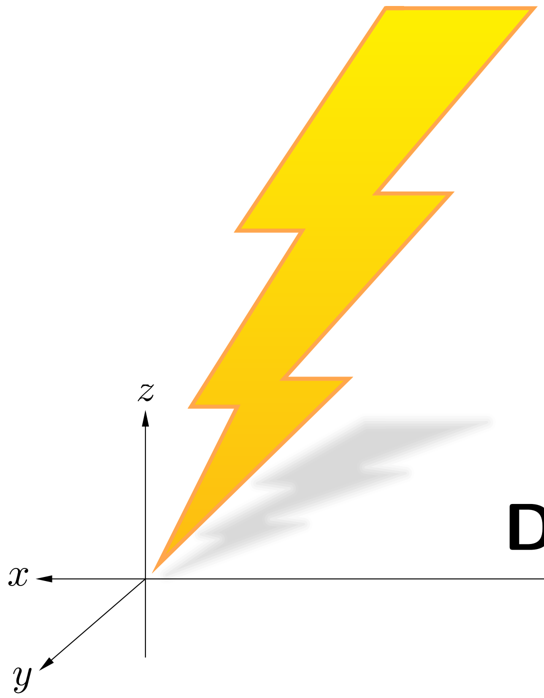


DynAdjust User's Guide

Version 1.1

Editors:
Roger Fraser, Frank Leahy, Philip Collier

May 2021

DynAdjust User's Guide
Dynamic network adjustment software
25 May 2021

For all technical and software related matters, please contact:

Dr. Roger Fraser
Chief Geospatial Scientist
Land Use Victoria
Department of Environment, Land, Water and Planning
Level 11, 2 Lonsdale St, Melbourne, Victoria, 3000
roger.fraser@delwp.vic.gov.au

DynAdjust User's Guide
Version 1.1

Dr. Roger Fraser
Chief Geospatial Scientist
Land Use Victoria
Melbourne

Dr. Frank Leahy
Associate Professor
University of Melbourne
Melbourne

Dr. Phil Collier
Principal Consultant
Collier Geodetic Solutions
Melbourne

Acknowledgements

The development of this software and user guide has benefited from the assistance provided by several individuals and organisations. In particular, the authors gratefully acknowledge the following people for their advice, support, feedback, supply of sample data files and contribution at various levels: Gary Johnston (retired), John Dawson, Nick Brown, Craig Harrison, Ted Zhou, Amy Peterson and Jonathan Mettes from Geoscience Australia (Commonwealth); Ben Menadue and Dale Roberts from the National Computational Infrastructure (Commonwealth); John Tulloch (retired), David Boyle, Alex Woods, Bob Ross (retired), Peter Growse (retired), James Leversha and Jason Heritage from the Department of Environment, Land, Water and Planning (Victoria); Matt Higgins, Steve Tarbit, Mike Cowie, Darren Burns and Peter Todd (retired) from the Department of Natural Resources, Mines and Energy (Queensland); Simon McElroy, Joel Haasdyk and Nic Gowans from the Department of Customer Service (New South Wales); Linda Morgan (retired), Irek Baran and Kent Wheeler from Landgate (Western Australia); Graeme Blick, Nic Donnelly, Chris Crook and Dave Collett from Land Information New Zealand (New Zealand); Scott Strong from the Department of Primary Industries, Parks, Water and Environment (Tasmania); Stephen Latham and Peter Stolz from the Department of Planning, Transport and Infrastructure (South Australia); Gavin Evans from the Department of Environment and Sustainable Development (ACT); Rob Sarib from the Department of Infrastructure, Planning and Logistics (Northern Territory); Simon Fuller from ThinkSpatial (Victoria); Neil Brown from LISTECH; Amir Bar-Maor, Tim Hodson and Terry Brinkman from Esri (Redlands); Peter Teunissen from Curtin University; Chris Rizos (retired) from the University of New South Wales; and Rod Deakin (retired) and Don Grant (retired) from the Royal Melbourne Institute of Technology.

Freely ye have received, freely give
(Matthew 10:8)

Remember the words of the Lord Jesus, how He said, “It is more blessed to give than to receive”
(Acts 20:35)

Contents

List of Figures

List of Tables

List of Abbreviations

AGD66	Australian Geodetic Datum 1966
AGD84	Australian Geodetic Datum 1984
AHD	Australian Height Datum. Commonly used in reference to Australian Height Datum 1971 (AHD71) and Australian Height Datum (Tasmania) 1983 (AHD–TAS83).
AHD–TAS83	The Australian Height Datum (Tasmania) 1983 is the normal–orthometric height datum for mainland Tasmania.
AHD71	The Australian Height Datum 1971 is the normal–orthometric height datum for mainland Australia.
CSV	Comma Separated Values
DNA	Dynamic Network Adjustment
DynaML	DynaNet Markup Language. XML format defined according to the DynaNet 2.0 XML schema.
EPSG	European Petroleum Survey Group
GDA2020	Geocentric Datum of Australia 2020. Realised by continuous analysis of over 400 Asia Pacific Reference Frame sites, referenced to the GRS80 ellipsoid and determined with respect to ITRF2014 at epoch 2020.0.
GDA94	Geocentric Datum of Australia 1994. Realised by the derived coordinates of the Australian Fiducial Network (AFN) geodetic stations, referenced to the GRS80 ellipsoid and determined with respect to ITRF92 at epoch 1994.0.
GLONASS	GLObal NAvigation Satellite System
GMA	Geodetic Model of Australia. Implemented via the adjustments termed GMA73, GMA80, GMA82 and GMA84.
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRS80	Geodetic Reference System 1980 reference ellipsoid, where a = 6378137.0 m, f = 1/298.257222101
ICSM	Intergovernmental Committee on Surveying and Mapping
IGS	International GNSS Service
ITRF	International Terrestrial Reference Frame — a realisation of the International Terrestrial Reference System (ITRS) produced by the International Earth Rotation and Reference Systems Service (IERS).
MGA94	Map Grid of Australia 1994. Universal Transverse Mercator projection of the Geocentric Datum of Australia 1994.
MSL	Mean sea level
PDF	Probability Density Function

RHEL	RedHat Enterprise Linux
SINEX	Solution INdependent EXchange format.
TM	Transverse Mercator
UTM	Universal Transverse Mercator
WGS84	World Geodetic System of 1994
XML	eXtensible Markup Language

List of Symbols

ϕ	Geodetic latitude.
λ	Geodetic longitude.
Φ	Astronomic latitude.
Λ	Astronomic longitude.
A	Astronomic azimuth.
θ_{12}	Geodetic azimuth between points 1 and 2.
α_{123}	Horizontal angle at point 1, between points 2 and 3.
s_{12}	Slope or direct distance between points 1 and 2.
c_{12}	Ellipsoid chord distance between points 1 and 2.
e_{12}	Ellipsoid arc distance between points 1 and 2.
m_{12}	Mean sea level arc distance between points 1 and 2.
ζ_{12}	Zenith distance from point 1 to 2.
ϑ_{12}	Vertical angle from point 1 to 2.
ν	Radius of curvature in the prime vertical.
ρ	Radius of curvature in the prime meridian.
a	Semi–major axis of the reference ellipsoid.
b	Semi–minor axis of the reference ellipsoid.
e^2	First eccentricity of the reference ellipsoid.
x, y, z	X, Y and Z coordinates in the cartesian reference frame.
ϕ, λ, h	Geodetic latitude, geodetic longitude and ellipsoid height in the geographic reference frame.
e, n, up	East, north and up coordinates in the local reference frame.
θ, ϑ, s	Geodetic azimuth, vertical angle and direct distance in the polar reference frame.
E, N, Zo	Easting, Northing and zone of a Universal Transverse Mercator (UTM) projection.
k_0	UTM central scale factor.
λ_0	Longitude of the central meridian of a UTM zone.
z_w	UTM zone width.
ξ	Deflection of the vertical in the prime meridian (north–south component).
η	Deflection of the vertical in the prime vertical (east–west component).
ϵ	Combined deflection of the vertical calculated from north–south and east–west components.
N or ζ	Geoid–ellipsoid separation or height anomaly.

H	Height of point p above an orthometric height surface.
h	Height of point p above the ellipsoid.
μ	The mean.
σ	Standard deviation of an estimated quantity.
σ^2	Variance or precision of an estimated quantity.
σ_0^2	Variance factor or sigma zero.
α	Probability or significance level, expressed as a percentage or value from 0 to 1.0.
k	Coverage factor corresponding to the level of significance.
χ_r^2	Chi–square statistic with r degrees of freedom.
w	Sum of the squares of the weighted corrections minimised in the solution of the least squares normal equations.
τ	Pelzer's measurement reliability criterion.
T	Pelzer's global network reliability criterion.

Chapter 1

Introduction

DynAdjust is a rigorous, high performance least squares adjustment application. It has been designed to estimate 3D station coordinates and uncertainties for both small and extremely large geodetic networks, and can be used for the adjustment of a large array of Global Navigation Satellite System (GNSS) and conventional terrestrial survey measurement types. On account of the phased adjustment approach used by DynAdjust, the maximum network size which can be adjusted is effectively unlimited, other than by the limitations imposed by a computer's processor, physical memory and operating system memory model. Example projects where DynAdjust can and has been used include the adjustment of small survey control networks, engineering surveys, deformation monitoring surveys, national and state geodetic networks and digital cadastral database upgrade initiatives.

DynAdjust provides the following capabilities:

- Import of data in geographic, cartesian and/or projection (UTM) coordinates contained in DNA, DynaML and SINEX data formats;
- Input of a diverse range of measurement types;
- Transformation of station coordinates and measurements between several static and dynamic reference frames;
- Rigorous application of geoid–ellipsoid separations and deflections of the vertical;
- Simultaneous (traditional) and phased adjustment modes;
- Automatic segmentation and adjustment of extremely large networks in an efficient manner;
- Rigorous estimation of positional uncertainty for all points in a network;
- Detailed statistical analysis of adjusted measurements and station corrections;
- Production of high quality network plots;
- Automated processing and analysis with minimal user interaction.

1.1 Brief history of phased adjustment and DynAdjust

Since Gauss published his treatment of the method of least squares in 1809, Tienstra (1956) was the first to develop the concept and principles of phased adjustment. In his work, Tienstra defines the principle property of phased adjustment as — "*Least squares problems may be divided into an arbitrary number of phases, provided that in each following phase(s), cofactors resulting from preceding phases are used.*" Using the method of condition equations and Ricci calculus, Tienstra demonstrated that rigorous parameter estimates for all stations could be derived in a step–wise fashion by treating the parameters estimated in one phase as quasi–measurements in the next phase.

Since the time of his publication, Tienstra's concept of phased adjustment has been studied by several authors, and has been extended and implemented in various forms. Initially, Professor Dr. Ir. Willem Baarda of the Computing Centre of the Delft Technological University (where Tienstra was Professor until his death in 1951) led the implementation of Tienstra's phased adjustment algorithm for the 1959 adjustment of the United European levelling network Alberda (1963). A few years later, Lambeck (1963) studied phased adjustment and demonstrated that Krüger's (1905) method of stacking normal equations was mathematically equivalent to Tienstra's phased adjustment technique. Schmid and Schmid (1965), in their discussion of the *generalised approach* to least squares, include an algorithm which solves least squares problems "*sequentially*" in steps. Their algorithm is essentially the same as Tienstra's phased adjustment algorithm, although it is designed to provide for the addition and removal of measurements after all parameters in the network have been estimated, rather than to provide a mechanism for handling the adjustment of a large network in stages.

Kouba (1970; 1972) studied the phased adjustment method and demonstrated that the sequential least squares technique developed by Schmid and Schmid (1965) was mathematically equivalent to that of Tienstra's. In his work, Kouba made an important contribution to the study of phased adjustment by demonstrating that all junction parameter estimates and variances must be carried between phases if rigorous results are to be achieved. Ying Chung–Chi (1970) summarised Tienstra's work using matrix algebra and extended the concept to adjustment by observation equations. Ying Chung–Chi (1970) also demonstrated mathematical equivalence between Tienstra's method and Krüger's method. Although Ying Chung–Chi does not formally prove that Tienstra's phased adjustment method will give the same estimates as a simultaneous adjustment, he demonstrates that it does so numerically for a simple level network.

The next major application of Tienstra's concept of phased adjustment was the Canadian Section Method, developed by Pinch and Peterson (1974). In this development, Pinch and Peterson formally prove that where two sections of a network are adjusted independently in a first stage adjustment, and the estimates of the parameters common to the two sections (with their variance matrices) are integrated in a second stage adjustment, the second stage estimates will be identical to those produced from a simultaneous adjustment. Whilst their method produces rigorous estimates for the junction station parameters and variance matrices, Gagnon (1976) notes that in the final phase of the process, rigorous variance matrix estimates are not produced for the inner[1] stations of each block.

The Canadian Section Method has been used widely in Australia for adjustments of the national network subsequent to the Australian Geodetic Datum 1966[2] (AGD66). These are referred to as various versions of the Geodetic Model of Australia (GMA) and include the adjustments termed GMA73, GMA80, GMA82 and GMA84, the latter of which led to the development of the Australian Geodetic Datum 1984 (AGD84) (Allman, 1983; Allman and Steed, 1980). The establishment of the North American Datum 1983 (NAD83) and the European Datum of 1987 (ED87) also made use of the Canadian Section Method, although they varied from Tienstra's approach in that they employed Helmert Blocking for the solution of the normal equations.

Around the time of the development of the early GMA adjustments, Cooper and Leahy (1977) and Leahy (1983) revisited the application of Tienstra's phased adjustment method to the adjustment of large geodetic networks. This led to further research (Leahy et al., 1986) and the development

1. Repeated mention of inner and junction stations will be made throughout this user guide. By way of preliminary definition, inner stations are the stations which are connected only by measurements in a single section, whereas junction stations are those which are connected by measurements from two or more sections. See §6.2 for more information.
2. Whilst the establishment of AGD66 was conducted in phases using the process of segmenting the network into smaller sections, the approach undertaken for the adjustment and integration of the respective sections didn't employ Tienstra's rigorous phased adjustment technique. This was largely driven by the challenges and complexities associated with undertaking a large scale adjustment on 1960's computational infrastructure.

of VicNet — a two–dimensional package designed to undertake phased adjustments of geodetic networks comprised of conventional terrestrial measurements. This software package was refined by Collier (1991) to accommodate three–dimensional adjustments and the integration of GPS baseline measurements. With these enhancements and other new capabilities and bug fixes, the software became known as MetNet and was used extensively for the adjustment of the Melbourne survey control network. Further research by Leahy (1999) led to the development of a fully automated network segmentation procedure which can handle networks of any size and configuration, and a rigorous approach for the extension and integration of networks.

Continued research on the automatic segmentation and phased adjustment of large geodetic networks gave rise to further refinements in the algorithm and the development of a new 32–bit Windows package (developed in Visual Basic 6) known as *Dynamic Network Adjustment (DNA)*. The initial development of DNA was undertaken by Leahy and Collier (1998) at the University of Melbourne and was made possible through funding provided by the Australian Research Council (ARC) and industry support from the Office of Surveyor–General Victoria, AUSLIG Geodesy and WBCM Surveys Pty Ltd. Following numerous enhancements and refinements in subsequent years, DNA was repackaged and released as *DynaNet* version 1.0. In turn, DynaNet 2.0 was developed to cater for new measurement types, and to provide several user enhancements.

1.1.1 DynaNet 3.0 — inital cross–platform version

Motivated by the need for a scalable, robust software package capable of undertaking repeated adjustments of continental–scale geodetic networks cromprised of millions of GNSS and terrestrial measurements, a new version of DynaNet was developed by the authors of this User Guide. Known as DynaNet 3.0, this version was completely re–engineered from scratch using C++ to provide cross–platform support (e.g. Windows, Linux, Mac OS X), and to take advantage of multi–core processors and modern supercomputing capabilities.

Through an Australian Cooperative Research Centre for Spatial Information (CRC–SI) project, several improvements were made to DynaNet 3.0, including the development of a comprehensive reference frame transformation library and various enhancements and bug fixes to the existing code base. Some of the software enhancements included replacing legacy Cholesky matrix inversion source code with external calls to Intel Math Kernel Library (MKL) Cholesky factorisation and inversion routines; adding functionality to rename stations, handle discontinuities, output of adjustment data in SINEX format, and elementary user interface tweaks; and modifications to existing file formats. This version has been used to maintain state geodetic networks, undertake digital cadastre adjustments, and was adopted for the computation of the GDA2020 adjustment Brown et al. (2018).

1.1.2 DynAdjust 1.0 — initial open source release

In view of providing a sustainable, long–term future for this software, and to maximise the benefits of releasing the software and code base to an international community of geodesists and developers, DynaNet 3.0 was refactored and released in 2018 as DynAdjust Version 1.0 under a Commonwealth of Australia Creative Commons Attribution 4.0 International Licence.

http://creativecommons.org/licenses/by/4.0/legalcode

1.1.3 DynAdjust Version 1.1 (current release)

The latest version of DynAdjust is available from the Intergovernmental Committee on Surveying and Mapping (ICSM) GitHub page: https://github.com/icsm-au/DynAdjust

The main specifications of DynAdjust Version 1.1, and the environments in which it has been compiled and tested include:

- Developed using C++ 14 and the C++ Standard Library;
- References Boost C++ Schäling (2014), Code Synthesis XSD Kolpackov (2017), Apache's Xerces C++ XML Parser Apache (2017) and Intel's Math Kernel Library Intel (2020);
- The modular architecture design and comprehensive API allows for development of new functional components and implementation via custom software development;
- Developed for Windows, Linux and Mac OS X platforms, and runs on 32–bit and 64–bit operating systems;
- Can be compiled using Microsoft C++ 2010 and 2017; Intel C++ 2014 and 2017; gcc 4.8.1 (C++ 11) up to gcc 10.1.1 (C++ 14). DynAdjust has not yet been compiled on C++ 17.
- Uses Cmake to handle compilation of binaries on Linux and Unix systems;
- DynAdjust has been compiled and successfully executed on the following operating systems:

Windows	XP, 7, 10 and 2003 Server.
Linux	CentOS 7 and 8; Fedora 27–32; OpenSUSE 10; Red Hat Enterprise Linux (RHEL) 7 and 8; Ubuntu 16–18.
Apple	Mac OS X LLVM (compiled using gcc 4.2.1 via clang-1001).

Geoscience Australia is committed to the active maintenance of DynAdjust through international collaboration. If you would like to formally express an interest in contributing to the development of DynAdjust, please contact `geodesy@ga.gov.au`.

Alternatively, to suggest an enhancement to the functionality of DynAdjust, or to report a defect or unexpected behaviour, please submit your query via the Issues page on the GitHub repository:

`https://github.com/icsm-au/DynAdjust/issues`

1.2 Conventions used in this document

The following typographical and mathematical conventions have been adopted throughout this document:

- DynAdjust program names are indicated by bold sans serif font. For example, the program **import** is the main program for importing data into DynAdjust.
- Program execution on the command line usage is denoted by fixed–width typewriter font. Examples of program execution at the command prompt are encapsulated by a grey box. If different syntax is required for Windows and UNIX/Linux platforms, syntax for both environments will be provided. Program options may be either inline with the text or placed within a grey box. For example, basic command line usage of **import** is given by:

```
> import -n network_name network.stn network.msr
```

  and the program option for specifying the input directory is `--input-folder`.
- File names and file contents are denoted by fixed–width typewriter font. File contents are encapsulated by a grey box with column positions shown in a separate box:

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
!#=DNA 3.00 STN     28.08.2013         GDA94                           43
```

- Math symbols are given in serif font. Variables are denoted by upper or lower case letters in italics, as in b or B. Matrices are denoted by upper case letters in bold font, as in $\mathbf{A}$, and vectors are denoted by lower case letters in bold font, as in $\mathbf{m}$. The identity matrix is denoted by $\mathbf{I}$, and the context in which it is used determines its dimensions. The term *variance matrix* is used to refer to the covariance matrix or variance–covariance matrix and is denoted by $\mathbf{V}$. Indexing of matrix and vector elements is denoted by $\mathbf{C}_{ij}$ or c_{ij}. Superscripts ^{T} and $^{-1}$ denote the transpose and inverse respectively. For a random variable x, the notation $E(x) \sim N(\mu, \sigma^2)$ means that x follows a Normal distribution with a mean or expected value of μ and variance σ^2.

1.3 Program overview

This section provides an overview of the DynAdjust software architecture and a summary of the various DynAdjust programs. The general philosophy of program execution and configuration via program options and arguments is also explained.

1.3.1 Software architecture and information flow

DynAdjust consists of several programs, the functionality of which is distributed across a number of executables and Dynamic Link Libraries (DLL) using a modular, service–oriented architecture. The modular architecture of DynAdjust affords several advantages, such as:

- Individual programs can be executed to perform a specific function relating to the processing and adjustment of geodetic networks;
- One or more programs can be chained together in a customised sequence to satisfy a specific user requirement;
- A sequence of program calls can be invoked via scripts at will, at scheduled times or as part of a larger automated datum maintenance environment;
- Routine and conventional processing of geodetic control surveys can be handled in a single step.

To assist with conventional end–to–end processing, the main program **dynadjust** serves as a wrapper application which can be used to coordinate the execution of one or more DynAdjust programs in a single step. The coordination of the various programs is illustrated in Figure 1.1.

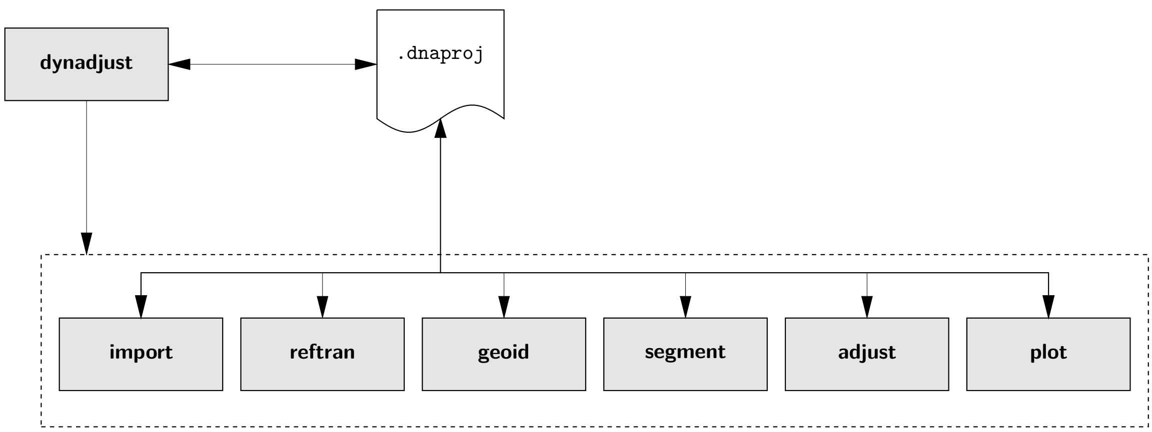


Figure 1.1: Coordination of DynAdjust program execution

A brief description of the various DynAdjust programs is given below:

import To create and process projects in DynAdjust, this program must be run first to define the network name, and to convert geodetic station coordinates and measurements from external file formats into the required binary and text file formats (see §2.2.2). Note that **import** only needs to be run once in order to import geodetic station coordinates and measurements into DynAdjust. Accordingly, **import** does not need to be run for repeated calls to other programs unless there is a change to the stations or measurements, or if some form of manipulation to the import process is required. Chapter 3 provides detailed information on how to use and configure **import**.

reftran This program performs reference frame transformations to align all imported stations and measurements to a common reference frame and epoch. See Chapter 4 for more information on using **reftran**.

geoid This program introduces geoid–ellipsoid separations and deflections of the vertical into a project from either an NTv2 formatted geoid model or ASCII text file. This program can optionally export interpolated geoid information to a DNA geoid text file. **geoid** also provides a capability to generate NTv2 formatted files from the legacy AUSGeoid DAT file format. See Chapter 5 for more information on using **geoid**.

segment This program segments a network into a series of inter–connected blocks for use by **adjust** in phased adjustment mode. See Chapter 6 for more information on using **segment**.

adjust This is the main parameter estimation program in DynAdjust. **adjust** can be executed in simultaneous or phased adjustment mode. When attempting to adjust large networks using phased adjustment, **adjust** may be executed in single–thread or multi–thread mode. For the former, users may opt to use staged mode if physical memory limits prevent normal program execution. A reverse, Block–1 only adjustment may also be undertaken if the desired outcome is a full variance matrix for a specific cluster of stations. See Chapter 8 for more information on using **adjust**.

plot This program generates PDF images of a network from the imported information and graphs relating to network segmentation and adjustment statistics. More information on **plot** and its use will be given in a subsequent version of this guide.

As is hinted by Figure 1.1, the coordination of DynAdjust programs is handled via a `.dnaproj` (DynAdjust project) file. The use of the project file will be discussed in more detail in Chapter 2.

The programs **import**, **reftran**, **geoid**, **segment**, **adjust** and **plot** read and write a range of formatted files in a way which permits the structured flow of information. Figure 1.2 illustrates the flow of information amongst the various programs and files.

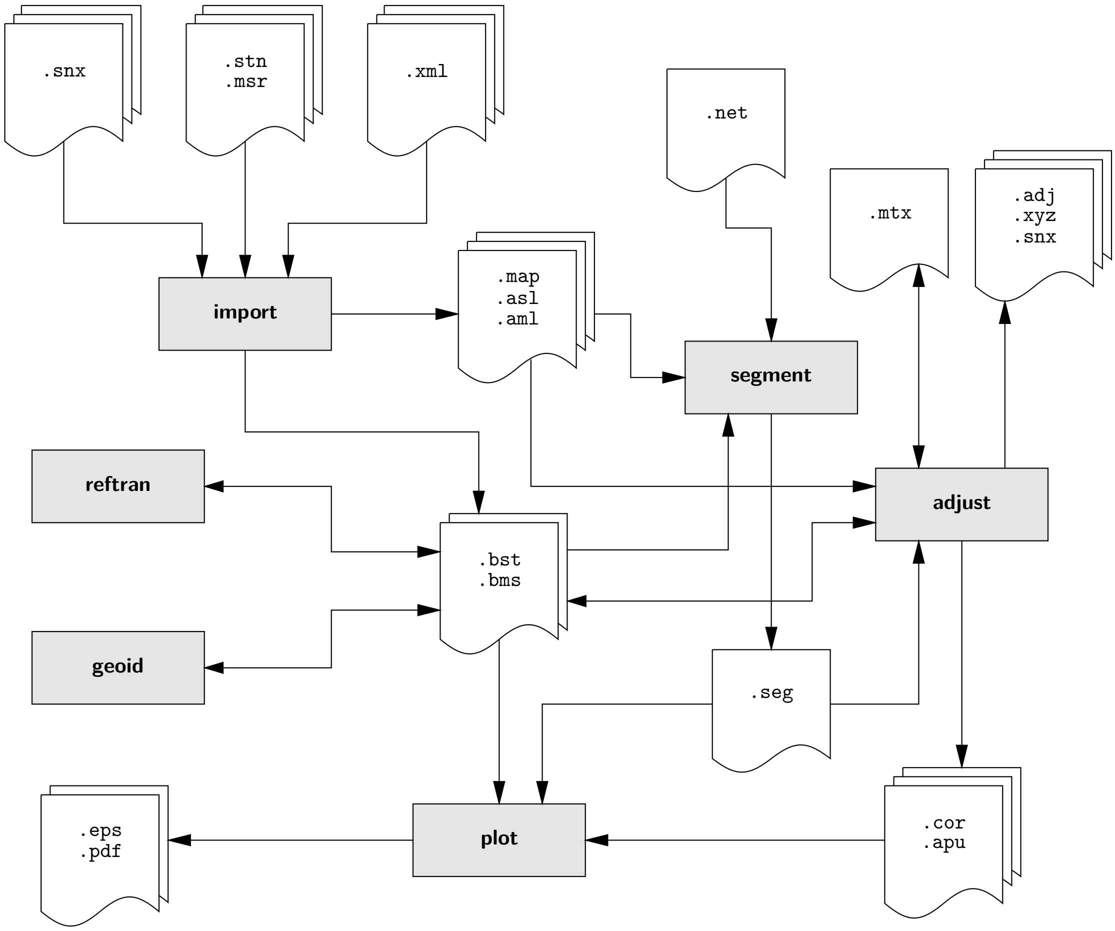


Figure 1.2: Flow of information between various DynAdjust programs and input and output files

As shown in Figure 1.2, information exported from one program often serves as input to other programs. This concept affords the user an ability to examine the output from one program before executing another program, and to re–run certain programs with different user options and to examine the variation in results. A brief description of the various file types is discussed in Chapter 2.

1.3.2 Program execution and command line options

DynAdjust programs can only be executed from the system command prompt (or *shell* or *terminal emulator*), via Windows batch files (`.bat` and `.cmd`) and UNIX/Linux shell scripts (`.sh`), or from system calls made within custom–developed programs. Hence, DynAdjust Version 1.1 does not provide a windows–based graphical user interface (GUI).

The convention for executing DynAdjust programs is to enter a program name, followed by one or more *options* that modify its behaviour and, if required, an *argument* upon which the program will act. The convention is as follows:

```
> program --option argument
```

The complete command line reference for all DynAdjust programs is provided in Chapter A. To display the command line reference for a DynAdjust program, type in the program name at the command prompt, followed by `--help` and press Enter:

```
> dynadjust --help
```

This will display all program options for **dynadjust**, organised into functional categories. To display the command line reference for a specific category of a program's functionality, type in the program name at the command prompt, followed by `--help-module` and a key word from the category name (e.g. `Standard options`, `Export options`, `Generic options`, etc.), then press Enter. For example, to display the command line reference on the export functions of **adjust**, type the following:

```
> adjust --help-module export
```

Any case–insensitive word or part of word (e.g. `expo`) from a help category heading can be used. Alternatively, if no program options are provided upon program execution, the program's version information and command line reference will be displayed on the screen.

There are over 140 program options which can be used to configure the way in which DynAdjust programs process geodetic network information. Each DynAdjust program will require certain options specific to its operation and depending on which option has been provided, additional arguments may be required. Several options, such as `--quiet`, `--version` and `--help` are common to all programs. If an option requires an argument, the command line reference will use the term `arg`.

All options are case–sensitive and must be preceded by two hyphens. Spaces must not be entered between the hyphen and the option text, however spaces are required between options. Options can be provided in any order whatsoever. Some options may be specified using an abbreviated form. For example, to display the version of a DynAdjust program, the abbreviated form `-v` may be used. In addition, all DynAdjust programs permit the use of partial option text, provided that the partial option text contains a sufficient number of characters to uniquely distinguish the required option from all other options. For example, the program **import** will export newly imported stations and measurements in DNA format if the option `--export-dna-files` is provided. This function can also be executed by providing `--export-d`. However, **import** will return an error if just `--export` is provided since there are five export options that commence with the text `export`.

When a program option requires an argument, input may require alpha–numeric entry and/or the selection of a multiple–choice option. If an argument must include spaces, such as a station name, enclose the argument with double quotes, such as:

```
> program --option "arguments with spaces"
```

For multiple choice options, DynAdjust will adopt the default value (denoted in the command line reference) unless the user overrides it by supplying an alternative argument value. For example, **import** provides an ability to specify the default reference frame for all stations and measurements contained in the user-supplied input files via the option `--reference-frame` (or `-r` in brief). As this option provides a multiple choice, only predefined reference frames are allowed. If this option is not provided, **import** will adopt the Geocentric Datum of Australia 1994 (GDA94) as the default value. Several options adopt this convention.

Following chapters will explain in detail the function of each program and how to configure program behaviour using the program options.

1.3.3 Program execution sequence

Figure 1.3 shows the program execution sequence employed by **dynadjust** when performing end–to–end processing. For the most part, this sequence will be adequate for conventional geodetic network processing and adjustment.

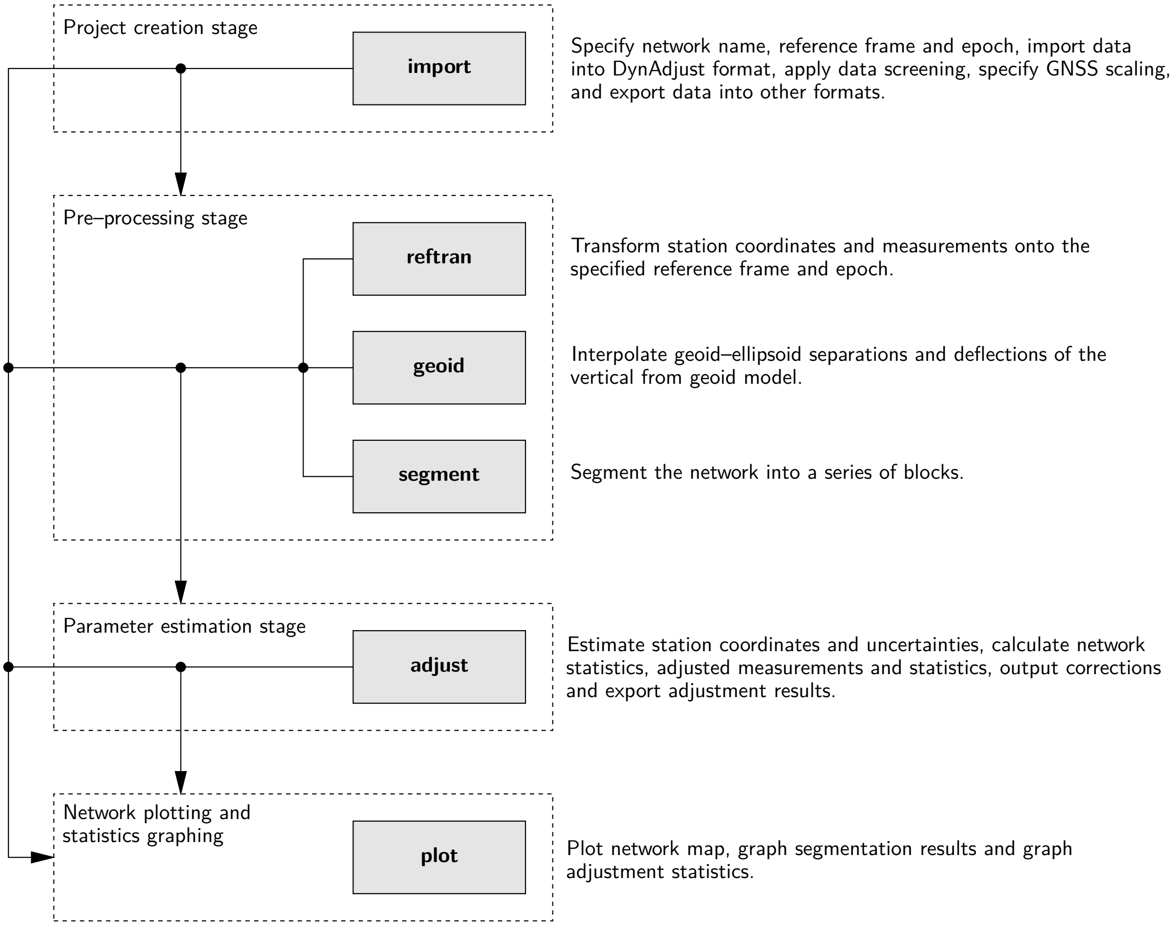


Figure 1.3: DynAdjust program execution sequence

There are two circumstances, however, where this sequence may not be appropriate or will need to be varied. Firstly, depending on the desired outcome, only some stages will be necessary. For this purpose, **dynadjust** provides the user with an ability to choose which DynAdjust programs to execute. The following examples explain some basic scenarios in which the user will require the execution of only a subset of the DynAdjust programs:

Ellipsoid–only adjustment To estimate coordinates on the ellipsoid from a small geodetic control survey of GNSS measurements which are aligned to a common reference frame, only **import** and **adjust** are required.

Multiple reference frames If the GNSS measurements in this survey are aligned to different reference frames, then the sequence **import**, **reftran** and **adjust** will be required.

Terrestrial measurements If terrestrial measurements (e.g. angles, distances and orthometric height differences) form part of this survey, then the sequence will change to **import**, **reftran, geoid** and **adjust**. Here, **geoid** is added to the sequence to obtain geoid–ellipsoid separations and deflections of the vertical which are used by DynAdjust to cater for the influence of gravity on the terrestrial measurements.

Generate basic network plot If only a plot of all stations and measurements in a network is required, then only **import** and **plot** are required.

Generate plot of error ellipses, uncertainties and corrections If the network plot should also include estimated error ellipses, circular confidence regions and *a–priori* station corrections derived from a least squares adjustment, then the sequence will be **import**, **adjust** and **plot**.

Secondly, network processing and adjustment may require data concatenation, screening (or filtering), scaling and multiple transformations between different reference frames before the network is suitable for processing by **adjust**. For these tasks, it is recommended that a script file be used to string together the needed program calls to achieve the required program execution sequence. In either case, knowing which programs to execute will require a knowledge of the data and an elementary knowledge of geodetic measurement, reference frames and adjustment theory.

1.4 Two–minute quick start tutorial

This section provides a quick start tutorial to processing geodetic network information in DynAdjust. The example used in this tutorial is a network comprised of six stations and nine GNSS baseline measurements. The the project is named `skye`, and the stations and measurements are contained in `2009-10-20-skye.stn` and `2009-10-20-skye.msr` respectively. Figure 1.4 shows the stations and measurements in the `skye` network.

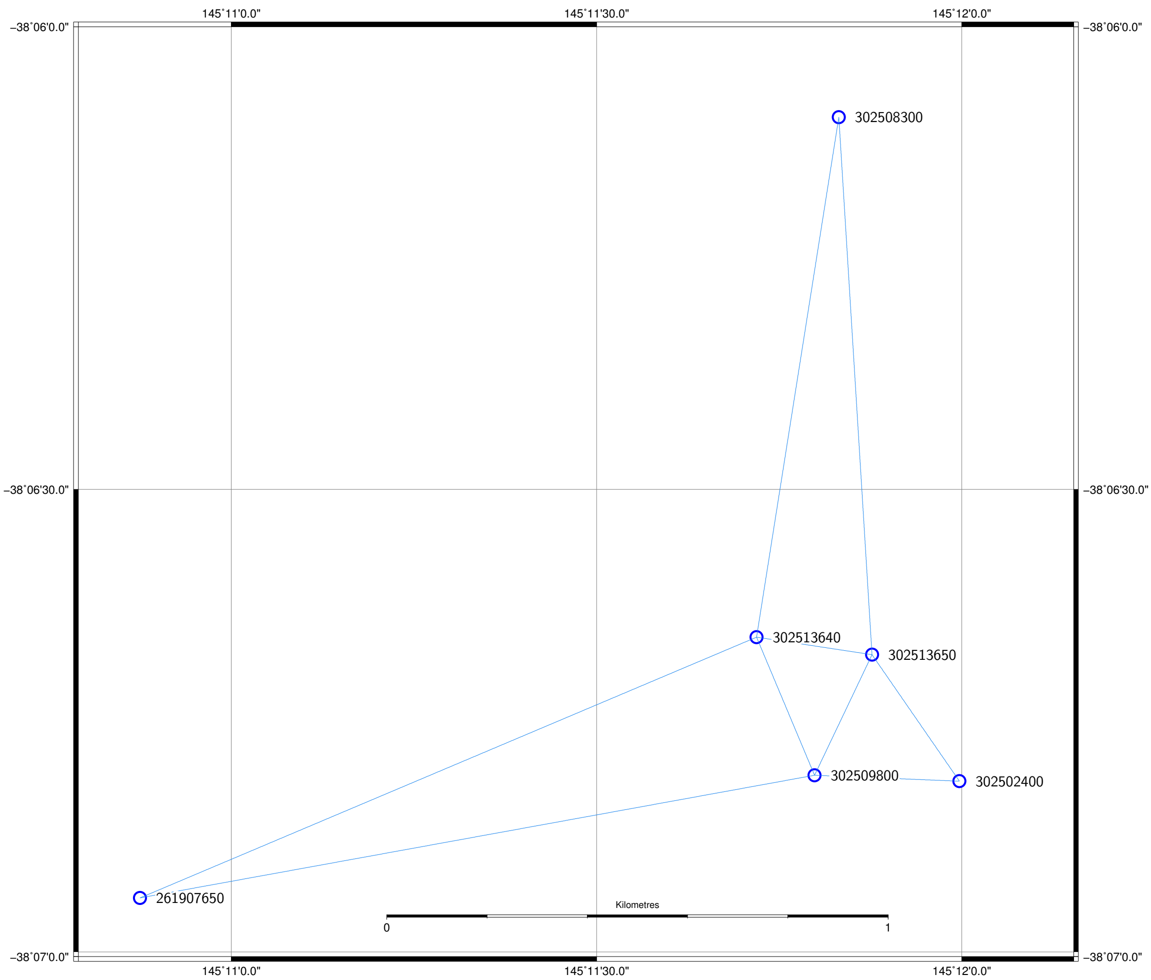


Figure 1.4: Stations and measurements in the `skye` network

The station file contains a mixture of ellipsoid heights and orthometric heights on the Australian Height Datum (AHD). The GNSS baseline measurements and variance matrices have been derived from conventional GNSS processing software and have been computed relative to the Geocentric Datum of Australia 1994 (GDA94). No scaling has been applied to the variance matrices.

Step 1 — Prepare the script

For this and many other examples provided in this user guide, it is assumed that users will want to make use of a script file to string together several DynAdjust program calls. Apart from some basic differences, calls to DynAdjust programs will be identical for both Windows and UNIX/Linux platforms and can therefore be copied directly to a Windows batch file or a UNIX/Linux shell script respectively.

For Windows platforms, prepare a file called `run_skye.bat` as follows:

```
echo off
rem Script to automate processing of skye geodetic network
```

For UNIX/Linux platforms, prepare a file called `run_skye.sh` as follows:

```
#!/bin/bash
# Script to automate processing of skye geodetic network
```

To execute this script using the UNIX/Linux shell, the execute permission for `run_skye.sh` will need to be granted to the current user (or group or all users). The file execute permission can be set by:

```
$ chmod +x ./run_skye.sh
```

Alternatively, this script can be executed from the command line using bash:

```
$ bash ./run_skye.sh
```

Step 2 — Import the data

The next step to be undertaken is to create a project named skye. In this step, a call to **import** is required to name the project and to introduce the station and measurement files. Setting the network name and importing files into DynAdjust is achieved by:

```
import -n skye 2009-10-20-skye.stn 2009-10-20-skye.msr
```

Upon running this command, the project `skye` is created and several files will be written to disk: `skye.bst`, `skye.bms`, `skye.map`, `skye.asl`, `skye.aml` and `skye.imp`. See Chapter 2 for an explanation of these file types.

Step 3 — Introduce geoid-ellipsoid separations

Since DynAdjust requires all station heights to be reduced to the ellipsoid prior to adjustment, the next step is to convert the orthometric station heights to ellipsoid heights. For this purpose, interpolation of geoid–ellipsoid separations from a geoid model is needed. To this end, the script would be expanded as follows:

```
import -n skye 2009-10-20-skye.stn 2009-10-20-skye.msr
geoid skye -g ./ausgeoid09.gsb --convert-stn-hts
```

Here, geoid-ellipsoid separations will be interpolated from `ausgeoid09.gsb`, which is a geoid grid file structured according to the National Transformation version 2.0 (NTv2) format. The option `--convert-stn-hts` informs **geoid** that all orthometric station heights should be converted to ellipsoid heights. After this command is executed, all stations in the binary station file `skye.bst` will contain geoid-ellipsoid separations and deflections of the vertical, and any orthometric heights will be reduced to ellipsoid heights.

Step 4 — Adjust the network

DynAdjust offers the flexibility to undertake constrained and minimally constrained (or *free*) least squares adjustments in a number of ways. Adjustments may also be performed in a single pass via simultaneous mode or sequentially in phases using phased adjustment mode. For this tutorial, the `skye` network will be adjusted using no constraints and, given the relatively small size of this control survey, via simultaneous mode. Assuming that station constraints have not been introduced into either station or measurement files, the script would be expanded as follows:

```
import -n 2009-10-20-skye.stn 2009-10-20-skye.msr
geoid skye -g ./ausgeoid09.gsb --convert-stn-hts
adjust skye --output-adj-msr
```

From this call to **adjust**, the resulting adjustment output file will be called `skye.simult.adj`. The option `--output-adj-msr` was added so as to print a table of adjusted measurements and statistics to the `.adj` file.

If it is decided that a station should be constrained, users can choose one of three options to apply station constraints — (1) set the station constraint flag within the station file, (2) add a station position measurement to the measurement file, including the measurement precision by which to constrain the station, or (3) specify the station and how it is to be constrained via the call to **adjust**. The third option, which in effect replicates the first option, can be achieved by the following change to the call to **adjust** (using station `302513640` as an example):

```
adjust skye --output-adj-msr --constraints 302513640,CCC
```

This section has provided a simple tutorial on using DynAdjust to perform a straightforward network adjustment of GNSS observations. Detailed help on program usage for numerous other processing tasks will be provided throughout the remainder of this user guide.

Chapter 2

Creating, editing and processing projects

2.1 Introduction

DynAdjust uses the concept of a *project* to manage the input, processing and output of geodetic network information. For each project, DynAdjust uses a project file to store default and user specified options. The user options contained in a project file configure the way in which the respective DynAdjust programs handle the geodetic network information relating to a project. At execution time, each program can be configured by providing a project file path as a program argument, or by specifying the respective options as program arguments. Since DynAdjust Version 1.1 can be executed from the system command prompt or from custom–developed programs, project files can be used to completely automate the processing of geodetic networks, and to capture the options used during program operation.

This chapter explains the various conventions used by DynAdjust for managing projects, how to prepare input files, and provides an overview of the basic program operation. More information about the various options for each program will be explained in subsequent chapters.

2.2 Conventions used in DynAdjust

2.2.1 File naming

Central to the management of projects is the *network name*. DynAdjust uses the network name to form the file names for all generated output files, and to determine which file to open when information generated from one program must be read as input by another program. The basic file naming convention is represented by `network_name.ext` where `network_name` is the user–supplied network name and `ext` is the file extension. In some cases, DynAdjust generates files in the form of `network_name.mode.ext` where `mode` represents either a mode in which a program has been executed or a user–specified program option.

To permit the input of files with a different file name to that which is expected from the default naming convention, the file name for certain input files may be overridden by providing the relevant command line argument. This feature will be covered in more detail in subsequent chapters describing the input and output options of the various programs.

The network name is a mandatory argument required by all DynAdjust programs except **import**. If a network name is not specified when running **import**, DynAdjust adopts the name '`network#`' where '`#`' represents the next available integer that yields a unique (or unused) file name in the folder of program execution.

The primary exceptions to the file naming convention are station and measurement files (e.g. `*.snx`, `*.stn`, `*.msr`, `*.xml`) provided as input to **import**, and the raw data files and formatted grid files (e.g. `*.dat`, `*.gsb`) provided as input to **geoid**. No restrictions are imposed upon the naming of these files other than that the file extension corresponds with the file format. The file extension restriction is imposed only for certain file types which prevent DynAdjust from automatically interpreting file content.

2.2.2 File types

As shown in Figure 1.2 on page 7, DynAdjust creates and/or updates a number of binary and text files, which are treated as either output and/or input files. The following is a list of file types created by DynAdjust in accordance with the file naming convention (assuming the network name is `network_name`):

Binary formatted file types

`network_name.aml`	Associated Measurements List. This file contains a list of measurements that a station appears in.
`.asl`	Associated Stations List. This file contains a count of the measurements connected to a station and the index of this station in the AML file.
`.bms`	Binary Measurements file. This file contains information about the measurements in a network. The BMS file is a binary formatted file created using an efficient file structure to provide maximum efficiency for retrieving measurement information.
`.bst`	Binary Stations file. This file contains information about the stations in a network. The BST file is a binary formatted file created using an efficient file structure to provide maximum efficiency for retrieving station information.
`.map`	Station Map. This file maintains the relationship between the supplied alphanumeric name and a unique (numerical) station identifier.
`.mtx`	Matrix file. This file stores matrices in a structured file format designed for efficient data storage.
`.dbid`	Database ID list. This file contains the user–supplied database IDs for measurements contained in DNA and DynaML formatted measurement files.

ASCII text file types

`network_name.imp`	**import** log. This file is a log of the station and measurement import process.
`.seg`	**segment** output. This file contains the station and measurement indices for the respective blocks created from network segmentation.
`.adj`	**adjust** output. This file contains the adjusted station coordinates and measurements and associated statistics.
`.xyz`	Adjusted Station Coordinates produced by **adjust**.
`.cor`	Station Coordinate Corrections produced by **adjust**. This file contains the corrections to the initial station coordinates.
`.apu`	Adjusted Positional Uncertainty produced by **adjust**. This file contains the positional uncertainties of the adjusted station coordinates.

`.dbg`	Debug output. This file contains detailed program output information to help assist with isolating network adjustment problems.
`.dst`	Duplicate Stations list. This file contains a list of stations that were identified as duplicates by **import**.
`.dms`	Duplicate Measurements list. This file contains a list of measurements that were identified as duplicates by **import**.
`.log`	**dynadjust** log. This file provides a time–stamped record of the progress of the individual programs that have been coordinated by **dynadjust**.

2.2.3 Directories

By default, all DynAdjust programs expect input files to exist in the directory in which the programs are run. DynAdjust will also generate output files in this directory. Optionally, an input folder and an output may be specified to inform the DynAdjust programs where to find input files and where to store output files. This feature will be covered in more detail in subsequent chapters.

2.3 Project setup and processing

2.3.1 Prepare station and measurement files

The first step in creating a project is to prepare the station and measurement files. DynAdjust supports a small range of file types. Appendix B provides a list of supported file types and the format specification for selected file types. Stations and measurements may be provided in one or more files, each of which may be encoded in any one of the supported file formats. DynAdjust does not impose any restrictions on how this information should be structured and so the user is left to decide which file type is chosen and how station and measurement information will be stored.

2.3.2 Create DynAdjust project file

In order to process projects in a single step using the main program **dynadjust**, a project file must be created. Note that a project file does not need to be created if the various DynAdjust programs will be executed manually, or executed via Windows batch files or UNIX/Linux shell scripts. In these cases however, a project file will be created and updated automatically as each program is executed.

There are two options for creating a DynAdjust project file — users may create this file manually or use **import** to create this file automatically. The file format for the project file is described in §B.2. Each time **import** is executed, a new project file will be created using the network name and the default or user–specified output folder path. If this file exists, it will be re–created using the options and arguments supplied. Options which have not been provided will assume default values. All other options and arguments supplied to **import**, such as network name and station and measurement files, will be printed to project file. Users not familiar with the project file format are encouraged to use **import** to create the project file, and to use a text editor to modify the project file with the desired options and arguments.

2.3.3 Automated project processing

Upon creating a project file, projects can be processed in a single step using the main program **dynadjust**. Using the project file shown in §B.2 for the `skye` project (see §1.4), the following command illustrates how end–to–end processing can be performed in a single step using **dynadjust**.

```
dynadjust -p skye.dnaproj --import --geoid --adjust
```

The first option (-p) and argument (skye.dnaproj) inform **dynadjust** where to find the project file. Since the folder path was not provided on the command line, DynAdjust will assume the project file is located in the folder from which **dynadjust** is executed. As shown by the general section in Figure B.13, §B.2, this folder is C:\Data\proj.

The second, third and fourth options tell **dynadjust** to execute **import**, **geoid** and **adjust** respectively using the options specified in the project file. The order of these options is not important since **dynadjust** will adopt the program execution sequence shown in Figure 1.3. With respect to **import**, the mandatory network name and input files are entered into the project file using --network-name (under #general) and --stn-msr-input-file (under #import) respectively. All other options use default arguments. For **geoid**, the NTv2 geoid grid file path and conversion of orthometric heights to ellipsoid heights are handled by --ntv2-filepath and --convert-stn-hts (under #geoid). Finally, configuring **adjust** to perform a simultaneous adjustment, holding station 302513640 fixed, and to produce a table of adjusted measurements and statistics in the adjustment output (.adj) file is managed by --adjustment-mode and --constraints (under #adjust), and --output-adj-msr (under #output). Note that the options and arguments under #reftran and #segment are ignored since the **dynadjust** argument to execute **reftran** (-reftran) was not provided, and **adjust** was configured to run a simultaneous adjustment.

As **dynadjust** runs, it will load the options contained in the project file and pass them to the respective programs. Upon execution, a log file named dynadjust.log will be created and will contain a time–stamped record of the progress of the individual programs that have been executed. A sample of the log file is shown below.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
```

```
+---------------------------------------------------------------------------
+ Title:         dynadjust
+ Description:  Geodetic network adjustment software
+ Version:       3.2.6, Release (64-bit)
+ Build:         Sep 22 2016, 16:23:21 (MSVC++ 10.0)
+ Copyright:     (C) 2016 F.J.Leahy, P.A.Collier, R.W.Fraser.
                 This software is released under a restricted license.
+ Contact:       geodesy@ga.gov.au
+                +61 407 062 507
+---------------------------------------------------------------------------


+ Executable   Start date and time      End date and time        Exit status
------------------------------------------------------------------------------------------
+ import       11-04-2014 6:04:06 PM    11-04-2014 6:04:07 PM    Ended successfully.
+ geoid        11-04-2014 6:04:07 PM    11-04-2014 6:04:08 PM    Ended successfully.
+ adjust       11-04-2014 6:04:08 PM    11-04-2014 6:04:09 PM    Ended successfully.

+ DynAdjust finished successfully.
```

Chapter 3

Import and export of geodetic network information

3.1 Introduction

DynAdjust supports a number of file formats for the exchange of geodetic network information. The primary file formats supported by DynAdjust, Version 1.1 for the import and export of stations, measurements and adjustment solutions include:

DNA (Dynamic Network Adjustment format) for the exchange of stations, measurements and geoid information in a simple, human–readable format. Versions 1 and 3 are supported.

DynaML (DynAdjust Markup Language format) for the exchange of stations and measurements in XML format defined according to the DynAdjust 2.0 XML schema

GeodesyML (Geodesy Markup Language format) for the exchange of station, measurement, solution and covariance information in XML format defined according to ICSM's eGeodesy GML application schema

SINEX (Solution-INdependent EXchange format) for the exchange of solutions and covariance information

Appendix B provides the file format specification for the DNA and DynaML file types. The specification for GeodesyML can be found at `http://geodesyml.org`. The specification for the SINEX file format can be found at

`http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html`

DynAdjust also reads and writes a number of binary and ASCII file formats for the exchange of geodetic network adjustment information. These will be described throughout this section as the need arises.

3.2 Importing station and measurement information

The primary program for importing and converting geodetic network stations and measurements into the format required by DynAdjust is **import**. To import station and measurement information into DynAdjust, type `import` at the command prompt, followed by one or more options and arguments. The complete command line reference for **import** is given in §A.2. If no program options are provided upon program execution, the command line reference for **import** will be displayed.

