CHAPTER ONE

INTRODUCTION

1.1: Background

It was not until the end of eighteenth century that instruments and methods attained a degree of refinement sufficient for the needs of figure determination. Since then, much geodetic work has been accomplished in Europe, America, Australia, India and Africa, and knowledge of the dimensions of the earth has steadily grown.

What is this subject of geodesy? Is it just an esoteric area of science with little modern practical application, or does it function quietly to the benefit of all but receive little public attention? Is it a product of the age of electronics, or does have a much longer history? In brief, what is it, how has it developed, what areas of live does it impinge upon, and how it put to the practical use? (Smith, J.R., 1997)

Definitions vary from the inevitable one line (Science of measuring the earth, or surveying any large part to it).

Geodesy, literally, means dividing the earth, and as a first objective of the practice of geodesy should provide an accurate framework for the control of national topographical surveys (Mertikas, 2011). Thus geodesy is the science that determines the figure of the earth and the interrelation of selected points on its surface by either direct or indirect techniques. (Torge, 1996)

These characteristics further makes it a branch of applied mathematics, one that must include observations that can be used to determine the size and shape of the earth and the definition of coordinate systems for three dimensional positioning, the variation of phenomena near to or on the surface, such as gravity, tides, earth rotation, crustal movement, and deflection of the plumb line (vertical); together with units of measurement, and methods of representing the curved earth surface on a flat sheet of paper.

Nowadays there is a trend today for the term geodesy to be applied in an umbrella manner particularly in the European community, to describe all activities from valuation, land management, soil testing, cartography, boundary surveys, land information systems, and in fact every activity except geodesy in its traditional definition! In addition we now are pressed to use the term Geomatics to cover almost as wide a selection as the above list. May be it will soon be possible to refer to geomatic and geodesy as covering everything that is understood to be under the authority of a surveying engineer. (Smith, 1997)

It is good to see that geodesy is one of the important subjects for endless applications, by the wide range of uses now being found for the global positioning system (GPS). It is an earth - centered system, relying on earth – orbiting satellites, but it has recreational uses in addition to survey applications and acceptable requirements from a few millimeters to many meters depending on the use.

However, before venturing into the realms of artificial satellites it will be instructive to trace the origins of geodesy from the first few centuries B.C. up to the present day. From the times of a flat earth concept, through the sphere and spheroid to the geoid; from the knotted rope for measurement to suspended wires, electromagnetic systems, laser ranging to the moon, and the use of orbiting satellites.

The results of geodetic measurements show that the earth very close lyapproximates to an oblate spheroid, which is the solid generated by rotation of an ellipse about its minor axis. The actual figure, considered as a gravitational equipotential surface at mean sea level, deviates slightly and irregularly from a true spheroid, and this is recognized by giving it the name "geoid" since, however, geodetic computations can be made with sufficient precision on the assumption of a spheroidal form; figure determinations are directed to ascertaining the dimensions of the spheroid which most nearly coincides with the actual figure. (Smith, 1997)

1.2 Objectives of the Thesis

1. To establish an optimum model for the densification of orthometric heights.
2. To establish an adequate model for densification of geoidal heights.
3. To enable topographic maps of different scales and contour intervals to be prepared using GPS / GIS techniques.

1.3 Thesis layout

The thesis consists of seven Chapters, including this introductory Chapter, in addition to two appendices. The other Chapters are summarized as below:-

Chapter two contains the different types of reference ellipsoids with various types of vertical datum, and mean sea level presented.

In Chapter three details the digital terrain modeling, data collection, measurements pattern, and modeling technique are presented.

Chapter four discusses the least squares collocation, the covariance matrices, the signal, the noise, and the solution of parameters.

Chapter five details the modeling program, study area, data collection for control work. It is also includes Instruments used and methodology, the programming language, back ground, design goals, and applications.

Chapter six outlines the results of the tests carried out using the developed program.

Chapter seven summarizes the conclusions and recommendations for future work.

CHAPTER TWO

REFERENCE ELLIPSOIDS AND VERTICAL DATUMS

2.1 Background

Mapping involves determining geographic locations of features on the earth, transforming these locations on flat maps through selected map projections and graphically symbolizing these features. Geographic locations are specified by geographic coordinates called latitude and longitude. To establish a system of geographic coordinates, we first must know the shape and size of the earth.

The earth is a very smooth geometrical figure. Much of the earth surface appears rugged and rough to us, but even the heights peaks and deepest ocean trenches are barely noticeable irregularities on the smoothly curving surfaces. (Robinson, Arthur Howard, 1995)

We have to examine three ever more accurate approximations to the earth's shape: the sphere, the ellipsoid, and the geoid.

2.1.1 Spherical earth

More than 2000 years ago most educated people knew that, if we disregard such features as hills and valleys, the earth is spherical in shape. This understanding was due in part to the teaching of Pythagoras (6th century B.C.) that human must live on a body of the " perfect shape "- a perfect sphere. More compelling, however, were Aristotle's (4th century B.C.) arguments for spherical earth. He noted that sailing ships always disappear from view hull first, mast last, rather than becoming ever smaller dots on the horizon of a flat earth. And the earth's spherical shape becomes widely accepted in ancient Greece and later civilizations with access to Greek writings. (Martin, 2005)

Determining the spherical earth's size was another matter. Again, the first calculation was made by a Greek Scholar. About 250 B.C. Eratosthenes, head of a great Egyptian library in Alexandria, came close to the figures for the earth's circumference we now accept.

2.1.2 Ellipsoidal earth

Until the late 1600s, the earth was thought to be perfectly spherical in shape. The change came around 1670, when Isaac Newton proposed , as a consequence of his theory of gravity, that there is a slight bulging of the earth at the equator due to the greater centrifugal force generated by the earth's rotation. This equatorial bulging would produce a slight flattening at the poles, predicted by Newton to be about 1/300th of the equatorial radius.

Newton's prediction was confirmed by measurements taken from 1735 to 1743 by expeditions sent to Ecuador and Finland to measure the ground distance for one degree of angular change (one degree of latitude) in equatorial and Polar Regions. The polar distance was found to be slightly greater due to flattening.

From 1800 to the present date, at least 20 determinations of the earth's radii and flattening (oblateness) have been made from measurements taken at widely different locations.

2.1.3 Geoidal Earth

An even more faithful figure of the earth, called the geoid (meaning earth like), deviates, ever so slightly, from the ellipsoid in an irregular manner. The geoid is the three dimensional shape that would be approximated by mean sea level in the oceans and the surface of a series of hypothetical sea level canals criss-crossing the continents. In more special terms, it is a sea level equipotential surface, the surface on which gravity is everywhere equal to its strength at mean sea level. If the earth were of uniform geological composition and devoid of mountain ranges, ocean basins, and other vertical irregularities, the geoid surface would match the ellipsoid exactly. However, due primarily to variations in rock density and topographic relief, the geoid surface deviates from the ellipsoid by up to 100m in certain locations. (Maximenko and Niiler, 2005).

Note that the "hills and valleys" on the geoid do not correspond with continents and oceans.

Indeed the highest point on the geoid is 75 meters above the ellipsoid in New Guinea and the lowest point is 104 meters below at the Southern tip of India.

2.1.4 The World Geoidetic System

The World Geodetic System of 72 and 84 ellipsoids, determined from satellite orbital data, are considered more accurate than the earlier ground measurement determinations, but may not give the best fit for a particular part of the earth. The Clarke 1866 ellipsoid, based on measurements taken in Europe, India, Peru, Russia, and South Africa, is of special interest in the United States, since it has been used for mapping in North America until recently. North America cartographers are now rapidly switching to the WGS 84 ellipsoid, which is a global standard.

2.2 Reference Ellipsoids

A reference ellipsoid, also called spheroid, is a simple mathematical model of the Earth’s shape. An ellipsoid of revolution, or simply an “ellipsoid,” is the shape that results from rotating an ellipse about one of its axes.

Oblate ellipsoids are used for geodetic purposes because the Earth’s polar axis is shorter than its equatorial axis.

2.3 Local Reference Ellipsoids

Datums and cartographic coordinate systems depend on a mathematical model of the Earth’s shape upon which to perform trigonometric computations to calculate the coordinates of places on the Earth and in order to transform between geocentric, geodetic, and cartographic coordinates. The transformation between geodetic and cartographic coordinates requires knowledge of the ellipsoid being used, (Bugayevskiy & Snyder 1995, Qihe, Snyder & Tobler 2000, Snyder 1987).

The transformation from geodetic to geocentric Cartesian coordinates is accomplished by Helmert’s projection, which also depends on an ellipsoid (Heiskanen & Moritz (1967)) as does the inverse relationship; Meyer (2002).

Measurements taken must be reduced to a common surface for geodetic surveying, and a reference ellipsoid provides that surface. Therefore, all geodetic horizontal datums depend on the availability of a suitable reference ellipsoid.

Until recently, the shape and size of reference ellipsoids were established from extensive, continentalsized triangulation networks and Gore, (1889), Crandall (1914), Shalowitz,(1938), Schwarz (1989), Dracup (1995), Keay (2000), although there were at least two different methods used to finally arrive at an ellipsoid

- The “arc” method for Airy 1830, Everest 1830, Bessel 1841 and Clarke 1866.
- The “area” method for Hayford (1909).

The lengths of (at least) one starting and ending baseline were measured with instruments such as rods, chains, wires, or tapes and the lengths of the edges of the triangles were subsequently propagated through the network mathematically by triangulation.

For early triangulation networks, vertical distances were used for reductions and typically came from trigonometric heights or barometric measurements the result of this was that each region in the world thus measured had its own ellipsoid, and this gave rise to a large number of them; (DMA (1995) and Meyer (2002)). It was impossible to create a single, globally applicable reference ellipsoid with triangulation networks due to the inability to observe stations separated by large bodies of water. Local ellipsoids did not provide a vertical datum in the ordinary sense, nor were they used as such.

Before GPS, all high-accuracy heights were measured with some form of leveling, and determining an ellipsoid height from an orthometric height requires knowledge of the deflection of the vertical, which is obtained through gravity and astronomical measurements (Heiskanen & Moritz (1967).

Deflections of the vertical, or high-accuracy estimations thereof, were not widely available prior to the advent of high-accuracy geoid models. Second, the location of a local ellipsoid was arbitrary in the sense that the center of the ellipsoid need not coincide with the center of the Earth (geometric or center of mass), so local ellipsoids did not necessarily conform to mean sea level in any obvious way.

In summary, local ellipsoids are essential in geodetic coordinates computation, geoidal determination, satellite orbit determination, and cartographic coordinate systems .As reported by Fischer (2004).

2.4 Equipotential Ellipsoids

Global reference ellipsoids have been created using Very Long Baseline Interferometry (VLBI) for GRS 80 (Moritz 2000)), satellite geodesy for the World Geodetic System 1984 (WGS 84) (DMA 1995), along with various astronomical and gravitational measurements. Very long baseline interferometry and satellite geodesy permit high-accuracy baseline measurement between stations separated by oceans. As a result, these ellipsoids model the Earth globally; they are not fitted to a particular local region.

Both WGS 84 and GRS 80 have size and shape such that they are a best-fit model of the geoid in a least-squares sense. Quoting Moritz (2000), The Geodetic Reference System 1980 has been adopted at the XVII General Assembly of the International Union of Geodesy and Geophysics (IUGG) in Canberra, December 1979, by declaring the following (recognizing that the Geodetic Reference System 1967 , no longer represents the size, shape, and gravity field of the Earth to an accuracy adequate for many geodetic, geophysical, astronomical and hydrographic applications and considering that more appropriate values are now available, recommends that the Geodetic Reference System 1967 be replaced by a new Geodetic Reference System 1980, also based on the theory of the geocentric equipotential ellipsoid, defined by the following constants)

- Equatorial radius of the Earth: a = 6378137 m;

- Geocentric gravitational constant of the Earth (including the atmosphere):

$$GM = 3,986,005 \times 10^3 \mathrm{m}^3\mathrm{s}^{-2}$$

- Dynamical form factor of the Earth, excluding the permanent tidal deformation:

$$J2 = 108,263 \times 10^{-8}; \text{ and}$$

- Angular velocity of the Earth:

$\omega = 7292115 \times 10^{-11} \text{rad}\, s^{-1}$.

Equipotential ellipsoid models of the Earth constitute local ellipsoids, which are purely geometric, whereas equipotential ellipsoids include the geometric but also concern with gravity. Indeed, GRS 80 is called an "equipotential ellipsoid" (Moritz 2000) and, using equipotential theory together with the aforementioned listed above, one derives the flattening of the ellipsoid rather than measuring it geometrically.

Datums that employ GRS 80 and WGS 84 (e.g., (North American Datum (NAD) 83, International Terrestrial Reference System (ITRS), and WGS 84) are intended to be geocentric, meaning that they intend to place the center of their ellipsoid at the Earth's center of gravity. It is important to note, however, that NAD 83 currently places the center of GRS 80 roughly two meters away from the center of ITRS and that WGS 84 is currently essentially identical to ITRS.

Equipotential ellipsoids are both models of the Earth's shape and first-order models of its gravity field. Somiglinana (1929) developed the first rigorous formula for normal gravity (Heiskanen & Moritz (1967)) and the first internationally accepted equipotential ellipsoid was established in 1930. It had the form: (Blakely 1995)

$$g0 = 9.78046(1 + 0.0052884 \sin^2 \varphi - 0.0000059 \sin^2 2 \varphi) \qquad (2.1)$$

Where

$g0$ = acceleration due to gravity at a distance 6,378,137 m from the center of the idealized Earth; and

φ = geodetic latitude

The value $g0$ is called theoretical gravity or normal gravity. The dependence of this formula on geodetic latitude will have consequences when closure errors arise in long leveling lines that run mostly north-south compared to those that run mostly east-west.

The most modern reference ellipsoids are GRS 80 and WGS 84. As given by (Blakely, 1995).The closed-form formula for WGS 84 normal gravity is:

$$g0=9.7803267714*(1+0.00193185138639\sin^2 \varphi)/(1-0.00669437999013 \sin^2 \varphi)^{½} \qquad (2.2)$$

2.5 Equipotential Ellipsoids as Vertical Datums

Equipotential ellipsoids are more suitable to be used as vertical datums in the ordinary sense than local ellipsoids and, in fact, they are used as such. In particular, GPS-derived coordinates expressed as geodetic latitude and longitude presents the third dimension as an ellipsoid height.

Equipotential ellipsoids are models of the gravity that would result from a highly idealized model of the Earth; one whose mass is distributed homogeneously but includes the Earth's oblate shape, and spinning like the Earth. The geoid is not a simple surface compared to an equipotential ellipsoid, which can be completely described by just the four parameters listed before. The geoid's shape is strongly influenced by the topographic surface of the Earth. The geoid is a convex surface by virtue of satisfying the Laplace equation, and its apparent concavity is a consequence of how the geoid is portrayed on a flat surface (Van´ıˇcek & Krakiwsky 1986).

Equipotential ellipsoids are useful as vertical datums, they are usually unsuitable as a surrogate for the geoid when measuring orthometric heights. Equipotential ellipsoids are "best fit" over the entire Earth and, consequently, they typically do not match the geoid particularly well in any specific place. (Meyer, T.H., Roman, D.R. and Zilkoski, D.B., 2006)

2.6 The World Geodetic System 1984 (WGS 84)

It is the reference frame used by the U.S. Department of Defense and is defined by the National Geospatial-Intelligence Agency (NGA) to cover the entire world in its scope. This is a great advantage over NAD 27 and NAD 83, even though it is set up similarly to NAD 83. WGS 84 is used by the Department of Defense for its mapping needs, including its GPS "broadcast" and "precise" orbits.

In January 1987, it became the default standard datum for coordinates stored in recreational and commercial GPS units. Since its inception, WGS 84 has been updated twice, most recently in 2008, to increase its operability with GPS systems.

2.6.1 Adindan Sudan local Datum

Adindan datum is the historical local datum of Sudan that all triangulation and traverse network observations has subsequently been reduced to it, also it has been used as Geodetic Datum in A Eritrea , Ethiopia Burkina Faso, Cameroon, Mali and Senegal.

Adindan references the Clarke 1880 (RGS) ellipsoid of a semi major axis of 6378249.145 m, and 293.465 reciprocal of the flattening (1/f), and the Greenwich prime meridian.

Adindan base terminal ZY was chosen as the origin of 22º 10' 7.1098" latitude (North) and 31º 29' 21.6079" longitude (East), with azimuth of 58º 14' 28.45" from the north to YY.ZY is now about 10 meters below the surface of Lake Nasser.

Adindan is a geodetic datum. The 12th parallel traverse of 1966-70 (Point 58 datum, code 6620) is connected to the Adindan network in western Sudan. This has given rise to misconceptions that the Adindan network is used in west Africa.

Since all existing maps and old survey information in Sudan are reduced to adindan national datum. Informations about Adindan network is needed. Relationship between the world geodetic system 1984 (WGS84) and the local geodetic datum need to be established.

2.7 Mean Sea Level

Mean sea level was long considered a satisfactory approximation to the geoid and therefore suitable for use as a reference surface. It is now known that mean sea level can differ from the geoid by up to a meter or more, but the exact difference is difficult to determine.

For heights, the most common datum is mean sea level. Using mean sea level for a height datum is perfectly natural because most human activity occurs at or above sea level. The NGS Glossary definition of mean sea level is “The average location of the interface between ocean and atmosphere, over a period of time sufficiently long so that all random and periodic variations of short duration average to zero.”

The National Oceanic and Atmospheric Administration’s (NOAA) National Ocean Service (NOS) Center for Operational Oceanographic Products and Services (CO-OPS) has set 19 years as the period suitable for measurement of mean sea level at tide gauges (National Geodetic Survey Reports 1986). The choice of 19 years was chosen because it is the smallest integer number of years larger than the first major cycle of the moon’s orbit around the Earth. This accounts for the largest of the periodic effects mentioned in the definition. Bomford (1980) and Zilkoski (2001). Local mean sea level is often measured using a tide gauge.

Figure 2.1 depicts a tide house, “a structure that houses instruments needed to measure and record the instantaneous water level inside the tide gauge and built at the edge of the body of water whose local mean level is to be determined.”

By this time it was a known fact that not all mean sea-level stations have the same height.

To begin with, all mean sea-level stations are at an elevation of zero by definition. Second, water seeks its own level, and the oceans have no visible constraints preventing free flow between the stations (apart from the continents).

According to differences in temperature, chemistry, ocean currents, and the water in the oceans is constantly moving at all depths. Seawater at different temperatures contains different amounts of salt and, consequently, ocean eddies, mean sea level is not at the same height everywhere.

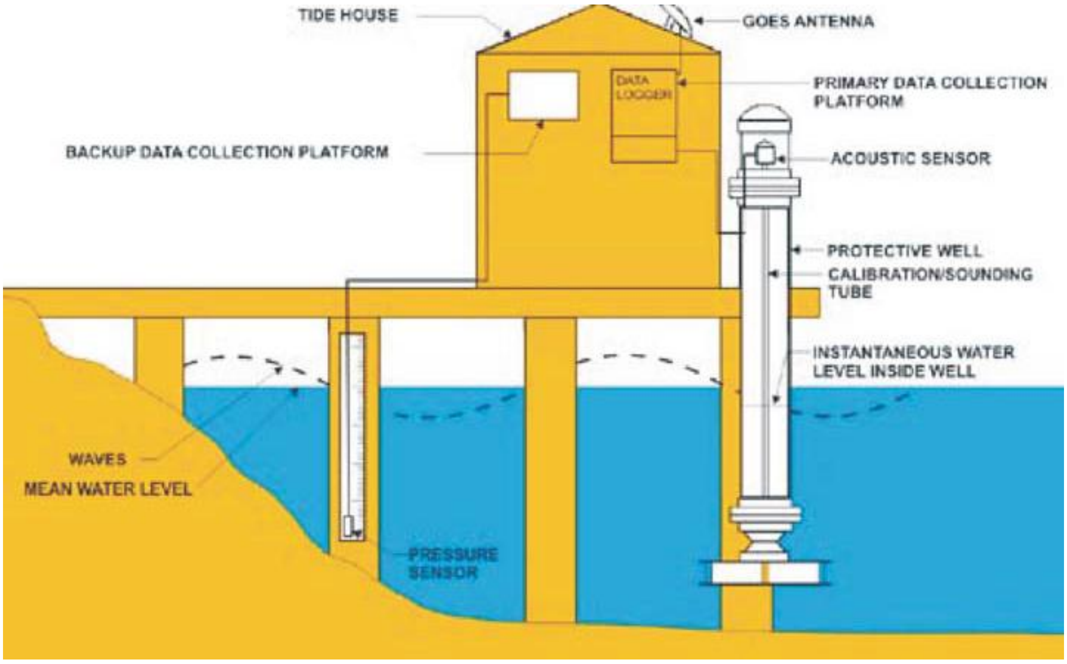

TIDE HOUSE
GOES ANTENNA
DATA LOGGER
PRIMARY DATA COLLECTION PLATFORM
BACKUP DATA COLLECTION PLATFORM
ACOUSTIC SENSOR
PROTECTIVE WELL
CALIBRATION/SOUNDING TUBE
INSTANTANEOUS WATER LEVEL INSIDE WELL
WAVES
MEAN WATER LEVEL
PRESSURE SENSOR

Figure 2.1: The design of a NOAA tide house and tide gauge used for measuring mean sea level. Source: (NOAA 2007).

has density gradients. These density gradients give rise to immense deep-ocean cataracts that constantly transport massive quantities of water from the poles to the tropics and back (Broecker 1983, Ingle 2000, Whitehead 1989). The sun's warming of surface waters causes the global-scale currents that are well-known to mariners in addition to other more subtle effects (Chelton, et al 2004). Geostrophic effects cause large scale, persistent ocean eddies that push water against or away from the continents, depending on the direction of the eddy's circulation. These effects can create sea surface topographic variations of more than 50 centimeters (Srinivasan 2004). As described by Zilkoski (2001), the differences are due to currents, prevailing winds and barometric pressures, water temperature and salinity differentials, topographic configuration of the bottom in the area of the gauge site, and other Physical causes (Tide is the major).

In essence, these factors push the water and hold it up shore or away-from-shore further than would be the case under the influence of gravity alone.

2.7.1 Sudan Mean Sea Levels (MSL) Height

There are two Datum's for mean sea level in Sudan namely irrigation datum and Sudan survey authority datum.

2.7.1.1 Irrigation Datum

The survey of bench-mark referred to this datum is made by survey of Egypt and it is divided into two sections:

(1) North of Khartoum which is referred to Alexandria mean sea level, and running a line of a precise leveling from Alexandria to
wadi Halfa. in 1906-and 1907 a second order leveling was run from Wadi Halfa to a polt Bench-mark in Sudan ministry of irrigation.

(2) South of Khartoum assuming gauge height of 360. 000 meters as Zero of Khartoum Gauge.

The bench mark monument referred to this datum (bolt and Pile) is similar to that of the Sudan survey authority.

2.7.1.2 Sudan Survey Authority Datum

This is the official Sudan Mean Sea Level datum (M.S.L), it's refereed to mean sea level at Alexandria Port. A precise leveling has been carried out from Alexandria and the Bench mark is of permanent nature, either a pile or a bolt. The bolt benchmark is the one chiseled into a wall, or that are permanently attached to a stable foundation, such as concrete posts, bridge, buildings, or a specifically constructed concrete block. These markers are then used as starting control points by subsequent surveyors and other users to establish the elevation of nearby points. Most of the Sudan survey authority's benchmarks still exist but others are damaged. The description of those Benchmarks is now misleading and needs to be updated to easily be found.

2.8 Tidal Datums

2.8.1 Principal Tidal Datums

A vertical datum is called a tidal datum when it is defined by a certain phase of the tide. Tidal datums are local datums and are referenced to nearby monuments. Since a tidal datum is defined by a certain phase of the tide there are many different types of tidal datums. Mean Higher High Water (MHHW), Mean High Water (MHW), Mean Sea Level (MSL), Mean Low Water (MLW), and Mean Lower Low Water (MLLW).

A determination of the principal tidal datums is based on the average of observations over a 19-year period,. A specific 19-year metonic cycle is denoted as a National Tidal Datum Epoch (NTDE). Users need to know which NTDE their data refer to.

- Mean Higher High Water (MHHW): MHHW is defined as the arithmetic mean of the higher high water heights of the tide observed over a specific 19-year metonic cycle denoted as the NTDE. Only the higher high water of each pair of high waters of a tidal day is included in the mean. For stations with shorter series, a comparison of simultaneous observations is made with a primary control tide station in order to derive the equivalent of the 19-year value (Marmer 1951).

- Mean High Water (MHW) is defined as the arithmetic mean of the high water heights observed over a specific 19-year metonic cycle. For stations with shorter series, a computation of simultaneous observations is made with a primary control station in order to derive the equivalent of a 19-year value. The survey carried out in 1958 to connect the two datum's show there're three meters difference between the two systems. (Marmer 1951).

- Mean Sea Level (MSL) is defined as the arithmetic mean of hourly heights observed over a specific 19-year metonic cycle. Shorter series are specified in the name, such as monthly mean sea level or yearly mean sea level (Marmer, 1951, Hicks, 1985).

- Mean Low Water (MLW) is defined as the arithmetic mean of the low water heights observed over a specific 19-year metonic cycle. For stations with shorter series, a comparison of simultaneous observations is made with a primary control tide station in order to derive the equivalent of a 19-year value (Marmer 1951).

- Mean Lower Low Water (MLLW) is defined as the arithmetic mean of the lower low water heights of the tide observed over a specific 19-year

Metonic cycle. Only the lower low water of each pair of low waters of a tidal day is included in the mean.

2.8.2 Other Tidal Datums

Other tidal values typically computed include the Mean Tide Level (MTL), Diurnal Tide Level (DTL), Mean Range (MR), Diurnal High Water Inequality (DHQ), Diurnal Low Water Inequality (DLQ), and Great Diurnal Range (GDR).

- Mean Tide Level (MTL) is a tidal datum which is the average of Mean High Water and Mean Low Water.

- Diurnal Tide Level (DTL) is a tidal datum which is the average of Mean Higher High Water and Mean Lower Low Water.

- Mean Range (MR) is the difference between Mean High Water and Mean Low Water.

- Diurnal High Water Inequality (DHQ) is the difference between Mean Higher High Water and Mean High Water.

- Diurnal Low Water Inequality (DLQ) is the difference between Mean Low Water and Mean Lower Low Water.

- Great Diurnal Range (GDR) is the difference between Mean Higher High Water and Mean Lower Low Water.

All of these tidal datums and differences have users that need a specific datum or difference for their particular use. The important point for users is to know which tidal datum their data are referred to. Like geodetic vertical datums, local tidal datums are all different from one another, but they can be related to each other.

CHAPTER THREE

DIGITAL TERRAIN MODELING

3.1 Introduction

Digital terrain modeling is a particular form of computer surface modelling that numerically represent the surface of the Earth. The initial concept of a digital terrain model (DTM) originated in the USA during the late 1950s (Miller and La Flarnme, 1958).

The term DTM originally referred to the use of cross-sectional height data to describe the terrain. Other terms are Digital Elevation Model (DEM), Digital Height Model (DHM), Digital Ground Model (DGM), and Digital Terrain Elevation Model (DTEM), are also used to describe the same process.

3.2 Sources of data:

The three main methods which can be used to acquire elevation data are:

(i) Ground survey methods.
(ii) Photogrammetric methods.
(iii) Graphics digitizing methods.

3.2.1 Ground survey methods

In ground survey method, elevation data can be acquired by using the total stations , Global Positioning System (GPS), and digital levels.

3.2.2 Photogrammetric methods

In photogrammetric methods elevation data can be acquired by using digital plotters.

3.2.3 Graphics digitizing methods

In this method the actual DTM spot height or elevation data is derived by interpolation from the digitized contour lines contained in existing topographic maps.

3.3 Measurement patterns

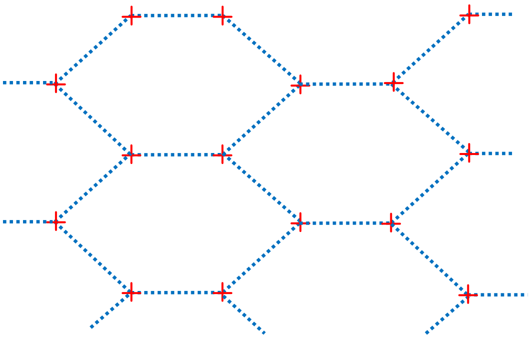
The required terrain elevation information may be obtained in any one of several sampling patterns.

3.3.1 Systematic sampling

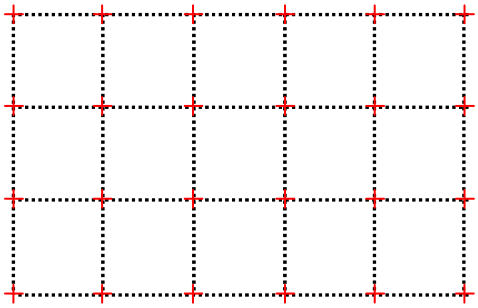
The spot heights may be measured in a regular geometric (square, rectangular, triangular) pattern (Fig. 3.1).

3.3.2 Progressive sampling

Originally proposed by Makarovic of the I.T.C., of the Netherlands (Makarovic, 1973, 1975).The measurement of grid points is varied in different parts of the grid, to matched the local roughness of the terrain surface (Fig. 3.2), to automatically or semi automatically optimize the relationship between specified accuracy, sampling density and terrain characteristics.



Hexagonal



Square

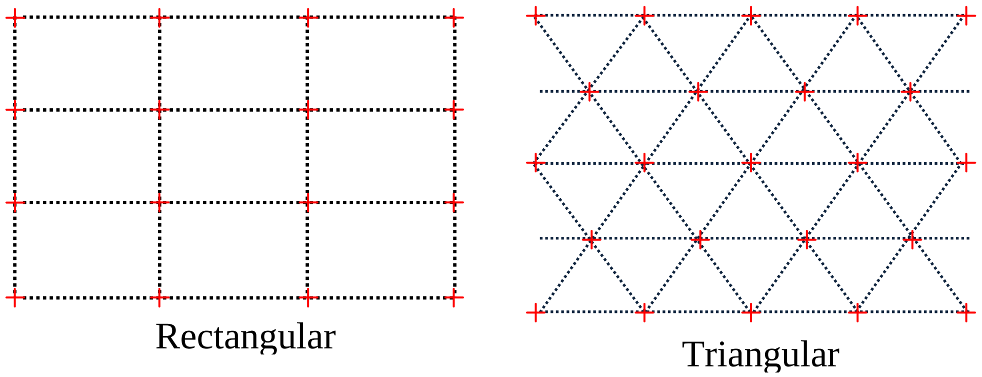


Figure 3.1 Regular grid patterns

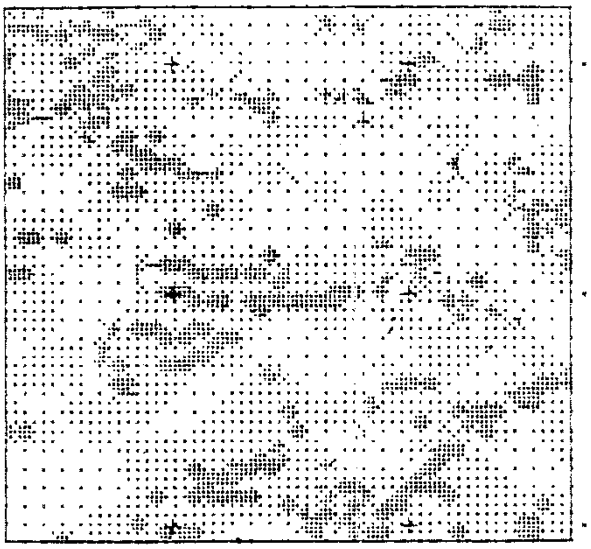


Figure 3.2 Progressive sampling

3.3.3 Random sampling

It is widely used by field surveyors and photogrammetrists, to measure heights selectively at significant points only-at the tops of hills, in hollows and along breaks of slope, ridge lines and streams. The measured points will be randomly located, that is, an irregular network of points during the reconnaissance and interpretation of the terrain features.

3.3.4 Composite sampling

It combines the elements of both of the above approaches (Makarovic, 1977).The basic grid-measuring pattern will be supplemented by the measurements made at significant points in the terrain, e.g. on hill tops, along break lines and streams, as, mentioned in 3.3.3.

3.3.5 Measured contours

To have the measurements in the form of digital coordinate data, contours are measured in a stereo model or from an existing topographic map over the whole area to be modeled.

3.4 Modelling techniques

Programs written for terrain modelling applications in surveying engineering, basically follow one or another of two main approaches:

(i) They make use of height data which has been collected or arranged in the form of a regular (rectangular or square) grid

(ii) They are based on a triangular network of irregular size, shape and orientation, based on randomly-located height data,

As Fig. 3.3 shows, these two approaches can be conducted either wholly independent of one another or they can be combined to give a composite or hybrid approach to terrain modeling and contouring.

3.4.1 Grid-based terrain modelling

The data comprising the terrain model is measured or collected in the form of a regular grid. Direct modelling of the grid can take place. A digital terrain data will often have been collected at specific locations, either in the field using GPS, or by photogrammetric methods. A preliminary random-to-grid interpolation must be carried out which converts this measured data to a suitably dimensioned regular grid.

Usually the following interpolation methods are distinguished:

(i) Point wise methods

(ii) Global methods

(iii) Patch wise methods

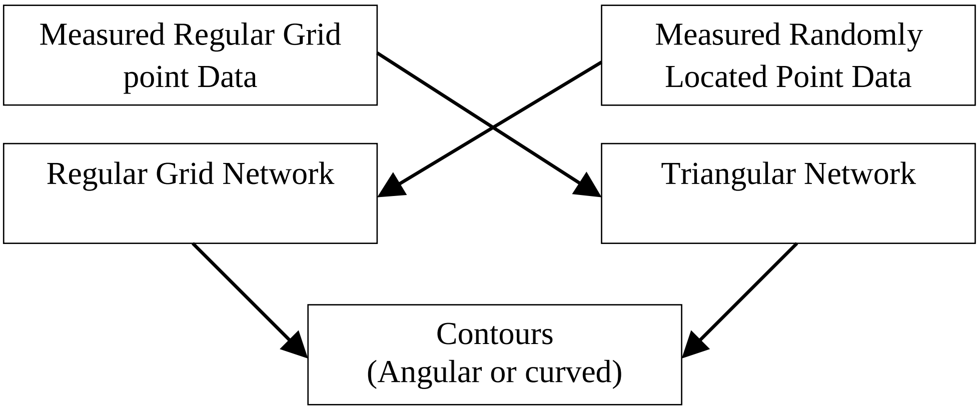

Measured Regular Grid point Data
Measured Randomly Located Point Data
Regular Grid Network
Triangular Network
Contours
(Angular or curved)

Figure 3.3 Overall relationships between measured point data, networks and contours in terrain modeling

3.4.1.1 Point wise methods

These involve the interpolation of the values of the terrain elevation at each specific grid node from its neighboring randomly-located measured height points. The determination of the height of each individual point are based on a search for the set of nearest neighbors, followed by the averaging of their heights weighted inversely by some function of their respective distances *d* from the position of the grid node. This weight $w= 1/d^m$ where ^{m} is the power used, typically in the range 0.5 to 4.

If the measured terrain model data takes the form of contours, another form of search may be implemented via the so-called sequential steepest slope algorithm described by Leberl and Olsen (1982) (Fig. 3.4). In this procedure, a search is made along each of the four lines passing through the required grid node and oriented along the grid directions (VV and HH) and their bisectors (U U and GG). The intersection of each of the eight directions with the nearest contours is established and the slope of each of the four lines calculated. The line with the steepest slope is then selected, and the value of the elevation of the grid node established by linear interpolation along this line-for instance, in the example shown in Fig. 3.4, search line GG is the steepest, and the height of the grid node *P* is derived from

$$H_p = [H_1 - H_5]/(1, 5)[(P, 5) + H_5$$

3.4.1.2 Global methods

These involve the fitting of a single three-dimensional surface defined by a high-order polynomial through all of the measured randomly-located terrain height points existing within the model points. (Fig. 3.5).

3.4.1.3 Patch wise methods

These lie in an intermediate position between the point wise method and the global method. The whole area to be modeled is divided into a series of equal-sized patches of identical shape.

The shape of each patch is in form, typically square or rectangular. The elevation of all the grid points falling within each individual patch can be interpolated using these parameters.

(i) Exact-fit patches (Fig. 3.6a) may be defined in which each patch abuts exactly on to its neighbors. The difficulty that may result from the use of such patches is that they may result in sharp discontinuities along their junctions, which show up markedly when the isoclines or contours are finally produced.

(ii) The alternative is to use an arrangement of overlapping patches (Fig. 3.6b), in which case there will be common points lying within the overlap which will be used in the computation of the parameters for each patch and, indeed, can be used to ensure a smooth continuity or transition between adjacent patches.

The advantages of using patch wise methods over global methods are that quite low order terms (parameters) can be used to satisfactorily describe each patch. So only few unknowns need to be solved, via simultaneous equations using least-squares method for each patch. Also, once the unknown parameters have been solved for, it is easy to calculate the derived points, that is, the grid nodes, by back-substitution in the functions or equations describing the patch. However, there are also some disadvantages of the patch wise method. In the first place, it needs much more organization of its data and of its processing than point wise or global methods. Also, the subdivision of the model surface into patches needs to be carried out with care.

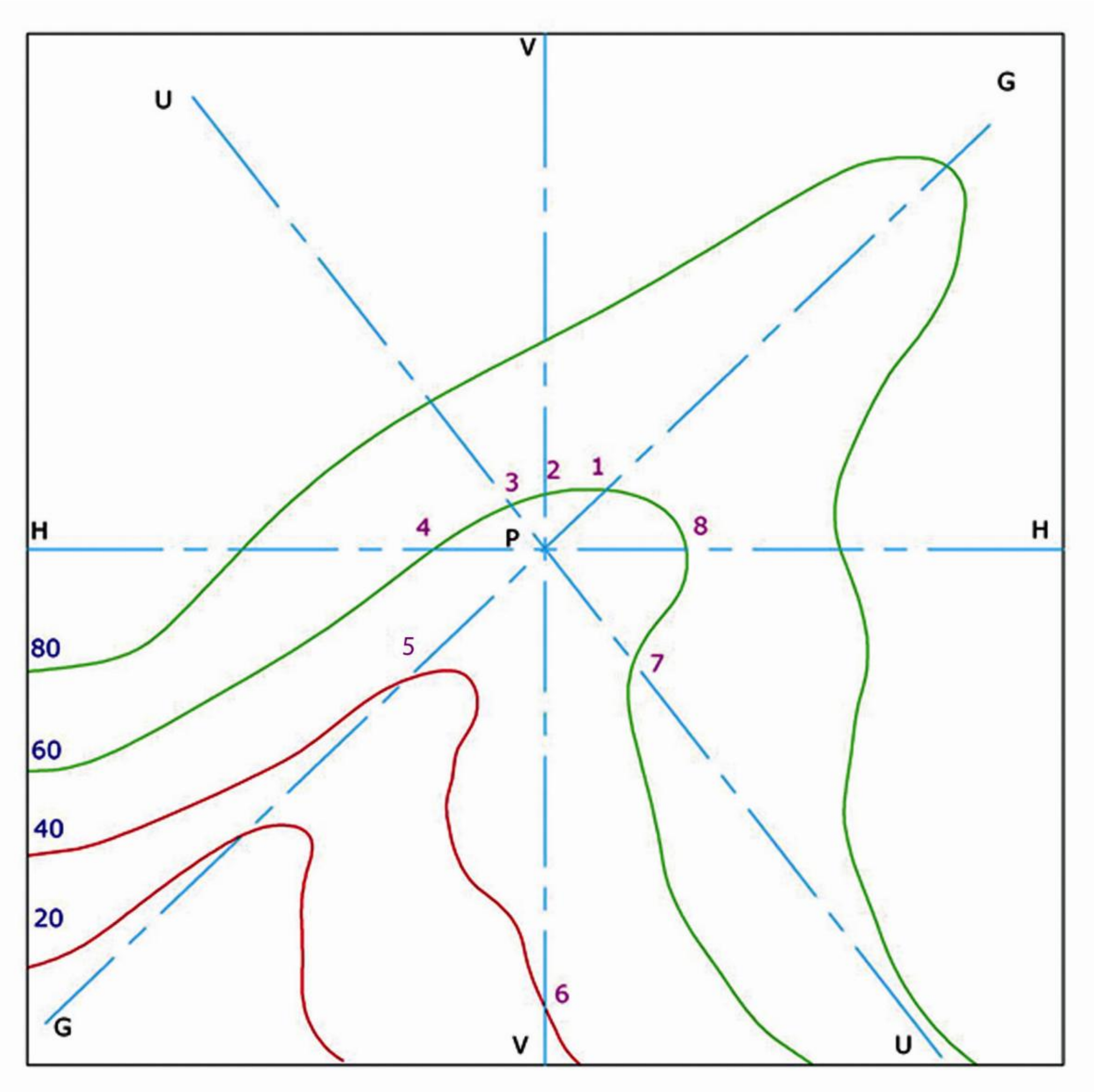


Figure 3.4 Sequential steepest slope algorithm showing cross-sections *HH*, *VV*, *UU* and *GG*. And the intersection points in the contours (Leberl and Olsen, 1982).

