix V

Sample Statistics

In statistics, a population is the complete set of similar objects or events which is of interest for some question or experiment. Populations can have a finite size, though potentially very big such as all bacteria on earth, or an infinite size such as all possible outcomes of flipping a coin. A sample is a subset of the entire population.

Let $x \equiv \{x_1, x_2, \ldots, x_N\}$ and $y \equiv \{y_1, y_2, \ldots, y_N\}$ be two data sets (samples, observations, measurements) of equal size. Any data set may be characterized by:

- Sample mean, $\bar{x} \equiv \frac{1}{N}\sum_{i=1}^{N} x_i$;
- Sample standard deviation, $S_x \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$;
- Sample variance $\mathrm{Var}(x) = S_x^2 \equiv \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$;
- Sample covariance, $\mathrm{Cov}(x,y) \equiv \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$;
- Sample correlation, $\mathrm{Corr}(x,y) = \rho(x,y) \equiv \frac{\mathrm{Cov}(x,y)}{S_x \cdot S_y}$.

Note that some authors prefer the other definition of the sample variance

$$S_x^2 \equiv \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

because it serves as a better (unbiased) estimator for the underlying population variance. Such questions lie outside the scope of this project which focuses entirely on algebraic aspects.

It is a simple but instructive exercise to show that

$$\mathrm{Var}(x) = S_x^2 = \mathrm{Cov}(x,x)$$

If α is a number (scalar, constant), we define

$$\alpha + x = x + \alpha \equiv \{x_1 + \alpha, x_2 + \alpha, \ldots, x_N + \alpha\}$$

$$\alpha x = x\alpha \equiv \{\alpha x_1, \alpha x_2, \ldots, \alpha x_N\}$$

It is an elementary exercise to prove that

$$\overline{x+\alpha} = \bar{x} + \alpha; \quad \overline{\alpha x} = \alpha\bar{x}; \quad S_{x+\alpha} = S_x; \quad S_{\alpha x} = \alpha S_x; \quad S_{\alpha x}^2 = \alpha^2 S_x^2.$$

It is logical to define a sum of two data sets by

$$x + y \equiv \{x_1 + y_1, x_2 + y_2, \ldots, x_N + y_N\}$$

We define now a product of two data sets as

$$x \cdot y \equiv xy \equiv \{x_1 y_1, x_2 y_2, \ldots, x_N y_N\}$$

In turn, when $x = y$, it induces the definition of power as

$$xx \equiv x^2 \equiv \{x_1^2, x_2^2, \ldots, x_N^2\}; \quad x^n \equiv \{x_1^n, x_2^n, \ldots, x_N^n\}.$$

Thus, we arrive at

$$\overline{xy} \equiv \frac{1}{N}\sum_{i=1}^{N} x_i y_i; \quad \overline{x^2} \equiv \frac{1}{N}\sum_{i=1}^{N} x_i^2; \quad \overline{x^n} \equiv \frac{1}{N}\sum_{i=1}^{N} x_i^n.$$

The sample variance and covariance may now be rewritten as

$$\mathrm{Var}(x) = S_x^2 = \overline{\left(x-\bar{x}\right)^2}; \quad \mathrm{Cov}(x,y) = \overline{\left(x-\bar{x}\right)\left(y-\bar{y}\right)}.$$

It is a simple exercise to establish the following very useful identities

$$S_x^2 = \overline{\left(x-\bar{x}\right)^2} = \overline{x^2} - \bar{x}^2; \quad \mathrm{Cov}(x,y) = \overline{\left(x-\bar{x}\right)\left(y-\bar{y}\right)} = \overline{xy} - \bar{x}\cdot\bar{y}.$$

The average of powers of a data set logically leads to a concept of raw moments, m_n, sometimes also called "crude moments". They are defined as

$$m_n \equiv \overline{x^n} = \frac{1}{N}\sum_{i=1}^{N} x_i^n$$

In particular,

$$m_0 = 1; \quad m_1 = \bar{x}.$$

Similarly, central moments are defined by

$$\mu_n \equiv \overline{\left(x-\bar{x}\right)^n} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i-\bar{x}\right)^n = \frac{1}{N}\sum_{i=1}^{N}\left(x_i-m_1\right)^n$$

In particular,

$$\mu_0 = 1; \quad \mu_1 = 0; \quad \mu_2 = S_x^2.$$

Using the binomial coefficients, it is easy to establish

$$\mu_n = \frac{1}{N}\sum_{i=1}^{N}\left(x_i-m_1\right)^n = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{n}\binom{n}{k}\left(-1\right)^{n-k} x_i^k m_1^{n-k} = \sum_{k=0}^{n}\binom{n}{k}\left(-1\right)^{n-k} m_k m_1^{n-k}$$

For example,

$$\mu_2 = m_2 - m_1^2; \quad \mu_3 = m_3 - 3m_1m_2 + 2m_1^3; \quad \mu_4 = m_4 - 4m_1m_3 + 6m_1^2m_2 - 3m_1^4.$$

Again using the binomial coefficients, one inverts the previous identity

$$m_n = \frac{1}{N}\sum_{i=1}^{N} x_i^n = \frac{1}{N}\sum_{i=1}^{N}\left[\left(x_i-m_1\right)+m_1\right]^n = \sum_{k=0}^{n}\binom{n}{k}\mu_k m_1^{n-k}$$