If a DynAdjust Project File (c.f. Chapter 2) is to be used, provide `--project-file` (or its shortcut `-p`) followed by the full path for the project file:

```
import -p uni_sqr.dnaproj
```

No other options are required, as **import** will be invoked using the options and arguments contained in the project file `uni_sqr.dnaproj`.

Alternatively, if all options and arguments are to be entered via the command line, type `import`, then `--network-name` (or its shortcut `-n`) followed by the network name and the names of the files that contain the station and measurement information:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml
```

If the `--network-name` option and argument are omitted, **import** will use the default "`network1`" name. If there is evidence in the current directory that "`network1`" has already been created, then "`network2`" will be used. The integer appended to "`network`" increases until the first available network name is found.

Whether using a project file or not, additional options and arguments can be supplied to configure the way in which this information is imported. When using a project file, any additional options and arguments provided on the command line will overwrite the values contained in the project file.

3.2.1 Station coordinate information

import accepts station coordinate information in three commonly accepted coordinate systems, namely, Geographic coordinates, Universal Transverse Mercator (UTM) projection coordinates, and Earth–centred cartesian coordinates. Geographic coordinates are expressed as *latitude*, *longitude* and *height*, where height may be *ellipsoidal* height or *orthometric* height. UTM projection coordinates are expressed as *Easting*, *Northing*, *zone* and *height*. Earth–centred cartesian coordinates are expressed as *X*, *Y* and *Z*. These coordinate systems and the methods by which DynAdjust handles transformations between them are explained in Chapter 4. With reference to the supported file formats described in the introduction, Table 3.1 lists the coordinate types handled by the respective file formats.

Table 3.1: Station coordinate types handled by the supported file formats

Format	**DNA**	**DynaML**	**GeodesyML**	**SINEX**
Geographic (d.mmss)	•	•	•	•
Geographic (d.dddd)			•	
UTM Projection	•	•	•	
Cartesian	•	•	•	•

Appendix B explains the formatting of station coordinate information for each of the supported file formats. §3.2.3 explains the concept of station constraints and how they are interpreted by DynAdjust. Configuring the default reference frame for stations is described in §3.3.1.

3.2.2 Supported measurement types

DynAdjust supports a wide range of GNSS and terrestrial measurement types. To facilitate the efficient management of these types, DynAdjust uses the concept of a measurement code. The measurement types and corresponding codes supported by DynAdjust are listed in Table 3.2.

Table 3.2: Supported measurement types and measurement codes

Code	Measurement type
A	Horizontal angles (uncorrelated)
B	Geodetic azimuth (or bearing)
C	Ellipsoid chord distance
D	Direction set
E	Ellipsoid arc distance
G	Single GNSS baseline $(\Delta x\, \Delta y\, \Delta z)$
H	Orthometric height
I	Astronomic latitude
J	Astronomic longitude
K	Astronomic (Laplace) azimuth
L	Orthometric height difference
M	Mean sea level (MSL) arc distance
P	Geodetic latitude
Q	Geodetic longitude
R	Ellipsoid height
S	Slope (direct) distance
V	Zenith distance
X	GNSS baseline cluster (full correlations)
Y	GNSS point cluster (full correlations)
Z	Vertical angle

A brief description of these measurement types, including the observation equations implemented within DynAdjust for relating them to the unknown parameters are described in §7.2. Appendix B explains the formatting of measurement information for each of the supported file formats.

3.2.3 Network constraints

DynAdjust permits networks to be constrained using two different forms — via station constraints and position measurement constraints. Multiple constraints of both forms may be applied to a network.

Station constraints

Station constraints can be imposed on a network as either *free* (`'F'`) or *constrained* (`'C'`), and either constraint may be applied to one, two or three station cardinals (hereafter referred to as parameters).

Station constraints may be provided in DNA, DynaML and GeodesyML formats (see Appendix B for help on formatting for the respective file formats).

Station parameters which the user considers fixed can be *constrained*, so that they will not be subject to variation by least squares adjustment. Station parameters held *free* are those in which the user expects variation according to network geometry, measurements and their uncertainties, and other station parameter constraints. Stations held constrained in three dimensions define the datum (position, rotation and scale) for other stations held free in the network. Care should be exercised when applying multiple station constraints, as constraining multiple stations with poor coordinate estimates can lead to network distortions and/or cause several measurements to fail (c.f. §9.3.2). By default, free stations are assigned an uncertainty of 10.0 m, and constrained an uncertainty of 0.001 mm. To alter these values, refer to the paragraph on default constraint values in §8.3.3 on page 125.

When all unknown parameters in a network are held free, the solution of the parameters is undertaken as a *free network adjustment*. Accordingly, the estimation of parameters derives solely from the measurements. When the minimum number of parameters required to estimate all dimensions of the network are constrained, the solution is undertaken as a *minimally constrained adjustment*. In this case, poor coordinate estimates for the fixed station will not distort the network nor cause measurement failures. The exception to this of course is if position measurements are supplied for the constrained station and there is a significant difference between the constrained station coordinates and the position measurements. In the absence of these inconsistencies, both free network adjustments and minimally constrained adjustments may be used to test the least squares adjustment and the reliability of the measurements and their uncertainties (see §9.3).

DynAdjust supports the application of any combination of free and constrained parameter constraints to any number of stations in a network. The advantage of combining Free and Constrained parameter constraints is illustrated as follows. At times, there may be a single parameter (e.g. a station height) which cannot be estimated from the available measurements. At other times, a particular plane or axis may contain no measurements and as such, a one–dimensional or two–dimensional adjustment is necessary. In either case, station constraints may be applied to fix parameters for which estimation is not required without compromising the least squares adjustment.

Position measurement constraints

Like constrained stations, position measurement constraints provide a means for defining the geodetic datum for all stations in a network. However, position measurement constraints are somewhat different in that they permit the least squares adjustment to vary the parameter estimates in accordance with network geometry, measurements and their uncertainties, and any other station parameters held constrained. Providing a position measurement with an extremely small value of uncertainty is identical to holding a station constrained (or fixed), whereas providing a position measurement with an extremely large value of uncertainty is identical holding a station free.

The measurement types which act as positional constraints include GNSS point cluster (Y), geodetic latitude (P), geodetic longitude (Q), astronomic latitude (I) astronomic longitude (J), orthometric height (H) and ellipsoid height (R). Hence, to constrain a network horizontally, provide P and Q measurements for the station to be constrained together with appropriate standard deviations to reflect the amount by which the station should be constrained. Similarly, H or R measurements with appropriate standard deviations may be supplied to constrain a network in the vertical direction.

A GNSS point cluster with full variance matrix (e.g. obtained from a GNSS Continuously Operating Reference Station (CORS) network, made available via a SINEX file) is the most rigorous approach for constraining a network in three dimensions, and for defining a datum for the stations of a network. As with station constraints, care should be exercised when supplying position measurements as poor measurement estimates or over–optimistic values of uncertainty can lead to network distortions and/or cause several measurements to fail.

3.2.4 Geoid information

In order to rigorously combine measurements subject to the influence of the anomalous gravity field with GNSS measurements within a single adjustment, geoid–ellipsoid separations and deflections of the vertical are required. For this purpose, **geoid** facilitates the interpolation of the required geoid information from a structured geoid grid file (see Chapter 5).

import also provides a capability for importing geoid information into DynAdjust for two primary purposes. Firstly, there may be instances where more accurate and higher resolution geoid information has been observed over an area but is not yet available within a geoid grid file. In this case, all information about a geodetic network can be imported in a single step. Secondly, **import** offers an option to simulate measurements observed in the local reference frame which contain realistic measures of geoid–ellipsoid separations and deflections of the vertical (see §3.4).

To introduce geoid information into DynAdjust, add `--geo-file` to the list of options for `import`, followed by the geoid file name:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --geo-file uni_sqr.geo
```

See §B.1.5 for help on formatting geoid information using the DNA geoid file format.

3.2.5 File name argument conventions

When supplying **import** with options and arguments relating to input station and measurement files, DynAdjust offers a few conventions to simplify the process of entering file names. Firstly, supplying a file name does not require the `--stn-msr-input-file` option (or its shortcut `-f`). Secondly, any number of station and measurement files may be supplied, whether of the same or different file formats. Thirdly, input files can be supplied in any order. The following hypothetical example shows how a number of files and file formats may be used:

```
import -n my_net constraints.snx gps_bsl.xml level.msr level.stn
```

This example creates a new network called "`my_net`", which is comprised of station constraints in `constraints.snx`, GNSS baselines and associated stations in `gps_bsl.xml`, and spirit levelling measurements in `level.msr` and stations in `level.stn`.

3.2.6 Progress reporting and import log

Once **import** has been executed with the relevant options and arguments, **import** will report to the screen a number of items relevant to the project (e.g. name, location of input and output files); the progress of file reading; summary of loaded station and measurement information; progress of various processing tasks; and final exit status (i.e. success or failure). Figure 3.1 shows the information reported to the screen upon program execution.

```
+ Options:
  Network name:                     uni_sqr
  Input folder:                     .
  Output folder:                    .
  Associated station file:          ./uni_sqr.asl
  Associated measurement file:      ./uni_sqr.aml
  Binary station output file:       ./uni_sqr.bst
  Binary measurement output file:   ./uni_sqr.bms
  Default reference frame           GDA94

+ Parsing:
  uni_sqrstn.xml...     Done. Loaded 149 stations in 0.035s
  uni_sqrmsr.xml...     Done. Loaded 1199 measurements in 0.04s

+ Reducing stations... Done.

+ File parsing summary:

  Read 149 stations:
 -------------------------------------------
  (CCC) 3D constrained:                   1
  (FFF) 3D free:                        145
  (CCF) 2D constrained, 1D free:          1
  (FFC) 2D free, 1D constrained:          1
  (CFF) custom 2D constraints:            1
 -------------------------------------------
  Total                                 149

  Read 1199 measurements:
 -------------------------------------------
  (A) Horizontal angle:                 251
  (B) Geodetic azimuth:                   1
  (C) Chord dist:                         0
  (D) Directions:                         0
  (E) Ellipsoid arc:                      0
  (G) GPS baseline:                     114
  (H) Orthometric height:                 1
  (I) Astronomic latitude:                0
  (J) Astronomic longitude:               0
  (K) Astronomic azimuth:                 1
  (L) Level difference:                  89
  (M) Mean sea level arc:                 1
  (P) Geodetic latitude:                  0
  (Q) Geodetic longitude:                 0
  (R) Ellipsoidal height:                 0
  (S) Slope distance:                   428
  (V) Zenith angle:                     300
  (X) GPS baseline cluster:               0
  (Y) GPS point cluster:                 12
  (Z) Vertical angle:                     1
 -------------------------------------------
  Total                                1199*
  * Includes 17 ignored measurements

+ Testing for duplicate stations... Done.
+ Sorting stations... Done.
+ Serialising station map... Done.
+ Mapping measurements to stations... Done.
+ Creating association lists... Done.
+ Serialising association lists... Done.
+ Serialising binary station file uni_sqr.bst... Done.
+ Serialising binary measurement file uni_sqr.bms... Done.

+ Total file handling process took 0.697s.
+ Binary station and measurement files are now ready for processing.
```

Figure 3.1: **import** progress reporting of data import, conversion and processing

The amount of information will vary depending on the options and arguments supplied to **import**, such as folder locations, input files, file contents, reference frame, data screening, GNSS measurement scaling and export. This information is also logged to a file named `uni_sqr.imp`. Any errors encountered during the various tasks will result in premature program termination and a description of the error.

3.2.7 Verification and error checking

When **import** is executed, a number of verification and error checking tasks are undertaken upon loading the information into DynAdjust. These tasks are briefly summarised below.

Empty strings

As data is loaded from the input files, **import** will verify that non–empty strings exist for mandatory fields and elements. The fields and elements which are mandatory will depend upon which station coordinate type and measurement is being supplied. For instance, zone is not required when supplying station coordinates in geographic format. Instrument and target heights are only required for slope distances, vertical angles and zenith angles. Refer to the file format specification in Appendix B for a list of mandatory fields and elements for the respective station coordinate types and measurement types.

Testing for duplicate stations and measurements

When supplying **import** with station and measurement information either in one file or in a number of files, **import** will identify and remove *duplicate* station information. Although input file arguments may be supplied in any order, **import** processes input files sequentially in the order they have been entered on the command line. If a station contained in an earlier input file is found in a subsequent file, the latter station will be regarded as a duplicate station and thereby ignored. All duplicate stations are printed to a file with a `.dst` extension. If it is essential that duplicate station information be maintained across multiple files, then the user must ensure that the file which contains the *desired* station information (e.g. station name, coordinates and constraints) is loaded first.

When concatenating networks contained in different files, the same station can sometimes appear in the station and measurement files twice with different station names. To assist users with the detection of two differently named stations which are intended to represent the same station, **import** provides the `--search-nearby-stn` option. See §3.3.2 for more information on using this option.

As for measurements, it is quite normal for several repeated measurements to be observed over the same set of stations. Sometimes, measurement files can contain two independent measurements with equal values. For this reason, **import** does not attempt to remove measurements which appear to be identical and so the responsibility for checking for duplicate measurements falls to the user. **import** affords the ability to test for duplicates via the `--search-similar-msr` option. See §3.3.2 for more information on using this option to search for duplicate measurements.

Station and measurement connectivity

Two tests are performed to ensure that the supplied station and measurement information is valid in the context of network connectivity. Firstly, a test is performed to verify that the station file(s) contain information about all stations connected to the measurements. For this test to pass, the station names referenced in each measurement must exist in the station file exactly as they appear in the measurements. This test is case sensitive and includes whitespace. Any stations which are not found in the station file will cause **import** to terminate prematurely with an error message. Secondly, any stations which are not connected to any measurements will be noted as unused and marked for exclusion from segmentation and adjustment processes. See the section on Flagging unused stations on page 33 for configuring the way in which unused stations are handled.

3.3 Configuring import options

3.3.1 Reference frame

DynAdjust provides a capability to transform station coordinates and GNSS measurements between several static and dynamic reference frames via **reftran** (see Chapter 4). The purpose of this functionality is to align stations and GNSS measurements on multiple reference frames to a common reference frame prior to least squares adjustment. To this end, **reftran** expects all stations and measurements to be tagged with a recognised reference frame abbreviation. The reference frames and abbreviations supported by DynAdjust are listed in Table 4.1 of Chapter 4. If the reference frame is dynamic, **reftran** will also require an epoch.

Since reference frame information is not always present in DNA, DynaML, GeodesyML and SINEX input files[1], **import** provides two options to efficiently associate a reference frame to stations and measurements upon loading the input files.

Default Reference frame

Before any input files are loaded, **import** by default assumes that the reference frame for all incoming stations and measurements is GDA2020. If an input file does not contain reference frame information, this default reference frame option will be adopted. However, if an input file contains reference frame information, this information will be retained. To specify a different default reference frame to be used for all stations and GNSS measurements contained in the input files, add the option `--reference-frame` (or its shortcut `-r`) followed by the reference frame abbreviation to the list of **import** options. For example, to assign the International Terrestrial Reference Frame 2008 (ITRF2008), run the following command:

```
import -n igs_net constraints.snx gps_bsl.xml level.stn level.msr -r ITRF2008
```

In this example, **import** will set ITRF2008 as the default reference frame for the GNSS Point cluster measurement loaded from `constraints.snx`, all GNSS baseline measurements loaded from `gps_bsl.xml`, and for all stations from `gps_bsl.xml` and `level.stn`. This is effected by setting the default reference frame abbreviation in the binary station (`.bst`) and binary measurement (`.bms`) files. Any GNSS measurement which does not have a reference frame abbreviation assigned to it will adopt this default abbreviation. The default epoch assigned to the GNSS baselines in `gps_bsl.xml` will be the reference epoch of the user–supplied frame. The reference epoch contained in the SINEX file `constraints.snx` is not altered by this option. Similarly, any GNSS measurements loaded from a measurement file that have already been assigned a reference frame will not be altered.

1. According to the SINEX 2.02 specification, a SINEX file must contain a reference epoch for the solution, however there is no provision for storing the reference frame. Both DynaML and GeodesyML contain elements for managing reference frame and epoch, although this information is not mandatory. DNA Version 3 provides the ability for reference frame and epoch to be stored, but again this information is optional. See Appendix B for more information.

Override input reference frame name

The option `--reference-frame` sets the default reference frame and thereby only comes into effect if a GNSS measurement has not already been assigned a reference frame (see, for example, Table B.8 in Appendix B). To override the assigned measurement reference frame with the default reference frame, provide the option `--override-input-ref-frame`.

Handling multiple different reference frames

Note that option `--reference-frame` will set the default reference frame for all input files in a single step. If a project is based upon multiple input files aligned to different reference frames, a blanket application of a single reference frame to all stations and measurements may lead to non–rigorous results. To ensure the correct reference frame is associated with the imported stations and measurements, a number of processing steps may be required. For instance, some or all of the following may be required:

- Where file format specification allows (e.g. DNA and DynaML formats. See B.1), set the default and measurement–specific reference frame of the stations and measurements within in the input file;
- Where file formats do not make provision for setting the default reference frame, use a two–step approach:
    1. Group files together which are aligned to a common reference frame. Set the default reference frame upon loading the input files with **import**.
    2. Execute **reftran** with the desired reference frame and export to the desired file format (see Chapter 4).
    3. Repeat steps 1. and 2. until the appropriate reference frame and epoch have been assigned to all stations and measurements.

Further details on performing reference frame transformations will be provided in Chapter 4.

3.3.2 Data screening

When undertaking network maintenance tasks, it is often necessary to screen or filter input data using some specialised criteria to produce a thinned or optimal subset of the network. At other times, it may be necessary to search input data for inconsistent, incorrect or duplicate information. The data screening helps provided by **import** include extracting stations and measurements based on geographical region, network connectivity, network segmentation, and measurement type. Helps for searching for inconsistent, incorrect or duplicate information include station nearness, measurement similarity and flagging and ignoring station and measurement information.

Geographical region

To import stations and measurements within a rectangular geographical region, pass the option `--bounding-box` to **import** followed by a comma delimited string in the form of latitude and longitude pairs defining the upper left and lower right corners of the bounding box. For example, considering the `skye` network shown in Figure 1.4 on page 10, the command:

```
import -n skye skye.stn skye.msr --bounding-box -38.0630,145.1130,-38.07,145.12
```

will import stations 302513640, 302513650, 302509800 and 302502400, and all connected GNSS baselines wholly within the bounding box. The stations and measurements that are imported using this command are shown in Figure 3.2.

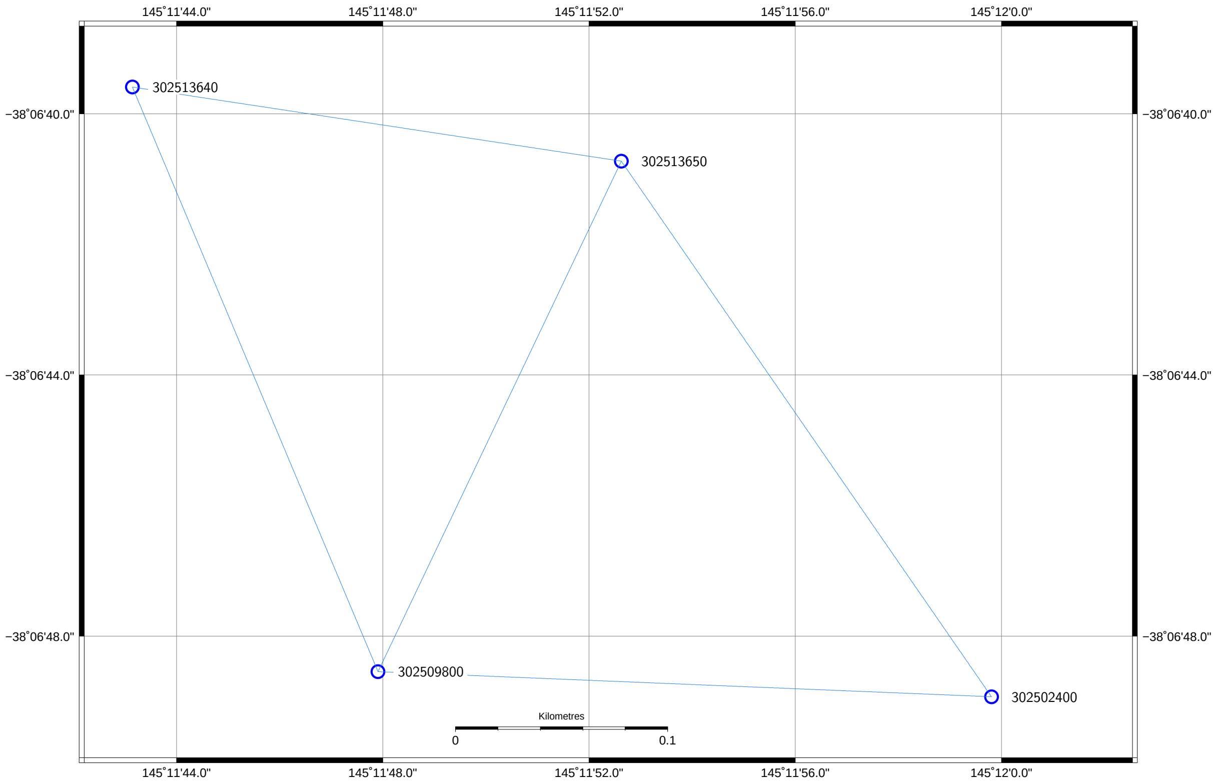


Figure 3.2: Stations and measurements in the `skye` network filtered by the `--bounding-box` option.

Note that only the stations and measurements which lie wholly within the limits of the bounding box will be imported. To include measurements which transcend the bounding box, add the option `--get-msrs-transcending-box`. By default, GNSS baseline clusters and GNSS point clusters which transcend a bounding box will not be split and as a consequence, will be ignored from the import process. If option `--bounding-box` is intended to be used to obtain a subset of a GNSS baseline or point cluster, add the option `--split-gnss-cluster-msrs`.

Including or excluding stations and measurements

To import only a set of specific stations and associated measurements, pass to **import** the option `--include-stns-assoc-msrs` followed by a comma delimited string defining the stations to be imported. This option will also import any stations connected to the associated measurements. For example, the command:

```
import -n skye skye.stn skye.msr --include-stns-assoc-msrs 302502400
```

will import station 302502400, all measurements associated with 302502400, and stations 302513650 and 302509800 which happen to be connected to the associated measurements. Any measurements between stations 302513650 and 302509800 will not be imported. For these measurements to be included, add 302513650 or 302509800 to the argument string. As with the bounding box function, if `--include-stns-assoc-msrs` is intended to be used to obtain a subset of a GNSS point or baseline cluster, add the option `--split-gnss-cluster-msrs`.

Conversely, to import all stations except a set of specific stations, pass to **import** the option `--exclude-stns-assoc-msrs` followed by a comma delimited string defining the stations to be excluded from the import process. This option will also exclude all measurements associated with that station, but not other stations. For example, the command:

```
import -n skye skye.stn skye.msr --exclude-stns-assoc-msrs 302513650
```

will import all stations except station `302513650` and all measurements associated with it. If the intention is to exclude one or more stations from a GNSS point or baseline cluster, add the option `--split-gnss-cluster-msrs`.

Segmented network block

DynAdjust provides a capability to segment a network into a series of connected blocks via **segment** (see Chapter 6). The purpose of this functionality is support phased adjustment. Once a network has been segmented, stations and measurements within a segmented block can be imported using the option `--import-block-stn-msr` and the desired block number as an argument:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --import-block-stn-msr 3
```

Using this command sequence, only the stations and measurements in block 3 will be imported. For this command sequence to work as intended, a segmentation file named `uni_sqr.seg` must exist in the folder from which **import** was executed, and this file must contain at least three blocks (see Chapter 6 for more information on the creation of segmentation files). To use an alternative segmentation file, pass to **import** the option `--seg-file` followed by the name of the segmentation file. If the segmentation file is located in another directory, the file name must include the full folder path.

Measurement type

For networks of mixed measurement types, options `--include-msr-types` and `--exclude-msr-types` can be used to include or exclude certain measurements from the import process. Both options take for their argument a non–delimited string defining the measurement types to be included or excluded. For example, the following command will only import GNSS measurements:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --include-msr GXY
```

Conversely, the following command will import all measurements except astronomic latitude and longitude:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --exclude-msr IJ
```

An additional option under the category of measurement type is `--prefer-single-x-as-g`. This option will convert all GNSS baseline cluster (X) measurements containing a single baseline into GNSS baseline (G) measurements.

Station renaming

On occasions, it will be necessary to rename multiple stations throughout a network, such as to correct stations which have been wrongly named in the field or during data processing, or to change from a well-known alias to another preferred name (or alias). To rename one or more stations upon importing stations and measurements, pass the option `--stn-renaming-file` to **import** followed by the name of the station renaming file:

```
import -n skye skye.stn skye.msr --stn-renaming-file skye.renaming
```

The structure and format of the station renaming file is shown in Figure 3.3. The first column is the preferred name to which all occurrences of the names listed in columns 1, 2, . . . , n will be changed. The number of characters reserved for each station name field will be defined according to the DNA STN file format version. For DNA version 3.0 onward, this width is 20. Rows beginning with an asterisk are treated as comments and are ignored.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```
```
!#=DNA 3.01 REN
* Station renaming file
* List the preferred name first, and then all other aliases a mark may have
* NEWNAME (20 chrs) ALIAS (20 chars)    ALIAS (20 chars)    ALIAS (20 chars)    ALIAS (20 chars)
302513640           1364
302509800           980
302508300           830
261907650           765
```

Figure 3.3: Station renaming file

Station discontinuities

To a greater or lesser extent, survey control marks and GNSS CORS sites will experience some form of physical movement over time, either suddenly or gradually. The causes of such movements are varied, and will typically result from natural and/or anthropogenic influences[2]. Left ignored, an adjustment of measurements taken before and after significant movement events may result in an unacceptable level of disagreement and may even cause the adjustment to fail. In addition to physical site movement, subtle variations in the computed position of a mark or GNSS CORS site may arise from a change in the observing system, such as changes in GNSS equipment (e.g. a new GNSS receiver, GNSS antenna or firmware), or a change of analysis software. In this context, the combination of daily coordinate estimates derived from routine GNSS CORS analysis before and after site movements or noticeable system changes may result in significant failures if the coordinate sets vary beyond what would normally be considered as random variations of the same position.

Traditionally, when the introduction of new measurements to an adjustment indicates that a survey control mark or GNSS site has moved, common practice has been to either ignore the troublesome measurements or to increase the uncertainty of older measurements such that the level of disagreement is no longer regarded a failure. Whilst the latter may alleviate some short term inconvenience, the net result is a less-than–optimal estimate of the survey control mark's true position. Provided that the spatial behaviour of the movement of marks over time can be quantified, the most rigorous approach is to account for that movement using a comprehensive deformation model. This is where

2. Natural influences are varied and can include earthquakes, ocean loading, ongoing subsidence and seasonal (cyclic) soil movement. Anthropogenic influences can include tunnelling, boring, underground resource (e.g. mineral, oil, gas, water) extraction, open–cut mining and construction.

the difficulty arises in handling mark movement — deformation models are not readily available at the resolution or precision required to account for site–specific movement.

In the absence of a comprehensive deformation model, an acceptable means for managing the presence of site movements within an adjustment without compromising the rigour of the solution is a station discontinuity file. As the title suggests, a station discontinuity represents a point in time when there was recognisable ground movement or a significant change in a series of coordinate estimates. A station discontinuity is simply recorded as an epoch against any mark or site at which time the movement or change took place. Figure 3.4 shows an example of a discontnuity appearing in time series data, caused by an external influence on a GNSS CORS site on 10 June 2004. Left ignored, the influence of the step will most likely cause measurement failures if it exceeds the expected repeatability in the estimates of position.

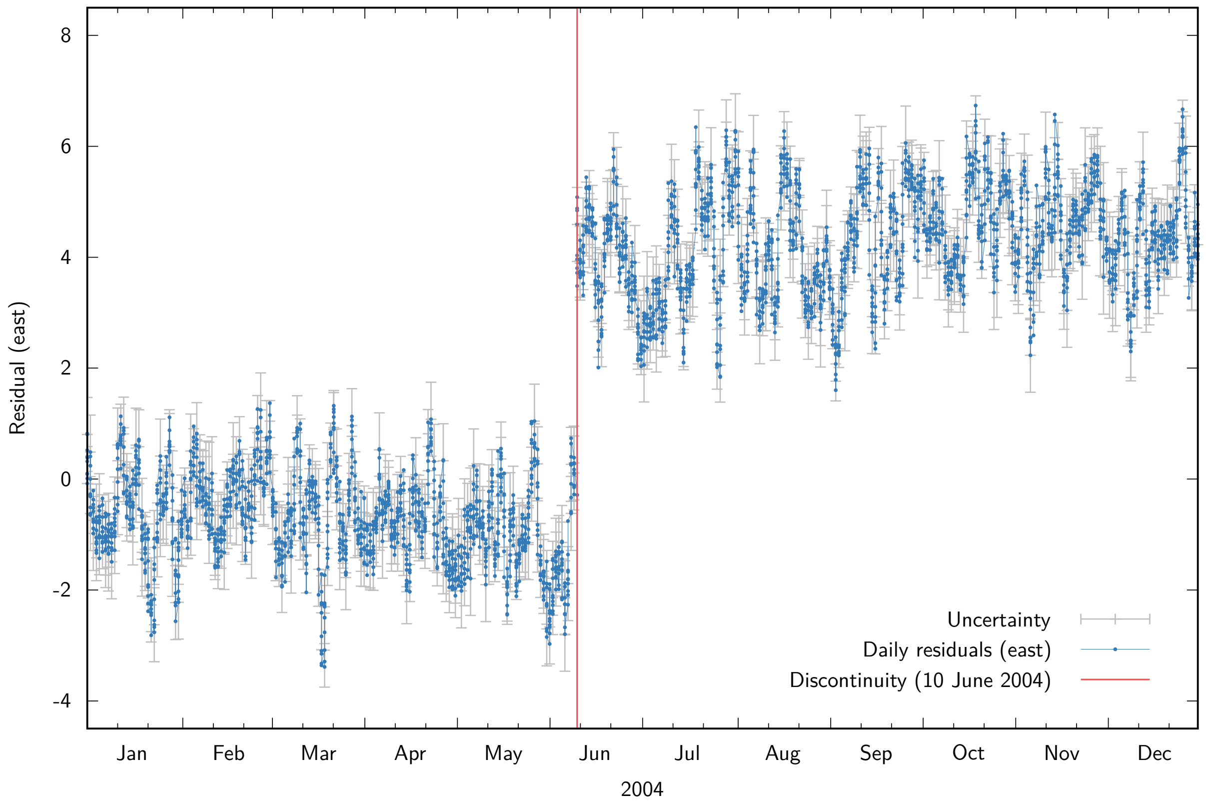


Figure 3.4: Daily GNSS residuals (east) showing evidence of a discontinuity in the time series analysis.

To introduce discontinuity information into an adjustment, pass the option `--discontinuity-file` to **import** followed by the name of the discontinuity file:

```
import -n apref_16947 apref_16947.SNX --discontinuity-file discontinuities.snx
```

Loading a discontinuity file at the time of importing station and measurment files will cause DynAdjust to make the following changes:

1. For the stations which appear in the discontinuity file, a new station is introduced for each discontinuity. The new stations are named according to the date of discontinuity. For example, the GNSS CORS site `ALIC` has two discontinuity events recorded over the lifetime of its

continuous operation — day 315 of year 2010 and day 201 of year 2011. The two stations introduced are named `ALIC_2010315` and `ALIC_2011201` respectively.

2. The original station will be renamed to a date corresponding to the earliest date it was observed. As this date will be ascertained from the measurement file, it is important that the measurements have accurate epoch information.
3. For each measurement connected to a station that appears in the discontinuity file, the 'from' and 'to' stations (or instrument and target stations) will be renamed according to the date the measurement was observed (e.g. Epoch in Table B.8 on page 189). The station name adopted will always correspond to the discontinuity immediately preceeding the date of observation. For example, if a GNSS baseline measurement to `ALIC` was observed on day 200 of year 2011, `ALIC` will be renamed to `ALIC_2010315`.

The net result of these changes from a least squares adjustment will be multiple coordinate estimates for a station according to the respective discontinuity dates.

Station nearness

When concatenating networks contained in different files, the same physical station can sometimes appear in the station and measurement files with different station names. To assist with identifying stations with different names, pass the option `--search-nearby-stn` to **import** and, if required, pass the option `--nearby-stn-buffer` followed by a distance in metres by which to search for nearby stations. For example, to search for stations in `uni_sqrstn.xml` within 1.5 metres of each other, run the following command:

```
import -n uni_sqr uni_sqrstn.xml --search-nearby-stn --nearby-stn-buffer 1.5
```

This command will produce the following output:

```
+ Testing for duplicate and nearby stations... Done.
- Error:   1 station was found to be separated by less than 1.5m.
  See ./unisqr.dst for details.

  If the names in each pair refer to the same station, then update the
  station and measurement files with the correct station name and re-run import.
  Alternatively, if the names in each pair are unique, either call import
  without the --search-nearby-stn option, or decrease the radial search
  distance using --nearby-stn-buffer.
```

Having identified one or more nearby stations, **import** will create a file called `uni_sqr.dst` in which will be printed a list of stations that are separated by less than 1.5 metres. The contents of this file are as follows:

```
--------------------------------------------------------------------------------
DUPLICATE STATION FILE

Version                            3.1.2
Build                              May 26 2014, 22:27:14 (MSVC++ 10.0)
File created                       Monday, 26 May 2014, 10:30:35 PM
File name                          C:\Data\unisqr.dst

Command line arguments             import -n unisqr uni_sqrstn.xml --search-nearby-stn --nearby-stn-buf 1.5

Network name                       unisqr
Stations file                      C:\Data\unisqr.bst
Measurements file                  C:\Data\unisqr.bms
Associated station file            C:\Data\unisqr.asl
Associated measurement file        C:\Data\unisqr.aml
Duplicate stations output file     C:\Data\unisqr.dst
Similar measurement output file    C:\Data\unisqr.dms
Input files                        uni_sqrstn.xml
Default reference frame            GDA94
Search for nearby stations         tolerance = 1.5m
--------------------------------------------------------------------------------

Nearby station search results:
1 station was found to be separated by less than 1.5m:

  First station       Nearby station      Separation (m)
  2120                NEWP                0.531
```

Once **import** has identified the stations that satisfy the search criteria, it is up to the user to verify if the stations listed in the `.dst` file are the same physical station or not. If a single station has been inadvertently given different names in the station and measurement files, redundant station information can be harmonised by:

1. Amalgamate the station information record(s) to form a single station with the preferred station name, coordinates and constraints.
2. Search and replace in the measurement file all instances of the redundant station name with the preferred station name

It is anticipated that a future version of DynAdjust will provide an automated approach for harmonising redundant station information.

Measurement similarity

As in the case for stations, concatenating networks contained in different files can easily result in duplicate measurements in the measurement files. Duplicate measurements can be difficult to detect as it is normal and legitimate for redundant measurements to exist. To assist with identifying duplicate measurements, pass the option `--search-similar-msr` to **import**:

```
import -n uni_sqr uni_sqrmsr.xml --search-similar-msr
```

This sequence of commands will produce the following output:

```
+ Searching for similar measurements...
- Error:   3 measurements were found to be very similar (if not identical)
  to other measurements.  See ./uni_sqr.dms for details.

  If the listed measurements are true duplicates, either remove each duplicate
  from the measurement file and re-run import, or re-run import with the
  --ignore-similar-msr option.  Alternatively, if each measurement
  is unique, then call import without the --search-similar-msr option.
```

As shown in the above output, **import** will create a file called `unisqr.dms` in which will be printed a list of measurements that appear to be similar or identical to other measurements. The format of the similar measurements is DynaML as follows:

```
- 3 measurements were found to be very similar (if not identical)
  to other measurements.

  <!--Type L Level difference-->
  <DnaMeasurement>
    <Type>L</Type>
    <Ignore/>
    <First>2214</First>
    <Second>2213</Second>
    <Value>0.0010</Value>
    <StdDev>0.0020</StdDev>
  </DnaMeasurement>
  <!--Type S Slope distance-->
  <DnaMeasurement>
    <Type>S</Type>
    <Ignore/>
    <First>2205</First>
    <Second>2213</Second>
    <Value>13.3010</Value>
    <StdDev>0.0050</StdDev>
    <InstHeight>0.000</InstHeight>
    <TargHeight>0.000</TargHeight>
  </DnaMeasurement>
  <!--Type S Slope distance-->
  <DnaMeasurement>
    <Type>S</Type>
    <Ignore/>
    <First>6002</First>
    <Second>1049</Second>
    <Value>136.4290</Value>
    <StdDev>0.0100</StdDev>
    <InstHeight>1.601</InstHeight>
    <TargHeight>1.337</TargHeight>
  </DnaMeasurement>
```

As with stations, the second and subsequent appearances of a measurement with similar attributes will be regarded as a duplicate. Once **import** has identified the measurements that satisfy the test for similarity, it is up to the user to verify if the measurements listed in the `.dms` file are the same measurement or different measurements. Measurements which are duplicates can either be removed from the measurement file, or set as an ignored measurement. For example, measurements can be ignored by providing an asterisk (*) for the ignore flag in DNA files (see Table B.3, §B.1.4) and DynaML files (see §B.3.2).

For very large networks which have been derived from hundreds of GNSS survey projects, it is possible for a single GNSS data set to be inadvertently processed twice, resulting in unique but *similar* GNSS baselines and GNSS baseline clusters. At times, these similar measurements can be difficult to detect due to slight variations in the generated GNSS vector and variance matrix components. To search for single GNSS baselines which may be similar to a GNSS baseline cluster, pass the option `--search-similar-gnss-msr` to **import**. This option will cause **import** to identify every single GNSS baseline where both terminal stations are contained within a GNSS baseline cluster and where both measurements the same reference frame and epoch.

Ignoring similar measurements

Upon verifying whether the measurements listed in the `.dms` file are duplicate measurements, the user may opt to remove the duplicate measurements or flag them as ignored for further verification. To automatically ignore all measurements identified as similar. add `--ignore-similar-msr` to the list of options for `import`:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --ignore-similar-msr
```

Executing this command will automatically flag any measurements deemed to be similar as other measurements in the network as ignored and thereby exclude them from all subsequent processing.

Removing ignored measurements

Measurements which are flagged as ignored, whether in the measurement file or by supplying the `--ignore-similar-msr` option, can be automatically removed from all subsequent processing as follows:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --remove-ignored-msr
```

This option can be combined with option `--ignore-similar-msr` in a single command:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --ignore-similst-msr --remove-ignored-msr
```

Executing this command will automatically flag similar measurements and, together with any other measurements already marked as ignored, will not be written to the binary measurement file.

Flagging unused stations

Upon loading the station and measurement information from the input files, any stations not connected to any measurements will be noted as unused and thereby marked for exclusion from segmentation and adjustment processes. However, **import** will retain these stations within the binary station file. Accordingly, functions such as data export (see §3.6) will include these stations.

To prevent unused stations from being used in any further processing, pass to **import** the option `--flag-unused-stations`:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --flag-unused-stations
```

3.3.3 GNSS variance matrix scaling

For the most part, GNSS measurement and variance matrix information will be sourced from files generated by a GNSS processing package. Depending on the processing strategy, it is possible that the variance matrices accompanying the GNSS measurements will need some form of scaling such that a more realistic estimate of the GNSS measurement precision is incorporated within an adjustment. GNSS measurement variance matrices may be scaled using the appropriate scalar fields within the input files (e.g. Table B.8 on page 189 for DNA file format, or Figure B.23 on page 201 for DynaML file format). In addition, **import** provides a capacity to scale GNSS variance matrices via five different scaling options. Table 3.3 lists the lists the various GNSS variance matrix scaling options (c.f. Table A.6 on page 167 in the Command line reference).

Table 3.3: GNSS variance matrix scaling options

Option	All msrs	Clusters	Matrix	North–south	East–west	Up
`--v-scale`	•	•	•			
`--p-scale`	•	•		•		
`--l-scale`	•	•			•	
`--h-scale`	•	•				•
`--baseline-scalar-file`			•	•	•	•

As shown in Table 3.3, GNSS variance matrix scaling can be applied to all GNSS measurements contained in all measurement files passed to **import**, or to individual GNSS baselines. Individual baseline scaling is based upon the contents of a baseline scalar file and does not apply to GNSS point or baseline clusters. Whether scaling all or individual GNSS variance matrices, scaling can be applied in two different ways — via a single matrix multiplier (or variance factor) or via individual scalars corresponding to the three dimensions of the local reference frame. One or more scaling options in any combination can be passed to **import**.

To scale all GNSS variance matrices by a single multiplier, pass to **import** the option `--v-scale` followed by a scalar value. For example, to scale all variance matrices in the `skye` project by a factor of `10.0`, run the following command:

```
import -n skye skye.stn skye.msr --v-scale 10
```

To scale all GNSS variance matrices by partial scalars in the north–south, east–west or vertical directions, pass to **import** the options `--p-scale`, `--l-scale` or `--h-scale` respectively, followed by the relevant scalar value. For example, to scale all variance matrices in the `skye` project by a factor of `2.0` in the horizontal direction and `5.0` in the vertical direction, run the following command:

```
import -n skye skye.stn skye.msr --p-scale 2 --l-scale 2 --h-scale 5
```

To scale specific GNSS baseline variance matrices, pass to **import** the option `--v-scale` followed by the name of the baseline scalar file:

```
import -n skye skye.stn skye.msr --baseline-scalar-file skye.scalars
```

The structure and format of the baseline scalar file is shown in Figure 3.5. 20 characters are reserved for the two station names, and 10 characters are reserved for `v-scale`, `p-scale`, `l-scale` and `h-scale`.

For all scaling options, §7.3.2.1 describes the procedure implemented by DynAdjust for scaling GNSS variance matrices.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------
GNSS BASELINE VARIANCE MATRIX SCALAR FILE

SCALARS

Station 1           Station 2                v-scale   p-scale   l-scale   h-scale
--------------------------------------------------------------------------------
409601230           409601240                  100.0         1         1         1
409601230           409601250                     10         2         2         5
409601230           409601260                      1         3         3        10
```

Figure 3.5: Baseline scalar file

3.4 Network measurement simulation

import provides a capability to simulate a network of GNSS and terrestrial measurements through the `--simulate-msr-file` option. In order to simulate measurements, two files are required — a station file and a measurement simulation control file. Upon execution, measurements are simulated for all measurement types and station names listed in the measurement simulation control file. Measurement values are calculated using the coordinates provided in the station file. The following command demonstrates the use of **import** to simulate measurements:

```
import -n simnet simnet.stn simnet-list.msr --simulate
```

Running this command will create a DNA measurements file named `simnet.simulated.msr` which will contain measurements generated from the station coordinates in `simnet.stn` and the types and station names in `simnet-list.msr`. The input station file `simnet.stn` must conform to the DNA station file format (c.f. §B.1.3). The input measurement simulation control file `simnet-list.msr` must conform to the DNA measurement file format (c.f. §B.1.4) in principle, the exceptions being that only measurement types, station names and cluster counts need to be present. Figure 3.6 shows a sample simulation control file.

The measurement values in the simulated measurements file are calculated exactly from the station coordinates. All other information is generated as follows:

- Standard deviations for measurement types C, E, L, M and S will be calculated from

$$3.0 \times \sqrt{(L/1000.0)}/100.0$$

  where L is the calculated distance in metres;
- Standard deviations for measurement types H and R will be set to 0.024 metres.
- Variance matrices for G, X and Y measurement types will be set to:

$$\begin{bmatrix} 4.022000\text{E-}05 & -1.369000\text{E-}05 & 3.975000\text{E-}05 \\ -1.369000\text{E-}05 & 1.487000\text{E-}05 & -2.035000\text{E-}05 \\ 3.975000\text{E-}05 & -2.035000\text{E-}05 & 6.803000\text{E-}05 \end{bmatrix}$$

- Station and target heights for measurements which require these heights (e.g. S, V, Z) will be set to 1.650 and 1.651 metres respectively.
- Standard deviations for measurement types A, B, D, K, V and Z will be set to 0.01 seconds.
- Standard deviations for measurement types I, J, P and Q will be set to 0.021 seconds.

```
123456789012345678901234567890123456789012345678901234567890123456789
```

```
!#=DNA 3.00 MSR    16.06.2014         GDA94    01.01.1994        20
* Measurement simulation control file
A 503                 304                 307
B 909                 1010
C 501                 309
D 602                 302                   2
                                          501
                                          503
E 503                 307
G 604                 606
H 105
K 101                 202
L 910                 1010
M 604                 706
P 105
Q 105
R 106
S 602                 302
V 109                 207
X 406                 501                 2
  109                 501
Y 406                 XYZ                 3
  203
  509
Z 503                 307
```

Figure 3.6: Example simulation measurements list

If orthometric heights are supplied in the station file, or if simulated terrestrial measurements are intended to contain a realistic measure of the influence of the gravity field, geoid–ellipsoid separations and deflections of the vertical can be added to the measurements through the use of a DNA geoid file. To this end, add the `--geo-file` option (c.f. §3.2.4), followed by the geoid file name:

```
import -n simnet simnet.stn simnet-list.msr --simulate --geo-file simnet.geo
```

If a DNA geoid file has not been created, **geoid** may be called to interpolate and export the geoid–ellipsoid separations and deflections of the vertical to a DNA geoid file. For this purpose, a binary station file for `simnet` is required. The steps are as follows:

1. Import the simulation station file to create the binary station file `simnet.bst`.
2. Interpolate the geoid information (from `simnet.bst`) and export to a DNA geoid file.
3. Simulate using the station file, simulation control file and the newly created geoid file.

The command line sequence for these steps is as follows:

```
import -n simnet simnet.stn
geoid simnet --ntv2-file ausgeoid09.gsb --export-dna-geo
import -n simnet simnet.stn simnet-list.msr --simulate --geo-file simnet.geo
```

See Chapter 5 for more help on interpolating geoid information from a geoid grid file.

3.5 Output station to measurement connectivity

To output a summary of the measurements connected to each station, add `--output-msr-to-stn` to the command line arguments for **import**:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml --output-msr-to-stn
```

This command will produce a file (`--uni_sqr.m2s`) containing a table with the frequency of each measurement type and the total number of measurements associated with each station. The table will be sorted row–wise by the station name, and column–wise by the measurement type (c.f. Table 3.2). For details on the content, structure and format of the table, refer to Table C.1 in Appendix C.3. Generating this table can be quite useful when attempting to identify measurement conflicts, or the presence of unnecessary measurements, measurement constraints or station constraints.

3.6 Data export

In addition to translating the external file formats into the binary formats required by DynAdjust, **import** provides a capacity to export geodetic network information into a number of different file formats. The options for exporting information include:

- Export of station and measurement information to DNA, DynaML, GeodesyML and SINEX file formats
- Export of geodetic network binary files (MAP, AML, ASL) to equivalent plain text variants

Export options can be passed to **import** together with any other option. Hence, the residual station and measurement information generated through the various configuration options can be stored for subsequent import, filtering and processing.

3.6.1 Station and measurement files

To export station and measurement information into DNA format, pass to **import** the option `--export-dna-files`. Similarly, to export station and measurement information into DynaML format, pass the option `--export-xml-files`. For example, the following command will export DNA station and measurement files comprised of only GNSS measurements found within the `uni_sqr` project, with the default reference frame set to `ITRF2005` and all GNSS variance matrices scaled by `10.0`:

```
import -n us_bsl uni_sqrstn.xml uni_sqrmsr.xml --v-sc 10 --ref itrf2005 --include-m GXY --export-dna --flag
```

Running this command will produce two files named `us_bsl.stn` and `us_bsl.msr` corresponding to the station and measurement files respectively. Since `--flag-unused-stations` was passed to **import**, stations which become disconnected from removed measurements (as a result of importing only GNSS measurements) will not be written to the station file.

The option `--export-xml-files` will produce two XML files named <network_name>stn.xml and <network_name>msr.xml corresponding to the station and measurement files respectively. To produce a single DynaML file, pass to **import** the option `--single-xml-file` with `--export-xml-files`.

Note that if the network name supplied in the above command matched the name of the input files according to the file naming convention described in §2.2.1 (i.e. `uni_sqr`), and data is exported to the same file format as the input files, then the original input files will be overwritten.

3.6.2 Association lists and station map

As discussed in §*2.2.2*, DynAdjust reads and writes a number of binary files. The Station Map (MAP), Associated Stations List (ASL) and Associated Measurements List (AML) may be exported to plain text to assist with detailed analysis of the network's geometry, measurement connectivity and measurement frequency. To export the ASL, AML and MAP to their plain text counterparts, run the following command:

```
import -n us_bsl us_bsl.stn us_bsl.msr --export-map --export-asl --export-aml
```

This command will produce three files named `us_bsl.map.txt`, `us_bsl.asl.txt` and `us_bsl.aml.txt`. The fields contained in the exported files are described in Tables 3.4, 3.5 and 3.6.

Table 3.4: Station Map fields

Field	Description
Station name	The station name as given in the input station and measurement files. The column header contains the number of stations detected in the input files.
Station index	The (zero–based) index of the station name in an alpha–numeric sorted list of all station names. DynAdjust uses this index when referring to stations during all internal processing.

Table 3.5: Associated Stations List fields

Field	Description
Station name	The station name as given in the input station and measurement files. The column header contains the number of stations detected in the input files.
No. connected msrs	The number of measurements connected to this station.
AML index	The (zero–based) index of the first record of this station's appearance in the Associated Measurements List.
Unused?	If an asterisk is present, this station is not connected to any measurements, or is only connected to ignored measurements.

Table 3.6: Associated Measurements List fields

Field	Description
Station name	The station name as given in the input station and measurement files. The column header contains the number of records in the file.
Msr index	The (zero–based) index of this measurement in the binary measurements file.
Msr type	The type of measurement (c.f. Table 3.2) and which measurement station this station refers to (first, second or third)
Cluster	If this measurement is part of a cluster, this number represents the index of the cluster in a list of all detected clusters.
Ignored msr?	If an asterisk is present, this measurement has been marked as ignored.

Chapter 4

Transformation of coordinates and measurements

4.1 Introduction

Recognising that GNSS surveying is the most highly utilised measurement technique for geodetic network establishment and maintenance, DynAdjust has been designed around the use of pre–processed GNSS measurements for the estimation of network station parameters. DynAdjust has also been designed to integrate measurements taken in the local reference frame with GNSS measurements in an earth–centred, cartesian reference frame.

But since the reference frame of GNSS measurements is directly related to the reference frame of the satellite orbit information used during GNSS processing, it is inevitable that GNSS measurements taken over several successive years will be based upon different reference frames and epochs. This is primarily due to the fact that since the launch of the first GNSS satellites, several reference frames (and versions) have been adopted for broadcast and precise ephemeris. The reference frames that may be relevant in this context can include:

- The PZ-90 coordinate system. Used for the Russian GLObal NAvigation Satellite System (GLONASS).
- The International Terrestrial Reference Frame (ITRF). Adopted by the International GNSS Service (IGS) for their precise orbit products.
- The World Geodetic System of 1994 (WGS84). Used for the Global Positioning System (GPS). Since its establishment, WGS84 has been aligned (and re–aligned) to several versions of ITRF.

In all cases, these frames can undergo several iterations (or revisions), such as ITRF88, ITRF92, ITRF2008, etc. WGS84 is regularly updated and aligned to ITRF. It follows that each frame realisation will have a different reference epoch. Accordingly, there is a fundamental need to handle transformations of GNSS measurements between these reference frames if the combination and adjustment of multiple GNSS measurements in a single solution is to be free from *reference frame bias*. To this end, **reftran** has been developed to transform stations and GNSS measurements between a selection of commonly used reference frames.

The first part of this chapter (§4.2 – §4.5) presents the theory of reference frames and coordinate systems, conversions between coordinate systems, transformation of coordinates and measurements between static and dynamic reference frames, and propagation of variances. The remainder of the chapter describes the use of **reftran** and its options.

4.2 Reference frame fundamentals

4.2.1 Cartesian reference frame

The cartesian reference frame is a three–dimensional system, the origin o of which is coincident with the centre of the reference ellipsoid. The referencing of point p is by means of orthogonal displacements parallel to the x, y and z axes. Figure 4.1 illustrates point p in relation to the axes of the cartesian reference frame, and the relationship between cartesian coordinates and geodetic latitude ϕ, geodetic longitude λ and ellipsoid height h.

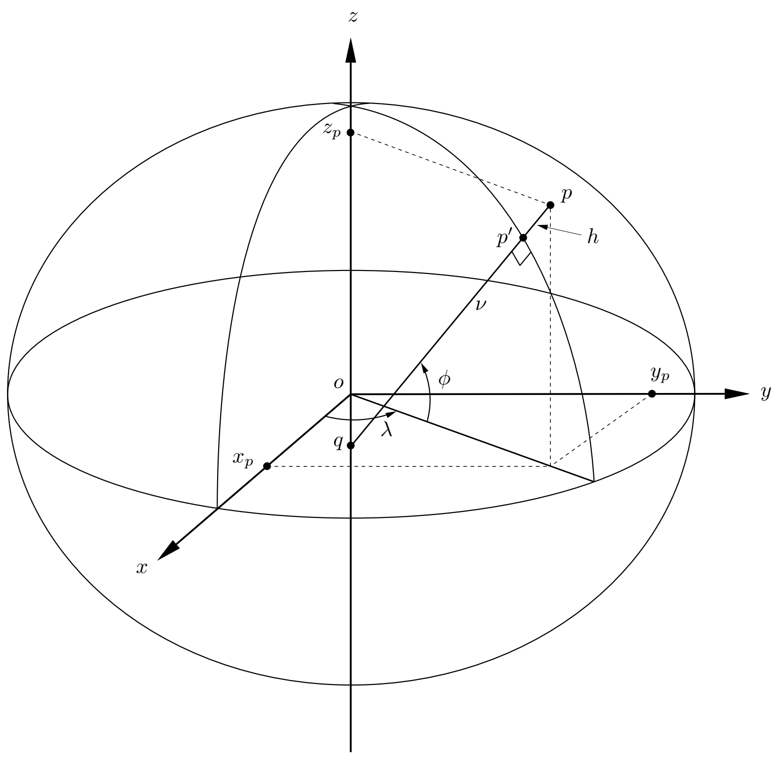


Figure 4.1: The ellipsoid–centred cartesian reference frame

A vector $\mathbf{x}_c$ between two points in the cartesian reference frame will have the elements:

$$\mathbf{x}_c = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{4.1}$$

As shown by Figure 4.1, geodetic latitude is the angle between the equatorial (x–y) plane and the normal to the ellipsoid through p, and geodetic longitude is the angle between the zero meridian and the meridian through p. Ellipsoid height is the distance of p above p', which is a point projected onto the reference ellipsoid along the ellipsoid normal. As Figure 4.1 indicates, the coordinates defining the location of p are dependent upon the size, shape and location of the reference ellipsoid.

4.2.2 Geographic reference frame

With respect to Figure 4.1, geographic reference frame (or geodetic) coordinates ϕ, λ and h are related to the corresponding cartesian coordinates by the following expressions:

$$
\begin{aligned}
x &= (\nu + h)\cos\phi\cos\lambda \\
y &= (\nu + h)\cos\phi\sin\lambda \\
z &= \left[\nu(1-e^2) + h\right]\sin\phi
\end{aligned}
\tag{4.2}
$$

where ν is the radius of curvature in the prime vertical, given by:

$$
\nu = \frac{a}{\sqrt{(1 - e^2\sin^2\phi)}} \tag{4.3}
$$

a is the quantity representing the semi–major axis of the reference ellipsoid and e^2 is the *first eccentricity* obtained from:

$$
e^2 = \frac{(a^2 - b^2)}{a^2} \tag{4.4}
$$

and b is the semi–minor axis of the ellipsoid (coincident with the z axis).

As will be required for later developments, the radius of curvature in the prime meridian ρ at ϕ is given by:

$$
\rho = \frac{a\left(1 - e^2\right)}{\left(1 - e^2\sin^2\phi\right)^{\frac{3}{2}}} \tag{4.5}
$$

and the distance oq which provides for the z coordinate of the point q — being at the intersection of the normal at p with the polar axis of the ellipsoid — is given by:

$$
oq = e^2\nu\sin\phi \tag{4.6}
$$

4.2.3 Local reference frame

The local reference frame is a three–dimensional orthogonal system that has its origin on or above any point on the ellipsoid, and is oriented with respect to the local geodetic meridian. Figure 4.2 illustrates the axes of the local reference frame in relation to the cartesian reference frame. The vector displacement up is coincident with the ellipsoid normal. The orientation of the local reference frame varies with respect to latitude ϕ and longitude λ.

Vectors in the cartesian reference frame can be represented in the local reference frame as $\mathbf{x}_l$:

$$
\mathbf{x}_l = \begin{bmatrix} \Delta e \\ \Delta n \\ \Delta up \end{bmatrix} \tag{4.7}
$$

A vector $\mathbf{x}_l$ in the local reference frame is related to the cartesian reference frame by:

$$
\mathbf{x}_c = \mathbf{R}_l\mathbf{x}_l \tag{4.8}
$$

where $\mathbf{R}_l$ is the rotation matrix with origin at latitude ϕ and longitude λ:

$$
\mathbf{R}_l = \begin{bmatrix} -\sin\lambda & -\sin\phi\cos\lambda & \cos\phi\cos\lambda \\ \cos\lambda & -\sin\phi\sin\lambda & \cos\phi\sin\lambda \\ 0 & \cos\phi & \sin\phi \end{bmatrix} \tag{4.9}
$$

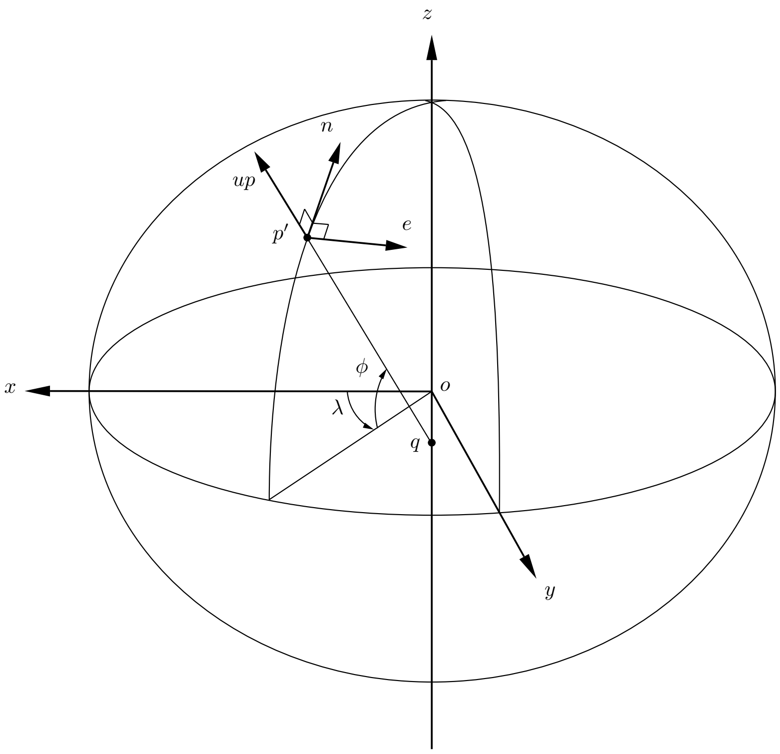


Figure 4.2: The local reference frame

Equation 4.8 can be written in expanded form as:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\sin\lambda(\Delta e) - \sin\phi\cos\lambda(\Delta n) + \cos\phi\cos\lambda(\Delta up) \\ \cos\lambda(\Delta e) - \sin\phi\sin\lambda(\Delta n) + \cos\phi\sin\lambda(\Delta up) \\ \cos\phi(\Delta n) + \sin\phi(\Delta up) \end{bmatrix} \tag{4.10}$$

Since $\mathbf{R}_l$ is orthogonal[1], a vector $\mathbf{x}_c$ in the cartesian reference frame is related to the local reference frame by:

$$\mathbf{x}_l = \mathbf{R}_l^T \mathbf{x}_c \tag{4.11}$$

which can be written in expanded form as:

$$\begin{bmatrix} \Delta e \\ \Delta n \\ \Delta up \end{bmatrix} = \begin{bmatrix} -\sin\lambda(\Delta x) + \cos\lambda(\Delta y) \\ -\sin\phi\cos\lambda(\Delta x) - \sin\phi\sin\lambda(\Delta y) + \cos\phi(\Delta z) \\ \cos\phi\cos\lambda(\Delta x) + \cos\phi\sin\lambda(\Delta y) + \sin\phi(\Delta z) \end{bmatrix} \tag{4.12}$$

4.2.4 Polar reference frame

At times it is necessary to transform vectors in a local reference frame into a displaced orthogonal polar reference frame. Figure 4.3 illustrates the displacement of a local reference frame at point p_2 in relation to the polar frame at point p_1 by geodetic azimuth (θ), vertical angle (ϑ) and direct distance (s).

1. A matrix is orthogonal if the inverse of that matrix is equal to its transpose, i.e. $\mathbf{R}^T = \mathbf{R}^{-1}$.

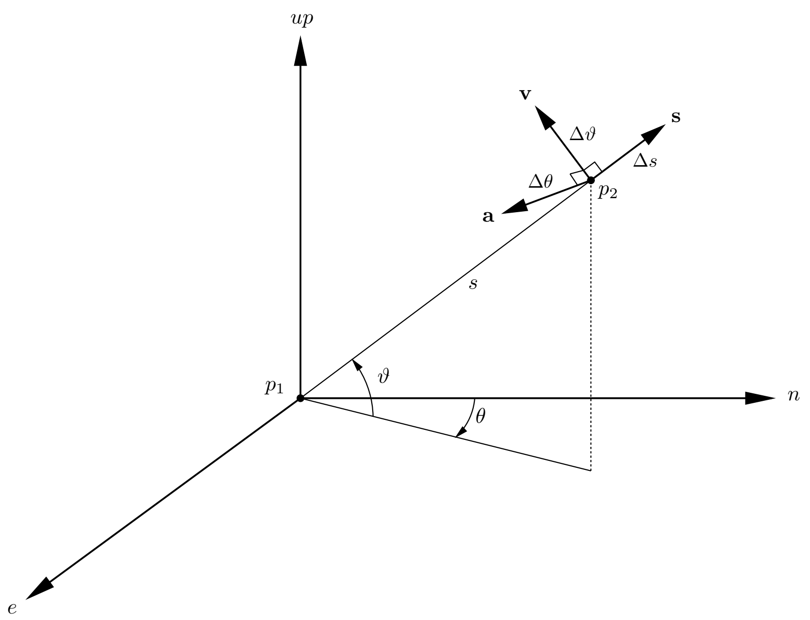


Figure 4.3: Orthogonal reference frames displaced by geodetic azimuth, vertical angle and distance

With respect to Figure 4.3, $\mathbf{a}$, $\mathbf{v}$ and $\mathbf{s}$ are unit vectors defining the displaced polar reference frame. Vectors $\mathbf{v}$ and $\mathbf{s}$ are in the vertical plane, whereas $\mathbf{a}$ is in the horizontal plane. A vector $\mathbf{x}_p$ in the polar reference frame has the elements:

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{a} \\ \mathbf{v} \\ \mathbf{s} \end{bmatrix} \tag{4.13}$$

A vector in the local reference frame is transformed to the displaced polar reference frame by:

$$\mathbf{x}_p = \mathbf{R}_p \mathbf{x}_l \tag{4.14}$$

where $\mathbf{R}_p$ is an orthogonal rotation matrix at origin p_2:

$$\mathbf{R}_p = \begin{bmatrix} -\cos\theta & -\sin\vartheta\sin\theta & \cos\vartheta\sin\theta \\ -\sin\theta & -\sin\vartheta\cos\theta & \cos\vartheta\cos\theta \\ 0 & \cos\vartheta & \sin\vartheta \end{bmatrix} \tag{4.15}$$

Polar coordinates of θ, ϑ and s are related to vectors in the local reference frame by:

$$\begin{aligned} \theta &= \arctan\left(\frac{\Delta e}{\Delta n}\right) \\ \vartheta &= \arctan\left(\frac{\Delta up}{\sqrt{\Delta e^2 + \Delta n^2}}\right) \\ s &= \sqrt{\Delta e^2 + \Delta n^2 + \Delta up^2} \end{aligned} \tag{4.16}$$

Equation 4.14 can be written in expanded form as:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{v} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} -\cos\theta(\Delta e) - \sin\vartheta\sin\theta(\Delta n) + \cos\vartheta\sin\theta(\Delta up) \\ -\sin\theta(\Delta e) - \sin\vartheta\cos\theta(\Delta n) + \cos\vartheta\cos\theta(\Delta up) \\ \cos\vartheta(\Delta n) + \sin\vartheta(\Delta up) \end{bmatrix} \tag{4.17}$$

Since $\mathbf{R}_p$ is orthogonal, a vector $\mathbf{x}_p$ in the displaced polar reference frame can be rotated into the local reference frame by:

$$\mathbf{x}_l = \mathbf{R}_p^T \mathbf{x}_p \tag{4.18}$$

which can be written in expanded form as:

$$\begin{bmatrix} \Delta e \\ \Delta n \\ \Delta up \end{bmatrix} = \begin{bmatrix} -\cos\theta(\mathbf{a}) - \sin\theta(\mathbf{v}) \\ -\sin\vartheta\sin\theta(\mathbf{a}) - \sin\vartheta\cos\theta(\mathbf{v}) + \cos\vartheta(\mathbf{s}) \\ \cos\vartheta\sin\theta(\mathbf{a}) + \cos\vartheta\cos\theta(\mathbf{v}) + \sin\vartheta(\mathbf{s}) \end{bmatrix} \tag{4.19}$$

4.3 Transformation between cartesian reference frames

Several methods are available for transforming coordinates and vectors from one cartesian reference frame to another. DynAdjust supports (conformal 3D) 7–parameter and 14–parameter similarity transformations using the Bursa–Wolf model.

Conformal transformations between two cartesian reference frames on the same epoch are managed by a 7–parameter similarity relationship. The relationship involves a three–dimensional shift, rotation and scale of the cartesian coordinate axes and is given by:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \Delta x_{12} \\ \Delta y_{12} \\ \Delta z_{12} \end{bmatrix} + (1+\delta) \begin{bmatrix} 1 & r_z & -r_y \\ -r_z & 1 & r_x \\ r_y & -r_x & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{4.20}$$

where Δx_{12}, Δy_{12} and Δz_{12} are the origin shifts, r_x, r_y and r_z are the coordinate axis rotations (in radians) and δ is the scale factor between the two systems. Where rotation of axes exceeds 10" of arc, the rotation matrix in equation 4.20 is replaced with the following:

$$\mathbf{R} = \begin{bmatrix} \cos r_z & \sin r_z & 0 \\ -\sin r_z & \cos r_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos r_y & 0 & -\sin r_y \\ 0 & 1 & 0 \\ \sin r_y & 0 & \cos r_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos r_x & \sin r_x \\ 0 & -\sin r_x & \cos r_x \end{bmatrix}$$

Conformal transformations between two cartesian reference frames on different epochs are managed by a 14–parameter similarity relationship. The relationship is identical to that given in equation 4.20, with the exception of introducing terms for the first order derivatives of origin shift, rotation and scale as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \Delta x_{12} + \dot{x}\,(t_2 - t_1) \\ \Delta y_{12} + \dot{y}\,(t_2 - t_1) \\ \Delta z_{12} + \dot{z}\,(t_2 - t_1) \end{bmatrix} + \Big(1 + \delta + \dot{\delta}\,(t_2 - t_1)\Big) \times \tag{4.21}$$
$$\begin{bmatrix} 1 & r_z + \dot{r_z}\,(t_2 - t_1) & -r_y - \dot{r_y}\,(t_2 - t_1) \\ -r_z - \dot{r_z}\,(t_2 - t_1) & 1 & r_x + \dot{r_x}\,(t_2 - t_1) \\ r_y + \dot{r_y}\,(t_2 - t_1) & -r_x - \dot{r_x}\,(t_2 - t_1) & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

Here, $[\ \dot{x} \quad \dot{y} \quad \dot{z}\ ]$, $[\ \dot{r_x} \quad \dot{r_y} \quad \dot{r_z}\ ]$ and $\dot{\delta}$ are the rates of change in origin shift, axis rotation and scale respectively, t_1 is the reference epoch of the reference frame for x_1, y_1 and z_1, and t_2 is the epoch of the reference frame for x_2, y_2 and z_2.

4.3.1 Transformation with projection of coordinates between epochs

When propagating coordinates between different epochs in a reference frame that is dynamic (e.g. ITRF2014, ITRF2008, etc.), a plate motion model that describes the temporal–spatial behaviour of positions on the earth is required. For this purpose, DynAdjust adopts Euler's concept of regarding displacements of positions on the earth's surface as a rotation with a defined angular velocity about a pole. The concept is illustrated in Figure 4.4, using the Australian tectonic plate (AU) as an example.

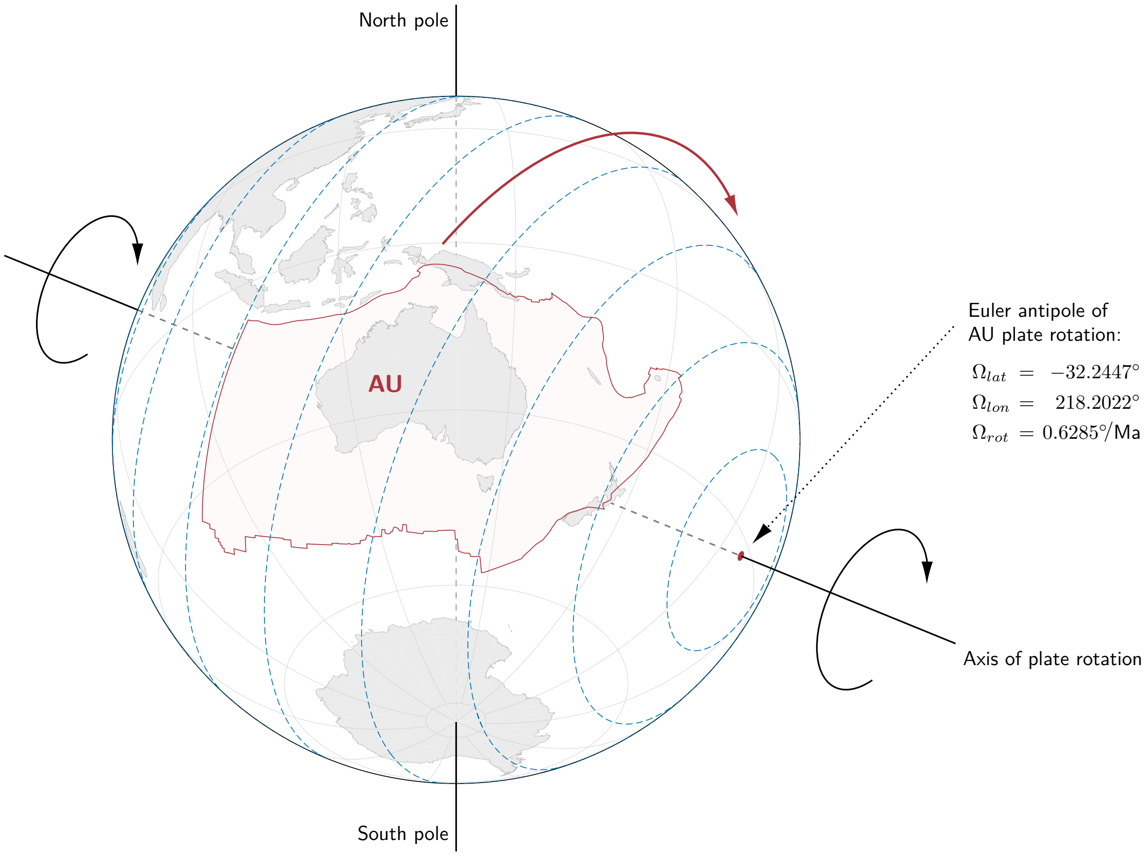


Figure 4.4: Euler plate motion of the AU plate

As illustrated by the Australian plate in Figure 4.4, the motions of all tectonic plates on the earth can be quantified in terms of angular velocity about an Euler pole. Mathematically, global plate motions are typically expressed in terms of either:

- Spherical angular velocity, given as Euler pole latitude Ω_{lat}, longitude Ω_{lon} (in decimal degrees) and a right–handed (anti–clockwise) rotation rate Ω_{rot} in degrees per million years, or;
- Earth–centred cartesian rotation rates, given as $\dot{r_x}$, $\dot{r_y}$ and $\dot{r_z}$ in milliarc seconds per year.

In either case, plate motions may be expressed relative to a single plate that is held fixed, or with respect to a "no-net-rotation" (NNR) reference frame that represents the motion of the earth's surface relative to the earth's origin in an absolute sense. The former allows for quantifying the movement of stations relative to an arbitrarily fixed plate; the latter for the movement of all stations relative to an absolute (earth–centred, earth–fixed) reference frame.

DynAdjust loads Euler pole parameters (as spherical coordinates and angular velocity) via user–supplied text files (explained in §4.6.3). In order to apply these parameters to the 14–parameter similarity relationship given in equation 4.21, rotation into the cartesian reference frame (c.f. §4.2.1) is required. The following expression is used to convert Euler pole parameters to cartesian elements:

$$\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{bmatrix} = \frac{\Omega_{rot}}{10^6} \begin{bmatrix} \cos(\Omega_{lat})\cos(\Omega_{lon}) \\ \cos(\Omega_{lat})\sin(\Omega_{lon}) \\ \sin(\Omega_{lat}) \end{bmatrix} \tag{4.22}$$

By default, DynAdjust applies the Australian plate motion model parameters defined in the ICSM Geocentric Datum of Australia 2020 Technical Manual ICSM (2020). Here, rotation rates for the Australian plate motion model have been supplied as cartesian elements ($\dot{r_x}$, $\dot{r_y}$ and $\dot{r_z}$) aligned with ITRF2014, which is by definition a NNR reference frame. In order to derive the corresponding Euler pole coordinates and rotation rate required for DynAdjust, the following expressions have been adopted:

$$\begin{aligned} \Omega_{lat} &= \arctan\left(\frac{\dot{r}_z^2}{\sqrt{\dot{r}_x^2 + \dot{r}_y^2}}\right) \\ \Omega_{lon} &= \arctan\left(\frac{\dot{r}_y^2}{\dot{r}_x^2}\right) \\ \Omega_{rot} &= \sqrt{\dot{r}_x^2 + \dot{r}_y^2 + \dot{r}_z^2} \end{aligned} \tag{4.23}$$

Using these expressions, the Euler pole parameters expressing the spatial behaviour of the Australian tectonic plate[2] over time are:

$$\begin{aligned} \Omega_{lat} &= 38.2022° \\ \Omega_{lon} &= 32.2447° \\ \Omega_{rot} &= 0.6285°/\text{Ma} \end{aligned} \tag{4.24}$$

If an alternative set of parameters is required, either for stations located on the Australian plate or for those on other plates, §4.6.3 outlines the way in which defined Euler pole parameters can be imported and applied to the reference frame transformation strategy implemented within **reftran**.

4.4 Propagation of variances between reference frames

4.4.1 Local reference frame

Variances in the local reference frame can be expressed in the cartesian reference frame by the law of propagation of variances:

$$\mathbf{V}_{\mathbf{x}_c} = \mathbf{R}_l \mathbf{V}_{\mathbf{x}_l} \mathbf{R}_l^T \tag{4.25}$$

where $\mathbf{V}_{\mathbf{x}_l}$ is a variance matrix in the local reference frame, $\mathbf{V}_{\mathbf{x}_c}$ is a variance matrix in the cartesian reference frame, and $\mathbf{R}_l$ is the rotation matrix given in equation 4.9. Conversely, propagation of variances from the cartesian system into the local reference frame is obtained by:

$$\mathbf{V}_{\mathbf{x}_l} = \mathbf{R}_l^{-1} \mathbf{V}_{\mathbf{x}_c} \left[\mathbf{R}_l^{-1}\right]^T \tag{4.26}$$

2. Note that Figure 4.4 shows the coordinates for the antipole for illustrative purposes.

Since $\mathbf{R}_l$ is orthogonal, $\mathbf{V}_{\mathbf{x}_l}$ may be expressed as:

$$\mathbf{V}_{\mathbf{x}_l} = \mathbf{R}_l^T \mathbf{V}_{\mathbf{x}_c} \mathbf{R}_l \tag{4.27}$$

4.4.2 Polar reference frame

The precision of polar coordinates can be derived from precisions in either cartesian reference frame or the local reference frame. For precisions in the cartesian system, propagation of variances is evaluated by a two step process, by (1) propagation into the local system as described above, and then (2) propagation into the polar form.

To transform a variance matrix $\mathbf{V}_{\mathbf{x}_l}$ in the local reference frame into the polar form $\mathbf{V}_{\mathbf{x}_p}$, propagation of variances is evaluated as:

$$\mathbf{V}_{\mathbf{x}_p} = \mathbf{P} \mathbf{V}_{\mathbf{x}_l} \mathbf{P}^T \tag{4.28}$$

where $\mathbf{P}$ is the Jacobian transformation matrix formed by linearising equations 4.16:

$$\mathbf{P} = \begin{bmatrix} \frac{\partial \theta}{\partial e} & \frac{\partial \theta}{\partial n} & \frac{\partial \theta}{\partial up} \\ \frac{\partial \vartheta}{\partial e} & \frac{\partial \vartheta}{\partial n} & \frac{\partial \vartheta}{\partial up} \\ \frac{\partial s}{\partial e} & \frac{\partial s}{\partial n} & \frac{\partial s}{\partial up} \end{bmatrix} \tag{4.29}$$

To transform a variance matrix directly from the cartesian system to the polar reference frame, propagation is computed by combining equations 4.27 and 4.28:

$$\mathbf{V}_{\mathbf{x}_p} = \mathbf{P} \mathbf{R}_l^T \mathbf{V}_{\mathbf{x}_c} \mathbf{R}_l \mathbf{P}^T \tag{4.30}$$

4.4.3 Geographic reference frame

To transform a variance matrix $\mathbf{V}_{\mathbf{x}_g}$ for geographic coordinates into the cartesian system, propagation of variances takes the familiar form:

$$\mathbf{V}_{\mathbf{x}_c} = \mathbf{T} \mathbf{V}_{\mathbf{x}_g} \mathbf{T}^T \tag{4.31}$$

$\mathbf{T}$ is the Jacobian transformation matrix formed by linearising equations 4.2:

$$\mathbf{T} = \begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \lambda} & \frac{\partial x}{\partial h} \\ \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \lambda} & \frac{\partial y}{\partial h} \\ \frac{\partial z}{\partial \phi} & \frac{\partial z}{\partial \lambda} & \frac{\partial z}{\partial h} \end{bmatrix} \tag{4.32}$$

For the reverse solution, propagation of variances into the geographic coordinate system is via:

$$\mathbf{V}_{\mathbf{x}_g} = \mathbf{T}^{-1} \mathbf{V}_{\mathbf{x}_c} \left[\mathbf{T}^{-1}\right]^T \tag{4.33}$$

4.5 Conversion of projection coordinates

It is often necessary to work with geographic information defined in terms of projection[3] coordinates. DynAdjust supports Universal Transverse Mercator (UTM) and Transverse Mercator (TM) projections[4]. UTM and TM projection coordinates are expressed in terms of Easting E, Northing N and Zone Zo. For UTM projections, the earth is divided into 60 zones. Each zone is $6°$ wide in longitude, extends from $-80°$ latitude to $+80°$ latitude, and is numbered consecutively beginning from zone 1 at $-180°$ to $-174°$ longitude and extending eastwards to zone 60 ($+174°$ to $+180°$ longitude). The origin of each zone is the point of intersection between the equator and the meridian central to the zone, and the central scale factor of the projected zones is 0.9996. To alleviate negative coordinate values in the southern hemisphere, the origin is assigned a false Northing of $10,000,000$m and a false Easting of $500,000$m. Since two points located in different zones can have equal values for Easting and Northing, projection coordinates are unique only when accompanied by the zone number.

The relationship between UTM and TM projection coordinates E, N and Zo and geographic coordinates ϕ, λ and h is given by Redfearn (1948):

$$
\begin{aligned}
E \;=\;& 500,000 + k_0 \Big\{ \nu\omega\cos\phi \;+\; \nu\frac{\omega^3}{6}\cos^3\phi\left(\psi - t^2\right) \\
& + \nu\frac{\omega^5}{120}\cos^5\phi\left[4\psi^3\left(1-6t^2\right) + \psi^2\left(1+8t^2\right) - \psi\left(2t^2\right) + t^4\right] \\
& + \nu\frac{\omega^7}{5040}\cos^7\phi\left(61 - 479t^2 + 179t^4 - t^6\right) \Big\}
\end{aligned}
\tag{4.34}
$$

$$
\begin{aligned}
N \;=\;& 10,000,000 + k_0 \Big\{ m + \nu\sin\phi\frac{\omega^2}{2}\cos\phi \\
& + \nu\sin\phi\frac{\omega^4}{24}\cos^3\phi\left(4\psi^3 + \psi - t^2\right) \\
& + \nu\sin\phi\frac{\omega^6}{720}\cos^5\phi\left[8\psi^4\left(11-24t^2\right) + 28\psi^3\left(1-6t^2\right)\right. \\
& \left. + \psi^2\left(1-32t^2\right) - \psi\left(2t^2\right) + t^4\right] \\
& + \nu\sin\phi\frac{\omega^8}{40320}\cos^7\phi\left(1385 - 3111t^2 + 543t^4 - t^6\right) \Big\}
\end{aligned}
\tag{4.35}
$$

$$
Zo = \frac{\omega}{z_w} \tag{4.36}
$$

3. As the reference ellipsoid is by nature a three–dimensional curved surface, projecting figures on the ellipsoid onto a flat, two–dimensional surface requires distortion in some form or another. Accordingly, different map projections have been designed to preserve one or more of the following geometric properties: shape *(conformal)*, direction (*azimuthal*), distance (*gnomic*) and area (*equal area*). To this end, several map projections are available — *Transverse Mercator* (conformal), *Lambert Conic* (conformal), *Polar Stereographic* (azimuthal, conformal) and *Albers Conic* (equal area).
4. A UTM projection is a TM projection with standardised values for zone width, origin, scale factor and false Easting and Northings.

where:

$$\begin{aligned}
\omega &= \lambda - \lambda_0 && (4.37)\\
z_w &= \text{zone width (in units of } \lambda) && (4.38)\\
\psi &= \frac{\nu}{\rho} && (4.39)\\
t &= \tan\phi && (4.40)\\
m &= a\left(A_0\phi - A_2\sin 2\phi + A_4\sin 4\phi - A_6\sin 6\phi\right) && (4.41)\\
A_0 &= 1 - \left(\frac{e^2}{4}\right) - \left(\frac{3e^4}{64}\right) - \left(\frac{5e^2}{256}\right) && (4.42)\\
A_2 &= \frac{3}{8}\left[e^2 + \left(\frac{e^4}{4}\right) - \left(\frac{15e^6}{128}\right)\right] && (4.43)\\
A_4 &= \frac{15}{256}\left(e^4 + \frac{3e^6}{4}\right) && (4.44)\\
A_6 &= \frac{35e^6}{3072} && (4.45)
\end{aligned}$$

k_0 is the central scale factor, λ_0 is the longitude of the central meridian, z_w is the projection zone width, and ρ is the radii of curvature in the meridian at ϕ.

Conversely, the relationship between geographic coordinates and UTM and TM projection coordinates is given by Redfearn (1948):

$$\begin{aligned}
\phi = \; & \phi' - \left(\frac{t'}{k_0\rho'}\right)\left(\frac{xE'}{2}\right) + \left(\frac{t'}{k_0\rho'}\right)\left(\frac{x^3E'}{24}\right)\left[-4\psi'^2 + 9\psi'\left(1 - t'^2\right) + 12t'^2\right]\\
& - \left(\frac{t'}{k_0\rho'}\right)\left(\frac{x^5E'}{720}\right)\left[8\psi'^4\left(11 - 24t'^2\right) - 12\psi'^3\left(21 - 71t'^2\right)\right.\\
& \left. + 15\psi'^2\left(15 - 98t'^2 + 15t'^4\right) + 180\psi'\left(5t'^2 - 3t'^4\right) + 360t'^4\right]\\
& + \left(\frac{t'}{k_0\rho'}\right)\left(\frac{x^7E'}{40320}\right)\left(1385 + 3633t'^2 + 4095t'^4 + 1575t'^6\right) && (4.46)
\end{aligned}$$

$$\begin{aligned}
\lambda = \; & \lambda_0 + \sec\phi' x - \left(\frac{x^3}{6}\right)\sec\phi'\left(\psi' + 2t'^2\right)\\
& + \left(\frac{x^5}{120}\right)\sec\phi'\left[-4\psi'^3\left(1 - 6t'^2\right) + \psi'^2\left(9 - 68t'^2\right) + 72\psi't'^2 + 24t'^4\right]\\
& - \left(\frac{x^7}{5040}\right)\sec\phi'\left(61 + 662t'^2 + 1320t'^4 + 720t'^6\right) && (4.47)
\end{aligned}$$

where:

$$
\begin{aligned}
E' &= E - E_f && (4.48)\\
N' &= N - N_f \text{ (southern hemisphere only)} && (4.49)\\
x &= \frac{E'}{k_0 \nu'} && (4.50)\\
\psi' &= \frac{\nu'}{\rho'} && (4.51)\\
\lambda_0 &= z_o z_w + \lambda_1 - z_w && (4.52)\\
\phi' &= \sigma + \left(\frac{3\eta}{2} - \frac{27\eta^3}{32}\right)\sin 2\sigma + \left(\frac{21\eta^2}{16} - \frac{55\eta^3}{32}\right)\sin 4\sigma \\
&\quad + \left(\frac{151\eta^3}{96}\right)\sin 6\sigma + \left(\frac{1097\eta^4}{512}\right)\sin 8\sigma && (4.53)\\
\eta &= \frac{f}{2-f} && (4.54)\\
G &= a\,(1-\eta)\left(1-\eta^2\right)\left(1 + \frac{9}{4}\eta + \frac{225}{64}\eta\right)\frac{\pi}{180} && (4.55)\\
\sigma &= \frac{\pi m}{180G} && (4.56)\\
m &= \frac{N'}{k_0} && (4.57)
\end{aligned}
$$

E_f and N_f are the false Easting and Northing values, λ_1 is the longitude of the central meridian of zone 1. ν' and ρ' are the radii of curvature evaluated at the foot–point latitude ϕ' (via equations 4.3 and 4.5), which is the latitude for which the meridian distance m equals N'/k_0.

4.6 Transformation of coordinates and measurements

The least squares adjustment model implemented within DynAdjust is based upon the cartesian reference frame. After loading station information from the input files, **import** converts all station coordinates in geographic and projection systems to their cartesian counterparts. Upon preparing for least squares adjustment, **adjust** forms observation equations in terms of the cartesian reference frame using the principles outlined in §4.2 and if required, performs propagation of variances of *a–priori* GNSS measurement variance matrices (c.f. §4.4). When reporting the adjustment results, **adjust** converts the estimated station coordinates to the user–specified coordinate systems (e.g. geographic and/or projection). If required, **adjust** undertakes propagation of variances of estimated station and GNSS measurement variance matrices to the user–specified system (e.g. cartesian, geographic, local or polar).

The primary purpose of **reftran** is to align all station coordinates and GNSS measurements to a common reference frame and epoch prior to least squares adjustment. **reftran** performs this task using the reference frame information supplied to **import** (either in the input files or via the `--reference-frame` option. c.f. §3.3.1), and the reference frame passed to **reftran**.

4.6.1 Supported reference frames and geodetic datums

DynAdjust supports several commonly used reference frames. To simplify the process of uniquely identifying each frame, DynAdjust adopts the naming convention, abbreviations and system of codes defined in the EPSG Geodetic Parameter Dataset[5] (version 7.9.0). Table 4.1 lists the EPSG codes and reference frames supported by DynAdjust.

Table 4.2 shows the various reference frame transformation combinations supported by DynAdjust and the respective references from which the transformation parameters have been sourced.

If transformation parameters between the reference frames of the input and output stations and measurements do not exist (e.g. ITRF93 – ITRF96), DynAdjust will attempt to perform a transformation in two stages using ITRF2014 as the intermediate step. This functionality maximises the number of transformation capabilities whilst at the same time, maintains as much rigour as possible. However, if parameters between ITRF2014 and either the input and output reference frames do not exist, **reftran** will terminate prematurely with an error message.

5. The EPSG Geodetic Parameter Dataset, or EPSG dataset, is maintained by the Geodesy Subcommittee of the Surveying and Positioning Committee of the International Association of Oil and Gas Producers (OGP). See http://www.epsg.org

Table 4.1: EPSG codes and reference frames

Code	Reference frame name	Abbreviated name
4939	Geocentric Datum of Australia 1994 (geographic)	GDA94
4347	Geocentric Datum of Australia 1994 (cartesian)	GDA94
7843	Geocentric Datum of Australia 2020 (geographic)	GDA2020
7842	Geocentric Datum of Australia 2020 (cartesian)	GDA2020
7789	International Terrestrial Reference Frame, 2014	ITRF2014
5332	International Terrestrial Reference Frame, 2008	ITRF2008
4896	International Terrestrial Reference Frame, 2005	ITRF2005
4919	International Terrestrial Reference Frame, 2000	ITRF2000
4918	International Terrestrial Reference Frame, 1997	ITRF97
4917	International Terrestrial Reference Frame, 1996	ITRF96
4916	International Terrestrial Reference Frame, 1994	ITRF94
4915	International Terrestrial Reference Frame, 1993	ITRF93
4914	International Terrestrial Reference Frame, 1992	ITRF92
4913	International Terrestrial Reference Frame, 1991	ITRF91
4912	International Terrestrial Reference Frame, 1990	ITRF90
4911	International Terrestrial Reference Frame, 1989	ITRF89
4910	International Terrestrial Reference Frame, 1988	ITRF88
4978	World Geodetic System, 1984	WGS84

Table 4.2: Supported transformation parameters

From reference frame	To reference frame	Transformation Parameters
GDA2020	ITRF96, ITRF97, ITRF2000, ITRF2005, ITRF2008, ITRF2014, GDA94	ICSM (2020)
GDA94	ITRF96, ITRF97, ITRF2000, ITRF2005, ITRF2008, GDA2020	Dawson and Woods (2010); ICSM (2020)
ITRF88, ITRF89, ITRF90, ITRF91, ITRF92, ITRF93, ITRF94	ITRF2000, ITRF2008	ITRF (2000, 2008)
ITRF96, ITRF97	ITRF2000, ITRF2008, GDA94, GDA2020	ITRF (2000, 2008); Dawson and Woods (2010)
ITRF2000	ITRF88, ITRF89, ITRF90, ITRF91, ITRF92, ITRF93, ITRF94, ITRF96, ITRF97, ITRF2005, ITRF2008, GDA94, GDA2020	ITRF (2000, 2005, 2008); Dawson and Woods (2010)
ITRF2005	ITRF2000, GDA94, GDA2020	ITRF (2005); Dawson and Woods (2010)
ITRF2008	ITRF88, ITRF89, ITRF90, ITRF91, ITRF92, ITRF93, ITRF94, ITRF96, ITRF97, ITRF2000, ITRF2005, GDA94, GDA2020	ITRF (2008); Dawson and Woods (2010)
ITRF2014	ITRF88, ITRF89, ITRF90, ITRF91, ITRF92, ITRF93, ITRF94, ITRF96, ITRF97, ITRF2000, ITRF2005, ITRF2008, GDA94, GDA2020	Dawson and Woods (2010); ITRF (2016); ICSM (2020)

4.6.2 Relationship between ITRF, IGS and WGS84 reference frames

When post–processing GNSS data to produce high–precision GNSS baselines or point clusters, a common prerequisite is the availability of precise GNSS satellite orbit information. IGS and its participating analysis centres (see http://www.igs.org/) are amongst the most reliable and well respected producers of precise GNSS orbits. Over the course of its history, IGS has aligned the reference frame for its satellite orbit products to a variety of ITRF reference frames. Table 4.3 lists the respective ITRF frames to which the orbit products are aligned and the periods in which those frames apply.

Table 4.3: Alignment of IGS precise orbit product reference frames with ITRF updates over time (source: http://acc.igs.org/igs-frames.html)

ITRF versions	IGS frame	Start date	End date	GPS weeks
ITRF92	ITRF92	02.01.1994	31.12.1994	0730 – 0781
ITRF93	ITRF93	01.01.1995	29.06.1996	0782 – 0859
ITRF94	ITRF94	30.06.1996	28.02.1998	0860 – 0946
ITRF94 retained	ITRF96	01.03.1998	31.07.1999	0947 – 1020
ITRF94 retained	ITRF97	01.08.1999	03.06.2000	1021 – 1064
ITRF97	IGS97	04.06.2000	01.12.2001	1065 – 1142
ITRF2000	IGS00	02.12.2001	10.01.2004	1143 – 1252
ITRF2000 retained	IGb00	11.01.2004	04.11.2006	1253 – 1399
ITRF2005	IGS05	05.11.2006	16.04.2011	1400 – 1631
ITRF2008	IGS08	17.04.2011	06.10.2012	1632 – 1708
ITRF2008 retained	IGb08	07.10.2012	28.01.2017	1709 – 1933
ITRF2014	IGS14	29.01.2017	...	1934 –

Upon processing GNSS data, the reference frame in which the baselines or points will be produced will be the reference frame of the satellite orbit coordinates used in the GNSS processing. This is of course the case if the processed GNSS baselines or points have not been transformed to a user–specified frame by the GNSS processing package. If the GNSS processing software is able to transform the GNSS data to the desired reference frame of the adjustment, no further steps are required. However, to ensure the correct reference frame is selected upon importing GNSS measurements into DynAdjust, Table 4.3 should be used to identify the ITRF frame corresponding to the frame of the orbit products.

For example, if a GNSS campaign was undertaken on 5 May 2005 and precise satellite orbits from IGS were used to produce a set of GNSS baselines and station coordinate estimates, the corresponding IGS frame of those satellite orbits would be IGb00. As the IGb00 frame was aligned to ITRF 2000, the reference frame and epoch of the GNSS baseline measurements should be set to ITRF 2000 and 05.05.2005 within the measurement input file. The following sample shows a GNSS baseline encoded in DNA format (c.f. Table B.8 in Appendix B.1.4 for DNA measurement files).

```
G N-2234             N-6200                                        1.00    1.00    1.00    1.00       ITRF2000     05.05.2005
                                                                    11675.0160 1.3617930e-04
                                                                   -20144.2370 5.0458000e-09 7.1068070e-05
                                                                   -26611.5240 8.5190970e-05-2.7839720e-05 7.6459650e-05
```

Of course, the user is free to select an alternative frame to which the input data should be transformed prior to adjustment.

In cases where broadcast ephemerides have been adopted (rather than precise satellite orbits), the processed data will be in the frame of the GNSS constellation transmitting the broadcast signals. In the case of GPS, the reference frame of the broadcast ephemerides will be WGS84. Despite its name, WGS84 is a dynamic frame which is continuously realised from routine analysis of the control centre stations over time. In order to maintain compatibility with international reference frames, WGS84 has been aligned to various versions of ITRF throughout its 30+ year history. Table 4.4 tabulates the different versions of WGS84, the ITRF version and epoch to which each WGS84 version has been aligned, and the date at which the alignment was adopted.

Table 4.4: Alignment of WGS84 realisations with ITRF updates over time (source: `https://confluence.qps.nl/pages/viewpage.action?pageId=29855173`)

WGS84 version	**Epoch**	**ITRF version and epoch**	**Offset (m)**	**Adopted**	**GPS weeks**
WGS84	1984.0	First realisation (DOPPLER)		1987	0001 – 0729
WGS84 (G730)	1994.0	ITRF 1991	0.7	29.06.1994	0730 – 0872
WGS84 (G873)	1997.0	ITRF 1994	0.2	29.01.1997	0873 – 1149
WGS84 (G1150)	2001.0	ITRF 2000	0.06	20.01.2002	1150 – 1673
WGS84 (G1674)	2005.0	ITRF 2008	0.01	08.02.2012	1164 – 1761
WGS84 (G1762)	2005.0	ITRF 2008 retained	0.01	16.10.2013	1762 –

To avoid misrepresentation of the appropriate frame to which stations and measurements are aligned, DynAdjust prevents the user from using WGS84. Hence, where WGS84 coordinates and/or GPS baselines are to be introduced into an adjustment, the user should substitute the corresponding ITRF version from Table 4.4 for the input GNSS measurements based on the date of that data. Although not provided here, the same will need to be applied for the other GNSS constellations (e.g. Beidou, Galileo, GLONASS, etc.).

4.6.3 Tectonic plate boundary and motion information

In addition to supporting transformations between a variety of dynamic reference frames, DynAdjust supports the projection of coordinates between epochs on a single (dynamic) frame using a plate motion model. By default, if no other options are provided, the Australian plate motion model ICSM (2020) is used (c.f. equation 4.24). From Version 1.1.0, DynAdjust provides an option to select and apply an alternative plate motion model to the transformation of stations and GNSS measurements that lie inside or outside the Australian tectonic plate.

At the present time, there is no single, authoritative list of global tectonic plates and associated rotation rates. Rather, several global and regional sets of tectonic plates with associated boundaries and Euler pole parameters have been published DeMets et al. (1990, 1994); Bird (2003); Prawirodirdjo and Bock (2004); DeMets et al. (2010); Argus et al. (2011); Kreemer et al. (2014); Altamimi et al. (2017, for example). In excess of 20 plate motion models have been established over the last three decades. Moreover, with a constantly changing crust, there is an ongoing need to monitor and re–estimate the motions of the assumed plates and their boundaries.

For these reasons, DynAdjust provides an option for users to select the plate motion model of choice (e.g. `PB2002`, `NNR-NUVEL-1A`, `NNR-MORVEL56`, `ITRF2014`, `GSRM 2014`, etc.). With this option, two input files are required — (1) a file containing sets of polygons representing tectonic plate boundaries and, (2) a file containing the Euler pole parameters associated with each tectonic plate. With the release of DynAdjust Version 1.1.0, plate boundary and associated Euler pole parameter input files for NNR-MORVEL56 Argus et al. (2011) and PB2002 Bird (2003) have been made available for download from the sampleData folder in the GitHub repository. These file pairs are named MORVEL56_plates.dig and NNR-MORVEL56_poles.dat, and PB2002_plates.dig and PB2002_poles.dat respectively and are accessible via:

https://github.com/icsm-au/DynAdjust/tree/master/sampleData

The structure and format of the tectonic plate boundaries file is shown in Figure 4.5. In this file, each plate is defined by a two–character identifier and a closed list of comma separated coordinate pairs (longitude, latitude) in decimal degrees, in anti–clockwise order. The last coordinate pair is the same as the first. Rows beginning with an asterisk are treated as comments and are ignored, except for lines commencing with '*** end'.

```
* PLATE ID (two char)
*  Longitude, Latitude (decimal degrees, counter clockwise)
*  ...
* *** end of line segment ***
NB
  -0.438,  -54.852
  -0.039,  -54.677
   ...
  -0.438,  -54.852
*** end of line segment ***
AM
  132.824,  30.754
  132.965,  30.970
  ...
  132.824,  30.754
*** end of line segment ***
 ...
```

Figure 4.5: Plate boundaries file

Figure 4.7 illustrates the fifty six NNR-MORVEL56 plate geometries. Note that forty nine of these are defined by PB2002. The other seven are either based on the work of MORVEL or adapted from PB2002.

The structure and format of the Euler pole parameters file is shown in Figure 4.6. In this file, the Euler pole parameters are defined by a two–character identifier (corresponding to the plate identifiers in the plate boundaries file), spherical latitude, spherical longitude (both in decimal degrees) and a right–handed (anti–clockwise) rotation rate in degrees per million years. Rows beginning with an asterisk are treated as comments and are ignored.

```
* ID (two char), Latitude (decimal degrees), Longitude (decimal degrees), Rotation (Deg/Ma Anti-clockwise)
NB   47.68    -68.44    0.292  Nubia
AM   63.17   -122.82    0.297  Amur
AN   65.42   -118.11    0.250  Antarctica
...
```

Figure 4.6: Euler pole parameters file

Note that the pole parameters in NNR-MORVEL56 are relative to the "no net rotation" (NNR) reference frame, meaning that they describe the motions of all plates with respect to the earth in an absolute sense. The pole parameters in PB2002 are treat the Pacific plate (PA) as fixed and express the motions of all other plates relative to it. Hence, care must be taken to ensure the correct model is selected, as propagating coordinates through time according to a frame that is inconsistent with the frame of the station and measurement data will lead to anomalous results.

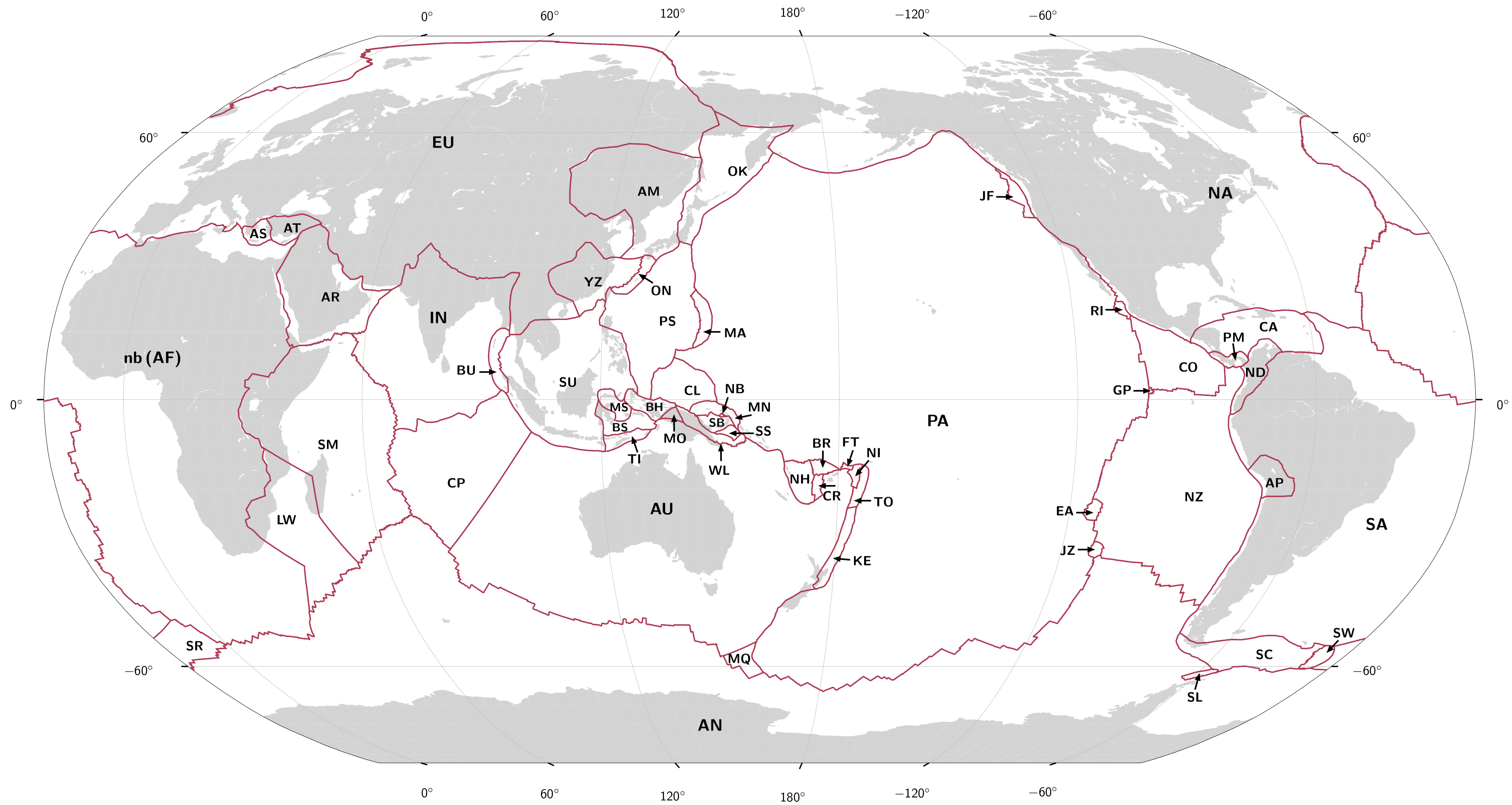


Figure 4.7: Tectonic plate boundaries used to derive the NNR-MORVEL56 plate motion model (Argus et al., 2011).

4.7 Transforming station coordinates and measurements

To transform station coordinates and measurements, type `reftran` on the command line followed by the network name, and then `--reference-frame` (or its shortcut `-r`) followed by the desired reference frame to which all imported stations and GNSS measurements should be aligned. For example, using the `skye` project, the following command shows how to transform the input stations and coordinates to GDA94:

```
reftran skye -r GDA94
```

If a DynAdjust Project File (c.f. Chapter 2) is to be used, type `dynadjust` on the command line, then `--project-file` (or its shortcut `-p`) followed by the full path for the project file, and then the option to invoke **reftran**:

```
dynadjust -p skye.dnaproj --reftran
```

No other options are required, as **reftran** will be invoked using the options and arguments contained in the project file `skye.dnaproj`.

Upon executing **reftran**, the binary station and measurement files `skye.bst` and `skye.bms` will be loaded into memory. For each station and GNSS measurement found in the binary files, **reftran** will verify whether the associated reference frame abbreviation is one of the supported reference frames (c.f. Table 4.1). If the user–specified reference frame is not supported, **reftran** will terminate prematurely with an error message. If the frame is in the list of supported frames, the appropriate reference frame parameters will be selected. As it is possible for the input stations and measurements to be aligned to multiple reference frames, **reftran** performs this test for every station and measurement. Then, depending on whether one or more dynamic frames are involved, either equation 4.20 or 4.21 will be applied. Once all stations and measurements have been transformed, the binary station and measurement files are updated.

As described in §4.6.1, if there are no direct transformation parameters relating the input and output reference frames, **reftran** will attempt to transform the coordinates in two stages using ITRF2014 as the intermediate step. If parameters between ITRF2014 and either the input and output reference frames do not exist, **reftran** will terminate prematurely with an error message.

At this point it should be noted that in order to achieve a rigorous transformation between reference frames, the correct reference frame of the input stations and measurements must be set within the input files or via the call to **import** (c.f. §3.3.1). Mis–identification of the input reference frame will lead to incorrect results.

4.7.1 Progress reporting and import log

Once **reftran** has been executed with the relevant options and arguments, **reftran** will report to the screen a number of items relevant to the project (e.g. name, location of input and output files); the number of stations and measurements transformed or, in the case of failure, an error message corresponding to the type of failure. Figure 4.8 shows the information reported to the screen upon normal program execution.