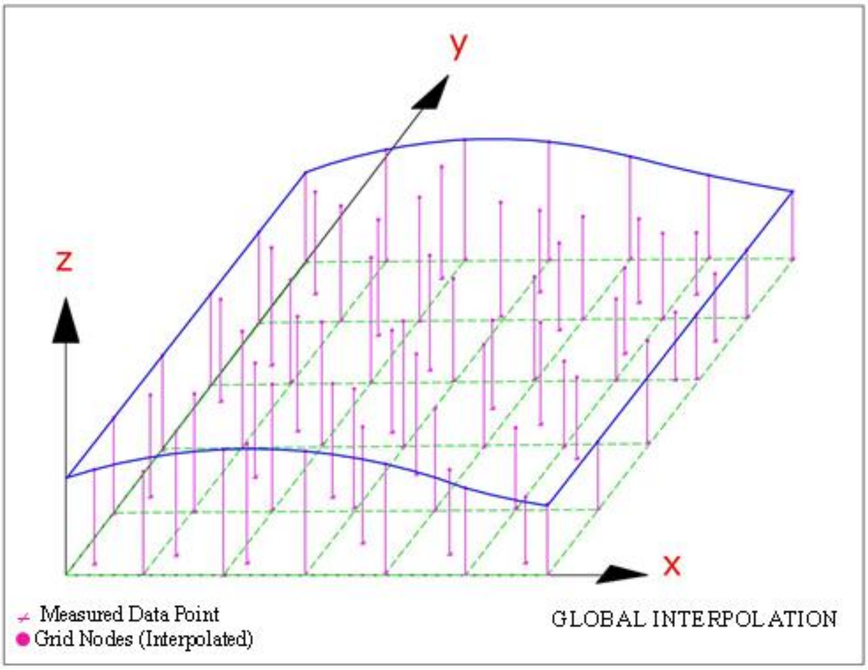


Figure 3.5 Global Interpolation

3.4.1.4 Polynomials used for surface representation

Polynomial equations are used to represent the terrain surfaces in the global and patch wise methods of interpolation. The basic general polynomial equation used is

$Z_i = a_o + a_1X_i + a_2Y_i + .,.$

as shown in Table 5.1

where Z_i is the height value of an individual point *i*

X_i , Y_i are the rectangular coordinates of the point *i*

a_o, a_1, a_2, etc., are the coefficients or parameters of the polynomial.

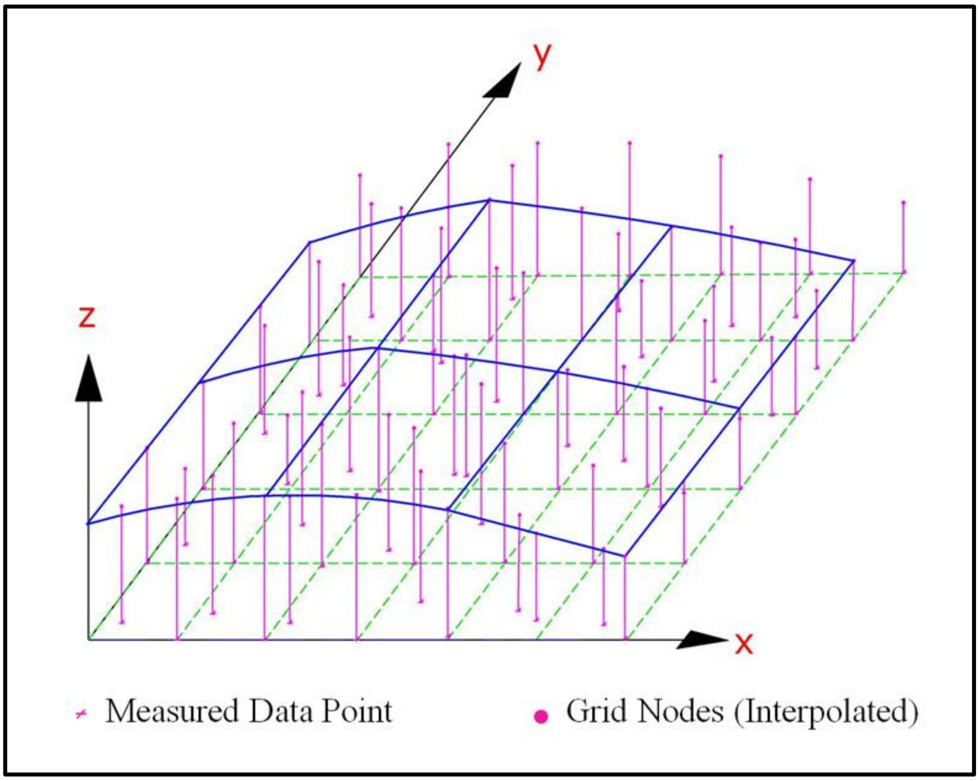


Figure (3.6a) Patch wise Interpolation (Exact Fit Patches)

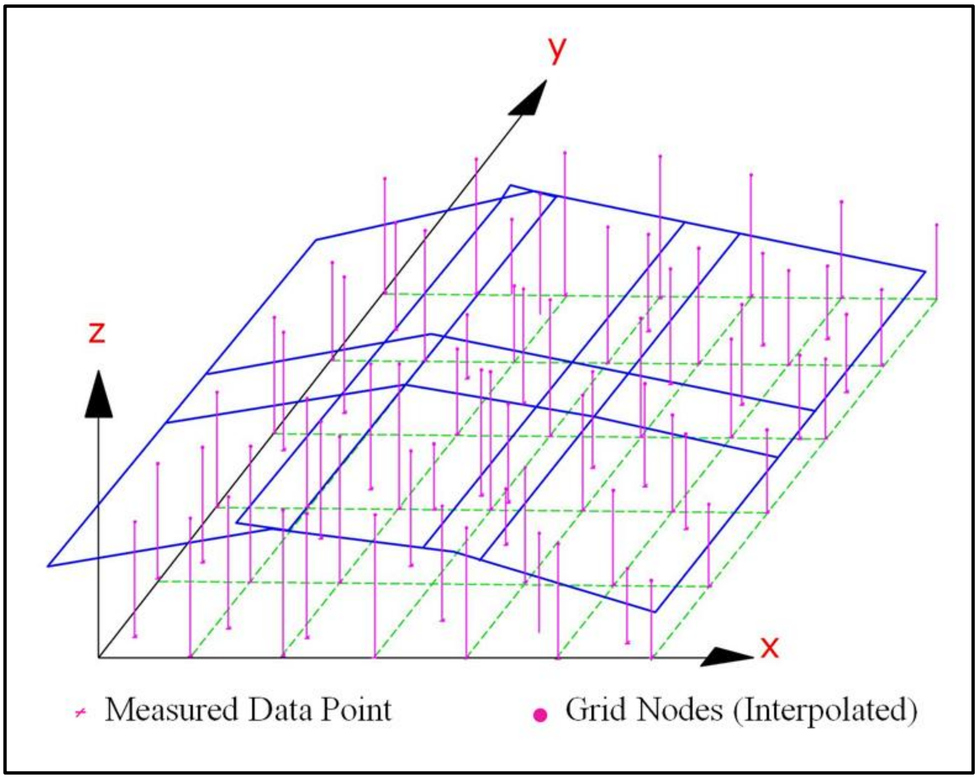


Figure (3.6b) Patch wise Interpolation(overlapping patches)

Individual terms	Order of term	Descriptive term	No. of terms
a_o	Zero	Planar	1
$+a_1X+a_2Y$	First	Linear	2
$+ a_3X^2+ a_4y^2+ a_5XY$	Second	Quadratic	3
$+ a_6X^3+ a_7y^3 + a_8X^2y + a_9Xy^2$	Third	Cubic	4
$+ a_{10}X^4+ a_{11}y^4 + a_{12}X^3 Y + a_{13}X^2y^2+ a_{14}Xy3$	Fourth	Quadratic	5
$+ a_{1S}X^s+$ ...etc	Fifth	Quintic	6

Table 3.1 Polynomial equation used for surface representation

One such equation will be generated for each individual point *i* with coordinates Xi, Y_i, Z_i, occurring in the terrain model. In the first step, the values of X, Y and Z are known for each measured point present in the overall data set or patch. Thus the values of the coefficients a_1, a_2, a_3··· can be determined from the set of simultaneous equations which have been set up, one for each data point. Once the values of the coefficientsa_1, a_2, a_3… have been determined, then for any given grid node point with known coordinates X, Y, the corresponding height value Z can be calculated.

To make a correct selection of the terms which will best represent or model the terrain surface, the surveying engineer must keep in mind the shape produced by each term in the polynomial equation (Fig. 3.7). Typical of the simpler types of surface used to model individual grid cells or Patches are:

(i) The 4-term bilinear polynomial

$$Z = a_o+ a_1X+ a_2Y+a_3XY \qquad \textbf{(3.2)}$$

(ii) The I0-term cubic polynomial

$$Z = a_o+ a_1X+a_2Y+ a_3XY+a_4X^2+ a_5Y^2+ a_6X^2Y+ a_7XY^2 +a_8X^3+ a_9Y^3 \qquad \textbf{(3.3)}$$

(iii) The 16-term bicubic polynomial

$$Z = a_o+ a_1X+a_2Y+ a_3XY+a_4X^2+ a_5Y^2+ a_6X^2Y+ a_7XY^2 +a_8X^2Y^2+ a_9X^3+ a_{10}Y^3$$

$$+ a_{11}X^3Y+ a_{12}XY^3+ a_{13}X^3Y^3+ a_{14}X^3Y^2+ a_{15}X^2Y^3 \qquad \textbf{(3.4)}$$

3.4.1.5 Contouring from grid data

For a single grid cell (Fig. 3.8), a simple linear interpolation is carried out along each of the four sides in turn, based on the values at the nodes. The positions of all the contour values are determined for each side.

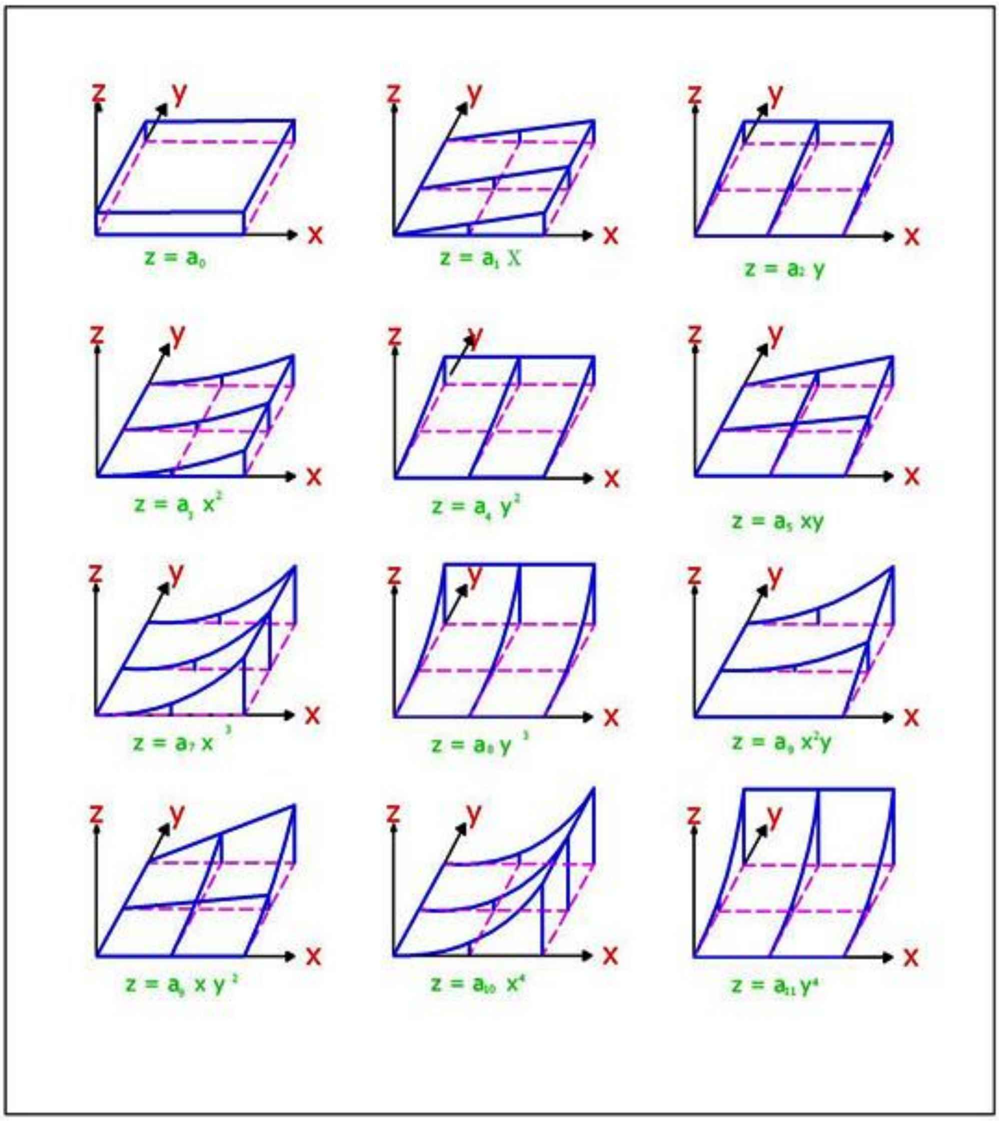
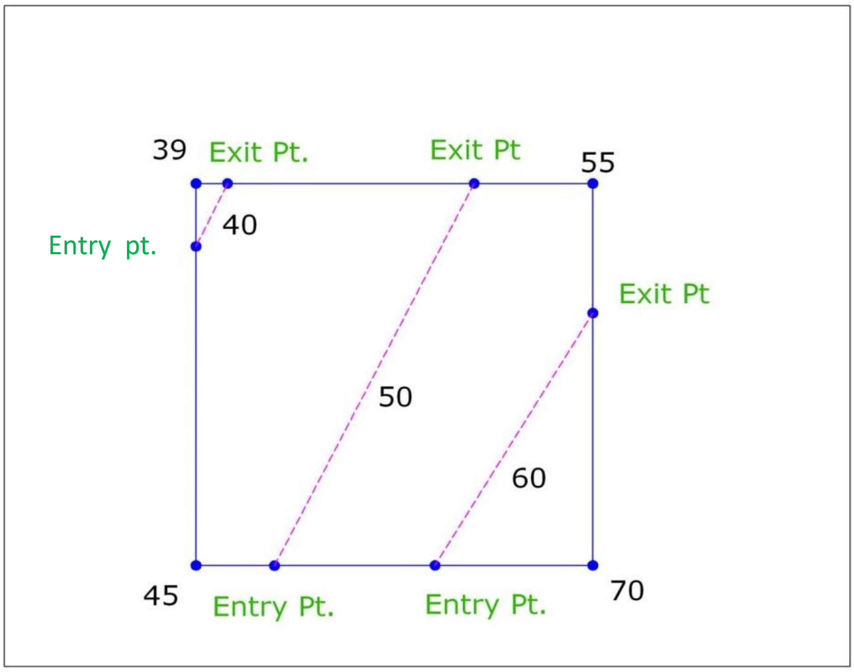


Fig (3.7) Surface shapes produced by individual terms in the general polynomial equation.

Taking the data points given in (Fig. 3.9), there are four possible solutions which give quite different positions for the contour and also a fifth (impossible) alternative.



.

Figure 3.8 Grid contouring: linear contour interpolation in a single cell.

3.4.2 Triangle-based terrain modelling

Is being used to an ever-increasing extent in terrain modeling,when every measured data point (vertices of the triangles) is used and honored directly, to model the terrain, from which the height of additional points may be determined by interpolation and the construction of contours undertaken.

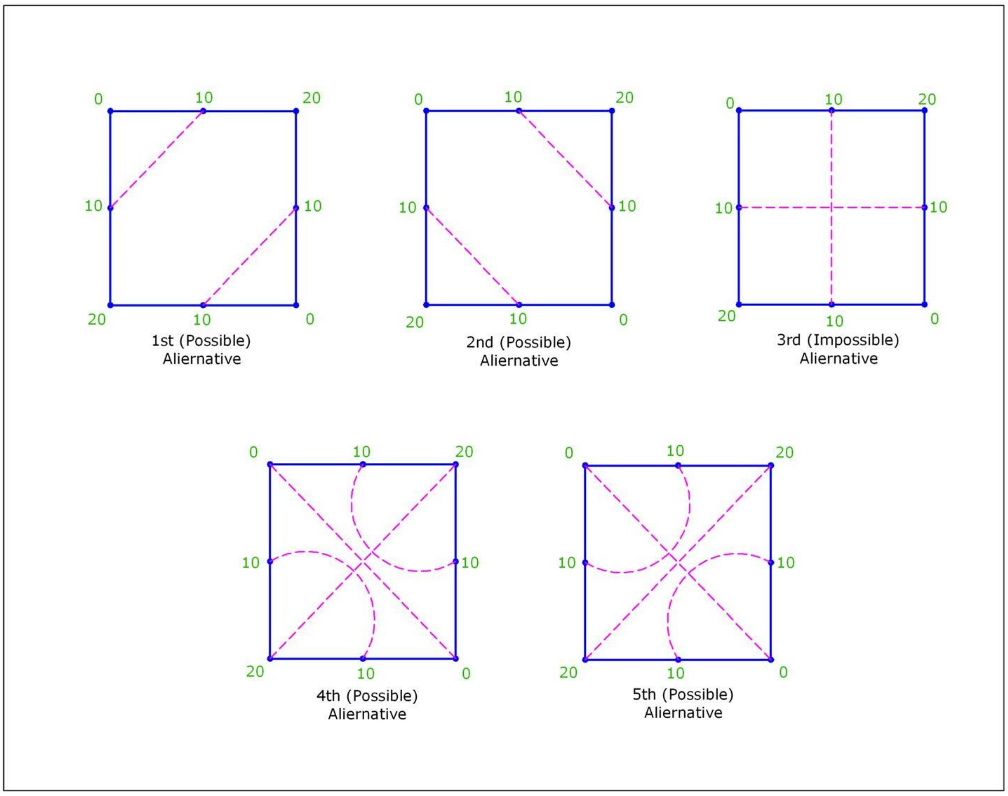


Figure 3.9 Grid contouring: ambiguity in contour threading in a single cell.

3.4.2.1 Contouring from triangulated data

As with contouring of regular gridded height points, so with randomly located triangulated height data, there are two main options for the contour threading.

(i) Simple linear interpolation of the contours.

(ii) Generation of curved smoothed contours using some type of function.

When the terrain model is based on triangulated data the use of direct linear interpolation for the contour generation gives a simple and robust solution. Ambiguities of directions can be resolved.

Digital terrain modeling is now a commonly used technique both in topographic mapping and civil engineering design. It is also used widely in other fields such as landscape planning, flight simulation and geological/geophysical exploration where generally the accuracy requirements of the elevation data are lower than for surveying and engineering applications .

CHAPTER FOUR

LEAST SQUARES COLLOCATION

4.1 Introduction:

The classical least squares adjustment theory of Gauss has been generalized and modified to include the theory of prediction and filtering of stochastic processes with stationary covariance signals. The generalized theory has been given the name “least squares collocation”. The mathematical definition of collocation is given by Noritz (1980) as “the determination of a function by fitting an analytical approximation to a given number of linear functions.

Historically least squares collocation is developed from least squares prediction of gravity anomalies . The technique is mainly used in surveying and geodesy to determine the values of quantities at points other than those at which measurements have been made (or at which information is not known).

An essential feature of the method is that quantities which are by nature deterministic are described in a statistical manner, particularly by the use of covariance matrices. we would need to establish a function (known as a covariance function) from which it would be possible to compute the covariance of the heights at any two points, to predict the unknown height of a point surrounded by a number of points of known height. This function would be in terms of quantities such as position and distance between the points.

4.2 Covariance matrices

Consider n points (Fig 4.1) at which we know the coordinate values of a quantity (X, Y, and H) u, i.e. we know u_1, u_2... u_n. To determine a covariance function. Assume that the correlation of the quantity between any two points i and j is a function only of the distance, d_{ij}, between them. Then using all n_1 pairs of points separated by a distance of up to r_1 meters we compute their covariance from

$$C_1 = \frac{1}{n_1}\sum u_i \, u_j \qquad (4.1)$$

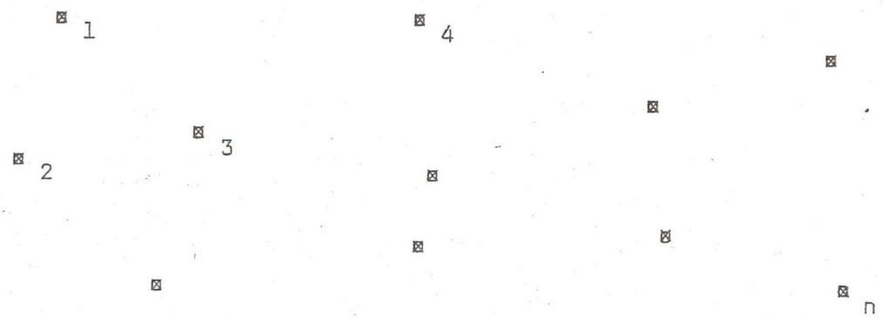


Fig 4.1 points with known heights

The process is then repeated using all n_2 pairs of points separated by a distance greater than r_1 and less than r_2 meters etc. Generally we can write, for the n_k pairs of points separated by a distance greater than r_{k-l} and less than r_k meters.

$$C_k = \frac{1}{n_k} \sum u_i \, u_j \qquad (4.2)$$

The covariance matrix for the n points is now written as

$$C_u = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix} \qquad (4.3)$$

Alternatively a mathematical function, e.g.

$$C_{ij} = a \exp(-b \, r_{ij}) \qquad (4.4)$$

Where a and b are constants, could be fitted to the data and subsequently used to compute each element of C_u.

We now extend the concept of the covariance matrix to cover the situation shown in Fig 4.2 where we have a quantity, u, known at points 1,2,3, etc.

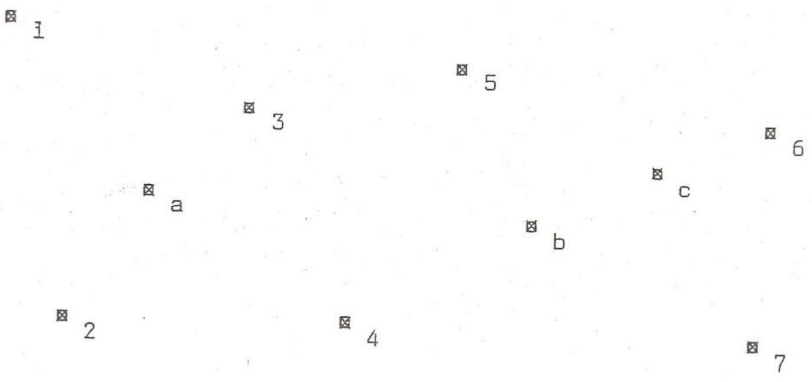


Fig 4.2 points for prediction of the unknown height

but unknown at points a, b, c. In this case we find it convenient to partition the complete vector u into two parts u_1 and u_2

$$u = \langle u_1 | u_2 \rangle^T \qquad (4.5)$$

Where u_1 contains the values of the quantity at points 1,2,3, etc. (called data points) and u_2 contains the values at points a,b,c, etc. (called computation points). Then the covariance matrix of u is correspondingly partitioned

$$C_u = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \qquad (4.6)$$

where the generally non—square matrices C_{12} and C_{21} (note that $C_{12} = C_{21}^T$) are often termed the Cross covariance matrices between the data and computation points.

4.3 Least squares prediction

As a preliminary to least squares collocation we will consider the simple case of least squares prediction. Referring to 4.1 and Fig.4.2, let u_1 be a vector of known quantities at points 1,2,3, etc. and let u_2 be the unknown values of the quantities at a, b, c, etc. Again it is emphasized that we are not here concerned with measurement errors, i.e. u_1 is perfectly known, but it is required to estimate u_2. Any linear estimates of u_2, say u_2^*, must be of the form

$$u_2^* = Qu_1 \qquad (4.7)$$

(PA Cross - 1983)

where Q is a linear transformation to be determined.

Let e* be the true error of the estimate u_2^*, then

$$e^* = u_2^* - u_2 \qquad (4.8)$$

and substituting 4.7 in 4.8 we have

$$e^* = Qu_1 - u_2 \qquad (4.9)$$

which can be rewritten as

$$e^* = \langle Q|-I\rangle \left[\frac{u_1}{u_2}\right] \qquad (4.10)$$

Then applying Gauss error propagation law

$$C_z = RC_yR^T \qquad (4.11)$$

to 4.10 and using 4.6 we obtain the covariance matrix of e^*

$$C_{e^*} = \langle Q|-I\rangle \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \left[\frac{Q^T}{-I}\right] \qquad (4.12)$$

and reorders to

$$\mathrm{C_{e^*} = C_{22} + QC_{11}Q^T - QC_{12} - C_{21}Q^T} \qquad (4.13)$$

Now $C_{21}C_{11}^{-1}C_{12}$ is subtracted and added to 4.13 to yield

$$C_{e^*} = C_{22} - C_{21}C_{11}^{-1}C_{12} + QC_{11}Q^T - QC_{12} - C_{21}Q^T + C_{21}C_{11}^{-1}C_{12} \qquad (4.14)$$

Since $C_{11}C_{11}^{-1} = I$ (4.14) can be written in the following expanded form

$$C_{e^*} = C_{22} - C_{21}C_{11}^{-1}C_{12} + QC_{11}Q^T - C_{11}C_{11}^{-1}C_{12}QC_{12} - C_{21}C_{11}C_{11}^{-1}Q^T + C_{21}C_{11}^{-1}C_{11}C_{11}^{-1}C_{12} \qquad (4.15)$$

which, after putting $C_{12} = C_{21}^T$, becomes

$$C_{e^*} = C_{22} - C_{21}C_{11}^{-1}C_{21}^T + (Q - C_{21}C_{11}^{-1})C_{11}(Q - C_{21}C_{11}^{-1})^T \qquad (4.16)$$

We can write (4.16) as a sum of two matrices viz.

$$C_{e^*} = F + G \qquad (4.17)$$

Where

$$F = C_{22} - C_{21}C_{11}^{-1}C_{21}^T \qquad (4.18)$$

and

$$G = (Q - C_{21}C_{11}^{-1})C_{11}(Q - C_{21}C_{11}^{-1})^T \quad (4.19)$$

In fact any choice of Q will yield a matrix G with none—negative diagonal elements.

This is because the ith diagonal element of C in (4.19) is given by the quadratic form

$$G_{ii} = gC_{11}g^T \qquad (4.20)$$

where g is the ith row of $Q - C_{21}C_{11}^{-1}$ and C$_{11}$ is positive—definite. Hence any choice of Q will make the variances of the error in each element of u_2^* equal to or larger than the diagonal elements of F. The minimum variance estimate will therefore be obtained when G is a null matrix, i.e.

$$Q - C_{21}C_{11}^{-1} = 0 \qquad (4.21)$$

or

$$Q = C_{21}C_{11}^{-1} \qquad (4.22)$$

Substituting (4.22) in (4.7) gives the best (in the sense of minimum variance) linear estimate of u_2 as

$$\hat{u}_2 = C_{21}C_{11}^{-1}u_1 \qquad (4.23)$$

Which, because of its minimum variance property, is also termed the least squares estimate. (4.23) is often written in the following manner for the prediction of u at any particular computation point p:

$$\hat{u}_p = [C_{p1}C_{p2} \dots C_{pn}] \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \qquad (4.24)$$

Where c_{pi} represents the vector of covariance between point p and the ith data point and all other symbols are as previously defined, i.e. u_1, u_2, ..., u_n are the values of the quantity at the data points and the square matrix to be inverted is the covariance matrix of the quantities at the data points.

4.4 Collocation Mathematical Model

The model associated with ordinary least squares adjustment by parameters is the general collocation model. This model is able to take into account measurement errors at the data points and the possible requirement to compute certain parameters during the prediction process.

Consider a set of data points at which we have made n observations. Let there also be q computation points and m parameters to be recovered. As usual we will denote the true values of the modeled observed quantities and the parameters by the vectors $\bar{\ell}$ and $\bar{x}$ respectively. We can write down n, generally non—linear, observation equations of the form.

$$F(\bar{x}) - \bar{\ell} = 0 \tag{4.25}$$

and

$$\ell = \bar{\ell} + e \tag{4.26}$$

where e is the total “error” in the observations (i.e. the difference between the observed and modeled quantities). In collocation this total error is considered to be the sum of two independent errors usually called the signal and noise, and denoted by the symbols S_1 and n respectively. Hence (4.26) is written as

$$\ell = \bar{\ell} + S_1 + n \tag{4.27}$$

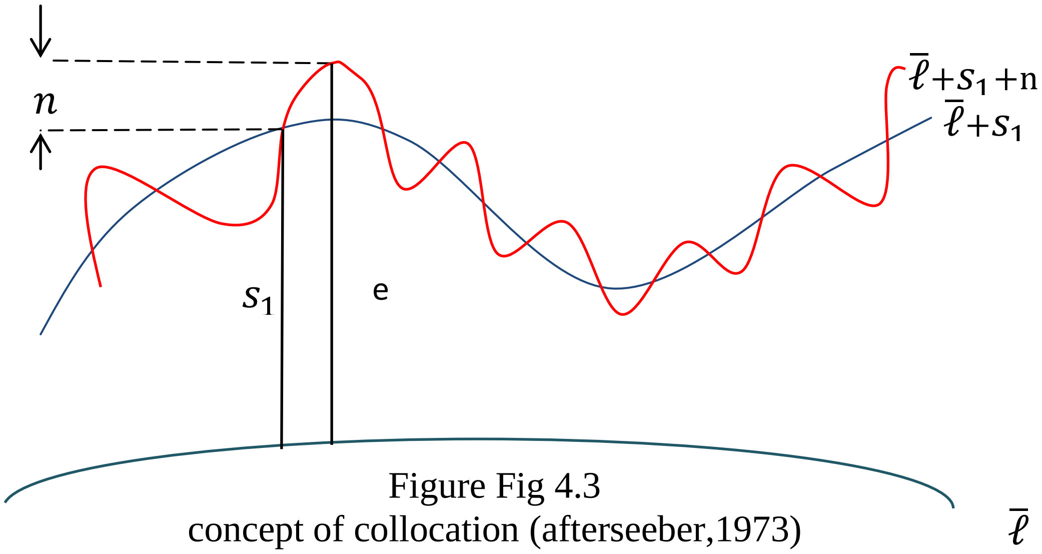


Figure Fig 4.3
concept of collocation (afterseeber,1973)

After linearising (4.25) using (4.27) we obtain

$$Ax — b + s_1 + n = 0 \tag{4.28}$$

4.4.1: The Signal

The signal represents the inability of the model to describe the exact relationship between the measurements ℓ and the unknown parameters $\bar{\ell}$. In addition, the signal may be considered to be external to the instrument and related to the behavior of the observable in a particular medium (Krakiwsky, 1975).

The signal may vary continuously and exists at points other than the measuring points, thus interpolation is possible (Moritz, 1972). Another property of the signal is that any one of its values is of an unpredictable, arbitrary nature. Thus the signal may be considered to be stochastic (Za'voti, 1977).

The signal quantities are statistically dependent by nature; that is the signal is characterized by a full covariance matrix in the domain defined by the observation and computation points. Thus; in collocation it is essential that the signal has known second moments (variance—covariance matrix), although the first moments (values of the signal) remain as unknowns to be determined. The vector s_1 is a random vector whose expectation or mean value, is zero

$$E(s) = 0 \qquad (4.29)$$

4.4.2: The Noise

In collocation terms, the measurement ℓ consists of a systematic part, $\bar{\ell}$, and two random parts, s_1 and n. The noise is likely to resemble a measuring error, and is internal to the instrument. The elements of the noise vector n are considered discrete values for each observation while the elements of S must be regarded as realizations of a continuous random quantity at the measuring points. Moreover, the measurement errors are assumed to be statistically independent from S and s_1 as they are peculiar to the measuring instrument (Krakiwsky, 1975).

The vector n is random vector whose expectation or mean value, is zero

$$E(n) = 0 \qquad (4.30)$$

The problem of collocation is now to estimate simultaneously the following:

(i) the parameters x

(ii) The signal s_1 and noise n at the data points.

(iii) The signal s_2 at the computation points. To apply the least squares collocation we have to know the covariance matrices of both the signal and the noise. The covariance matrix for the noise, c1 is obtained in the usual way (equivalent to C_ℓ) and Cs, the signal covariance matrix for both the data and computation points, by a study (equation (4.1)) of the variation of the signal where it is known or can be estimated. Cn will often be diagonal but Cs will invariably be a full matrix as the whole point of differentiating between the signal and the noise is that the signal is highly spatially (or possibly temporally) correlated and has completely different statistical properties to the noise. The derivation of the collocation equations now proceeds as follows (PA Cross-1983).

Let s be a vector containing the signal at both the data and computation points, i.e.

$$s = [s_1|s_2]^T \qquad (4.31)$$

Then (4.28) can be rewritten as

$$Ax — b + Bs + n \;=\; 0 \qquad (4.32)$$

with

$$B = [I|0] \qquad (4.33)$$

Note that if we have q computation points then, B will have dimensions (n + q) x n with I, a unit matrix, being n x n and 0, a null matrix, being q×n.

We now wish to estimate x, s and n in (4.32) using the method of least squares,i.e. minimizing

$$s^T C_s^{-1} s + n^T C_n^{-1} n$$

Hence, using Lagrange's method of undetermined multipliers as in the following equation

$$\Phi = v^T wv + 2k^T (Ax + cv - b) \qquad (4.34)$$

we have

$$\Phi = s^T C_s^{-1} s + n^T C_n^{-1} n + 2k^T (Ax - b + Bs + n) \qquad (4.35)$$

which is minimized by differentiating Φ with respect to the unknowns and equation to zero as follows

$$\frac{\partial \Phi}{\partial x} = 2A^T \hat{k} = 0 \qquad (4.36)$$

$$\frac{\partial\Phi}{\partial s} = 2C_s^{-1}\hat{s} + 2B^T\hat{k} = 0 \qquad (4.37)$$

$$\frac{\partial\Phi}{\partial \mathrm{n}} = 2\mathrm{C}_\mathrm{n}^{-1}\hat{\mathrm{n}} + 2\hat{\mathrm{k}} = 0 \qquad (4.38)$$

Also the least squares estimates must satisfy (4.32),i.e.

$$A\hat{x} - b + B\hat{s} + \hat{n} = 0 \qquad (4.39)$$

Now after dividing (4.36) to (4.38) by 2 and combining then with (4.39) obtain the following least squares through hyper matrix.

$$\begin{bmatrix} C_n^{-1} & 0 & I & 0 \\ 0 & C_n^{-1} & B^T & 0 \\ I & B & 0 & A \\ 0 & 0 & A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{n} \\ \hat{s} \\ \hat{k} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \\ 0 \end{bmatrix} \qquad (4.40)$$

which, can be written as

$$\begin{aligned} C_n^{-1}\hat{n} \ + \ \hat{k} \ &= 0 \\ \hat{n} + B\hat{s} + \hat{n} + A\hat{x} \ &= b \qquad (4.41) \\ C_s^{-1}\hat{s} + B^T\hat{k} &= 0 \\ A^T\hat{k} &= 0 \end{aligned}$$

The first two equations can be used to eliminate $\hat{n}$ as follows:-

$$\hat{n} \ = \ -c_n \ \hat{k}$$

which when substituted in the second equation yields

$$-c_n \ \hat{k} + B\hat{s} + A\hat{x} \ = b$$

This equation and the last two equations of (4.41) can be written as

$$\begin{bmatrix} C_n^{-1} & B^T & 0 \\ B & -c_n & A \\ 0 & A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{s} \\ \hat{k} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \tag{4.42}$$

from (4.42) we can write the following equations

$$C_n^{-1} + B^T\hat{k} = 0$$

$$B\hat{s} - c_n\,\hat{k} + A\hat{x} = b$$

$$A^T\hat{k} = 0$$

From the first of the above three equations it is not difficult to see that

$\hat{s} = -c_n\, B^T\hat{k}$

Which when substituted in the second equation yields

$$-Bc_n\, B^T\hat{k} - c_n\,\hat{k} + A\hat{x} = 0$$

And when combined with the third equation, the result could be written as:-

$$\begin{bmatrix} -(C_n + BC_sB^T) & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{k} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{4.43}$$

Substituting equation (4.33) into (4.43) gives

$$\begin{bmatrix} [-C_n - [\mathrm{I} \quad 0]] \begin{bmatrix} C_{S_1} & C_{S_1S_2} \\ C_{S_2S_1} & C_{S_2} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{k} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{4.44}$$

$$[-C_n - [1 \quad 0]] \begin{bmatrix} C_{S_1} & C_{S_1S_2} \\ C_{S_2S_1} & C_{S_2} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{4.45}$$

where C_{S_1} and C_{S_2} are the variance covariance matrices of the signal at the data computation points respectively and $C_{S_1S_2}$ and $C_{S_2S_1}$ are their cross-covariance matrices.

or

$$\begin{bmatrix} -(C_n + C_{s_1}) & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{k} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \qquad (4.46)$$

4.4.3: Solution for the Parameters

From (4.46) we can write:

$-(C_n + C_{s_1})\,\hat{k} + A\hat{x} = b$ and

$A^T\hat{k} = 0$

From the first equation we obtain

$$\hat{k} = \left(C_n + C_{S_1}\right)^{-1} A\hat{x} - (C_n + C_{s_1}) \qquad (4.47)$$

Substituting this expression for $\hat{k}$ into the second equation we obtain

$$A^T\left(C_n + C_{S_1}\right)^{-1} A\,\hat{x} - A^T\left(C_n + C_{S_1}\right)^{-1} b = 0$$

from which

$$\hat{x} = \left(A^T\left(C_n + C_{S_1}\right)^{-1} A\right)^{-1} A^T\left(C_n + C_{s_1}\right)^{-1} b \qquad (4.48)$$

Substituting (4.48) into (4.47) results in

$$\hat{k} = -\left(C_n + C_{s_1}\right)^{-1}(b - A\hat{x}) \qquad (4.49)$$

Then substituting (4.49) and rearranging leads to

From

$$\hat{s} = -c_n\, B^T\hat{k}$$

developed, and substituting in (4.49) we get

$$\hat{S} = C_s B^T \left(C_n + C_{s_1}\right)^{-1} (b - A\hat{x}) \qquad (4.50)$$

which is the least squares collocation expression for the signal at both the data and computation points. The noise at the data points is obtained by substituting 4.49 yield

from equation (4.41) ,

$$\hat{n} \ = \ -c_n \ \hat{k}$$

And when equation $\hat{k}$ from (4.49) is substituted, we get

$$\hat{n} = C_n \left(C_n + C_{s_1}\right)^{-1} (b - A\hat{x}) \qquad (4.51)$$

To derive the corresponding covariance matrices we need an expression for the covariance matrix of the vector b.

From

$$\underset{(\mathrm{r} \times 1)}{-\mathrm{b}} = \begin{bmatrix} f_1(x^0, t) \\ f_2(x^0, t) \\ . \\ . \\ . \\ . \\ . \\ . \\ f_r(x^0, t) \end{bmatrix} \qquad (4.52)$$

in the special case of observation equations

$$b = F(x^o) - \ell \qquad (4.53)$$

substituting 4.27 gives

$$b = F(x^o) - \bar{\ell} - n - s_1 \qquad (4.54)$$

Then applying (4.11) to (4.54), whilst noting that F(x°) and $\bar{\ell}$ are not stochastic and that we have already assumed n and s_1 to be independent, we have

$$C_b = C_n + C_{S_1} \qquad (4.55)$$

substitute (4.55) in (4.48) to have

$$\hat{x} = [(A^T C_b^{-1} A)^{-1} A^T C_b^{-1}] b \qquad (4.56)$$

Application of (4.11) to (4.56) gives the following expression for the covariance matrix of the parameters:

$$C_{\hat{x}} = [(A^T C_b^{-1} A)^{-1} A^T C_b^{-1}] C_b [(A^T C_b^{-1} A)^{-1} A^T C_b^{-1}]^T \qquad (4.57)$$

which can be simplified to

$$C_{\hat{x}} = (A^T C_b^{-1} A)^{-1} \qquad (4.58)$$

$$= \left[A^T \left(C_n + C_{s_1}\right)^{-1} A\right]^{-1} \quad \text{(from 4.55))} \qquad (4.59)$$

For the covariance matrix of the least squares estimates of the signal, at both the computation and data points, substitute (4.55) and (4.56) in (4.50) to obtain

$$\hat{S} = \{C_S B^T C_b^{-1} [I - A(A^T C_b^{-1} A)^{-1} A^T C_b^{-1}]\} b \qquad (4.60)$$

Then substituting (4.58) and applying (4.11) to (4.60) leads to

$$\begin{aligned} C_{\hat{s}} = \{C_S B^T C_b^{-1} [I \\ - A C_{\bar{x}} A^T C_b^{-1}]\} C_b \{C_s B^T C_b^{-1} [I \\ - A C_{\hat{x}} A^T C_b^{-1}]\}^T \end{aligned} \qquad (4.61)$$

which can be simplified to

$$C_{\hat{s}} = C_S B^T C_b^{-1} B C_s - C_S B^T C_b^{-1} A C_{\hat{x}} A^T C_b^{-1} B C_s \qquad (4.62)$$

For the signal covariance matrix substitute (4.55) and (4.58) in (4.62) to obtain

$$\begin{aligned} C_{\hat{s}} = C_S B^T \left(C_n + C_{s_1}\right)^{-1} B C_s \\ - C_S B^T \left(C_n + C_{s_1}\right)^{-1} A \left[A^T \left(C_n + C_{s_1}\right)^{-1} A\right]^{-1} A^T \left(C_n \right. \\ \left. + C_{s_1}\right)^{-1} B C_s \end{aligned} \qquad (4.63)$$

A similar treatment of (4.51) leads to the covariance matrix for the least squares estimates of the noise at the data points

$$C_{\hat{n}} = C_n\left(C_n + C_{s_1}\right)^{-1}C_n - C_n\left(C_n + C_{s_1}\right)^{-1}A\left[A^T\left(C_n + C_{s_1}\right)^{-1}A\right]^{-1}A^T\left(C_n + C_{s_1}\right)^{-1}C_n \qquad (4.64)$$

Equations (4.48), (4.50), (4.51), (4.59), (4.63) and (4.64) are the working formulae for least squares collocation.