```
+ Options:
  Network name:                     skye
  Input folder:                     .
  Output folder:                    .
  Binary station file:              ./skye.bst
  Binary measurement file:          ./skye.bms
  Target reference frame:           GDA94

+ Transforming stations and measurements... done.
+ Transformed 6 stations.
+ Transformed 27 measurements.
```

Figure 4.8: **reftran** progress reporting of station and measurement transformations

The amount of information will vary depending on the options and arguments supplied to **reftran**, such as folder locations, input files, target reference frame, target epoch, tectonic plate information and export options. This information is also logged to a file named `skye.rft`. Any errors encountered during the various tasks will result in premature program termination and a description of the error.

4.7.2 Configuring reference frame transformations

Transformations involving different frames and epochs

When supplying the reference frame via `--reference-frame` (or its shortcut `-r`), **reftran** will undertake a transformation using the default reference frame epoch. For static reference frames (such as GDA94 and GDA2020), no further options are required. However, when transformations between dynamic reference frames are required, an optional epoch argument may be supplied to align the coordinates to a particular epoch. For this purpose, add `--epoch` (or its shortcut `-e`) followed by the desired epoch in the form of `dd.mm.yyyy`:

```
reftran skye -r itrf2005 -e 05.05.2005
```

If a transformation using an epoch corresponding with today's date is required, simply type `today`:

```
reftran skye -r itrf2005 -e today
```

Projections of coordinates to different epochs on the same frame

If the coordinates of the project need to be projected to another epoch on the same reference frame, re–enter the reference frame with a new reference epoch (e.g. 01 January 2030):

```
reftran skye -r itrf2005 -e 01.01.2030
```

Applying an alternative (or global) plate motion model

Using the previous command, and provided there are no changes to the reference frame, the transformation will only involve a propagation of coordinates using Australian plate motion model parameters (equation 4.24 in §4.3.1). If there is a change to both reference frame and epoch, **reftran** will apply both reference frame transformation parameters (i.e. from the supported list given in Table 4.2) and the Euler pole parameters.

Should an alternative plate motion model be used, pass to **reftran** the option `--plate-model-option` followed by `1`. This will cause **reftran** to interpolate plate motion model parameters based on the location of each station with respect to a defined set of global tectonic plates. For this option, a global tectonic plate boundary file and corresponding Euler plate motion model parameters file must be provided via the options `--plate-boundary-file` (or `-b`) and `--plate-pole-file` (or `-m`) respectively, as follows:

```
reftran skye -r itrf2005 -e 01.01.2030 --plate-model-option 1 -b PB2002_plates.dig -m PB2002_poles.dat
```

4.8 Data export

Upon transforming stations and measurements to an alternative reference frame, **reftran** provides a capacity to export the transformed data into a number of different file formats, including DNA, DynaML and GeodesyML file formats.

To export transformed station and measurement information into DNA format, pass to **reftran** the option `--export-dna-files`. Similarly, to export station and measurement information into DynaML format, pass the option `--export-xml-files`. For example, the following command will export the stations and measurements in the `skye` project to DNA format in `GDA94`:

```
reftran skye -r GDA94 --export-dna
```

Running this command will produce two files named `skye.GDA94.stn` and `skye.GDA94.msr`, which correspond to the station and measurement files respectively.

Similarly, the option `--export-xml-files` will produce two XML files named `skyestn.GDA94.xml` and `skyemsr.GDA94.xml` corresponding to the station and measurement files respectively. To produce a single DynaML file, pass to **reftran** the option `--single-xml-file` with `--export-xml-files`.

Chapter 5

Import and export of geoid information

5.1 Introduction

In recognition of the inescapable influence of the anomalous gravity field upon terrestrial geodetic measurements, and the difference that exists between ellipsoidal and gravity–based height systems, DynAdjust provides an efficient means for incorporating geoid information within an adjustment project. Introducing this information into a project permits the rigorous combination of GNSS measurements and terrestrial measurements within a single adjustment, the efficient transformation of heights from one height system to another, and the rigorous estimation of parameters in cartesian, geodetic and local reference frames.

The task of introducing geoid information into a DynAdjust project can be performed using a geoid file as described in §3.2.4, or via automatic interpolation from a geoid model using **geoid**. The purpose of this chapter is to explain the latter. This chapter also explains how to interpolate geoid information in interactive and text file modes, how to transform between ellipsoid heights and orthometric heights, how to convert geoid models in the legacy WINTER file format to binary grid files in National Transformation version 2.0 (NTv2) grid file format Junkins and Farley (1995), and how to display the metadata contained in an existing NTv2 binary grid file. The application of geoid information to the observation equations for terrestrial geodetic measurements is dealt with in Chapter 7.

5.2 Fundamental concepts

Height determination demands a reasonable level of care due to the number of systems to which a height can be related. To minimise the error that may be introduced into an adjustment through incorrectly defined station heights and/or height measurements, users should be aware of the intrinsic differences between these systems. This section provides a basic overview of height systems and geoid models, and will define the terms used throughout this document. For further information of height systems and geoid modelling, the reader is encouraged to refer to Heiskanen and Moritz (1993), Hofmann–Wellenhof and Moritz (2006), Vaniček and Krakiwsky (1986), Bomford (1980), Rapp (1961) and Featherstone and Kuhn (2006).

Consider the surfaces labelled natural terrain, ellipsoid, geoid, mean sea level and sea surface shown in Figure 5.1. The terrain is the well understood ground surface upon which features of interest and survey control marks are located. The ellipsoid is the mathematical surface chosen to best represent the geometric size and shape of the Earth — either as a whole, or in a particular region. It serves as the fundamental basis upon which positioning and navigation, map projections and geodetic calculations can be based. It is also the surface to which GNSS measurements on the terrain are related (c.f. Figure 4.1). As shown in Figure 5.1, ellipsoid height h represents the height of point p

above the ellipsoid, measured along the normal to the ellipsoid at point p'.

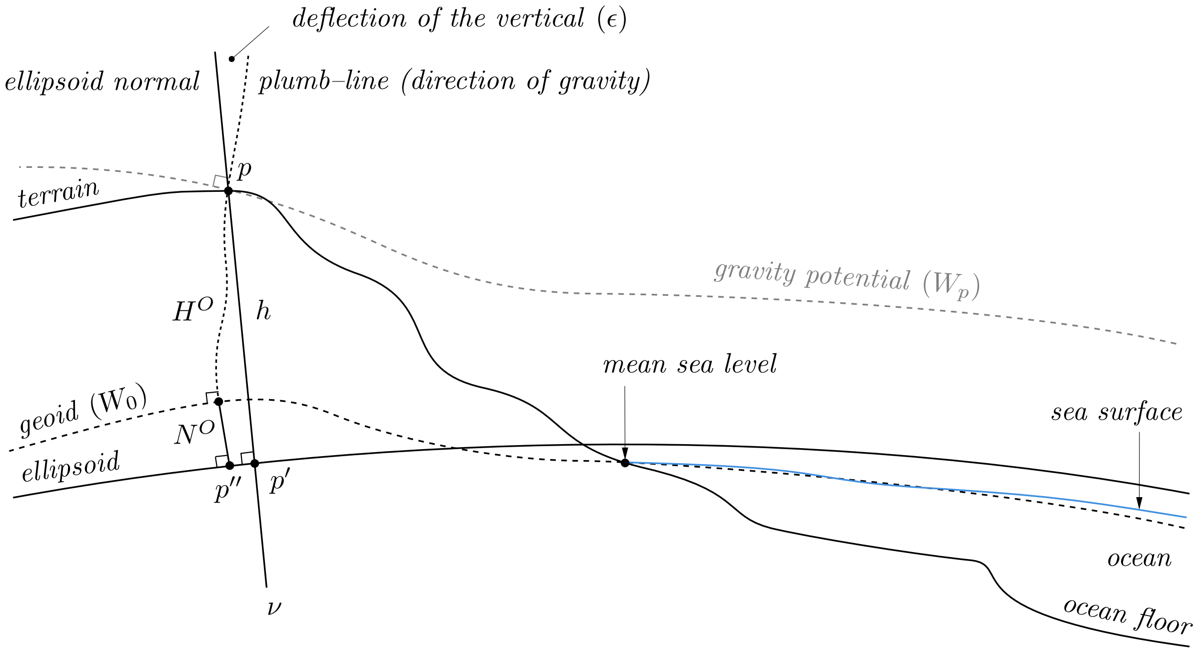


Figure 5.1: The relationships between the natural terrain, ellipsoid, geoid, mean sea level, and sea surface.

Mean sea level (MSL) is an observed tidal datum and is used as the conventional reference surface to which heights on the terrain (e.g. contours, heights of mountains, flood plains, etc.) and other tidal datums are related. The geoid is the surface of constant gravity potential (or equipotential) chosen to best approximate MSL. It is often shown in the literature as W_0 in order to distinguish it as the reference equipotential surface to which other gravity potentials (such as the potential W_p at point p) are related. Being a gravity potential surface (as opposed to a geometrical surface like the ellipsoid), the geoid is consistently perpendicular to the direction of gravity (i.e. the plumb–line). In this sense, it represents the "spirit–level" surface at which fluids stabilise.

As suggested by the curved plumb–line, the direction of gravity varies along the path of the plumb–line on account of variations in land (or subsurface) mass–density and other gravity anomalies. The deflection of the vertical ϵ is the angle between the direction of the plumb–line and the normal to the ellipsoid and is generally expressed in two dimensions, as a north–south component (deflection in the prime meridian, ξ) and an east–west component (deflection in the prime vertical, η). The sea surface varies from the geoid due to the dynamic topography of the ocean caused by variations in temperature, regional currents, gravitational effects, and salinity, for example. For similar reasons, MSL also varies from the geoid.

As shown in Figure 5.1, true orthometric height H^O is the length of the curved plumb–line between point p and the geoid. The geoid–ellipsoid separation N^O is the length of the normal to the ellipsoid, from p'' to the geoid. In practice, it is not a simple matter to compute the true orthometric height since the integral mean of gravity between p and the geoid is a rather complex and practically challenging phenomenon to quantify. For this reason, several approximations of H^O have been proposed, such as Helmert orthometric (H^H), normal (H^N) and normal orthometric (H^{NO}). It is not necessary to explain all approximations for H^O, although brief comments will be made on the normal orthometric height system due to its frequent use in establishing national vertical datums.

Consider Figure 5.2. Normal orthometric height H^{NO} is the length of the curved normal gravity plumb–line between point p and the quasigeoid. It is based on a theoretical (ellipsoidal) normal gravity field and can be readily computed without knowing the nature of the subsurface mass–density or needing to take gravity observations. As a result of approximating the Earth's gravity field with a normal gravity field, propagation of normal orthometric heights through spirit levelling requires a relatively simple correction (known as the normal orthometric correction) to account for the change in gravity according to the change in latitude along the levelling run.

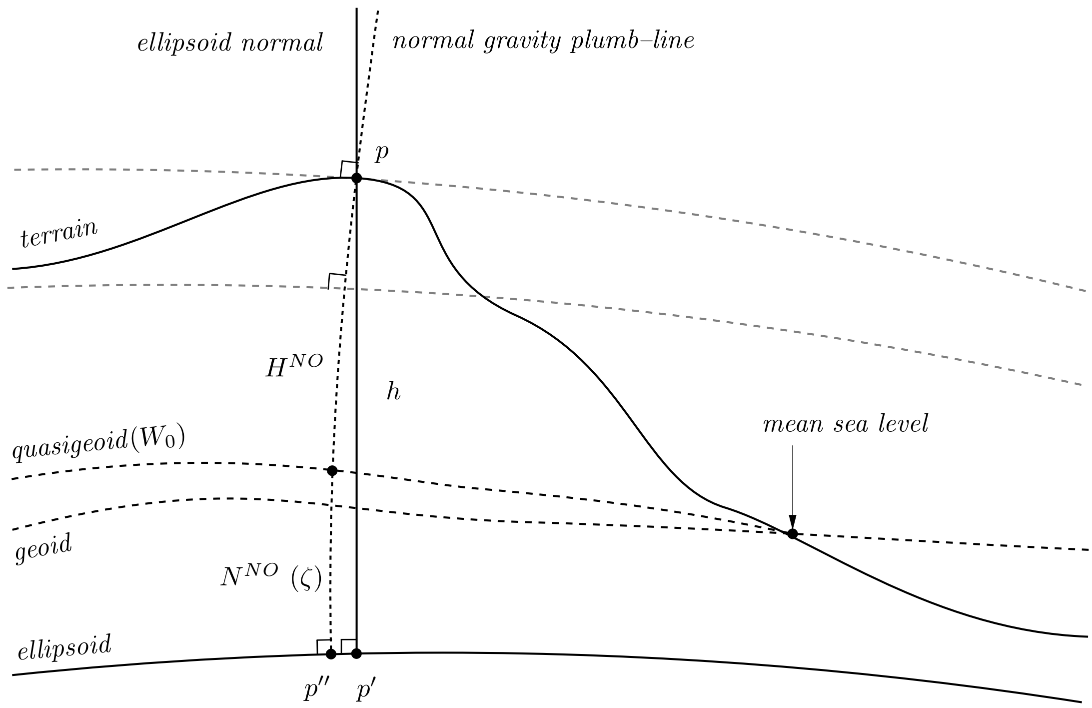


Figure 5.2: The relationship between the geoid and quasigeoid, and between ellipsoid height (h) and normal orthometric height (H^{NO})

The height of the quasigeoid above the ellipsoid is labelled N^{NO} and is often cited in the literature as the height anomaly ζ. The quasigeoid is given its name since it is not an equipotential surface and thereby does not properly describe the flow of fluids. The geoid and quasigeoid are coincident at MSL, but diverge, more or less, over land with respect to elevation. According to Featherstone and Kuhn (2006), this divergence may reach up to 3.4 metres in the Himalayas and 0.15 metres in Australia. The point to note is that the quasigeoid, being an approximation, does not represent true gravity and hence, does not provide for estimating true orthometric heights and height differences.

5.2.1 Geoid models

There are numerous geoid and quasigeoid models (so called) in use throughout the world. A comprehensive review of all such models is beyond the scope of this document. However, it is worthwhile noting the subtle differences that may be found between such models.

Firstly, reflecting on the concepts discussed in the previous section, a gravimetric geoid model provides N^{O} values, whereas a gravimetric quasigeoid model generally provides approximations of it, such as N^{NO} or N^{H}. Secondly, the adopted reference ellipsoid for each model is not unique. Although international practice shows a global trend towards the use of the Geodetic Reference System 1980 (GRS80), some older models may be based upon the World Geodetic System 1984 (WGS84). Thirdly, a geoid or quasigeoid model can be derived from a global equipotential model alone, such as EGM96 or EGM2008, or a combination of gravimetric data sourced from a global model, regional gravity observations and other gravity modelling.

Another point to note is that there are models which also include a geometric component to account for known distortions in a particular vertical datum. In the Australian context, AHD was established using a normal–orthometric height system. Due to various reasons, AHD contains numerous distortions of various magnitudes and does not coincide uniformly with the quasigeoid. Of the four national geoid models published since 1990 for use over the Australian continent[1], only AUSGeoid09 includes a geometric component that models the difference between the gravimetric quasigeoid and the zero surface of the AHD Featherstone et al. (2011). Hence, AUSGeoid09 provides for converting between the GRS80 ellipsoid heights and AHD heights, not heights relative to the gravimetric quasigeoid.

Therefore, when selecting a geoid model from which to interpolate geoid/quasigeoid–ellipsoid values and deflections of the vertical, it is imperative that the model supplied to **geoid** is compatible with the adjustment project reference frame (and ellipsoid), and the supplied station heights and height measurements.

5.2.2 Conventions used in DynAdjust

Although there is a tangible difference between the geoid and the quasigeoid (and other approximations of the geoid), the term *geoid* will be used throughout this document to refer to gravimetric geoid and quasigeoid models generally. Similarly, the term *geoid–ellipsoid* separation (N) will be used to refer to both the separation between the geoid and the ellipsoid (N^O) and the separation between the quasigeoid (and other geoid approximations) and the ellipsoid (N^N, ζ). Likewise, *orthometric height* (H) will be used to refer to true orthometric (H^O), Helmert (H^H), normal (H^N) and normal orthometric (H^{NO}) height. The practical implication is that whenever these terms are cited, the user is responsible for interpreting them according to the context of the adjustment project.

5.3 Grid file interpolation

By far, the most simple, convenient, efficient and repeatable means for retrieving geoid information from a geoid model is to use interpolation from a structured grid file defined by regularly–spaced grid nodes. Using a regular grid file, geoid information can be interpolated at will for arbitrarily located points via one of several interpolation methods. The information that can be incorporated within a DynAdjust adjustment includes values for geoid–ellipsoid separation N, the north–south and east–west deflections of the vertical ξ and η, and the respective uncertainties σ_N, σ_ξ and σ_η. When attempting to interpolate this information from a gridded geoid model, **geoid** provides an ability to derive the unknown parameter values using bilinear interpolation and bicubic interpolation. These methods are briefly described in the following sections.

5.3.1 Bilinear interpolation

The method of bilinear interpolation uses a quadratic form to calculate the desired parameter values for an arbitrary point from a two–dimensional array of grid nodes. Bilinear interpolation is well suited to the task of interpolating from grids having a relatively high resolution with respect to variability in the grid node parameter values, or where higher order curve (or polynomial) fitting offers minimal improvement in accuracy. In this method, only the four surrounding grid nodes are required. Figure 5.3 illustrates the concept of interpolating an unknown geoid–ellipsoid separation N at point p from

1. AUSGeoid90, AUSGeoid93, AUSGeoid98 and AUSGeoid09

the surrounding grid nodes using bilinear interpolation. The same concept applies to interpolating ξ and η.

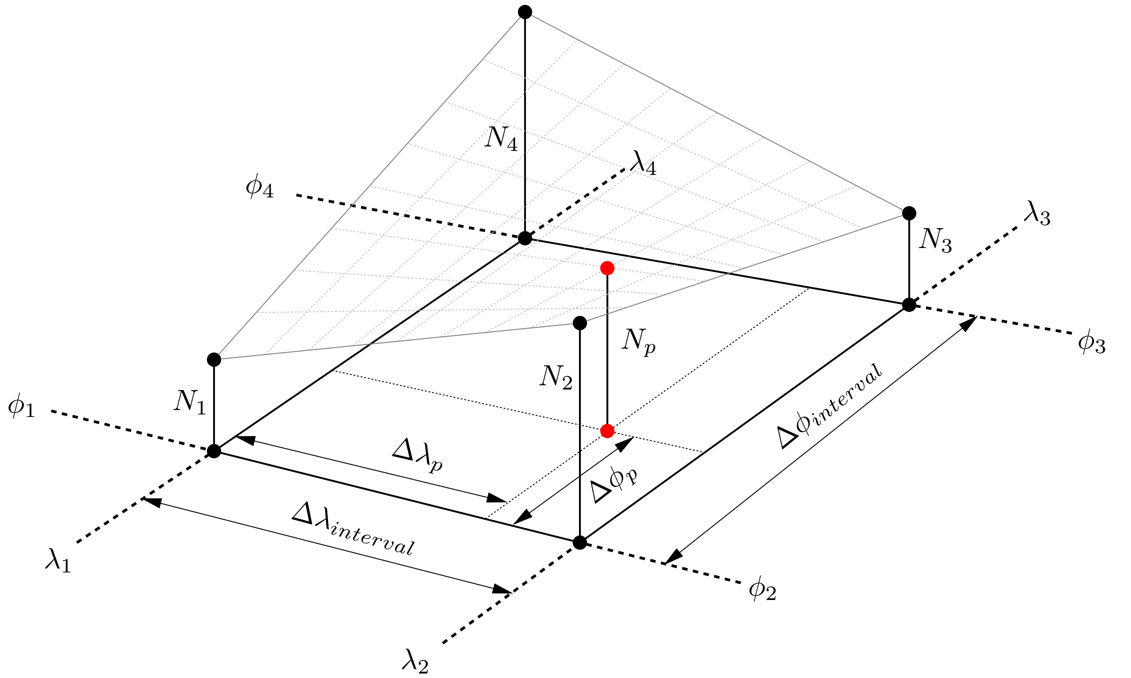


Figure 5.3: Bilinear interpolation concept

The formula to compute N at point p using bilinear interpolation is given by Press et al. (2002):

$$N_p = \alpha_{00} + \alpha_{10}t + \alpha_{01}u + \alpha_{11}tu \tag{5.1}$$

$$= \sum_{i=0}^{1}\sum_{j=0}^{1} \alpha_{ij}t^i u^j \tag{5.2}$$

where:

$$a_{00} = N_1 \tag{5.3}$$

$$a_{10} = N_2 - N_1 \tag{5.4}$$

$$a_{01} = N_4 - N_1 \tag{5.5}$$

$$a_{11} = N_1 + N_3 - N_2 - N_4 \tag{5.6}$$

$$t = \frac{\Delta\lambda_p}{\Delta\lambda_{int}} \tag{5.7}$$

$$u = \frac{\Delta\phi_p}{\Delta\phi_{int}} \tag{5.8}$$

$$\Delta\lambda_p = \lambda_p - \lambda_1 \tag{5.9}$$

$$\Delta\phi_p = \phi_p - \phi_1 \tag{5.10}$$

Here, N_1, N_2, N_3 and N_4 are retrieved from the grid file. $\Delta\lambda_{int}$ and $\Delta\phi_{int}$ are the grid node intervals in the east–west and north–south directions respectively. Since the majority of grid files are orthogonal with regularly spaced grid nodes, $\Delta\lambda_{int}$ and $\Delta\phi_{int}$ will be constant for throughout each grid. The exception to this convention is when a geoid grid file contains one or more sub–grids which have a higher resolution than the parent grid. However, in these instances, the values for $\Delta\lambda_{int}$ and $\Delta\phi_{int}$ will most likely be constant for each sub–grid. To interpolate ξ_p and η_p, the N terms in equations 5.3, 5.4, 5.5 and 5.6 are replaced with the corresponding ξ and η values for grid nodes 1 to 4.

5.3.2 Bicubic interpolation

Bicubic interpolation uses a third degree polynomial, in two dimensions, to approximate a continuous (or smooth) surface from which to calculate the unknown quantities. For this purpose, bicubic interpolation requires the parameter values at sixteen grid nodes surrounding the arbitrary point. It is computationally less efficient, but may be necessary where the grid node spacing is large or there is high variability in the grid node parameter values.

Conceptually, the method of bicubic interpolation involves two steps. Firstly, a two–dimensional polynomial is fitted to the sixteen grid node parameter values and, secondly, the value at the arbitrary point is evaluated. To fit the third order polynomial to the surrounding parameter values, sixteen values must be evaluated for the four grid nodes — four parameter values, four east–west gradients, four north–south gradients, and four cross–product gradients. All gradients (i.e. partial derivatives) are computed using numerical differentiation from the sixteen parameter values $N_{1\ldots16}$.

The formula to compute N at point p using bicubic interpolation is given by Press et al. (2002):

$$N_p = \sum_{i=0}^{3}\sum_{j=0}^{3}\alpha_{ij}t^i u^j \tag{5.11}$$

Evaluating α involves a linear transformation of a vector x representing the four parameter values and twelve derivatives using a coefficient matrix $\mathbf{C}$ and is given by:

$$\alpha = \mathbf{C}^{-1}x$$

where $\mathbf{C}^{-1}$ is the inverse of the coefficient matrix:

$$\mathbf{C}^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -3 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
-3 & 3 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3 & 0 & 0 & -2 & -1 & 0 & 0 \\
9 & -9 & 9 & -9 & 6 & 3 & -3 & -6 & 6 & -6 & -3 & 3 & 4 & 2 & 1 & 2 \\
-6 & 6 & -6 & 6 & -4 & -2 & 2 & 4 & -3 & 3 & 3 & -3 & -2 & -1 & -1 & -2 \\
2 & -2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 1 & 1 & 0 & 0 \\
-6 & 6 & -6 & 6 & -3 & -3 & 3 & 3 & -4 & 4 & 2 & -2 & -2 & -2 & -1 & -1 \\
4 & -4 & 4 & -4 & 2 & 2 & -2 & -2 & 2 & -2 & -2 & 2 & 1 & 1 & 1 & 1
\end{bmatrix}$$

The elements of x are calculated as follows:

$$\begin{aligned}
x_1 &= N_1 & x_5 &= \frac{\partial N_1}{\partial x}\Delta\lambda_{int} & x_9 &= \frac{\partial N_1}{\partial y}\Delta\phi_{int} & x_{13} &= \frac{\partial^2 N_1}{\partial x \partial y}\Delta\lambda_{int}\Delta\phi_{int} \\
x_2 &= N_2 & x_6 &= \frac{\partial N_2}{\partial x}\Delta\lambda_{int} & x_{10} &= \frac{\partial N_2}{\partial y}\Delta\phi_{int} & x_{14} &= \frac{\partial^2 N_2}{\partial x \partial y}\Delta\lambda_{int}\Delta\phi_{int} \\
x_3 &= N_3 & x_7 &= \frac{\partial N_3}{\partial x}\Delta\lambda_{int} & x_{11} &= \frac{\partial N_3}{\partial y}\Delta\phi_{int} & x_{15} &= \frac{\partial^2 N_3}{\partial x \partial y}\Delta\lambda_{int}\Delta\phi_{int} \\
x_4 &= N_4 & x_8 &= \frac{\partial N_4}{\partial x}\Delta\lambda_{int} & x_{12} &= \frac{\partial N_4}{\partial y}\Delta\phi_{int} & x_{16} &= \frac{\partial^2 N_4}{\partial x \partial y}\Delta\lambda_{int}\Delta\phi_{int}
\end{aligned}$$

The twelve gradient values $\partial N_i/\partial x$, $\partial N_i/\partial y$ and $\partial^2 N_i/\partial x\partial y$ $(i = 1, \ldots, 4)$ are evaluated as follows:

$$\begin{aligned}
\frac{\partial N_1}{\partial x} &= \frac{N_2 - N_{16}}{2\Delta\lambda_{int}} & \frac{\partial N_1}{\partial y} &= \frac{N_4 - N_6}{2\Delta\phi_{int}} & \frac{\partial^2 N_1}{\partial x \partial y} &= \frac{N_3 - N_7 - N_{15} + N_5}{4\Delta\lambda_{int}\Delta\phi_{int}} \\
\frac{\partial N_2}{\partial x} &= \frac{N_9 - N_1}{2\Delta\lambda_{int}} & \frac{\partial N_2}{\partial y} &= \frac{N_3 - N_7}{2\Delta\phi_{int}} & \frac{\partial^2 N_2}{\partial x \partial y} &= \frac{N_{10} - N_8 - N_4 + N_6}{4\Delta\lambda_{int}\Delta\phi_{int}} \\
\frac{\partial N_3}{\partial x} &= \frac{N_{10} - N_4}{2\Delta\lambda_{int}} & \frac{\partial N_3}{\partial y} &= \frac{N_{12} - N_2}{2\Delta\phi_{int}} & \frac{\partial^2 N_3}{\partial x \partial y} &= \frac{N_{11} - N_9 - N_{13} + N_1}{4\Delta\lambda_{int}\Delta\phi_{int}} \\
\frac{\partial N_4}{\partial x} &= \frac{N_3 - N_{15}}{2\Delta\lambda_{int}} & \frac{\partial N_4}{\partial y} &= \frac{N_{13} - N_1}{2\Delta\phi_{int}} & \frac{\partial^2 N_4}{\partial x \partial y} &= \frac{N_{12} - N_2 - N_{14} + N_{16}}{4\Delta\lambda_{int}\Delta\phi_{int}}
\end{aligned}$$

Values for t, u, $\Delta\lambda_p$ and $\Delta\phi_p$ are evaluated from equations 5.7, 5.8, 5.9 and 5.10, and $\Delta\lambda_{int}$ and $\Delta\phi_{int}$ are the grid node intervals. Finally, N is evaluated at point p from equation 5.11 using the following pseudocode:

$N_p = 0$
for $i = 0$ to 4 **do**
  $N_p = N_p * t + a_{i,4} * u + a_{i,3} * u + a_{i,2} * u + a_{i,1}$ ▷ equation 5.11
end for

5.4 Import of geoid information

To introduce geoid information into a project, type `geoid` at the command prompt, followed by one or more options and arguments. The complete command line reference for **geoid** is given in §A.4. If no program options are provided upon program execution, the command line reference for **geoid** will be displayed.

If a DynAdjust Project File (c.f. Chapter 2) is to be used, provide `--project-file` (or its shortcut `-p`) followed by the full path for the project file:

```
geoid -p skye.dnaproj
```

No other options are required, as **geoid** will be invoked using the options and arguments contained in the project file `skye.dnaproj`.

To import geoid information using command line options and arguments, type `geoid` followed by the network name[2], and then `--ntv2-file` (or its shortcut `-g`) followed by the full path for the NTv2 binary geoid file:

```
geoid skye -g ausgeoid09.gsb
```

Upon executing **geoid**, the binary station file `skye.bst` and the header records from the NTv2 geoid grid file `ausgeoid09.gsb` will be loaded into memory. For each station found in the binary station file, **geoid** will attempt to interpolate the geoid–ellipsoid separations and the north–south and east–west deflections of the vertical from the surrounding grid nodes. If the station lies outside the limits of the geoid grid file, no information will be retrieved for that station. Once the information for all stations has been interpolated, the binary station file and corresponding DynAdjust project file will be updated.

By default, bicubic interpolation (c.f. §5.3.2) is used to interpolate values from the surrounding grid nodes. To interpolate geoid information using bilinear interpolation (c.f. §5.3.1), add to the `geoid` command the option `--interpolation-method` (or its shortcut `-m`) followed by a zero (`0`):

```
geoid skye -g ausgeoid09.gsb -m 0
```

Since the least squares adjustment model implemented within **DynAdjust** is based upon the cartesian reference frame, all orthometric heights contained in the station file must be transformed to ellipsoid heights. This can be achieved by adding the option `--convert-stn-hts`:

```
geoid skye -g ausgeoid09.gsb --convert-stn-hts
```

When the option `--convert-stn-heights` is provided, **geoid** will determine whether or not to transform a station's height by examining the coordinate type (c.f. §B.1.3, §B.3.1). If the coordinate type is `LLH` or `UTM`, **geoid** will treat the supplied height as an orthometric height and transform it to an ellipsoidal height using the interpolated N value. If the coordinate type is `LLh`, **geoid** will treat the station height as ellipsoidal, leaving it as–is. The application of interpolated geoid information to terrestrial geodetic measurements prior to running an adjustment will be explained in Chapter 7.

When complete, **geoid** will report the number of stations for which geoid information was interpolated or, in the case of failure, an error message corresponding to the type of failure. Using the `geoid` project, Figure 5.4 shows the information reported to the screen upon normal program execution.

2. Note that the option `--network-name` is the default option to **geoid** and as such, the network name may be passed to **geoid** without the option text.

```
+ Options:
  Network name:                   skye
  Input folder:                   .
  Output folder:                  .
  Binary station file:            ./skye.bst
  Convert orthometric heights:    Yes
  Geoid grid file:                ausgeoid09.gsb
  Interpolation method:           Bi-cubic
  Input coordinate format:        Degrees minutes seconds

+ Binary station file interpolation mode.

+ Opening grid file... done.
+ Interpolating geoid components and reducing
  heights to the ellipsoid... done.
+ Interpolated data for 6 points.

+ Geoid file interpolation took 0.007s
```

Figure 5.4: **geoid** progress reporting of geoid grid file interpolation

5.5 Arbitrary interpolation and height transformation

In addition to introducing geoid information into an adjustment project, **geoid** offers two modes for interpolating geoid information from a geoid grid file — interactive mode and text file mode. With the latter, it is also possible to transform a file of heights from one height system (or vertical datum) to another.

5.5.1 Interactive mode

Interpolating geoid information in interactive mode involves using the command line to specify arbitrary interpolation points one at a time. For instance, to retrieve information for an arbitrary location of -33° 24' 36" latitude and 149° 39' 06" longitude (provided in degrees, minutes and seconds), type `geoid`, the NTv2 grid file option and file path, then `--interactive` (or its shortcut `-e`) to indicate that coordinates will be entered via the command line, followed by options and arguments for latitude and longitude:

```
geoid -g ausgeoid09.gsb -e --latitude -33.2436 --longitude 149.3906
```

When complete, **geoid** will report the N value and the north–south and east–west deflections of the vertical ξ and η interpolated for the supplied coordinates. In the case of failure, an error message corresponding to the type of failure will be displayed. Figure 5.5 shows the information reported to the screen upon normal program execution.

```
+ Options:
  Geoid grid file:                 ausgeoid09.gsb
  Interpolation method:            Bi-cubic
  Input coordinate format:         Degrees minutes seconds

+ Opening grid file... done.
+ Interpolation results for -33.2436, 149.3906 (ddd.mmssss):

  N value          =   25.4 metres
  Deflections:
  - Prime meridian =  -4.41 seconds
  - Prime vertical =  -6.48 seconds
```

Figure 5.5: Interactive geoid grid file interpolation

By default, interactive input requires coordinates to be supplied in degrees, minutes and seconds format. To interpolate geoid information using decimal degrees, simply add the `--decimal-degrees` option:

```
geoid -g ausgeoid09.gsb -e --latitude -33.4100 --longitude 149.6517 --decimal-degrees
```

Note that latitudes and longitudes, whether in decimal degrees or degrees, minutes and seconds, must be supplied in HP notation (i.e. dd.dddddddd or dd.mmssssss). Spaces between degrees, minutes and seconds will cause **geoid** to treat values following the spaces as independent option arguments. Since **geoid** does not check or correct latitude values for sign, positive and negative latitude values will be treated as being in the northern and southern hemispheres respectively. Similarly, positive and negative longitude values will be treated as being east or west of the zero meridian (i.e. Greenwich meridian) respectively.

To interpolate geoid information using bilinear interpolation (c.f. §5.3.1), add `--interpolation-method` (or its shortcut `-m`) followed by a zero (`0`).

5.5.2 Text file mode

Text file mode has been designed to provide an efficient means for (1) interpolating geoid information for a list of coordinates and (2) transforming heights from one system to another using simple ASCII text file formats. For these purposes, **geoid** supports Formatted Text files (e.g. `*.dat`, `*.prn`, `*.txt`) and Comma Separated Values (CSV) files (`*.csv`). Appendix B.6 provides the file format specification for these file types.

To interpolate geoid information and transform a file of heights, type `geoid` on the command line, followed by the NTv2 grid file option and file path, then `--text-file` (or its shortcut `-t`) with the input text file path and, if required, options and arguments for interpolation method, coordinate format and conversion direction. For example, to transform a file of heights in a file named `pipeline.txt` from orthometric to ellipsoidal, where the desired interpolation method is bilinear and the input coordinates are in decimal degrees, type:

```
geoid -g ausgeoid09.gsb -t pipeline.txt -m 0 --decimal -r 0
```

Upon transforming a text file of heights, **geoid** will report to the screen the items relevant to the text file transformation, the progress of interpolation and transformation of heights, and any errors or warnings related to the file transformation. Figure 5.6 shows the information reported to the screen upon normal program execution.

```
+ Options:
  Input folder:                  .
  Output folder:                 .
  ASCII file:                    pipeline.txt
  Output file:                   pipeline_out.txt
  Geoid grid file:               ausgeoid09.gsb
  Interpolation method:          Bi-linear
  Input coordinate format:       Decimal degrees

+ ASCII file interpolation mode.

+ Opening grid file... done.
+ Interpolating geoid components... done.
+ Interpolated data for 17 points.
+ Warning: data for 1 point could not be interpolated.
  See pipeline_out.txt for more information.

+ Geoid file interpolation took 0.012s
```

Figure 5.6: **geoid** progress reporting of geoid interpolation and text file transformation

During this process, **geoid** will attempt to determine the file type from the file extension and contents of the input file. Any problems encountered with opening the input file will be reported to the screen. The name of the output file will be the same as the input file name, but with a "`_out`" inserted between the file name and the file extension. During the transformation process, geoid information will be interpolated using the default or user–specified interpolation method, and the height supplied on each line will be transformed according to the transformation direction. The original height, transformed height, interpolated N value and deflections of the vertical ξ and η for each point in the file will be appended to the original record from the input file, which is in turn printed to the output file. Since **geoid** is only concerned with heights, the input latitude and longitude are written directly to the output file without any alteration.

In the case of an error, the input record prefixed by the string '`ERROR (#)`', where '`#`' is a number indicating the type of error, will be printed to the output file. Typical examples of error include instances when the coordinates cannot be read from the input file (possibly due to an incorrect coordinate type or file record format), or if the coordinates lie outside the extents of the geoid grid file. Table 5.3 in §5.7.4 lists the possible range of error codes and their meanings.

5.6 Exporting interpolated information

During the process of interpolating geoid information from a geoid grid file, **geoid** can export this information to a DNA geoid file. Generating a DNA geoid file is only necessary when it is desirable to introduce geoid information into a DynAdjust project via **import** (c.f. §3.2.4). To export interpolated geoid information to a DNA geoid file, simply add the `--export-dna-geo-file` option when executing **geoid**. The following examples show how to export geoid information during the process of (1) introducing geoid information into a project (c.f. §5.4) and (2) interpolating in text file mode (c.f. §5.5.2):

```
geoid skye -g ausgeoid09.gsb --export-dna-geo-file
```

```
geoid -g ausgeoid09.gsb -t pipeline.txt --export-dna-geo-file
```

5.7 Working with NTv2 geoid grid files

5.7.1 Reporting NTv2 geoid grid file metadata

At times, it may be necessary to view a summary of the metadata contained within an NTv2 geoid grid file. NTv2 grid file metadata is contained within an array of block header records, and provides specific information relating to, for example, geoid grid file version, geoid and ellipsoid parameters, grid file extents and grid node interval. To display the header information for a grid file, type `geoid` on the command line, followed by the NTv2 grid file option and file path, then `--summary` (or its shortcut `-u`) :

```
geoid -g ausgeoid09.gsb -u
```

Running this command will produce a summary similar to that which is shown in Figure 5.7.

```
+ Options:
  Geoid grid file:                 ausgeoid09.gsb
  Input coordinate format:         Degrees minutes seconds

Grid properties for "ausgeoid09.gsb":
+ GS_TYPE  = SECONDS
+ VERSION  = 1.0.0.0
+ SYSTEM_F = GDA94
+ SYSTEM_T = AHD_1971
+ MAJOR_F  = 6378137.000
+ MAJOR_T  = 6378137.000
+ MINOR_F  = 6356752.314
+ MINOR_T  = 6356752.314
+ NUM_OREC = 11
+ NUM_SREC = 11
+ NUM_FILE = 1
  + SUBGRID 0:
    SUB_NAME = AUSGEOID
    PARENT   = NONE
    CREATED  = 01012010
    UPDATED  = 01012010
    S_LAT    = -165540.000
    N_LAT    = -28800.000
    E_LONG   = -575940.000
    W_LONG   = -388800.000
    LAT_INC  = 60.000
    LONG_INC = 60.000
    GS_COUNT = 7113600
```

Figure 5.7: Example NTv2 grid file summary

In relation to Figure 5.7, Tables 5.1 and 5.2 describe the NTv2 grid file header fields and the individual sub grid header fields.

Table 5.1: NTv2 grid file overview fields

Field	Description
`NUM_OREC`	Number of header records in the overview block.
`NUM_SREC`	Number of header records in each sub grid block.
`NUM_FILE`	Number of sub grids contained in the geoid grid file. A grid file may contain several sub grids of different densities. The limits of these sub grids will all be contained within the parent grid. If there is only one sub grid, then it is the parent grid.
`GS_TYPE`	The units of the grid nodes.
`VERSION`	The geoid grid file version.
`SYSTEM_F`	The "from" reference ellipsoid (or geodetic datum with well known ellipsoid)
`SYSTEM_T`	The "to" height system or vertical datum.
`MAJOR_F`	The ellipsoid semi–major axis of the "from" system.
`MAJOR_T`	The ellipsoid semi–major axis of the "to" system.
`MINOR_F`	The ellipsoid semi–minor axis of the "from" system.
`MINOR_T`	The ellipsoid semi–minor axis of the "to" system.

Table 5.2: NTv2 grid file sub grid fields

Field	Description
`SUB_NAME`	The name of this particular sub grid.
`PARENT`	The parent sub grid name. '`NONE`' if this is the parent grid.
`CREATED`	Grid file creation date
`UPDATED`	Grid file modification date
`S_LAT`	Lower latitude extent
`N_LAT`	Upper latitude extent
`E_LONG`	Lower longitude extent
`W_LONG`	Upper longitude extent
`LAT_INC`	Latitude interval
`LONG_INC`	Longitude interval
`GS_COUNT`	Grid node count

5.7.2 Importing WINTER DAT geoid grid files

AUSGeoid09 has been publicly released in the form of ASCII text files using the legacy WINTER DAT file format. Whilst serving as a simple, human–readable format, the WINTER DAT file format is not designed to provide an efficient means for instantaneous interpolation. Accordingly, DynAdjust requires the geoid information contained in these files to be converted into a structured binary format. DynAdjust uses the NTv2 format for structuring, storing and interpolating geoid model information.

To convert geoid information in the WINTER DAT file format to the NTv2 binary format, type `geoid` at the command prompt, then `--create-ntv2` (or its shortcut `-c`), followed by the options and arguments relating to the input and output files and the geoid model parameters. For example, to create a new NTv2 binary grid file named `geoid_model.gsb` from geoid information stored in a WINTER DAT file named `geoid_model.dat` using the default options, run the following command:

```
geoid -c -g ./geoid_model.gsb -d geoid_model.dat
```

When converting geoid files to the NTv2 binary format, **geoid** will determine from the raw data the upper and lower grid extents and the north–south and east–west grid node intervals. Checks are also performed to ensure the data contains a sufficient number of nodes to construct an orthogonal grid file. The parameters derived from the raw data will be written to the grid file header records (c.f. Tables 5.1 and 5.2). Since the WINTER DAT file format contains data for a single grid, the resultant NTv2 grid file will contain a single sub–grid which will be the parent grid.

To facilitate the capture of other metadata relating to the geoid grid file, **geoid** provides options to specify the units for the deflections of the vertical (in radians or seconds), the grid file version, the name of the ellipsoid system, the name of the height system or vertical datum, the semi–major and semi–minor ellipsoid parameters of the two systems, the name of the sub–grid, and the dates of creation and update. For example, to specify a version of `2.5.0.0`, the name `MYGEOID`, ellipsoid and height system names `GDA94` and `AHD71`, a creation date of 18 March 2015 and an update date of 5 May 2015, run the following command:

```
geoid -c -g ./geoid_model.gsb -d geoid_model.dat --VERSION 2.5.0.0 --SUB_NAME MYGEOID --SYSTEM_F GDA94
--SYSTEM_T AHD71 --CREATED 18032015 --UPDATED 05052015
```

The option `--grid-shift-type` followed by either `seconds` or `radians` can be used to specify the units in which the grid extents and the deflections of the vertical will be stored. The options `--semi-major-from`, `--semi-major-to`, `--semi-minor-from` and `--semi-minor-to` are not used during grid file interpolation and can be provided for metadata purposes only.

Figure 5.8 shows the information reported to the screen upon normal program execution. Once the NTv2 binary grid file has been created, a summary of the grid file parameters (c.f. Figure 5.7) will be displayed.

```
+ Options:
  Input folder:                 .
  Output folder:                .
  Geoid grid file:              geoid_model.gsb
  WINTER DAT file:              geoid_model.dat

+ Creating NTv2 file from WINTER DAT file format:

+ Reading contents of WINTER DAT file... done.
+ WINTER DAT file structure appears OK.
+ Creating NTv2 gsb file... done.

+ Grid properties for geoid_model.gsb:
  - GS_TYPE  = SECONDS
  - VERSION  = 2.5.0.0
  - SYSTEM_F = GDA94
  - SYSTEM_T = AHD71
  - ...
  - ...
```

Figure 5.8: Progress of NTv2 grid file creation

5.7.3 Exporting NTv2 geoid grid files

geoid provides a capacity to export a binary geoid grid file to ASCII file format, and vice versa. For example, the following command will export a binary NTV2 grid file to its ASCII equivalent. Using this option, the output file name will be generated by appending the `.asc` extension to the supplied binary NTv2 grid file name.