4.4.4 Special cases

The following three special cases can be identified.

(i) Collocation without parameters

In cases when the elements of the matrix confidents are equal to zero. A = 0 and 4.50, 4.63, 4.51 and 4.64 simplify to

$$\hat{S} = C_s B^T\left(C_n + C_{s_1}\right)^{-1}b \qquad (4.65)$$

with

$$C_{\hat{S}} = C_s B^T\left(C_n + C_{s_1}\right)^{-1}BC_s \qquad (4.66)$$

and

$$\hat{n} = C_n\left(C_n + C_{s_1}\right)^{-1}b \qquad (4.67)$$

with

$$C_{\hat{n}} = C_n\left(C_n + C_{s_1}\right)^{-1}C_n \qquad (4.68)$$

(ii) Collocation without parameters and without noise

When random errors are insignificant or the statistics of the observations are unknown, the least squares collocation equations for the signal (4.50) and (4.63) can be simplified to

$$\hat{S} = C_s B^T {C_{s_1}}^{-1}b \qquad (4.69)$$

with

$$C_{\hat{s}} = C_s B^T {C_{s_1}}^{-1} B C_s \tag{4.70}$$

(iii) Least squares prediction

When the observed (without noise) quantities and the signal are the same, using the notation of (4.2), we have

$$s = [u_1 | u_2]^T \tag{4.71}$$

$$b = u_1 \tag{4.72}$$

$$C_{s_1} = C_{u_1 s} \tag{4.73}$$

and

$$C_s = \begin{bmatrix} C_{s_1} & C_{s_1 s_2} \\ C_{s_2 s_1} & C_{s_2} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{4.74}$$

and, using (4.33) and (4.74), gives

$$C_s B^T = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \left[\frac{I}{0}\right] \tag{4.75}$$

$$= \begin{bmatrix} C_{11} \\ C_{21} \end{bmatrix} = [C_{11} | C_{21}]^T \tag{4.76}$$

Substituting (4.76) in (4.69) to obtain

$$\hat{S} = [C_{11} | C_{21}]^T C_{11}^{-1} b \tag{4.77}$$

$$= \left[\frac{b}{C_{21} C_{11}^{-1} b}\right] \tag{4.78}$$

Finally, substituting (4.71) and (4.72) in the left and right hand sides respectively of (4.78), we have

$$\left[\frac{\hat{u}_1}{\hat{u}_2}\right] = \left[\frac{u_1}{C_{21} C_{11}^{-1} u_1}\right] \tag{4.79}$$

i.e.

$$\hat{u}_1 = u_1 \tag{4.80}$$

and

$$\hat{u}_2 = C_{21} C_{11}^{-1} u_1 \tag{4.81}$$

with (4.81) being identical to (4.23). Hence we have shown least squares prediction to be a special case of least squares collocation.

CHAPTER FIVE

THE MODELING PROGRAM

5.1 Study area (Sudan)

In river Nile state, on the left bank of the River Nile and to the north west of Shandi Town. The study area is bounded by the following coordinates shown in table (5.1)

Between Latitude	N16˚ 35' 49"	Latitude	N17˚ 00' 14"
And Longitude	E32˚ 45' 55"	Longitude	E33˚ 14' 06"
Approximate area	100000 hectares		

Table 5.1-Study area coordinates

The maps below show the borders of the study area produced on the basis of the above given coordinates (See figures 5.1&5.2).

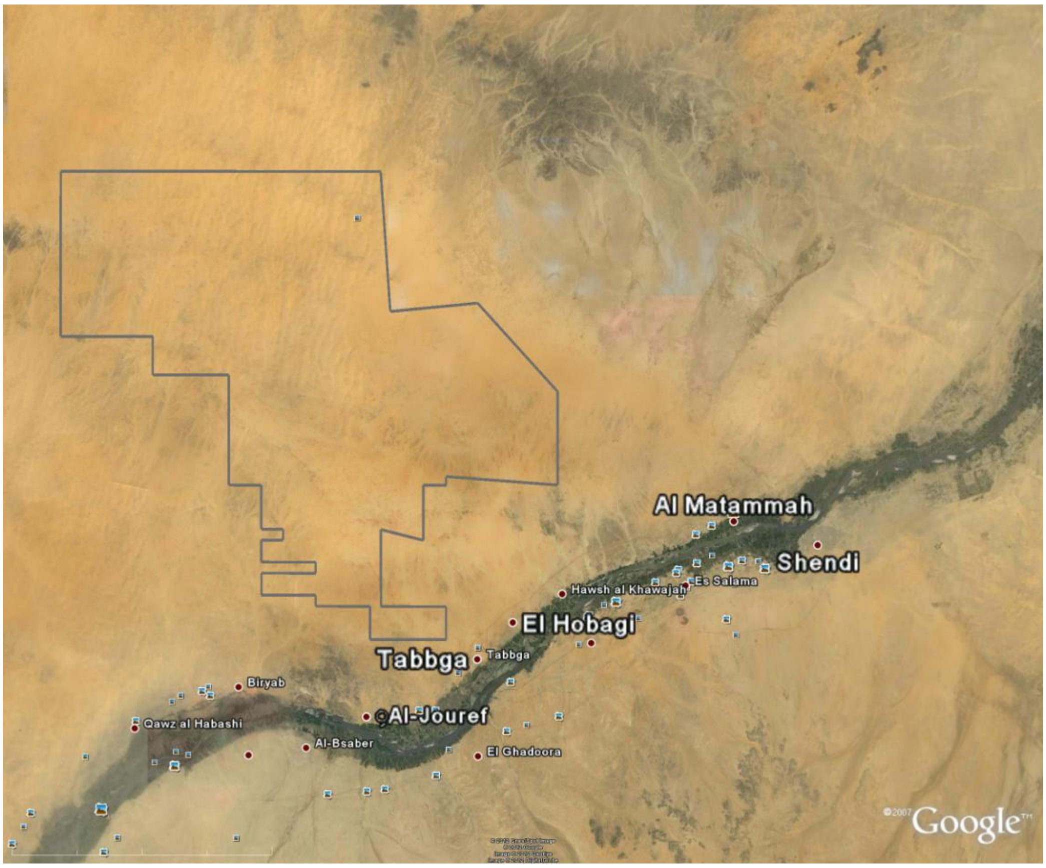


Figure 5.1(Study area location By Google)

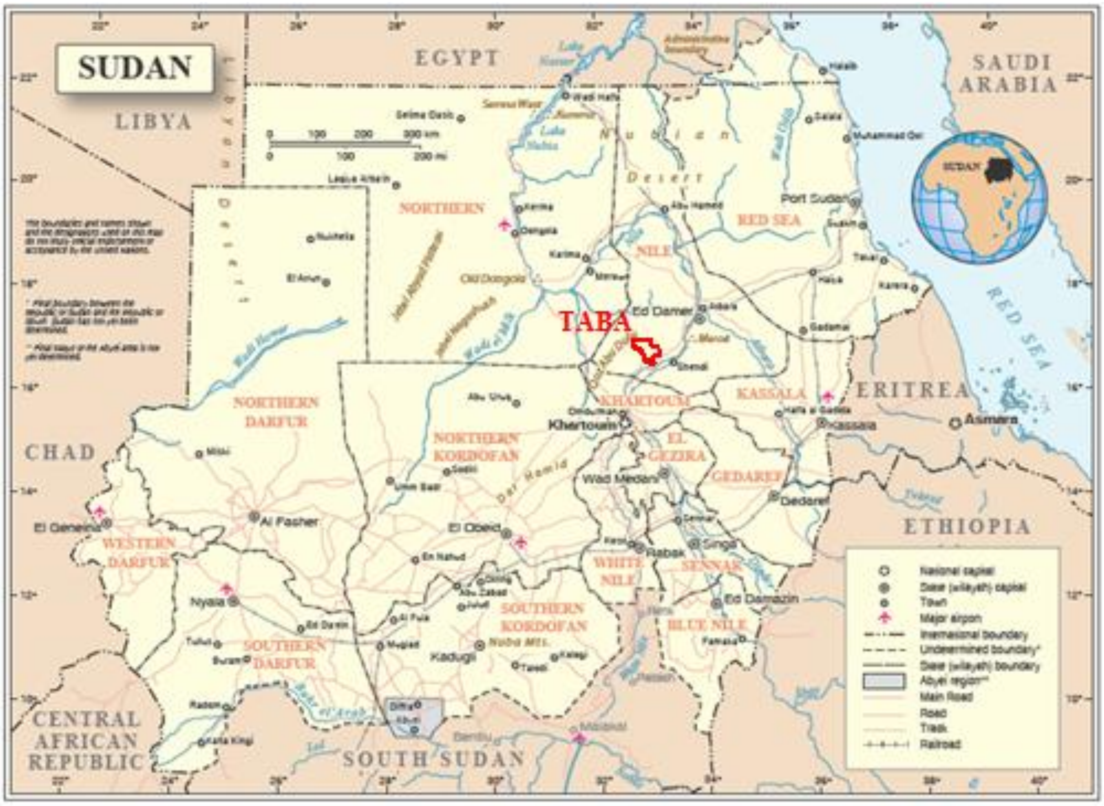


Figure 5.2 (Study area location - Map of Sudan (National Surveying Authority(NSA)))

5.2 Data collection for control work

- Establishing and observing new control points to be distributed with maximum distance of 4 kilometers between each two points within the study area, and observed by GPS and an Automatic level.
- RTK technique (Real Time Kinematic GPS) to carry out the detail survey and spot heights in grid lines which shall be at approximately 200X50m interval with more dense measurements on topographic changes (e.g. water courses, mountains etc...) for the study area and create a contour with the appropriate software. This is to be used later to check the developed mathematical modeling programs.

5.3 Instruments used and methodology

The filed surveying activities were carried out according to the following procedure and methodology:

The field work was started, by carrying out reconnaissance surveying within the study area in order for the surveying engineers to be acquainted with the study area environment and its physical and man-made surroundings (villages, roads, land cover, reference control points)

A cross check for the study area boundaries before commencing the topographical survey works was made.

The surveying works commenced by establishing 66 new control points {K1, K2, …, K66} in grid of 4Km×4Km covering 5 loops of 12KmX12K as shown in the diagram 1(Fig.5.3).

The new control points (K1-K66) coordinates, were determined using GPS receivers in static mode with a minimum 1 hour observations for control points within 12Km distance apart and other control points were measured by

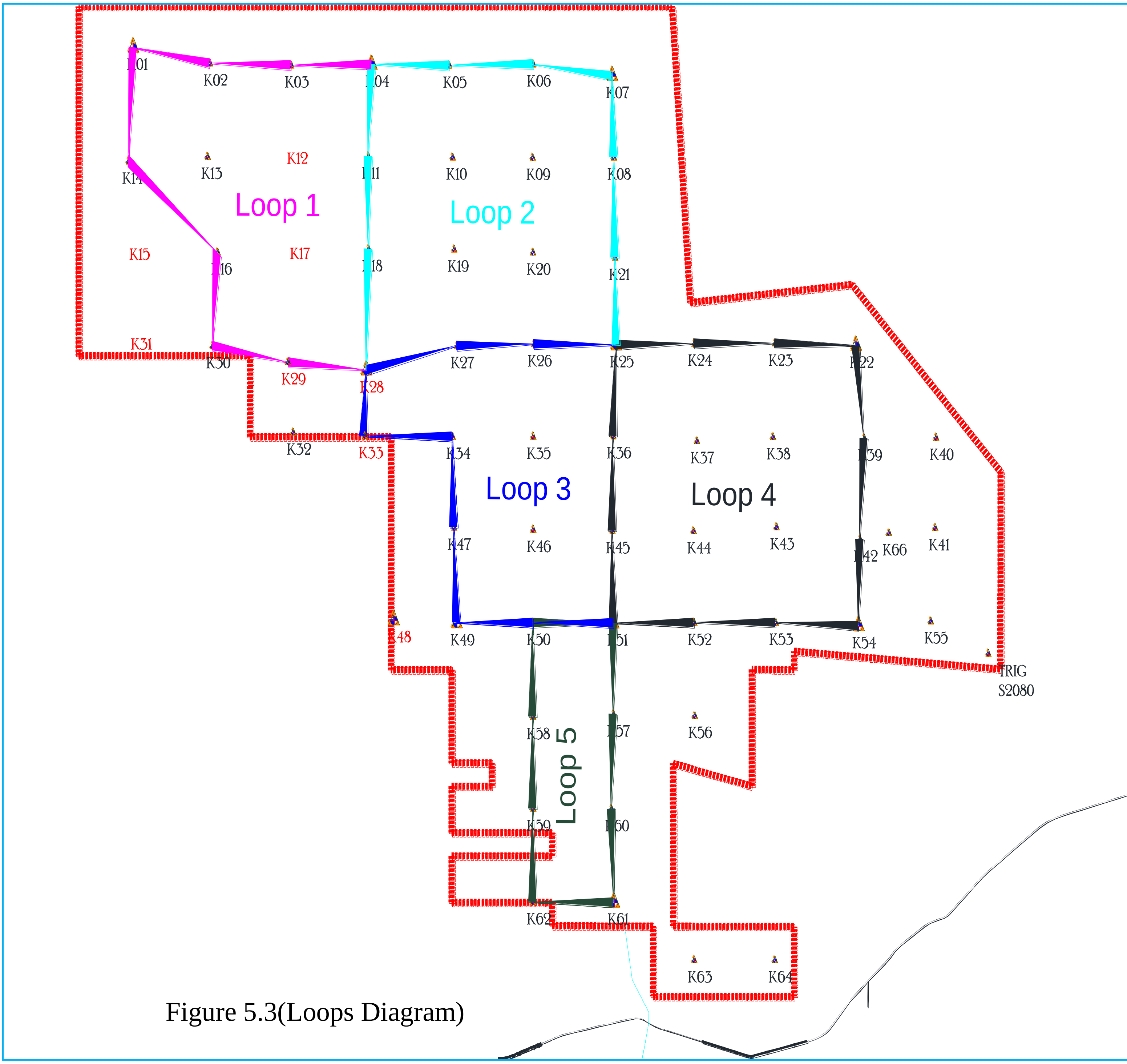


Figure 5.3(Loops Diagram)

observation points technique in RTK mode. These GPS observations were tied to the given reference control point S2080 {Second order geodetic, point, established by the National Surveying, Authority}.

The coordinates determined by GPS methods, were in the World Geodetic System 1984 (WGS84), and were transformed to the Universal Transverse Mercator (UTM) coordinates system. Also, the ellipsoid height was converted to orthometric height {relative to mean sea level} by Earth Gravitational Model 2008 {EGM 2008}. The site calibration method was accomplished by using points having heights determined by automatic levels.

Double run leveling was carried out between any two consecutive points as shown in (Figure 5.3); thus the major leveling loop (5 in number) have dimensions 12Km×12 Km. and the dimensions of the other smaller ones are not precisely measured. Therefore, the loop closures were checked by two techniques:

1- Double Run Method.
2- Close to Known points.

5.4 The programming language

A program was written in C# (pronounced “See Sharp”) is a simple, modern, object-oriented, and type-safe programming language.

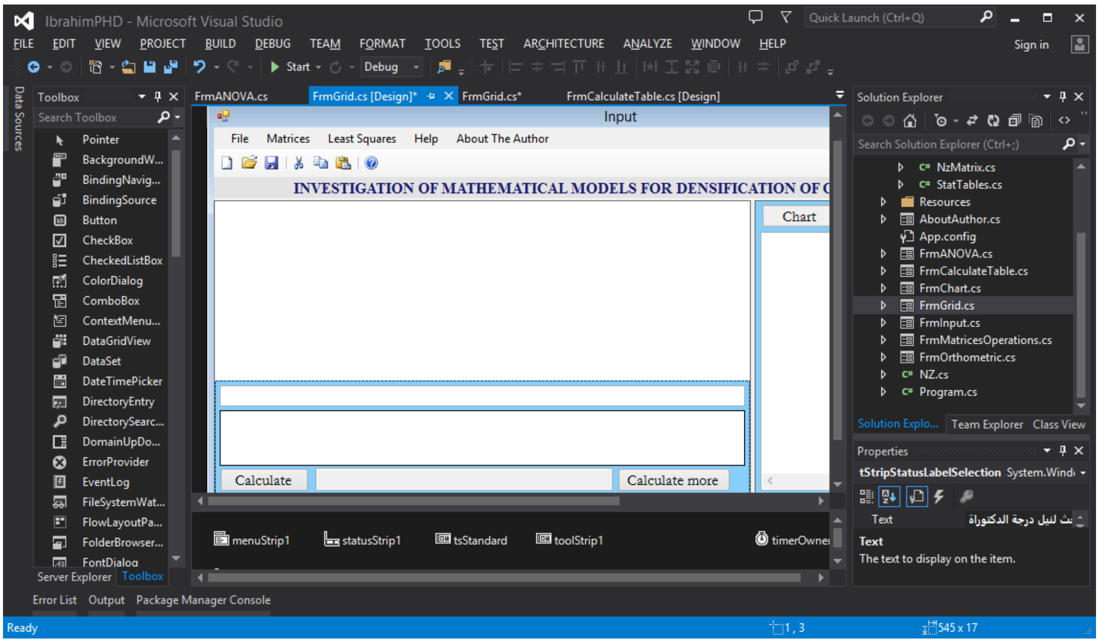
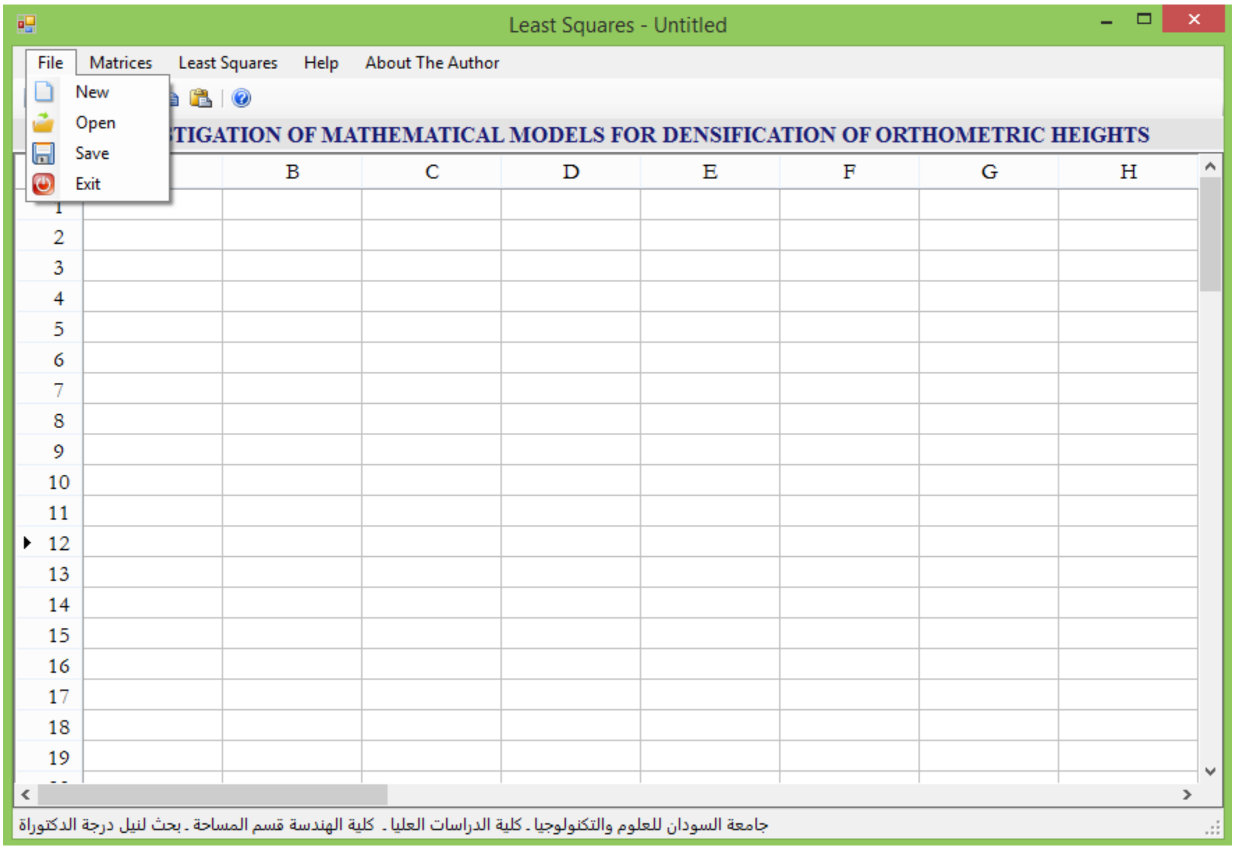


Figure 5.4-Program main menu screen

5.4.1 Back ground

- During the .NET Framework development, a managed code compiler system called Simple Managed C (SMC) used to write the original class libraries. In January 1999, Anders Hejlsberg formed a team to build a new language at the time called C-like Object Oriented Language (COOl).
- The .NET project was publicly announced in July 2000 and during the Professional Developers Conference, the language had been renamed C#, and the class libraries and ASP.NET runtime had been ported to C#.



جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

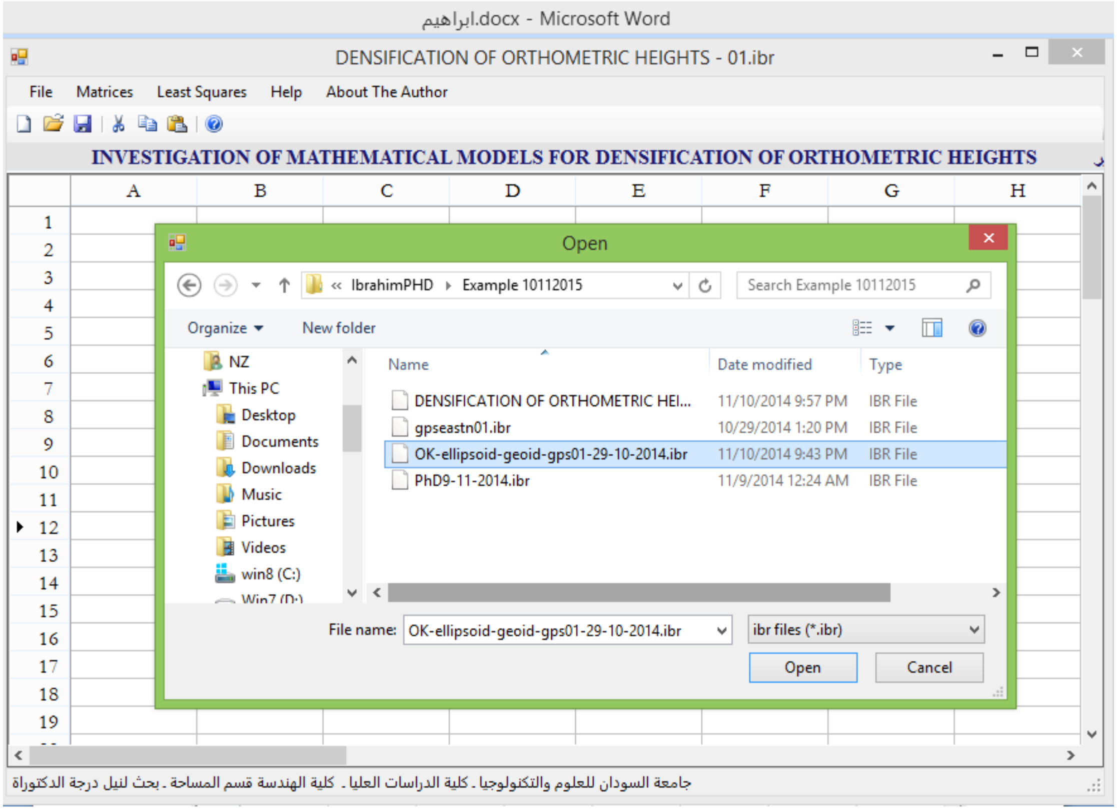
Figure 5.5 –The main menu and file pull-down menu.

C#'s designer is Anders Hejlsberg, who was previously involved with the design of Turbo Pascal, Embarcadero Delphi (formerly Code Gear Delphi, Inprise Delphi and Borland Delphi), and Visual J++.

C# makes use of reification to provide "first-class" generic objects that can be used like any other class, with code generation performed at class-load time. ((C-sharp-(Programming-language)# Cite-note-25)

Furthermore, C# has added several major features to accommodate functional-style programming, culminating in the Language integrated Query (LINQ) extensions released with C# 3.0 and its supporting framework of lambda expressions, extension methods, and anonymous types. These features enable C# programmers to use functional programming techniques, such as closures, when it is advantageous to their application.

The LINQ extensions and the functional imports help developers reduce the amount of "boilerplate" code that is included in common tasks like querying a database, parsing an xml file, or searching through a data structure, shifting the emphasis onto the actual program logic to help improve readability and maintainability.((C-sharp-(Programming-language)# Cite-note-28)


ابراهيم.docx - Microsoft Word
DENSIFICATION OF ORTHOMETRIC HEIGHTS - 01.ibr
File
Matrices
Least Squares
Help
About The Author
INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS
A
B
C
D
E
F
G
H
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Open
« IbrahimPHD › Example 10112015
Search Example 10112015
Organize
New folder
NZ
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
win8 (C:)
Win7 (D:)
Open
Cancel
جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

Name	Date modified	Type
DENSIFICATION OF ORTHOMETRIC HEI...	11/10/2014 9:57 PM	IBR File
gpseastn01.ibr	10/29/2014 1:20 PM	IBR File
OK-ellipsoid-geoid-gps01-29-10-2014.ibr	11/10/2014 9:43 PM	IBR File
PhD9-11-2014.ibr	11/9/2014 12:24 AM	IBR File

File name: OK-ellipsoid-geoid-gps01-29-10-2014.ibr

ibr files (*.ibr)

Figure 5.6 Drive and Geoid –Ellipsoid File selection

5.4.2 Design goals

The ECMA standard lists these design goals for C#

- The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.
- The language is intended for use in developing software components suitable for deployment in distributed environments.
- Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
- Support for internationalization is very important.
- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.
- Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language.

An exciting way to create versatile distributed applications. Using the same simple syntax regardless of the language used to create a Web service or the system on which it resides. For more advanced capabilities, you can also create Windows Communication Foundation (WCF) services.

Any of these types may also require some form of database access, which can be achieved using the Active Data Objects .NET (ADO.NET) section of the .NET Framework, through the ADO.NET Entity Framework, or through the Language Integrated Query (LINQ) capabilities of C#. Many other resources can be drawn on, such as tools for creating networking components, outputting graphics, performing complex mathematical tasks, and so on

5.5 About the Program

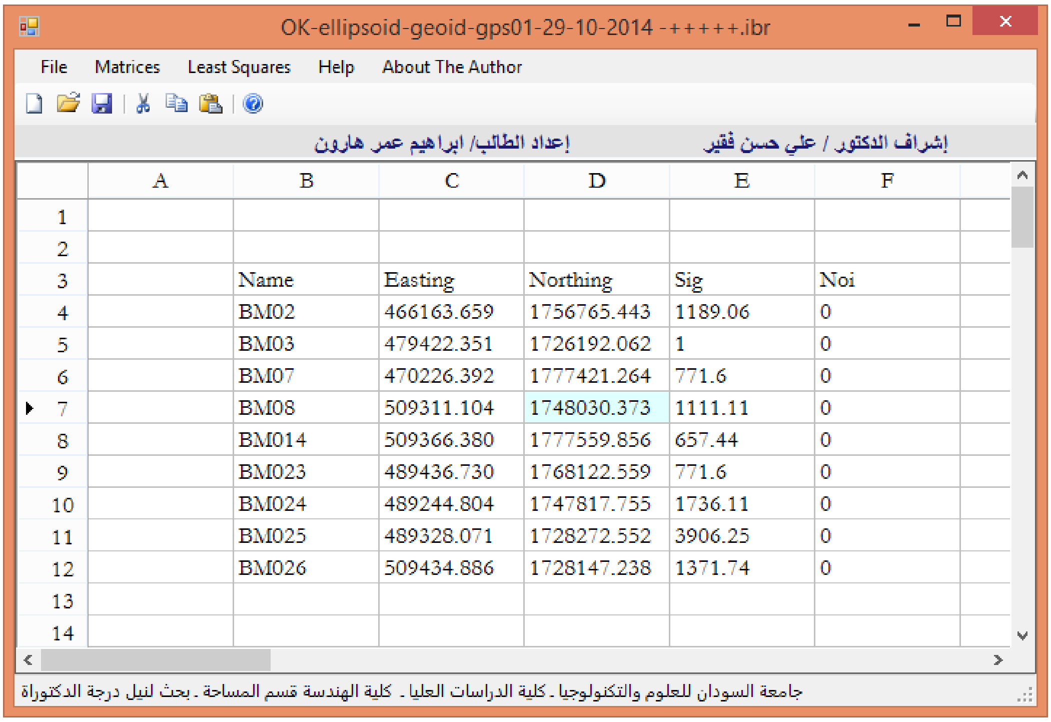
In this program, a worksheet is the file in which you work and store your data .The work sheet is used to list and analyze the data which can be entered, edited and perform required calculations needed to the entered data.

A worksheet consists of cells organized into columns and rows.

Save a group of workbooks in a worksheet :you open a worksheet file by using the Open command (File menu), and you must continue to save changes you make to the worksheet using Save command (File menu).

Enter numbers, text in the cells:

1. Click the cell where you want to enter data.
2. Type the data and press ENTER or TAB.



إعداد الطالب/ ابراهيم عمر هارون

إشراف الدكتور / علي حسن فقير

	A	B	C	D	E	F
1						
2						
3		Name	Easting	Northing	Sig	Noi
4		BM02	466163.659	1756765.443	1189.06	0
5		BM03	479422.351	1726192.062	1	0
6		BM07	470226.392	1777421.264	771.6	0
7		BM08	509311.104	1748030.373	1111.11	0
8		BM014	509366.380	1777559.856	657.44	0
9		BM023	489436.730	1768122.559	771.6	0
10		BM024	489244.804	1747817.755	1736.11	0
11		BM025	489328.071	1728272.552	3906.25	0
12		BM026	509434.886	1728147.238	1371.74	0
13						
14						

جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

Figure 5.7 open Geoids – Ellipsoid data file delete an entry :

To delete a completed entry select the cells , rows, or columns you want to delete , press del key

Copy and paste cells :

You can display, change, find, rearrange, analyze, relate, copy, paste and print any data in your file.

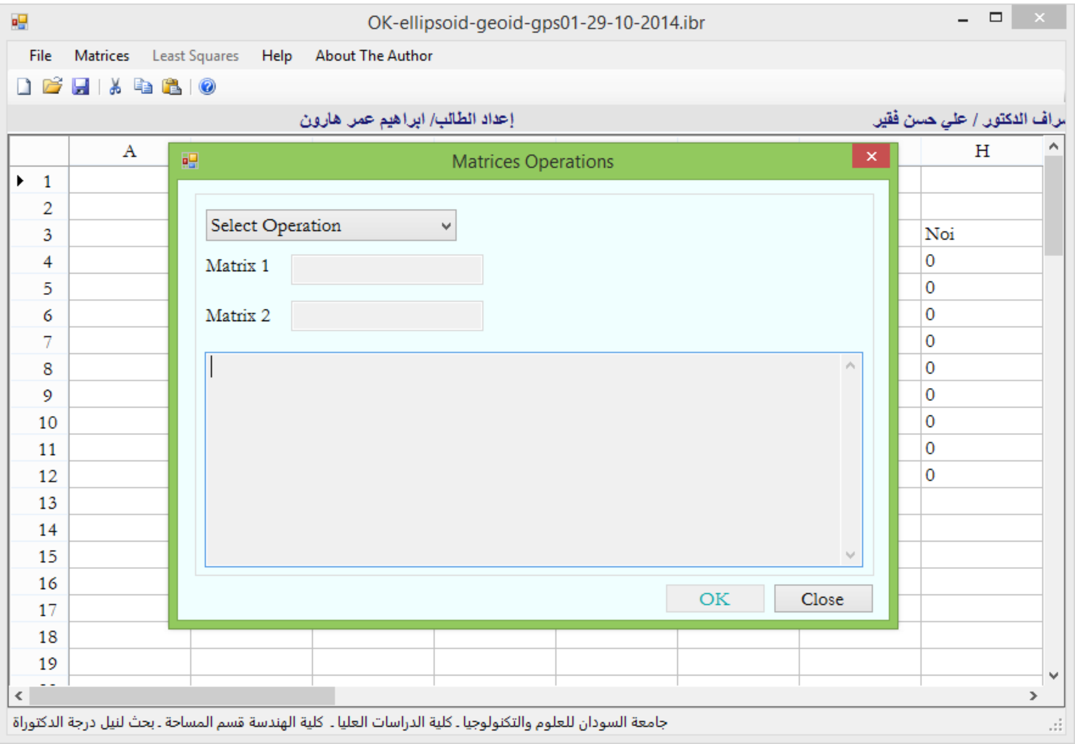
The examples used here are based on field observations from real established field points in two states (Khartoum State & River Nile State).

The program main menu:

A menu is a list of options from which you select the operation you want. (Figure 5.5).

In this program the main menu contains the following:-

- File. (figure 5.6).
- Matrices. (Figure 5.8), (Figure 5.9).



إعداد الطالب/ ابراهيم عمر هارون

راف الدكتور / علي حسن فقير

جامعة السودان للعلوم والتكنولوجيا . كلية الدراسات العليا . كلية الهندسة قسم المساحة . بحث لنيل درجة الدكتوراة

Figure 5.8 Select matrices operations form

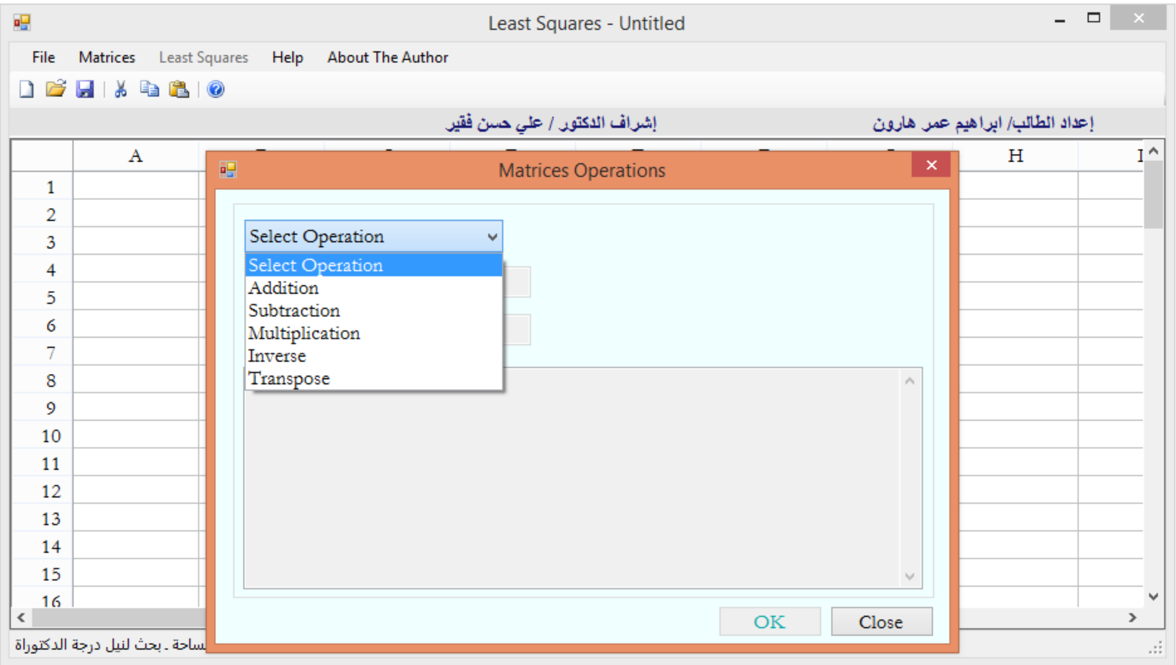


Figure 5.9 Matrices pull-down menu screen

- Least Squares. (Figure 5.10), (Figure 5.11), (Figure 5.12), (Figure 5.13),(Figure 5.14).
- Help.
- About the Author.
- The assistant menu itself contains other, more specialized menus, whose names appear across the top of your screen. In reality, only one menu at a time can be opened.

- Opening Menus:-

- Right now, one of these menus, the file menu, is currently open . Its options appear in a pull– down menu.

- ○ Press → once.
- Now the Matrices menu opens, revealing its pull-down menu.

- Select operation:-
- ▪ Addition.
- ▪ Subtraction.

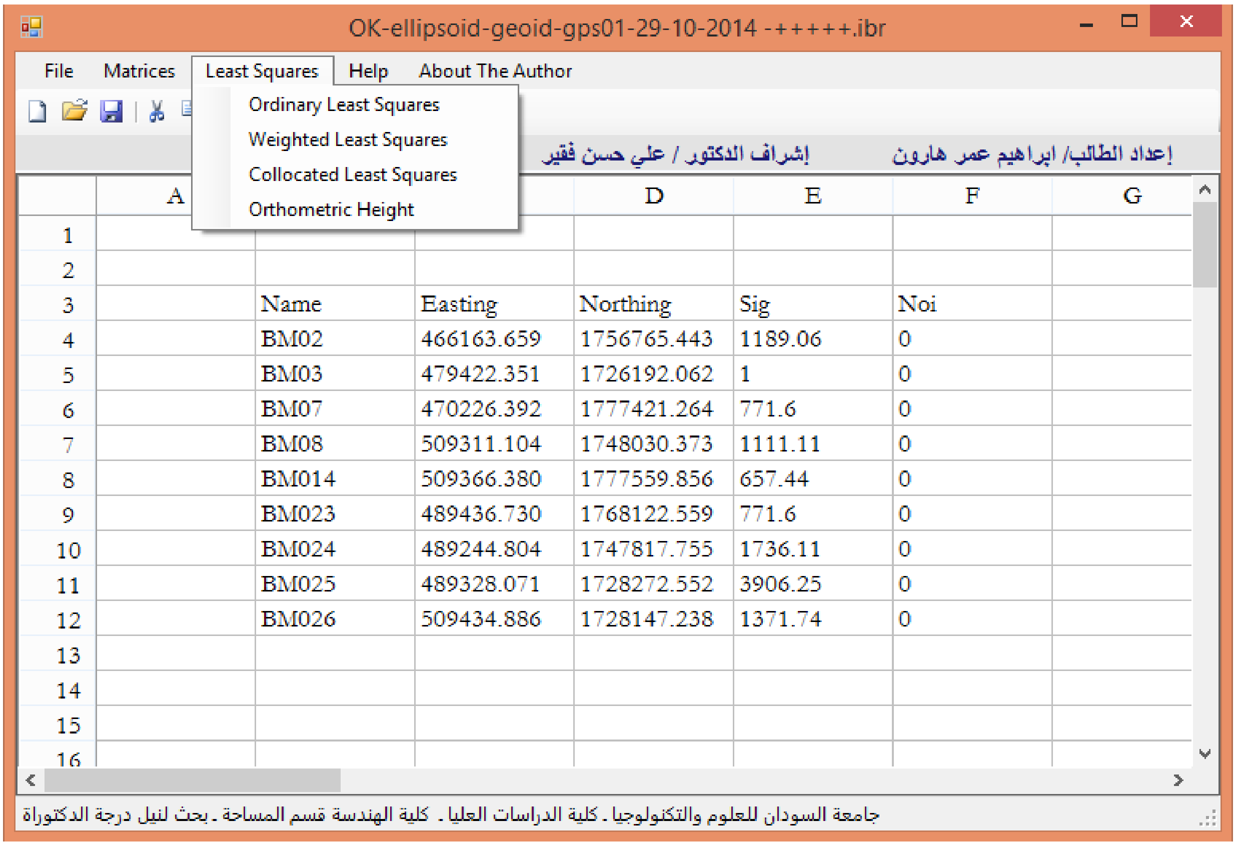
- ▪ Multiplication.
- ▪ Inverse.
- ▪ Transpose.
- See Figure (Figure 5.9).

- ○ Press → once.
- Now the Least Squares menu opens, revealing its pull-down menu. (Figure 5.10).
- ▪ Ordinary Least Squares. (Figure 5.11).
- ▪ Weighted Last Squares. (Figure 5.12).
- ▪ Collocated Least Squares→ Undulations. (Figure 5.13).
- ▪ Orthometric Height. (Figure 5.14).
- Selecting menu options :-
- The current menu is highlighted on your screen. To select a menu option, use ↑ and ↓ or mouse to position the highlight over that option and then press ┘ enter or click the mouse.
- A drive selection sub menu appears.
- ○ Press → once.
- For Help.
- ○ Press → once.
- For Author C.V.

○ Use → and ← or a mouse to back or forth across the menu bar, opening

Menus as you go.

○ Reopen the file menu.


OK-ellipsoid-geoid-gps01-29-10-2014 -+++++.ibr
File
Matrices
Least Squares
Help
About The Author
Ordinary Least Squares
Weighted Least Squares
Collocated Least Squares
Orthometric Height
إعداد الطالب/ ابراهيم عمر هارون
إشراف الدكتور / علي حسن فقير
جامعة السودان للعلوم والتكنولوجيا . كلية الدراسات العليا . كلية الهندسة قسم المساحة . بحث لنيل درجة الدكتوراة

	A			D	E	F	G
1							
2							
3		Name	Easting	Northing	Sig	Noi	
4		BM02	466163.659	1756765.443	1189.06	0	
5		BM03	479422.351	1726192.062	1	0	
6		BM07	470226.392	1777421.264	771.6	0	
7		BM08	509311.104	1748030.373	1111.11	0	
8		BM014	509366.380	1777559.856	657.44	0	
9		BM023	489436.730	1768122.559	771.6	0	
10		BM024	489244.804	1747817.755	1736.11	0	
11		BM025	489328.071	1728272.552	3906.25	0	
12		BM026	509434.886	1728147.238	1371.74	0	
13							
14							
15							
16							

Figure 5.10 Least Squares pull- down menu

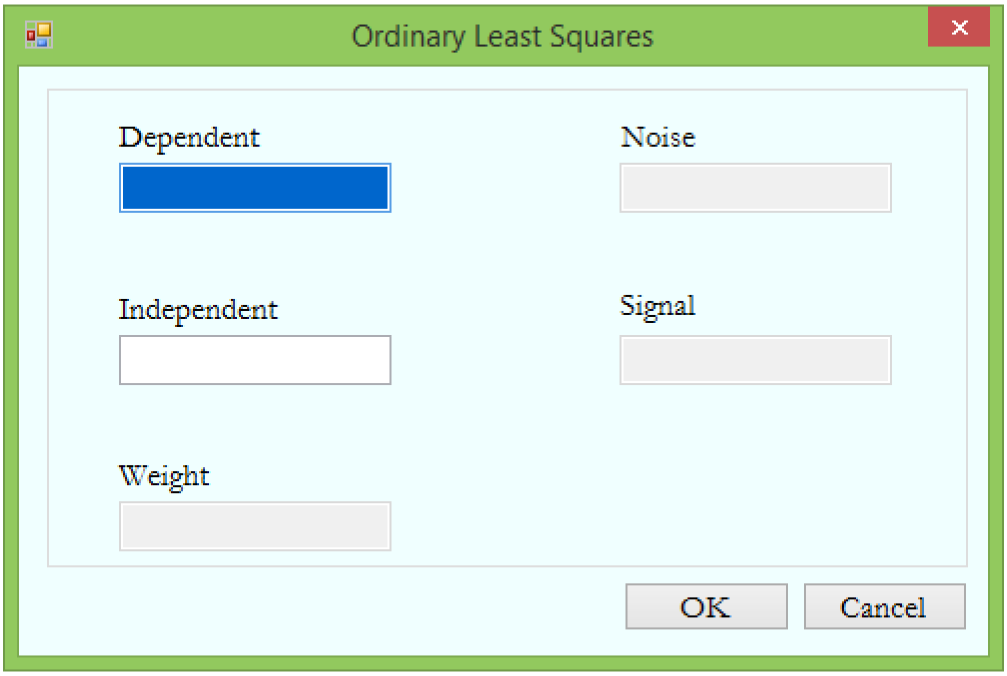


Figure 5.11 Blank Ordinary Least Squares Entry form

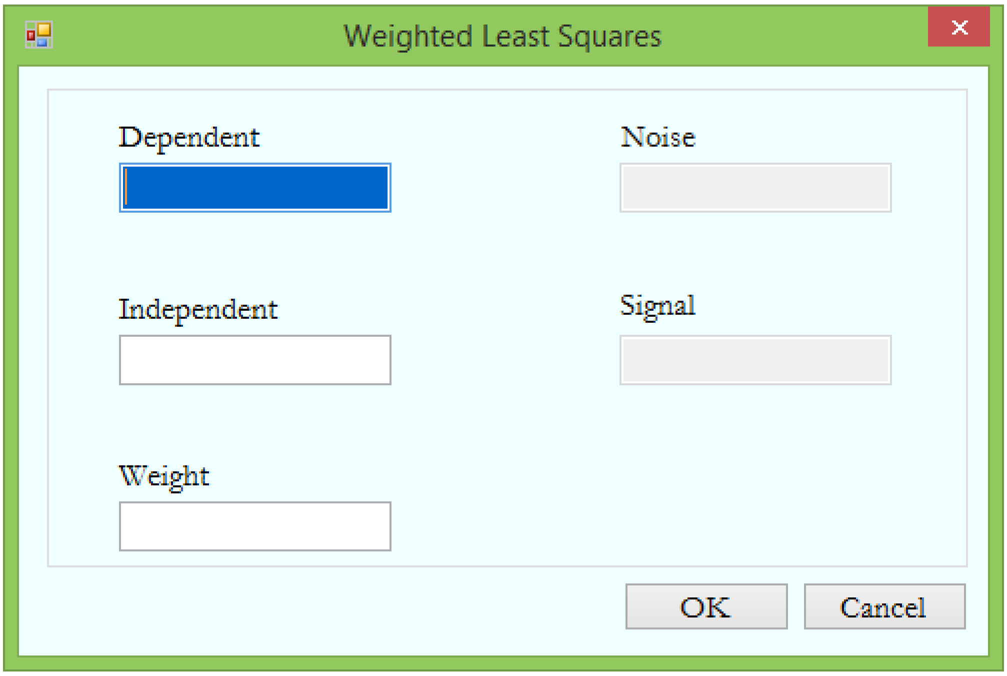


Figure 5.12 Blank Weighted Least Squares Entry form

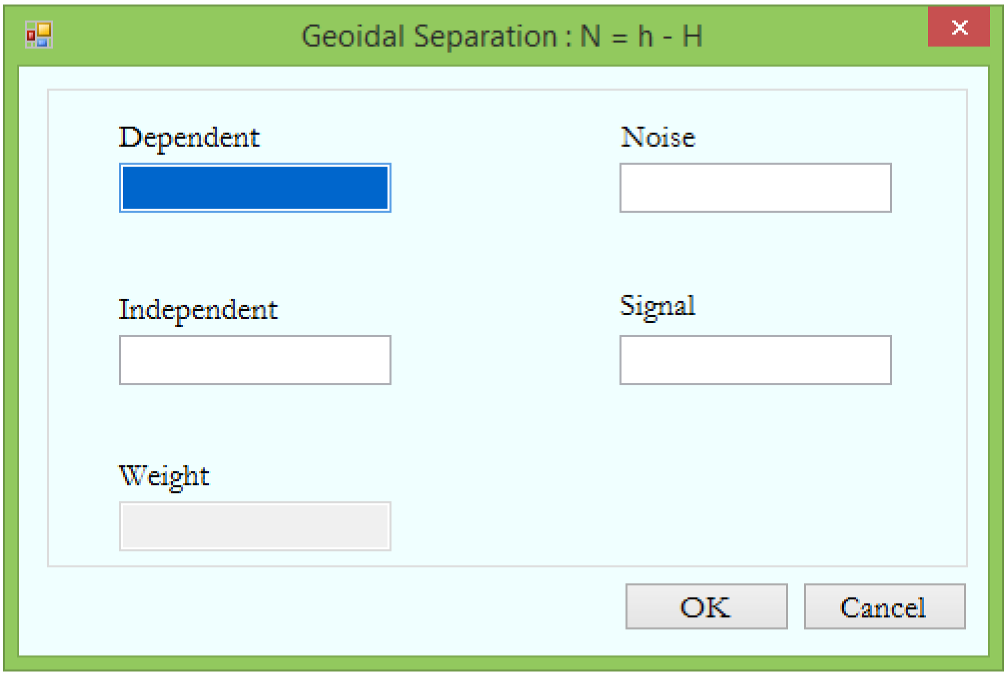
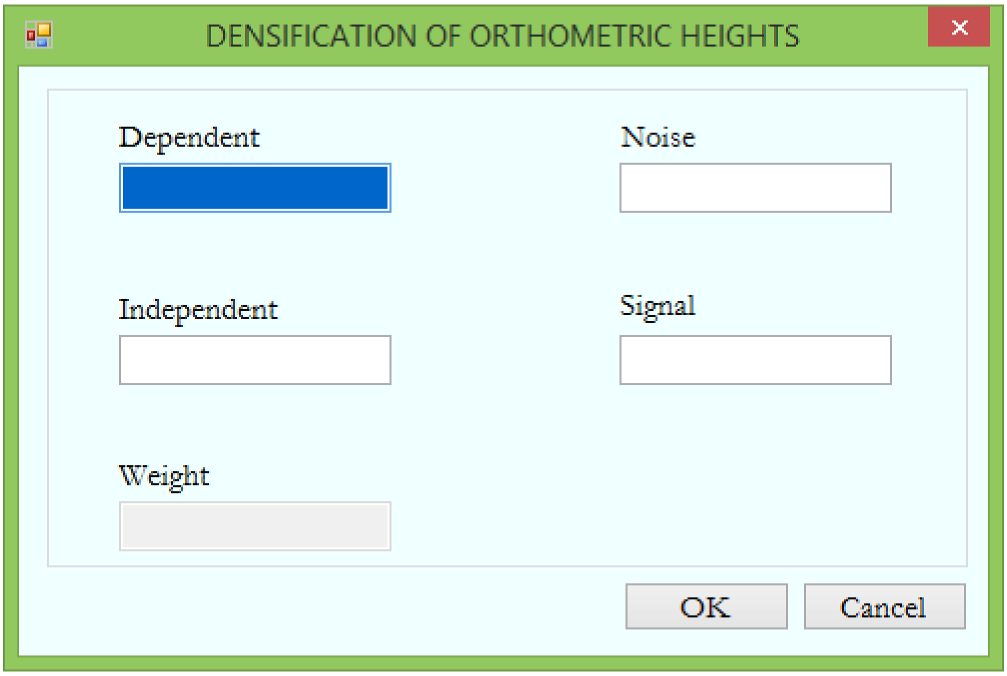


Figure 5.13 Blank Geoidal Separation Entry form



DENSIFICATION OF ORTHOMETRIC HEIGHTS

Dependent

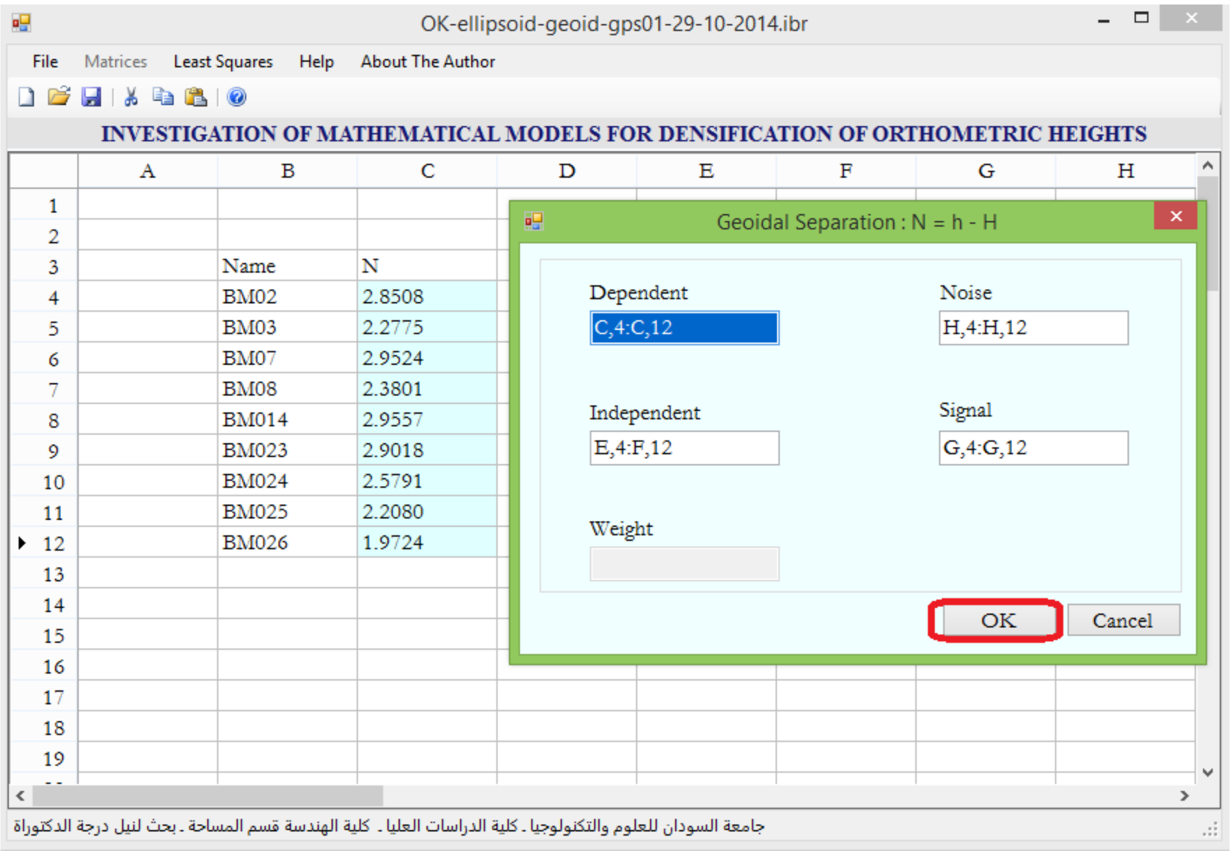
Noise

Independent

Signal

Weight

Figure 5.14 Blank Densification of Orthometric Heights Entry form



OK-ellipsoid-geoid-gps01-29-10-2014.ibr

INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS

	A	B	C
1			
2			
3		Name	N
4		BM02	2.8508
5		BM03	2.2775
6		BM07	2.9524
7		BM08	2.3801
8		BM014	2.9557
9		BM023	2.9018
10		BM024	2.5791
11		BM025	2.2080
12		BM026	1.9724
13			
14			
15			
16			
17			
18			
19			

Geoidal Separation : N = h - H

Dependent: C,4:C,12

Noise: H,4:H,12

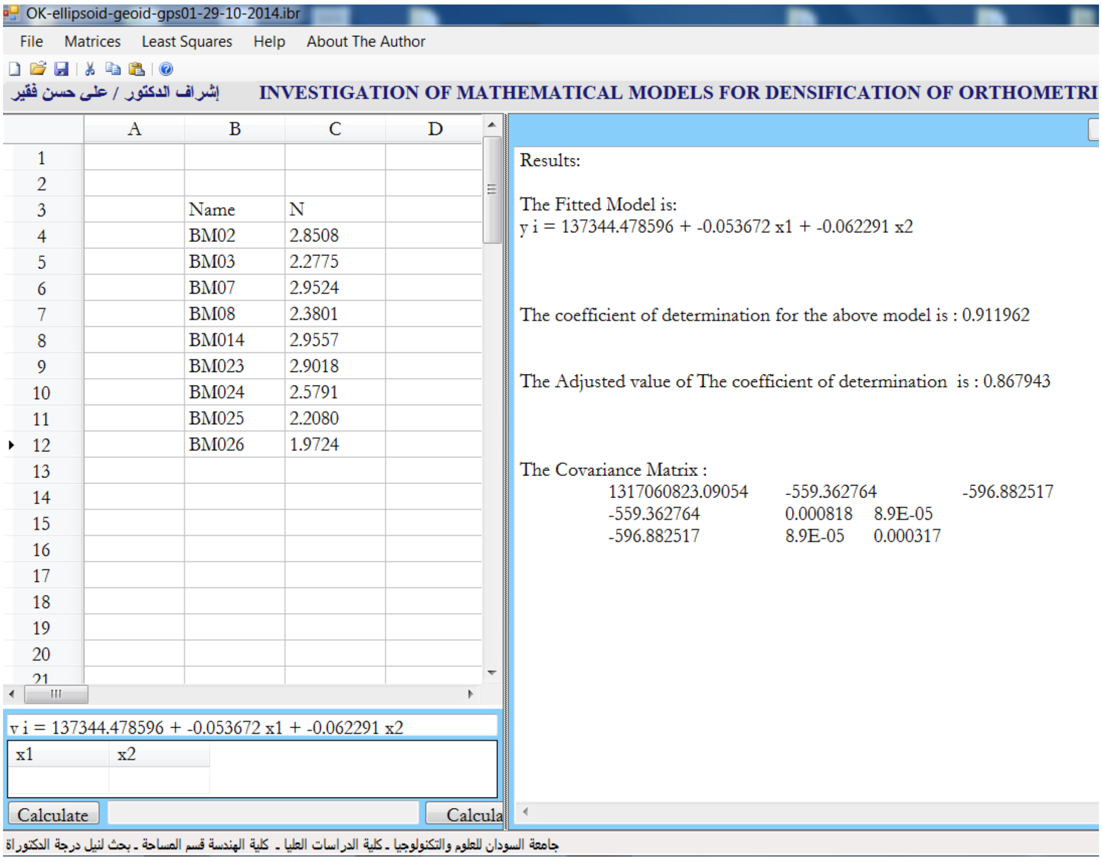
Independent: E,4:F,12

Signal: G,4:G,12

Weight

جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

Figure 5.15 Completed Geoidal Separation Entry form before pressing OK



OK-ellipsoid-geoid-gps01-29-10-2014.ibr
File Matrices Least Squares Help About The Author
إشراف الدكتور / على حسن فقير
INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRI
Calculate
Calcula
جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

	A	B	C	D
1				
2				
3		Name	N	
4		BM02	2.8508	
5		BM03	2.2775	
6		BM07	2.9524	
7		BM08	2.3801	
8		BM014	2.9557	
9		BM023	2.9018	
10		BM024	2.5791	
11		BM025	2.2080	
12		BM026	1.9724	

Results:

The Fitted Model is:
y i = 137344.478596 + -0.053672 x1 + -0.062291 x2

The coefficient of determination for the above model is : 0.911962

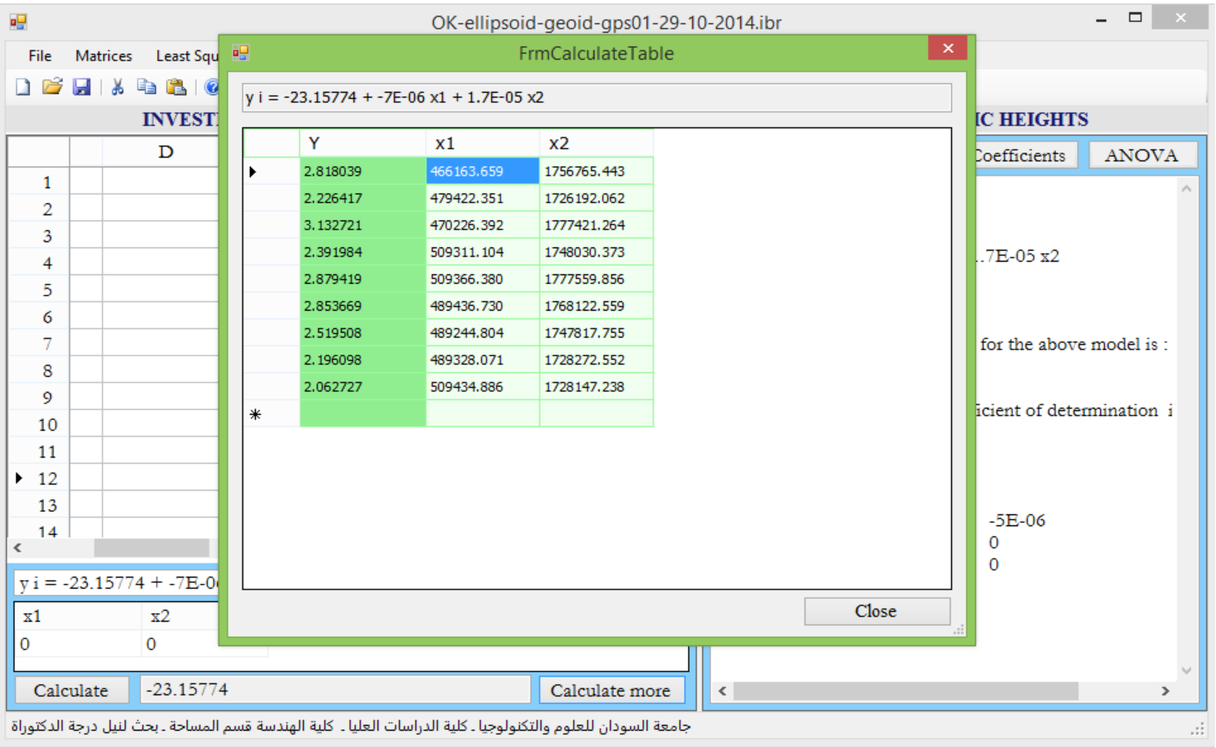
The Adjusted value of The coefficient of determination is : 0.867943

```
The Covariance Matrix :
          1317060823.09054     -559.362764          -596.882517
          -559.362764          0.000818   8.9E-05
          -596.882517          8.9E-05    0.000317
```

y i = 137344.478596 + -0.053672 x1 + -0.062291 x2

x1	x2

Figure 5.16 Geoidal Separation Result after pressing OK



OK-ellipsoid-geoid-gps01-29-10-2014.ibr
File Matrices Least Squ
INVEST
IC HEIGHTS
Coefficients
ANOVA
FrmCalculateTable
Close
Calculate
Calculate more
جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

y i = -23.15774 + -7E-06 x1 + 1.7E-05 x2

Y	x1	x2
2.818039	466163.659	1756765.443
2.226417	479422.351	1726192.062
3.132721	470226.392	1777421.264
2.391984	509311.104	1748030.373
2.879419	509366.380	1777559.856
2.853669	489436.730	1768122.559
2.519508	489244.804	1747817.755
2.196098	489328.071	1728272.552
2.062727	509434.886	1728147.238

y i = -23.15774 + -7E-0

x1	x2
0	0

-23.15774

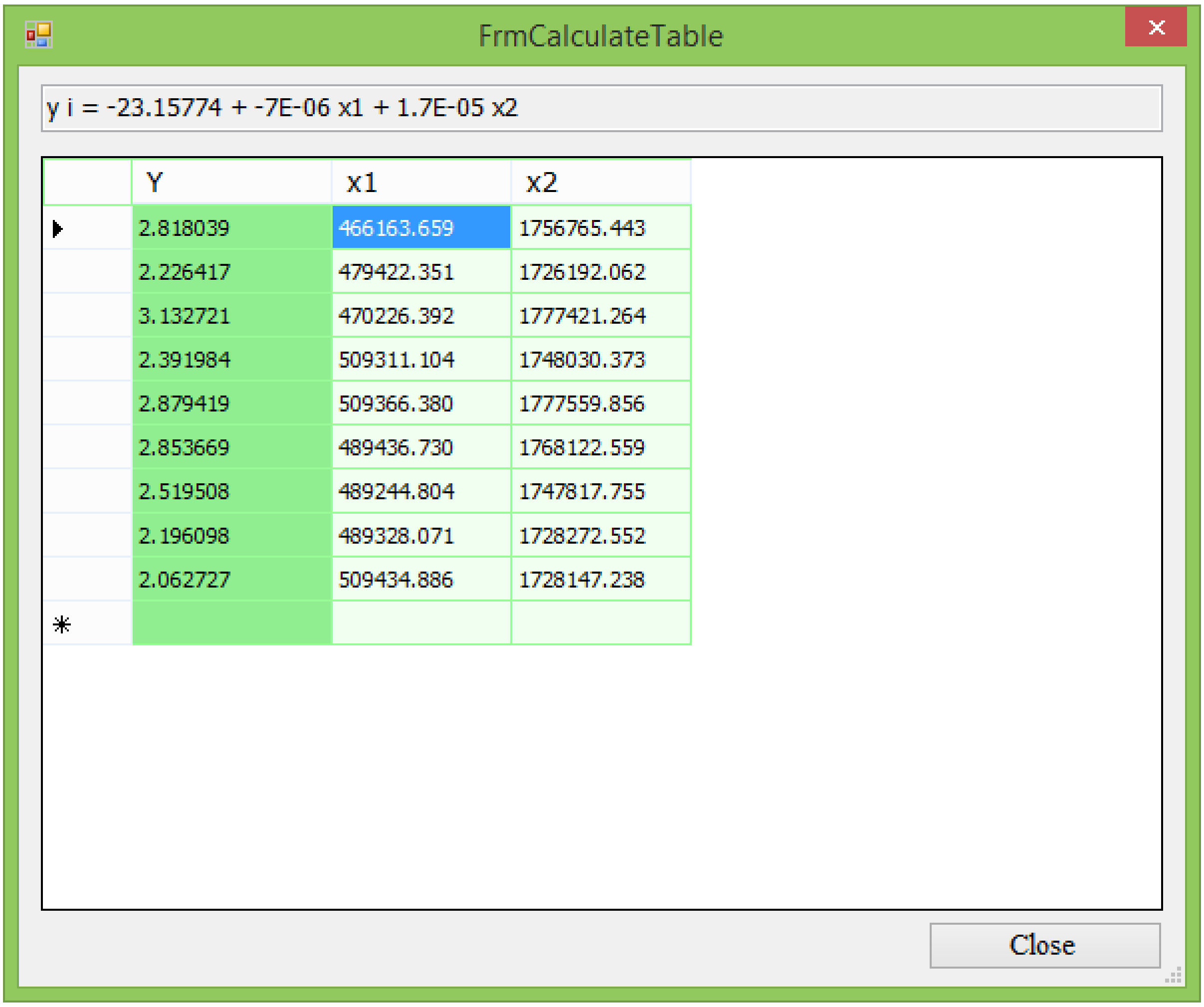
.7E-05 x2

for the above model is :

icient of determination i

-5E-06
0
0

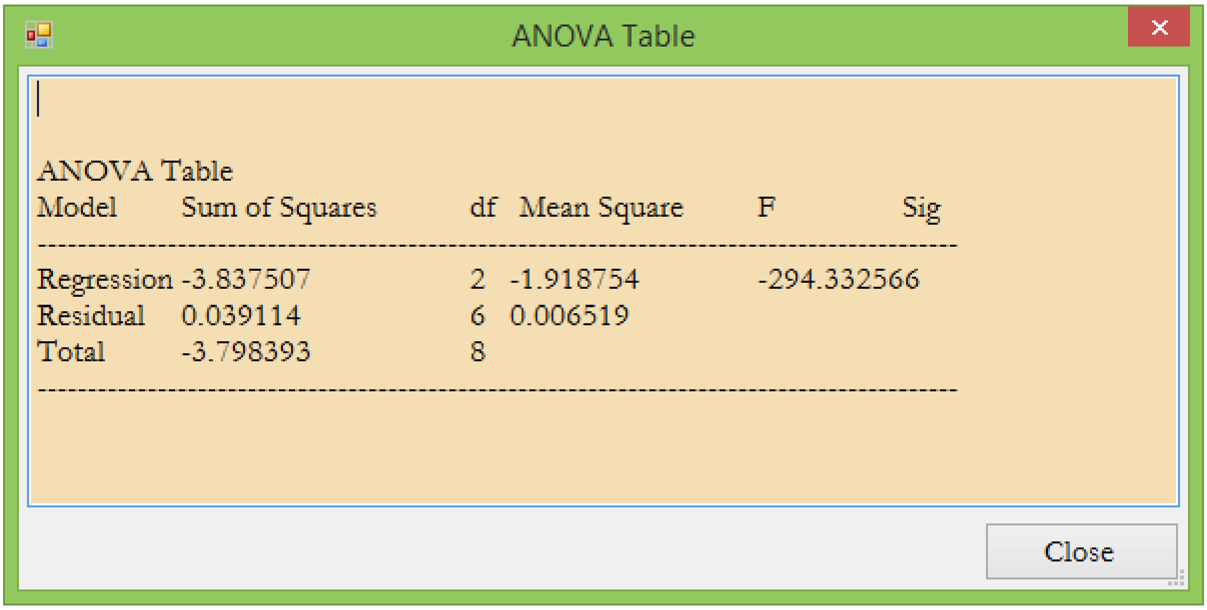
Figure 5.17 Geoidal Separation Prediction Form



y i = -23.15774 + -7E-06 x1 + 1.7E-05 x2

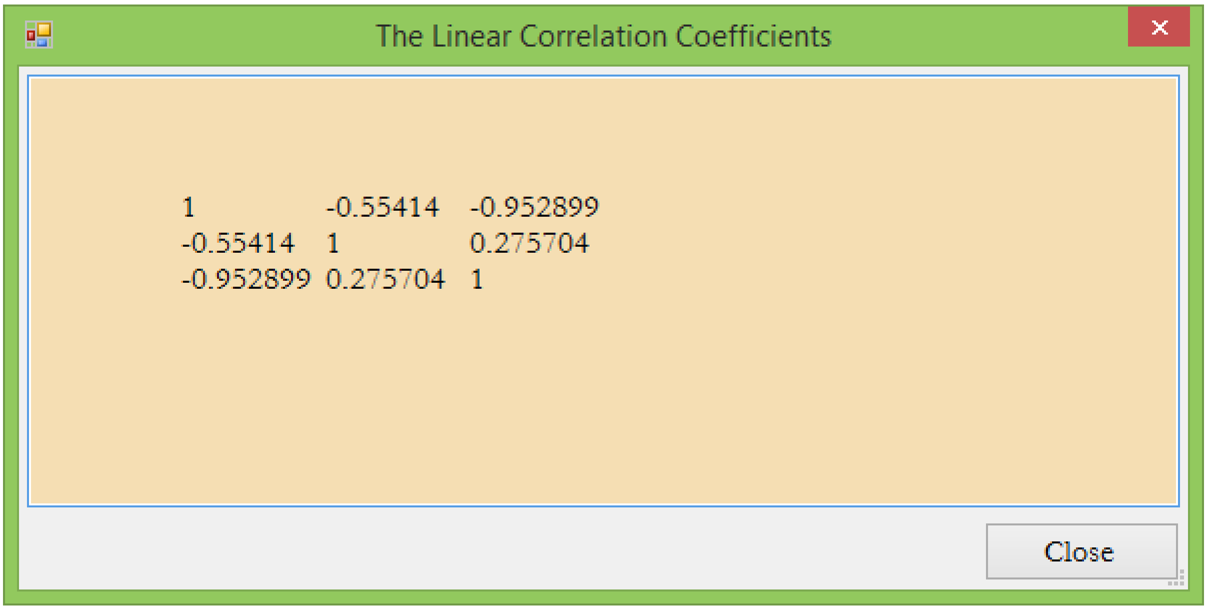
	Y	x1	x2
▶	2.818039	466163.659	1756765.443
	2.226417	479422.351	1726192.062
	3.132721	470226.392	1777421.264
	2.391984	509311.104	1748030.373
	2.879419	509366.380	1777559.856
	2.853669	489436.730	1768122.559
	2.519508	489244.804	1747817.755
	2.196098	489328.071	1728272.552
	2.062727	509434.886	1728147.238
✳			

Figure 5.18 Predicted Geoidal Separation



```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F              Sig
--------------------------------------------------------------------------------
Regression -3.837507                2   -1.918754        -294.332566
Residual    0.039114                6    0.006519
Total      -3.798393                8
--------------------------------------------------------------------------------
```

Figure 5.19 ANOVA Table (Geoidal Separation)



```
1          -0.55414   -0.952899
-0.55414   1          0.275704
-0.952899  0.275704   1
```

Figure 5.20 The Linear Correlation (Geoidal Separation)

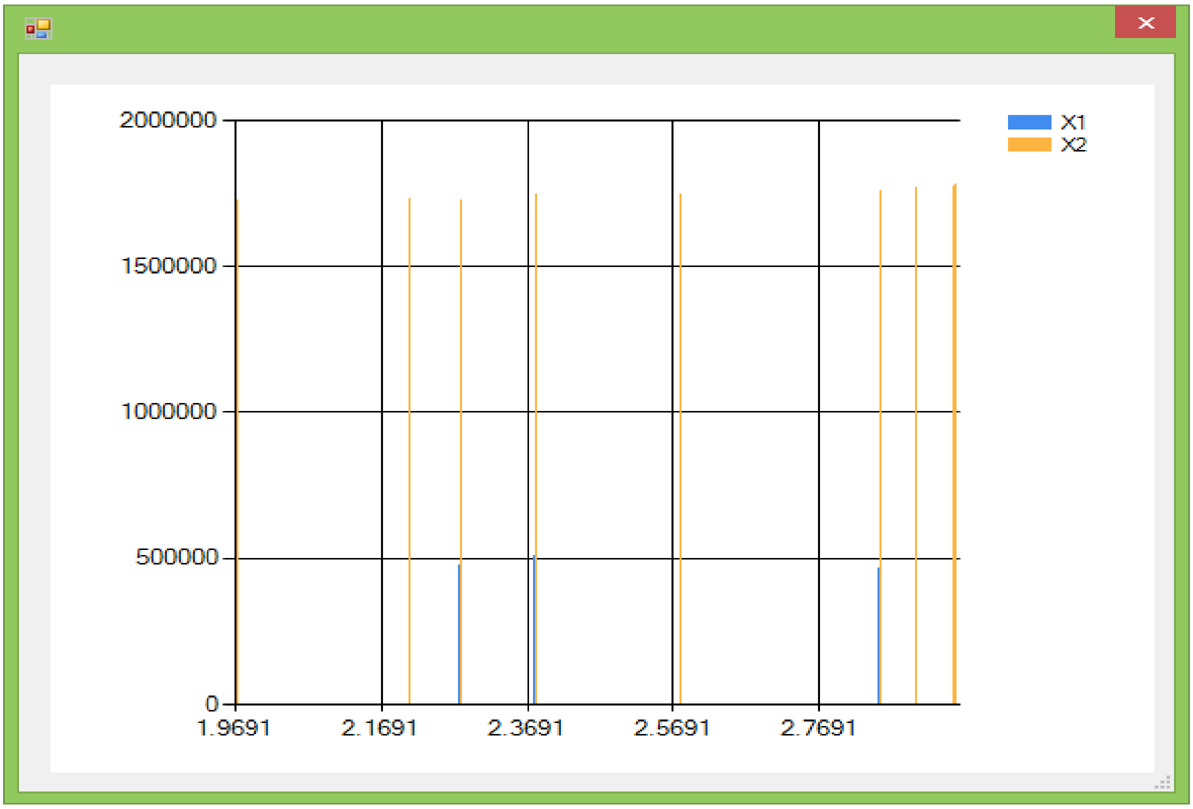
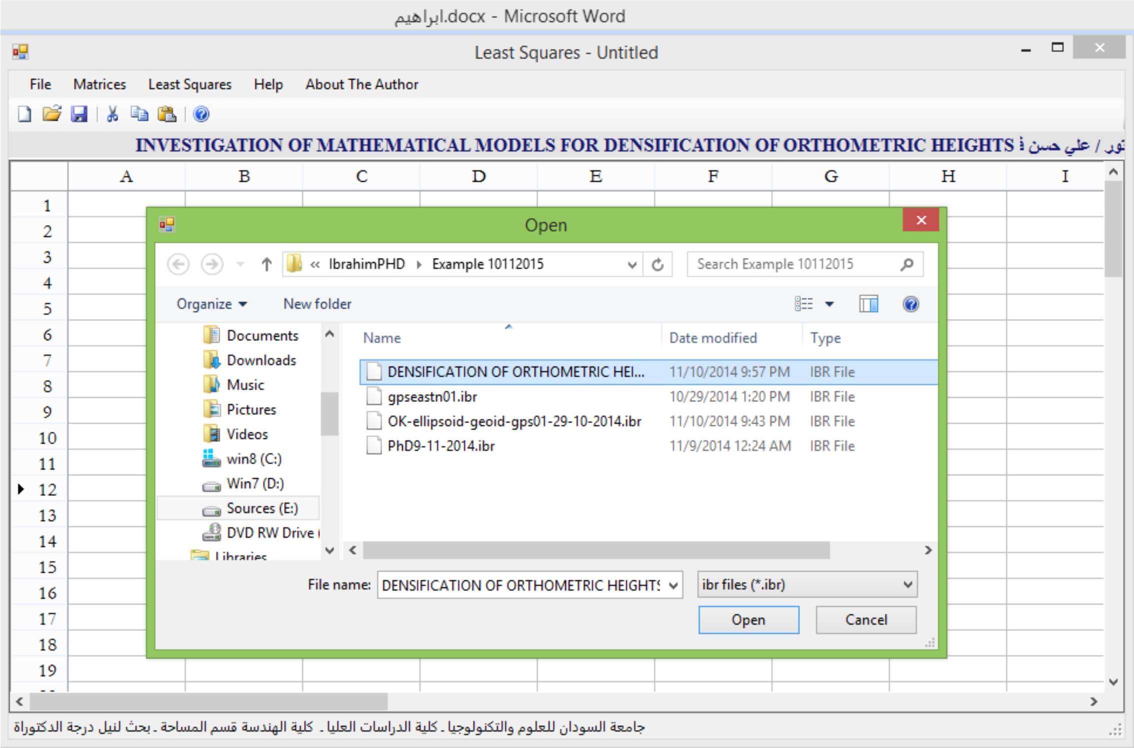


Figure 5.21 Graph for Minimum And Maximum (Geoidal Separation)

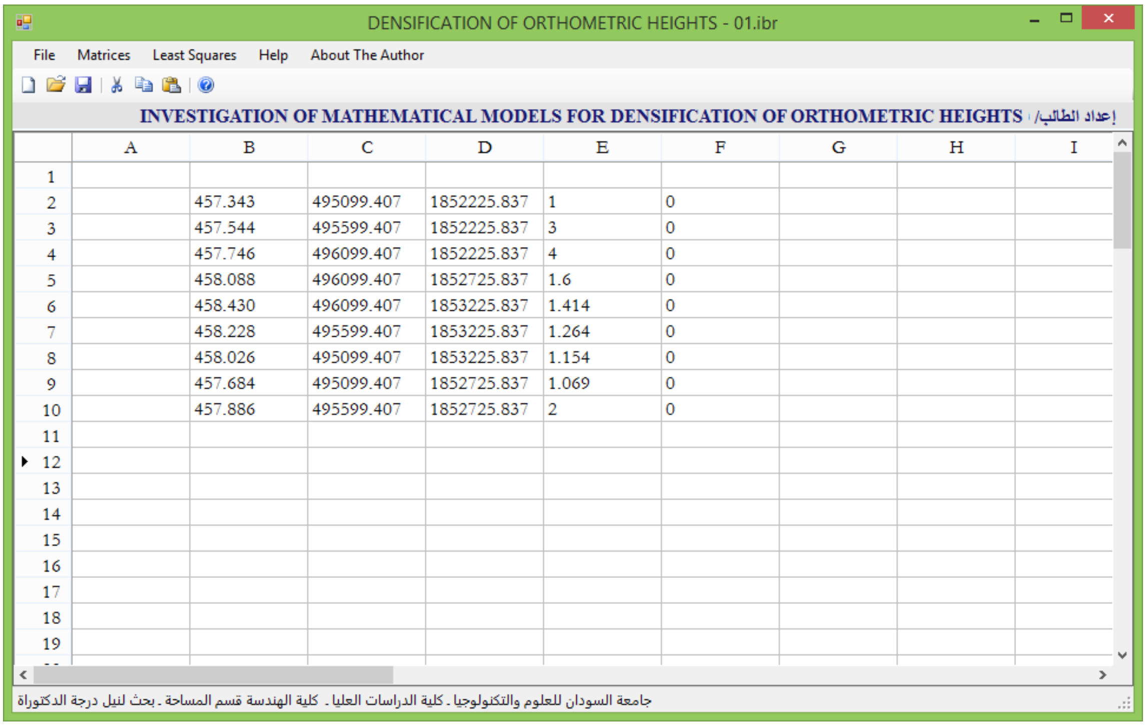

ابراهيم.docx - Microsoft Word
Least Squares - Untitled
File
Matrices
Least Squares
Help
About The Author
INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS نور / علي حسن ا
Open
« IbrahimPHD › Example 10112015
Search Example 10112015
Organize
New folder
Documents
Downloads
Music
Pictures
Videos
win8 (C:)
Win7 (D:)
Sources (E:)
DVD RW Drive
Libraries
Open
Cancel
جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

Name	Date modified	Type
DENSIFICATION OF ORTHOMETRIC HEI...	11/10/2014 9:57 PM	IBR File
gpseastn01.ibr	10/29/2014 1:20 PM	IBR File
OK-ellipsoid-geoid-gps01-29-10-2014.ibr	11/10/2014 9:43 PM	IBR File
PhD9-11-2014.ibr	11/9/2014 12:24 AM	IBR File

File name: DENSIFICATION OF ORTHOMETRIC HEIGHT

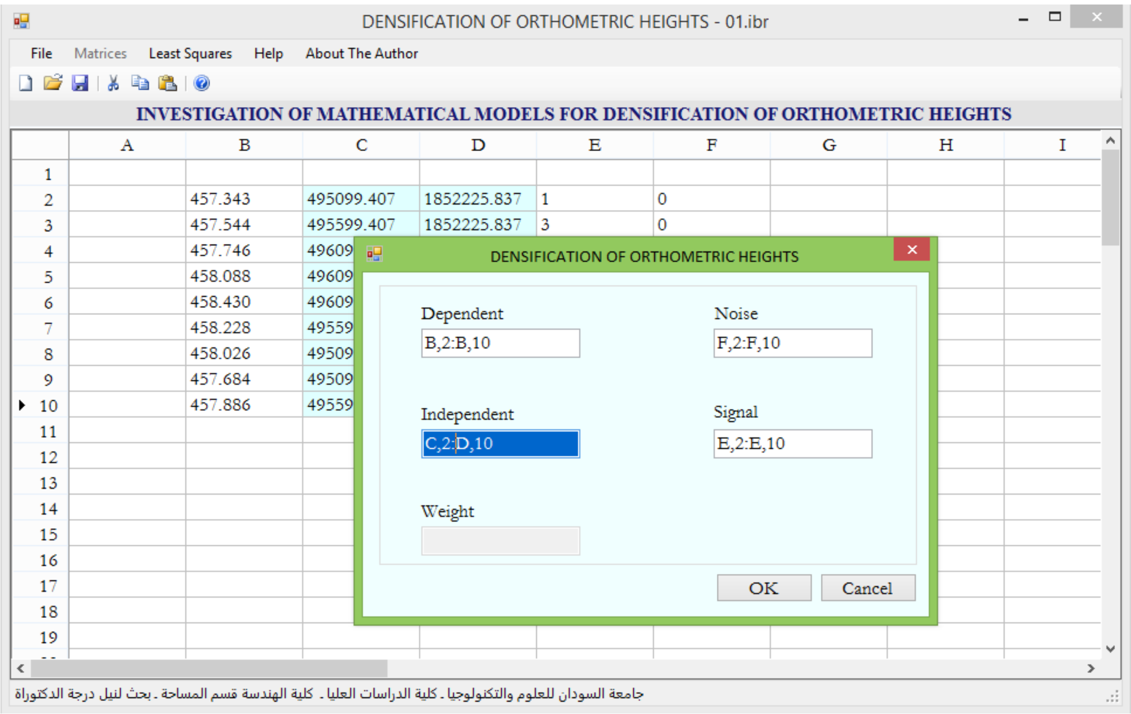
ibr files (*.ibr)