```
geoid -g ./geoid_model.gsb --export-ntv2-asc-file
```

Running this command will produce a file named `geoid_model.gsb.asc`. Exporting to ASCII can be useful in isolating issues relating to geoid interpolation, importing data to other surveying and GIS software packages, or transforming published geoid files in NTv2 format to an alternative format.

5.7.4 Grid file interpolation errors

If an error occurs at any time during the geoid interpolation process, a non-zero value is printed to the output file to indicate the type of error. Table 5.3 lists the errors that may be encountered upon geoid grid file interpolation.

Table 5.3: Grid file interpolation error codes and descriptions

Code	Description
-6	The NTv2 version of the specified grid file is not supported.
-5	The parameters for the specified grid file do not match the number of records.
-4	Could not allocate the required memory.
-3	A grid file has not been opened.
-2	Cannot locate the required sub grid.
1	The specified grid file could not be opened.
2	Cannot read this type of grid file.
3 or 14	Found an unrecoverable error in the specified grid file: `<grid-file-name>` It is likely that this file was downloaded or produced incorrectly.
4	Could not read from the specified input file.
5	Could not write to the specified output file.
6	Cannot read this type of input file.
7	Cannot produce this type of output file.
8	No data was contained within the last input record.
9	The record `<data-record>` is too short to contain valid data.
10	The record `<data-record>` does not contain valid numeric input.
11	The required data could not be extracted from the record `<data-record>`.
12	The specified zone is invalid.
13 or -1	The point `<data-record>` lies outside the extents of the specified grid file.
15	The interpolation shifts could not be retrieved from the ASCII grid file.
16	The interpolation shifts could not be retrieved from the Binary grid file.
17	The csv record `<data-record>` does not contain sufficient records.

Chapter 6

Network segmentation

6.1 Introduction

For relatively small survey control networks, the task of estimating unknown station coordinates and their uncertainties can be readily achieved in a matter of seconds using a modern computer. In these circumstances, network segmentation can be omitted, and least squares adjustments can be undertaken in a conventional, simultaneous fashion (c.f. §8.3.1). When dealing with extremely large networks however, network adjustment becomes computationally intensive and time consuming, taking up to several hours, if not days to complete. This is primarily due to the number of floating point operations that must be undertaken when computing the inverse of the least squares normal equation matrix — an essential step for the full analysis of the precision of the estimated parameters. Invariably, the sheer size of state, national and continental scale geodetic networks (i.e. those which have tens of thousands of stations and hundreds of thousands of measurements) presents a substantial obstacle to performing regular re–adjustment as demanded by datum maintenance. In such circumstances, it becomes more efficient to segment and adjust the network using phased least squares adjustment (c.f. §8.3.2).

DynAdjust has been specifically designed to make the phased adjustment process as simple and efficient as possible. This is achieved by the use of an automated network segmentation algorithm which eliminates the need for the user to manually subdivide a network into blocks. The first part of this chapter (§6.2 – §6.3) describes the concept of network segmentation and the automated segmentation algorithm employed by DynAdjust. The second part describes how to use **segment** and its options to segment networks and to constrain the way in which networks are segmented.

6.2 The concept of network segmentation

The concept of network segmentation involves partitioning a network into smaller–sized blocks. Figure 6.1 illustrates the concept of a geodetic network segmented into two blocks. In Figure 6.1, Block A is defined by the set of measurements $\mathbf{m_a}$ with variance matrix $\mathbf{V_a}$, and Block B is defined by the set of measurements $\mathbf{m_b}$ with variance matrix $\mathbf{V_b}$. The parameters to be estimated in the entire network are represented by $\mathbf{x_p}$, $\mathbf{x_q}$ and $\mathbf{x_r}$, of which $\mathbf{x_p}$ and $\mathbf{x_q}$ appear in Block A, and $\mathbf{x_q}$ and $\mathbf{x_r}$ appear in Block B. Thus, $\mathbf{x_q}$ represents the set of *junction stations* which link Blocks A and B through measurements $\mathbf{m_a}$ and $\mathbf{m_b}$, and $\mathbf{x_p}$ represents the set of *inner stations* relating to Block A. Since all stations represented by $\mathbf{x_q}$ are connected to measurements in Block B, both $\mathbf{x_q}$ and $\mathbf{x_r}$ represent the inner stations relating to Block B. $\mathbf{x_p}$, $\mathbf{x_q}$ and $\mathbf{x_r}$ each represent a cluster of stations which can be of any size. Likewise, $\mathbf{m_a}$ and $\mathbf{m_b}$ each represent clusters of measurements of any size. With this concept in mind, Figure 6.2 shows a trivial GNSS network segmented into two blocks.

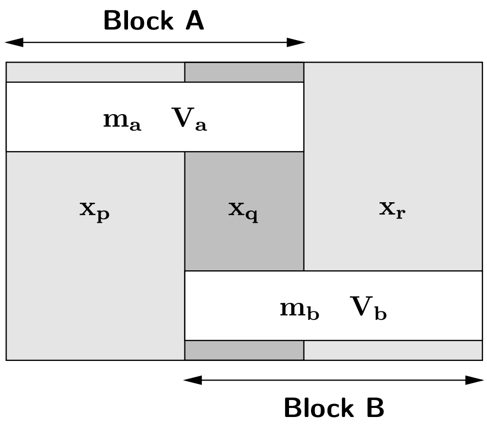


Figure 6.1: Network segmentation concept

(a) First block (Block A)

(b) Second block (Block B)

Figure 6.2: A trivial GNSS network segmented into two blocks

In Figures 6.2a and 6.2b, the stations with a blue circle are inner stations. The stations with a red circle are the junction stations which link the two blocks. With reference to Figure 6.1 and the network shown in Figure 6.2, Table 6.1 summarises the distribution of the stations and measurements amongst the two blocks. An important note worth emphasising is that whilst a particular station can appear as a junction station in both Blocks A *and* B, each measurement in a network must only ever appear once — either in Block A *or* in Block B.

Table 6.1: Distribution of parameters and measurements of a network segmented into two blocks

Block	**Parameters and stations**		**Measurements**
A	$\mathbf{x_p}$	236300210, 309100090, 310211240, 365300060, 409600170, 409704930	$\mathbf{m_a}$
	$\mathbf{x_q}$	335800500, 360100260	
B	$\mathbf{x_q}$	335800500, 360100260	$\mathbf{m_b}$
	$\mathbf{x_r}$	212000820, 230900140, 269100210, 299000080, 409700110	

Although the preceding text has outlined the concept of network segmentation using two blocks, a geodetic network can be segmented into an almost unlimited number of blocks. It follows that the phased adjustment technique can be applied to geodetic networks of almost any number of stations and measurements. The primary factors which define the way in which a geodetic network is segmented are (a) the desired maximum number of stations each block should contain (i.e. the maximum block size), and (b) the level of connectivity in a network (i.e. the number of other stations each station is connected to by the set of measurements). Concerning the latter, networks with few measurements per station can be segmented in a far more flexible manner than highly connected networks, whereas networks having a high level of connectivity cannot always be segmented into the desired number of stations. In any case, to achieve a rigorous solution for the unknown parameters, the network must be correctly segmented so that:

- each block contains the total number of measurements that exist between all inner stations in that block, and
- junction station estimates and variances can be passed between successive blocks.

segment takes care of these concerns using a fully automated procedure, and offers the flexibility for users to nominate (a) the maximum number of stations each block should contain and (b) which stations should appear in the first block. Nominating the maximum block size is essential to optimising network adjustment performance according to network size and computer hardware capability. The ability to nominate which stations should appear in the first block affords the ability to derive the relative uncertainties between any two or more stations in the network. These options will be described in §6.5.1.

6.3 Segmentation algorithm

At the most fundamental level, the segmentation algorithm partitions a large network into a series of blocks with the characteristics of the two adjacent blocks as shown in Figure 6.1. To help conceptualise a network segmented into multiple blocks, Figure 6.3 illustrates the concept of a network segmented into four blocks.

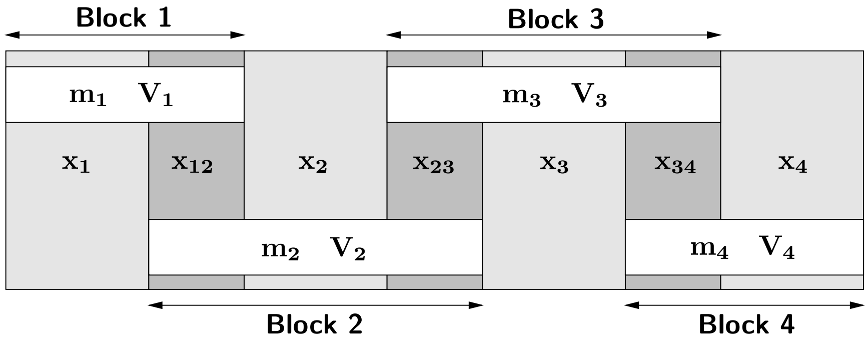


Figure 6.3: Network segmentation concept involving four blocks

In summary, each block is made up of inner stations $(\mathbf{x}_i)$, for which all their associated measurements $(\mathbf{m}_i)$ make up the measurement–set, and junction stations $(\mathbf{x}_{ij,\,j=i+1})$, which are connected by measurements to the inner stations but are also associated with other measurements to stations in other blocks. The segmentation algorithm moves forward (e.g. block i, block $i+1$, ..., block $n-1$, block n) by selecting the inner stations for the next block from the junction stations of the last block.

The total process of selecting stations and measurements for the series of connected blocks can be summarised as:

1. Select the inner stations for the first block.
2. Select all measurements associated with inner stations.
3. Select as junction stations, all stations connected by measurements to the inner stations.
4. The structure of the block is now complete and is made up of inner stations, junction stations and associated measurements.
5. Move to the next block by selecting its inner stations from the previous block's junction stations.
6. Repeat steps 2, 3 and 4 until all measurements have been selected for one of the blocks.

The result of this process is to segment a large network into a series of smaller blocks which can be adjusted sequentially. After each block is adjusted, the estimates of the parameters associated with the junction stations are passed as *pseudo measurements* to the next block for further processing, and the parameters associated with the inner stations drop from the procedure.

The stations selected as inner stations of the first block do not have to be in a close spatial proximity or connected by direct measurement, and may indeed be on opposite sides of a continent. In many applications of DynAdjust, the selection of the inner stations of the first block is critical as it provides a mechanism for placing a particular set of stations in one block. If these are the only stations for which a rigorous estimate and the full variance matrix is required, it makes for great efficiency in adjustment as only one pass through the full series of blocks is required (see §6.4.4). The efficiency of the adjustment process is aided by having a high ratio of inner to junction stations and various strategies are available at step 5 to enhance this (see §6.4.2).

6.4 Accommodating variations in network design, size, user preferences and computer performance

In general, conventional survey control networks used for land development are those which have been developed over several decades with an almost random pattern of extension, reinforcement and re–measurement. They are not structures that are characterised by uniform network design and homogeneous station–to–measurement ratios, such as is found in a regular lattice, a honeycomb pattern, or triangulated terrain model. Rather, survey control networks vary considerably in geometry, station count, measurement count, types of measurements, the number of measurements per station, the number of stations which are connected by a single measurement, and so on. In order to permit phased adjustment to be performed on *any* network, the automated segmentation algorithm has been designed to be extremely flexible so as to handle a virtually innumerable number of network designs and sizes whilst at the same time, permitting the user to influence the way in which a network is segmented. To achieve this flexibility, the segmentation algorithm takes several aspects into consideration, including:

- Optimum block size;
- Inevitable influences on block size;
- Factors influencing the rate of segmentation;
- Datum deficient blocks;
- Non–contiguous networks, and;
- The need for coordinate estimates and full variance matrix for a particular set of stations.

The aspects are explained in the sections that follow.

6.4.1 Optimum block size and maximum adjustment efficiency

Considering that the linear CPU time taken to adjust a network is a factor of n^3, where n is the number of parameters to be estimated, logic would suggest that adopting smaller block sizes would lead to greater phased adjustment efficiency. However, this notion is true only to a certain degree. Experience shows that for each network and computer there is an optimum block size limit, beyond which phased adjustment efficiency decreases. Three primary influences (positive and negative) on the optimum block size are noted:

1. **Smaller than optimum block sizes can lead to a larger number of parameters to estimate.** Two things need to be noted in this context. Firstly, as smaller block sizes lead to larger numbers of blocks to adjust, the total aggregate time to perform a phased adjustment on the set of segmented blocks may be more than the time taken to adjust fewer blocks. Secondly, smaller block sizes increase the likelihood that the same station will appear as a junction station in several blocks. Inevitably, multiple occurrences of a junction station increases the number of total parameters to be estimated in a phased adjustment and thereby, increases the adjustment time. Generally speaking, networks which are partitioned using less than the optimal number of stations per block will take additional CPU time to adjust.
2. **Adjustment times improve with higher performing processor (CPU) architectures and frequencies.** Intuitively, the time taken to adjust a network varies, more or less, according to the CPU frequency of the computer on which an adjustment is run. As faster CPU frequencies lead to faster adjustment times, it follows that the optimum block size will also increase. With the widespread availability of multiple CPUs, hyper–threading and multi–core CPUs[1], adjustment times can be reduced significantly by the use of multi–threaded phased adjustments (c.f. §8.3.2.2). Accordingly, the optimum block size will be different again. In almost every case, the optimum block size varies according to computer capability.
3. **Measurement connectivity**. Networks having a high number of measurements per station will take more time to adjust than networks with lower ratios, particularly if larger than expected blocks are generated (c.f. §6.4.2). Accordingly, the optimum block size for the former case

1. In some cases, CPUs with a large number of cores have a lower frequency per core than those with fewer cores. Hence, a quad–core CPU will not always lead to faster adjustment times than a dual–core.

may be lower than the latter case, despite the two networks having a similar number of stations and (total) measurements.

It is for these reasons that the optimum block size will vary from one project to another, and from one computer to another. To achieve the maximum phased adjustment efficiency for any one network on any given computer, the segmentation algorithm affords a capacity for users to nominate the maximum block size by which the network is segmented.

6.4.2 Inevitable influences on the generated block sizes

Despite careful efforts to determine the optimum block size for segmenting a network, there are some inevitable influences which cause the segmentation algorithm to generate blocks which exceed the user–specified block size. Two primary influences are (1) the amount of station–to–station connectivity created from the set of measurements and (2) the presence of large clusters of correlated GNSS baselines or GNSS points. Left unmanaged, these influences have the potential to create excessively large block sizes.

In the first instance, large numbers of measurements from a single station can cause the algorithm to introduce large numbers of stations into a block. In some cases, the number of stations introduced to a block may exceed the user–specified block size. An example of this scenario is shown in Figure 6.4, whereby the total block size regularly exceeds the Max station limit (650) because many of the stations in this network are connected to large numbers of measurements. Figure 6.5 shows a typical station with a high level of connectivity (MORANG PM 48, associated with over 770 GNSS baseline, distance and direction set measurements). Conversely, Figure 6.6 shows the segmentation results from a spirit levelling network. In this case, all stations in the network are connected to less than five measurements and as such, the total block size closely follows the Max station limit (385).

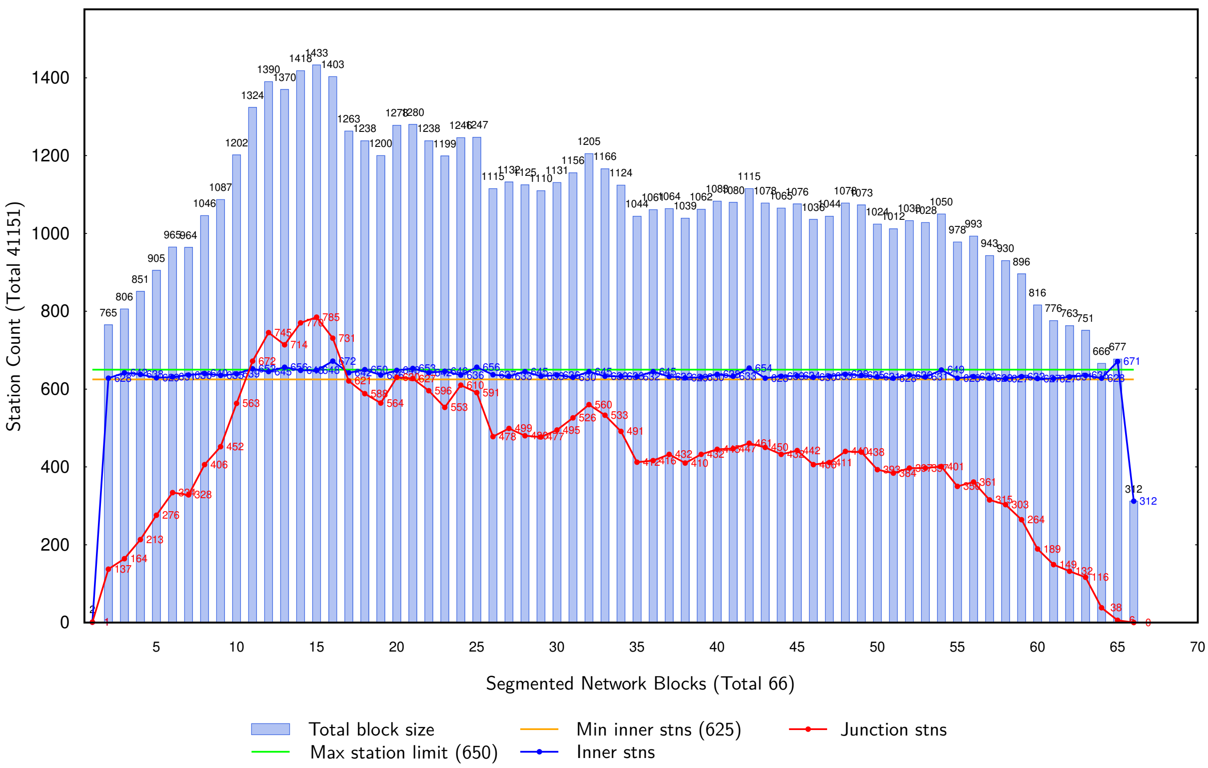


Figure 6.4: Segmentation of a geodetic network having stations with large numbers of measurements.

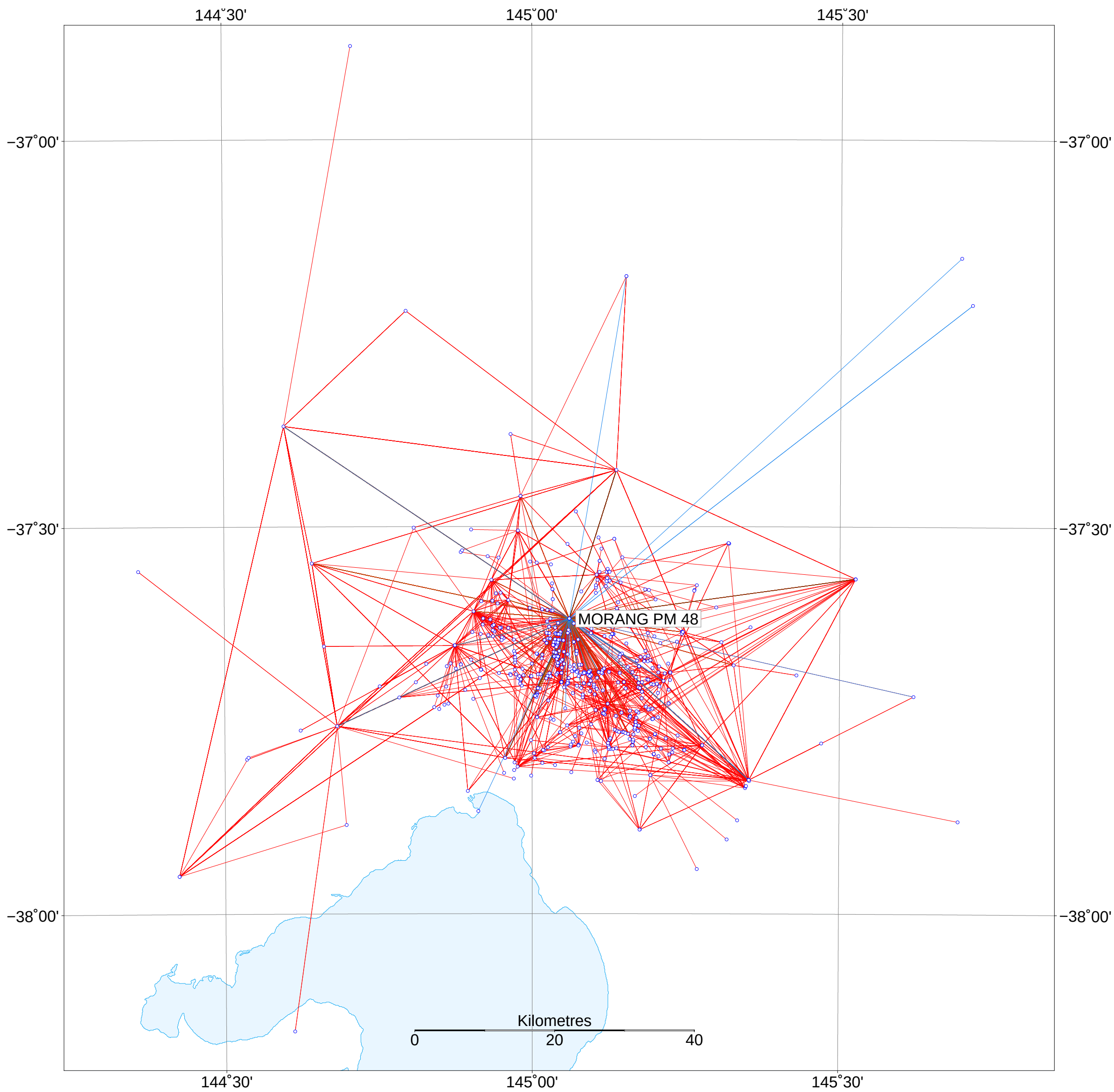


Figure 6.5: GNSS, direction set and distance measurements in the Victorian geodetic network associated with MORANG PM 48.

Secondly, owing to the fact that GNSS clusters are correlated sets of measurements with a fully populated variance matrix, each GNSS cluster is introduced to a block as a single group. Hence, GNSS clusters cannot be "split" and spread across multiple blocks without compromising the rigour of the adjustment. When the segmentation algorithm adds a large GNSS baseline or point cluster to a block (such as a national GNSS CORS network solution), it is not unusual for the aggregate number of stations introduced by the cluster to exceed the user–specified block size.

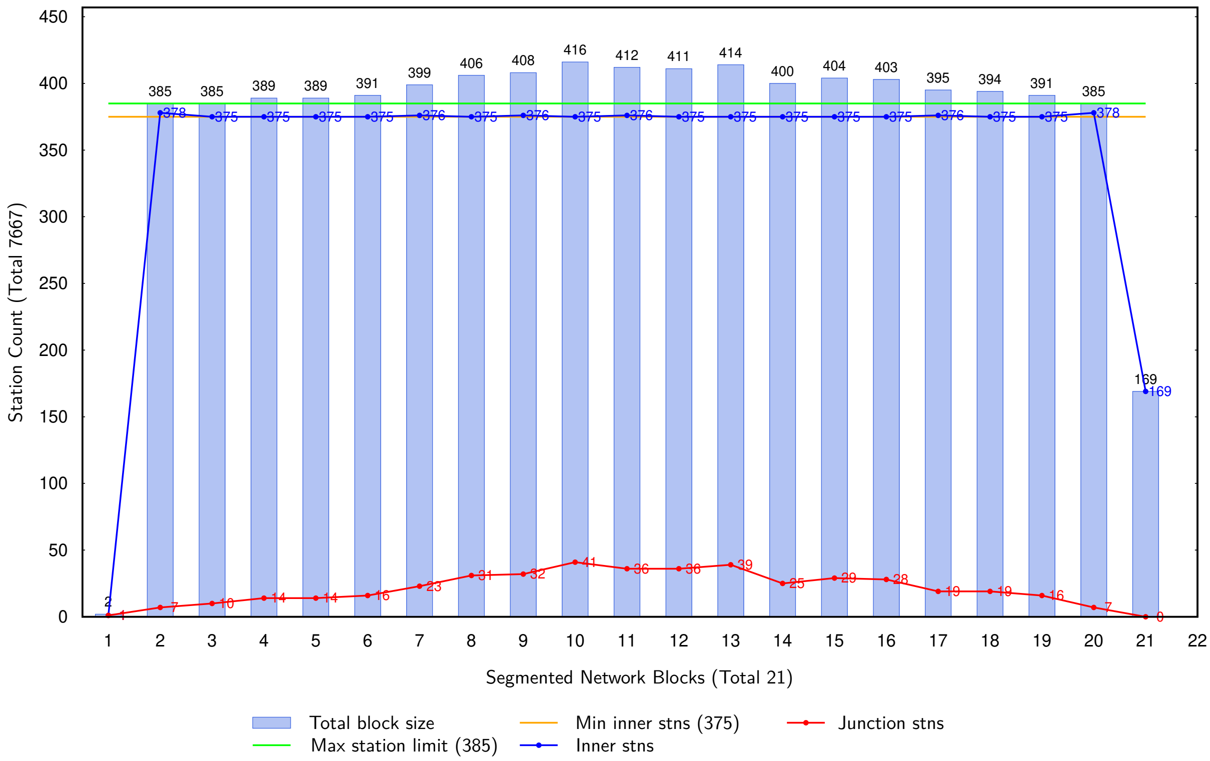


Figure 6.6: Segmentation of a spirit levelling network having stations with a low measurement count.

Although these influences are for the most part inevitable, the segmentation algorithm includes a strategy to minimise the influence of stations having excessive numbers of measurements, and those which form part of a large correlated cluster of measurements. When forming new blocks, the algorithm introduces stations in order by taking the next available station which has the least number of associated measurements. By delaying the introduction of large groups of measurements in this way, the segmentation algorithm has a greater opportunity to advance through the network at rapid rates. In addition, the segmentation algorithm examines each junction station prior to its introduction to verify whether its associated measurements will result in an excessively large block size. In this case, the algorithm advances to alternative junction stations which can maximise the rate of segmentation and thereby, minimise the resulting block size.

6.4.3 Factors influencing the segmentation rate

When the segmentation algorithm is forced to introduce a large set of measurements to a block, which in turn results in the insertion of a large number of junction stations to that block, the rate at which the segmentation advances through the network will slow down considerably if there is little flexibility in permitting measurements associated with those junction stations from being added. Whilst this is somewhat counter–intuitive to maintaining block size thresholds, failure to provide flexibility in this regard will result in the creation of an excessive number of blocks due to the way in which junction stations are carried from one block to the next.

For example, if a block is created with 400 inner stations and 200 junction stations, the segmentation algorithm will commence building the next block with 200 junction stations. That *all* junction stations are carried to the next block is an essential criteria for rigorous phased adjustment. However, if a block size limit of 200 stations has been imposed, a dead–end situation (or stalemate) immediately arises if there is no flexibility to permit the introduction of new measurements and in turn, more stations. To circumvent the algorithm from reaching stalemate, the segmentation algorithm affords a capacity for the user to specify the minimum number of inner stations that each block must contain.

6.4.4 Generating coordinate estimates and full variance matrix for a user–defined set of stations

At different times and for various reasons, the need for the coordinates and full variance matrix for a particular set of stations in a geodetic network will arise — whether for those that surround a certain project, or those which lie within or along the extent a defined area. Whilst this may seem to be a rather trivial objective to fulfil, three obstacles to achieving this objective must be borne in mind. Firstly, it is highly unlikely that the full variance matrix for extremely large networks (i.e. state, national or continental scale networks of which a small portion survey control is needed) has been produced and archived in an easily accessible form. Secondly, some or all of the stations selected for a project may have never been directly connected by measurement. Thirdly, the only plausible way to obtain a rigorous variance matrix for the defined set of stations is to incorporate the contribution from all measurements in the network.

Recognising the reality of these issues, the segmentation algorithm has been designed to offer a capacity to meet the requirement for producing, at will, coordinate estimates and the full variance matrix for randomly selected stations. This is achieved by permitting users to nominate the stations which must appear together in a single block. Upon running a phased adjustment of the entire network with these stations all appearing in the last block to be adjusted, rigorous coordinates and the full variance matrix will be available. If the latest, most rigorous station coordinates have been used as the a–priori estimates, a single pass in Block–1 only mode (c.f. §8.3.2.3) is all that will be required to generate a rigorous variance matrix.

6.4.5 Datum deficient blocks

During segmentation, it is quite common for one or more segmented blocks to be generated without a sufficient number of measurements to connect the inner and/or junction stations to datum. In addition, it is equally common for a block to not contain a sufficient number of measurements to solve all parameter estimates. This particular case arises when the user has nominated a specific set of stations to be contained in the first block and those stations are not connected by the associated measurements. The absence of a sufficient number of measurements is a characteristic feature of many blocks generated by the segmentation algorithm. The consequential outcome of forming blocks with insufficient measurements is that the normal equation matrix will be singular and therefore, non–invertible. This is despite the fact that the coordinate estimates and uncertainties for the junction stations are carried forward from the previous block.

DynAdjust accommodates for the problem of singularity during phased adjustment using the following approach. Prior to running a phased adjustment, all stations which have an insufficient number of measurements or connections to datum are identified and considered to be known with a (default) variance matrix of 10 metres^2. This ensures the non–singularity of the normal matrix for each block and that the solution can be taken. Thus, in the early stages of the sequence of phased adjustments, the variances of the adjusted coordinates will be of the order of 10 metres^2 if no connection to datum has yet been found in any of the blocks so far adjusted. Eventually however, the influence of datum will arise through the remaining measurements and the variance of the estimated coordinates will reduce to the appropriate values.

6.4.6 Non–contiguous networks

Experience with several large geodetic networks developed over long periods of time shows that, despite the best intentions of those who have managed them, geodetic networks are not always contiguous. That is, geodetic networks inevitably contain isolated networks comprised of stations and measurements which are not connected to other parts of the network. At times, these isolated networks are not connected to datum. DynAdjust has been developed to handle multiple isolated networks in a contiguous fashion and to deal with the associated implications during segmentation and phased adjustment.

6.5 Segmenting a network

Once **import** has created the binary station (`.bst`) and measurement (`.bms`) files, and the associated station (`.asl`) and measurement (`.aml`) lists from the input station and measurement files, the network is ready for segmentation. To segment a network using all default settings, type `segment` on the command line followed by the network name:

```
segment uni_sqr
```

Upon running this command, a segmentation output file (`uni_sqr.seg`) will be created. This file contains the station and measurement indices for the respective blocks. Appendix C.5 describes the structure and format of the segmentation output file. The content of the segmentation output file is structured in a way to provide for the efficient loading of network segmentation information. In particular, **adjust** relies on the formatting of this file to obtain metrics relating to block count, individual block sizes, and block station and measurement counts. These values are needed for the creation and management of several lists and matrices used in the phased adjustment process. As this file is automatically generated by **segment**, any changes to this file will be lost if **segment** is executed after those changes have been made.

If a DynAdjust Project File (c.f. Chapter 2) is to be used, type `dynadjust` on the command line, then `--project-file` (or its shortcut `-p`) followed by the full path for the project file, and then the option to invoke **segment**:

```
dynadjust -p uni_sqr.dnaproj --segment
```

No other options are required, as **segment** will be invoked using the options and arguments contained in the project file `uni_sqr.dnaproj`.

When complete, **segment** will report to the screen a number of items relevant to the project (e.g. name, location of input and output files); user–specified options; the progress of network segmentation; brief statistical summary of the formed blocks; and final exit status (i.e. success or failure). Using the `uni_sqr` project, Figure 6.7 shows the information reported to the screen upon program execution.

Note that each time a project has been created using **import**, **segment** must be executed before **adjust** can be called to perform phased adjustment. The importance of this program execution sequence (c.f. Figure 1.3) is illustrated as follows. Consider a situation where a network project has been created using **import**, and then **segment** has been executed to divide the network into several blocks. If at a later stage new measurements and/or stations are added to the original station and measurement files and **import** is again executed to reload the network, the BST, BMS, ASL and

AML elements and the relationships between them will be different to what was previously created. Consequently, the original segmentation file (`.seg`) will be deficient, failing to take into account the new measurements and/or stations. For this reason, **adjust** performs a test on the SEG file to ensure that it has been created subsequent to the creation of the BST, BMS, ASL and AML files.

```
+ Options:
  Network name:                    uni_sqr
  Input folder:                    .
  Output folder:                   .
  Associated station file:         ./uni_sqr.asl
  Associated measurement file:     ./uni_sqr.aml
  Binary station file:             ./uni_sqr.bst
  Binary measurement file:         ./uni_sqr.bms
  Segmentation output file:        ./uni_sqr.seg
  Minimum inner stations:          5
  Block size threshold:            65
  No initial station specified. The first station will be used.

+ Loading binary files... done.
+ Adopting 1018 as the initial station in the first block.
+ Creating blocks... done.
+ Segmentation statistics:

  No. blocks          Max size   Min size   Average    Total size
  ------------------------------------------------------------------------
  4                   65         3          45.75      183

+ Verifying station connections... done.
+ Printing blocks to uni_sqr.seg... done.

+ Network segmentation took 0.001s
+ uni_sqr is now ready for sequential phased adjustment.
```

Figure 6.7: **segment** progress reporting

6.5.1 Configuring segmentation behaviour

Recognising the need to accommodate variations in network size, station–to–measurement connectivity and computer performance, and to produce coordinate estimates and their variance matrix for a particular set of stations, the following sections describe how to configure the behaviour of **segment** to achieve the optimal phased adjustment outcome.

6.5.1.1 Specifying stations to appear in the first block

To specify the stations that should be contained in the first block, provide the list of stations in a text file and/or via the command line interface. To specify the stations using a text file, add the option `--net-file`:

```
segment uni_sqr --net-file
```

In this case, **segment** will look for a file named `uni_sqr.net`. This file must be formatted as a `.net` file. To specify starting stations on the command line, add the option `--starting-stns` followed by a comma delimited string of station names. For example, to ensure stations 2202 and 2203 are contained in the first block, execute the following (remembering that station names with spaces will require double quotes, c.f. §1.3.2):

```
segment uni_sqr --starting-stns 2202,2203
```

Both `--net-file` and `--starting-stns` options can be used at the same time.

6.5.1.2 Achieving optimum block sizes

To achieve the optimum segmentation block size, add the option `--max-block-stns` followed by the block size threshold (i.e. the number of stations by which to limit the block sizes). For example, to specify a block size threshold of 45, execute the following:

```
segment uni_sqr --starting-stns 2202,2203 --max-block-stns 45
```

Upon running this command, seven blocks will be created. Figure 6.8 shows the segmentation summary from `uni_sqr.seg`.

```
SEGMENTATION SUMMARY

No. blocks produced              7
-----------------------------------------------------------------------------------------
  Block       Network ID    Junction stns   Inner stns      Measurements    Total stns
  1           0             14              2               90              16
  2           0             17              28              187             45
  3           0             23              23              152             46
  4           0             27              18              120             45
  5           0             25              20              119             45
  6           0             8               37              183             45
  7           0             0               21              243             21
-----------------------------------------------------------------------------------------
```

Figure 6.8: Segmentation summary for `uni_sqr` with a block threshold of 45

As shown in Figure 6.8, each row summarises the station and measurement counts for a block. The block number is given in the first column. If the option to form separate blocks for isolated networks has been provided (c.f. §6.5.1.3), the values in the `Network ID` column will increment. Otherwise, a 0 (zero) will be given for all blocks. Based on the above command, the first block will include stations `2202` and `2203`, and all associated measurements and stations connected to those measurements. Due to the connectivity of the network, fourteen junction stations were introduced to the first block — this is an inevitable result of the connectivity of the network as was explained in §6.4.2.

To increase the number of inner stations per block, add the option `--min-inner-stns` followed by the minimum number of inner stations to be included in each block. For example, to specify a block size threshold of 45 with a minimum inner station count of 35, execute the following:

```
segment uni_sqr --starting-stns 2202,2203 --max-block-stns 45 --min-inner-stns 35
```

Upon running this command, six blocks will be created. Figure 6.9 shows the revised segmentation summary. Note that each block (except the first) now has at least 35 inner stations. There is also a decrease in the total number of blocks, which, in turn, has resulted in a decrease in the total number of parameters to be estimated (c.f. `Total stns`).

Whilst increasing the minimum number of inner stations from 5 to 35 yielded negligible improvements for the uni_sqr network, increasing the minimum inner station count can offer substantial performance gains on large networks. Hence, the user should examine the output from a range of alternative block size threshold and minimum inner station values to achieve the optimum block size.

```
SEGMENTATION SUMMARY

No. blocks produced                5
------------------------------------------------------------------------------------------
  Block       Network ID     Junction stns    Inner stns       Measurements    Total stns
  1           0              14               2                90              16
  2           0              19               37               248             56
  3           0              30               35               249             65
  4           0              17               38               179             55
  5           0              0                37               328             37
------------------------------------------------------------------------------------------
```

Figure 6.9: Segmentation summary for uni_sqr with a block threshold of 45 and minimum inner station count of 35

To assist with this analysis, **segment** provides an option to specify the segmentation output file name, so that various segmentation options and their results can be stored in different files. By maintaining several segmentation output files, different phased adjustments can be run by providing each segmentation file to the input segmentation file option in **adjust**. To specify the output segmentation file, simply add the option --seg-file followed by the desired file name to the list of **segment** options. For example, to force the segmentation output of two different segmentation runs to uni_sqr_test1.seg and uni_sqr_test2.seg, run the following command:

```
segment uni_sqr --max-block-stns 20 --min-inner-stns 20 --seg-file uni_sqr_test1.seg
segment uni_sqr --max-block-stns 80 --min-inner-stns 80 --seg-file uni_sqr_test2.seg
```

6.5.1.3 Isolated networks

As discussed in §6.4.6, DynAdjust automatically caters for isolated networks and provides a capacity to estimate parameters for all stations in a network, despite the fact that the network may not be contiguous. In this context, **segment** permits the option to integrate isolated networks with other parts of the network, or to keep those isolated networks as individual blocks. The default option is the former. To form individual blocks for isolated networks, provide a 0 (zero) to the --contiguous-blocks option as follows:

```
segment uni_sqr --contiguous-blocks 0
```

If there are isolated networks, the values in the Network ID column in the SEGMENTATION SUMMARY section of the segmentation output file (e.g. Figure 6.9 — discussed in more detail in Appendix C.5) will increment.

The option to separate isolated networks is particularly useful when attempting to identify which stations and measurements are disconnected from the rest of the network. To achieve this outcome, run the following sequence of commands:

```
import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml
segment uni_sqr --contiguous-blocks 0
import -n uni_sqr_block_# uni_sqrstn.xml uni_sqrmsr.xml --seg-file uni_sqr.seg --import-block-stn-msr #
  --export-xml-files
```

According to this sequence of commands, the first and second commands will import and segment the entire network into blocks, separating isolated networks (if any) into individual blocks. If there are any stations and measurements which are disconnected from the rest of the network, the values in the `Network ID` column will increment. In this circumstance, record the block number(s) for every change in the `Network ID` column. If the call to **segment** above causes an isolated network to be split into two or more blocks, increase the maximum block size until a unique `Network ID` value appears on a single row and hence, is associated with a unique number in the `Block` column. Using the block number of the isolated network, the second import command with the `--seg-file` option (using the segmentation output file generated from the previous command), and the `--import-block-stn-msr` option (where `#` equals the isolated block number) will export the isolated network to a set of DynaML files named `uni_sqr_block_#stn.xml` and `uni_sqr_block_#msr.xml`.

Chapter 7

Mathematical models for dynamic network adjustment

7.1 Introduction

The purpose of this chapter is to describe the functional models implemented within DynAdjust for the formulation of the least squares normal equations. Since proper testing of measurements and estimated results is of fundamental importance to the verification of least squares adjustments, a summary of relevant probability distributions is provided. The stochastic models for scaling a–priori variance matrices and expressing quality and reliability are also described.

7.2 Observation equations

The measurement types handled by DynAdjust are listed in Table 3.2 on page 19. DynAdjust recognises that certain measurements may be aligned with the direction of the plumb–line rather than the ellipsoid normal, and that certain measurements may be related to the astronomic meridian rather than local meridian. In addition, measurements may be related to a different ellipsoid, reference frame and/or epoch than that chosen for the adjustment. In any case, a correction will need to be made prior to adjustment to ensure all measurements are aligned with the adopted datum. To this end, DynAdjust makes the following assumptions and pre–adjustment corrections upon loading each of the supported measurement types:

- Horizontal angles (A), directions (D), astronomic latitude (I) and longitude (J), astronomic azimuths (K), zenith distances (V) and vertical angles (Z) have not been corrected for deflection of the vertical. Hence, corrections for the deflection of the vertical are applied using the east–west and north–south deflection components contained within the station file. The algorithms used to cater for the deflection of the vertical will be developed in related subsections.
- Geodetic azimuths (B) relate to the local meridian and are thereby not subject to the deflection of the vertical. Since all observation equations have been developed in the cartesian reference frame using vector geometry rather than on the surface of the ellipsoid, skew normal corrections are not required for azimuths.
- Ellipsoid chord (C) and arc (E) distances, MSL arc distances (M), slope distances (S), zenith distances (V) and vertical angles (Z) have already been corrected for coefficient of refraction.
- GNSS baselines (G) and baseline clusters (X), GNSS point clusters (Y), geodetic latitude (P) and longitude (Q) and ellipsoid height (R) are aligned with the adjustment ellipsoid, reference frame and epoch. If a different ellipsoid, reference frame and/or epoch has been provided with such measurements, DynAdjust applies a transformation (as per §4.3) to align each measurement to the adjustment reference frame.

7.2.1 Slope distances (S)

Figure 7.1 shows a typical slope (or direct) distance measurement s_{12} in the local reference frame from p_1 to p_2.

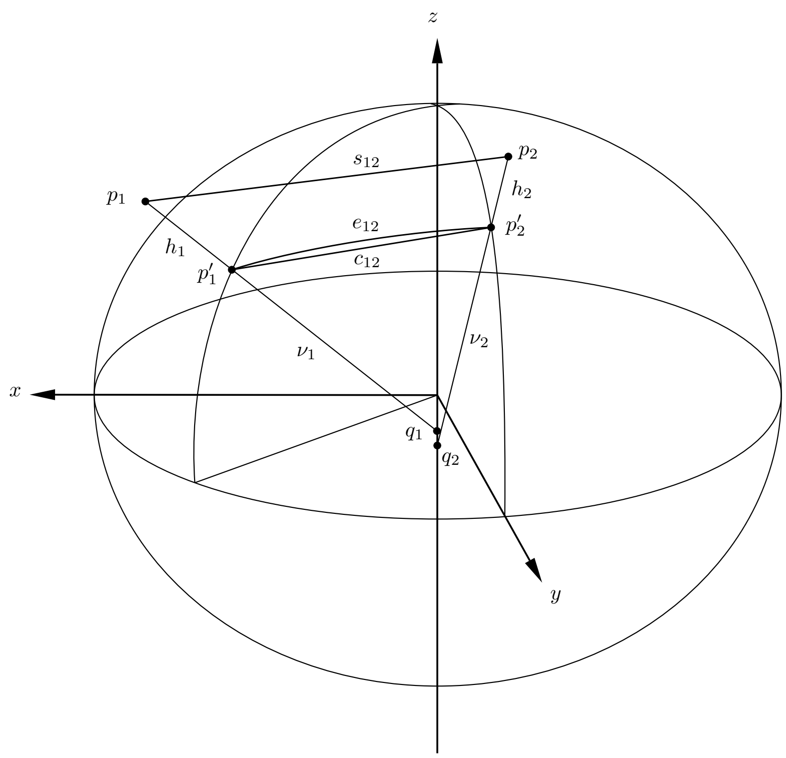


Figure 7.1: Slope, ellipsoid arc and ellipsoid chord distances

The slope distance measurement s_{12} is related to the coordinates of p_1 and p_2 in the cartesian reference frame by:

$$s_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{7.1}$$

Slope distances are observed either between instrument and target or directly between stations. Accordingly, the cartesian elements in 7.1 reflect the coordinates of the instrument and target or stations p_1 and p_2. The coordinates relating to the former are derived from supplied instrument and target heights. Direct distances between stations are dealt with by supplying a zero height for the instrument and target.

Note that for all slope distances, it is assumed that first and second velocity corrections have already been applied to the observed distance.

7.2.2 Ellipsoid arc (E) and ellipsoid chord (C) distances

Figure 7.1 shows typical ellipsoid arc and ellipsoid chord distance measurements e_{12} and c_{12} from p_1' to p_2'. The formation of observation equations for ellipsoid arc measurements is handled by reducing each measurement to its chord counterpart first. An ellipsoid arc e_{12} is related to the corresponding chord c_{12} by:

$$c_{12} = 2r \sin\left(\frac{e_{12}}{2r}\right) \tag{7.2}$$

where r is the radius of curvature of the ellipsoid in the direction of the chord, given by:

$$r = \frac{\rho\nu}{\nu\cos^2\theta_{12} + \rho\sin^2\theta_{12}} \tag{7.3}$$

and θ_{12} is the bearing from p'_1 to p'_2. Therefore, using equation 7.1, the ellipsoid chord distance measurement c_{12} can be related to the coordinates of p'_1 and p'_2 in the cartesian reference frame by:

$$c_{12} = \sqrt{(x'_2 - x'_1)^2 + (y'_2 - y'_1)^2 + (z'_2 - z'_1)^2} \tag{7.4}$$

The coordinates of points p'_1 and p'_2 in the cartesian reference frame are obtained as follows:

$$for\ i = 1, 2$$

$$x'_i = x_i \frac{\nu_i}{\nu_i + h_i} \tag{7.5}$$

$$y'_i = y_i \frac{\nu_i}{\nu_i + h_i} \tag{7.6}$$

$$z'_i = (z_i + {}_qz_i)\frac{\nu_i}{\nu_i + h_i} - {}_qz_i \tag{7.7}$$

where ${}_qz_i$ is the z coordinate element of the point q_i evaluated from equation 4.6, ν_i is the radius of curvature in the prime vertical at p_i, and x_i, y_i and z_i are the cartesian coordinates of p_i.

7.2.3 Mean sea level (MSL) arc distances (M)

Figure 7.2 shows a typical MSL arc distance measurement m_{12} and the corresponding MSL chord mc_{12} from p_1 to p_2. As indicated by Figure 7.2, MSL arcs are assumed to be coincident with the geoid, where $N_{1,2}$ are the geoid–ellipsoid separations at $p_{1,2}$ respectively.

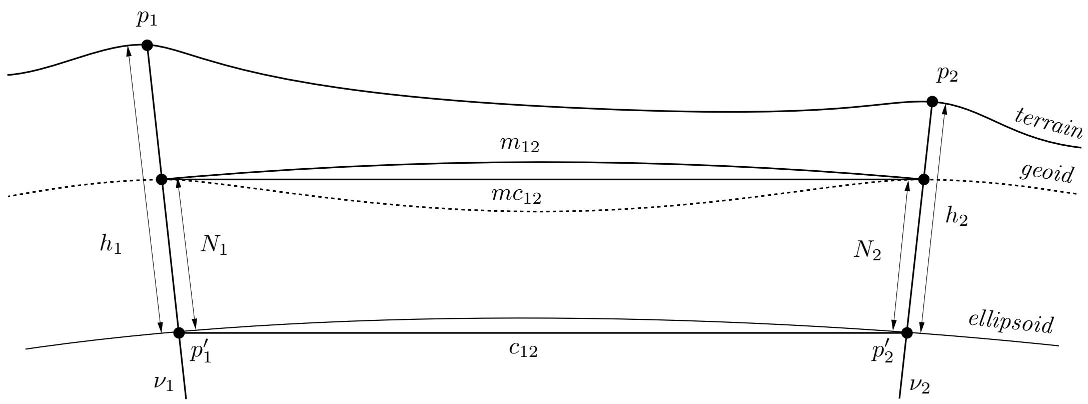


Figure 7.2: Mean sea level arc distances

Similar to the procedure for handling ellipsoid arcs, the task of forming observation equations for MSL arc distances is handled by firstly reducing each MSL arc m_{12} to its chord counterpart mc_{12} and secondly reducing the MSL chord to an ellipsoid chord distance c_{12}. A MSL arc m_{12} is related to the corresponding MSL chord mc_{12} using equation 7.2, repeated here as:

$$mc_{12} = 2r\sin\left(\frac{m_{12}}{2r}\right) \tag{7.8}$$

Here, r is the radius of curvature at MSL and is approximated by:

$$r = \sqrt{\rho\nu} + \left(\frac{N_1 + N_2}{2}\right) \tag{7.9}$$

Then, the ellipsoid chord c_{12} is obtained from:

$$c_{12} = \sqrt{\frac{mc_{12}^2 - (N_2 - N_1)^2}{\left(1 + \frac{N_1}{R}\right)\left(1 + \frac{N_2}{R}\right)}} \tag{7.10}$$

where R is the average radius of curvature approximated from the two principle radii of curvature:

$$R \doteqdot \sqrt{\rho\nu} \tag{7.11}$$

As explained in §7.2.2, the ellipsoid chord distance measurement c_{12} can be related to the coordinates of p_1' and p_2' in the cartesian reference frame using equation 7.4.

For the sake of clarity, Figure 7.3 is provided to show the relationships between slope distance, MSL arc distance, ellipsoid arc distance and ellipsoid chord distance measurements.

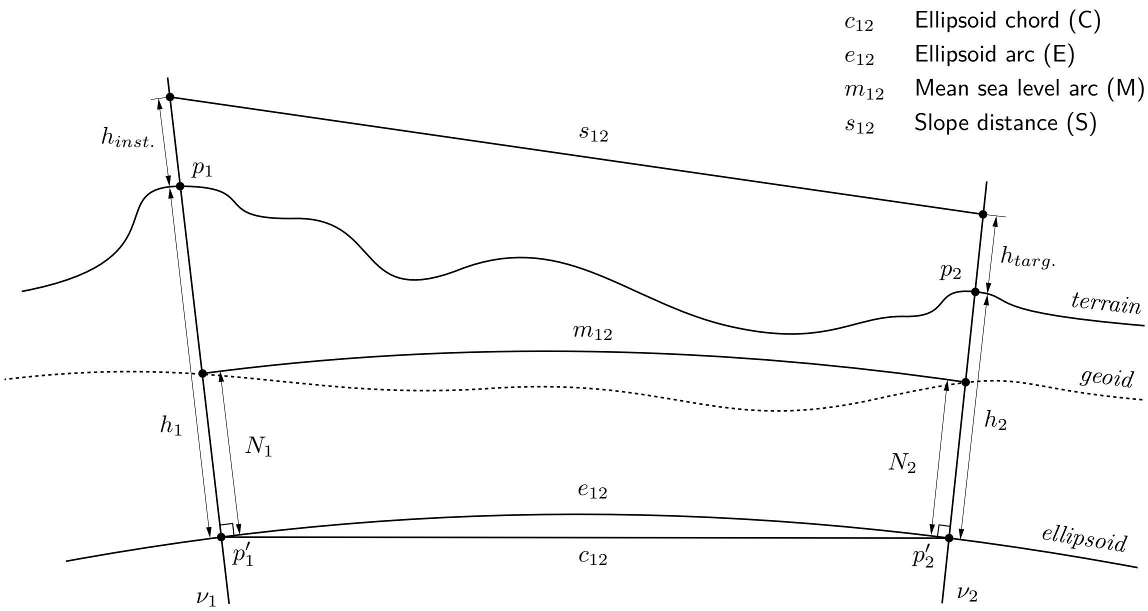


Figure 7.3: Relationships between slope distance, MSL arc distance, ellipsoid arc distance and ellipsoid chord distance measurements

7.2.4 Ellipsoid heights (R) and height differences

Figure 7.4 illustrates typical height measurements h_1 and h_2 observed above the ellipsoid at p_1 and p_2.

An ellipsoid height measurement h_1 is related to the coordinates of p_1 in the cartesian reference frame by:

$$\begin{aligned} h_1 &= (q_1 p_1 - \nu_1) \\ &= \sqrt{x_1^2 + y_1^2 + (z_1 + {}_q z_1)^2} - \nu_1 \end{aligned} \tag{7.12}$$

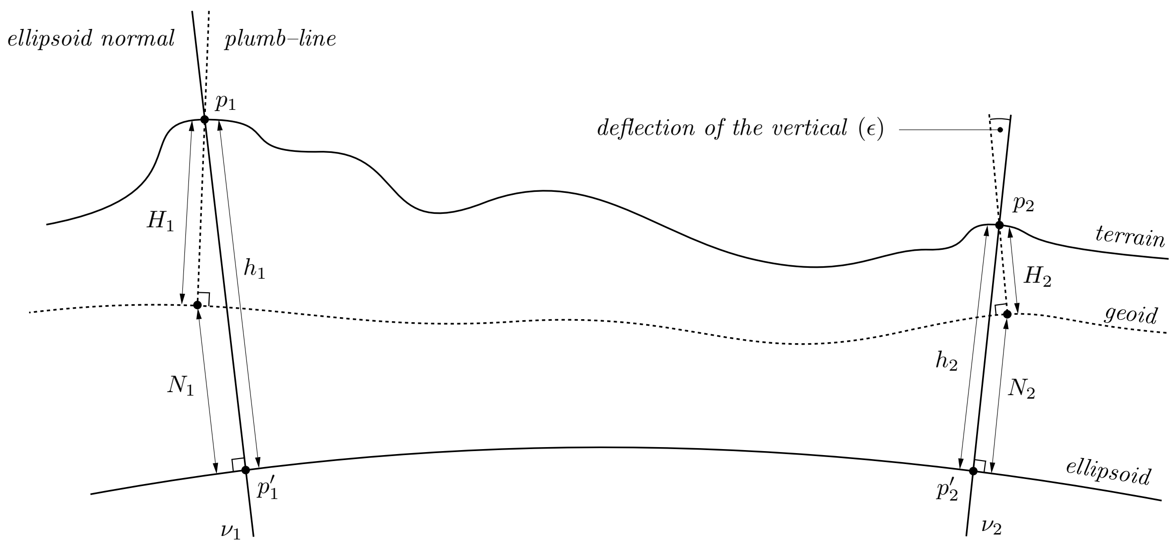


Figure 7.4: Heights and height differences

where q_1 is the intersection of the normal at p_1 with the polar axis of the ellipsoid (see Figure 7.1), ${}_qz_1$ is the z coordinate element of the point q_1 evaluated from equation 4.6, and ν_1 is the radius of curvature in the prime vertical at p_1. Hence, q_1p_1 is the total distance from q_1 to p_1.

An ellipsoid height difference Δh_{12} is related to the coordinates of p_1 and p_2 in the cartesian reference frame by:

$$\begin{aligned}\Delta h_{12} &= h_2 - h_1 \\ &= (q_2p_2 - \nu_2) - (q_1p_1 - \nu_1) \\ &= \sqrt{x_2^2 + y_2^2 + (z_2 + {}_qz_2)^2} - \sqrt{x_1^2 + y_1^2 + (z_1 + {}_qz_1)^2} - \nu_2 + \nu_1\end{aligned} \tag{7.13}$$

Here, q_2, ${}_qz_2$ and ν_2 relate to p_2 as per p_1 in equation 7.12.

7.2.5 Orthometric heights (H) and height differences (L)

Figure 7.4 illustrates typical orthometric height measurements H_1 and H_2 observed above the geoid at p_1 and p_2. The development of observation equations for orthometric height measurements is handled by reducing them to their ellipsoid counterparts first. With respect to Figure 7.4, an orthometric height measurement H_1 is related to the corresponding ellipsoid height h_1 by:

$$h_1 = H_1 \cos\epsilon_1 + N_1 \tag{7.14}$$

where ϵ_1 is the deflection in the direction of gravity from the ellipsoid normal (or deflection of the vertical) at p_1. However, since $\cos\epsilon$ always very close to unity, the correction to h_1 is generally at the $1/100$ millimetre level and can be safely ignored. Accordingly, h_1 can be derived rigorously by:

$$h_1 = H_1 + N_1 \tag{7.15}$$

Thus, h_1 can be related to the coordinates of p_1 using equation 7.12.

A height difference ΔH_{12} observed by spirit levelling can be reduced to an ellipsoidal difference by:

$$\Delta h_{12} = \Delta H_{12} + (N_2 - N_1) \tag{7.16}$$

Note that the observation equations developed in this section assume the geoid to be coincident with the orthometric height datum to which spirit levelling observations (i.e. H_1, H_2 and ΔH_{12}) refer. In practice, however, orthometric height datums derived from spirit levelling observations[1] are rarely coincident with the geoid, especially those developed over large regions. To cater for the separation between an established orthometric height datum and the geoid, separations N_1 and N_2 in equations 7.15 and 7.16 should incorporate both a geoid–ellipsoid separation component and a geoid–ortho separation component.

7.2.6 Cartesian coordinates and GNSS point clusters (Y)

For direct cartesian coordinate measurements, derived either by GNSS or resulting from a previously adjusted network, the relationship between the measured coordinates x_m, y_m and z_m and the coordinates x_1, y_1 and z_1 of p_1 is:

$$\begin{aligned} x_m &= x_1 \\ y_m &= y_1 \\ z_m &= z_1 \end{aligned} \tag{7.17}$$

If GNSS point cluster measurements are provided in terms of latitude ϕ, longitude λ and orthometric height H, the equivalent cartesian coordinates x, y and z are obtained by transformation using equation 4.2, repeated here as:

$$\begin{aligned} x &= (\nu + h)\cos\phi\cos\lambda \\ y &= (\nu + h)\cos\phi\sin\lambda \\ z &= \left[\nu(1 - e^2) + h\right]\sin\phi \end{aligned} \tag{7.18}$$

where h has been derived from H using equation 7.15. In these instances, the point cluster variance matrix $\mathbf{V}_{\mathbf{x}_g}$ (supplied in units of $radians^2_\phi$, $radians^2_\lambda$ and $metres^2_H$) is propagated into the cartesian reference frame using equation 4.31.

When multiple GNSS points are processed simultaneously, correlations will exist between all the points in the cluster. Similarly, correlations will exist for coordinates resulting from network adjustment. In such cases where a fully correlated variance matrix is available, DynAdjust takes these correlations into account within the a–priori variance matrix so as to rigorously propagate the quality of the measurement set into the solution.

7.2.7 GNSS baselines (G) and GNSS baseline clusters (X)

A direct cartesian vector (or GNSS baseline) measurement $[\Delta x_{12}\,\Delta y_{12}\,\Delta z_{12}]$ is related to the coordinates of p_1 and p_2 in the cartesian reference frame by:

$$\begin{aligned} \Delta x_{12} &= x_2 - x_1 \\ \Delta y_{12} &= y_2 - y_1 \\ \Delta z_{12} &= z_2 - z_1 \end{aligned} \tag{7.19}$$

As with a cluster of simultaneously processed GNSS points, DynAdjust takes into account the correlations between simultaneously processed GNSS vectors when forming the a–priori variance matrix.

1. For example, the Australian Height Datum of 1971 (AHD71) and the North American Vertical Datum of 1988 (NAVD88).

7.2.8 2D position via geodetic latitude (P) and longitude (Q)

§7.2.6 provides a rigorous way for dealing with a full set of geographic coordinates (ϕ, λ and H). In practice however, it is often the case that the full set of geographical coordinates for a single point is not available. This is due primarily to the way in which the horizontal and vertical components of geodetic networks have been maintained as separate networks. On some occasions, only latitude and longitude will be available for a single station within a horizontal network. On other occasions only height will be available where the station forms part of the vertical network. For these circumstances, the observation equations for the estimates of the horizontal and vertical components are developed as follows.

If Δx, Δy and Δz represent small changes in the cartesian coordinates of a station due to small changes $\Delta\phi$, $\Delta\lambda$ and Δh in the geographical coordinates, which are represented as:

$$\mathbf{x}_c = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \qquad \mathbf{x}_g = \begin{bmatrix} \Delta\phi \\ \Delta\lambda \\ \Delta h \end{bmatrix} \tag{7.20}$$

then:

$$\mathbf{x}_c = \mathbf{T}\mathbf{x}_g \tag{7.21}$$

$$\mathbf{x}_g = \mathbf{T}^{-1}\mathbf{x}_c \tag{7.22}$$

where $\mathbf{T}$ is the Jacobian matrix given by equation 4.32. In expanded form, equation 7.22 can be written as:

$$\mathbf{x}_g = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \mathbf{x}_c$$

Here, t_{ij} are the individual elements of $\mathbf{T}^{-1}$. In this form, observation equations can be formed for any one of $\Delta\phi$, $\Delta\lambda$ orΔh. Thus, the observation equations for measured latitude ϕ_m and longitude λ_m coordinates and can be written as:

$$\phi_m - \phi_c = t_{11}\Delta x + t_{12}\Delta y + t_{13}\Delta z \tag{7.23}$$

$$\lambda_m - \lambda_c = t_{21}\Delta x + t_{22}\Delta y + t_{23}\Delta z \tag{7.24}$$

$$h_m - h_c = t_{31}\Delta x + t_{32}\Delta y + t_{33}\Delta z \tag{7.25}$$

where ϕ_c, λ_c and h_c are the latitude, longitude and ellipsoid height coordinates computed from the most recent estimates of x, y and z, and t_{11}, t_{12}, $\ldots$, t_{33} are the differentiations corresponding to x, y and z. If ϕ_m and λ_m have been derived from a previous adjustment, then a full variance matrix for $\mathbf{x}_c$ can be developed from the variance matrix relating to ϕ_m and λ_m.

Incidentally, observations of this kind provide a convenient, yet rigorous way to constrain a network with respect to the local reference frame in a single horizontal direction. By virtue of providing ϕ_m or λ_m as a unique measurement, a network can be horizontally constrained in the n or e dimension without the up dimension being influenced by the correlations that would normally be introduced through equation 4.31. The same also applies to constraining a network in the up dimension using equation 7.25, however it is computationally more efficient to do so using ellipsoid height (R) measurements (see §7.2.4). This is due to the necessity for the inversion of $\mathbf{T}$ before t_{31}, t_{32} and t_{33} can be obtained.

7.2.9 Geodetic azimuths and horizontal bearings (B)

Figure 7.5 shows a typical geodetic azimuth (or bearing corrected for the deflection of the vertical) measurement θ_{12} in the local reference frame from p_1 to p_2.

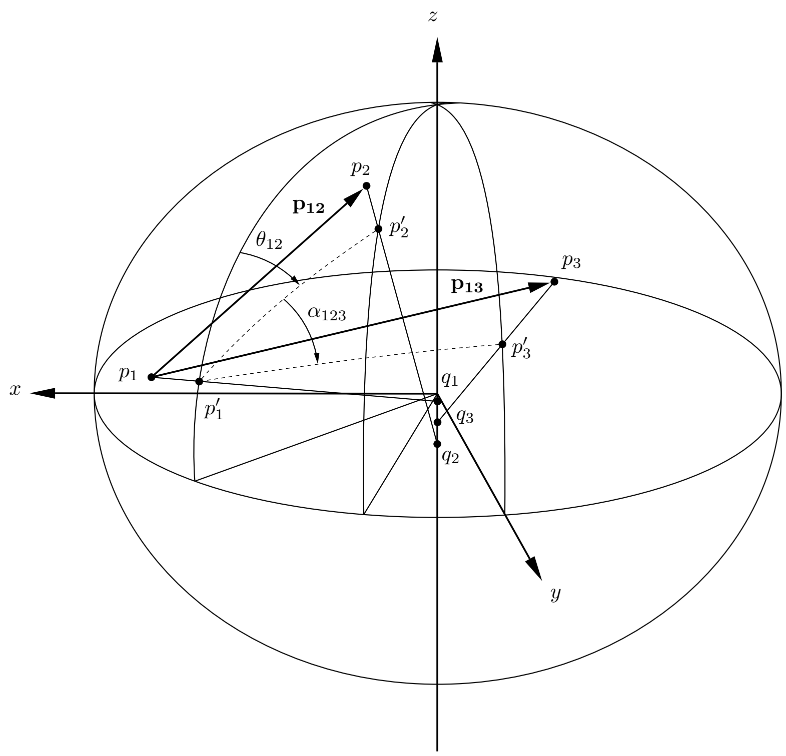


Figure 7.5: Bearings and horizontal angles

The azimuth between p_1 and p_2 is represented by vector $\mathbf{p_{12}}$. The elements of $\mathbf{p_{12}}$ are related to the cartesian reference frame via equation 4.11:

$$\mathbf{p_{12}} = \begin{bmatrix} \Delta e_{12} \\ \Delta n_{12} \\ \Delta up_{12} \end{bmatrix} = \mathbf{R}_1^T \begin{bmatrix} \Delta x_{12} \\ \Delta y_{12} \\ \Delta z_{12} \end{bmatrix} \tag{7.26}$$

where $\mathbf{R}_1$ is the rotation matrix defined by equation 4.9 evaluated at ϕ_1, λ_1. From equation 4.12, this can be written in expanded form as:

$$\begin{bmatrix} \Delta e_{12} \\ \Delta n_{12} \\ \Delta up_{12} \end{bmatrix} = \begin{bmatrix} -\sin\lambda(x_2 - x_1) + \cos\lambda(y_2 - y_1) \\ -\sin\phi\cos\lambda(x_2 - x_1) - \sin\phi\sin\lambda(y_2 - y_1) + \cos\phi(z_2 - z_1) \\ \cos\phi\cos\lambda(x_2 - x_1) + \cos\phi\sin\lambda(y_2 - y_1) + \sin\phi(z_2 - z_1) \end{bmatrix}$$

Hence, the azimuth θ_{12} between stations p_1 and p_2 is related to the vector elements with:

$$\theta_{12} = \arctan\left(\frac{\Delta e_{12}}{\Delta n_{12}}\right) \tag{7.27}$$

Traditionally, skew normal corrections are applied to azimuth measurements as part of the process of reducing such measurements from the terrain to the ellipsoid. The correction takes account of the fact that the normal section through the instrument station and containing the target station does not contain the normal through the target station. In DynAdjust, skew normal corrections are not required because the observation equation for an azimuth measurement (equation 7.27) is developed in the cartesian reference frame using vector geometry rather than on the surface of the ellipsoid.

7.2.10 Horizontal angles (A)

Figure 7.5 shows a typical angle measurement α at p_1 in the local reference frame between two directions (or bearings) from p_1 to p_2 and p_3. The local vectors between p_1 and p_2, and p_1 and p_3 are given in short as $\mathbf{p_{12}}$ and $\mathbf{p_{13}}$ respectively. As per equation 7.26, the local vector elements of $\mathbf{p_{13}}$ can be related to the cartesian reference frame via equation 4.11.

The horizontal angle α_{123} at station p_1 is related to the local vector elements with:

$$\alpha_{123} = \arctan\left(\frac{\Delta e_{13}}{\Delta n_{13}}\right) - \arctan\left(\frac{\Delta e_{12}}{\Delta n_{12}}\right) \tag{7.28}$$

Prior to evaluating the design matrix elements for horizontal angles, DynAdjust corrects observed angles for the deflection of the vertical. If ${}_m\alpha_{123}$ is an observed angle, the influence of the deflection of the vertical ϵ_1 at p_1 is corrected by:

$$\alpha_{123} = {}_m\alpha_{123} - \epsilon_1 \tag{7.29}$$

where ϵ_1 is the deflection of the vertical at p_1 given by Hofmann–Wellenhof and Moritz (2006):

$$\epsilon_1 = (\xi_1 \sin\theta_{13} - \eta_1 \cos\theta_{13}) \cot\zeta_{13} - (\xi_1 \sin\theta_{12} - \eta_1 \cos\theta_{12}) \cot\zeta_{12} \tag{7.30}$$

Here, ϕ_1 and λ_1 are the geodetic latitude and longitude at p_1, ξ_1 is the deflection in the prime meridian and η_1 is the deflection in the prime vertical evaluated at p_1, and ζ_{12} and ζ_{13} are the zenith angles (corrected for the deflection of the vertical) at p_1 to p_2 and p_3 respectively.

As will be gathered from equation 7.30, if the zenith angles ζ_{12} and ζ_{13} are equal, such as would be the case for angles in the horizontal plane, ϵ_1 will be negligible and can be safely ignored.

7.2.11 Horizontal direction sets (D)

Figure 7.6 shows a typical cluster of direction measurements θ_{12}, θ_{13}, θ_{14} and θ_{15} in the local reference frame from p_1 to p_2, p_3, p_4 and p_5.

The estimation of parameters from direction sets is handled by reducing the respective directions to angles α_{123}, α_{134} and α_{145}. The relationship between a cluster of directions and corresponding angles can be written as:

$$\alpha = \mathbf{Ad} \tag{7.31}$$

which can be written in expanded form as:

$$\begin{bmatrix} \alpha_{123} \\ \alpha_{134} \\ \alpha_{145} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_{12} \\ \theta_{13} \\ \theta_{14} \\ \theta_{15} \end{bmatrix} \tag{7.32}$$

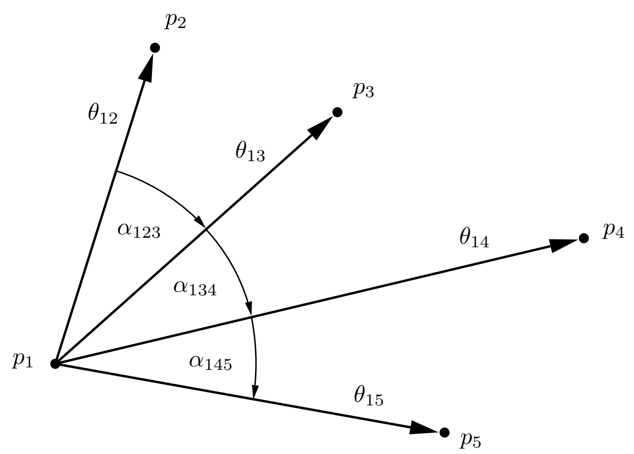


Figure 7.6: Cluster of horizontal directions

In this way, the observation equations which relate α_{123}, α_{134} and α_{145} to the x, y and z elements of p_1, p_2, p_3, p_4 and p_5 have the same form as those developed for angles in §7.2.10.

As it is common practice to observe a cluster of directions at the same time, correlations will exist between the derived angles α_{123}, α_{134}, α_{145}. Accordingly, the a–priori variance matrix must take these correlations into account. Given standard deviations $\sigma_{\theta_{12}}$, $\sigma_{\theta_{13}}$, $\sigma_{\theta_{14}}$ and σ_{θ_5} for the cluster of directions, the full variance matrix $\mathbf{V}_\alpha$ for the derived angles is obtained by propagation of variances:

$$\mathbf{V}_\alpha \; = \; \mathbf{A}\mathbf{V}_\theta\mathbf{A}^T \tag{7.33}$$

where $\mathbf{A}$ is the functional model relating the derived angles to the observed directions given in equation 7.32 and $\mathbf{V}_\theta$ is the (diagonal) variance matrix of the directions, given by:

$$\mathbf{V}_\theta \; = \; \begin{bmatrix} \sigma^2_{\theta_{12}} & 0 & 0 & 0 \\ 0 & \sigma^2_{\theta_{13}} & 0 & 0 \\ 0 & 0 & \sigma^2_{\theta_{14}} & 0 \\ 0 & 0 & 0 & \sigma^2_{\theta_{15}} \end{bmatrix} \tag{7.34}$$

The zero terms in $\mathbf{V}_\theta$ reflect the fact that the observed directions are independent measurements. Whilst the directions are uncorrelated, however, $\mathbf{V}_\alpha$ will be a non–diagonal matrix containing non–zero off-diagonal terms which represent the correlations between adjacent angles.

7.2.12 Zenith distances (V)

Figure 7.7 shows a typical zenith distance measurement ζ_{12} from p_1 to p_2. The vector between p_1 and p_2 is represented by $\mathbf{p_{12}}$.

The elements of $\mathbf{p_{12}}$ are related to the cartesian reference frame via equation 7.35:

$$\zeta_{12} \; = \; \arctan\left(\frac{\sqrt{\Delta e^2_{12} + \Delta n^2_{12}}}{\Delta up_{12}}\right) \tag{7.35}$$

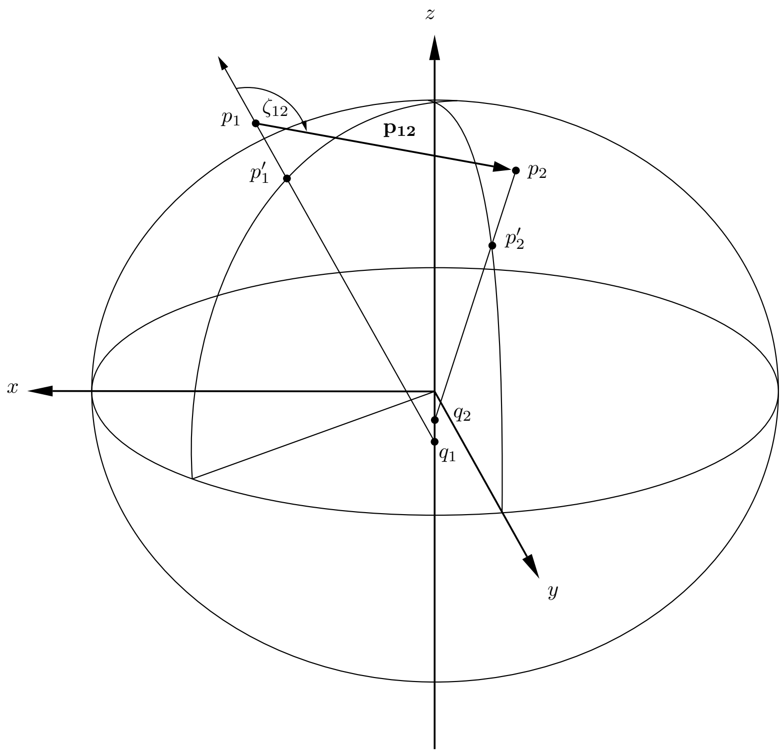


Figure 7.7: Zenith distances

Since zenith distances are observed between instrument and target, the elements in 7.35 relate to the vector formed between the instrument and target. These elements are derived by incorporating instrument and target heights into the computation of Δe, Δn and Δup.

Prior to evaluating the design matrix elements for zenith distances, DynAdjust corrects observed zenith distances for the deflection of the vertical. If ${}_m\zeta_{12}$ is an observed zenith distance, the influence of the deflection of the vertical is corrected by:

$$\zeta_{12} = {}_m\zeta_{12} - \epsilon_1 \tag{7.36}$$

where ϵ_1 is the deflection of the vertical at p_1 given by Hofmann–Wellenhof and Moritz (2006):

$$\epsilon_1 = \xi_1 \cos\theta_{12} + \eta_1 \sin\theta_{12} \tag{7.37}$$

Here, ξ_1 is the deflection in the prime meridian and η_1 is the deflection in the prime vertical evaluated at p_1, and θ_{12} is the azimuth (corrected for the deflection of the vertical) of the line from p_1 to p_2.

7.2.13 Vertical angles (Z)

Figure 7.8 shows a typical vertical angle measurement ϑ_{12} in the local reference frame from p_1 to p_2. The vector between p_1 and p_2 is represented by $\mathbf{p_{12}}$, the elements of which are related to the cartesian reference frame via equation 7.26.

The vertical angle ϑ_{12} between stations p_1 and p_2 is related to the vector elements with:

$$\vartheta_{12} = \arctan\left(\frac{\Delta up_{12}}{\sqrt{\Delta e_{12}^2 + \Delta n_{12}^2}}\right) \tag{7.38}$$

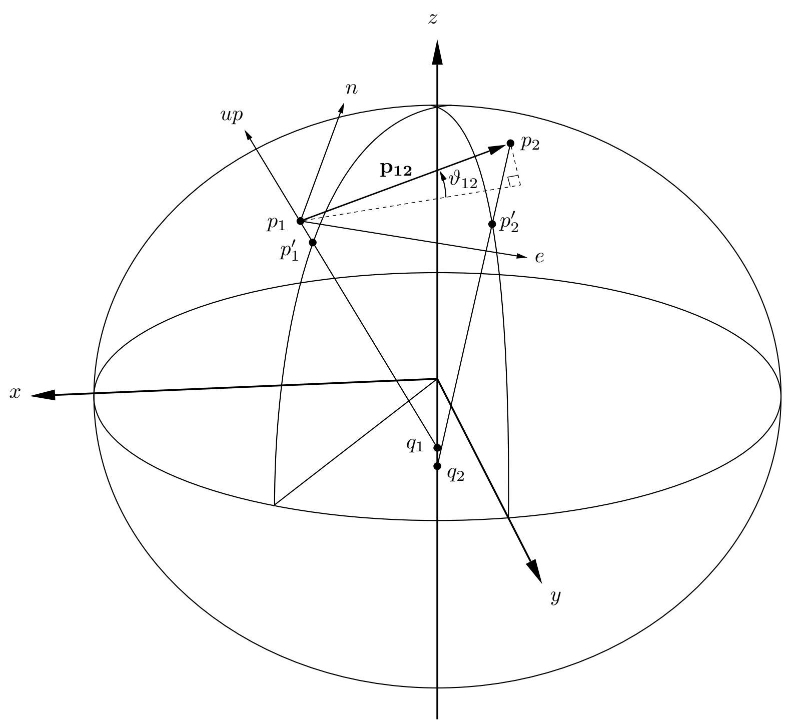


Figure 7.8: Vertical angles

As with zenith distances, vertical angles are observed between instrument and target. Hence, the elements in 7.38 relate to the vector formed between the instrument and target. These elements are derived by incorporating instrument and target heights into the computation of Δe, Δn and Δup.

Prior to evaluating the design matrix elements for vertical angles, DynAdjust corrects observed vertical angles for the deflection of the vertical using equation 7.29, which is repeated here for vertical angles as:

$$\vartheta_{12} \; = \; {}_m\vartheta_{12} - \epsilon_1 \tag{7.39}$$

where ϵ_1 is the deflection of the vertical at p_1 given by equation 7.37.

7.2.14 Astronomic latitude (I) and longitude (J)

The treatment of astronomic latitude Φ and longitude Λ measurements at point p is handled by reducing the values for Φ and Λ to their geodetic counterparts ϕ and λ. Astronomic latitude and longitude are related to geodetic latitude and longitude by Hofmann–Wellenhof and Moritz (2006):

$$\begin{aligned} \phi \; &= \; \Phi - \xi \\ \lambda \; &= \; \Lambda - \eta \sec \phi \end{aligned} \tag{7.40}$$

where ξ is the deflection in the prime meridian and η is the deflection in the prime vertical at p. As explained in §7.2.8, the reduced ϕ and λ can be related to the coordinates of p in the cartesian reference frame using equations 7.23 and 7.24 respectively.

7.2.15 Astronomic (or Laplace) azimuth (K)

The treatment of astronomic azimuth measurement A_{12} from point p_1 to p_2 is handled by reducing A_{12} to a geodetic azimuth θ_{12} prior to adjustment. Astronomic azimuths are related to geodetic azimuths via:

$$\theta_{12} \;=\; A_{12} - \Delta\alpha_1 \tag{7.41}$$

where $\Delta\alpha_1$ is the Laplace correction at p_1 given by Hofmann–Wellenhof and Moritz (2006):

$$\Delta\alpha_1 \;=\; \eta\tan\phi_1 + (\xi_1\sin\theta_{12} - \eta_1\cos\theta_{12})\cot\zeta_{12} \tag{7.42}$$

Here, ϕ_1 and λ_1 are the geodetic latitude and longitude at p_1, ξ_1 is the deflection in the prime meridian and η_1 is the deflection in the prime vertical evaluated at p_1, and ζ_{12} is the zenith angle (corrected for the deflection of the vertical) at p_1.

As explained in §7.2.9, θ_{12} can be related to the coordinates of p_1 and p_2 using equation 7.27.

7.3 Stochastic modelling and reporting

7.3.1 Probability distributions used in DynAdjustfor testing

7.3.1.1 Normal distribution

Survey measurements, free of blunders and systematic error, are assumed to be random variables which are Normally distributed about a *mean* of μ with a *variance* of σ^2. It follows that the errors resulting in random measurement variations can also be considered as random variables with a mean of zero. Likewise, estimates which are based on those measurements are assumed to possess the same random characteristics. For these reasons, DynAdjust uses the *Normal* or *Gaussian distribution* as the basic model for quantifying and testing the statistical behaviour of measurements, measurement corrections and parameter estimates. The *probability density function* (*PDF*) of the Normal distribution is given by:

$$f(x) \;=\; \frac{1}{\sigma_x\sqrt{2\pi}}\exp - \left[\frac{(x-\mu_x)^2}{2\sigma_x^2}\right], \qquad -\infty < x < \infty \tag{7.43}$$

Equation 7.43 is used to estimate the probability of a measurement correction or estimated coordinate coinciding with the mean or falling within a certain interval (e.g. $\mu_x - \sigma_x \;\leq\; \hat{x} \;\leq\; \mu_x + \sigma_x$). Figure 7.9 shows the typical bell shaped curve of the Normal PDF.

The probability that a measurement correction or estimated coordinate will fall within a certain confidence interval $\mu - \sigma$ and $\mu + \sigma$ is given by:

$$P\left[\,\mu - \sigma \;\leq\; \bar{x} \;\leq\; \mu + \sigma\,\right] \;\approx\; \int_{-\infty}^{+\infty} f(x)dx \;\approx\; \alpha \tag{7.44}$$

where α is a value ranging from 0 to 1 commonly expressed as a percentage. The statistical inference is that P is the level of uncertainty that the estimated parameter will fall within the interval $\pm\sigma$ about the mean μ. When expressing the level of probability as a specific multiple of $\pm\sigma$, k (or coverage factor) is used. For example, for $k = 1$, $\alpha = 0.683$ (or 68.3%). Common confidence intervals based upon multiples of σ and their respective probabilities are given in Table 7.1.

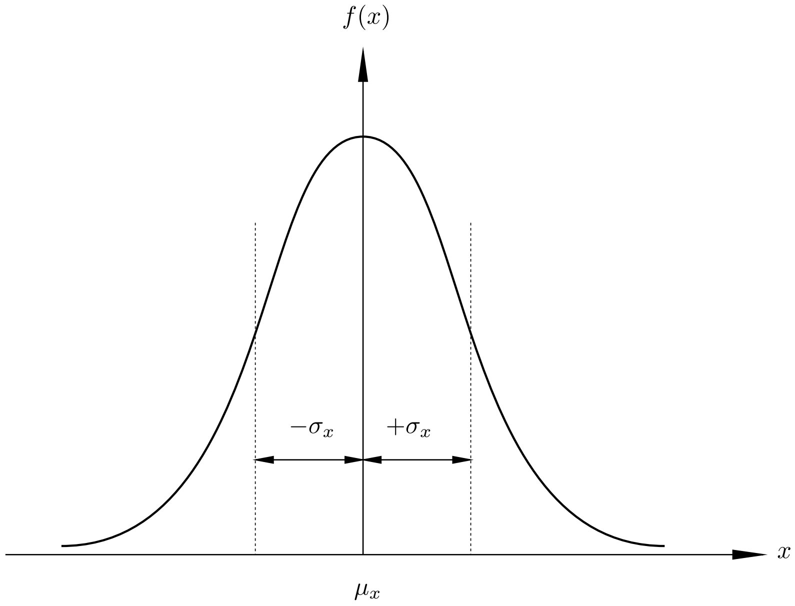


Figure 7.9: The Normal distribution

Table 7.1: Confidence intervals for $\pm 1\sigma$ to $\pm 4\sigma$.

k	$P\,[\,\mu - k\sigma \ \leq \ x \ \leq \ \mu + k\sigma\,]$
1.00	68.3 %
1.96	95.0 %
2.00	95.4 %
2.58	99.0 %
3.00	99.73 %
3.29	99.9 %
4.00	99.994 %

By default, DynAdjust adopts the 95% confidence interval for all statistical reporting and testing. See §8.3.3 on page 124 for selecting an alternative confidence interval by which to quantify and test measurements, their assumed variances and the parameters estimated from an adjustment. The use of the Normal distribution to test the quality of supplied measurements will be discussed in §9.3.2.

7.3.1.2 Chi–square distribution

DynAdjust uses the *Chi–square* (χ^2) distribution to test the reliability of a least squares adjustment. By definition, the quantity w obtained from the sum of the squares of a set of Normally distributed random variables is a χ^2 random variable and is given by:

$$w_{\chi^2} \ = \ \mathbf{v}^T\mathbf{v} \ = \ \left[\, x_1 \ \ x_2 \ \ \cdots \ \ x_u \,\right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_u \end{bmatrix} \tag{7.45}$$

A χ^2 random variable may also be obtained from the *weighted* sum of the squares:

$$w_{\chi^2} = \mathbf{v}^T\mathbf{V}^{-1}\mathbf{v} = \begin{bmatrix} x_1 & x_2 & \cdots & x_u \end{bmatrix} \begin{bmatrix} \sigma^2_{x_1} & \sigma_{x_1x_2} & \cdots & \sigma_{x_1x_n} \\ \sigma_{x_2x_1} & \sigma^2_{x_2} & & \vdots \\ \vdots & & \ddots & \\ \sigma_{x_nx_1} & \cdots & & \sigma^2_{x_u} \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_u \end{bmatrix} \tag{7.46}$$

where $\mathbf{V}$ is the variance matrix which weights the respective contribution of each variable $x_1, x_2, ..., x_u$ to w_{χ^2}. The PDF of χ^2 random variables is given by:

$$f(x) = \frac{x^{\frac{r}{2}-1}e^{-\frac{x}{2}}}{\Gamma\left(\frac{r}{2}\right)2^{\frac{r}{2}}} \tag{7.47}$$

where $\Gamma\left(\frac{1}{2}r\right)$ is a *gamma function* and r is the degrees of freedom in the model. The χ^2 PDFs for random variables having 4, 6, 10 and 18 degrees of freedom respectively are shown in Figure 7.10.

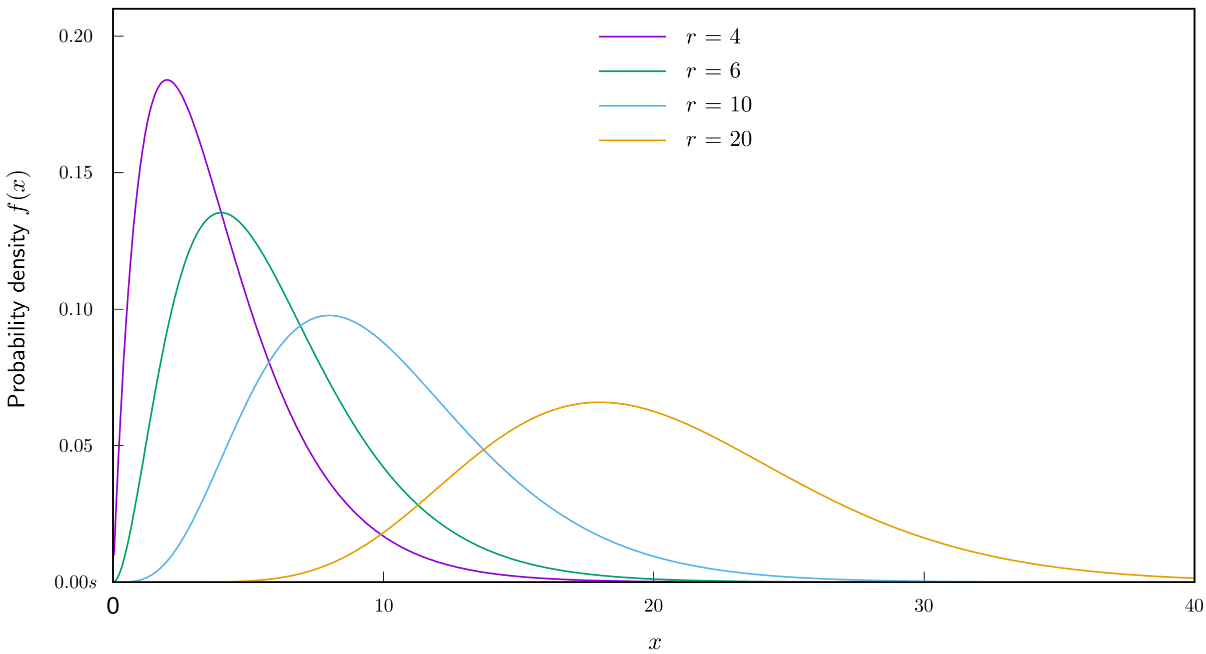


Figure 7.10: Chi–square (χ^2) distributions for 4, 6, 10 and 20 degrees of freedom

As shown in Figure 7.10, as r increases to ∞, the χ^2 distribution will take the form of the Normal distribution. The expected value of a χ^2 random variable w is defined by the degrees of freedom:

$$E\left[w_{\chi^2_r}\right] = r \tag{7.48}$$

and its variance is given by:

$$\sigma^2\left(w_{\chi^2_r}\right) = 2r \tag{7.49}$$

Therefore, as $r \to \infty$, $\mu = r$ and $\sigma^2 = 2r$. The probability of a χ^2 random variable being greater than a particular *critical value* c is denoted as α and is given by:

$$\alpha = P\left[w_{\chi^2_r} > c\right] = \int_c^{\infty} f(x)dy \tag{7.50}$$

where α is the level of significance. The use of this distribution to test the quality of a least squares adjustment will be explained in §9.3.1.

7.3.1.3 Student's t distribution

When a network contains a large set of measurements having reliable estimates of precision, the Normal distribution offers a reliable means for testing measurement quality. However, when a network is comprised of a limited number of measurements, or the assumed measurement precisions are doubtful, the level of confidence that can be gained from testing against the Normal distribution decreases. To obtain a higher level of confidence from a solution in such circumstances, DynAdjust can optionally compute statistics according to the Student's t (t) distribution which provides for more stringent testing on the supplied measurements.

The PDF of the t distribution is given by:

$$f(x) = \frac{\Gamma\left(\frac{r+1}{2}\right)}{\sqrt{r\pi}\,\Gamma\left(\frac{r}{2}\right)} \cdot \frac{1}{\left(1+\frac{x^2}{r}\right)^{\frac{r+1}{2}}} \qquad (7.51)$$

where $\Gamma(\ldots)$ is the gamma function and r is the degrees of freedom. Four t PDFs having 2, 5, 10 and 50 degrees of freedom respectively are shown in Figure 7.11.

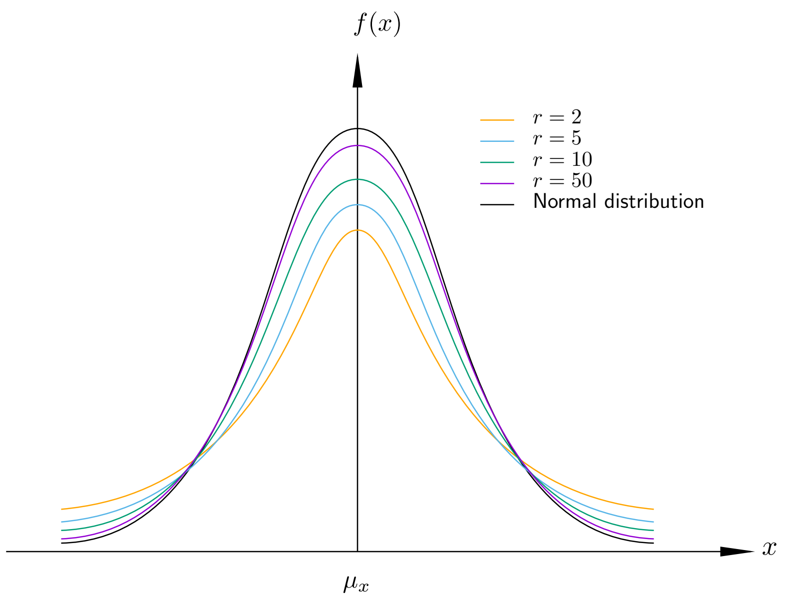


Figure 7.11: Student's t distributions for 2, 5, 10 and 50 degrees of freedom

As indicated by Figure 7.11, as r increases, the t distribution approaches the Normal distribution. What is distinctly different however, is that for low values of r, the t distribution has the characteristic of causing variations from the mean to disperse quickly beyond the confidence intervals giving a lower tendency to mark variations as outliers.

The Expected value of a t random variable is:

$$E\left[w_{t_r}\right] = 0 \qquad (r > 1) \qquad (7.52)$$

and its variance is given by:

$$\sigma^2 (w_{t_r}) \; = \; \frac{r}{r-2} \qquad (r \geq 2) \tag{7.53}$$

The probability that a measurement correction will fall within a certain confidence interval $-\sigma$ and $+\sigma$ is given by:

$$\alpha \; = \; P\left[w_{t_r} > c\right] \; = \; \int_c^{\infty} \; f(x)dy \tag{7.54}$$

The use of this distribution to analyse small networks with measurements having doubtful precisions will be explained in §9.3.2.2.

7.3.2 Preparation of measurement precisions and variance matrices

Since a least squares adjustment of a geodetic network depends upon a reliable stochastic model, it is imperative that realistic estimates of precision are assigned to the measurements. Whilst some may argue that the use of least squares to estimate the precision of measurements is *theoretically* legitimate, the *practical* limitation of this argument is that the true statistical behaviour of a measurement and its contribution to the estimation of station coordinates will be biased.

In the absence of having a rigorous instrument calibration uncertainty[2], sound geodetic practice for achieving reliable measurement precisions involves estimating the standard deviation from a large sample of measurements. Traditionally, the procedure employed by many geodetic agencies was, for each instrument type, to (i) take a large sample of measurements under normal operating conditions (and over different parts of the year); (ii) estimate the standard deviation from those measurements; and (iii) scale and/or add additional terms to the estimated standard deviation to account for external influences not discernible from the sample (such as errors which have a systematic effect and need to be estimated from experience). The quantity derived from (ii) is sometimes termed *Type A* uncertainty or *internal precision*, whereas the quantity obtained during (iii) is termed *Type B* uncertainty or *external precision*. Using elementary statistical principles, the precision for a measurement at (ii) may be obtained by:

$$\sigma^2 \; = \; \frac{1}{(n-1)} \sum_1^n (m_1 - \overline{m})^2 \tag{7.55}$$

where n is the sample size, m_i relates to each measurement in the sample and $\overline{m}$ is the theoretical mean of the sample. Following the procedure outlined above, any measurement taken from an instrument thereafter would be assigned the total (one–sigma) uncertainty estimate.

Nowadays however, this procedure is almost totally forgotten or regarded as being unimportant with the availability of high precision equipment. Instead, the practice of taking an adjustment and scaling the supplied variances by the estimated variance factor (c.f. §9.3.1.1) has become a favourable option due to the relative ease and efficiency with which it can be computed. Whilst this may be true for several modern digital surveying instruments, the same cannot be said for *all* instruments or GNSS baselines and point clusters which have been derived from GNSS analysis.

Experience shows that GNSS variance matrices derived from GNSS analysis software are almost always over optimistic, giving only an estimate of the internal precision. More often than not, the uncertainties from all collatoral measurement sources (e.g. carrier phase measurement precision;

2. A rigorous uncertainty computed by instrument calibration and expanded to include uncertainties for all known collatoral measurement sources and unmodelled systematic error.

receiver clock estimates; antenna phase centre models; satellite orbit and clock estimates) and systematic models (e.g. effect of double–difference decorrelation; troposphere and ionosphere models; ocean loading models) are not taken into account. From another perspective, variance matrices derived from short occupation periods almost always fail to convey the imprecision in the resulting measurements. Failing to take the uncertainties of all of these factors results in variance matrices which are, on the whole, over–optimistic. Ultimately, this problem causes a number of down–stream difficulties when attempting to combine measurements taken over the same stations under different circumstances, different observation techniques, different equipment and over different times of the year.

DynAdjust cannot generate realistic matrices, but can be used to assist with attaining more reliable estimates of measurement precision. Two such tools include GNSS variance matrix scaling (discussed in the next section) and local testing via the Student's t distribution (c.f. §7.3.1.3 and §9.3.2.2).

7.3.2.1 Scaling GNSS variance matrices

As described in §3.3.3, DynAdjust offers a capacity for a–priori GNSS variance matrices to be scaled prior to adjustment. The options for scaling a variance matrix include a total variance matrix scalar and individual (or partial) variance matrix scalars. A total variance matrix scalar is in the form of a single multiplier or variance factor. Partial matrix scalars are given in terms of east–west, north–south and up scalars. Both forms of scaling are stored within the binary measurement file and applied upon executing an adjustment. Hence, the original input files are not modified.

Variance matrix scaling using a single multiplier is often carried out once the quality of the solution has been determined. Scaling a variance matrix by a single multiplier is performed automatically prior to the least squares adjustment as:

$$\mathbf{V}_{\mathbf{m}}^{scaled} = s^2\mathbf{V}_{\mathbf{m}} \tag{7.56}$$

where $\mathbf{V}_{\mathbf{m}}$ is the a–priori variance matrix and s is the variance matrix scalar.

In the case of partial matrix scaling, it may be desirable to, for example, scale a variance matrix by a greater magnitude in the vertical axis than in the horizontal plane. To effect this outcome, three operations are performed. Firstly, the a–priori variance matrix is propagated into the local reference frame from either the cartesian or geographic reference frame such that the variance matrix terms are in units of east, north and up. This is achieved by applying both equations 4.31 and 4.27 if the input variance matrix is in the geographic reference frame or equation 4.27 alone if the input variance matrix is in the cartesian reference frame.

Secondly, partial variance matrix scalars are applied via propagation of variances:

$$\mathbf{V}_{\mathbf{x}_l}^{scaled} = \mathbf{S}_{\mathbf{enu}}\mathbf{V}_{\mathbf{x}_l}\mathbf{S}_{\mathbf{enu}}^T \tag{7.57}$$

where the functional model is expressed as:

$$\mathbf{S}_{\mathbf{enu}} = \begin{bmatrix} \sqrt{s_{east}} & 0 & 0 \\ 0 & \sqrt{s_{north}} & 0 \\ 0 & 0 & \sqrt{s_{up}} \end{bmatrix} \tag{7.58}$$

If $\mathbf{V}_{\mathbf{x}_l}$ is larger than a (3x3) matrix — that is, it applies to a GNSS cluster rather than a single baseline or position — the same procedure as described above is used except the $\mathbf{S}_{\mathbf{enu}}$ matrix is constructed so as to match the dimensions of the variance matrix that is to be scaled. The scaling matrix will be block–diagonal, with each (3x3) sub-matrix on the leading diagonal being as shown in equation 7.58 and all off–diagonal terms will be zero. If a single multiplier scalar is supplied with partial matrix scalars, the partial scalar values are multiplied by the single multiplier first prior to propagation of variances.

Thirdly, the scaled variance matrix is propagated into the cartesian reference frame using equation 4.8, such that it is in a ready state for the development of the normal equation matrix.

7.3.3 Expressing estimates of quality and reliability

As will be seen in §8.2, a by–product of least squares adjustment is an estimate of the precision of the estimated parameters and adjusted measurements. To assist with analysing the quality and reliability of a network, DynAdjust performs additional computations to quantify and express the following indicators:

- Quality of a least squares adjustment
- Quality of the estimated station coordinates (in one, two or three dimensions)
- Quality of supplied and adjusted measurements
- Reliability of the supplied measurements
- Reliability of the whole network

As discussed in the previous section, DynAdjust uses the Chi–square statistic to quantify the quality of the solution of a network. To express the quality of estimated station coordinates, standard deviations directly obtained from the estimated variance matrix are used by default. Optionally, station coordinate quality in the horizontal plane can be quantified in terms of the standard error ellipse and a circular 95% confidence region, known as *Positional Uncertainty*. On reporting the quality of measurements, residuals *normalised* (or *standardised*) to certain confidence intervals of the unit Normal distribution and unit Student's t distribution may be computed. Measurement reliability and network reliability are expressed using quantities based on supplied and adjusted measurement precisions. The formula used by DynAdjust to compute these indicators are described in the following sections.

7.3.3.1 Error ellipses

The *error ellipse* of constant probability is the locus of points (x, y) described by the intersection between a horizontal plane at a height k above the xy plane and the continuous PDF surface. Figure 7.12a illustrates the error ellipse at height k. Figure 7.12b shows the relevant parameters of the error ellipse.

The size, shape and orientation (i.e. the complete spatial behaviour) of the ellipse is obtained when the exponent term in the PDF of the Normal distribution is set to a specific value k^2:

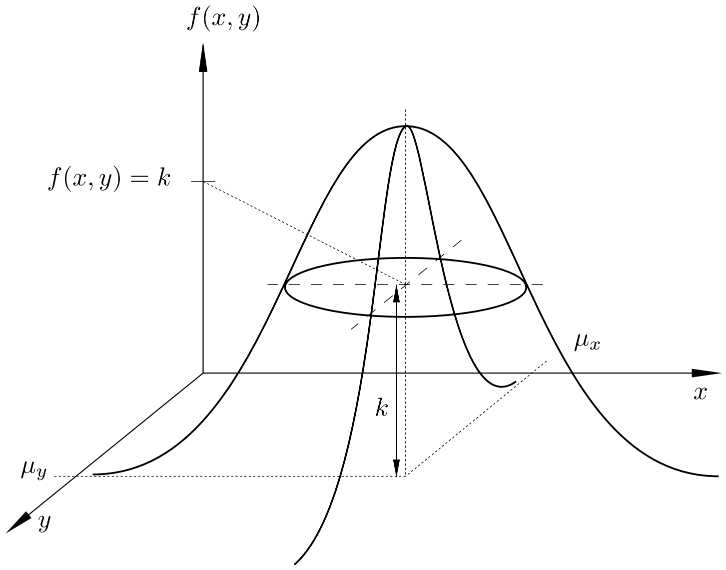
$$(\mathbf{m} - \mu_{\mathbf{m}})^T \mathbf{V}_{\mathbf{m}}^{-1} (\mathbf{m} - \mu_{\mathbf{m}}) \ = \ k^2 \tag{7.59}$$

The centre of the ellipse coincides with the most probable values for x and y, being μ_x and μ_y respectively. The semi–major a and semi–minor b axes are defined by the square roots of the relevant eigenvalues of $\mathbf{V_m}$ Mikhail (1976):

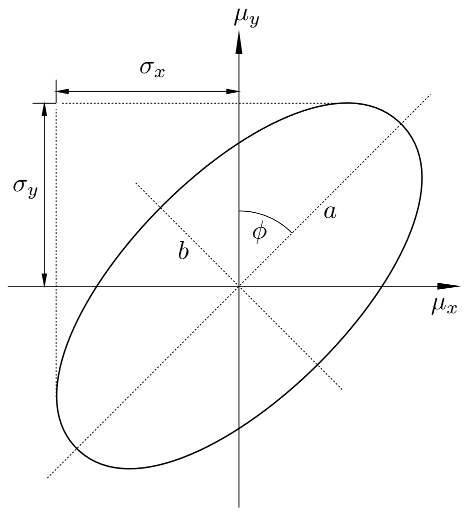
$$a = \left[\frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 + z\right)\right]^{\frac{1}{2}} \tag{7.60}$$

$$b = \left[\frac{1}{2}\left(\sigma_x^2 + \sigma_y^2 - z\right)\right]^{\frac{1}{2}} \tag{7.61}$$

$$z = \left[\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2\right]^{\frac{1}{2}}$$



(a) The intersection of a horizontal plane at height k



(b) The error ellipse of constant probability

Figure 7.12: The error ellipse and its dependence on coverage factor k

The estimated variances σ_x^2 and σ_y^2 are dimensioned along the x and y axes of the two–dimensional reference frame. When the estimates of x and y are uncorrelated, the axes of the ellipse will be coincident with the x and y axes. When $\rho \neq 0$, the ellipse axes are rotated relative to the x and y axes by the angle ϕ. The orientation ϕ of the error ellipse is a function of the estimated variances and covariance of x and y, and is given by:

$$\phi = \frac{1}{2}\arctan\left[\frac{2\sigma_{xy}}{\sigma_y^2 - \sigma_x^2}\right] \tag{7.62}$$

When the correlation between Normal random variables under the same probability distribution is equal to zero, the orientation ϕ is equal to zero.

At any time, users may opt to present these quantities in higher confidence intervals than the standard. For example, if users wish to express standard deviations for one dimensional components (such as height) at the 95% confidence level, the uncertainty value is computed by scaling the estimated (1 sigma) standard deviation by coverage factor $k = 1.960$. To express the standard two–dimensional error ellipse at the 95% confidence level, the axes of the 95% error ellipse are obtained by scaling the (1 sigma) axes by coverage factor $k = 2.448$. Similarly, the axes of the 95% three dimensional error ellipsoid are obtained by scaling the (1 sigma) axes by coverage factor $k = 2.796$.

7.3.3.2 Positional uncertainty

According to the *Guideline for the Adjustment and Evaluation of Survey Control* ICSM (2014), *Positional Uncertainty* (PU) is the uncertainty of the horizontal and/or vertical coordinates of a survey control mark with respect to the defined datum and represents the combined uncertainty of the existing datum realisation and the new control survey. PU is expressed in SI units at the 95% confidence level.

For the horizontal circular confidence region, the 95% uncertainty value is calculated from the standard (1 sigma) error ellipse and is expressed as a single quantity, being the radius of the circular confidence region. The radius r of the circular confidence region is computed by:

$$r = a \times K \tag{7.63}$$

$$K = q_0 + q_1 C + q_2 C^2 + q_3 C^3 \tag{7.64}$$

$$C = \frac{b}{a} \tag{7.65}$$

where:

a is the semi–major axis of the standard error ellipse computed from equation 7.60

b is the semi–minor axis of the standard error ellipse computed from equation 7.61

$q_0 = 1.960790$

$q_1 = 0.004071$

$q_2 = 0.114276$

$q_3 = 0.371625$

Values for a and b are derived from the full a–posteriori variance matrix (or precision of the estimated coordinates) obtained from least squares adjustment (discussed in §8.2 and evaluated by equation 8.16).

7.3.3.3 Measurement reliability and network reliability

Amongst the reliability factors which may be used to determine measurement reliability, a common and relatively simple indicator is Pelzer's measurement reliability criterion Prószynski (1994). Pelzer's criterion is computed as:

$$\tau = \frac{\sigma_i^2}{\sigma_i^2 - \hat{\sigma}_i^2} \tag{7.66}$$

where σ_i^2 is the a–priori measurement precision and $\hat{\sigma}_i^2$ is the a–posteriori measurement precision arising from least squares adjustment. Based on this formula, τ will vary between unity and infinity. The larger the value for τ, the poorer the reliability. For instance, as measurement redundancy increases, $\hat{\sigma}_i^2$ will decrease and cause τ to tend towards unity.

Once the reliability criterion has been computed for each measurement via equation 7.66, the global (or whole) network reliability can be computed as:

$$T^2 = \sum_n^1 (\tau^2 - 1) \tag{7.67}$$

Chapter 8

Estimation of station parameters

8.1 Introduction

DynAdjust estimates unknown station parameters using the technique of least squares via the traditional observation equations method. The parameters which may be estimated include station coordinates and their uncertainties, adjusted measurements and the precision of adjusted measurements. DynAdjust also computes a small range of statistics to aid in the validation of the supplied measurements and their precisions, and in the analysis and verification of the estimated parameters. The program which performs all adjustment operations is **adjust**.

The objectives of this chapter are twofold — firstly, to provide a basic overview of the least squares algorithm employed by **adjust** and secondly, to describe how **adjust** may be configured to achieve maximum computational efficiency and to produce the desired output. Chapter 9 explains how the statistics computed by **adjust** can be used to validate the precision of the measurements and the reliability of the least squares solution as a whole.

8.2 Overview of least squares estimation

According to the least squares observation equations method, the solution of unknown parameters is achieved using linearised observation equations (or measurement functions) without any need for condition or constraint equations. For each measurement, the observation equation is developed as:

$$m = f(x_1, x_2, \ldots, x_u) \tag{8.1}$$

where $x_1, x_2, \ldots, x_u$ are the parameters to be estimated and f is the function that expresses the deterministic relationship between a measurement and the unknown parameters. The observation equations presented in §7.2 have been developed in the form of equation 8.1. In matrix form, a system of n linearised observation equations can be related to a set of u unknown parameters by the functional model:

$$\mathbf{m} = \mathbf{A}\mathbf{x} \tag{8.2}$$

where:

- $\mathbf{x}$ is a vector of unknown parameters (size $u \times 1$)
- $\mathbf{m}$ is a vector of measurements (size $n \times 1$)
- $\mathbf{A}$ is the matrix of coefficients, or design matrix (size $n \times u$)

When the number n of functionally independent measurements exceeds the number u of unknown parameters, the redundancy r (or degrees of freedom) will be defined as:

$$r = n - u \tag{8.3}$$

If $n < u$, the solution is indeterminate. If $n = u$, there is only one possible solution for the unknown parameters. If $n > u$, an inconsistency in $\mathbf{m} = \mathbf{Ax}$ will arise. This inconsistency arises from the fact that some form of error will exist in a set of measurements which, in turn, leads to a range of possible parameter values that could satisfy the system of observation equations.

To arrive at a unique set of values for the parameters, the least squares algorithm adjusts the system of measurements $\mathbf{m}$ by a set of estimated corrections (or residuals*)* $\hat{\mathbf{v}}$ such that the functional model is satisfied. In matrix form, the adjusted measurements $\hat{\mathbf{m}}$ are related to the original measurements $\mathbf{m}$, residuals $\hat{\mathbf{v}}$ and estimated parameters $\hat{\mathbf{x}}$ as follows:

$$\hat{\mathbf{m}} = \mathbf{m} - \hat{\mathbf{v}} \tag{8.4}$$

$$= \mathbf{A}\hat{\mathbf{x}} \tag{8.5}$$

This allows equation 8.2 to be expressed as:

$$\mathbf{m} - \hat{\mathbf{v}} = \mathbf{A}\hat{\mathbf{x}} \quad \text{with} \quad E(\mathbf{m}) = \boldsymbol{\mu} = \mathbf{Ax} \quad \text{and} \quad \mathrm{E}(\hat{\mathbf{v}}) = 0 \tag{8.6}$$

The expectation $E(\hat{\mathbf{v}}) = 0$ is based on the hypothesis that the measurements $\mathbf{m}$ are Normally distributed random variables with an expected mean of $\boldsymbol{\mu} = \mathbf{Ax}$, and that $\mathbf{v}$ represents random error in $\mathbf{m}$. The subject of dealing with larger than expected values for $\mathbf{v}$ will be taken up in §9.3.2.

For the observation equations in the functional model which are non–linear with respect to the parameters, the linear form of functions f_1, f_2, f_3, $\ldots$, f_n in $\mathbf{A}$ are approximated by the zero and first–order terms of a Taylor series expansion:

$$\begin{aligned}
m_1 - \hat{v}_1 &= f_1(x_1', x_2', \ldots, x_u') + \frac{\partial f_1}{\partial x_1}\Delta x_1 + \frac{\partial f_1}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_1}{\partial x_u}\Delta x_u \\
m_2 - \hat{v}_2 &= f_2(x_1', x_2', \ldots, x_u') + \frac{\partial f_2}{\partial x_1}\Delta x_1 + \frac{\partial f_2}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_2}{\partial x_u}\Delta x_u \\
&\vdots \\
m_n - \hat{v}_n &= f_n(x_1', x_2', \ldots, x_u') + \frac{\partial f_n}{\partial x_1}\Delta x_1 + \frac{\partial f_n}{\partial x_2}\Delta x_2 + \cdots + \frac{\partial f_n}{\partial x_u}\Delta x_u
\end{aligned} \tag{8.7}$$

where x_1', x_2', $\ldots$, x_u' are the starting values used to compute $\mathbf{m}$. Hence, $x_i = x_i' + \Delta x_u$. Simplified in matrix form, equation 8.7 is written as:

$$\mathbf{m} - \hat{\mathbf{v}} = \mathbf{m}' + \mathbf{A}\hat{\mathbf{x}} \tag{8.8}$$

where $\mathbf{m}'$ denotes the vector of measurements computed from the starting values and $\mathbf{A}$ is the full Jacobian (or *design*) matrix of partial derivatives of f with respect to the parameters x_1, x_2, $\ldots$, x_u. Since the linearised functional model is approximated using a first order Taylor series, the solution will require iteration. In steps, the vector of *reduced measurements* $\mathbf{m}_r$ is computed with an initial approximate for $\hat{\mathbf{m}}$, denoted by $\hat{\mathbf{m}}'$:

$$\mathbf{m}_r = \mathbf{m} - \hat{\mathbf{m}}' \tag{8.9}$$

and then the *most probable values* are evaluated by:

$$\hat{\mathbf{m}} = \mathbf{m}_r + \mathbf{A}\hat{\mathbf{x}} \tag{8.10}$$

The solution is iterated (i.e. $\hat{\mathbf{m}}$ is substituted for $\hat{\mathbf{m}}'$) until Δx_1, $\Delta x_2'$, ..., $\Delta x_u'$ in equation 8.7 converge to zero (i.e. when $\mathbf{m}_r = 0$). Assuming that the measurements are Normally distributed, the least squares algorithm estimates values for $\hat{v}_1$, $\hat{v}_2$, ..., $\hat{v}_n$ so that the values for $\hat{\mathbf{x}}$ will have a maximum probability under the Normal distribution. This is achieved by minimising the exponent of the normal distribution PDF (c.f. equation 7.43) for $\mathbf{m}$:

$$w = (\mathbf{m} - \mathbf{A}\mathbf{x})^T V_{\mathbf{m}}^{-1} (\mathbf{m} - \mathbf{A}\mathbf{x}) \tag{8.11}$$

where $\mathbf{V_m}$ is a symmetric positive definite variance matrix (size $n \times n$) representing the stochastic behaviour of the measurements , and w is the weighted sum of the squares of the corrections. After substituting $\mathbf{m} - \mathbf{A}\mathbf{x}$ with $\mathbf{v}$ (from equation 8.6), w can be rewritten in the familiar form:

$$w = \mathbf{v}^T V_{\mathbf{m}}^{-1} \mathbf{v} \tag{8.12}$$

The minimum value for w is achieved by minimising equation 8.11 as:

$$\frac{\partial w}{\partial \mathbf{x}} = 0 \tag{8.13}$$

which after expansion and differentiation of equation 8.11 leads to the least squares normal equations:

$$\left(\mathbf{A}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{A}\right) \mathbf{x} = \mathbf{A}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{m} \tag{8.14}$$

Solving for $\mathbf{x}$ enables the least squares solution of the unknown parameters and corresponding variance matrix to be written as:

$$\hat{\mathbf{x}} = \left(\mathbf{A}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{m} \tag{8.15}$$

$$\mathbf{V}_{\hat{\mathbf{x}}} = \left(\mathbf{A}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{A}\right)^{-1} \tag{8.16}$$

Here, distinction is made between $\mathbf{x}$ and $\hat{\mathbf{x}}$ in that the latter represents the estimates of the former — the most probable values according to the functional model $\mathbf{A}$ and stochastic model $\mathbf{V_m}$. Explanation of $\mathbf{V}_{\hat{\mathbf{x}}}$ is left until Chapter 9. If the functional model is non–linear, equation 8.15 will need to be iterated to arrive at the final coordinate and precision estimates.

8.3 Adjusting a network

Once **import** has created the binary station (`.bst`) and measurement (`.bms`) files, and the associated station (`.asl`) and measurement (`.aml`) lists from the input station and measurement files, the network is ready for adjustment. This assumes that all necessary reference frame transformations (c.f. Chapter 4) and geoid file interpolations (c.f. Chapter 5) have been undertaken. For phased adjustments, the only additional precondition apart from running **import** is that **segment** has been run to create a segmentation file (c.f. Chapter 6).

With **adjust**, users may undertake adjustments in either simultaneous mode or phased adjustment mode. By default, **adjust** will perform network adjustments in simultaneous mode. In phased adjustment mode, several options are provided so as to maximise computational efficiency based upon the computer's available processor, physical memory and hard disk resources. For both modes, several options are provided to configure the adjustment behaviour and output. If there is a requirement to re–print the results from a previous adjustment using different configuration options, **adjust** may be executed in report results mode. In this mode, no adjustment is undertaken but rather, the output files are regenerated. The complete command line reference for **adjust** is given in §A.6. If no program options are provided upon program execution, the command line reference will be displayed.

If a DynAdjust Project File (c.f. Chapter 2) is to be used, type `dynadjust` on the command line, then `--project-file` (or its shortcut `-p`) followed by the full path for the project file, and then the option to invoke **adjust**:

```
dynadjust -p uni_sqr.dnaproj --adjust
```

No other options are required, as **adjust** will be invoked using the options and arguments contained in the project file `uni_sqr.dnaproj`.

8.3.1 Simultaneous mode

As described in previous sections, simultaneous adjustment mode is ideally suited to smaller–sized networks. To adjust a network in simultaneous mode using all default settings, type `adjust` on the command line followed by the network name:

```
adjust uni_sqr
```

Upon running this command, **adjust** will report to the screen a number of items relevant to the project (e.g. name, location of input and output files); user–specified options; the progress of network adjustment including adjustment iterations; brief statistical summary of the adjustment; and final exit status (i.e. success or failure). Using the `uni_sqr` project, Figure 8.1 shows the information reported to the screen upon program execution. By and large, the screen output shown in Figure 8.1 will be similar to the output for all other adjustment modes.

Upon invoking **adjust**, a variety of matrices will be formed such that the station parameter estimates (c.f. 8.15) and uncertainties (c.f. equation 8.16) can be solved. This stage of the adjustment process is reported to the screen as "`+ Preparing for adjustment...` " and can take anywhere from milliseconds to several minutes depending on the size of the network, the number of computations that need to be undertaken to form the normals matrix, and the amount of data that needs to be held in physical memory. If there is insufficient memory to load the network in memory, DynAdjust will report an error, in which case, users may need to revert to the use of phased adjustment mode in either single-thread mode or staged adjustment mode (c.f. §8.3.2).

```
+ Options:
  Network name:                     uni_sqr
  Input folder:                     .
  Output folder:                    .
  Associated station file:          ./uni_sqr.asl
  Associated measurement file:      ./uni_sqr.aml
  Binary station file:              ./uni_sqr.bst
  Binary measurement file:          ./uni_sqr.bms
  Adjustment output file:           ./uni_sqr.simult.adj
  Coordinate output file:           ./uni_sqr.simult.xyz
  Reference frame:                  GDA94
  Epoch:                            01.01.1994
  Geoid model:                      ./ausgeoid09.gsb

+ Simultaneous adjustment mode
+ Preparing for adjustment...  done.
+ Adjusting network...
  Iteration  1, max station corr:   -0.5850 e
  Iteration  2, max station corr:   -0.0023 e
  Iteration  3, max station corr:    0.0000 n
+ Done.
+ Solution: converged after 3 iterations.
+ Network adjustment took 0.06s.
+ Generating statistics... done.
+ Adjustment results:

+-------------------------------------------------------------------------------
  Number of unknown parameters      440
  Number of measurements            1182  (72 potential outliers)
  Degrees of freedom                742
  Chi squared                       627.01
  Rigorous sigma zero               0.845
  Global (Pelzer) Reliability       4.008   (excludes non redundant measurements)
  Chi-Square test (95.0%)           0.901 < 0.845 < 1.104         *** WARNING ***
+-------------------------------------------------------------------------------

+ Serialising adjustment matrices... done.
+ Printing adjusted station coordinates... done.
+ Updating binary station and measurment files... done.