Figure 5.22 Drive and Densification of Orthometric Heights File selection


DENSIFICATION OF ORTHOMETRIC HEIGHTS - 01.ibr
File
Matrices
Least Squares
Help
About The Author
INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS إعداد الطالب/
جامعة السودان للعلوم والتكنولوجيا ـ كلية الدراسات العليا ـ كلية الهندسة قسم المساحة ـ بحث لنيل درجة الدكتوراة

	A	B	C	D	E	F	G	H	I
1									
2		457.343	495099.407	1852225.837	1	0			
3		457.544	495599.407	1852225.837	3	0			
4		457.746	496099.407	1852225.837	4	0			
5		458.088	496099.407	1852725.837	1.6	0			
6		458.430	496099.407	1853225.837	1.414	0			
7		458.228	495599.407	1853225.837	1.264	0			
8		458.026	495099.407	1853225.837	1.154	0			
9		457.684	495099.407	1852725.837	1.069	0			
10		457.886	495599.407	1852725.837	2	0			
11									
12									
13									
14									
15									
16									
17									
18									
19									

Figure 5.23 Densification of Orthometric Heights Open Data File



INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS

	A	B	C	D	E	F	G	H	I
1									
2		457.343	495099.407	1852225.837	1	0			
3		457.544	495599.407	1852225.837	3	0			
4		457.746	49609						
5		458.088	49609						
6		458.430	49609						
7		458.228	49559						
8		458.026	49509						
9		457.684	49509						
10		457.886	49559						
11									
12									
13									
14									
15									
16									
17									
18									
19									

DENSIFICATION OF ORTHOMETRIC HEIGHTS

Dependent: B,2:B,10

Noise: F,2:F,10

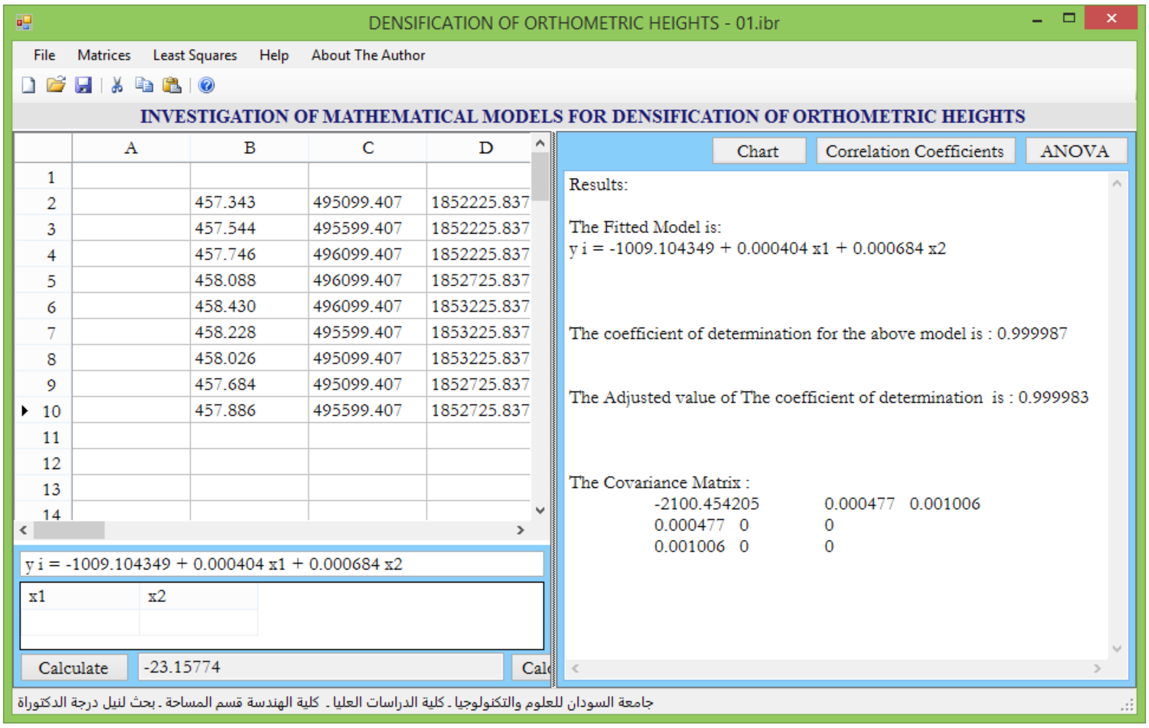
Independent: C,2:D,10

Signal: E,2:E,10

Weight:

جامعة السودان للعلوم والتكنولوجيا . كلية الدراسات العليا . كلية الهندسة قسم المساحة . بحث لنيل درجة الدكتوراة

Figure 5.24 Completed Densification of Orthometric Heights Entry form before pressing OK



INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS

	A	B	C	D
1				
2		457.343	495099.407	1852225.837
3		457.544	495599.407	1852225.837
4		457.746	496099.407	1852225.837
5		458.088	496099.407	1852725.837
6		458.430	496099.407	1853225.837
7		458.228	495599.407	1853225.837
8		458.026	495099.407	1853225.837
9		457.684	495099.407	1852725.837
10		457.886	495599.407	1852725.837
11				
12				
13				
14				

y i = -1009.104349 + 0.000404 x1 + 0.000684 x2

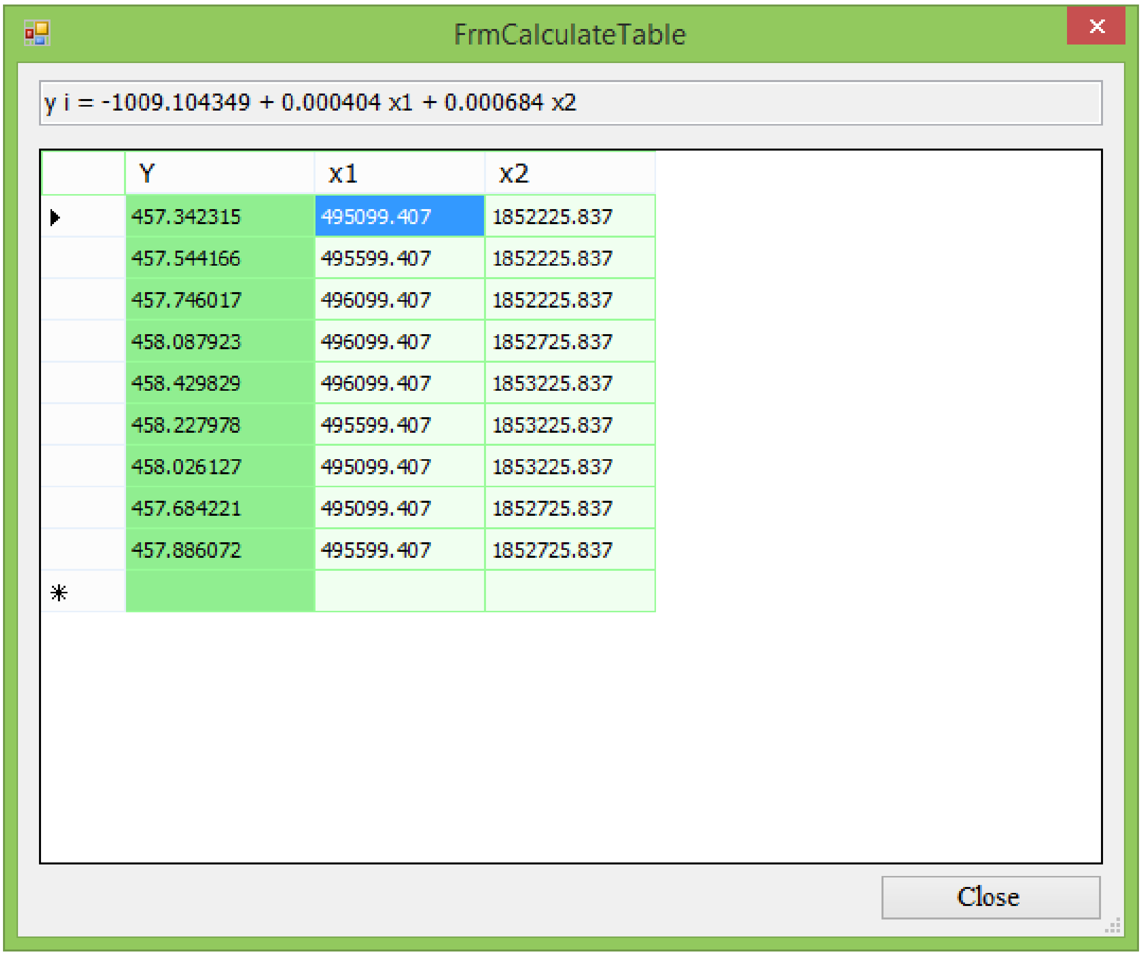
x1	x2

Calculate -23.15774 Cal

```
Results:

The Fitted Model is:
y i = -1009.104349 + 0.000404 x1 + 0.000684 x2


The coefficient of determination for the above model is : 0.999987


The Adjusted value of The coefficient of determination  is : 0.999983


The Covariance Matrix :
            -2100.454205        0.000477   0.001006
            0.000477   0              0
            0.001006   0              0
```

جامعة السودان للعلوم والتكنولوجيا . كلية الدراسات العليا . كلية الهندسة قسم المساحة . بحث لنيل درجة الدكتوراة

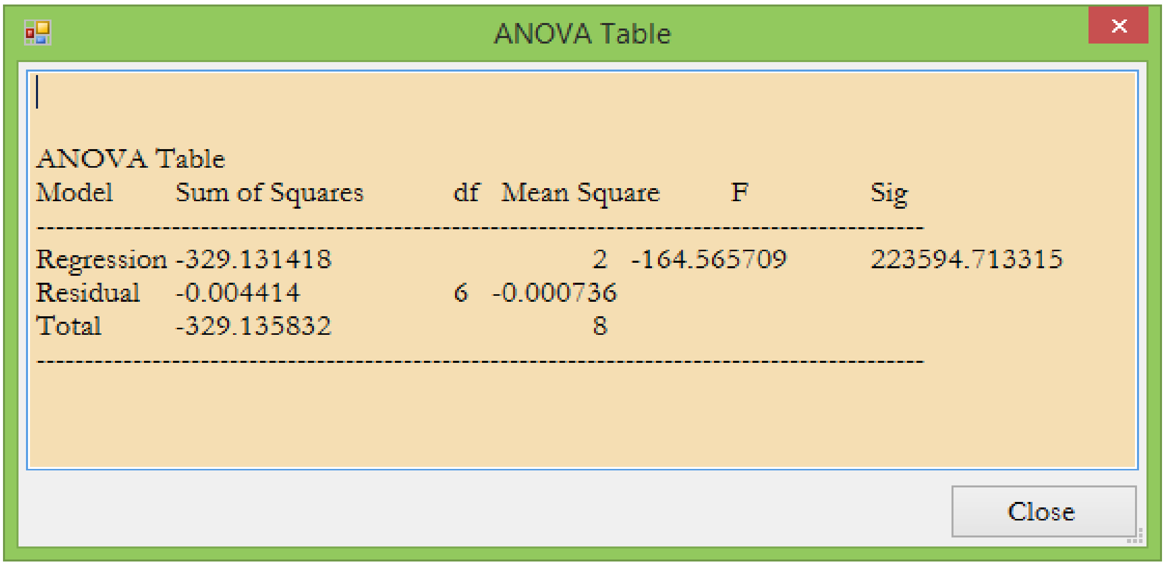
Figure 5.25 Densification of Orthometric Heights Results after pressing OK



y i = -1009.104349 + 0.000404 x1 + 0.000684 x2

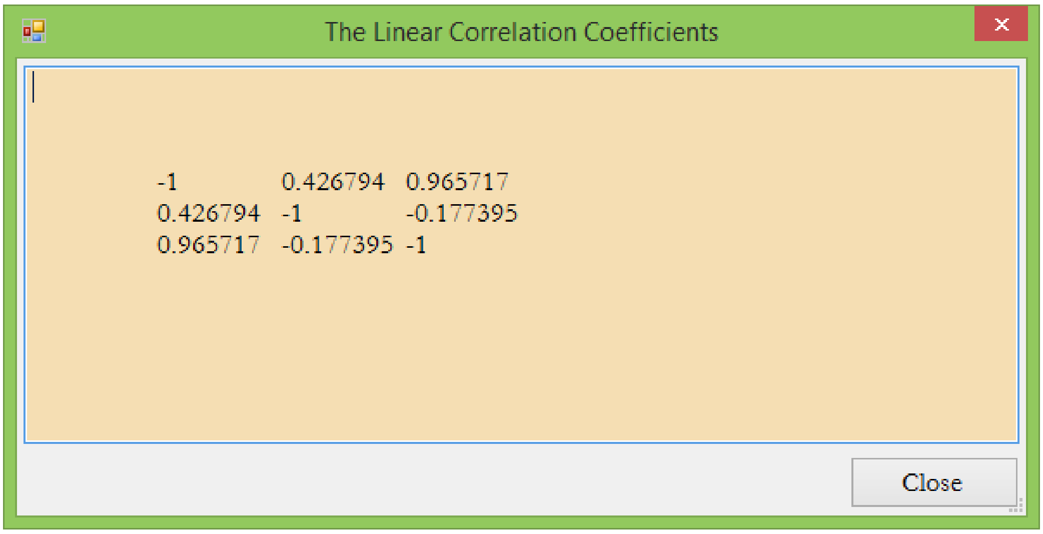
	Y	x1	x2
▶	457.342315	495099.407	1852225.837
	457.544166	495599.407	1852225.837
	457.746017	496099.407	1852225.837
	458.087923	496099.407	1852725.837
	458.429829	496099.407	1853225.837
	458.227978	495599.407	1853225.837
	458.026127	495099.407	1853225.837
	457.684221	495099.407	1852725.837
	457.886072	495599.407	1852725.837
*			

Figure 5.26 Predicted Orthometric Heights



```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F              Sig
---------------------------------------------------------------------------------
Regression -329.131418                      2   -164.565709        223594.713315
Residual     -0.004414               6   -0.000736
Total        -329.135832                    8
---------------------------------------------------------------------------------
```

Figure 5.27 ANOVA Table (Orthometric Heights)



```
-1          0.426794   0.965717
0.426794   -1          -0.177395
0.965717   -0.177395  -1
```

Figure 5.28 The Linear correlation coefficients (Orthometric Heights)

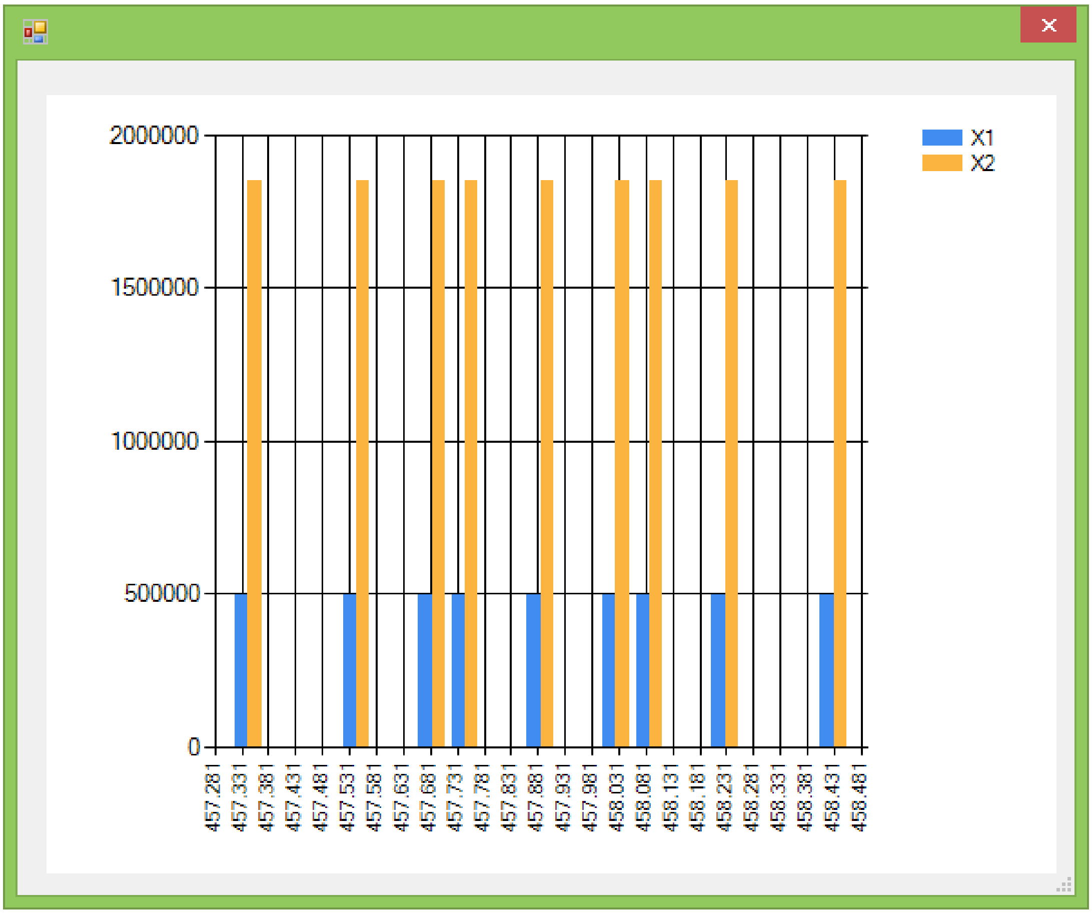


Figure 5.29 Graph for Minimum And Maximum (Orthometric Heights)

CHAPTER SIX

TESTS AND RESULTS

6.1 Introduction

This chapter discusses the results of various tests that were carried out with different selected real leveling networks.
The objectives of these tests are:-

(i) To investigate a mathematical model for densification of orthometric heights.

(ii) To find out the parameters of the following polynomials:-

$$H_i = a_1 + a_2X + a_3Y \qquad 6.1$$

$$H_i = a_1 + a_2X + a_3Y + a_4XY \qquad 6.2$$

$$H_i = a_1 + a_2X + a_3Y + a_4XY + a_5X^2 \qquad 6.3$$

$$H_i = a_1 + a_2X + a_3Y + a_4XY + a_5Y^2 \qquad 6.4$$

Using the three dimensions of known existing points.

(iii) Re compute the height (Hi) of the points using the parameters (a_1, a_2, a_3, a_4, a_5), and the two dimensions of points.

(iv) To obtain the residuals from the difference of (ii) and (iii) heights.

(v) To find out the height (Hi) of the same points using the mathematical least squares collocation model.

(vi) Compute the residuals of model predicted points.

(vii) Compare the residuals of (iv) and (vi).

(viii) To obtain the least number of height points.

(ix) To find the optimum distribution of height points.

Using the polynomials ***6.1, 6.2***, **6.**3and 6.4

In each case the parameters of the above polynomials were obtained fitting the polynomials to the observed data (The three dimensions of known existing points).

(x) Re compute the height (Hi) of the points using the parameters (a_1, a_2, a_3, a_4, a_5), and the two dimensions of the same existing points.

(xi) Obtain the residuals from the difference of existing and computed (predicted) heights.

(xii) The results are as in table 6.3

6.2 Computation and Methodology

The Trimble Business Center (TBC) software was used for computing the coordinates of the newly established control points observed by RTK technique. The surveying teams adhered to the following steps in their computation methodology:

- GPS base line processing.
- Survey network adjustment.
- Quality assurance and quality control of data (QA/QC).
- Survey data import and export.
- Digital terrain modeling and contouring.
- Datum transformation and projection.
- Survey project management.

6.3 Mapping methodology

The following softwares were used as appropriate for producing the deliverable products:

- Arc GIS 10.
- Land Development 2009.
- Civil 3D 2012.
- Trimble Business Center.

6.4 Map Datum

Coordinate System	**UTM**
Zone	36N
Study area Datum Spheroid	WGS84
Geoid Model	EGM 2008
Coordinate Units	Meters
Distance Units	Meters
Height Units	Meters
Origin North	500000
Origin East	0000

Table 6.1-Map Datum

6.5 Adjustment report for GPS observations and leveling

6.5.1 Quality Assurance and Quality Control (QA/QC) of data and result achievements

6.5.1.1 Quality Assurance

Quality Assurance aims to assure that quality survey and quality results will be built in before survey is done.

6.5.1.2 Quality control

Quality control aims to determine that quality survey and quality results did occur after survey was done.

In surveying the quality of a network is considered to be made up of three factors: Economy, Precision, and Reliability (Teunissen, P.J.G. (1985). Economy declares the total cost of designing, measuring, adjusting and validating the survey network. Network precision, as described by the a posteriori covariance matrix of the network coordinates (network's characteristics in propagating random errors). And reliability, as described by the minimal detectable biases, expresses the ability of the redundant observations to detect and identify specific modeling errors (internal reliability, together with the networks characteristics in propagating these modeling errors (external reliability).
From the given reference geodetic point S2080 to the first established control point K01, for a distance of 50km for GPS observation and 100km for double leveling method.(50km from S2080 to K01 + 50km from K01 to S2080).

For acceptance or rejection, a comparison was made between the obtained GPS and the double leveling height .The difference between the two heights is 6cm.

In addition to that the topographic surveying data (detail survey and spot heights in grid lines) were within the following parameters:

HZ: ± 10cm
V: + 5cm
Where HZ is the horizontal error and V is the vertical error.

And the control point's data were within the following parameters:

HZ: ± 3cm

V: + 10 $\sqrt{K}$

Where HZ is the horizontal control misclosure , V is the allowable vertical control misclosure and K is the length in km.

Finally all field survey works and results where within the required accuracy tolerance (acceptance).

6.6 Least Squares surface fitting

To check my prediction model program and compare it with least squares fitting I select the South West corner of loop 3, 4X4 KM controlled by K46, K47, and K49&K50.

Four suggested cases were used to investigate the optimum distribution and suitable location of data points (4, 5, 6 E and 6N). See table 6.2 and figures (6.2a to 6.8d).

Test Area Figure	Block Size (Km)	Grid Cell (M)	No. of observed points	Observed Heights		
				Maximum Height(m)	Minimum Height(m)	Difference in Height(m)
Fig 6.2	1X1	200X100	66	458.985	457.524	1.461
Fig 6.3	1X2	200X100	126	460.253	457.524	2.729
Fig 6.4	2X1	200X100	121	459.581	457.524	2.057
Fig 6.5	2X2	200X100	231	460.253	457.524	2.729
Fig 6.6	2X4	200X100	451	463.398	457.524	5.874
Fig 6.7	4X2	200X100	441	462.200	457.524	4.676
Fig 6.8	4X4	200X100	861	463.398	455.033	8.365

Table 6.2-Gridded observed height

6.7 Data points configuration:-

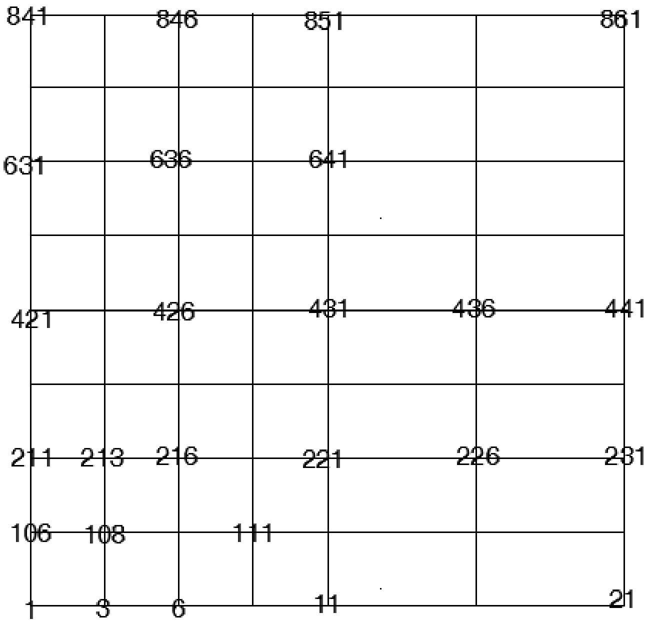


Fig 6.1 Master

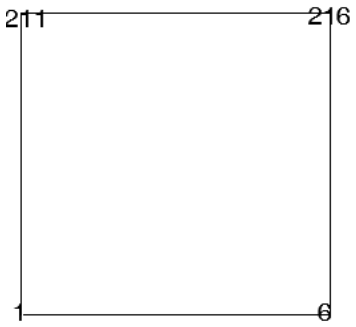


Fig 6.2a-1X1 4pts

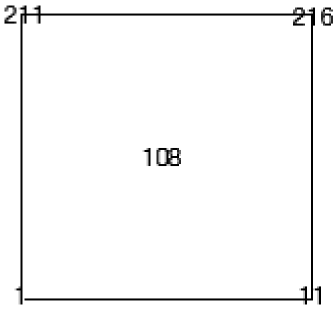


Fig 6.2b- 1X1 5pts

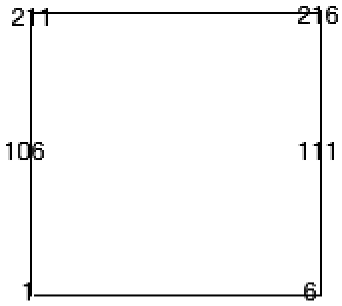


Fig 6.2c- 1X1 6pts1

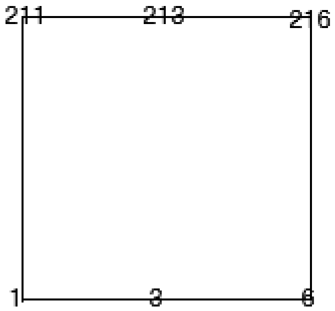


Fig 6.2d- 1X1 6pts2

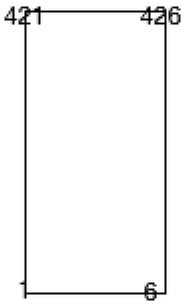


Fig 6.3a 1X2 4pts

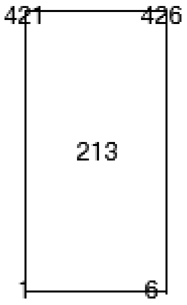


Fig 6.3b 1X2 5pts

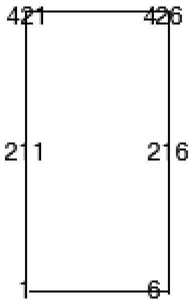


Fig 6.3c 1X2 6pts1

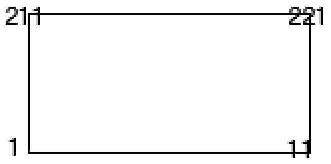


Fig 6.4a 2X1 4pts

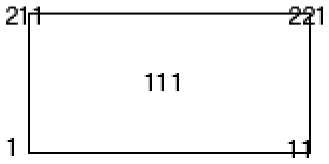


Fig 6.4b 2X1 5pts

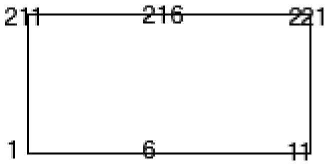


Fig 6.4c 2X1 6pts2

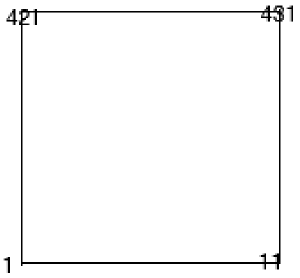


Fig 6.5a 2X2 4pts

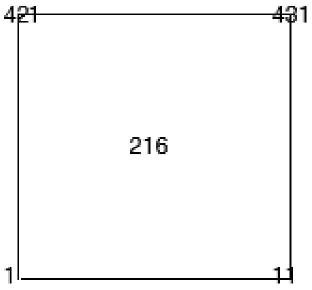


Fig 6.5b 2X2 5pts

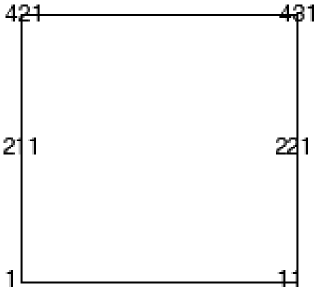


Fig 6.5c 2X2 6pts1

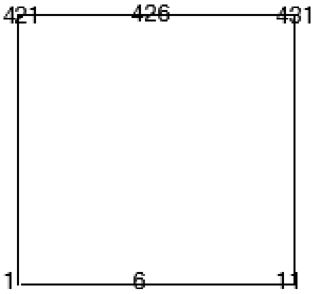


Fig 6.5d 2X2 6pts2

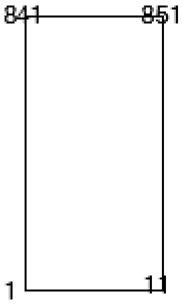


Fig 6.6a 2X4 4pts

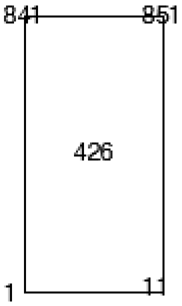


Fig 6.6b 2X4 5pts

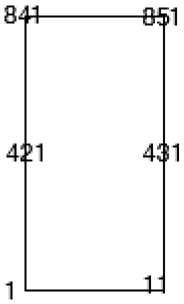


Fig 6.6c 2X4 6pts1

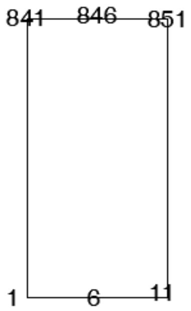


Fig 6.6d 2X4 6pts2

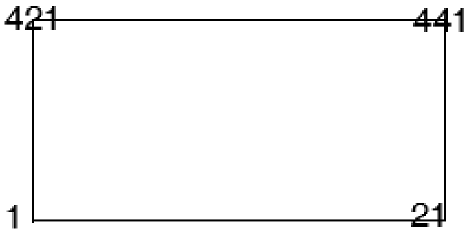


Fig 6.7a 4X2 4pts

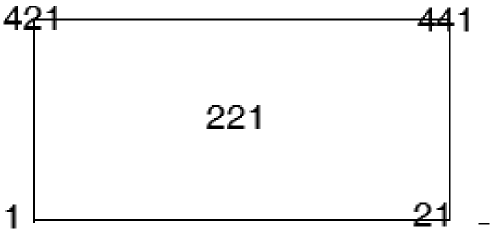


Fig 6.7b 4X2 5pts

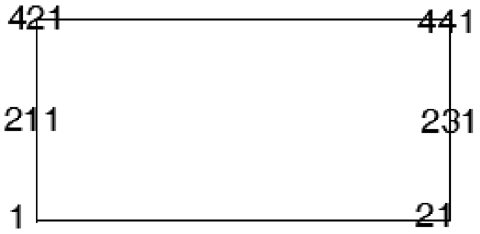


Fig 6.7c 4X2 6pts1

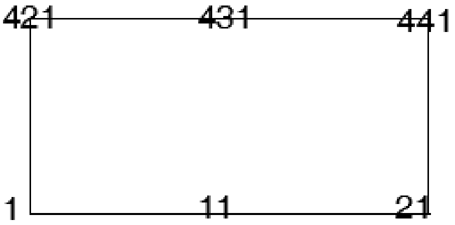


Fig 6.7d 4X2 6pts2

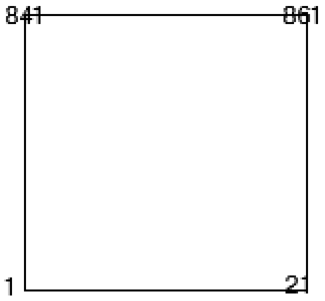
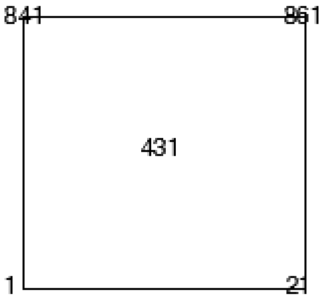
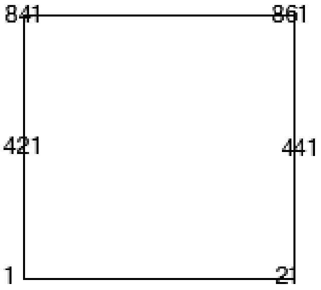


Fig 6.8 a 4X4 4pts





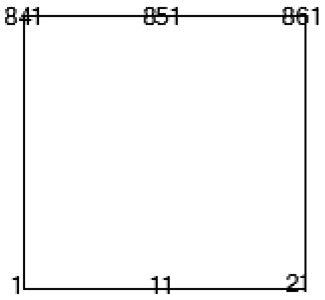


Fig 6.8b 4X4 5pts Fig 6.8c 4X4 6pts1 Fig 6.8d 4X4 6ptsS2

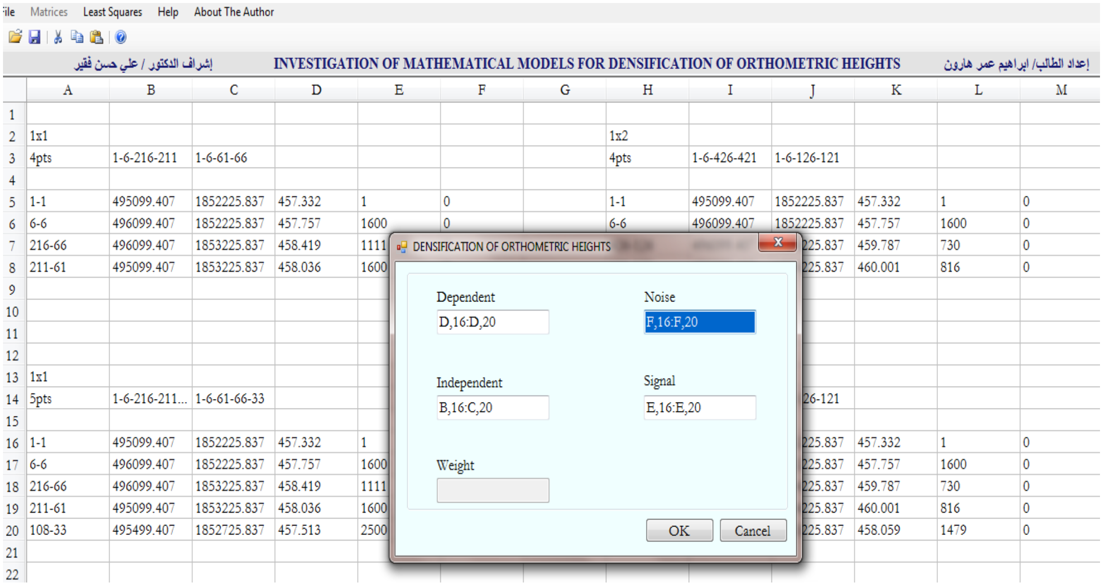
Area Size (km)	Grid Figure	No. of data points	Polynomial			
			$H_i = a_1+ a_2X+ a_3Y$	$H_i = a_1+ a_2X+ a_3Y+a_4XY$	$H_i = a_1+ a_2X+a_3Y+ a_4XY+a_5X^2$	$H_i = a_1+ a_2X+a_3Y+ a_4XY+ a_6Y^2$
			Residual (m)	Residual (m)	Residual (m)	Residual (m)
1X1	6.2a	4	0.11553	0.1153		
	6.2b	5	0.16423	0.16423	0.16423	0.16423
	6.2c	6	0.12344	0.12344	0.12344	0.12344
	6.2d	6	0.18440	0.18440	0.18440	0.18440
1X2	6.3a	4	0.37347	0.37347		
	6.3b	5	0.39807	0.39807	0.39807	0.39807
	6.3c	6	0.38730	0.38730	0.38730	0.38730
		6	0.38730	0.38730	0.38730	0.38730
2X1	6.4a	4	0.47000	0.47000		
	6.4b	5	0.52843	0.52843	0.52843	0.52843
	6.4c	6	0.51705	0.51705	0.51705	
		6				
2X2	6.5a	4	2.05057	2.05057		
	6.5b	5	2.08791	2.08791	2.08791	2.08791
	6.5c	6	2.05310	2.05310	2.05310	2.05310
	6.5d	6	2.07506	2.07506	2.07506	2.07506
2X4	6.6a	4	3.08087	3.08087		
	6.6b	5	3.08356	3.08356	3.08356	3.08356
	6.6c	6	3.09067	3.09067	3.09067	3.09067
4X2	6.7a	4	3.21617	3.21617	3.21617	3.21617
	6.7b	5	3.22300	3.22300	3.22300	3.22300
	6.7c	6	3.23397	3.23397	3.23397	3.23397
4X4	6.8a	4	5.31513	5.31513		
	6.8b	5	5.31597	5.31597		
	6.8c	6	5.31957	5.31957		5.31957
	6.8d	6	5.32382	5.32382	5.32382	

Table 6.3-Application of Polynomials in different areas

6.8 The mathematical modeling program surface fitting

To find out the height (Hi) of the same points using the mathematical least squares collocation.

Compute the residuals of predicted points using the program (fig 6.9)



إشراف الدكتور / علي حسن فقير INVESTIGATION OF MATHEMATICAL MODELS FOR DENSIFICATION OF ORTHOMETRIC HEIGHTS إعداد الطالب/ ابراهيم عمر هارون

DENSIFICATION OF ORTHOMETRIC HEIGHTS

Dependent: D,16:D,20

Noise: F,16:F,20

Independent: B,16:C,20

Signal: E,16:E,20

Weight:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	1x1							1x2					
3	4pts	1-6-216-211	1-6-61-66					4pts	1-6-426-421	1-6-126-121			
4													
5	1-1	495099.407	1852225.837	457.332	1	0		1-1	495099.407	1852225.837	457.332	1	0
6	6-6	496099.407	1852225.837	457.757	1600	0		6-6	496099.407	1852225.837	457.757	1600	0
7	216-66	496099.407	1853225.837	458.419	1111					225.837	459.787	730	0
8	211-61	495099.407	1853225.837	458.036	1600					225.837	460.001	816	0
9													
10													
11													
12													
13	1x1												
14	5pts	1-6-216-211...	1-6-61-66-33							26-121			
15													
16	1-1	495099.407	1852225.837	457.332	1					225.837	457.332	1	0
17	6-6	496099.407	1852225.837	457.757	1600					225.837	457.757	1600	0
18	216-66	496099.407	1853225.837	458.419	1111					225.837	459.787	730	0
19	211-61	495099.407	1853225.837	458.036	1600					225.837	460.001	816	0
20	108-33	495499.407	1852725.837	457.513	2500					225.837	458.059	1479	0
21													
22													

Fig 6.9 Enter the dependent by shading the height column.

Enter the dependent by shading the height column.

Enter the independent by shading the coordinates (E and N) columns.

Enter the signal columns.

Enter the noise column.

Press OK.

Area Size (km)	Grid Figure	ANOVA Figure	No. of data points	Computer Mathematical Model
				Residual(m)
1X1	6.2a	6.12a	4	0.000159
	6.2b	6.12b	5	0.069757
			6	0.036254
	6.2d	6.12c	6	0.034233
1X2	6.3a	6.13a	4	0.027454
	6.3b	6.13b	5	0.440888
	6.3c	6.13c	6	0.486649
			6	0.490078
2X1	6.4a	6.14a	4	0.02525
	6.4b	6.14b	5	0.276836
	6.4c	6.14c	6	0.425709
			6	
2X2	6.5a	6.15a	4	0.317741
	6.5b	6.15b	5	0.883924
	6.5c	6.15c	6	1.471481
	6.5d	6.15d	6	1.363111
2X4	6.6a	6.16a	4	1.325778
	6.6b	6.16b	5	3.153142
	6.6c	6.16c	6	3.567449
	6.6d	6.16d	6	2.814597
4X2	6.7a	6.17a	4	0.089396
	6.7b	6.17b	5	2.543554
	6.7c	6.17c	6	1.981809
	6.7d	6.17d	6	1.607955
4X4	6.8a	6.18a	4	0.656096
	6.8b	6.18b	5	1.725409
	6.8c	6.18c	6	5.608839
	6.8d	6.18d	6	5.402513

Table 6.4-Comparison of residual according to no. of data points

Area Size (km)	Grid Figure	ANOVA Figure	No. of data points	Polynomial	Computer Mathematical Model
				Residual(m)	Residual(m)
1X1	6.2a	6.12a	4	0.12	0.00
	6.2b	6.12b	5	0.16	0.07
	6.2c	6.12c	6	0.12	0.04
	6.2d	6.12d	6	0.18	0.03
1X2	6.3a	6.13a	4	0.37	0.03
	6.3b	6.13b	5	0.40	0.44
	6.3c	6.13c	6	0.39	0.49
	6.3d	6.13d	6	0.39	0.49
2X1	6.4a	6.14a	4	0.47	0.03
	6.4b	6.14b	5	0.53	0.28
	6.4c	6.14c	6	0.52	0.43
		6.14d	6		
2X2	6.5a	6.15a	4	2.05	0.32
	6.5b	6.15b	5	2.09	0.88
	6.5c	6.15c	6	2.05	1.47
	6.5d	6.15d	6	2.08	1.36
2X4	6.6a	6.16a	4	3.08	1.33
	6.6b	6.16b	5	3.08	3.15
	6.6c	6.16c	6	3.09	3.57
		6.16d	6		2.81
4X2	6.7a	6.17a	4	3.22	0.09
	6.7b	6.17b	5	3.22	2.54
	6.7c	6.17c	6	3.23	1.98
		6.17d	6		1.61
4X4	6.8a	6.18a	4	5.32	0.66
	6.8b	6.18b	5	5.32	1.73
	6.8c	6.18c	6	5.32	5.61
	6.8d	6.18d	6	5.32	5.40

Table 6.5-Comparison of residuals between polynomial and mathematical model.

6.9 Comparison of Results

To compare between Least Squares surface fitting and the mathematical modeling program surface fitting see Table 6.5

From Table (6.3), identical residuals were obtained by applying the polynomials given by equations (6.1…6.4) for all test areas, which means that there is no difference between the terms.

An area 1kmx1km of Figure 6.2a (4points) and Figure 6.2c (6points) is suitable for prediction of heights and it gives reasonably accurate results.

For an area 1x2 km (extending North - South), 4points located at the corners is the better configuration.

For an area 2x1 km (extending east - west), 4points located at corners is preferred.

For area 2x2 2km towards east and 2km towards north 4points located at corners and 5points (four points located at the corners and one point on the middle of the area) 4 points is better than 5 points.

For an area 2x4 km (2km towards east and 4km towards north) all points give bad results and so also for an area 4x4 km (4km towards east and 4km towards west).

For an area 4x2 (4km towards east and 2km towards north) 4points located at the corners is better to be used.

From the above results and for precise works, an area of 1x1km is the best.

CHAPTER SEVEN

Conclusions and recommendations

7.1 Conclusions

For the future, the use of terrain modeling methods will undoubtedly continue to develop and expand, particularly with continued improvements in the price performance ratio of computer systems. National and regional terrain databases based on (existing topographic maps, are now being developed in many parts of the world, and these will play an increasingly important role in terrain visualization during the preliminary planning stages of engineering projects. For small site, however, the primary source of terrain data is likely to continue to be directly measured spot heights.

For engineering work contouring is very important. To draw contour you need a corrected orthometric height referenced to local main sea level (Geoid).

I found that orthometric height costs more by direct leveling methods. To avoid that costing and time consuming job, Global Positioning System (GPS) ellipsoidal height should be done.

To convert the ellipsoidal height to orthometric height, we need many three dimensional control points with corrected height and known datum.

Sometimes in the office work surveyors need height points and discover that they have to go back field.

Till then Surveyors solve the problem in the field, when they come to the office and need more points they have no choice without going to the field again.

This mathematical model solved this problem, by picking out the height of points, when entering the horizontal coordinates (X, Y) of the desired point.

The surface fitting equation

$$H_i = a_1 + a_2X + a_3Y$$

is adequate for production of heights in area 1kmX1km .

The least squares prediction model, using a covariance matrix developed from an empirical function, gives the best results (minimum errors at check points).

The method predicts heights for drawing contour plans with acceptable contour intervals.

From the test it was found that the least number of data points required to model the area is four points located at the corners.

It was found that the shape of the area and the direction of the slope are very important.

The east - west direction and vice versa is the better direction for running the level line.

It gave results for control point’s data within the allowable error.

The final result is that the degree of perfection used with the field data is equal to the degree of perfection obtained by the modeling program and both are satisfying the solution in least squares surface fitting sense.

Finally the developed program satisfies the factors of quality, Economic, accuracy, reliability (internal and external).

7.2 Recommendations:-

1- Information about the surface topography should be accounted for, in addition to the distance and direction of the lines.

2- For GPS (RTK) it is important to build redundancy into a survey by, for example, occupying stations more than once.

3- The starting coordinates used in processing should be very accurate, avoiding the errors in the ephemeris that can affect the overall quality of the base line stations.

4- Information such as, starting time, navigated position, point number, antenna height are useful in processing.

REFERENCES

Ambrus Kenyeres, "Technology Development for GPS Heighting in Hungary", Paper presented at the 5th International Seminar on GPS in Central Europe 5-7 May, I 999, Penc, Hugary.

Anonym, 1999. Golden Software, Surfer8, and User's Guide: Contouring and 3D surface mapping for scientist and engineers, Colorado, USA.

Baarda, W.(1973), S-transformaions and Criterion Matrices, Neth. Geoid. Com on Geodesy. Delft.

Bomford, G. (1980) Geodesy, Fourth Edition, Oxford University Press, Oxford, United Kingdom, 855 pp.

Bugayevskiy & Snyder 1995, Qihe, Snyder & Tobler 2000, Snyder 1987.

Cooper,M.A.R (1974), Fundamentals of survey measurement and Analysis,Crosby Lockwood Staples.

Cooper MAR, Cross P.A. (1988), Statistical Concepts and their Application in Photogrammetry and Surveying, paper published in the photogrammetric record vol XII no. 71.

Cross PA (1983), Advanced Least Squares as Applied to The position Fixing, Working Paper no 6, North East London Polytechnic, department of land surveying.

Defense Mapping Agency. Datums, Ellipsoids, Grids , and Reference Systems , DMA TM 8358.1.

Ellmann A. (2005a), Computation of three stochastic modifications of Stokes's formula for regional geoid determination. Computers and Geosciences, 31/6, pp. 742-755.

Fagir, AH. (1984), Covariance Matrices: their structure and application to the optimal design of geodetic networks, PhD thesis, Department of Land Surveying, North East London Polytechnic, London, England.

Fotopoulos, G. (2003), An Analysis on the Optimal combination of Geoid, Orthometric and Ellipsoidal Height Data. PhD Thesis, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, UCGE Reports Number 201 85.

Fuad, Kassim.A (1986), An Evaluation of Three Prediction Techniques for the prediction of Gravity Anomalies in Canada, University of New Brunswick ,Fredericton.

Gulsen, Taskin ,Hasan Saygin,Multin Demiralp,Mustafa Yanalak(2002), Least Squares Curve Fitting via High Dimensional Model Representation for Digital elevation Model, paper presented at Informatics Institute, Istanbul Technical University.

Heiskanen, W.A. and Moritz, H. (1967), Physical Geodesy, W.H. Freeman and Company, San Francisco, USA, 364 pp.

Ibrahim A (1993), Interpretation of Gravity and magnetic data from the Central African Rift system [Sudan]: (Ph.D. Thesis), University of Leeds.

Ibrahim A.A.(2009),Structure & Application of Covariance Functions in Geodesy, Ph.D. Thesis, Department of Surveying Engineering, Sudan University of Science & Technology,Khartoum, Sudan.

Ibrahim A.M. (1984), The use of Eigen Values in the Analysis of Geodetic Networks, University of London.

Kotsakis C, Sideris MG (1999), On the adjustment of combined GPS/levelling/geoid networks. Journal of Geodesy, Vol 73(8), 412-421.

Krakiwsky, E.J. (1965), Heights, Master of Science Thesis, Department of Geodetic Science and Surveying, Ohio State University, Columbus, USA, 157 pp.

Leick, Alfred (1990), GPS Satellite Surveying, John Wei1y & sons.

Marmer, H. (1951), Tidal datum planes, Technical Report Special Publication No. 135, NOAA National Ocean Service, U.S. Coast and Geodetic Survey.

Martin, G.J., 2005. All possible worlds: A history of geographical ideas. OUP Catalogue.

Maximenko, N.A. and Niiler, P.P., 2005. Hybrid decade-mean global sea level with mesoscale resolution. Recent advances in marine science and technology, pp.55-59.

Mertikas, S.P., 2011. Geodesy, Ground Positioning and Leveling. In Encyclopedia of Solid Earth Geophysics (pp. 316-323). Springer Netherlands.

Meyer, T.H., Roman, D.R. and Zilkoski, D.B., 2006. What does height really mean? Part I: Introduction. Surveying and Land Information Science, 66(2), pp.127-137.

Moritz, H. (1976) ,Covariance Functions in Least Squares Collocation, Technical Report, Ohio University

Niethammer, T. (1932), Levelling and weight as means for the computation of true sea level heights, Schweizersche Geodatische Kommission, Berne, 76 pp.

Paul (1973), A method of evaluating the truncation error coefficients for geoidal height, Bulletin Geodesique, Vol. 47, No.4, pp. 413-425, doi: 10.1007/BF0252l951.

Paul MK (1973), A method for evaluating the truncation error coefficients for geoid heights. Bull Geo 110: 413-425.

Robinson, Arthur Howard, 1995. Early Thematic Mapping in the History of Cartography .University of Chicago Press, 266pp

Smith, J.R., 1997. Introduction to geodesy: the history and concepts of modern geodesy (Vol. 1). John Wiley & Sons.

Strang, G. (1980), Linear Algebra and Applications, Academic Press ,414 pp. Oliver Schabenberger, Carol A-Gotway (2005), Statistical Methods for "Spatial Analysis, Champman & CRC press.

Taha F. M.(2008), Determination of Orthometric Height Using Differential GPS Positioning,M.Sc.Thesis,Department of surveying, Sudan university of Science & Technology,Khartoum,Sudan.

Tenzer, R., Vani_ek, P., Santos, M., Featherstone, W.E. and Kuhn, M. (2005) ,The rigorous determination of orthometric heights, Journal of Geodesy, Vol. 78,No. 1-3, pp. 82-92, doi: 10.1007/s00190-005-0445-2.

Teunissen, P.J.G. (1985): Quality Control in Geodetic Networks. In: Optimization and Design of Geodetic Networks, Springer Verlag, pp.526-547.

The International Centre for Global Earth Models, ttp://icgem.gfz- potsdam .dell/ICGEM/ evaluation/evaluation.html).

Thomas H. Meyer, Daniel R. Roman, What Does Height Really Mean? Part IV: GPS Heighting Paper motivated by the National Geodetic Survey's (NGS).

T.J.M. Kennie and G. Petrie, Engineering Surveying Technology, John Wiley & Sons, Inc. New York.

Torge, H.W., 1996. The International Association of Geodesy (IAG)-More than 130 years of international cooperation. Journal of Geodesy, 70(12), pp.840-845.

Wilson, C., 1996, Assessment of Two Interpolation Methods, Inverse Distance Weighting and Geostatistical Kriging, University of Ottawa, course # geg-5306, Canada.

Wolf R, Charles, D, Ghilani (1996), Adjustment Computations, John Wiley &sons New York.

Zhilin Li, Qing Zhu (2005), Digital Terrain Model; Principles and Methodology, Champman & CRC press.

APPENDIX A

THE PROGRAM CODE

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace IbrahimPHD.Matrices

{


    public  class NzMatrix

    {

        private  double[,] _matrix;

        private int rowCount = 0;

        private int columnCount = 0;


        public  Double this[int row, int column]

        {

            get { return _matrix[row,column]; }

            set { _matrix[row,column] = value; }

        }


        public NzMatrix(double[,] values)

        {

            _matrix = values;

            rowCount = _matrix.GetLength(0);

            columnCount = _matrix.GetLength(1);
        }
        public NzMatrix(int rows, int columns)
        {
            double[,] values = new Double[rows,columns] ;
            _matrix = values;
            rowCount = rows;
            columnCount = columns;
        }
        public int ColumnCount { get { return  columnCount ; } }
        public int RowCount { get { return rowCount  ; } }

        public static implicit operator NzMatrix(Double[,] dataArray)
        {
            return new NzMatrix(dataArray);
        }

        public static NzMatrix operator +(NzMatrix matrix1, NzMatrix matrix2)
        {
              return NzMatrix.Addition(matrix1, matrix2);
        }
        protected static NzMatrix Addition(NzMatrix matrix1, NzMatrix
matrix2)
        {
            if ((matrix1.ColumnCount != matrix2.ColumnCount) ||
(matrix1.RowCount != matrix2.RowCount))
            {
                throw new System.Exception("IbrMatrix dimensions donot
agree");
            }
            NzMatrix result = new
NzMatrix(matrix1.RowCount,matrix1.ColumnCount) ;
```

```
            for (int i = 0; i < matrix1.RowCount  ; i++)
                for (int j = 0; j < matrix1.ColumnCount; j++)
                    result[i, j] = matrix1[i, j] + matrix2[i, j];

            return result;
        }

        public static NzMatrix operator -(NzMatrix matrix1, NzMatrix matrix2)
        {
            return NzMatrix.Subtraction(matrix1, matrix2);
        }
        protected static NzMatrix Subtraction(NzMatrix matrix1, NzMatrix
matrix2)
        {
            if ((matrix1.ColumnCount != matrix2.ColumnCount) ||
(matrix1.RowCount != matrix2.RowCount))
            {
                throw new System.Exception("IbrMatrix dimensions donot
agree");
            }
            NzMatrix result = new NzMatrix(matrix1.RowCount,
matrix1.ColumnCount);

            for (int i = 0; i < matrix1.RowCount; i++)
                for (int j = 0; j < matrix1.ColumnCount; j++)
                    result[i, j] = matrix1[i, j] - matrix2[i, j];

            return result;
        }

        public static NzMatrix operator /(NzMatrix matrix, Double scalar)
        {
            return NzMatrix.Division(matrix, scalar);
        }
        protected static NzMatrix Division(NzMatrix matrix, Double scalar)
        {

            NzMatrix result = new NzMatrix(matrix.RowCount,
matrix.ColumnCount);
            for (int i = 0; i < matrix.RowCount; i++)
                for (int j = 0; j < matrix.ColumnCount; j++)
                    result[i, j] = matrix[i, j] / scalar;

            return result;
        }



        public virtual NzMatrix StdDeviation { get { return
NzMatrix.StdDv(this); } }
        protected static NzMatrix StdDv(NzMatrix matrix)
        {
            int n = matrix.RowCount;
            int k = matrix.ColumnCount;
            Double[,] columnSum = new Double[1, k];
            Double[,] columnMean = new Double[1, k];
            Double[,] StD = new Double[1, k];
            Double[,] sumOf_OminusM = new Double[1, k];
            for (int j = 0; j < k; j++)
            {
                columnSum[0, j] = 0;
                for (int i = 0; i < n; i++)
                {
                    columnSum[0, j] = columnSum[0, j] + matrix[i, j];
                }
                columnMean[0, j] = columnSum[0, j] / n;
                sumOf_OminusM[0, j] = 0;
                for (int i = 0; i < n; i++)
                {
                    sumOf_OminusM[0, j] = sumOf_OminusM[0, j] +
Math.Pow(matrix[i, j] - columnMean[0, j], 2);
                }
                StD[0, j] = Math.Sqrt(sumOf_OminusM[0, j] / (n - 1));
            }
            NzMatrix result = new NzMatrix(matrix.RowCount, 0);
            result = StD;
            return result;
        }

        public virtual NzMatrix Mean { get { return NzMatrix.mean(this); } }
        protected static NzMatrix mean(NzMatrix matrix)
        {
            int n = matrix.RowCount;
            int k = matrix.ColumnCount;
            Double[,] columnSum = new Double[1, k];
            Double[,] columnMean = new Double[1, k];
            for (int j = 0; j < k; j++)
            {
                columnSum[0, j] = 0;
                for (int i = 0; i < n; i++)
                {
                    columnSum[0, j] = columnSum[0, j] + matrix[i, j];
                }
                columnMean[0, j] = columnSum[0, j] / n;
```

```
    }
    NzMatrix result = new NzMatrix(matrix.RowCount, 0);
    result = columnMean;
    return result;
}
public virtual NzMatrix CM { get { return NzMatrix.cm(this); } }
protected static NzMatrix cm(NzMatrix matrix)
{
    NzMatrix result = new NzMatrix(matrix.RowCount, 0);
    /*
    int n = matrix.RowCount;
    int k = matrix.ColumnCount;
    Double[,] columnSum = new Double[1, k];
    Double[,] columnMean = new Double[1, k];
    for (int j = 0; j < k; j++)
    {
        columnSum[0, j] = 0;
        for (int i = 0; i < n; i++)
        {
            columnSum[0, j] = columnSum[0, j] + matrix[i, j];
        }
        columnMean[0, j] = columnSum[0, j] / n;
    }
    result = columnMean;
    */
   //////// CMArrayyy = VarYYY * (XX.Transposed * WW * XX).Inverse;

    return result;

}
```

```
        public static double[,] Weight(NzMatrix weight)
        {

            double[,] result = new double[weight.RowCount, weight.RowCount];
            Double sumW = 0;
            for (int i = 0; i < weight.RowCount; i++)
            {
                sumW = sumW + weight[i, 0];
            }

            for (int i = 0; i < weight.RowCount; i++)
            {
                for (int j = 0; j < weight.RowCount; j++)
                {
                    result[i, j] = 0;
                }
                result[i, i] = weight[i, 0] * weight.RowCount / sumW;
            }
            return result;
        }

        public static Double[,] GetLinearCorCof(NzMatrix cm)
        {
            Double[,] lccArray = new Double[cm.ColumnCount, cm.ColumnCount];
// Linear correlation coefficients
            for (int i = 0; i < cm.ColumnCount; i++)
            {
                for (int j = 0; j < cm.ColumnCount; j++)
                {
                    lccArray[i, j] = cm[i, j] / Math.Sqrt(cm[i, i] * cm[j,
j]);
```

```
                }
            }
            return lccArray;
        }

        public static NzMatrix operator *(NzMatrix matrix1, NzMatrix matrix2)
        {
            return NzMatrix.Multiplication(matrix1, matrix2);
        }
        public static NzMatrix operator *(double scalar, NzMatrix matrix)
        {
            return NzMatrix.Multiplication(matrix, scalar);
        }

        public static NzMatrix operator *(NzMatrix matrix, double scalar)
        {
            return NzMatrix.Multiplication(matrix, scalar);
        }

        protected static NzMatrix Multiplication(NzMatrix matrix1, NzMatrix
matrix2)
        {
            if (matrix1.ColumnCount != matrix2.RowCount)
                throw new ArithmeticException("Number of columns in first
matrix does not equal number of rows in second matrix.");

            NzMatrix result = new NzMatrix(matrix1.RowCount,
matrix2.ColumnCount);

            for (int j = 0; j < result.RowCount; j++)
                for (int i = 0; i < result.ColumnCount; i++)
                {
                    Double value = 0;
                    for (int k = 0; k < matrix2.RowCount; k++)
                        value += matrix1[j, k] * matrix2[k, i];
                    result[j, i] = value;
                }
            return result;
        }
        protected static NzMatrix Multiplication(NzMatrix matrix, Double 
scalar)
        {
            NzMatrix result = new NzMatrix(matrix.RowCount, 
matrix.ColumnCount);

            for (int i = 0; i < matrix.RowCount; i++)
                for (int j = 0; j < matrix.ColumnCount; j++)
                    result[i, j] = matrix[i, j] * scalar;
            return result;
        }

        public virtual NzMatrix Transposed { get { return 
NzMatrix.Transpose(this); } }
        protected static NzMatrix  Transpose(NzMatrix matrix)
        {
            NzMatrix result =  new NzMatrix(matrix.ColumnCount, 
matrix.RowCount);
            for (Int32 i = 0; i < matrix.RowCount; i++)
            {
                for (Int32 j = 0; j < matrix.ColumnCount; j++)
                {
                    result[j, i] = matrix[i, j];
                }
            }
            return result;
        }

        public NzMatrix Inverse { get { return NzMatrix.Invert(this); } }

        protected static NzMatrix Invert(NzMatrix matrix)
        {
            double[,] a = matrix._matrix;
            int ro = a.GetLength(0);
            int co = a.GetLength(1);
            try
            {
                if (ro != co) { throw new System.Exception(); }
            }
            catch { Console.WriteLine("Cannot find inverse for an non square
matrix"); }

            int q; double[,] b = new double[ro, co]; double[,] I = eyes(ro);
            for (int p = 0; p < ro; p++) { for (q = 0; q < co; q++) { b[p, q]
= a[p, q]; } }
            int i; double det = 1;
            if (a[0, 0] == 0)
            {
                i = 1;
                while (i < ro)
                {
                    if (a[i, 0] != 0)
                    {
                        NzMatrix.interrow(a, 0, i);
                        NzMatrix.interrow(I, 0, i);
                        det *= -1;
                        break;
                    }
                    i++;
                }
            }
            det *= a[0, 0];
            NzMatrix.rowdiv(I, 0, a[0, 0]);
            NzMatrix.rowdiv(a, 0, a[0, 0]);
            for (int p = 1; p < ro; p++)
            {
                q = 0;
                while (q < p)
                {
                    NzMatrix.rowsub(I, p, q, a[p, q]);
                    NzMatrix.rowsub(a, p, q, a[p, q]);
                    q++;
                }
                if (a[p, p] != 0)
                {
                    det *= a[p, p];
                    NzMatrix.rowdiv(I, p, a[p, p]);
                    NzMatrix.rowdiv(a, p, a[p, p]);
                }
                if (a[p, p] == 0)
                {
                    for (int j = p + 1; j < co; j++)
                    {
                        if (a[p, j] != 0)
                        {
```

```
                            throw new System.Exception("Unable to deteremine
the Inverse");
                        }
                    }

                }
            }
            for (int p = ro - 1; p > 0; p--)
            {
                for (q = p - 1; q >= 0; q--)
                {
                    NzMatrix.rowsub(I, q, p, a[q, p]);
                    NzMatrix.rowsub(a, q, p, a[q, p]);
                }
            }
            for (int p = 0; p < ro; p++)
            {
                for (q = 0; q < co; q++)
                {
                    a[p, q] = b[p, q];
                }
            }

            return (I);
        }
        static void rowdiv(double[,] a, int r, double s)
        {
            int co = a.GetLength(1);
            for (int q = 0; q < co; q++)
            {
```

```
        a[r, q] = a[r, q] / s;
    }
}
static void rowsub(double[,] a, int i, int j, double s)
{
    int co = a.GetLength(1);
    for (int q = 0; q < co; q++)
    {
        a[i, q] = a[i, q] - (s * a[j, q]);
    }
}
static double[,] interrow(double[,] a, int i, int j)
{
    int ro = a.GetLength(0);
    int co = a.GetLength(1);
    double temp = 0;
    for (int q = 0; q < co; q++)
    {
        temp = a[i, q];
        a[i, q] = a[j, q];
        a[j, q] = temp;
    }
    return (a);
}
static double[,] eyes(int n)
{
    double[,] a = new double[n, n];
    for (int p = 0; p < n; p++)
    {
        for (int q = 0; q < n; q++)
        {
            if (p == q)
            {
                a[p, q] = 1;
            }
            else
            {
                a[p, q] = 0;
            }

        }
    }
    return (a);
}


public override String ToString()
{
    double[,] matrix = this._matrix;

    int row = matrix.GetLength(0);
    int col = matrix.GetLength(1);

    String result = String.Empty;
    if (matrix.Length == 0) return "[empty]";
    result += String.Format("{0} Rows x {1} Columns\n", row, col);
    for (int i = 0; i < row; i++)
    {
        String rowStr = String.Empty;
        for (int j = 0; j < col; j++)
```

```
                {
                    rowStr += String.Format(" {0} ", matrix[i,
j].ToString());
                }
                result += String.Format("[{0}]\n", rowStr);
            }
            return result;
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Reflection;
using System.Windows.Forms;
using System.Runtime.CompilerServices;
using System.IO;
using IbrahimPHD.Matrices;
namespace IbrahimPHD
{

    public static class General
    {
        private static bool fileChanged = false;
        private static string fullFileName = string.Empty;

        public static NzMatrix Beta { get; set; }
        public static NzMatrix XAxis { get; set; }
        public static NzMatrix YAxis { get; set; }
        public static bool FileChanged
        {
            get
            {
                return fileChanged;
            }
            set
            {
                fileChanged = value;
            }
        }
        public static string FullFileName
        {
            get
            {
                return fullFileName;
            }
            set
            {
                fullFileName = value;
            }
        }
        public static string FileName
        {
            get
```

```
            {
                FileInfo file = new FileInfo(General.FullFileName);
                return file.Name;
            }
        }

        public static int GridRows
        {
            get
            {
                return 99;
            }
        }

        public static int GridColumns
        {
            get
            {
                return 26;
            }
        }
/*******************************************************************/
        public static void CopyToClipboard(DataGridView dGView)
        {
            //Copy to clipboard
            DataObject dataObj = dGView.GetClipboardContent();
            if (dataObj != null)
                Clipboard.SetDataObject(dataObj);
        }
        public static  void PasteClipboardValue(DataGridView dGView)
        {
            //Show Error if no cell is selected
            if (dGView.SelectedCells.Count == 0)
            {
                MessageBox.Show("Please select a cell", "Paste",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            //Get the starting Cell
            DataGridViewCell  startCell = GetStartCell(dGView);
            //Get the clipboard value in a dictionary
            Dictionary<int, Dictionary<int, string>> cbValue =
                    ClipBoardValues(Clipboard.GetText());

            int iRowIndex = startCell.RowIndex;
            foreach (int rowKey in cbValue.Keys)
            {
                int iColIndex = startCell.ColumnIndex;
                foreach (int cellKey in cbValue[rowKey].Keys)
                {
                    //Check if the index is within the limit
                    if (iColIndex <= dGView.Columns.Count - 1
                    && iRowIndex <= dGView.Rows.Count - 1)
                    {
                        DataGridViewCell cell = dGView[iColIndex, iRowIndex];

                        //Copy to selected cells if 'chkPasteToSelectedCells'
is checked
```

```
                        ///////////          if
((chkPasteToSelectedCells.Checked && cell.Selected) ||
(!chkPasteToSelectedCells.Checked))

                        cell.Value = cbValue[rowKey][cellKey];

                    }

                    iColIndex++;

                }

                iRowIndex++;

            }

        }


        private static DataGridViewCell GetStartCell(DataGridView dgView)

        {

            //get the smallest row,column index

            if (dgView.SelectedCells.Count == 0)

                return null;


            int rowIndex = dgView.Rows.Count - 1;

            int colIndex = dgView.Columns.Count - 1;


            foreach (DataGridViewCell dgvCell in dgView.SelectedCells)

            {

                if (dgvCell.RowIndex < rowIndex)

                    rowIndex = dgvCell.RowIndex;

                if (dgvCell.ColumnIndex < colIndex)

                    colIndex = dgvCell.ColumnIndex;

            }


            return dgView[colIndex, rowIndex];

        }
        private static Dictionary<int, Dictionary<int, string>>
ClipBoardValues(string clipboardValue)
        {
            Dictionary<int, Dictionary<int, string>>
            copyValues = new Dictionary<int, Dictionary<int, string>>();

            String[] lines = clipboardValue.Split('\n');

            for (int i = 0; i <= lines.Length - 1; i++)
            {
                copyValues[i] = new Dictionary<int, string>();
                String[] lineContent = lines[i].Split('\t');

                //if an empty cell value copied, then set the dictionary with
an empty string
                //else Set value to dictionary
                if (lineContent.Length == 0)
                    copyValues[i][0] = string.Empty;
                else
                {
                    for (int j = 0; j <= lineContent.Length - 1; j++)
                        copyValues[i][j] = lineContent[j];
                }
            }
            return copyValues;
        }
/*******************************************************************/
    }
    public static class ExtensionMethods
    {
        public static void DoubleBuffered(this DataGridView dgv, bool
setting)
        {
            Type dgvType = dgv.GetType();
            PropertyInfo pi = dgvType.GetProperty("DoubleBuffered",
BindingFlags.Instance | BindingFlags.NonPublic);
            pi.SetValue(dgv, setting, null);
        }

    }

    public class Range
    {
        int firstRow;
        int firstColumn;
        int lastRow;
        int lastColumn;
        public int FirstRow
        {
            get { return firstRow; }
            set { firstRow = value; }
        }
        public int LastRow
        {
            get { return lastRow; }
            set { lastRow = value; }
        }
        public int FirstColumn
        {
            get { return firstColumn; }
            set { firstColumn = value; }
```

```
}
public int LastColumn
{
    get { return lastColumn; }
    set { lastColumn = value; }
}
public int RowCount
{
    get { return lastRow - firstRow + 1; ; }
}
public int ColumnCont
{
    get { return lastColumn - firstColumn + 1; ; }
}
public Range(string txtRange)
{
    string[] str;
    string[] topLeftCell;
    string[] buttomRightCell;

    if (txtRange.Contains(':'))
    {
        str = txtRange.Split(':');
        topLeftCell = str[0].Split(',');
        buttomRightCell = str[1].Split(',');
    }
    else
    {
        topLeftCell = txtRange.Split(',');
        buttomRightCell = txtRange.Split(',');
```

```
            }
            firstRow = int.Parse(topLeftCell[1]) - 1;
            firstColumn = (int)char.Parse(topLeftCell[0]) - 65;
            lastRow = int.Parse(buttomRightCell[1]) - 1;
            lastColumn = (int)char.Parse(buttomRightCell[0]) - 65;
        }
    }


}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using IbrahimPHD.Matrices;


namespace IbrahimPHD
{
    public partial class FrmInput : Form
    {
        FrmGrid frmGrid;

        Boolean validatedData = true;

        public FrmInput(FrmGrid _frmGrid)

        {

            frmGrid = _frmGrid;


            InitializeComponent();

        }


        private void FrmInput_Load(object sender, EventArgs e)

        {

            frmGrid.matrixToolStripMenuItem.Enabled = false;

        }


        private void FrmInput_FormClosed(object sender, FormClosedEventArgs
e)

        {

            //NZ.IsVisibleFrmInput = false;

            frmGrid.matrixToolStripMenuItem.Enabled = true;

        }


        private void txtBox_Enter(object sender, EventArgs e)

        {

            frmGrid.dGInput.ClearSelection();

            txtDependent.Tag = string.Empty;

            txtIndependent.Tag = string.Empty;

            txtWeight.Tag = string.Empty;

            txtNoise.Tag = string.Empty;

            txtSignal.Tag = string.Empty;
            TextBox txtBox = (TextBox)sender;
            txtBox.BackColor = SystemColors.HotTrack;
            txtBox.ForeColor = SystemColors.Window;
            txtBox.Tag = "Focused";
            string txtRange = txtBox.Text;
            if (txtRange != string.Empty)
            {
                Range selectionRange = new Range(txtRange);
                frmGrid.dGInput.ClearSelection();
                for (int i = selectionRange.FirstRow; i <=
selectionRange.LastRow; i++)
                {
                    for (int j = selectionRange.FirstColumn; j <=
selectionRange.LastColumn; j++)
                    {
                        frmGrid.dGInput.Rows[i].Cells[j].Selected = true;
                    }
                }
            }
        }

        private double[,] selection(string txtRange)
        {
            double[,] dblArray = null ;
            if (txtRange != string.Empty)
            {
                Range selectionRange = new Range(txtRange);
                dblArray = new double[selectionRange.RowCount,
selectionRange.ColumnCont];

                frmGrid.dGInput.ClearSelection();
                int row = 0;

                int column = 0;

                double number;



                for (int i = selectionRange.FirstRow; i <=
selectionRange.LastRow; i++)

                {

                    for (int j = selectionRange.FirstColumn; j <=
selectionRange.LastColumn; j++)

                    {

                        //frmGrid.dGInput.Rows[i].Cells[j].Selected = true;

                        //if
(double.TryParse(frmGrid.dGInput.Rows[i].Cells[j].Value.ToString())

                        //if
(Double.TryParse(string.IsNullOrEmpty(frmGrid.dGInput.Rows[i].Cells[j].Value.
ToString()).ToString(), out number))

                        if
(Double.TryParse(frmGrid.dGInput.Rows[i].Cells[j].Value.ToString(), out
number))

                        {

                            dblArray[row, column] = number;

                        }

                        else

                        {

                            frmGrid.dGInput.Rows[i].Cells[j].Style.BackColor
= Color.Red ;

                            validatedData = false;

                        }

                        column = column + 1;

                    }

                    column = 0;

                    row = row + 1;

                }
```

```
            }

            return dblArray;

        }


        private void txtBox_Leave(object sender, EventArgs e)

        {

            frmGrid.dGInput.ClearSelection();

            txtDependent.Tag = string.Empty;

            txtIndependent.Tag = string.Empty;

            txtWeight.Tag = string.Empty;

            txtNoise.Tag = string.Empty;

            txtSignal.Tag = string.Empty;

            TextBox txtBox = (TextBox)sender;

            txtBox.BackColor = SystemColors.Window;

            txtBox.ForeColor = SystemColors.WindowText;

        }





        private void btnOk_Click(object sender, EventArgs e)

        {

            if (txtDependent.Text == string.Empty) { txtDependent.Focus();
return; }

            if (txtIndependent.Text == string.Empty) {
txtIndependent.Focus(); return; }

            switch(this.Tag.ToString())

            {

                case "OLS" :

                    break;

                case "WLS" :
                    if (txtWeight.Text == string.Empty) { txtWeight.Focus();
return; }

                    break;

                case "WLSC" :

                    if (txtNoise.Text == string.Empty) { txtNoise.Focus();
return; }

                    if (txtSignal.Text == string.Empty) { txtSignal.Focus();
return; }

                    break;

            }

            NzMatrix Dependent = null;

            NzMatrix Independent = null;

            NzMatrix Weight = null;


            double[,] weight = null;

            double[,] dependent = selection(txtDependent.Text);

            Dependent = dependent;


            double[,] _independent = selection(txtIndependent.Text);

            int rMax = _independent.GetLength(0);

            int cMax = _independent.GetLength(1);


            double[,] independent = new double[rMax, cMax + 1];

            // Fill the first column of independent with zeroes (for p = 0)
or ones (for p = 1), the rest with the data in the x-column(s)

            for (int i = 0 ; i < rMax ; i++)

            {

                independent[i, 0] = 1; // p;

            }


            for (int j = 1 ; j <= cMax ; j++)

            {
```

```
    for (int i = 0 ; i < rMax ; i++)
    {
        independent[i, j] = _independent[i, j - 1];
    }
}
Independent = independent;

if (txtWeight.Enabled == false)
{
    if (txtSignal.Enabled == true && txtNoise.Enabled == true)
    {
        double[,] noise = null;
        double[,] signal = null;
        noise = selection(txtNoise.Text);
        signal = selection(txtSignal.Text);
        NzMatrix Noise = noise;
        NzMatrix Signal = signal;
        Weight = Noise + Signal;

        weight = selection(txtWeight.Text);
    }
    else
    {
        weight = new double[rMax, 1];
        for (int i = 0; i < rMax; i++)
        {
            weight[i, 0] = 1; // p;
        }
        Weight = weight;
    }
```

```
            }
            else
            {
                weight = selection(txtWeight.Text);
                Weight = weight;
            }



            if (validatedData == true)
            {
                //Regress(dependent, independent, weight);
                Regress(Dependent, Independent, Weight);
                frmGrid.splitContainer.Panel2Collapsed = false;
                frmGrid.splitContainer1.Panel2Collapsed = false;
            }
            else
            {
                frmGrid.splitContainer.Panel2Collapsed = true;
                frmGrid.splitContainer1.Panel2Collapsed = false;

                MessageBox.Show("Input values not corrected!");
            }
            this.Close();
        }

        //public void Regress(Double[,] Dependent, Double[,] Independent, 
Double[,] Weight)
        public void Regress(NzMatrix Dependent, NzMatrix Independent, 
NzMatrix Weight)
        {
            NzMatrix Y = Dependent;
            NzMatrix X = Independent;
            NzMatrix W = NzMatrix.Weight(Weight);
            General.XAxis = X;
            General.YAxis = Y;

            //int n = Dependent.Length;
            //int k = Independent.Length / n;
            int n = Dependent.RowCount;
            int k = Independent.ColumnCount;



            NzMatrix XtWX = X.Transposed * W * X;
            NzMatrix XtWXi = XtWX.Inverse;

            NzMatrix XtWY = X.Transposed * W * Y;
            NzMatrix YtWY = Y.Transposed * W * Y;
            General.Beta = null;
            NzMatrix B = XtWXi * XtWY;  // B = (X' W X)" X' W Y  -------
Var(b) = (X_WX)−1(X_WΣ0WX)(X_WX)−1
            General.Beta = B;
            /////////////////////////////////////////Calculate The Regression
Equation///////////////////////////////////
            string RegressionEquation = string.Empty ;
            for (int i = 0; i < B.RowCount; i++)
            {
                if (i == 0)
                {
                    RegressionEquation = "y i = " + Math.Round(B[0, 0],
6).ToString();
                }
                else
                {
                    RegressionEquation = RegressionEquation + " + " +
Math.Round(B[i, 0], 6).ToString() + " x" + i.ToString();
                }
            }



/////////////////////////////////////////////////////////////////////////////
///////////////////////////////
            double SSRnz = Math.Round((Y.Transposed * W * Y)[0, 0] -
(B.Transposed * X.Transposed * W * Y)[0,0], 6);
            double varY = SSRnz / (n - k);   // The variance of y
            NzMatrix CMArray =  XtWXi * varY  ; // The Covariance Matrix


            NzMatrix LCC = NzMatrix.GetLinearCorCof(CMArray); // Linear
correlation coefficients

/////////////////////////////////////////////////////////////////////////////
///////////////////////////////




            //////////////////////////////////////////////ANOVA
Table///////////////////////////////////////////////////
            double sumDependent = 0 ;
            for (int i = 0; i < n; i++)
            {
                sumDependent = sumDependent + Dependent[i,0];
            }
            double Ybar = Math.Pow(sumDependent, 2) / n;
            double SSE = Math.Round(YtWY[0, 0] - (B.Transposed * XtWY)[0, 0],
6); // SSE = Y' Y - B'X'Y -- Sum of squares due to error
            double SSR = Math.Round( (B.Transposed * XtWY)[0, 0] - Ybar,6);
// SSR = B' X' Y - sum(Y)^2 / N  -- Sum of squares to due regression
            double SST = Math.Round( YtWY[0, 0] - Ybar,6) ; // Y' Y -
sum(Y)^2 / N -- Total sum of squares of dependent variable.
```

```
            double DF_Residuals = k - 1;
            double DF_Error = n - k ;
            double DF_Total = n - 1 ;
            double MSR = Math.Round( SSR / (k - 1) , 6) ;
            double MSE = Math.Round( SSE / (n - k),6) ;
            double Fstat = Math.Round( MSR / MSE , 6); // F statistics
            double R2 = Math.Round ( SSR / SST , 6); // Coefficient of 
determination
            double R2adj = Math.Round ( 1 - (1 - R2) * (n - 1) / (n - k) , 
6); // Adjusted value of R2




////////////////////////////////////OutPut///////////////////////////////////
///
            string cMArray = string.Empty;
            for (int i = 0; i < CMArray.RowCount; i++)
            {
                for (int j = 0; j < CMArray.ColumnCount; j++)
                {
                    cMArray = cMArray + "\t" + Math.Round(CMArray[i, j], 6);
                }
                cMArray = cMArray + "\r\n";

            }
            string lcc = string.Empty;
            for (int i = 0; i < LCC.RowCount; i++)
            {
                for (int j = 0; j < LCC.ColumnCount; j++)
                {
                    lcc = lcc + "\t" + Math.Round(LCC[i, j], 6);
                }
                lcc = lcc + "\r\n";
            }
            String result = string.Empty;
            result = "Results:\r\n\r\nThe Fitted Model is: \r\n" +
RegressionEquation +
                    "\r\n\r\n\r\n\r\nThe coefficient of determination for the
above model is : " + R2 +
                    "\r\n\r\n\r\nThe Adjusted value of The coefficient of
determination  is : " + R2adj +
                    "\r\n\r\n\r\n\r\nThe Covariance Matrix : \r\n" + cMArray
+ "\r\n\r\n"  ;



            String ANOVA = "\r\n\r\nANOVA Table\r\nModel\tSum of Squares\tdf
Mean Square\tF\tSig\r\n" +
            "----------------------------------------------------------------
----------------------------\r\n" +
            "Regression\t" + SSR + "\t\t" + DF_Residuals + "   " + MSR + "\t"
+ Fstat + "\r\n" +
            "Residual\t" + SSE + "\t\t" + DF_Error + "   " + MSE + "\r\n" +
            "Total\t" + SST + "\t\t" + DF_Total + "\r\n" +
            "----------------------------------------------------------------
----------------------------\r\n";

            frmGrid.txtModel.Text = RegressionEquation;
            //frmGrid.txtResult.Text = result + ANOVA;
            frmGrid.txtResult.Text = result ;
            frmGrid.btnANOVA.Tag = ANOVA;
            frmGrid.btnLCC.Tag = lcc;
            frmGrid.dGModel.Columns.Clear();
            frmGrid.dGModel.DoubleBuffered(true);

            //frmGrid.dGModel.BackgroundColor =
System.Drawing.Color.Aquamarine;

            //frmGrid.dGModel.Dock = System.Windows.Forms.DockStyle.Fill;

            for (int i = 0; i < B.RowCount; i++)

            {

                //char c = (char)(i + 65);

                frmGrid.dGModel.Columns.Add("x" + i, "x"  + i );

                frmGrid.dGModel.Columns[i].Tag = Math.Round(B[i, 0],
6).ToString();

            }

            frmGrid.dGModel.Rows.Add(1);

            frmGrid.dGModel.Columns[0].Visible = false;

            frmGrid.dGModel.RowHeadersVisible = false;

            frmGrid.dGModel.Rows[0].Cells[0].Value = 1;



/////////////////////////////////////////////////////////////////////////////
///

            /*

            ANOVAa,b

            Model             Sum of Squares  df        Mean Square         F
              Sig.