+ Open uni_sqr.simult.adj to view the adjustment details.
```

Figure 8.1: **adjust** progress reporting

Following the preparation of adjustment matrices, **adjust** will commence adjusting the network, iterating as many times as necessary until the iteration threshold (or convergence criteria) is met or the maximum number of iterations has been reached. If either of these conditions fail, **adjust** will report to the screen an error message describing the reason for failure.

The time each iteration will take will vary according to the network size and the computer's computational power, including its processor (type, CPU frequency and architecture), physical memory and operating system memory model (32–bit or 64–bit). For these reasons, it is not possible to provide an estimate of computation progress or time to completion. Only experience with running several network adjustments will give the user a sense for how long an adjustment iteration may take.

Upon the completion of each iteration, the largest difference between the previously estimated station coordinates and the newly estimated station coordinates (denoted as "`max station corr`") will be displayed. These values will be particularly helpful in understanding the agreement between the starting estimates and the measurement system, and may be used to help diagnose problematic adjustments. For instance, a well configured and appropriately weighted measurement system will result in a rapid convergence (e.g. three or less iterations), despite the quality of the initial station coordinate estimates. However, a system with poorly configured station constraints, measurement blunders, insufficient measurements, inappropriate measurement uncertainties and/or wrongly named stations may cause unexpected convergence rates (i.e. very small or large, erratic changes to the maximum station correction values) or a failure to converge at all.

When the largest correction becomes less than the iteration threshold, **adjust** will deem the solution to have converged. Once the solution has converged, **adjust** will compute a range of values and statistics for the output files. This task may take anywhere from milliseconds to several minutes depending on the size of the network and the user–specified options for configuring the output of adjusted measurements and statistics (c.f. §8.3.4).

Provided there are no significant errors, the next block of information will be a summary of statistics. Referring to Figure 8.1, the reported statistics include:

Number of unknown parameters In DynAdjust, every station is treated by default as having three unknown parameters (i.e. x, y, z). Any one or more cardinals may be fixed by the use of station constraints. The number of unknown parameters therefore corresponds to the number (u) of parameters to be estimated, excluding those station coordinates which have been held fixed in the station file. See §3.2.3 for more information on the concept of station constraints and how they are interpreted by DynAdjust.

Number of measurements This value reports the number (n) of measurement components. All measurements except GNSS measurement types are treated as having a single measurement component. Each GNSS baseline and GNSS point is regarded as having three measurement components (i.e. x, y, z). The number of measurement components in a cluster of GNSS baselines or GNSS points will be equal to three times the number of GNSS baselines or points in the cluster. If a measurement correction exceeds the specified confidence interval and thereby fails the local test, the measurement will be flagged as a potential outlier (c.f. Figure 8.1 where 72 measurements have been flagged as potential outliers). In such cases, the adjustment output file should be inspected to identify whether the potential outlier is a genuine failure and to rectify the cause of that failure. Details on the local test are given in §9.3.2.

Degrees of freedom This value (r) is calculated directly from u and n (c.f. equation 8.3), and is the parameter against which the global rigorous sigma zero value is tested. As described earlier, u excludes all fixed station coordinates.

Chi squared This quantity (w) is calculated directly from the supplied measurement precisions and the measurement residuals of the least squares adjustment (c.f. equation 8.12). With a well–configured measurement system, w will approach r.

Rigorous sigma zero This value ($\hat{\sigma}^2$) is derived from w and r (c.f. equation 9.7) and is used to test the least squares solution as a whole (c.f. §9.3.1).

Global (Pelzer) Reliability This quantity (T^2) is calculated using equation 7.67 according to Pelzer's measurement reliability criterion. It is intended to give an assessment of the reliability of the measurement system as a whole.

Chi–Square test (95.0%) This statement reports the result of the global test of $\hat{\sigma}^2$ using the default or–user specified confidence interval (c.f. §8.3.3, page 124). This test will be explained in §9.3.1. Note that it is possible for a least squares adjustment to have passed this global test and at the same time, one of the measurements in the set to have failed the local test. For this reason, the adjustment output file should be inspected in conjunction with this statistic.

Users should become acquainted with the statistical indicators presented in this summary and how they are calculated in order to make an informed decision about the quality of the solution and any measurements flagged as outliers.

The final statements reported in the screen output include references to various binary and text files that are created or updated with the latest adjustment results. The primary output file which contains the adjustment results is the adjustment output file. This file will be named using the convention `<network-name>.<adjustment-mode>.adj` (e.g. `uni_sqr.simult.adj`). The file content, format and structure of the adjustment output file is described in Appendix C.7. Hints on interpreting the adjusted measurements and associated statistics contained in this file will be given in Chapter 9.

By default, the adjustment output file contains the global adjustment statistics (as shown in the screen output) and estimated station coordinates. Further information on configuring the information printed to this file is given in §8.3.4.

8.3.2 Phased adjustment mode

When adjusting large adjustments in phased adjustment mode, the total network adjustment task is broken up into several smaller adjustments which are undertaken in an ordered sequence. The only pre–requisite for phased adjustment mode is that the network has been segmented (c.f. Chapter 6). There are three primary ways in which phased adjustment mode can be executed — single–thread mode, multi–thread mode and Block–1 only mode. In addition, when running an adjustment in single–thread mode, staged adjustment mode may be used. Brief mention is made of these modes to help determine which mode is appropriate for the network and the computer on which it is being adjusted.

The processing of single–threaded phased adjustments is illustrated in Figure 8.2. In single–thread mode, all adjustment operations are executed in a sequential (or procedural) fashion on one thread managed by a single core. If a computer's CPU has two cores, only 50% of the computer's processing power will be utilised. Similarly, if the CPU has four cores, only 25% will be used.

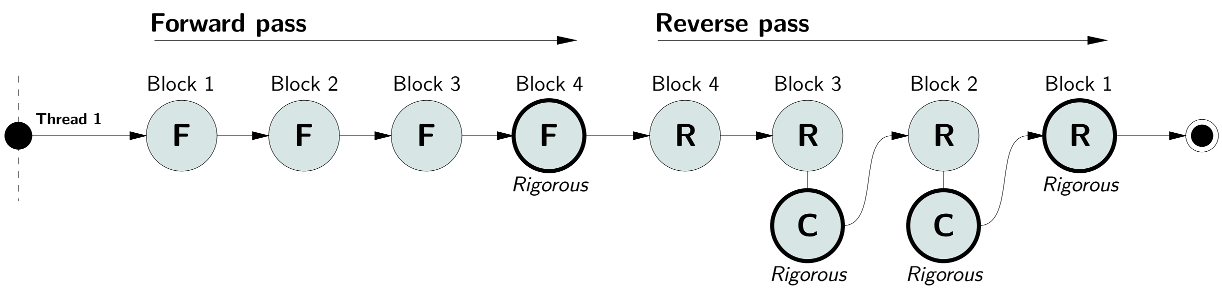


Figure 8.2: Single–thread mode

Information pertaining to the entire network is managed in physical memory (RAM). If there is insufficient RAM to retain the entire network in memory, staged adjustment mode should be used. In this mode, DynAdjust uses *memory mapping* — a technique which treats the hard disk as if it were physical memory. Due to the I/O operations that must be performed to exchange information between the hard disk and memory, this mode may take longer than single–thread mode, however it permits the adjustment of extremely large networks on computers with small amounts of RAM.

The processing of multi–threaded phased adjustments is illustrated in Figure 8.3. In this mode, the forward and reverse pass adjustments are executed on two separate threads concurrently. If a computer's CPU has at least two cores, these threads will be executed in parallel by the operating system on two cores. When a block has been adjusted by both forward and reverse adjustments, a new thread will be invoked to undertake the combination adjustment of that block. This new thread will remain active for as long as it takes to adjust that block.

If a computer has at least four cores, all concurrent threads will be executed in parallel which will achieve the fastest adjustment time. Hence, running adjustments in multi–thread mode on a quad core CPU will result in 100% utilisation of the computer's processing capability. With hyper–threading technology (now almost a standard commodity in all modern computers), extreme performance gains can be achieved in this mode.

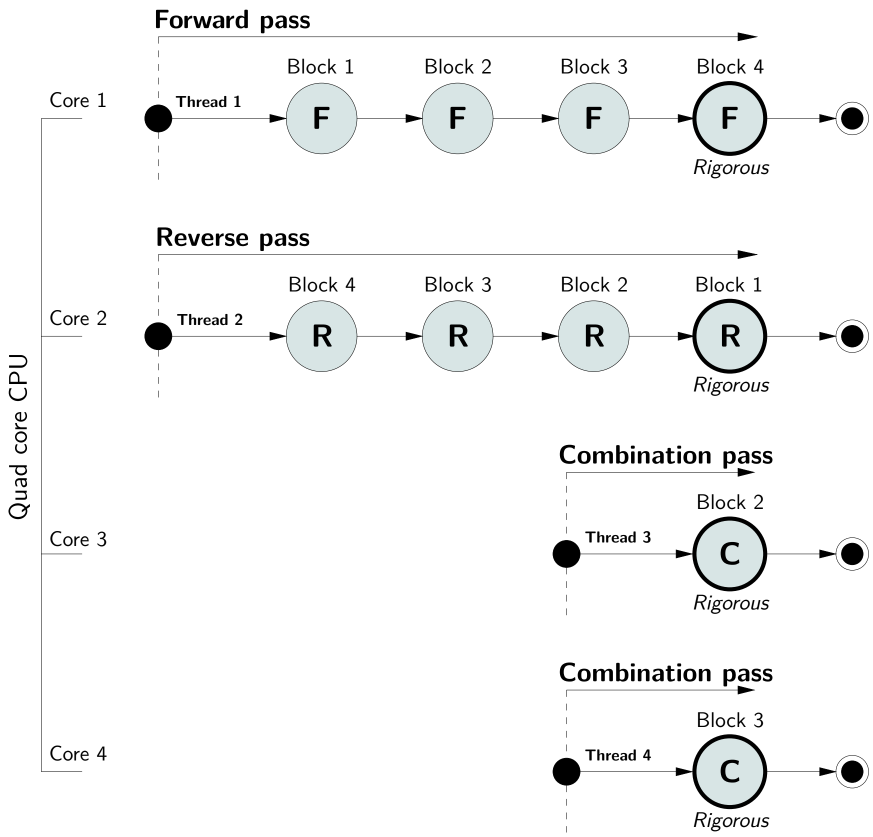


Figure 8.3: Multi–thread mode on four cores

If the number of concurrent threads exceeds the number of cores available on a CPU, the processing of individual tasks by those threads will be *interleaved* (or time–sliced) on the available core(s) such that all tasks appear to advance in parallel. The concept of interleaving between two threads on a single core is shown in Figure 8.4.

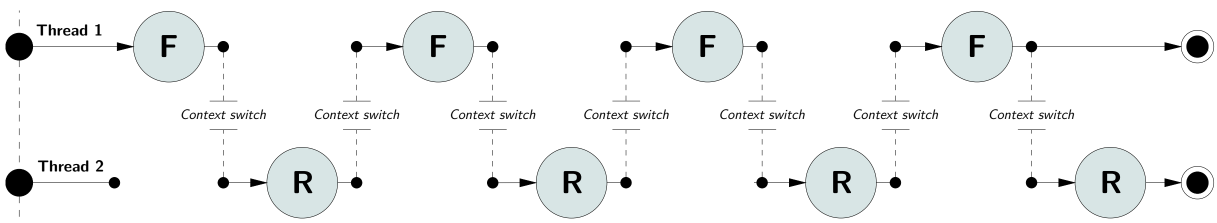


Figure 8.4: Thread switching on a single core

When interleaving from one task to the other, the CPU must perform a *context switch*, which involves storing and recalling the state of each process. Depending on the size of the blocks, these operations can be computationally intensive. Therefore, running multi–thread adjustments on a single core CPU may take longer than the time required for a simultaneous or single–threaded phased adjustment due to the additional time that is needed to switch from the adjustment of one block to another. Note that since the adjustment of a single block involves numerous sub–tasks, context switching can occur at any stage of the adjustment.

The concept of a Block–1 only phased adjustment is illustrated in Figure 8.5. In this case, all blocks are adjusted once in a sequential fashion on a single thread. Since rigorous coordinates are required for the first block only, forward and combination passes are not required. Therefore, only block 1 will contain rigorous coordinates as it will be the only block that will take contribution from all other blocks in the network.

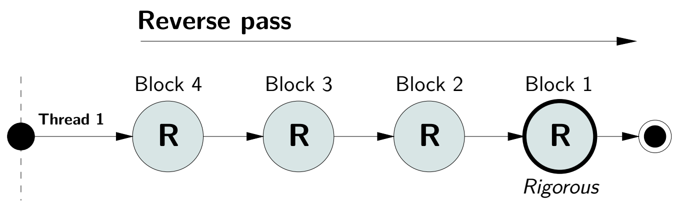


Figure 8.5: Block–1 only mode

Block–1 only phased adjustment mode is by far the most efficient if only a subset of the station coordinate estimates (i.e. those which have been selected to appear in the first block) are required. However, care must be taken to ensure that the most rigorous coordinates are available for all stations in the network prior to running an adjustment. The reason for this is because only a single reverse pass is performed and no further iterations will be undertaken if the solution has not converged.

8.3.2.1 Single–thread mode

To adjust a network using phased adjustment in single–thread mode, type `adjust` on the command line followed by the network name, and then `--phased-adjustment`:

```
adjust uni_sqr --phased-adjustment
```

Figure 8.6 shows the information reported to the screen pertinent to executing an adjustment in single–thread mode (omitting information common to the output from a simultaneous adjustment). As shown in the screen output, an adjustment output file named `uni_sqr.phased.adj` will be created, the contents of which will be very similar to the simultaneous adjustment output.

The main difference in the behaviour of this mode over that of simultaneous mode is that the progress (“`+ Adjusting network...`”) will indicate which block is being adjusted and thereby, provide a more realistic indicator of the rate at which the adjustment is progressing. Since each block size will vary according to the behaviour of the network segmentation, it is not possible to give an estimate of time to completion.

```
+ Options:
  ...

  Adjustment output file:          ./uni_sqr.phased.adj
  Coordinate output file:          ./uni_sqr.phased.xyz
  ...

+ Rigorous sequential phased adjustment mode

+ Preparing for a 5 block adjustment...  done.
+ Adjusting network...
  ...

+ Done.
+ Successfully adjusted 5 blocks.
  ...

+ Open uni_sqr.phased.adj to view the adjustment details.
```

Figure 8.6: Progress reporting using single–thread adjustment mode

8.3.2.2 Multi–thread mode

To adjust a network using phased adjustment in multi–thread mode, type `adjust` on the command line followed by the network name, and then `--multi-thread`:

```
adjust uni_sqr --multi-thread
```

Figure 8.7 shows the information reported to the screen pertinent to executing an adjustment in multi–thread mode.

```
+ Options:
  ...

  Adjustment output file:          ./uni_sqr.phased-mt.adj
  Coordinate output file:          ./uni_sqr.phased-mt.xyz
  ...

+ Rigorous sequential phased adjustment mode
+ Optimised for concurrent processing via multi-threading.

+ The active CPU supports the execution of 8 concurrent threads.

+ Preparing for a 5 block adjustment...  done.
+ Adjusting network...
  ...

+ Done.
+ Successfully adjusted 5 blocks.
  ...

+ Open uni_sqr.phased-mt.adj to view the adjustment details.
```

Figure 8.7: Progress reporting using multi–thread adjustment mode

When reporting the progress of the adjustment, information on iterations following ("`+ Adjusting network...`") will indicate which block is being adjusted by the combination pass, not the forward and reverse passes. Since a combination pass on a block is invoked only once both forward and reverse passes have completed their adjustments of that block, the order of blocks adjusted by the combination passes will appear to be random. The exact order is subject to the order in which forward and reverse adjustments are completed and the way in which the operating system marshals a list of threads waiting in a queue.

8.3.2.3 Block–1 only mode

To adjust a network using phased adjustment in Block–1 only mode, type `adjust` on the command line followed by the network name, and then `--block1-phased`:

```
adjust uni_sqr --block1-phased
```

Figure 8.8 shows the information reported to the screen pertinent to executing an adjustment in Block–1 only mode. As described earlier, this mode is ideally suited to the situation where final coordinate estimates for all stations in the network have already been produced, and there is a requirement for the full variance matrix for a subset of stations in the network.

```
+ Options:
  ...

  Adjustment output file:          ./uni_sqr.phased-block1.adj
  Coordinate output file:          ./uni_sqr.phased-block1.xyz
  ...

+ Sequential phased adjustment resulting in rigorous estimates for Block 1 only

+ Preparing for a 5 block adjustment...  done.
  ...

+ Solution: estimates solved for Block 1 only.

- Warning: Depending on the quality of the apriori station estimates, further
  iterations may be needed. --block1-phased mode should only be used once
  rigorous estimates have been produced for the entire network.
  ...

+ Open uni_sqr.phased-block1.adj to view the adjustment details.
```

Figure 8.8: Progress reporting using Block–1 only adjustment mode

8.3.2.4 Staged adjustment mode

In staged adjustment mode, **adjust** relies upon the hard disk, rather than RAM, to store the adjustment matrices and estimated results for the entire network. Only the block being adjusted is loaded into memory. To this end, it is necessary to inform **adjust** whether to create a set of binary matrix files or to read from existing matrix files created from a previous adjustment. To adjust a network for the first time using phased adjustment in staged mode whereby new binary matrix files must be created, type `adjust` on the command line followed by the network name, and then `--staged-adjustment` and `--create-stage-files`:

```
adjust uni_sqr --staged --create-stage-files
```

Upon running the command above, **adjust** will serialise all matrix and vector information required to execute an adjustment in files with a `.mtx` extension. All the necessary information to re–run the adjustment will be contained within the `.mtx` files. If a repeat adjustment is to be executed using the information serialised from a previous staged adjustment, the option `--create-stage-files` is not required. For example, the following command:

```
adjust uni_sqr --staged
```

will cause **adjust** to look for and open files named `uni_sqr-*.mtx`. Any failure associated with opening or reading these files will cause **adjust** to terminate prematurely.

8.3.2.5 Report results mode

To simply regenerate the results of a previous adjustment, type `adjust` on the command line followed by the network name, and then `--report-results` and the mode in which the previous adjustment was run (e.g. `--phased-adjustment`). If the mode is not supplied, **adjust** will assume simultaneous by default:

```
adjust uni_sqr --report-results
```

During this mode, the results can be re–produced at any time and using different output options. Refer to §8.3.4 for details on configuring the output of the results of an adjustment. The critical thing to remember when running report results mode is to supply the adjustment mode in which the last adjustment was run.

8.3.3 Adjustment configuration options

adjust permits the configuration of certain parameters which effect the behaviour of an adjustment independent of the adjustment mode. These include the confidence interval for testing the global statistic and measurement corrections; the least squares convergence threshold to test whether further iterations are required; the maximum number of iterations; additional station constraints; the level of precision by which fixed and free stations are constrained; and an option to scale the normal matrix prior to computing its inverse. The full command line reference for configuring adjustment behaviour is given in Table A.24.

Confidence interval

Confidence intervals represent the range of probability within which a parameter will equal the true value (c.f. §7.3.1 and §7.3.3.1). Based upon the assumption that a set of measurements will be Normally distributed random variables, and that the corrections represent the random error in those measurements (c.f. equation 8.6), assessments of the quality of a measurement set can be undertaken using the Normal distribution. The criteria by which a measurement is regarded as having passed or failed, is whether or not the measurement correction is within a probabilistic confidence interval. Hence, it is imperative that an appropriate confidence interval be chosen if assertions are to be made about the reliability of measurements and estimated quantities. The theoretical basis and application of statistical testing of measurements is given in more detail in Chapter 9.

The default interval used by **adjust** for all testing is the 95% confidence interval. To change the confidence interval to another value, such as 68.3% (or one sigma), pass the option `--conf-interval` followed by the interval as a percentage:

```
adjust uni_sqr --conf-interval 68.3
```

Note that reducing the confidence interval implies a more stringent testing criteria is required. It means decreasing the probability that measurements will fall within an acceptable range of uncertainty and thereby, less probability that measurements will be deemed reliable. For example, testing the `uni_sqr` network with a 68.3% confidence interval leads to the identification of 220 potential outliers, whereas a 99.73% (or three sigma) identifies only 22 potential outliers. As shown from these two intervals, the number of outliers that exist in an adjustment largely depends on how stringent a probability test is imposed.

Iteration threshold and maximum number of iterations

Since the functional model adopted for the least squares solution is a linearised model (approximated using a first order Taylor series), a solution will often require iteration until the parameter estimates converge to the most probable values. Or, more accurately, until the least squares normal equations are satisfied (c.f. equation 8.14). Depending on the quality of the initial station coordinates and the quality of the measurements (both as to their accuracy and their stated precision), the solution may require several iterations before equality is reached. One way of limiting excessive or unnecessary iterations from being undertaken is to impose a threshold by which the normal equations are deemed to have been satisfied.

The default iteration threshold used by **adjust** is 0.5mm. This means that the solution will keep iterating while corrections to the station coordinates exceed this amount, and until the maximum number of iterations has been reached. To change the convergence threshold to another value, such as 1mm, pass the option `--iteration-threshold` followed by the value in metes:

```
adjust uni_sqr --iteration-threshold 0.001
```

Care should be exercised when increasing this amount as too large a value may mean that incorrect (or less than acceptable) estimates will be derived.

The default maximum number of iterations **adjust** will undertake before terminating is 10. Under normal circumstances where the measurements are of good quality and the starting coordinates are close to their most probable values, this number will be more than adequate. However, in circumstances where a system is defective and troubleshooting is required, this iteration count can be set to, for example, 1 (one) to force an early termination. To change the maximum iteration limit, pass to **adjust** the option `--max-iterations` followed by the desired value:

```
adjust uni_sqr --max-iterations 3
```

Additional station constraints and default constraint values

adjust affords a capacity to introduce station constraints to an adjustment in addition to those which were obtained from the input station file(s). If it is decided that a station constraint should be added, modified or removed, users can specify the station and how it is to be constrained via the option --constraints. Using station 5000 as an example, the following call to **adjust** demonstrates how to constrain this station in the horizontal axis:

```
adjust uni_sqr --constraints 302513640,CCF
```

This option replicates the functionality to apply station constraints supplied in the input station file (c.f. §3.2.3) . Therefore, the order of the three characters in the user–supplied constraint string corresponds to the cardinals of the supplied station coordinate. Since station 5000 was supplied in the form of UTM coordinates, the constraints CCF correspond to Easting, Northing and height up respectively.

Whilst this method of imposing constraints is not the recommended means for constraining a network to a datum, it does provide a useful mechanism for the detection of outliers and analysis of problematic adjustments.

By default, values of 10.0 m and 0.001 mm are adopted for *free* ('F') and *constrained* ('C') station constraints respectively. To alter these values, pass the options --free-stn-sd and/or --fixed-stn-sd to **adjust**:

```
adjust uni_sqr --free-stn-sd 5.0 --fixed-stn-sd 0.003
```

Scaling the normal matrix to unity

At times, computing the inverse of the normal matrix will be problematic if there is a very large numerical difference across the normal matrix elements, or if the matrix tends towards singularity due to very small quantities. This difference can arise from large variations in measurement precisions and/or linearisations which generate very large and very small quantities. These characteristics can cause numerical instability and/or loss of precision at the time of computing the inverse of the normal matrix. A method of overcoming such problems is to scale the normal matrix to unity prior to computing the inverse. To cause the normal matrix to be scaled to unity prior to computing the inverse, pass the option --scale-normals-to-unity to **adjust**:

```
adjust uni_sqr --scale-normals-to-unity
```

Note that this action is only to avoid instability in inverting the various variance matrices. The effect is reversed after matrix inversion so that the computed values are as if no scaling has taken place. This option will most likely only be required for cases where there is a very large variation in the precisions supplied for the set of measurements.

8.3.4 Output configuration options

By default, the adjustment output file (*.adj) contains basic adjustment statistics and adjusted coordinates using all default settings. Optionally, users can activate and configure the output and format of results that **adjust** generates from an adjustment. The full command line reference for configuring the adjustment output is given in Table A.26.

Measurements to station summary

To output a table of measurements connected to each station, add `--output-msr-to-stn` to the command line arguments for **adjust**:

```
adjust uni_sqr --output-msr-to-stn
```

This command will produce a table commencing with the title block `Measurements to Station` that shows the frequency of each measurement type and the total number of measurements associated with each station. The table will be sorted row–wise by the station name, and column–wise by the measurement type (c.f. Table 3.2). Appendix C.7.2 mentions the output of the measurements to stations table to the adjustment output file. For details on the content, structure and format of the table, refer to Table C.1 in Appendix C.3. Generating this table can be quite useful when attempting to identify measurement conflicts, or the presence of unnecessary measurements, measurement constraints or station constraints.

Printing and configuring adjusted measurements

To output a table showing the adjusted measurements and their associated statistics, add the option `--output-adj-msr` to the command line arguments. The table will commence with the title block `Adjusted Measurements` and will contain several columns containing information derived from the adjustment. Appendix C.7.3 provides further details on the content, structure and format of the adjusted measurements table.

Upon supplying this command, several options are available for configuring the adjusted measurement table. These include changing the units (Cartesian, local, polar) in which adjusted GNSS baselines are printed (via `--output-adj-gnss-units`); printing the t statistic (via `--output-tstat-adj-msr`); printing the optional database IDs (via `--output-database-ids`); sorting the adjusted measurements (via `--sort-adj-msr-field`); printing the adjusted measurements according to the segmented blocks (via `--output-msr-blocks`, for phased adjustments only); changing the precision to which linear and angular measurements are printed (via `--precision-msr-linear` and `--precision-msr-angular`); and altering the type and format in which angular measurements are printed (via `--angular-msr-type` and `--dms-msr-format`).

For each of these options, the `--output-adj-msr` option must be provided (either before or after the additional options). The configuration options can be supplied in any order.

For example, the following records show adjusted output information for a GNSS baseline in the default cartesian reference frame (station information excluded):

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
Adjusted Measurements
------------------------------------------

...    C            Measured            Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD     N-stat  Pelzer Rel  Pre Adj Corr
...    ----------------------------------------------------------------------------------------------------------------------------------
...    X         -10325.4980         -10325.5078     -0.0098       0.0661       0.0494       0.0440      -0.22        1.50        0.0000
...    Y          39327.8660          39327.8795      0.0135       0.0487       0.0365       0.0322       0.42        1.51        0.0000
...    Z          44763.3780          44763.3639     -0.0141       0.0593       0.0447       0.0389      -0.36        1.52        0.0000
```

The following records show the corresponding adjustment information in the local reference frame via the `--output-adj-gnss-units` option:

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
Adjusted Measurements
------------------------------------------

...  C            Measured           Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD     N-stat  Pelzer Rel  Pre Adj Corr
...  ------------------------------------------------------------------------------------------------------------------------------------
...  e         -25623.1954        -25623.2005     -0.0051       0.0112       0.0084       0.0073      -0.70        1.53        0.0000
...  n          54776.0622         54776.0599     -0.0022       0.0130       0.0099       0.0085      -0.26        1.54        0.0000
...  u           -303.0373          -303.0162      0.0211       0.0998       0.0748       0.0660       0.32        1.51        0.0000
```

The following records show the output information having also passed the `--precision-msr-linear` option with a value of 3 to adjust:

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
Adjusted Measurements
------------------------------------------

...  C            Measured           Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD     N-stat  Pelzer Rel  Pre Adj Corr
...  ------------------------------------------------------------------------------------------------------------------------------------
...  e          -25623.195         -25623.201      -0.005        0.011        0.008        0.007      -0.70        1.53        0.0000
...  n           54776.062          54776.060      -0.002        0.013        0.010        0.008      -0.26        1.54        0.0000
...  u            -303.037           -303.016       0.021        0.100        0.075        0.066       0.32        1.51        0.0000
```

Information on interpreting the adjusted measurement statistics is provided in §9.3.2.

Configuring adjusted coordinates

By default, **adjust** prints the adjusted coordinates to the adjustment output file. The options available for configuring the coordinate output include printing the adjusted station coordinates in sections according to the segmented blocks (via `--output-stn-blocks`, for phased adjustments only); sorting the stations on the sort order found in the input file (via `--sort-stn-order-orig`); selecting which coordinate types should be printed (via `--stn-coord-types`); printing the optional corrections to the original station coordinates (via `--stn-corrections`); and changing the precision in which linear and angular station coordinate values are printed (via `--precision-stn-linear` and `--precision-stn-angular`).

The following sample records show adjusted station coordinate output information having also passed to adjust the `--precision-stn-linear` option with a value of 3, the `--stn-coord-types` option followed by `ENzH`, and the `--stn-corrections` option (omitting station names and descriptions):

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
Adjusted Coordinates
------------------------------------------

...  Const       Easting       Northing    Zone   H(Ortho)       SD(e)      SD(n)    SD(up)       Corr(e)     Corr(n)    Corr(up)  ...
...  ------------------------------------------------------------------------------------------------------------------------------------
...  FFF      751542.610    6066631.957      54     88.056       2.774      2.774     2.775        -0.027       0.009       0.056  ...
...  FFF      668103.768    6069681.066      54     56.127       2.774      2.774     2.774        -0.021       0.020       0.043  ...
...  FFF      609169.911    6181529.819      54     62.476       2.774      2.774     2.774        -0.012      -0.004      -0.009  ...
...  FFF      722194.458    6055677.636      54    123.898       2.774      2.774     2.774        -0.024       0.016       0.036  ...
```

Appendix C.7.4 mentions the output of the adjusted station coordinates to the adjustment output file. For details on the content, structure and format of this information, refer to Table C.6 in Appendix C.6.

Adjustment results at each iteration

Various adjustment results can be generated upon each iteration to help identify and solve adjustment failure, such as excessive measurement corrections or a failure to converge. The results that can be printed to the output file upon each iteration include the adjusted station coordinates (via `--output-iter-adj-stn`); the summary of adjustment statistics (via `--output-iter-adj-stat`); the measurements computed prior to an adjustment (via `--output-iter-cmp-msr`); and the adjusted measurements (via `--output-iter-adj-msr`). Refer to Appendix C.7.6 for further information on the behaviour of this output.

Printing ignored measurements

Measurements which have been marked as ignored in the input measurement files (c.f. §B.1.4, §B.3.2) will be exluded from all subsequent processing and adjustment. At times, it can be extremely advantageous to see how ignored measurements might perform in an adjustment without being included in an adjustment. For this purpose, ignored measurements can be printed to the output file using the `--output-ignored-msrs` option. When this option is provided, each ignored measurement will be tabulated in the adjustment output file with a corresponding measurement computed from the rigorous (a–posteriori) station coordinate estimates, the difference between the measured and computed measurements and any applicable pre–adjustment corrections (e.g. geoid–ellipsoid separation, deflection of the vertical or Laplace correction). This feature can prove invaluable when attempting to evaluate troublesome measurements or measurements with doubtful or unknown measurement precisions.

The following records show the output of ignored measurements to the adjustment output file. Refer to Table C.9 in Appendix C.7.5 for more information on the structure and content of this table.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
Ignored Measurements (a-posteriori)
------------------------------------------

M Station 1        ...           *  C            Measured           Computed  Difference     Meas. SD  Pre Adj Corr
--------------     ...           ----------------------------------------------------------------------------------
H 276400801        ...           *               162.4102           162.4000     -0.0102       0.3000        2.8346
D 246000120        ...           *  5
                   ...                      22 01 30.7400      22 01 30.8377      0.0977       0.9998       -0.0032
                   ...                      62 39 34.6700      62 39 34.6853      0.0153       0.9998       -0.0066
                   ...                      68 36 00.3960      68 36 00.3893     -0.0067       1.1500       -0.0003
                   ...                     119 02 43.8250     119 02 43.7639     -0.0611       1.9330        0.0070
                   ...                     128 32 00.1460     128 32 00.1630      0.0170       2.5594        0.0027
```

There will be occasions when measurements marked as ignored in the input files will not appear in the list of ignored measurements in the adjustment output file. This behaviour occurs when an ignored measurement is connected to stations that are not connected to any other measurement, or stations which are only connected to ignored measurements. Stations of this nature are marked as *unused* (c.f. Station and measurement connectivity in §3.2.7) and cannot be estimated. Accordingly, DynAdjust will exclude all ignored measurements connected to these stations. To identify any stations which are unused, whether as a result of not being connected to measurements or being connected by ignored measurements only, see §3.6.2 for exporting the Associated Stations List (ASL).

8.3.5 Export configuration options

adjust provides a capacity to export the results of an adjustment into a variety of different files and formats. These include the adjusted positional uncertainty (`*.apu`) file; the station corrections file (`*.cor`); and the DNA (`*.stn` and `*.msr`), DynaML (`*.xml`) and SINEX (`*.snx`) file formats.

Full uncertainty information

To produce a file containing the rigorous uncertainties of the estimated parameters, pass the option `--output-pos-uncertainty` to **adjust**:

```
adjust uni_sqr --output-pos-uncertainty
```

Running this command will produce a file named `uni_sqr.simult.apu`. Similar files will be produced for the various phased adjustment modes. By default, this file will contain the adjusted coordinates; horizontal PU; vertical PU; error ellipse semi–major (a), semi–minor (b) and orientation (ϕ); and the upper–triangular component of the (3x3) variance matrix for each station. To output the covariances for all stations, add the option `--output-all-covariances`:

```
adjust uni_sqr --output-pos-uncertainty --output-all-covariances
```

If the above command is executed in phased adjustment mode, the uncertainty information will be printed in sections according to the segmented blocks. By default, the variance matrix is in the cartesian reference frame. To output the covariances in the local reference frame, pass the option `--output-apu-vcv-units` followed by 1:

```
adjust uni_sqr --output-pos-uncertainty --output-apu-vcv-units 1
```

Appendix C.9 provides further details on the content, structure and format of this file.

At times, there may be a need for the latest station coordinate estimates and a full variance matrix for a specific subset of stations from a large geodetic network, which, owing to the size of the network, must be adjusted in phased adjustment mode. This need can be met via the Block–1 only phased adjustment mode, whereby **adjust** is supplied the uncertainty output options described above. For example, the following command:

```
adjust uni_sqr --block1-phased --output-pos-uncertainty --output-all-covariances
```

will write the latest station coordinate estimates to `uni_sqr.phased-block1.adj` and the full variance matrix to `uni_sqr.phased-block1.apu`.

Station corrections

To produce a file containing the corrections to the original station coordinates, pass the option `--output-corrections-file` to **adjust**:

```
adjust uni_sqr --output-corrections-file
```

Running this command will produce a file named `uni_sqr.simult.cor`. Similar files will be produced for the various phased adjustment modes. This file will contain the adjusted station coordinate corrections in terms of horizontal azimuth; vertical angle; slope distance; horizontal distance; and the three-dimensional shifts in the local reference frame (east, north and up). Appendix C.8 provides further details on the content, structure and format of the station coordinate corrections file.

To restrict the information printed to the corrections file to those corrections which exceed a horizontal and/or vertical threshold, pass the --hz-corr-threshold and/or --vt-corr-threshold options followed by the respective thresholds. For example, to restrict the output of corrections to those which are larger than 25 mm horizontally *and* 25 mm vertically, run the following command:

```
adjust uni_sqr --output-corrections-file --hz-corr 0.025 --vt-corr 0.025
```

Exporting adjusted station coordinates to DynaML, DNA and SINEX

Adjusted station coordinates can be exported to DynAdjust XML (DynaML), DNA and/or SINEX formats. These export functions are invoked by **adjust** upon adding the command line options --export-xml-stn-file, --export-dna-stn-file and/or --export-sinex-stn-file respectively. All three may be passed at the same time, as:

```
adjust uni_sqr --export-xml-stn-file --export-dna-stn-file --export-sinex-stn-file
```

Appendix B provides the file format specification for the DynaML, DNA and SINEX file types.

Adjusted station coordinates exported from **adjust** can be imported in subsequent adjustments as a–priori coordinate information. Importing adjusted station coordinates into an adjustment offers the advantage of reducing the number of iterations. The reason DynAdjust does not support the export of adjusted measurement information is that properly configured network adjustments should not be based upon adjusted measurements and adjusted measurement precisions, but rather, reduced (or processed) and appropriately weighted measurements.

When exporting adjustment results to SINEX format from a phased adjustment, an independent SINEX file will be created for each block using the convention <network-name>-block<n>.<frame>.snx. Independent files are created for each block of a phased adjustment since the SINEX standard does not permit multiple blocks of data in a single file. For each file created, <n> corresponds to the block number and <frame> corresponds to the adjustment reference frame. Each file will contain the full variance matrix for the subject block. To assist with managing the information contained in separate SINEX files, brief comments are provided in the FILE/REFERENCE and FILE/COMMENT sections. The following example shows the preliminary sections from the first of two SINEX files produced for the trivial GNSS network shown in Figure 6.2.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
%=SNX 2.00 DNA 16:320:28126 DNA 94:001:00000 94:001:00000 P 00039 0 S
*-------------------------------------------------------------------------------
+FILE/REFERENCE
*INFO_TYPE_________ INFO________________________________________________________
 DESCRIPTION        Network 50kex
 OUTPUT             Phased adjustment results. Block 1 of 2
 SOFTWARE           dynadjust, Geodetic network adjustment software
 SOFTWARE           Version:      3.2.6, Release (64-bit)
 SOFTWARE           Build:        Nov  2 2016, 17:41:32 (MSVC++ 10.0)
 HARDWARE           Win 64
 INPUT              50k_exgpsmsr.xml
 INPUT              50k_exgpsstn.xml
-FILE/REFERENCE
*-------------------------------------------------------------------------------
+FILE/COMMENT
 This file contains the rigorous estimates for block 1 of a segmented
 network comprised of 2 blocks. Due to the way in which junction stations
 are carried through successive blocks, stations appearing in this file
 may also be found in other SINEX files relating to this network, such as
 50kex-block1.snx, 50kex-block2.snx, etc.
-FILE/COMMENT
*-------------------------------------------------------------------------------
+SITE/ID
*CODE PT __DOMES__ T _STATION DESCRIPTION__ APPROX_LON_ APPROX_LAT_ _APP_H_
 2120  A 212000820 P TUTCHEWOP              143 46 25.0 -35 30 43.4    94.9
 2691  A 269100210 P POLA                   143 27 11.9 -35 37 03.8   130.1
 4097  A 409700110 P GENOE                  143 29 34.3 -35 01 04.8    79.3
-SITE/ID
*-------------------------------------------------------------------------------
+SOLUTION/STATISTICS
*_STATISTICAL PARAMETER________ __VALUE(S)____________
 NUMBER OF OBSERVATIONS                             90
 NUMBER OF UNKNOWNS                                 39
 NUMBER OF DEGREES OF FREEDOM                       51
 VARIANCE FACTOR                              0.141551
-SOLUTION/STATISTICS
...
```

If the adjustment contains a station name exceeding the maximum four–character code limit imposed by the SINEX standard, a new file with the extension `.snx.err` will be created. This file will contain an error statement for every station exceeding four characters in length. For example, the following shows the `.snx.err` file corresponding to the (block 1) SINEX file shown above.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
--------------------------------------------------------------------------------
DYNADJUST SINEX OUTPUT WARNINGS FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Nov  2 2016, 17:41:32 (MSVC++ 10.0)
File created:                      Tuesday, 15 November 2016, 7:48:46 AM
File name:                         c:\Data\dist\50kex-block1.GDA94.snx.err
--------------------------------------------------------------------------------

Station name 212000820 exceeds four characters.
Station name 269100210 exceeds four characters.
Station name 409700110 exceeds four characters.
```

To overcome this problem, users should consider the use of a renaming file (c.f. §3.3.2, page 28) when executing **import** to properly handle station names greater than four characters.

Chapter 9

Estimating uncertainty and testing least squares adjustments

9.1 Introduction

The previous chapter summarised the least squares estimation theory implemented within **adjust** and the various modes of its execution to determine rigorous station coordinates for all stations in a network. The purpose of this chapter is to elaborate on two secondary, yet equally important functions of **adjust** — (1) estimating the precision of the estimated coordinates and adjusted measurements and (2) the computation of statistical indicators for the verification of least squares adjustments and adjustment results.

The first part of this chapter presents the theory for determining the precision of estimated coordinates and statistical indicators generated by **adjust**. The remainder of the chapter explains the meaning of these values, where they appear in the adjustment output file, and how they can be used to identify erroneous measurements, less–than–optimal (i.e. over– or under–optimistic) measurement precisions, invalid constraints and other inconsistencies in the measurement system.

9.2 Algorithms for estimating uncertainty

9.2.1 Precision of the estimated parameters

The precision of the estimated parameters expresses the quality (or uncertainty) of the adjusted station coordinates. For many applications, determining the uncertainty of the adjusted station coordinates is of vital importance and forms the basis for making decisions about the location of real–world objects and their movement over time. So that a high–level of confidence can be placed upon the computed values, it is imperative that the computation of the precision of the estimated parameters is completely rigorous.

The precision of the estimated parameters is directly obtainable from the least squares solution via equation 8.16, repeated here as:

$$\mathbf{V}_{\hat{\mathbf{x}}} = (\mathbf{A}^T\mathbf{V}_{\mathbf{m}}^{-1}\mathbf{A})^{-1} \tag{9.1}$$

For simultaneous adjustments, $\mathbf{V}_{\hat{\mathbf{x}}}$ will contain the precisions for the entire network and will be derived in a single step upon the solution of the normal equations. For phased adjustments, a separate variance matrix will be produced for the parameters in each block ($\mathbf{V}_{\hat{\mathbf{x}}_i}$, $i = 1..n$). Owing to the way in which the phased adjustment algorithm carries the precision of the junction station estimates through to the solution of successive blocks (in forward, reverse and combination adjustments), these variance matrices will be completely rigorous.

The variance matrix $\mathbf{V}_{\hat{\mathbf{x}}}$ is a square, symmetric positive definite matrix. The square dimension of $\mathbf{V}_{\hat{\mathbf{x}}}$ is equal to the number u of parameters being estimated. Since the system of observation equations for the linear functional model $\mathbf{A}$ is required to have functional independence in order to achieve non–singularity in the normal equation matrix, it follows from the evaluation of $(\mathbf{A}^T\mathbf{V}_{\mathbf{m}}^{-1}\mathbf{A})^{-1}$ that the elements of $\mathbf{V}_{\hat{\mathbf{x}}}$ will also have functional independence. Therefore, $\mathbf{V}_{\hat{\mathbf{x}}}$ will be non–singular.

Note that $\mathbf{V}_{\hat{\mathbf{x}}}$ may appear in some texts and least squares adjustment programs as $\mathbf{Q}_{\hat{\mathbf{x}}}$ and termed the *a–posteriori variance–covariance matrix* or *cofactor matrix* which is multiplied by a single variance factor to obtain a more *realistic* variance matrix $\mathbf{V}_{\hat{\mathbf{x}}}$. The scaling of cofactor matrices will be explained in §9.3.1.1.

Spatial correlation between the diagonal precision elements (i.e. the respective uncertainties of the estimated parameters) is held by the off–diagonal elements. Where there is no correlation between the estimated parameters, the off–diagonal elements will be zero.

By default, $\mathbf{V}_{\hat{\mathbf{x}}}$ will be expressed in the cartesian reference frame. When the precision of the estimated parameters is to be expressed in another reference frame (i.e. geographic, local or polar), the law of propagation of variances is applied as described in §4.4.

9.2.2 Precision of the adjusted measurements

The precision of the adjusted measurements $\mathbf{V}_{\hat{\mathbf{m}}}$ is derived by applying the law of propagation of variances to the precision of estimated parameters:

$$\mathbf{V}_{\hat{\mathbf{m}}} = \mathbf{A}\mathbf{V}_{\hat{\mathbf{x}}}\mathbf{A}^T \tag{9.2}$$

$\mathbf{V}_{\hat{\mathbf{m}}}$ is a square matrix, the dimensions of which are equal to the number n of measurements in the system. Where $\mathbf{V}_{\hat{\mathbf{m}}}$ holds values for the precision of estimated parameters of different types (e.g. angular and linear quantities), the corresponding elements will be in the respective units of measure (the square of radians or metres).

Once the precision of the adjusted measurements has been computed, the precision of the corrections to the measurements can be computed as:

$$\mathbf{V}_{\hat{\mathbf{v}}} = \mathbf{V}_{\mathbf{m}} - \mathbf{V}_{\hat{\mathbf{m}}} \tag{9.3}$$

where $\mathbf{V}_{\mathbf{m}}$ represents the a–priori precision of the measurements.

9.3 Testing least squares adjustments

Following the tasks of estimating unknown station coordinates and their uncertainties, it is of primary importance to test the estimated results to validate the measurements, the assumed precision of those measurements, and the reliability of the network as a whole. There are three primary reasons for which testing is essential for the proper management of geodetic networks:

1. It is quite common for the a–priori precision assumed for a measurement to not be *realistic*, whether over– or under–optimistic, giving the impression that the measurements are better or worse than reality (c.f. §7.3.2). This situation frequently arises during the processing of

raw GNSS data to produce GNSS baselines and/or positions and their accompanying variance matrices. In this context, GNSS variance matrices typically represent the internal precision of the GNSS processing and fail to take into account the external influences that degrade the solution. Similarly, manufacturer–stated precisions for distance and angle measuring equipment which have been derived from laboratory–like conditions are not always representative of the performance that is achieved from equipment operating under real–world conditions.

2. It is possible that erroneous, sub–standard measurements (or *outliers*) have been introduced to the system — despite every care being undertaken during observation and measurement reduction. Examples of this are numerous, including an erroneous instrument or GNSS antenna height, an incorrectly transcribed measurement value and station mis–identification. Estimated parameters influenced by outlier measurements will not only be biased as to their physical location, but also as to their estimated quality.
3. When there is a shortcoming in the functional model adopted for the observation equations. A well–known example of this is movement of the station mark which occurs between successive (or repeat) observation campaigns. Movements of this nature yield in an inconsistency in the solution, the inevitable result of which will be measurement failure and error in the station coordinates if the change in the mark's physical location is not taken into account. Failing to deal with such influences in a rigorous way can lead to the false assumption that the precision of older measurements (to the marks which have moved) are no longer representative and therefore need some tweaking or scaling.

DynAdjust cannot provide an automated means for fixing these issues. Rather, it has been designed to assist the user in identifying and resolving the cause of inconsistencies. The following sections outline the statistical indicators calculated by **adjust** to assist the user in these processes.

9.3.1 Testing the least squares adjustment as a whole

Recall from equation 8.12 that the sum of the squares of the weighted corrections that is minimised in the solution of the least squares normal equations is:

$$w \;=\; \mathbf{v}^T \mathbf{V}_{\mathbf{m}}^{-1} \mathbf{v}$$

The minimised quantity w is a Chi–square random variable with r degrees of freedom — written as χ^2_r (c.f. §7.3.1.2). In the context of network adjustment, r is equal to the number of redundant measurements:

$$E\left[w\right] \;=\; r \tag{9.4}$$

$$w \;\sim\; \chi^2_r \tag{9.5}$$

To assist the user with validating the reliability of the least squares solution, **adjust** performs a test to quantify how *significantly* different w is to r. This test is referred to as the *global test* (or *goodness–of–fit test*), since it tests the least squares solution as a whole. The global test is undertaken immediately following the solution of the normal equations and calculation of the set of corrections $\mathbf{v}$. Whilst equality between w and r is the desired outcome, the estimation can be considered statistically reliable if w is within the upper and lower limits (or critical values) of the confidence interval at significance level α. The test for reliability is usually written as:

$$P\left[\chi^2_{1-\alpha/2,\,r} \;\leq\; \hat{\sigma}^2_0 \;\leq\; \chi^2_{\alpha/2,\,r}\right] \;\approx\; \alpha \tag{9.6}$$

where $\hat{\sigma}_0^2$ is the *estimated* variance factor (or variance of unit weight, or sigma zero) obtained from:

$$\hat{\sigma}_0^2 = \frac{w}{r} \tag{9.7}$$

This quantity is particularly useful in estimating the precision of measurements of the same type and will be elaborated on in §9.3.2.

The outcome of the global test, together with the upper and lower limits of the confidence interval to which $\hat{\sigma}_0^2$ is compared, is reported in the last line of the statistical summary printed to the screen and the adjustment output file (`*.adj`) at the completion of an adjustment (c.f. §8.3.1, page 118). For example, the following shows the results of an adjustment where r is `1482`, w is `823.99`, $\hat{\sigma}_0^2$ is `0.556`, and the lower ($\chi^2_{1-\alpha/2,\,r}$) and upper ($\chi^2_{\alpha/2,\,r}$) critical values of the 95% confidence interval are `0.929` and `1.073` respectively. A warning is issued from the global test since the estimated variance factor is less than the lower limit of the confidence interval:

```
+--------------------------------------------------------------------------------
Number of unknown parameters       495
Number of measurements             1977  (22 potential outliers)
Degrees of freedom                 1482
Chi squared                        823.99
Rigorous Sigma Zero                0.556
Global (Pelzer) Reliability        12.886  (excludes non redundant measurements)

Chi-Square test (95.0%)            0.929 < 0.556 < 1.073                          *** WARNING ***
+--------------------------------------------------------------------------------
```

There are three possible outcomes from the global test undertaken by **adjust**:

`PASSED` In this case, the estimated value $\hat{\sigma}_0^2$ is within the lower and upper limits of the confidence interval at probability α. Whilst unity is desirable, any value within the lower and upper limits can be regarded as statistically reliable. Users should be aware that it is possible for a least squares adjustment to pass the global test despite the presence of one or more measurements which have failed the local test. For this reason, the number of measurement corrections which exceed the specified confidence interval (in the local test) will be noted in the record which provides the measurement count (e.g. `(22 potential outliers)`).

`FAILED` This means that the calculated quantity for w is significantly greater than the expected value. That is, $\hat{\sigma}_0^2$ has exceeded the confidence interval upper limit of the probability α that w is equal to r. When this test fails, it is clear that an unacceptable number of measurement corrections have failed.

`WARNING` When the estimated value $\hat{\sigma}_0^2$ is less than the lower limit, a warning is issued. In some cases, it means that the set of measurements was better than indicated by the supplied precisions. In this context, it may be implied that the global test has not failed, despite w being significantly different to r. However, a warning is issued to indicate that some of the measurements in the set and their variance matrix require attention. In particular, it indicates that the variances allocated could be too large and as a result, the estimates of the precision of the parameters would suffer.

Elaboration of all possible reasons for which w differs significantly from r will not be given here. However, as discussed earlier, three common reasons for which a failure is most likely to occur are an incorrect a–priori variance matrix $\mathbf{V_m}$, outlier measurements and/or shortcomings in the functional model. Another three reasons why an adjustment might fail include inappropriate station constraints, an incorrect choice of measurement type in the measurement file and/or failure to apply an essential reference frame transformation or geoid correction. It is the responsibility of the user to discern which of these reasons has caused a failure to occur and how to resolve it.

9.3.1.1 Rectifying over–optimistic variance matrices of the same type

As described earlier, it is quite common for a set of GNSS variance matrices to be over–optimistic, giving the false impression that the measurement set is of a higher quality than reality. In this context, it is not that there is a measurement blunder or that there is a bias unaccounted for in the processing, but that the GNSS processing has simply failed to produce a reliable estimate of the *true* measurement precision. Upon performing an adjustment of newly–processed GNSS baselines with over–optimistic variance matrices, under minimal constraints without any other modification, it is not uncommon for the adjustment to fail the global test.

For this reason, the a–priori measurement variance matrix is often referred to as a cofactor matrix $\mathbf{Q_m}$, implying that the *true* variance matrix $\mathbf{V_m}$ is never really known. Based on the fact that $\mathbf{v}^T\mathbf{V_m}^{-1}\mathbf{v}$ is a Chi–square random variable, the *expected* value for $\mathbf{v}^T\mathbf{Q_m}^{-1}\mathbf{v}$ can be written as:

$$E\left[\mathbf{v}^T\mathbf{Q_m}^{-1}\mathbf{v}\right] = E\left[\sigma_0^2\,\mathbf{v}^T\mathbf{V_m}^{-1}\mathbf{v}\right] \tag{9.8}$$
$$= \sigma_0^2\,r \tag{9.9}$$

From this relationship, it can be assumed that an *unbiased* estimate of variance matrix $\mathbf{V_m}$ for measurement sets of the same type can be estimated after the least squares solution by a global matrix scalar operation:

$$\mathbf{V_m} = \hat{\sigma}_0^2\,\mathbf{Q_m} \tag{9.10}$$

where $\hat{\sigma}_0^2$ is the variance factor *estimated* from the adjustment with an a–priori variance matrix $\mathbf{Q_m}$. From this, the *unbiased* variance for each measurement can be determined by:

$$\hat{\sigma}_i^2 = \hat{\sigma}_0^2\,\sigma_i^2 \tag{9.11}$$

where σ_i^2 is the a–priori variance of the original measurement m_i.

To produce an unbiased variance matrix for a new set of GNSS measurements, the following sequence of steps may be applied. For this example, the `skye` network shown in Figure 1.4 will be used. As a general rule for all other networks to which this procedure is to be applied, all stations in the station file should be held free (no constraints). In addition, all measurements in the measurement file should be captured under similar measurement conditions and a similar observation methodology, produced from the same processing strategy, and be of the same category (i.e. baselines only, baseline clusters only, or point clusters only).

Step 1 — Import the data

Using **import**, create a project introducing the required station and measurement files:

```
import -n skye skye.stn skye.msr
```

Step 2 — adjust the network

Call **adjust** using the default settings, effecting the condition of minimum constraints:

```
adjust skye --output-adj-msr
```

Inspect the adjustment output file to ensure the measurement set is free from significant blunders. Make a note of the estimated rigorous sigma zero $\hat{\sigma}_0^2$ value reported in the adjustment statistics summary. For the `skye` network shown in Figure 1.4, the estimated rigorous sigma zero ($\hat{\sigma}_0^2$) shown in the adjustment statistics summary is 2.648, as follows:

```
+------------------------------------------------------------------------------
  Number of unknown parameters     18
  Number of measurements           27  (3 potential outliers)
  Degrees of freedom               9
  Chi squared                      23.84
  Rigorous sigma zero              2.648
  Global (Pelzer) Reliability      1.204   (excludes non redundant measurements)

  Chi-Square test (95.0%)          0.300 < 2.648 < 2.114          *** FAILED ***
+------------------------------------------------------------------------------
```

Step 3 — re–import the data and scale all measurement variance matrices

Using **import** again, create a new project (e.g. `skye-scaled`) introducing the original station and measurement files, with the option `--v-scale` (c.f. §3.3.3) followed by the rigorous sigma zero value obtained from Step 2. At this step, the variance matrix scaling can be exported to new station and measurement files (e.g. via `--export-dna`):

```
import -n skye-scaled skye.stn skye.msr --v-scale 2.648 --export-dna
```

At this point, the value 2.648 will be assigned to the v-scale element in the binary measurement file for every GNSS baseline in the network. Hence, the original variance matrices are retained. Then, when **adjust** is executed again using the command:

```
adjust skye-scaled --output-adj-msr
```

equation 9.10 will be applied to all variance matrices prior to adjustment. The net result from this sequence of steps is a set of a–priori measurement variance matrices which can be considered *unbiased*, and a new rigorous sigma zero closer to unity:

```
+------------------------------------------------------------------------------
  Number of unknown parameters     18
  Number of measurements           27  (1 potential outlier)
  Degrees of freedom               9
  Chi squared                      9.00
  Rigorous sigma zero              1.000
  Global (Pelzer) Reliability      1.204   (excludes non redundant measurements)

  Chi-Square test (95.0%)          0.300 < 1.000 < 2.114          *** PASSED ***
+------------------------------------------------------------------------------
```

This action is generally only performed once and, to maintain the highest rigour in the measurement precisions, should only be used for scaling variance matrices of measurements of the same type, measurement characteristics, category and processing strategy.

9.3.2 Testing for the presence of outliers

Following the global test, **adjust** also undertakes a *local test* on each measurement correction to help detect the presence of individual outliers. There are two general cases which provide adequate justification for testing every measurement. Firstly, if there is a blunder in the set of measurements, it is possible that the measurement correction will be so large that it will, on its own, cause the adjustment to fail the global test and several other reliable measurements to fail the local test. For this reason, all measurements which have failed the local test are denoted *potential* outliers. The user will need to take steps to distinguish the *true* outliers which require attention (e.g. removal, reprocessing, or variance matrix scaling) from the measurements (marked as potential outliers) which are indeed reliable. Secondly, it is equally possible for an outlier to exist amongst the measurements despite the solution passing the global test. In certain situations where there is a high degree of redundancy in the measurement system, the confidence interval may become so small that marginal inconsistencies between the measurement precisions and the corrections will cause a higher likelihood of local test failures. In this context, it may be entirely acceptable to ignore such failures as any modification to the measurement system to achieve a unanimous pass may have a negligible effect on the estimated parameters. In any case, it is the responsibility of the user to understand the nature of the outlier, to identify the cause of failure and thereupon, take steps to address it.

Drawing from equation 8.6, the set of measurement corrections $\hat{\mathbf{v}}$ is evaluated from the original measurements $\mathbf{m}$, estimated parameters $\hat{\mathbf{x}}$ and design matrix $\mathbf{A}$ via the expression:

$$\hat{\mathbf{v}} \;=\; \mathbf{m} - \mathbf{A}\hat{\mathbf{x}} \tag{9.12}$$

Being estimated quantities, the corrections $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_i$ are considered in DynAdjust as Normally distributed random variables, having an *expected* value of zero:

$$E\left[v\right] \;=\; 0 \tag{9.13}$$

$$v \;\sim\; N(0, \sigma_v) \tag{9.14}$$

adjust uses two distributions to assist with testing each correction — the Normal distribution (c.f. §7.3.1.1) and the Student's t distribution (c.f. §7.3.1.3). By default, **adjust** performs all local testing using the Normal distribution at the 95% confidence interval. Optionally, the confidence interval for all statistical testing may be changed by providing the option `--conf-interval` followed by the desired level of significance in the command line call to **adjust**. The user may also undertake local testing using the Student's t distribution upon providing the option `--output-tstat-adj-msr` to **adjust**. These distributions have been selected to help the user in addressing outliers with reliable (or well–known, trusted) and doubtful (or suspect) measurement variances respectively.