            -----------------------------------------------------------------
--------

            1Regression   545.885             2        272.943
3477.072     .000c

                Residual  .628          8        .078

                Total         546.513         10

            -----------------------------------------------------------------
--------

            a Dependent Variable: y
            b Weighted Least Squares Regression - Weighted by weight

            c Predictors: (Constant), x2, x1

             */
        }


        private void txtDependent_TextChanged(object sender, EventArgs e)
        {


        }



        private void btnCancel_Click(object sender, EventArgs e)
        {
            this.Close();
        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml;

using System.IO;

using IbrahimPHD;


namespace IbrahimPHD

{

    public partial class FrmGrid : Form

    {

        FrmInput frmInput = null;

        FrmMatricesOperations frmMatricesOperations = null ;

        private int sRow = -1;

        private int sColumn = -1;

        private int eRow = -1;

        private int eColumn = -1;

        private int startrow = -1;

        private int startcolumn = -1;

        private int endrow = -1;

        private int endcolumn = -1;

        public FrmGrid()

        {

            InitializeComponent();

            this.StartPosition = FormStartPosition.CenterScreen;

            //tStripStatusLabelSelection.Text = string.Empty;

        }


        private void FrmGrid_Load(object sender, EventArgs e)

        {

            lblTitle.BackColor = toolStrip1.BackColor;
            lblTitle.Left = this.Width / 2 - lblTitle.Width / 2;

            lblOwner.BackColor = toolStrip1.BackColor;
            lblSupervisor.BackColor = toolStrip1.BackColor;
            lblOwner.Left = this.Left - lblOwner.Width;
            lblSupervisor.Left =  this.Width   ;

            dGInput.DoubleBuffered(true);
            //dGModel.DoubleBuffered(true);
            dGInput.BackgroundColor = System.Drawing.Color.White;
            dGInput.Dock = System.Windows.Forms.DockStyle.Fill;
            //dGModel.BackgroundColor = System.Drawing.Color.Aquamarine;
            //dGModel.Dock = System.Windows.Forms.DockStyle.Fill;
            for (int i = 0; i < General.GridColumns; i++)
            {
                char c = (char)(i + 65);
                dGInput.Columns.Add(c.ToString(), c.ToString());
                dGInput.Columns[c.ToString()].SortMode = 
DataGridViewColumnSortMode.NotSortable;

                //dGModel.Columns.Add(c.ToString(), c.ToString());
            }
            dGInput.SelectionMode = 
DataGridViewSelectionMode.ColumnHeaderSelect;
            dGInput.Rows.Add(General.GridRows);

            //dGModel.Rows.Add(3);
            //dGModel.ColumnHeadersVisible = false;
            //dGModel.RowHeadersVisible = false;

            for (int row = 1; row <= General.GridRows ; row++)
```

```
    {
        dGInput.Rows[row - 1].HeaderCell.Value = row.ToString();
    }
    startNewFile();
}
private void startNewFile()
{
    splitContainer.Panel2Collapsed = true;
    splitContainer1.Panel2Collapsed = true;
    this.Text = "Least Squares - Untitled";
    General.FullFileName = string.Empty;
    dGInput.EndEdit() ;
    for (int i = 0; i < General.GridRows; i++)
        for (int j = 0; j < General.GridColumns; j++)
        {
            dGInput.Rows[i].Cells[j].Value = string.Empty;
            dGInput.Rows[i].Cells[j].Style.BackColor = Color.White;
        }


    dGInput.ClearSelection();
    General.FileChanged = false;
    txtResult.Text = string.Empty;

}
private void newfile(object sender, EventArgs e)
{
    dGInput.EndEdit();
    if (General.FileChanged == true)
    {
```

```
                DialogResult result = MessageBox.Show("Do you want to save
the changes you made to file?","",  MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Exclamation);

                switch (result)

                {

                    case DialogResult.Yes :

                        if (saveOperation(General.FullFileName) == true)

                        {

                            startNewFile();

                        }

                        else

                        {

                            return;

                        }

                        break;

                    case DialogResult.No:

                        startNewFile();

                        break;

                    case DialogResult.Cancel :

                        break;

                }

            }

            else

            {

                startNewFile();

            }

        }

        private void savefile(object sender, EventArgs e)

        {

            dGInput.EndEdit();

            saveOperation(General.FullFileName);
        }
        private bool saveOperation(string fullFileName)
        {
            General.FullFileName = fullFileName;
            if (fullFileName == string.Empty)
            {
                SaveFileDialog saveFileDialog = new SaveFileDialog();
                saveFileDialog.Filter = "ibr files (*.ibr)|*.ibr";
                saveFileDialog.FilterIndex = 2;
                saveFileDialog.RestoreDirectory = true;
                DialogResult result = saveFileDialog.ShowDialog();

                if (DialogResult.OK == result)
                {
                    General.FullFileName = saveFileDialog.FileName;
                }
                else if (DialogResult.Cancel == result)
                {
                    return false;
                }
            }

            using (BinaryWriter bw = new
BinaryWriter(File.Open(General.FullFileName, FileMode.Create)))
            {
                bw.Write(dGInput.Columns.Count);
                bw.Write(dGInput.Rows.Count);
                foreach (DataGridViewRow dgvR in dGInput.Rows)
                {
                    for (int j = 0; j < dGInput.Columns.Count; ++j)
                    {
                        object val = dgvR.Cells[j].Value;
                        if (val == null)
                        {
                            bw.Write(false);
                            bw.Write(false);
                        }
                        else
                        {
                            bw.Write(true);
                            bw.Write(val.ToString());
                        }
                    }
                }
            }
            this.Text = General.FileName;
            General.FileChanged = false;
            return true;
        }
        private void openfile(object sender, EventArgs e)
        {
            dGInput.EndEdit();
            if (General.FileChanged == true)
            {
                DialogResult result = MessageBox.Show("Do you want to save
the changes you made to file?", "", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Exclamation);
                switch (result)
                {
                    case DialogResult.Yes:
                        if (saveOperation(General.FullFileName) == true)
                        {
                            openOperation();
                            //splitContainer.Panel2Collapsed = true;
                        }
                        else
                        {
                            return;
                        }
                        break;
                    case DialogResult.No:
                        openOperation();
                        break;
                    case DialogResult.Cancel:
                        break;
                }
            }
            else
            {
                openOperation();
            }
        }
        private void openOperation()
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "ibr files (*.ibr)|*.ibr";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                string fullFileName = openFileDialog.FileName;
```

```
                try
                {
                    using (BinaryReader bw = new
BinaryReader(File.Open(fullFileName, FileMode.Open)))
                    {
                        int n = bw.ReadInt32();
                        int m = bw.ReadInt32();
                        for (int i = 0; i < m; ++i)
                        {
                            for (int j = 0; j < n; ++j)
                            {
                                dGInput.Rows[i].Cells[j].Value = null;
                                if (bw.ReadBoolean())
                                {
                                    dGInput.Rows[i].Cells[j].Style.BackColor
= Color.White;
                                    dGInput.Rows[i].Cells[j].Value =
bw.ReadString();
                                }
                                else bw.ReadBoolean();
                            }
                        }
                    }
                    General.FullFileName = fullFileName;
                    this.Text = General.FileName;
                    General.FileChanged = false;
                    dGInput.ClearSelection();
                    splitContainer.Panel2Collapsed = true;
                    splitContainer1.Panel2Collapsed = true;

                    txtResult.Text = string.Empty;
                    //this.Text = "Least Squares - Untitled";
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }
        private void exitproject(object sender, EventArgs e)
        {
            this.Close();
        }



        private void dGInput_CellEnter(object sender, 
DataGridViewCellEventArgs e)
        {
            if (dGInput.SelectedCells.Count == 1)
            {
                startrow = e.RowIndex;
                startcolumn = e.ColumnIndex;
                endrow = e.RowIndex;
                endcolumn = e.ColumnIndex;
            }
            else
            {
                endrow = e.RowIndex;
                endcolumn = e.ColumnIndex;
            }
            sRow = Math.Min(startrow, endrow);
            sColumn = Math.Min(startcolumn, endcolumn);
            eRow = Math.Max(startrow, endrow);
            eColumn = Math.Max(startcolumn, endcolumn);
            string range = string.Empty;

            ///////////////////////////



            // if (frmInput != null && frmInput.Visible ||
frmMatricesOperations != null && frmMatricesOperations.Visible)



            if (frmInput != null && frmInput.Visible)
            {
                if (sRow == eRow && sColumn == eColumn)
                {
                    //$C$4:$D$6
                    range = dGInput.Columns[sColumn].HeaderText + "," +
dGInput.Rows[sRow].HeaderCell.Value;
                }
                else
                {
                    range = dGInput.Columns[sColumn].HeaderText + "," +
dGInput.Rows[sRow].HeaderCell.Value + ":" + dGInput.Columns[eColumn].Name +
"," + dGInput.Rows[eRow].HeaderCell.Value;
                }
                if ((string)frmInput.txtDependent.Tag == "Focused")
                    frmInput.txtDependent.Text = range;
                if ((string)frmInput.txtIndependent.Tag == "Focused")
                    frmInput.txtIndependent.Text = (string)range;
                if ((string)frmInput.txtWeight.Tag == "Focused")

                    frmInput.txtWeight.Text = range;

                if ((string)frmInput.txtSignal.Tag == "Focused")

                    frmInput.txtSignal.Text = range;

                if ((string)frmInput.txtNoise.Tag == "Focused")

                    frmInput.txtNoise.Text = range;

            }



            if (frmMatricesOperations != null &&
frmMatricesOperations.Visible)

            {

                if (sRow == eRow && sColumn == eColumn)

                {

                    //$C$4:$D$6

                    range = dGInput.Columns[sColumn].HeaderText + "," +
dGInput.Rows[sRow].HeaderCell.Value;

                }

                else

                {

                    range = dGInput.Columns[sColumn].HeaderText + "," +
dGInput.Rows[sRow].HeaderCell.Value + ":" + dGInput.Columns[eColumn].Name +
"," + dGInput.Rows[eRow].HeaderCell.Value;

                }

                if ((string)frmMatricesOperations.txtMatrix1.Tag ==
"Focused")

                    frmMatricesOperations.txtMatrix1.Text = range;

                if ((string)frmMatricesOperations.txtMatrix2.Tag ==
"Focused")

                    frmMatricesOperations.txtMatrix2.Text = (string)range;

            }
```

```
    //tStripStatusLabelSelection.Text = range;
}
private void FrmGrid_SizeChanged(object sender, EventArgs e)
{
    switch (this.WindowState)
    {
        case FormWindowState.Maximized:
            foreach (Form frm in Application.OpenForms)
            {
                if (frm.TopMost == true)
                {
                    frm.Visible = true;
                }
            }
            break;
        case FormWindowState.Minimized:
            foreach (Form frm in Application.OpenForms)
            {
                if (frm.TopMost == true)
                {
                    frm.Visible = false;
                }
            }
            break;
        case FormWindowState.Normal:
            break;
        default:
```

```
                    break;
                }
        }

        private void dGInput_CellBeginEdit(object sender,
DataGridViewCellCancelEventArgs e)
        {
            General.FileChanged = true;
        }

        private void FrmGrid_FormClosing(object sender, FormClosingEventArgs
e)
        {
            if (General.FileChanged == true)
            {
                DialogResult result = MessageBox.Show("Do you want to save
the changes you made to file?", "", MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Exclamation);
                switch (result)
                {
                    case DialogResult.Yes:
                        if (saveOperation(General.FullFileName) == true)
                        {
                            startNewFile();
                        }
                        else
                        {
                            return;
                        }
                        break;
                    case DialogResult.No:
                        break;
                    case DialogResult.Cancel:
                        e.Cancel = true;
                        break;
                }
            }

        }

        private void dGInput_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
        {
            if (dGInput.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor
== Color.Red)
            {
                dGInput.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor
= Color.White;
            }
        }

        private void timerOwner_Tick(object sender, EventArgs e)
        {
            if (lblOwner.Left + lblOwner.Width == lblTitle.Left)
            {
                lblTitle.Visible = false;
            }
            else if (lblOwner.Left == lblTitle.Left + lblTitle.Width)
            {
                lblTitle.Visible = true;
            }
            if (lblOwner.Left > this.Width)

            {

                lblOwner.Left = this.Left  - lblOwner.Width;

            }


            if (lblSupervisor.Left == this.Left - lblSupervisor.Width)

            {

                lblSupervisor.Left = this.Width;

            }

            lblOwner.Left =  lblOwner.Left + 1;

            lblSupervisor.Left = lblSupervisor.Left - 1;

        }


        private void helpToolStripButton1_Click(object sender, EventArgs e)

        {


        }




        private void OLStsm_Click(object sender, EventArgs e)

        {

            splitContainer.Panel2Collapsed = true;

            splitContainer1.Panel2Collapsed = true;

            txtResult.Text = string.Empty;

            if (((Form)Application.OpenForms["frmInput"] == null))

            {

                frmInput = new FrmInput(this);

                frmInput.TopMost = true;

                frmInput.Tag = "OLS";
                frmInput.Text = "Ordinary Least Squares";
                frmInput.txtNoise.Enabled = false;
                frmInput.txtSignal.Enabled = false;
                frmInput.txtWeight.Enabled = false;
                frmInput.Show();
            }
            else
            {
                frmInput.BringToFront();
            }
        }

        private void WLStsm_Click(object sender, EventArgs e)
        {
            splitContainer.Panel2Collapsed = true;
            splitContainer1.Panel2Collapsed = true;
            txtResult.Text = string.Empty;
            if (((Form)Application.OpenForms["frmInput"] == null))
            {
                frmInput = new FrmInput(this);
                frmInput.TopMost = true;
                frmInput.Tag = "WLS";
                frmInput.Text = "Weighted Least Squares";
                frmInput.txtNoise.Enabled = false;
                frmInput.txtSignal.Enabled = false;
                frmInput.txtWeight.Enabled = true;
                frmInput.Show();
            }
            else
            {
```

```
                frmInput.BringToFront();
            }
        }

        private void CLStsm_Click(object sender, EventArgs e)
        {
            splitContainer.Panel2Collapsed = true;
            splitContainer1.Panel2Collapsed = true;
            txtResult.Text = string.Empty;
            if (((Form)Application.OpenForms["frmInput"] == null))
            {
                frmInput = new FrmInput(this);
                frmInput.Tag = "WLSC";
                frmInput.Text = "Geoidal Separation : N = h - H";
                frmInput.TopMost = true;
                frmInput.txtNoise.Enabled = true;
                frmInput.txtSignal.Enabled = true;
                frmInput.txtWeight.Enabled = false;
                frmInput.Show();
            }
            else
            {
                frmInput.BringToFront();
            }
        }

        private void aboutTheAuthorToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            AboutAuthor aboutAuthor = new AboutAuthor();
            aboutAuthor.ShowDialog();
        }

        private void FrmGrid_Resize(object sender, EventArgs e)
        {
            lblTitle.Left = this.Width / 2 - lblTitle.Width / 2;
            lblOwner.Left = this.Left - lblOwner.Width;
            lblSupervisor.Left = this.Width;
        }



        private void btnCalculateYi_Click(object sender, EventArgs e)
        {
            double number;
            double summation = 0 ;
            Boolean validatedData = true;
            for (int i = 0 ; i < dGModel.Columns.Count ; i++)
            {
                if (dGModel.Rows[0].Cells[i].Value == null)
                {
                    dGModel.Rows[0].Cells[i].Value = 0 ;
                }
                if
(Double.TryParse(dGModel.Rows[0].Cells[i].Value.ToString(), out number))
                {
                    //dblArray[row, column] = number;
                    summation = summation + (number *
double.Parse(dGModel.Columns[i].Tag.ToString()));
                }
                else
                {
                    dGModel.Rows[0].Cells[i].Style.BackColor = Color.Red;
                    validatedData = false;
                }
            }
            if (validatedData == true)
            {
                txtYi.Text = summation.ToString();
            }
            else
            {
                txtYi.Text = "Correct the input data";
            }
            dGModel.ClearSelection();
        }

        private void dGModel_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
        {
            if (dGModel.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor
== Color.Red)
            {
                dGModel.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor
= Color.White;
            }
        }

        private void matrixToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            /*
            splitContainer.Panel2Collapsed = true;
            splitContainer1.Panel2Collapsed = true;

            txtResult.Text = string.Empty;

            */

            if (((Form)Application.OpenForms["frmInput"] == null))

            {

                frmMatricesOperations = new FrmMatricesOperations(this);

                frmMatricesOperations.TopMost = true;

                frmMatricesOperations.Text = "Matrices Operations";

                frmMatricesOperations.Show();

            }

            else

            {

                frmMatricesOperations.BringToFront();

            }

        }


        private void aNOVATableToolStripMenuItem_Click(object sender,
EventArgs e)

        {

            FrmANOVA frmANOVA = new FrmANOVA();

            frmANOVA.ShowDialog();

        }


        private void btnANOVA_Click(object sender, EventArgs e)

        {

            FrmANOVA frmANOVA = new FrmANOVA();

            frmANOVA.Text = "ANOVA Table";

            frmANOVA.txtANOVA.Text = btnANOVA.Tag.ToString();

            frmANOVA.ShowDialog();

        }
        private void btnLCC_Click(object sender, EventArgs e)

        {

            FrmANOVA frmANOVA = new FrmANOVA();

            frmANOVA.Text = "The Linear Correlation Coefficients";

            frmANOVA.txtANOVA.Text = "\r\n\r\n\r\n" + btnLCC.Tag.ToString();

            frmANOVA.ShowDialog();

        }


        private void orthometricHeightToolStripMenuItem1_Click(object sender,
EventArgs e)

        {

            //FrmOrthometric frmOrthometric = new FrmOrthometric();

            //frmOrthometric.ShowDialog();

            splitContainer.Panel2Collapsed = true;

            splitContainer1.Panel2Collapsed = true;

            txtResult.Text = string.Empty;

            if (((Form)Application.OpenForms["frmInput"] == null))

            {

                frmInput = new FrmInput(this);

                frmInput.Tag = "WLSC";

                frmInput.Text = "DENSIFICATION OF ORTHOMETRIC HEIGHTS";

                frmInput.TopMost = true;

                frmInput.txtNoise.Enabled = true;

                frmInput.txtSignal.Enabled = true;

                frmInput.txtWeight.Enabled = false;

                frmInput.Show();

            }

            else

            {
```

```
            frmInput.BringToFront();
        }
    }

    private void btnYi_Click(object sender, EventArgs e)
    {
        FrmCalculateTable frmCalculateTable = new FrmCalculateTable();
        frmCalculateTable.txtModel.Text = txtModel.Text;
        frmCalculateTable.ShowDialog();
    }

    private void btnChart_Click(object sender, EventArgs e)
    {
        FrmChart frmChart = new FrmChart();
        frmChart.ShowDialog();
    }

    private void cutToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Copy to clipboard
        General.CopyToClipboard(dGInput);

        //Clear selected cells
        foreach (DataGridViewCell dgvCell in dGInput.SelectedCells)
            dgvCell.Value = string.Empty;
    }

    private void copyToolStripMenuItem_Click(object sender, EventArgs e)
    {
        General.CopyToClipboard(dGInput);
        }

        private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Perform paste Operation
            General.PasteClipboardValue(dGInput);
            General.FileChanged = true;
        }

        private void dGInput_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            if (dGInput.SelectedCells.Count > 0)
                dGInput.ContextMenuStrip = contextMenuStrip1;
        }

        private void tsStandard_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
        {

        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;


namespace IbrahimPHD

{

    public partial class FrmCalculateTable : Form

    {

        public FrmCalculateTable()

        {

            InitializeComponent();

        }


        private void FrmCalculateTable_Load(object sender, EventArgs e)

        {

            dGModel.Columns.Clear();

            dGModel.DoubleBuffered(true);

            dGModel.Columns.Add("y" , "Y");

            dGModel.Columns[0].DefaultCellStyle.BackColor = Color.LightGreen;


            for (int i = 1; i < General.Beta.RowCount; i++)

            {

                dGModel.Columns.Add("x" + i, "x" + i);

                dGModel.Columns[i].Width = 90 ;

            }

            dGModel.Rows.Add(1);

            dGModel.Columns[0].ReadOnly = true;

            //dGModel.RowHeadersVisible = false;
        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void dGModel_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
        {
            double number;
            double summation = General.Beta[0, 0];
            Boolean validatedData = true;
            for (int i = 1; i < dGModel.Columns.Count; i++)
            {
                if (dGModel.Rows[e.RowIndex].Cells[i].Value == null)
                {
                    dGModel.Rows[e.RowIndex].Cells[i].Value = 0;
                }
                if
(Double.TryParse(dGModel.Rows[e.RowIndex].Cells[i].Value.ToString(), out
number))
                {
                    //MessageBox.Show(General.Beta[i, 0].ToString());
                    summation = summation + (number * General.Beta[i, 0]);
                }
                else
                {
                    dGModel.Rows[e.RowIndex].Cells[i].Style.BackColor =
Color.Red;
                    validatedData = false;
                }
            }
            if (validatedData == true)
            {
                dGModel.Rows[e.RowIndex].Cells[0].Value =
Math.Round(summation, 6);
            }
            else
            {
                dGModel.Rows[e.RowIndex].Cells[0].Value = "X";
            }
            //dGModel.ClearSelection();
        }

        private void dGModel_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            if (dGModel.SelectedCells.Count > 0)
                dGModel.ContextMenuStrip = contextMenuStrip1;
        }

        private void cutToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Copy to clipboard
            General.CopyToClipboard(dGModel);

            //Clear selected cells
            foreach (DataGridViewCell dgvCell in dGModel.SelectedCells)
                dgvCell.Value = string.Empty;
        }
        private void copyToolStripMenuItem_Click(object sender, EventArgs e)
        {
            General.CopyToClipboard(dGModel);
        }

        private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Perform paste Operation
            General.PasteClipboardValue(dGModel);
        }

        ///////////////////////////////



    }
}




using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IbrahimPHD
```

```
{
    public partial class FrmANOVA : Form
    {
        public FrmANOVA()
        {
            InitializeComponent();
        }

        private void FrmANOVA_Load(object sender, EventArgs e)
        {
            txtANOVA.SelectionStart = 0 ;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

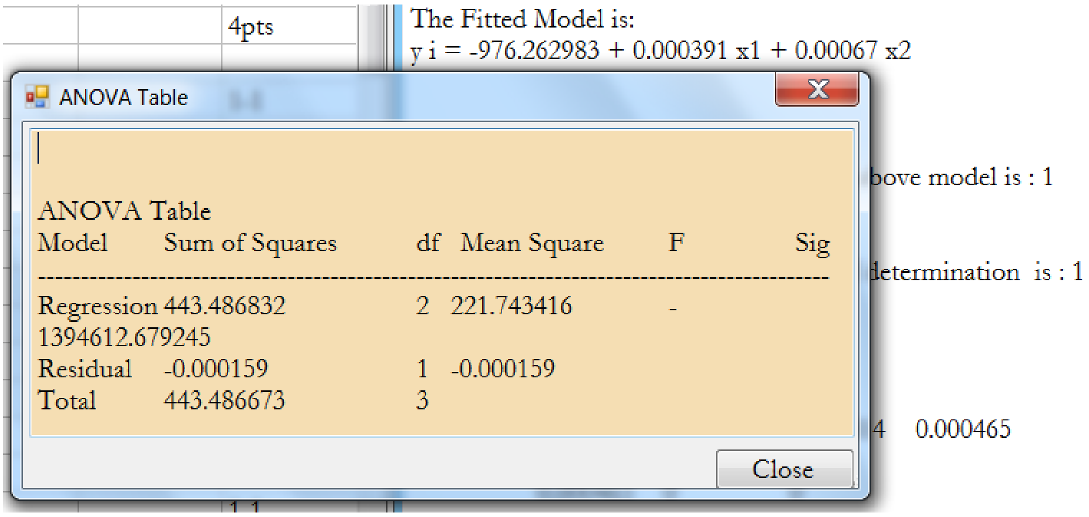
APPENDIX B

ANALYSES OF VARIANCE

Press ANOVA to obtain the analyses of variance for fitting regression see fig (AppB.1a to AppB.7d) and table 6.4.

The table summarizes the sum of squares, regression, residuals, the degree of freedom (d.f.) and the mean squares which are sums of squares divided by degrees of freedom.

The analysis of variance table is simply a convenient summary of the steps involved in calculating an F-statistic.



The Fitted Model is:
y i = -976.262983 + 0.000391 x1 + 0.00067 x2

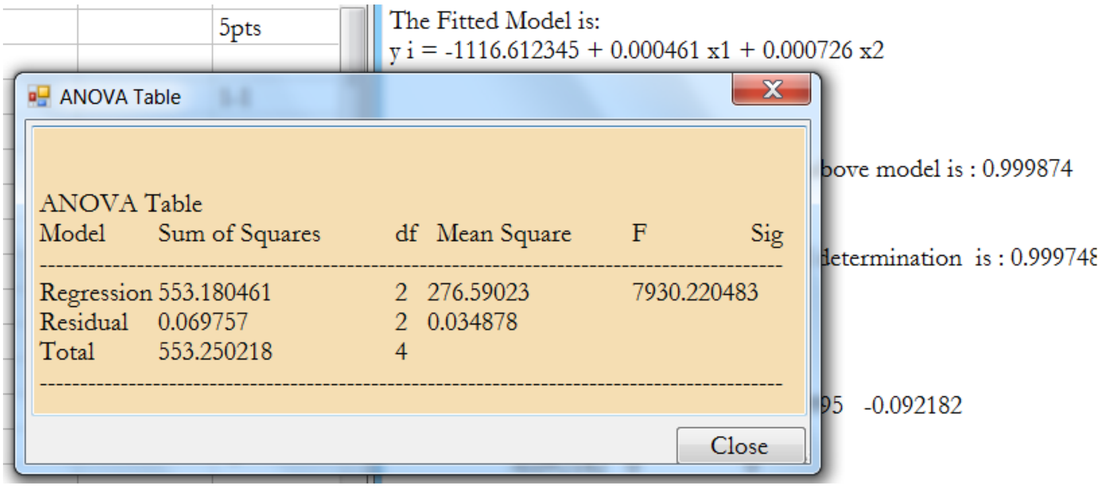
bove model is : 1

determination is : 1

4 0.000465

```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F            Sig
---------------------------------------------------------------------------------
Regression 443.486832                2    221.743416         -
1394612.679245
Residual     -0.000159               1    -0.000159
Total        443.486673              3
```

Fig AppB.1a-1X1 4Pts-ANOVA Table (see fig 6.2a)



The Fitted Model is:
y i = -1116.612345 + 0.000461 x1 + 0.000726 x2

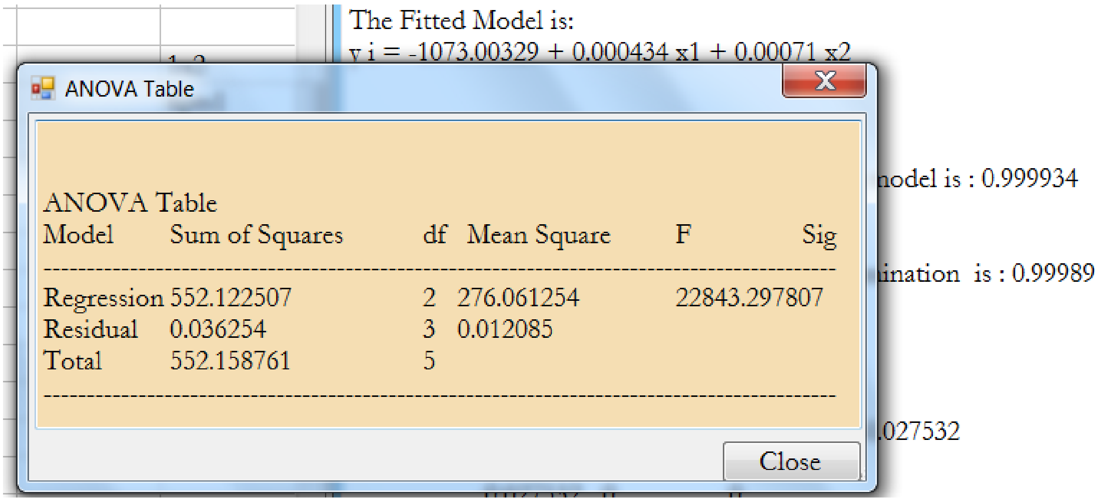
bove model is : 0.999874

determination is : 0.99974ℇ

95 -0.092182

```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F            Sig
---------------------------------------------------------------------------------
Regression 553.180461                2    276.59023          7930.220483
Residual     0.069757                2    0.034878
Total        553.250218              4
---------------------------------------------------------------------------------
```

Fig AppB.1b-1X1 5Pts ANOVA Table (see fig 6.2b)



The Fitted Model is:
y i = -1073.00329 + 0.000434 x1 + 0.00071 x2

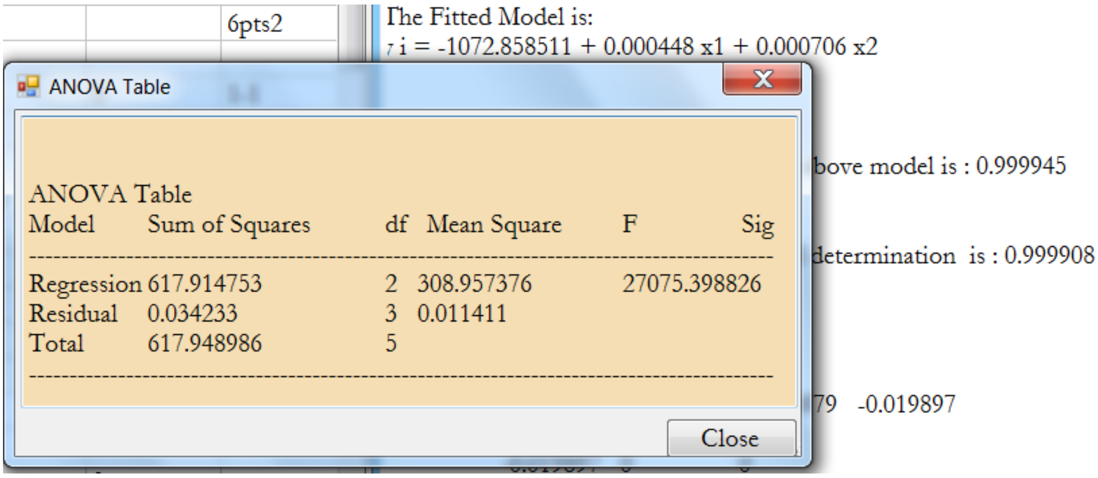
```
ANOVA Table
Model        Sum of Squares          df  Mean Square       F              Sig
--------------------------------------------------------------------------------
Regression 552.122507             2   276.061254         22843.297807
Residual     0.036254                3   0.012085
Total          552.158761             5
--------------------------------------------------------------------------------
```

nodel is : 0.999934

ination is : 0.99989

.027532

Fig AppB.1c-1X1 6Pts1 ANOVA Table (see fig 6.2c)



The Fitted Model is:
i = -1072.858511 + 0.000448 x1 + 0.000706 x2

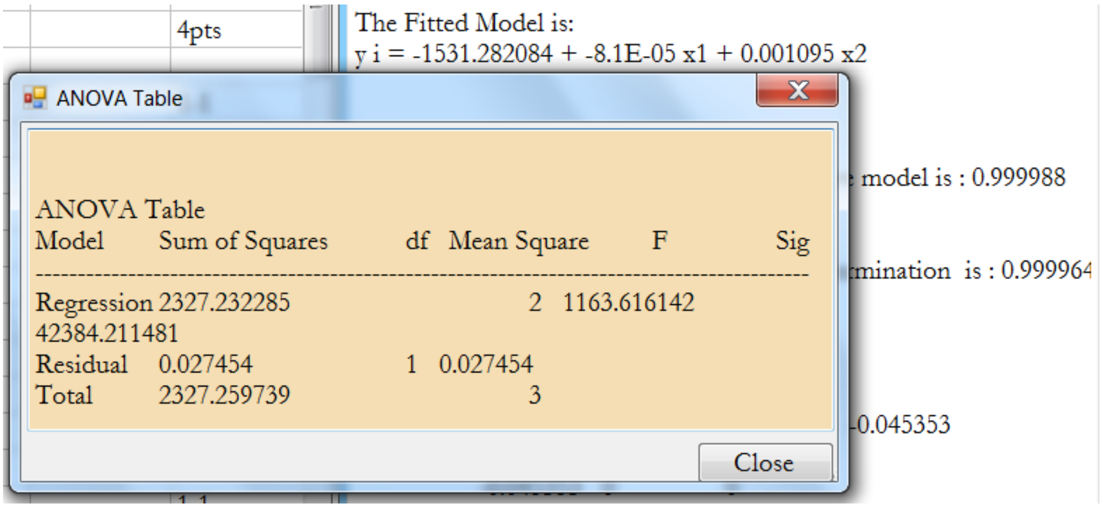
```
ANOVA Table
Model        Sum of Squares          df  Mean Square       F              Sig
--------------------------------------------------------------------------------
Regression 617.914753             2   308.957376         27075.398826
Residual     0.034233                3   0.011411
Total          617.948986             5
--------------------------------------------------------------------------------
```

bove model is : 0.999945

determination is : 0.999908

79 -0.019897

Fig AppB.1d-1X1 6Pts2 ANOVA Table (see fig 6.2d)



The Fitted Model is:
y i = -1531.282084 + -8.1E-05 x1 + 0.001095 x2

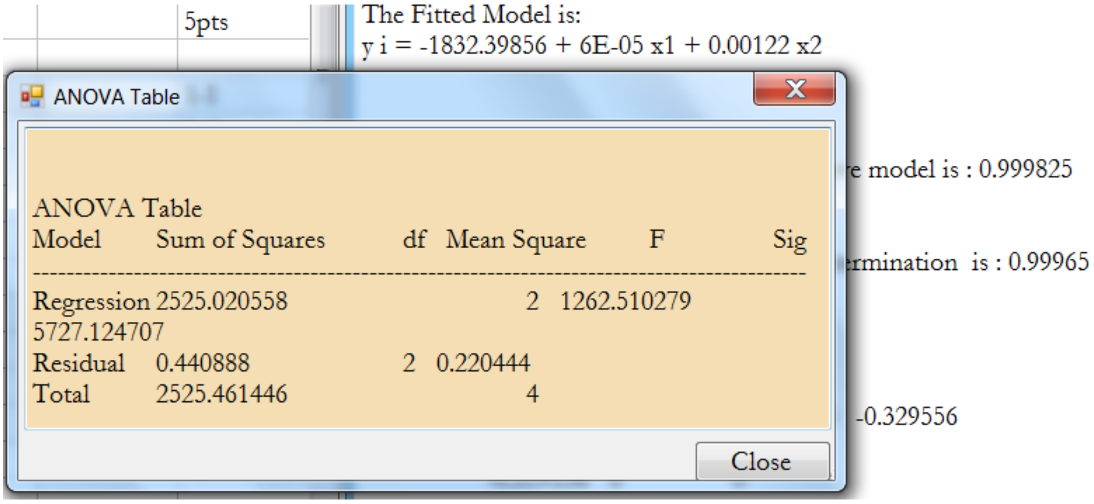
```
ANOVA Table
Model        Sum of Squares          df  Mean Square       F              Sig
--------------------------------------------------------------------------------
Regression 2327.232285                       2   1163.616142
42384.211481
Residual     0.027454                1   0.027454
Total          2327.259739                    3
```

model is : 0.999988

mination is : 0.999964

-0.045353

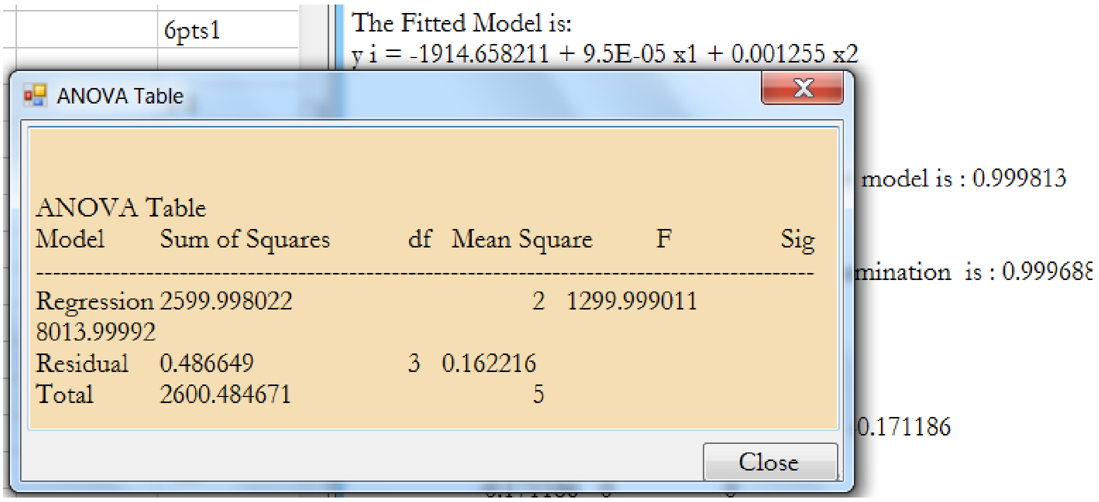
Fig AppB.2a-1X2 4ptS ANOVA Table



```
The Fitted Model is:
y i = -1832.39856 + 6E-05 x1 + 0.00122 x2

e model is : 0.999825
rmination  is : 0.99965
-0.329556
```

```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F             Sig
---------------------------------------------------------------------------
Regression 2525.020558                        2   1262.510279
5727.124707
Residual    0.440888               2   0.220444
Total       2525.461446                       4
```

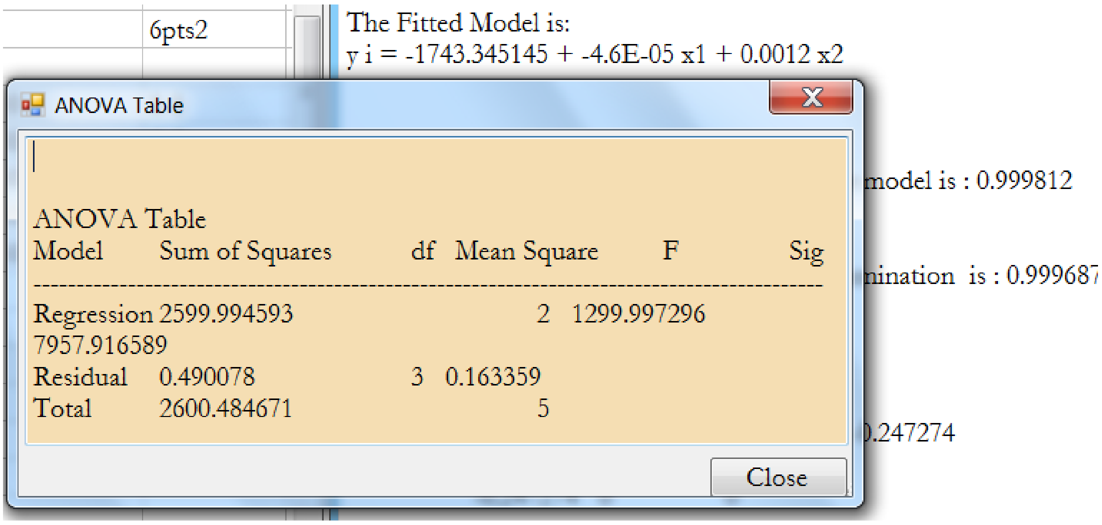
Fig AppB.2.b-1X2 5Pts ANOVA Table



```
The Fitted Model is:
y i = -1914.658211 + 9.5E-05 x1 + 0.001255 x2

model is : 0.999813
mination  is : 0.999688
0.171186
```

```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F             Sig
---------------------------------------------------------------------------
Regression 2599.998022                        2   1299.999011
8013.99992
Residual    0.486649               3   0.162216
Total       2600.484671                       5
```

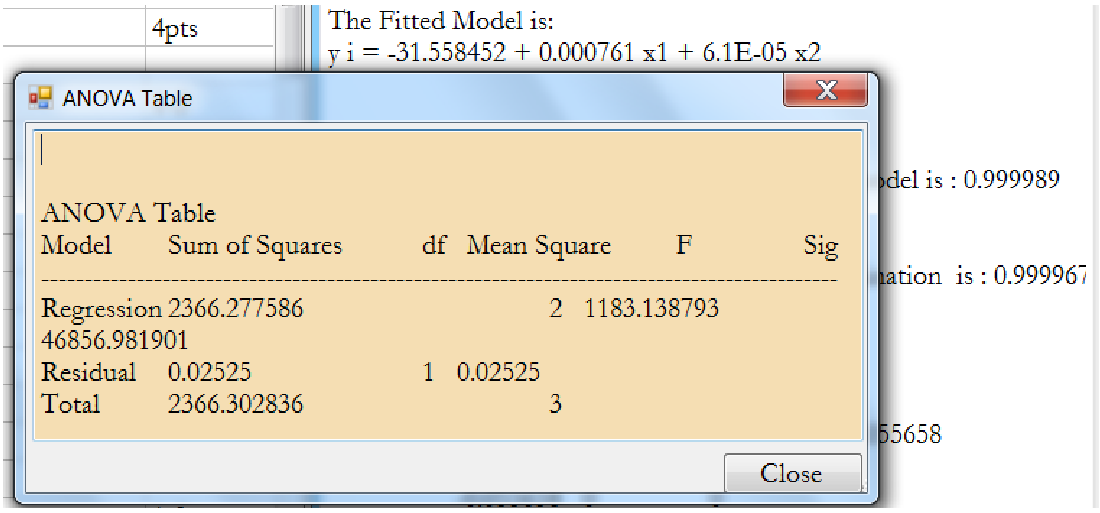
Fig AppB.2c-1X2 6Pts1 ANOVA Table



```
The Fitted Model is:
y i = -1743.345145 + -4.6E-05 x1 + 0.0012 x2

model is : 0.999812
nination  is : 0.999687
).247274
```

```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F             Sig
---------------------------------------------------------------------------
Regression 2599.994593                        2   1299.997296
7957.916589
Residual    0.490078               3   0.163359
Total       2600.484671                       5
```

Fig AppB.2d-1X2 6Pts2 ANOVA Table



The Fitted Model is:
y i = -31.558452 + 0.000761 x1 + 6.1E-05 x2

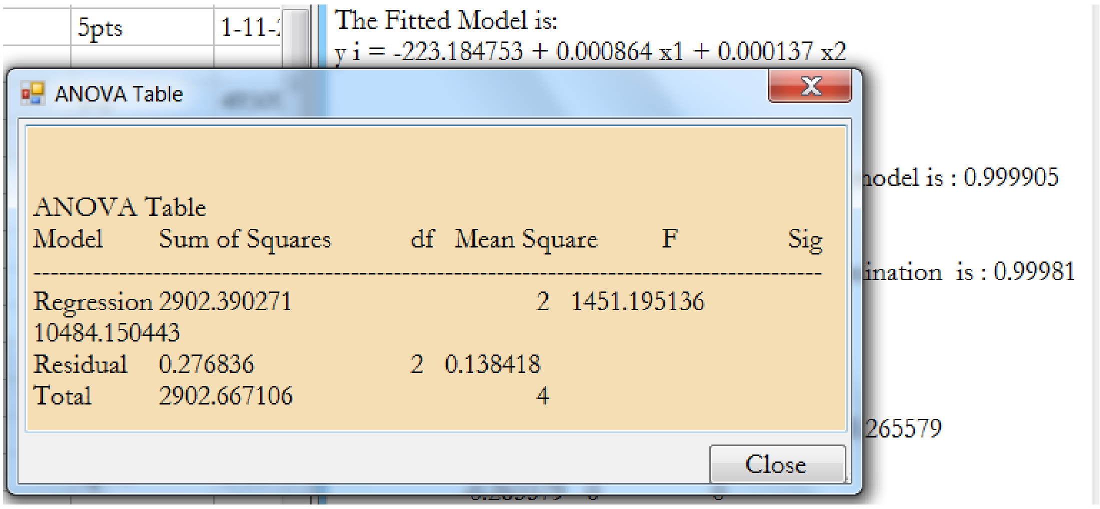
```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F            Sig
--------------------------------------------------------------------------------
Regression 2366.277586                      2   1183.138793
46856.981901
Residual    0.02525              1    0.02525
Total       2366.302836                     3
```

odel is : 0.999989

ation is : 0.999967

55658

Fig AppB.3a-2X1 4Pts ANOVA Table



The Fitted Model is:
y i = -223.184753 + 0.000864 x1 + 0.000137 x2

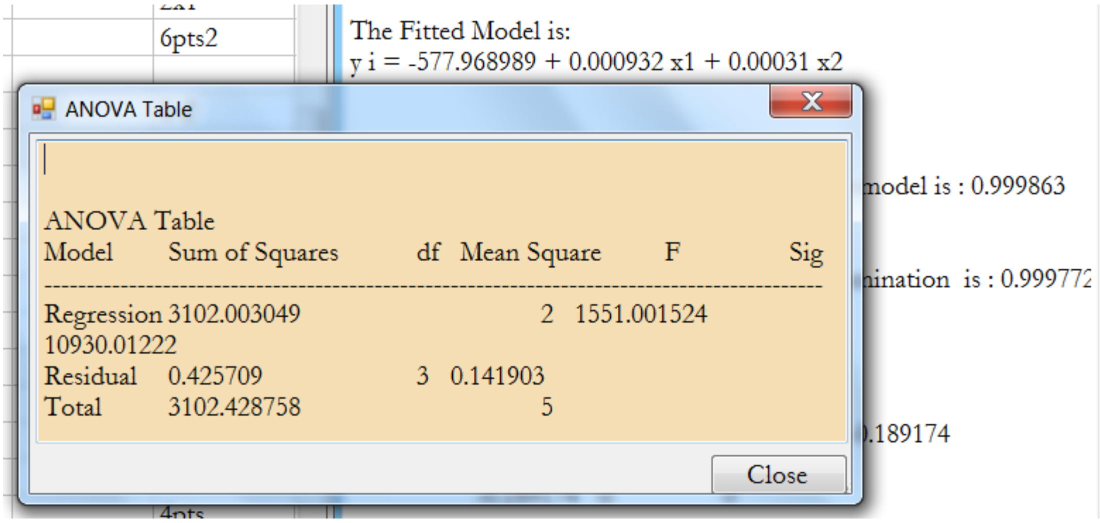
```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F            Sig
--------------------------------------------------------------------------------
Regression 2902.390271                      2   1451.195136
10484.150443
Residual    0.276836             2    0.138418
Total       2902.667106                     4
```

nodel is : 0.999905

ination is : 0.99981

265579

Fig AppB.3b-2X1 5Pts ANOVA Table



The Fitted Model is:
y i = -577.968989 + 0.000932 x1 + 0.00031 x2

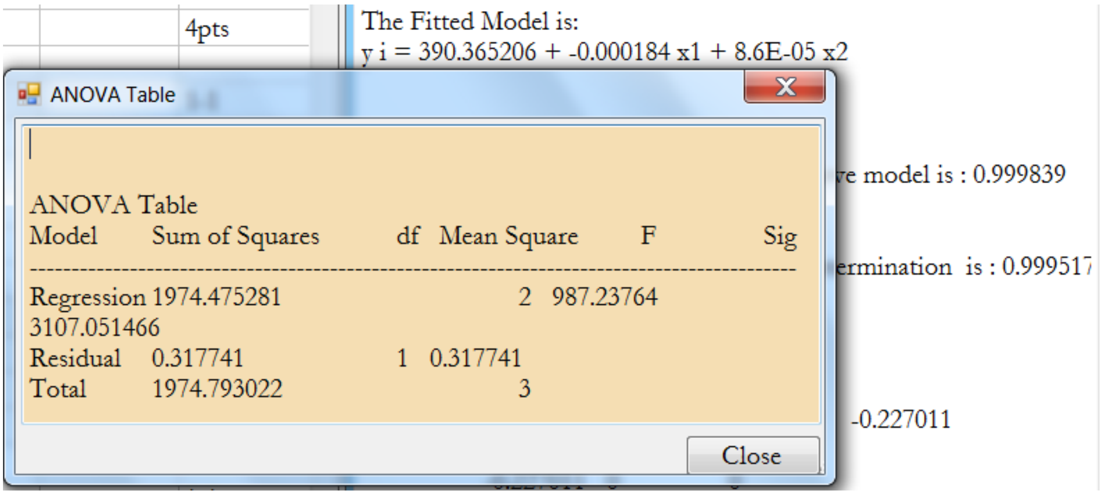
```
ANOVA Table
Model       Sum of Squares          df  Mean Square       F            Sig
--------------------------------------------------------------------------------
Regression 3102.003049                      2   1551.001524
10930.01222
Residual    0.425709             3    0.141903
Total       3102.428758                     5
```

model is : 0.999863

nination is : 0.99977

.189174

Fig AppB.3c-2X1 6Pts1 ANOVA Table

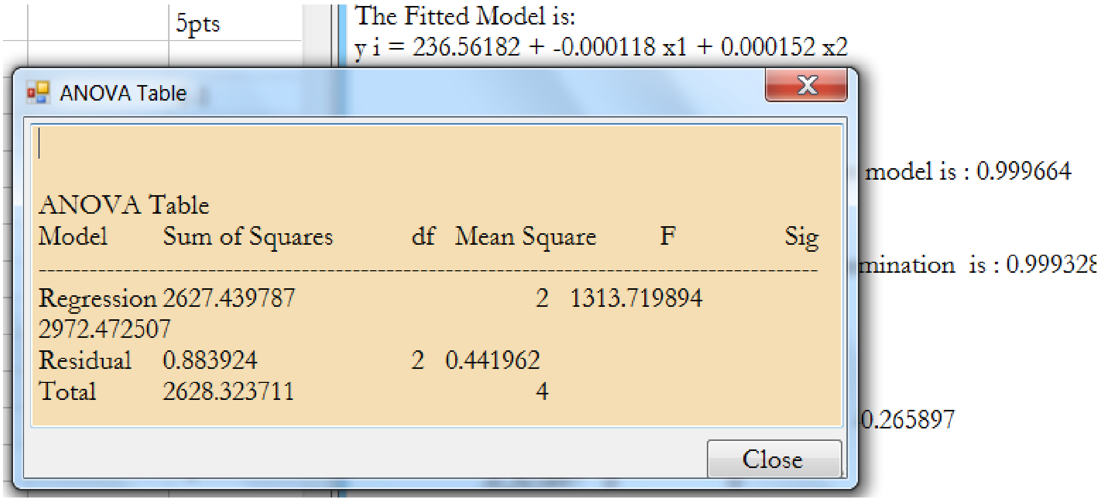


```
The Fitted Model is:
y i = 390.365206 + -0.000184 x1 + 8.6E-05 x2
```

```
ANOVA Table
Model       Sum of Squares          df   Mean Square       F            Sig
-----------------------------------------------------------------------------------
Regression 1974.475281                    2   987.23764
3107.051466
Residual    0.317741            1   0.317741
Total       1974.793022                   3
```

```
e model is : 0.999839
ermination  is : 0.999517
-0.227011
```

Fig AppB.4a-2X2 4Pts ANOVA Table

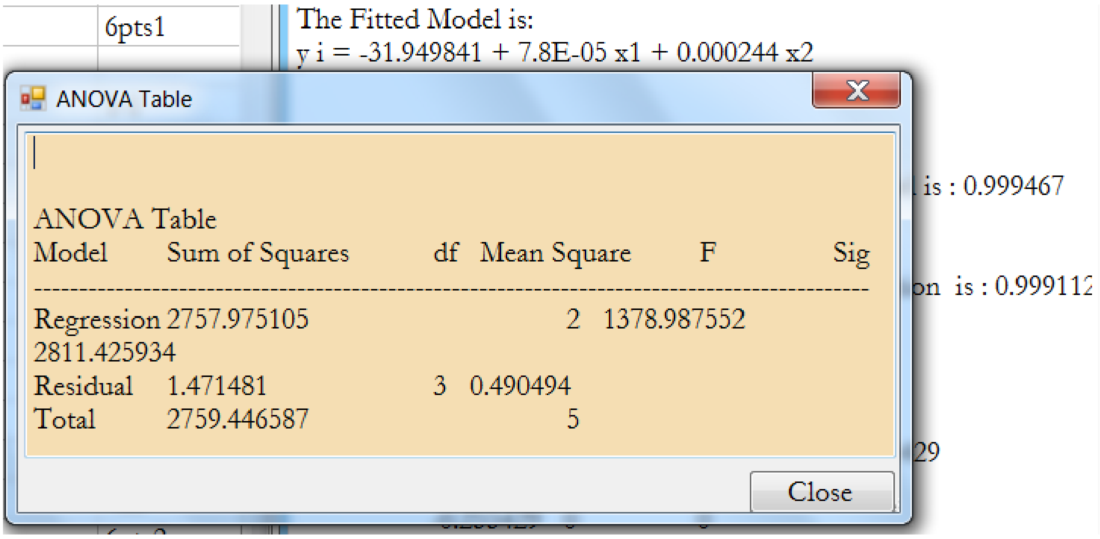


```
The Fitted Model is:
y i = 236.56182 + -0.000118 x1 + 0.000152 x2
```

```
ANOVA Table
Model       Sum of Squares          df   Mean Square       F            Sig
-----------------------------------------------------------------------------------
Regression 2627.439787                    2   1313.719894
2972.472507
Residual    0.883924            2   0.441962
Total       2628.323711                   4
```

```
model is : 0.999664
mination  is : 0.99932
0.265897
```

Fig AppB.4b-2X2 5Pts ANOVA Table

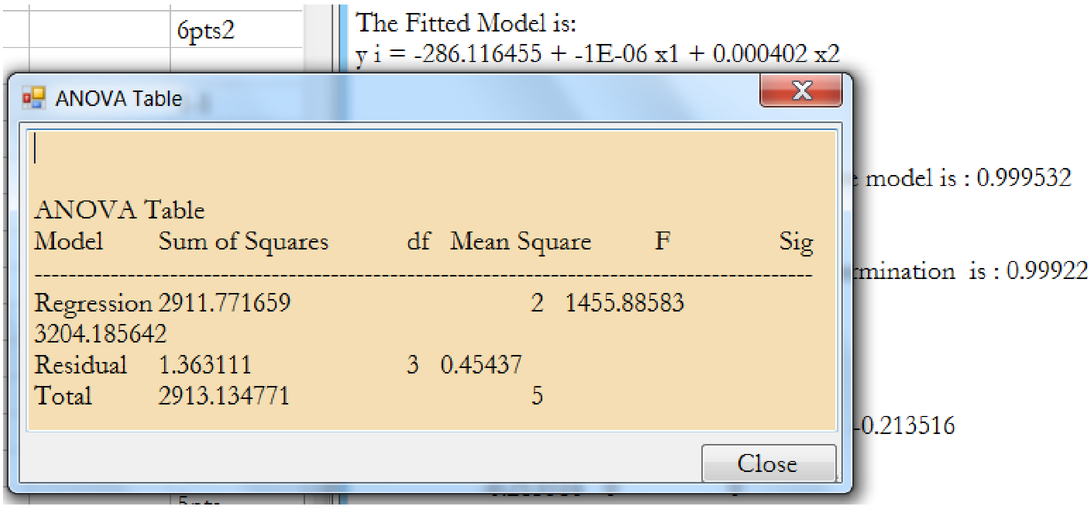


```
The Fitted Model is:
y i = -31.949841 + 7.8E-05 x1 + 0.000244 x2
```

```
ANOVA Table
Model       Sum of Squares          df   Mean Square       F            Sig
-----------------------------------------------------------------------------------
Regression 2757.975105                    2   1378.987552
2811.425934
Residual    1.471481            3   0.490494
Total       2759.446587                   5
```

```
is : 0.999467
on  is : 0.99911
29
```

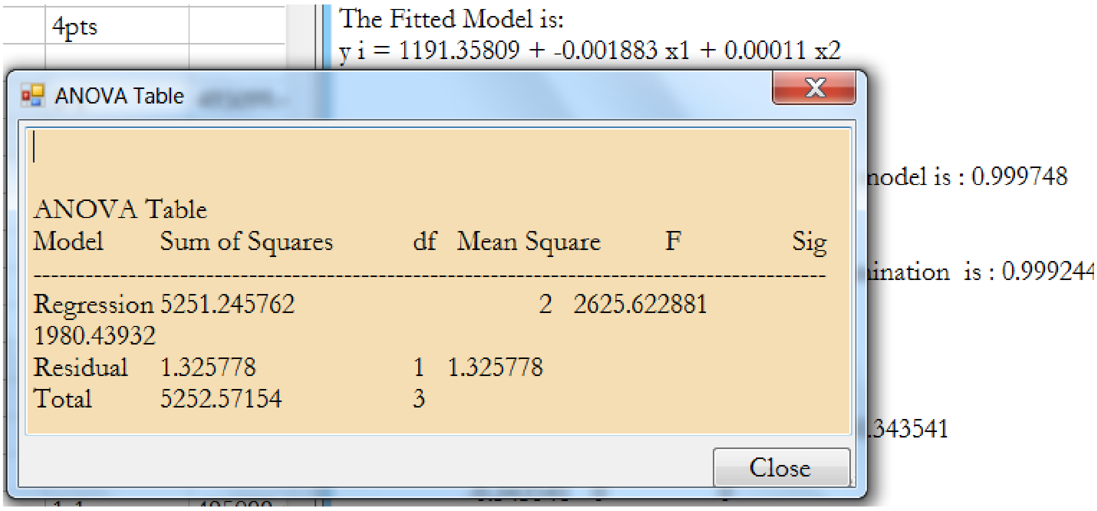
Fig AppB.4c-2X2 6Pts1 ANOVA Table



```
The Fitted Model is:
y i = -286.116455 + -1E-06 x1 + 0.000402 x2
```

```
ANOVA Table
Model       Sum of Squares         df  Mean Square        F              Sig
--------------------------------------------------------------------------------
Regression 2911.771659                          2   1455.88583
3204.185642
Residual    1.363111               3   0.45437
Total       2913.134771                         5
```

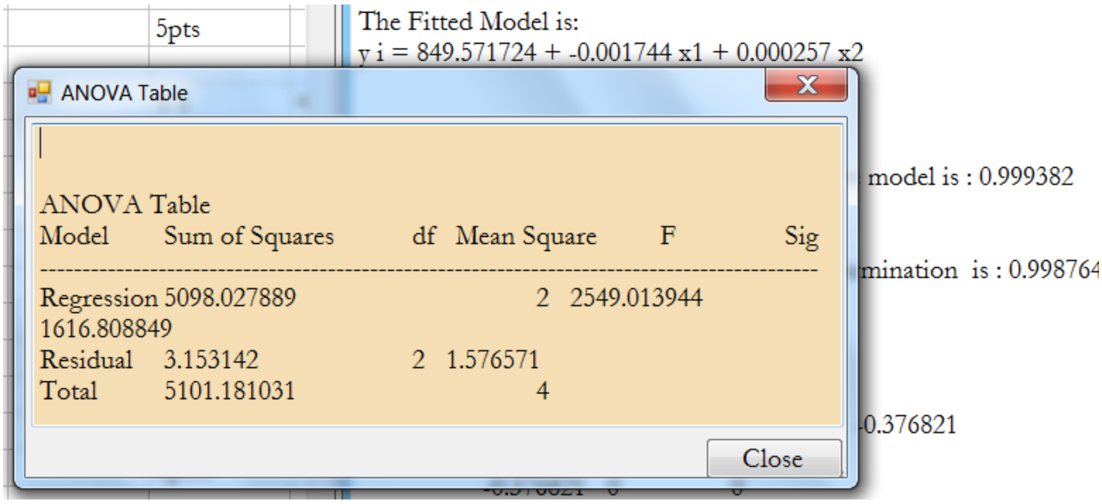
Fig AppB.4d-2X2 6Pts2 ANOVA Table



```
The Fitted Model is:
y i = 1191.35809 + -0.001883 x1 + 0.00011 x2
```

```
ANOVA Table
Model       Sum of Squares         df  Mean Square        F              Sig
--------------------------------------------------------------------------------
Regression 5251.245762                          2   2625.622881
1980.43932
Residual    1.325778               1   1.325778
Total       5252.57154             3
```

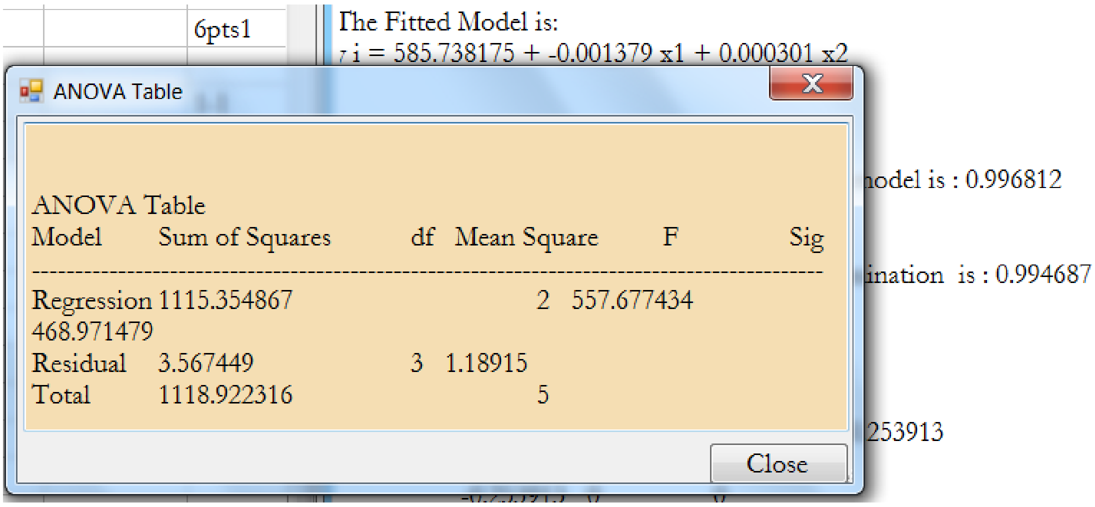
Fig AppB.5a-2X4 4Pts ANOVA Table



```
The Fitted Model is:
y i = 849.571724 + -0.001744 x1 + 0.000257 x2
```

```
ANOVA Table
Model       Sum of Squares         df  Mean Square        F              Sig
--------------------------------------------------------------------------------
Regression 5098.027889                          2   2549.013944
1616.808849
Residual    3.153142               2   1.576571
Total       5101.181031                         4
```

Fig AppB.5b-2X4 5Pts ANOVA Table

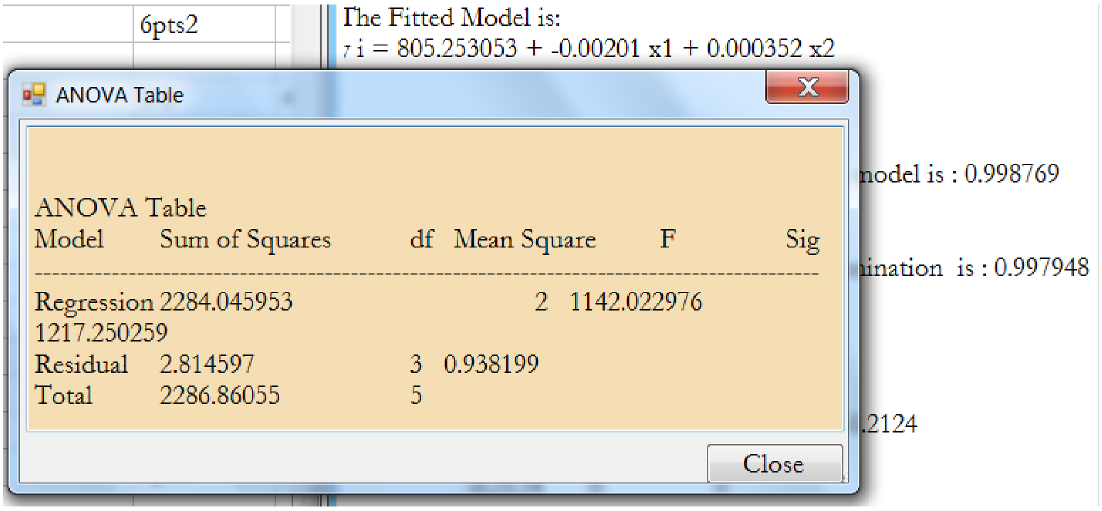


```
The Fitted Model is:
y i = 585.738175 + -0.001379 x1 + 0.000301 x2
```

```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F              Sig
-------------------------------------------------------------------------------
Regression 1115.354867                       2    557.677434
468.971479
Residual     3.567449                 3   1.18915
Total        1118.922316                     5
```

```
odel is : 0.996812
ination  is : 0.994687
253913
```

Fig AppB.5c-2X4 6Pts1 ANOVA Table

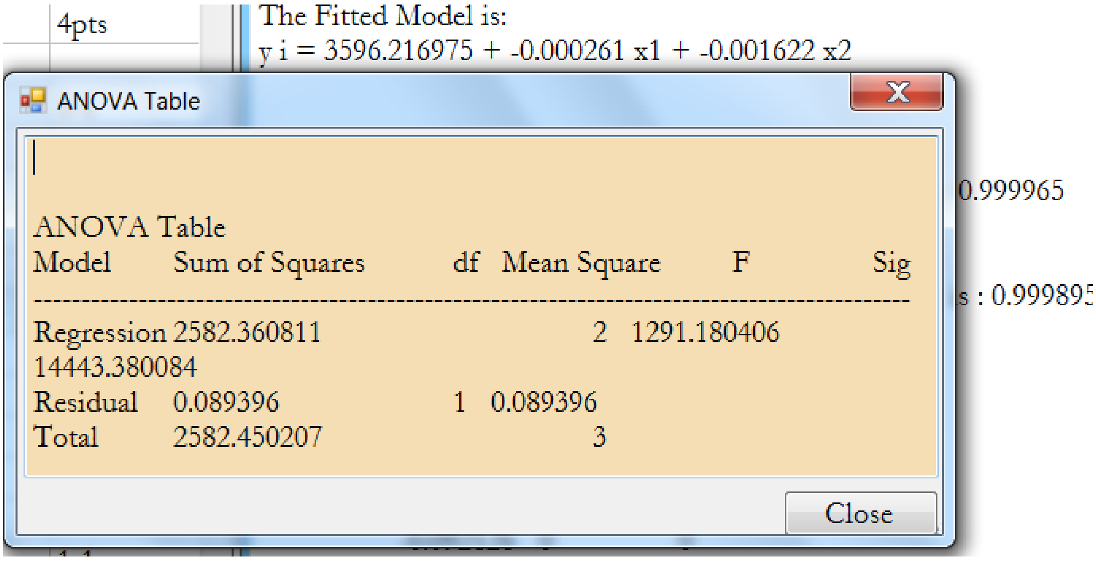


```
The Fitted Model is:
y i = 805.253053 + -0.00201 x1 + 0.000352 x2
```

```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F              Sig
-------------------------------------------------------------------------------
Regression 2284.045953                       2    1142.022976
1217.250259
Residual     2.814597                 3   0.938199
Total        2286.86055               5
```

```
odel is : 0.998769
ination  is : 0.997948
.2124
```

Fig AppB.5d-2X4 6Pts2 ANOVA Table

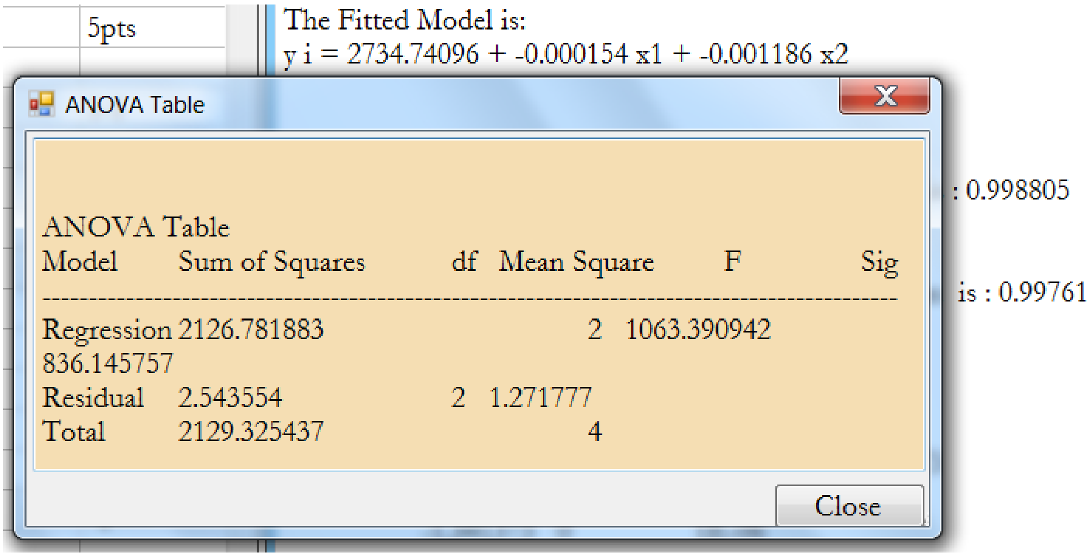


```
The Fitted Model is:
y i = 3596.216975 + -0.000261 x1 + -0.001622 x2
```

```
ANOVA Table
Model        Sum of Squares          df   Mean Square        F              Sig
-------------------------------------------------------------------------------
Regression 2582.360811                       2    1291.180406
14443.380084
Residual     0.089396                 1    0.089396
Total        2582.450207                     3
```

```
0.999965
s : 0.999895
```

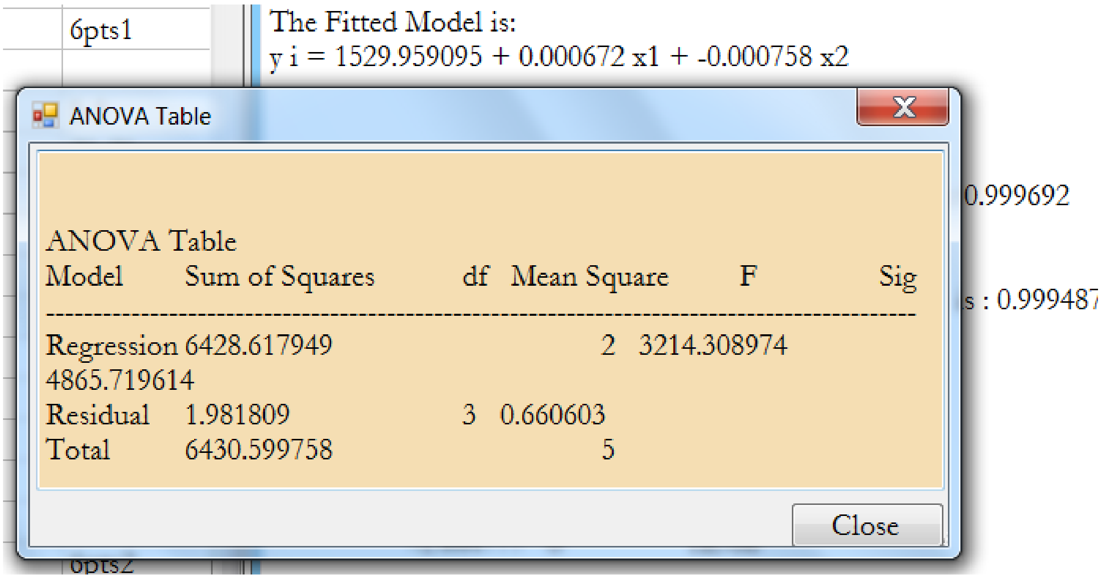
Fig AppB.6a-4X2 4Pts ANOVA Table



```
The Fitted Model is:
y i = 2734.74096 + -0.000154 x1 + -0.001186 x2
```

```
ANOVA Table
Model       Sum of Squares         df  Mean Square       F            Sig
------------------------------------------------------------------------------------
Regression 2126.781883                    2   1063.390942
836.145757
Residual    2.543554             2   1.271777
Total       2129.325437                   4
```

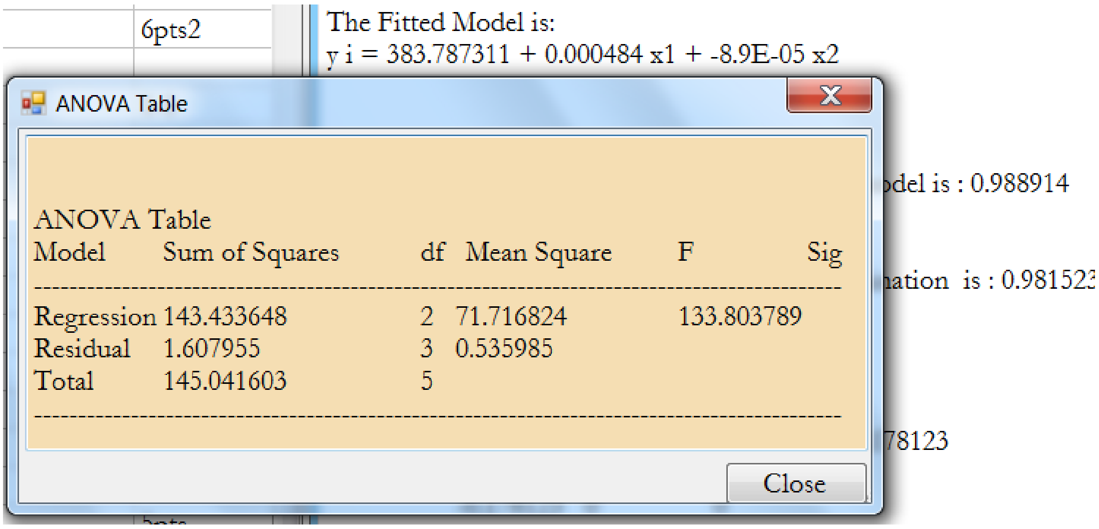
Fig AppB.6b-4X2 5Pts ANOVA Table



```
The Fitted Model is:
y i = 1529.959095 + 0.000672 x1 + -0.000758 x2
```

```
ANOVA Table
Model       Sum of Squares         df  Mean Square       F            Sig
------------------------------------------------------------------------------------
Regression 6428.617949                    2   3214.308974
4865.719614
Residual    1.981809             3   0.660603
Total       6430.599758                   5
```

Fig AppB.6c-4X2 6Pts1 ANOVA Table



6pts2

The Fitted Model is:
y i = 383.787311 + 0.000484 x1 + -8.9E-05 x2

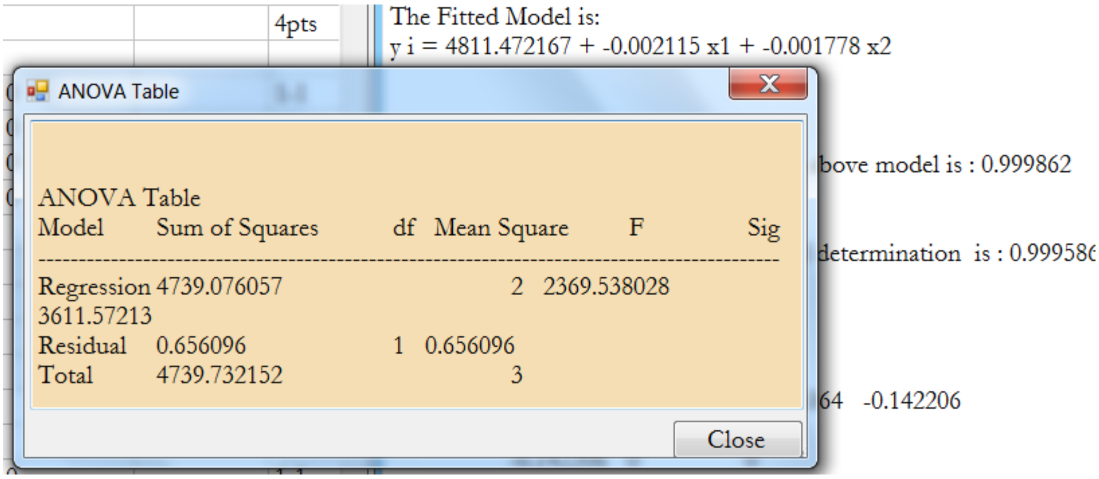
odel is : 0.988914

ation is : 0.981523

78123

```
ANOVA Table
Model        Sum of Squares        df  Mean Square        F              Sig
------------------------------------------------------------------------------
Regression 143.433648              2   71.716824          133.803789
Residual   1.607955                3   0.535985
Total      145.041603              5
------------------------------------------------------------------------------
```

Fig AppB.6d-4X2 6Pts2 ANOVA Table



4pts

The Fitted Model is:
y i = 4811.472167 + -0.002115 x1 + -0.001778 x2

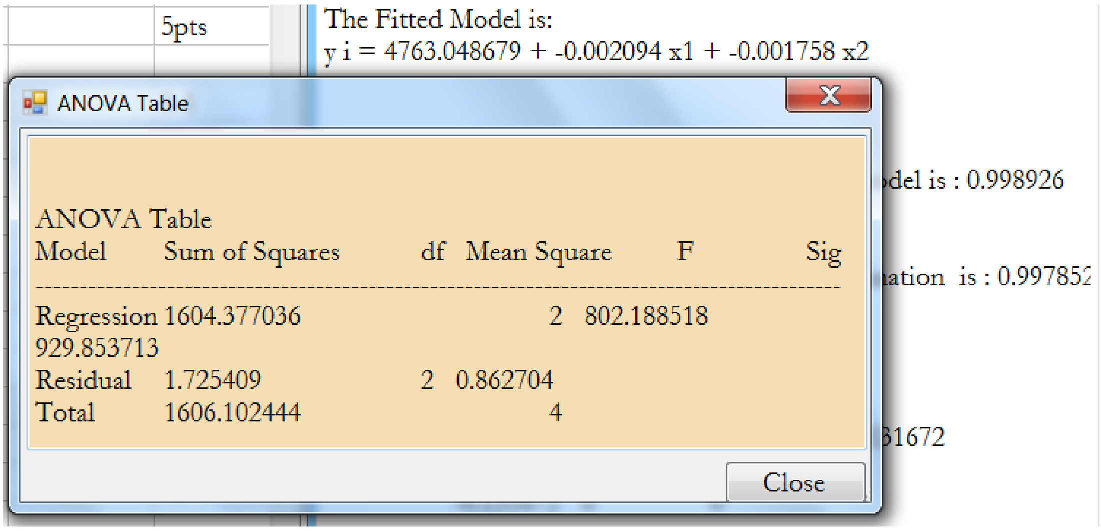
bove model is : 0.999862

determination is : 0.999586

64 -0.142206

```
ANOVA Table
Model        Sum of Squares        df  Mean Square        F              Sig
------------------------------------------------------------------------------
Regression 4739.076057                         2   2369.538028
3611.57213
Residual   0.656096                1   0.656096
Total      4739.732152                         3
```

Fig AppB.7a-4X4 4Pts ANOVA Table



5pts

The Fitted Model is:
y i = 4763.048679 + -0.002094 x1 + -0.001758 x2

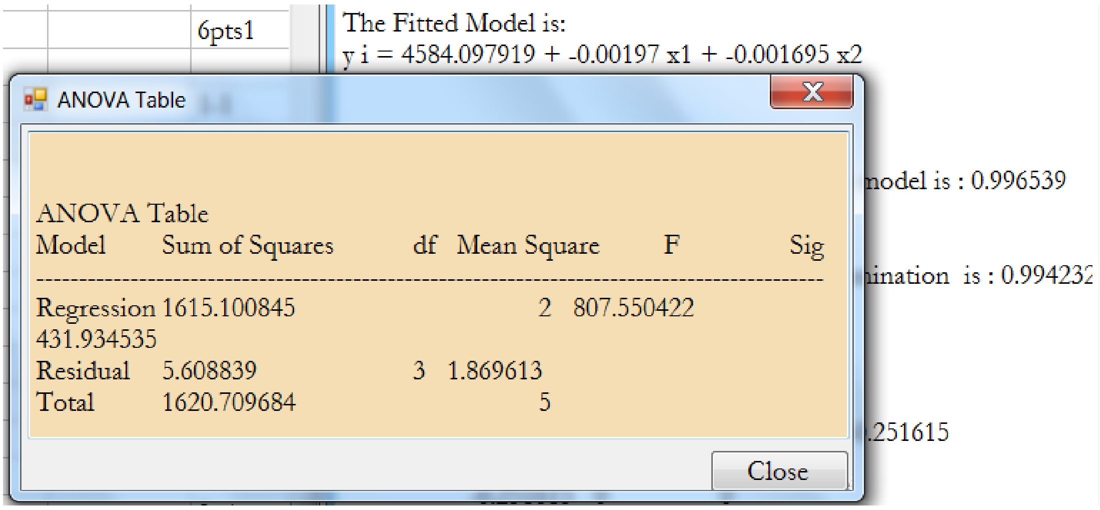
odel is : 0.998926

ation is : 0.997852

31672

```
ANOVA Table
Model        Sum of Squares        df  Mean Square        F              Sig
------------------------------------------------------------------------------
Regression 1604.377036                         2   802.188518
929.853713
Residual   1.725409                2   0.862704
Total      1606.102444                         4
```

Fig AppB.7b-4X4 5Pts ANOVA Table



The Fitted Model is:
y i = 4584.097919 + -0.00197 x1 + -0.001695 x2

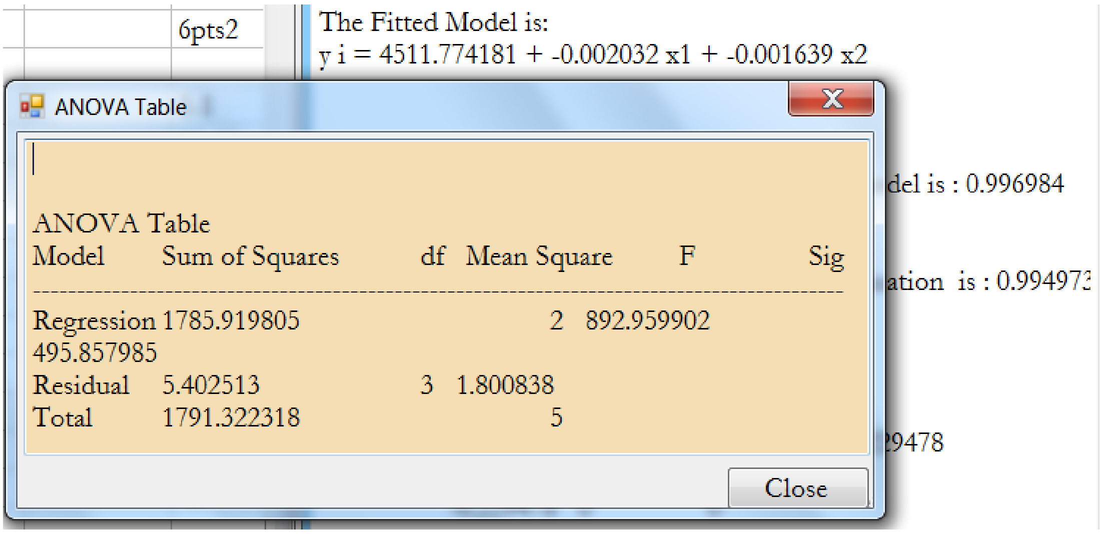
nodel is : 0.996539

nination is : 0.994232

.251615

```
ANOVA Table
Model       Sum of Squares         df  Mean Square       F            Sig
--------------------------------------------------------------------------------
Regression 1615.100845                   2   807.550422
431.934535
Residual    5.608839              3   1.869613
Total       1620.709684                  5
```

Fig AppB.7c-4X4 6Pts1 ANOVA Table



The Fitted Model is:
y i = 4511.774181 + -0.002032 x1 + -0.001639 x2

del is : 0.996984

ation is : 0.994973

29478

```
ANOVA Table
Model       Sum of Squares         df  Mean Square       F            Sig
--------------------------------------------------------------------------------
Regression 1785.919805                   2   892.959902
495.857985
Residual    5.402513              3   1.800838
Total       1791.322318                  5
```

Fig AppB.7d-4X4 6Pts2 ANOVA Table