9.3.2.1 Testing measurements with reliable estimates of precision

If the a–priori variance matrix $\mathbf{V_m}$ is known or considered to be reliable, the magnitude of the corrections can be satisfactorily tested using the Normal distribution (c.f. §7.3.1.1). The test to verify whether a correction v_i falls within the upper and lower limits of the confidence interval at significance α can be written as:

$$P\left[-z_{(1-\alpha)/2} \times \sigma_{v_i} \;\le\; v_i \;\le\; +z_{(1-\alpha)/2} \times \sigma_{v_i}\right] \approx \alpha \tag{9.15}$$

Here, the confidence interval for each correction v_i is centred upon an *expected* zero mean with a standard deviation σ_{v_i}. The quantity σ_{v_i} is the standard deviation of the correction calculated from equation 9.3. The upper and lower limits are calculated from the correction standard deviation and the (Normal) z coefficient which is evaluated from the inverse of the Normal *cumulative distribution function* (CDF) at significance α. For example, when $\alpha = 95\%$, $z_{(1-0.95)/2} = 1.96$. The test is two–tailed since the corrections may be either positive or negative. Since the PDF will vary according to σ_{v_i}, the upper and lower limits must be calculated for each correction.

To reduce the time taken to perform this test for every measurement in the network, DynAdjust simplifies the testing of corrections by computing a *standardised Normal statistic* (N–statistic, or normalised residual) for each correction:

$$v'_i = \frac{v_i}{\sigma_{v_i}} \tag{9.16}$$

This permits **adjust** to test all measurement corrections using the *standard* (or *unit*) Normal distribution. The test interval for v'_i is expressed as:

$$P\left[-k \leq v'_i \leq +k\right] \approx \alpha_k \tag{9.17}$$

The upper and lower critical values will be the coverage factor k corresponding to the level of significance (c.f. Table 7.1). For example, the critical value of the unit Normal distribution for the default 95% confidence interval is ±1.96:

$$P\left[-1.96 \leq v'_i \leq +1.96\right] \approx 0.95 \tag{9.18}$$

To examine the local testing undertaken by **adjust**, add the option `---output-adj-msr` to the list of options on the command line:

```
adjust uni_sqr --output-adj-msr --sort-adj-msr-field 7
```

Adding this option will cause **adjust** to print to the adjustment output file a table showing the adjusted measurements (c.f. §8.3.4 on page 127) and their associated statistics. The additional option and argument `--sort-adj-msr-field 7` sorts the list of adjusted measurements according to the N–statistic. Appendix C.7.3 provides further details on the content, structure and format of the adjusted measurements table.

The example on page 142 shows the adjustment statistics summary and some of the adjusted measurements arising from a simultaneous adjustment of the `uni_sqr` network. Note in this case, the rigorous sigma zero is less than the lower limit of the χ^2 confidence interval. At first glance, this might suggest that the solution is performing better than expected (leading to a warning). However, there are 72 measurements which have a correction that exceeds the critical value of the unit Normal distribution and therefore, are noted as potential outliers. For a concise summary of each of the columns in the Adjusted measurements table, please refer to Appendix C.7.3.

As shown by the example on page 142, measurements which have been identified as potential outliers are flagged using an asterisk in the column labelled `Outlier?`. This occurs when the N–statistic v'_i exceeds ±1.96, which is the critical value for the 95% confidence interval. The N–statistic for each measurement is in the column labelled `N-stat` and is calculated from `Correction/Residual` (c.f. equation 9.16). For example, the N–statistic for the first measurement in the list — a zenith distance (c.f. §7.2.12) — is calculated as:

$$\begin{aligned} v' &= v/\sigma_v \\ 7.09 &= 124.1655''/17.5063'' \end{aligned}$$

Knowing that the critical value for the 95% confidence interval is ± 1.96, the values in the `Outlier?` and `N-stat` columns can be quickly scanned to determine which measurements have failed the local test and how significant the failure is. At this point, the user should follow well–defined standards and practices for identifying the nature and cause of the failure using the values contained in the columns labelled `Correction`, `Meas. SD`, `Adj. SD`, `Corr. SD` and `Pelzer Rel`.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

--------------------------------------------------------------------------------
SOLUTION                           Converged
Total time                         00:00:00.052000

Number of unknown parameters       440
Number of measurements             1182  (72 potential outliers)
Degrees of freedom                 742
Chi squared                        627.00
Rigorous Sigma Zero                0.845
Global (Pelzer) Reliability        4.008   (excludes non redundant measurements)

Chi-Square test (95.0%)            0.901 < 0.845 < 1.104         *** WARNING ***


Adjusted Measurements
----------------------------------------

M Station 1           Station 2           Station 3           *  C            Measured            Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD      N-stat  Pelzer Rel  Pre Adj Corr    Outlier?
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
V PELH                NEWP                                                 90 01 47.5000     90 03 51.7413    124.1655      20.0000       9.6711      17.5063        7.09        1.14       -0.0758           *
V 1044                2024                                                 91 22 57.5000     91 21 28.9924    -80.8657      20.0000      14.5085      13.7660       -5.87        1.45        7.6419           *
S 1016                1014                                                       58.4800           58.4955      0.0155       0.0050       0.0041       0.0029        5.39        1.74        0.0000           *
V 1033                1034                                                 89 37 39.0000     89 38 33.0618     56.9413      20.0000      16.6592      11.0666        5.15        1.81        2.8795           *
V 2029                2028                                                 91 56 47.0000     91 56 16.6740    -27.3081      20.0000      19.0357       6.1352       -4.45        3.26        3.0180           *
V 2029                2030                                                 88 28 38.5000     88 29 12.7387     31.1717      20.0000      18.7277       7.0194        4.44        2.85       -3.0669           *
...
S 1015                1014                                                       73.9910           73.9791     -0.0119       0.0050       0.0041       0.0029       -4.16        1.75        0.0000           *
V 2024                2016                                                 90 34 22.0000     90 33 43.7114    -43.7058      20.0000      16.9095      10.6804       -4.09        1.87       -5.4173           *
S 1002                1034                                                       46.5210           46.5155     -0.0055       0.0050       0.0048       0.0014       -3.92        3.54        0.0000           *
V NEWP                PELH                                                 90 14 44.2000     90 15 51.1019     66.9760      20.0000       9.6706      17.5065        3.83        1.14        0.0741           *
G HOSP                5000                                   X                 -528.4051         -528.4158     -0.0107       0.0058       0.0021       0.0054       -2.00        1.07        0.0000           *
G HOSP                5000                                   Y                 -571.8662         -571.8585      0.0077       0.0051       0.0020       0.0047        1.65        1.09        0.0000
G HOSP                5000                                   Z                  168.7984          168.7830     -0.0154       0.0060       0.0021       0.0057       -2.73        1.07        0.0000           *
V BOWL                2106                                                 89 39 50.8000     89 40 50.9127     53.9767      20.0000      13.7470      14.5265        3.72        1.38       -6.1361           *
V BOWL                2106                                                 90 31 18.3000     90 30 31.3701    -53.0660      20.0000      13.7419      14.5313       -3.65        1.38       -6.1361           *
G HOSP                2215                                   X                 -187.7390         -187.7434     -0.0044       0.0032       0.0020       0.0025       -1.79        1.29        0.0000
G HOSP                2215                                   Y                 -416.2030         -416.1971      0.0059       0.0032       0.0021       0.0024        2.47        1.32        0.0000           *
G HOSP                2215                                   Z                  -66.1290          -66.1338     -0.0048       0.0032       0.0021       0.0024       -2.00        1.32        0.0000           *
S 1033                1034                                                       60.5550           60.5473     -0.0077       0.0050       0.0045       0.0021       -3.60        2.33        0.0000           *
V 1044                2011                                                 90 33 54.5000     90 34 55.6139     59.5189      20.0000      11.0894      16.6441        3.58        1.20       -1.5949           *
V 1002                1034                                                 93 02 12.3000     93 01 34.0019    -34.4605      20.0000      17.4263       9.8144       -3.51        2.04        3.8376           *
S 1050                1011                                                       84.5670           84.5609     -0.0061       0.0050       0.0047       0.0017       -3.49        2.87        0.0000           *
S 1011                1012                                                       61.5440           61.5376     -0.0064       0.0050       0.0046       0.0019       -3.35        2.61        0.0000           *
A BARR                NEWP                2118                            330 59 16.7000    330 58 24.6325    -52.0901      20.0000      12.5594      15.5647       -3.35        1.28       -0.0227           *
V 2028                1030                                                 91 41 25.5000     91 40 31.3886    -51.0524      20.0000      12.9284      15.2597       -3.35        1.31        3.0590           *
G 6004                6003                                   X                  138.3607          138.4002      0.0395       0.0241       0.0066       0.0232        1.70        1.04        0.0000
G 6004                6003                                   Y                  226.0200          225.9922     -0.0278       0.0142       0.0051       0.0132       -2.10        1.07        0.0000           *
G 6004                6003                                   Z                   32.1482           32.1860      0.0378       0.0216       0.0060       0.0207        1.82        1.04        0.0000
...
S 4024                4023                                                      167.2120          167.2120      0.0000       0.0100       0.0054       0.0084        0.00        1.19        0.0000
S 4019                1040                                                      240.2420          240.2420      0.0000       0.0100       0.0055       0.0084        0.00        1.20        0.0000
S 2024                2023                                                       49.0460           49.0460      0.0000       0.0050       0.0034       0.0037        0.00        1.36        0.0000
A 4015                4013                1019                             74 15 30.0000     74 15 30.2309      0.0225      20.0000      13.9325      14.3487        0.00        1.39       -0.2083
A 2214                5000                2239                            130 53 36.0000    130 53 35.9631     -0.0161      20.0000      13.7392      14.5339        0.00        1.38        0.0208
V 1015                1018                                                 89 51 16.0000     89 51 23.0383      0.0000      20.0000      20.0000       0.0044        0.00      999.99       -7.0383
S 2007                2021                                                       33.6070           33.6070      0.0000       0.0050       0.0034       0.0037        0.00        1.35        0.0000
A 1015                1007                1018                            258 50 05.0000    258 50 04.7316      0.0000      20.0000      20.0000       0.0044        0.00      999.99        0.2684
A 2108                2120                2102                            254 51 23.0000    254 51 23.3363     -0.0031      20.0000      10.3436      17.1175        0.00        1.17       -0.3395
V 1029                1030                                                 91 04 55.0000     91 04 47.5012      0.0000      20.0000      20.0000       0.0106        0.00      999.99        7.4988
S 1015                1018                                                       22.5200           22.5200      0.0000       0.0050       0.0050       0.0000        0.00      999.99        0.0000
A 1030                1029                1010                             91 29 15.0000     91 29 15.2429      0.0000      20.0000      20.0000       0.0106        0.00      999.99       -0.2429
S 1029                1025                                                      130.6050          130.6050      0.0000       0.0100       0.0100       0.0000        0.00      999.99        0.0000
```

9.3.2.2 Testing measurements with doubtful estimates of precision

When testing newly observed or processed measurements for the first time, or when the processed variance matrix $\mathbf{V_m}$ is in doubt or in any way suspect, the magnitude of the corrections $v_1, v_2, \ldots, v_i$ can be reliably validated using a Student's t test. The test to verify whether a correction v_i falls within the upper and lower limits of the confidence interval at significance α can be written as:

$$P\left[-t_{(\alpha,r-1)} \times \sigma_{v_i} \leq v_i \leq +t_{(\alpha,r-1)} \times \sigma_{v_i}\right] \approx \alpha \tag{9.19}$$

The upper and lower limits are calculated from the correction standard deviation and the Student's t coefficient, evaluated from the inverse of the Student's t CDF at significance α and degrees of freedom r. As was introduced in §7.3.1.3, as the redundancy in a network increases, the critical values approach those of the Normal distribution. Table 9.1 lists the critical values for $r = 2 \ldots 1000$ computed at the 95% confidence interval.

Table 9.1: Student's t confidence intervals for $\alpha = 95\%$, $r = 2 \ldots 1000$

r	$t_{\alpha=95\%,r}$
2	4.303
5	2.571
10	2.228
20	2.086
50	2.009
100	1.984
200	1.972
500	1.965
1000	1.962

Since the Student's t distribution approaches a Normal distribution once the r approaches 100 or so, local testing using Student's t is ideally suited to smaller sets of measurements.

As with testing corrections against the Normal distribution, DynAdjust simplifies tests against the Student's t distribution by computing a standardised Student's t statistic (T–statistic):

$$\begin{aligned} v_i'' &= \frac{v_i'}{\hat{\sigma}_0} \\ &= \frac{v_i}{\hat{\sigma}_0\,\sigma_{v_i}} \end{aligned} \tag{9.20}$$

By definition, the quantity v_i'' is a Student's t random variable and can be tested against the standard (or unit) Student's t distribution. This quantity is similar to the N–statistic v_i' (c.f. equation 9.16), the only difference is that the quotient σ_{v_i} is scaled by the square root of the variance factor $\hat{\sigma}_0^2$ estimated from equation 9.7 — a computation undertaken to produce an *unbiased* estimate of the uncertainty (c.f. §9.3.1.1).

To illustrate how the T–statistic may assist in validating the quality of measurements and measurement precisions, consider the adjustment summary in Figure 9.1 on the next page and the adjusted measurements table in Figure 9.2 on page 145. These results have been produced from a minimally constrained simultaneous adjustment of a network of 10 stations and 19 GNSS baselines. The a–priori variance matrix (obtained directly from the GNSS processing software) was not scaled, and the option `--output-tstat-adj-msr` was provided to **adjust** so that the T–statistic would be printed to the adjusted measurements table.

```
--------------------------------------------------------------------------------
SOLUTION                          Converged
Total time                        00:00:00

Number of unknown parameters      30
Number of measurements            57  (34 potential outliers)
Degrees of freedom                27
Chi squared                       391.33
Rigorous Sigma Zero               14.494
Global (Pelzer) Reliability       1.027   (excludes non redundant measurements)

Chi-Square test (95.0%)           0.540 < 14.494 < 1.600         *** FAILED ***
```

Figure 9.1: Adjustment summary from a sample GNSS network

From these results, the following points are noted:

- The adjustment fails the global test significantly.
- Over one half of the measurements in the set fail the local test using the Normal distribution. These measurements are marked as potential outliers since the N–statistic (column `N-stat`) has exceeded the critical value of the unit Normal distribution for the default 95% confidence interval, being ± 1.96.
- Around two thirds of the GNSS baselines have corrections larger than their a–priori standard deviation. Of those baselines, many have X, Y and/or Z corrections which are *significantly* larger than the corresponding standard deviations.
- All measurements pass the local test using the Student's t distribution

Clearly something is amiss, but as to whether there are blunders, over–optimistic precisions, or both, remains to be answered. The T–statistic for each measurement is in the column labelled `T-stat` and is calculated from `N-stat`$/\hat{\sigma}_0$ (c.f. equation 9.20). For example, the T–statistic for the first measurement in the list — the X component of a GNSS baseline — is calculated as:

$$\begin{aligned} v'' &= v'/\hat{\sigma}_0 \\ -0.97 &= -3.69/\sqrt{14.494} \end{aligned}$$

Here it can be seen that the Student's t distribution is more forgiving in that, whereas the N–statistic value of -3.69 exceeded the critical value of -1.96, the T–statistic value of -0.97 did not. This is the case for all measurements, which suggests that the issue is not necessarily confined to one or a small number of measurements. Rather, the results seem to indicate that there is an underlying issue affecting all measurements.

Considering the form of equation 9.20, it can be seen from the measurement statistics in Figure 9.2 that the T–statistic is equivalent to what would result for the N–statistic if the variance factor was used to scale the a–priori variance matrix prior to adjustment. That is, by scaling the a–priori variance matrix by $\hat{\sigma}_0^2 = 14.494$ prior to adjustment, the new a–posteriori $\hat{\sigma}_0^2$ would become 1.0 and the N–statistic would become equal to the T–statistic. By following the procedure outlined in §9.3.1.1, the adjusted measurements table in Figure 9.3 on page 146 shows the net effect of scaling the a–priori variance matrix by $\hat{\sigma}_0^2 = 14.494$. Since all measurements now pass the local test using the Normal distribution, it may be inferred — in this case — that the issue lay solely with the reliability of the variance matrix produced by the GNSS processing package. In particular, the variance matrix was over–optimistic.

1234567890123456789

M	Station 1	Station 2	Station 3	*	C	Measured	Adjusted	Correction	Meas. SD	Adj. SD	Corr. SD	N-stat	T-stat	Pelzer Rel	Pre Adj Corr	Outlier?
G	409700110	299000080			X	44663.6990	44663.6813	-0.0177	0.0062	0.0039	0.0048	-3.69	-0.97	1.29	0.0000	*
G	409700110	299000080			Y	34616.4430	34616.4636	0.0206	0.0048	0.0030	0.0037	5.57	1.46	1.29	0.0000	*
G	409700110	299000080			Z	-21377.6470	-21377.6702	-0.0232	0.0055	0.0035	0.0043	-5.41	-1.42	1.29	0.0000	*
G	310211240	360100260			X	-11412.3870	-11412.3644	0.0226	0.0067	0.0039	0.0054	4.19	1.10	1.24	0.0000	*
G	310211240	360100260			Y	-59911.3030	-59911.3220	-0.0190	0.0050	0.0030	0.0040	-4.69	-1.23	1.25	0.0000	*
G	310211240	360100260			Z	-40068.0740	-40068.0528	0.0212	0.0060	0.0035	0.0049	4.37	1.15	1.23	0.0000	*
G	310211240	236300210			X	14084.5430	14084.5358	-0.0072	0.0033	0.0028	0.0018	-4.02	-1.06	1.85	0.0000	*
G	310211240	236300210			Y	-10691.6250	-10691.6185	0.0065	0.0025	0.0021	0.0013	4.78	1.26	1.86	0.0000	*
G	310211240	236300210			Z	-25853.7910	-25853.7963	-0.0053	0.0029	0.0025	0.0016	-3.31	-0.87	1.84	0.0000	*
G	212000820	269100210			X	22694.2880	22694.2983	0.0103	0.0062	0.0049	0.0039	2.65	0.70	1.60	0.0000	*
G	212000820	269100210			Y	19348.4820	19348.4690	-0.0130	0.0046	0.0036	0.0029	-4.53	-1.19	1.60	0.0000	*
G	212000820	269100210			Z	-9559.1510	-9559.1364	0.0146	0.0058	0.0045	0.0037	3.98	1.05	1.57	0.0000	*
G	212000820	409700110			X	-10325.4980	-10325.5079	-0.0099	0.0066	0.0050	0.0044	-2.27	-0.60	1.51	0.0000	*
G	212000820	409700110			Y	39327.8660	39327.8792	0.0132	0.0049	0.0037	0.0032	4.14	1.09	1.52	0.0000	*
G	212000820	409700110			Z	44763.3780	44763.3636	-0.0144	0.0059	0.0045	0.0039	-3.71	-0.98	1.53	0.0000	*
G	269100210	299000080			X	11643.8660	11643.8751	0.0091	0.0056	0.0035	0.0044	2.06	0.54	1.28	0.0000	*
G	269100210	299000080			Y	54595.8870	54595.8739	-0.0131	0.0044	0.0027	0.0035	-3.79	-1.00	1.27	0.0000	*
G	269100210	299000080			Z	32944.8150	32944.8299	0.0149	0.0050	0.0031	0.0039	3.81	1.00	1.28	0.0000	*
G	269100210	230900140			X	26690.7580	26690.7616	0.0036	0.0036	0.0031	0.0017	2.09	0.55	2.05	0.0000	*
G	269100210	230900140			Y	47953.9560	47953.9509	-0.0051	0.0028	0.0024	0.0014	-3.76	-0.99	2.03	0.0000	*
G	269100210	230900140			Z	10466.4910	10466.4965	0.0055	0.0032	0.0028	0.0016	3.46	0.91	2.02	0.0000	*
G	335800500	360100260			X	-46576.9390	-46576.9589	-0.0199	0.0062	0.0032	0.0053	-3.75	-0.99	1.16	0.0000	*
G	335800500	360100260			Y	-18603.6710	-18603.6595	0.0115	0.0047	0.0024	0.0040	2.86	0.75	1.17	0.0000	*
G	335800500	360100260			Z	36786.5760	36786.5669	-0.0091	0.0055	0.0028	0.0047	-1.93	-0.51	1.16	0.0000	
G	230900140	299000080			X	-15046.8920	-15046.8865	0.0055	0.0046	0.0036	0.0030	1.85	0.49	1.57	0.0000	
G	230900140	299000080			Y	6641.9310	6641.9230	-0.0080	0.0036	0.0028	0.0023	-3.45	-0.91	1.57	0.0000	*
G	230900140	299000080			Z	22478.3250	22478.3333	0.0083	0.0041	0.0032	0.0026	3.19	0.84	1.58	0.0000	*
G	335800500	236300210			X	-21080.0670	-21080.0586	0.0084	0.0048	0.0033	0.0034	2.45	0.64	1.40	0.0000	*
G	335800500	236300210			Y	30616.0510	30616.0439	-0.0071	0.0037	0.0026	0.0026	-2.71	-0.71	1.40	0.0000	*
G	335800500	236300210			Z	51000.8130	51000.8234	0.0104	0.0042	0.0030	0.0030	3.43	0.90	1.40	0.0000	*
G	269100210	409700110			X	-33019.8010	-33019.8062	-0.0052	0.0056	0.0038	0.0041	-1.28	-0.34	1.37	0.0000	
G	269100210	409700110			Y	19979.4010	19979.4103	0.0093	0.0041	0.0029	0.0030	3.12	0.82	1.39	0.0000	*
G	269100210	409700110			Z	54322.5110	54322.5000	-0.0110	0.0050	0.0034	0.0037	-2.97	-0.78	1.36	0.0000	*
G	409700110	360100260			X	20600.9200	20600.9186	-0.0014	0.0075	0.0047	0.0058	-0.25	-0.07	1.29	0.0000	
G	409700110	360100260			Y	65606.4490	65606.4602	0.0112	0.0058	0.0037	0.0045	2.49	0.65	1.29	0.0000	*
G	409700110	360100260			Z	32812.8170	32812.8063	-0.0107	0.0065	0.0042	0.0050	-2.12	-0.56	1.30	0.0000	*
G	236300210	360100260			X	-25496.8870	-25496.9003	-0.0133	0.0064	0.0036	0.0052	-2.54	-0.67	1.22	0.0000	*
G	236300210	360100260			Y	-49219.7100	-49219.7034	0.0066	0.0048	0.0028	0.0039	1.67	0.44	1.22	0.0000	
G	236300210	360100260			Z	-14214.2510	-14214.2565	-0.0055	0.0056	0.0032	0.0046	-1.18	-0.31	1.22	0.0000	
G	299000080	360100260			X	-24062.7700	-24062.7628	0.0072	0.0057	0.0038	0.0042	1.73	0.45	1.36	0.0000	
G	299000080	360100260			Y	30990.0030	30989.9965	-0.0065	0.0044	0.0030	0.0032	-2.00	-0.53	1.36	0.0000	*
G	299000080	360100260			Z	54190.4720	54190.4765	0.0045	0.0050	0.0034	0.0037	1.23	0.32	1.36	0.0000	
G	310211240	365300060			X	25366.5730	25366.5783	0.0053	0.0051	0.0035	0.0037	1.43	0.38	1.38	0.0000	
G	310211240	365300060			Y	29789.3220	29789.3168	-0.0052	0.0039	0.0027	0.0029	-1.83	-0.48	1.38	0.0000	
G	310211240	365300060			Z	-2397.0210	-2397.0195	0.0015	0.0045	0.0031	0.0033	0.48	0.12	1.38	0.0000	
G	236300210	365300060			X	11282.0440	11282.0424	-0.0016	0.0047	0.0033	0.0033	-0.48	-0.13	1.41	0.0000	
G	236300210	365300060			Y	40480.9310	40480.9353	0.0043	0.0036	0.0025	0.0026	1.67	0.44	1.41	0.0000	
G	236300210	365300060			Z	23456.7760	23456.7768	0.0008	0.0041	0.0029	0.0029	0.29	0.07	1.41	0.0000	
G	299000080	335800500			X	22514.2000	22514.1961	-0.0039	0.0055	0.0040	0.0037	-1.04	-0.27	1.47	0.0000	
G	299000080	335800500			Y	49593.6580	49593.6561	-0.0019	0.0043	0.0031	0.0030	-0.65	-0.17	1.45	0.0000	
G	299000080	335800500			Z	17403.9060	17403.9096	0.0036	0.0049	0.0036	0.0033	1.08	0.28	1.46	0.0000	
G	335800500	360100260			X	-46576.9660	-46576.9589	0.0071	0.0061	0.0032	0.0052	1.36	0.36	1.17	0.0000	
G	335800500	360100260			Y	-18603.6570	-18603.6595	-0.0025	0.0047	0.0024	0.0040	-0.62	-0.16	1.17	0.0000	
G	335800500	360100260			Z	36786.5680	36786.5669	-0.0011	0.0053	0.0028	0.0045	-0.24	-0.06	1.18	0.0000	
G	335800500	365300060			X	-9798.0120	-9798.0162	-0.0042	0.0060	0.0039	0.0045	-0.93	-0.24	1.32	0.0000	
G	335800500	365300060			Y	71096.9790	71096.9792	0.0002	0.0047	0.0030	0.0036	0.07	0.02	1.31	0.0000	
G	335800500	365300060			Z	74457.6030	74457.6002	-0.0028	0.0053	0.0034	0.0040	-0.70	-0.18	1.32	0.0000	

Figure 9.2: Adjusted measurements table from a sample GNSS network — without a–priori variance matrix scaling

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

M Station 1            Station 2             Station 3            *  C          Measured           Adjusted  Correction    Meas. SD      Adj. SD     Corr. SD     N-stat     T-stat  Pelzer Rel  Pre Adj Corr     Outlier?
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
G 409700110            299000080                                     X        44663.6990         44663.6813     -0.0177      0.0235       0.0149       0.0182      -0.97      -0.97        1.29        0.0000
G 409700110            299000080                                     Y        34616.4430         34616.4636      0.0206      0.0181       0.0114       0.0141       1.46       1.46        1.29        0.0000
G 409700110            299000080                                     Z       -21377.6470        -21377.6702     -0.0232      0.0210       0.0133       0.0163      -1.42      -1.42        1.29        0.0000
G 310211240            360100260                                     X       -11412.3870        -11412.3644      0.0226      0.0254       0.0150       0.0205       1.10       1.10        1.24        0.0000
G 310211240            360100260                                     Y       -59911.3030        -59911.3220     -0.0190      0.0192       0.0115       0.0154      -1.23      -1.23        1.25        0.0000
G 310211240            360100260                                     Z       -40068.0740        -40068.0528      0.0212      0.0228       0.0134       0.0185       1.15       1.15        1.23        0.0000
G 310211240            236300210                                     X        14084.5430         14084.5358     -0.0072      0.0125       0.0105       0.0068      -1.06      -1.06        1.85        0.0000
G 310211240            236300210                                     Y       -10691.6250        -10691.6185      0.0065      0.0096       0.0081       0.0051       1.26       1.26        1.86        0.0000
G 310211240            236300210                                     Z       -25853.7910        -25853.7963     -0.0053      0.0112       0.0094       0.0061      -0.87      -0.87        1.84        0.0000
G 212000820            269100210                                     X        22694.2880         22694.2983      0.0103      0.0237       0.0185       0.0148       0.70       0.70        1.60        0.0000
G 212000820            269100210                                     Y        19348.4820         19348.4690     -0.0130      0.0176       0.0137       0.0110      -1.19      -1.19        1.60        0.0000
G 212000820            269100210                                     Z        -9559.1510         -9559.1364      0.0146      0.0220       0.0170       0.0140       1.05       1.05        1.57        0.0000
G 212000820            409700110                                     X       -10325.4980        -10325.5079     -0.0099      0.0252       0.0189       0.0166      -0.60      -0.60        1.51        0.0000
G 212000820            409700110                                     Y        39327.8660         39327.8792      0.0132      0.0185       0.0140       0.0122       1.09       1.09        1.52        0.0000
G 212000820            409700110                                     Z        44763.3780         44763.3636     -0.0144      0.0226       0.0171       0.0147      -0.98      -0.98        1.53        0.0000
G 269100210            299000080                                     X        11643.8660         11643.8751      0.0091      0.0214       0.0133       0.0168       0.54       0.54        1.28        0.0000
G 269100210            299000080                                     Y        54595.8870         54595.8739     -0.0131      0.0167       0.0103       0.0132      -1.00      -1.00        1.27        0.0000
G 269100210            299000080                                     Z        32944.8150         32944.8299      0.0149      0.0190       0.0119       0.0148       1.00       1.00        1.28        0.0000
G 269100210            230900140                                     X        26690.7580         26690.7616      0.0036      0.0135       0.0118       0.0066       0.55       0.55        2.05        0.0000
G 269100210            230900140                                     Y        47953.9560         47953.9509     -0.0051      0.0106       0.0092       0.0052      -0.99      -0.99        2.03        0.0000
G 269100210            230900140                                     Z        10466.4910         10466.4965      0.0055      0.0123       0.0107       0.0061       0.91       0.91        2.02        0.0000
G 335800500            360100260                                     X       -46576.9390        -46576.9589     -0.0199      0.0235       0.0121       0.0202      -0.99      -0.99        1.16        0.0000
G 335800500            360100260                                     Y       -18603.6710        -18603.6595      0.0115      0.0179       0.0093       0.0153       0.75       0.75        1.17        0.0000
G 335800500            360100260                                     Z        36786.5760         36786.5669     -0.0091      0.0209       0.0107       0.0180      -0.51      -0.51        1.16        0.0000
G 230900140            299000080                                     X       -15046.8920        -15046.8865      0.0055      0.0177       0.0136       0.0113       0.49       0.49        1.57        0.0000
G 230900140            299000080                                     Y         6641.9310          6641.9230     -0.0080      0.0137       0.0106       0.0088      -0.91      -0.91        1.57        0.0000
G 230900140            299000080                                     Z        22478.3250         22478.3333      0.0083      0.0157       0.0121       0.0099       0.84       0.84        1.58        0.0000
G 335800500            236300210                                     X       -21080.0670        -21080.0586      0.0084      0.0182       0.0127       0.0130       0.64       0.64        1.40        0.0000
G 335800500            236300210                                     Y        30616.0510         30616.0439     -0.0071      0.0139       0.0098       0.0099      -0.71      -0.71        1.40        0.0000
G 335800500            236300210                                     Z        51000.8130         51000.8234      0.0104      0.0161       0.0113       0.0115       0.90       0.90        1.40        0.0000
G 269100210            409700110                                     X       -33019.8010        -33019.8062     -0.0052      0.0212       0.0145       0.0155      -0.34      -0.34        1.37        0.0000
G 269100210            409700110                                     Y        19979.4010         19979.4103      0.0093      0.0157       0.0109       0.0113       0.82       0.82        1.39        0.0000
G 269100210            409700110                                     Z        54322.5110         54322.5000     -0.0110      0.0192       0.0131       0.0141      -0.78      -0.78        1.36        0.0000
G 409700110            360100260                                     X        20600.9200         20600.9186     -0.0014      0.0286       0.0181       0.0221      -0.07      -0.07        1.29        0.0000
G 409700110            360100260                                     Y        65606.4490         65606.4602      0.0112      0.0221       0.0139       0.0171       0.65       0.65        1.29        0.0000
G 409700110            360100260                                     Z        32812.8170         32812.8063     -0.0107      0.0249       0.0159       0.0192      -0.56      -0.56        1.30        0.0000
G 236300210            360100260                                     X       -25496.8870        -25496.9003     -0.0133      0.0242       0.0138       0.0199      -0.67      -0.67        1.22        0.0000
G 236300210            360100260                                     Y       -49219.7100        -49219.7034      0.0066      0.0182       0.0105       0.0149       0.44       0.44        1.22        0.0000
G 236300210            360100260                                     Z       -14214.2510        -14214.2565     -0.0055      0.0215       0.0122       0.0176      -0.31      -0.31        1.22        0.0000
G 299000080            360100260                                     X       -24062.7700        -24062.7628      0.0072      0.0216       0.0146       0.0159       0.45       0.45        1.36        0.0000
G 299000080            360100260                                     Y        30990.0030         30989.9965     -0.0065      0.0167       0.0113       0.0123      -0.53      -0.53        1.36        0.0000
G 299000080            360100260                                     Z        54190.4720         54190.4765      0.0045      0.0189       0.0129       0.0139       0.32       0.32        1.36        0.0000
G 310211240            365300060                                     X        25366.5730         25366.5783      0.0053      0.0193       0.0133       0.0140       0.38       0.38        1.38        0.0000
G 310211240            365300060                                     Y        29789.3220         29789.3168     -0.0052      0.0150       0.0103       0.0109      -0.48      -0.48        1.38        0.0000
G 310211240            365300060                                     Z        -2397.0210         -2397.0195      0.0015      0.0171       0.0118       0.0124       0.12       0.12        1.38        0.0000
G 236300210            365300060                                     X        11282.0440         11282.0424     -0.0016      0.0177       0.0125       0.0126      -0.13      -0.13        1.41        0.0000
G 236300210            365300060                                     Y        40480.9310         40480.9353      0.0043      0.0138       0.0097       0.0098       0.44       0.44        1.41        0.0000
G 236300210            365300060                                     Z        23456.7760         23456.7768      0.0008      0.0157       0.0111       0.0111       0.07       0.07        1.41        0.0000
G 299000080            335800500                                     X        22514.2000         22514.1961     -0.0039      0.0208       0.0153       0.0141      -0.27      -0.27        1.47        0.0000
G 299000080            335800500                                     Y        49593.6580         49593.6561     -0.0019      0.0165       0.0119       0.0114      -0.17      -0.17        1.45        0.0000
G 299000080            335800500                                     Z        17403.9060         17403.9096      0.0036      0.0185       0.0135       0.0127       0.28       0.28        1.46        0.0000
G 335800500            360100260                                     X       -46576.9660        -46576.9589      0.0071      0.0232       0.0121       0.0198       0.36       0.36        1.17        0.0000
G 335800500            360100260                                     Y       -18603.6570        -18603.6595     -0.0025      0.0180       0.0093       0.0154      -0.16      -0.16        1.17        0.0000
G 335800500            360100260                                     Z        36786.5680         36786.5669     -0.0011      0.0202       0.0107       0.0171      -0.06      -0.06        1.18        0.0000
G 335800500            365300060                                     X        -9798.0120         -9798.0162     -0.0042      0.0227       0.0148       0.0173      -0.24      -0.24        1.32        0.0000
G 335800500            365300060                                     Y        71096.9790         71096.9792      0.0002      0.0179       0.0115       0.0137       0.02       0.02        1.31        0.0000
G 335800500            365300060                                     Z        74457.6030         74457.6002     -0.0028      0.0201       0.0131       0.0153      -0.18      -0.18        1.32        0.0000
```

Figure 9.3: Adjusted measurements table from a sample GNSS network — with a–priori variance matrix scaling

In the context of identifying outliers, consider the adjustment summary and adjusted measurements table shown in Figure 9.4 (a) on the following page. These results are from a simultaneous adjustment of a small spirit levelling network of 5 stations and 8 orthometric height difference measurements. Two Permanent Marks with published reduced level (RL) values were held constrained (`C`). The a–priori measurement standard deviations have been evaluated by scaling the manufacturer–stated instrument precisions by a factor of $12\sqrt{k}$, where k is the distance in kilometres of each level run. The height differences were reduced from two–way levelling and as such, the measurement standard deviations are considered reliable. The geoid–ellipsoid separations were interpolated for all points using **geoid**, the effect of which on each height difference is tabulated in the column `Pre Adj Corr`. The option `--output-tstat-adj-msr` was provided to **adjust** so that the T–statistic would be printed to the adjusted measurements table.

From the results in Figure 9.4 (a), the following points are noted:

- The adjustment fails the global test significantly.
- Three of the eight measurements fail the local test using the Normal distribution. These measurements are marked as potential outliers since the N–statistic has exceeded ± 1.96, being the critical value of the unit Normal distribution for the default 95% confidence interval. All three measurements have corrections larger than the corresponding standard deviation.
- Of the measurements which are marked as potential outliers, one measurement (`2` to `BM-3`) has a correction which is *significantly* larger than the corresponding standard deviation. This measurement is the only measurement which fails the local test using the Student's t distribution.

From this elementary analysis, it would appear that the measurement `2` to `BM-3` is suspect. To prove whether this is indeed the case, a repeat adjustment will be run with this measurement removed from the set. As a general rule, only one change to a measurement set should be undertaken at a time so as to not adversely affect other measurements which might be of reasonable quality. The results from this adjustment are shown in Figure 9.4 (b) on the next page.

From the repeat adjustment, the following points are noted:

- The global test has produced a warning, indicating that the measurement set performed better than assumed by the standard deviations.
- No measurements fail the local test using the Normal distribution or the Student's t distribution.

From this analysis, it can be safely assumed that measurement `2` to `BM-3` is indeed a blunder. Whilst the estimated heights would vary marginally from scaling the a–priori standard deviations by the variance factor (c.f. §9.3.1.1), doing so is recommended if (a) more reliable estimates of uncertainty are required on those heights or (b) if these levelling measurements are to be integrated with other measurement sets in a larger network adjustment.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

--------------------------------------------------------------------------------
SOLUTION                           Converged
Total time                         00:00:00

Number of unknown parameters       3
Number of measurements             8  (3 potential outliers)
Degrees of freedom                 5
Chi squared                        38.23
Rigorous Sigma Zero                7.646
Global (Pelzer) Reliability        0.833   (excludes non redundant measurements)

Chi-Square test (95.0%)            0.166 < 7.646 < 2.567          *** FAILED ***


Adjusted Measurements
----------------------------------------

M Station 1           Station 2           Station 3           *  C            Measured           Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD     N-stat     T-stat  Pelzer Rel  Pre Adj Corr    Outlier?
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
L 2                   BM-3                                                          -5.180             -5.115       0.065        0.012        0.006        0.011       6.18       2.23        1.14        0.0069           *
L BM-1                2                                                             16.164             16.183       0.019        0.010        0.006        0.008       2.28       0.82        1.22        0.0074           *
L 2                   3                                                             -8.119             -8.129      -0.010        0.008        0.006        0.005      -2.08      -0.75        1.58       -0.0061           *
L 1                   2                                                             10.756             10.766       0.010        0.010        0.007        0.007       1.54       0.56        1.51       -0.0059
L 3                   BM-3                                                           3.024              3.015      -0.009        0.009        0.006        0.007      -1.36      -0.49        1.33        0.0130
L 1                   BM-3                                                           5.662              5.651      -0.011        0.012        0.007        0.009      -1.11      -0.40        1.26        0.0011
L BM-1                3                                                              8.046              8.053       0.007        0.012        0.006        0.010       0.69       0.25        1.15        0.0013
L BM-1                1                                                              5.410              5.417       0.007        0.015        0.007        0.013       0.50       0.18        1.15        0.0132
```

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

--------------------------------------------------------------------------------
SOLUTION                           Converged
Total time                         00:00:00

Number of unknown parameters       3
Number of measurements             7
Degrees of freedom                 4
Chi squared                        0.06
Rigorous Sigma Zero                0.015
Global (Pelzer) Reliability        0.934   (excludes non redundant measurements)

Chi-Square test (95.0%)            0.121 < 0.015 < 2.786         *** WARNING ***


Adjusted Measurements
----------------------------------------

M Station 1           Station 2           Station 3           *  C            Measured           Adjusted  Correction     Meas. SD      Adj. SD     Corr. SD     N-stat     T-stat  Pelzer Rel  Pre Adj Corr    Outlier?
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
L BM-1                1                                                              5.410              5.408      -0.002        0.015        0.007        0.013      -0.18      -1.44        1.15        0.0132
L 1                   BM-3                                                           5.662              5.660      -0.002        0.012        0.007        0.009      -0.18      -1.43        1.28        0.0011
L BM-1                3                                                              8.046              8.045      -0.001        0.012        0.006        0.010      -0.12      -0.97        1.16        0.0013
L 3                   BM-3                                                           3.024              3.023      -0.001        0.009        0.006        0.007      -0.11      -0.92        1.36        0.0130
L BM-1                2                                                             16.164             16.164       0.000        0.010        0.006        0.008      -0.03      -0.21        1.31        0.0074
L 1                   2                                                             10.756             10.756       0.000        0.010        0.008        0.006       0.02       0.16        1.55       -0.0059
L 2                   3                                                             -8.119             -8.119       0.000        0.008        0.006        0.005      -0.01      -0.08        1.68       -0.0061
```

Figure 9.4: (a) Level run adjustment with all original measurements, and (b) Repeat level run adjustment with the suspect measurement removed

9.3.3 Hints on addressing measurement failures

There is no single recipe or prescription that may be applied to each adjustment to identify and address inconsistencies in the measurement system. As stated earlier, the user should follow well–defined standards and practices for identifying the nature and cause of such inconsistencies using the values contained in the columns labelled `Correction`, `Meas. SD`, `Adj. SD`, `Corr. SD` and `Pelzer Rel`.

As a general guide however, the following hints may be of some practical value:

1. Attempt to identify measurement blunders and defective variance matrices as soon as possible after measurement processing and/or reduction.
2. The process of detecting outliers and manipulating measurement variance matrices should be undertaken on measurement sets of the same type, measurement characteristics, category and processing strategy.
3. It is easier to identify anomalies in smaller–sized measurement sets, and to rectify inconsistencies prior to aggregating different measurement sets into a larger network.
4. If a GNSS baseline variance matrix produced from GNSS processing software is over–optimistic, it is likely that many other variance matrices from the same campaign will also bear the same optimism. Therefore, it will be advantageous to consider the measurement set as a whole, and to verify whether the problem may be rectified by scaling the a–priori variance matrix.
5. Attempt to distinguish from the list of measurements which have failed the measurements which are blunders and those which may be reliable. Blunder measurements are those which are characterised by a very high N–statistic value or a very large correction, the failure of which is usually *significantly* different to the other measurements. Reliable measurements connected to a station (to which a blunder is also connected) can often appear to be a blunder. Therefore, when ignoring or deleting suspected blunders from the input measurement file, measurement removal and re–adjustment should be undertaken one measurement at a time. This procedure will minimise the effect on correlated measurements which are of satisfactory quality.
6. Repeat the adjustment and note the differences in the global test and local testing of each measurement.
7. Repeat steps 4–6 until there are no unsatisfactory measurement failures.

Bibliography

Alberda, J. E. (1963). Report on the Adjustment of the United European Levelling Net and Related Computations. In *Publications on Geodesy — New Series*, Volume 1, Number 2. Netherlands Geodetic Commission, Delft.

Allman, J. (1983). *Final Technical Report to the National mapping Council. Geodetic Model of Australia 1982*. University of New South Wales.

Allman, J. and Steed, J. (1980). *Geodetic Model of Australia 1980*. Technical Report 29. Division of National Mapping, Canberra.

Allman, J. and Veenstra, C. (1984). *Geodetic Model of Australia 1982*. Technical Report 33. Division of National Mapping, Canberra.

Altamimi, Z., Collilieux, X., and Métivier, L. (2011). ITRF2008: an improved solution of the international terrestrial reference frame. *Journal of Geodesy 85*, 457–473.

Altamimi, Z., Métivier, L., Rebischung, P., Rouby, H., and Collilieux, X. (2017). ITRF2014 plate motion model. *Geophysical Journal International 209*(3), 1906–1912.

Altamimi, Z., Rebischung, P., Métivier, L., and Collilieux, X. (2016). ITRF2014: A new release of the International Terrestrial Reference Frame modeling nonlinear station motions. *Journal of Geophysical Research: Solid Earth 121*(8), 6109–6131.

Apache (2017). *Apache Xerces XML parser (compiler software). Version 3.1.* Apache Software Foundation. Available online: http://xerces.apache.org/.

Argus, D. F., Gordon, R. G., and DeMets, C. (2011). Geologically current motion of 56 plates relative to the no-net-rotation reference frame. *Geochemistry, Geophysics, Geosystems 12*(11), 1–13.

Baarda, W. (1967). *Statistical Concepts in Geodesy*, Volume 2. Netherlands Geodetic Commission.

Bird, P. (2003). An updated digital model of plate boundaries. *Geochemistry, Geophysics, Geosystems 4*(3), 1–52.

Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.

Blewitt, G. (1996). GPS Data Processing Methodology. In *GPS for Geodesy*, Kleusberg, A. and Teunissen, P. (Eds.), pp. 231–270. Springer–Verlag, Berlin.

Bomford, A. (1967). The geodetic adjustment of Australia. *Survey Review 19*(144), 52–71.

Bomford, G. (1980). *Geodesy* (4th ed.). Clarendon Press, London.

Bowditch, N. (1939). *The American Practical Navigator: An Epitome of Navigation — Originally by Nathaniel Bowditch* (1938 ed.). Pub. No. 9, [First published in 1802] United States Government Printing Office, Washington.

Bowring, B. (1985). The accuracy of geodetic latitude and height equations. *Survey Review 28*(218), 202–206.

Brown, N., Harrison, C., Fraser, R., and Dawson, J. (2018). Using an avalanche of measurements to improve national datums. In *Proceedings of the 23rd Association of Public Authority Surveyors Conference (APAS2018)*, Volume 14 of *Jindabyne*.

Buttari, A., Langou, J., Kurzak, J., and Dongarra, J. (2009). A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Computing 35*(1), 38–53.

Choi, J., Demmel, J., Dhillon, I., Dongarra, J., Ostrouchov, S., Petitet, A., Stanley, K., Walker, D., and Whaley, R. C. (1996). ScaLAPACK: A portable linear algebra library for distributed memory computers — design issues and performance. In *Applied Parallel Computing Computations in Physics, Chemistry and Engineering Science*, Dongarra, J., Madsen, K., and Waśniewski, J. (Eds.), pp. 95–106. Springer Berlin Heidelberg.

Collier, P. (1991). *Status Report on the Re–adjustment of the MMBW Horizontal Survey Control Network*. MMBW Survey and Mapping Group — Internal Report.

Collier, P. and Leahy, F. (1992). Readjustment of Melbourne's Survey Control Network. *Australian Surveyor 37*(4), 275–288.

Collier, P., Leahy, F., and Argeseanu, V. (1996). *Transition to the Geocentric Datum of Australia*. Consultants Report to the Office of Surveyor General, Victoria. Department of Geomatics, The University of Melbourne.

Collier, P. A. (1988). *Integrated Geodesy Applied to Precise Engineering Surveys*. Ph. D. thesis, University of Melbourne.

Cooper, M. (1974). *Aspects of Modern Land Surveying — Fundamentals of Survey Measurement and Analysis*. Crosby Lockwood Staples, London.

Cooper, M. (1987). *Control Surveys in Civil Engineering*. Collins Professional and Technical Books, London.

Cooper, M. and Leahy, F. (1977). Phased adjustment of engineering control surveys. In *XV International Congress of Surveyors, Stockholm*.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms* (2 ed.). The MIT Press.

Costa, S. M. A., Pereira, K. D., and Beattie, D. (1998). The integration of brazilian geodetic network into sirgas. In *Advances in Positioning and Reference Frames*, Brunner, F. K. (Ed.), pp. 187–192, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cross, P. A. (1990). *Advanced Least Squares Applied to Position–Fixing, Working paper No. 6* (1st ed.). Polytechnic of East London, Department of Land Surveying. Reprinted with corrections.

Dagum, L. and Menon, R. (1998). OpenMP: An industry–standard API for shared–memory programming. *IEEE Comput. Sci. Eng. 5*(1), 46–55.

Dawson, J., Deo, M., Jia, M., Hu, G., and Ruddick, R. (2012). Asia Pacific Reference Frame (APREF): Densification of the ITRF in the Asia Pacific. In *EGU General Assembly Conference Abstracts*, Abbasi, A. and Giesen, N. (Eds.), Volume 14 of *EGU General Assembly Conference Abstracts*, pp. 3940.

Dawson, J. and Woods, A. (2010). ITRF to GDA94 coordinate transformations. *Journal of Applied Geodesy 4*, 189–199.

DeMets, C., Gordon, R. G., and Argus, D. F. (2010). Geologically current plate motions. *Geophysical Journal International 181*(1), 1–80.

DeMets, C., Gordon, R. G., Argus, D. F., and Stein, S. (1990). Current plate motions. *Geophysical Journal International 101*(2), 425–478.

DeMets, C., Gordon, R. G., Argus, D. F., and Stein, S. (1994). Effect of recent revisions to the geomagnetic reversal time scale on estimates of current plate motions. *Geophysical Research Letters 21*(20), 2191–2194.

Demmel, J., Grigori, L., Hoemmen, M., and Langou, J. (2012). Communication-optimal parallel and sequential QR and LU factorizations. *SIAM Journal on Scientific Computing 34*(1), 206–239.

Dongarra, J. J., Gates, M., Haidar, A., Kurzak, J., Luszczek, P., Wu, P., Yamazaki, I., YarKhan, A., Abalenkovs, M., Bagherpour, N., Hammarling, S., SÃstek, J., Stevens, D., Zounon, M., and Relton, S. D. (2019). PLASMA: Parallel Linear Algebra Software for Multicore using OpenMP. *ACM Transactions on Mathematical Software 45*(2), 1–35.

Ehrnsperger, W. (1991). The ED87 adjustment. *Bulletin géodésique 65*, 28–43.

Elmroth, E. and Gustavson, F. (2000). Applying recursion to serial and parallel QR factorization leads to better performance. *IBM Journal Research and Development 44*(4), 605–624.

Featherstone, W., Kirby, J., Hirt, C., Filmer, M., Claessens, S., Brown, N., Hu, G., and Johnston, G. (2011). The AUSGeoid09 model of the Australian Height Datum. *Journal of Geodesy 85*(3), 133–150.

Featherstone, W. and Kuhn, M. (2006). Height systems and vertical datums: a review in the Australian context. *Journal of Spatial Science 51*(1), 21–42.

Fortes, L. P. S. (1998). An overview of the sirgas project. In *Advances in Positioning and Reference Frames*, Brunner, F. K. (Ed.), pp. 167–167, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fraser, R., Leahy, F., and Collier, P. (2020). Automatic segmentation and parallel phased least squares for highly efficient parameter estimation.

Gagnon, P. (1976). *Step by Step Adjustment Procedures for Large Horizontal Geodetic Networks. Technical Report No. 38*. University of New Brunswick, Frederickton.

Gauss, C. F. (1809). *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*. [English Translation (1857) by C. H. Davis, Little, Brown & Co., Boston.], Hamburg.

Golub, G. H. and Plemmons, R. J. (1980). Large–scale geodetic least–squares adjustment by dissection and orthogonal decomposition. *Linear Algebra Applications 34*, 3–28.

Golub, G. H., Plemmons, R. J., and Sameh, A. (1988). Parallel block schemes for large–scale least–squares computations. In *High–Speed Computing, Scientific Applications and Algorithm Design*, pp. 171–179.

Hayes, B. (2002). Science on the farther shore. *American Scientist 90*(6), 499–502.

Heiskanen, W. A. and Moritz, H. (1993). *Physical Geodesy*. W.H.Freeman an Company San Francisco, Reprint, Institute of Physical Geodesy, Technical University, Graz, Austria.

Helmert, F. R. (1880). *Die Mathematischen und Physikalischen Theorien der höheren Geodäsie, 1 Teil, Leipzig, Germany*. Teubner, Leipzig.

Hofmann–Wellenhof, B., Lichtenegger, H., and Collins, J. (2001). *Global Positioning System — Theory and Practice* (5th ed.). Springer–Verlag/Wien, New York.

Hofmann–Wellenhof, B. and Moritz, H. (2006). *Physical Geodesy* (2nd ed.). Springer–Verlag/Wien, New York.

Husbands, P. and Yelick, K. (2007). Multi-threading and one-sided communication in parallel LU factorization. In *Proceedings of the ACM/IEEE Conference on High Performance Networking and Computing, SC 2007, November 10-16, 2007, Reno, Nevada, USA*, Verastegui, B. (Ed.), pp. 31. ACM Press.

ICSM (2002). *Geocentric Datum of Australia Technical Manual*. Intergovernmental Committee on Surveying and Mapping, Canberra, Australia. Version 2.2.

ICSM (2014). *Guideline for the Adjustment and Evaluation of Survey Control. Special Publication No. 1*. Intergovernmental Committee on Surveying and Mapping (ICSM), Version 2.1.

ICSM (2017). *Geocentric Datum of Australia 2020 Interim Release Note*. Intergovernmental Committee on Surveying and Mapping (ICSM), Version 1.01.

ICSM (2020). *Geocentric Datum of Australia 2020 Technical Manual*. Intergovernmental Committee on Surveying and Mapping (ICSM), Version 1.5.

IGS (2005). *IGS Products Website*. International GPS Service for Geodynamics. Available online: http://igscb.jpl.nasa.gov/components/prods.html.

Intel (2020). *Developer Guide for Intel Math Kernel Library 2020*. Intel Corporation. Available online: https://software.intel.com/content/www/us/en/develop/documentation/mkl-linux-developer-guide/top.html.

ISO (1995). *Guide to the Expression of Uncertainty in Measurement*. International Standards Organisation, Geneva, Switzerland.

ITRF (2000). *ITRF2000 transformation parameters*. Institut Géographique National. Available online: http://itrf.ensg.ign.fr/doc_ITRF/Transfo-ITRF2000_ITRFs.txt.

ITRF (2005). *ITRF2005 transformation parameters*. Institut Géographique National. Available online: http://itrf.ensg.ign.fr/doc_ITRF/Transfo-ITRF2005_ITRF2000.txt.

ITRF (2008). *ITRF2008 transformation parameters*. Institut Géographique National. Available online: http://itrf.ensg.ign.fr/doc_ITRF/Transfo-ITRF2008_ITRFs.txt.

ITRF (2016). *ITRF2014 transformation parameters*. Institut Géographique National. Available online: http://itrf.ensg.ign.fr/ITRF_solutions/2014/tp_14-08.php.

Jin, S. and Zhu, W. (2004). A revision of the parameters of the nnr-nuvel-1a plate velocity model. *Journal of Geodynamics 38*(1), 85–92.

Junkins, D. R. and Farley, S. A. (1995). *NTv2 Developer's Guide*. Geodetic Survey Division, Geomatics Canada.

Kolpackov, B. (2017). *Codesynthesis XSD — XML data binding for C++ (compiler software). Version 4.0.* Code Synthesis. Available online: http://codesynthesis.com/products/xsd/.

Kouba, J. (1970). Generalized sequential least squares expressions and matlan programming. Master's thesis, University of New Bunswick, Fredericton.

Kouba, J. (1972). Principle property and least squares adjustment. *The Canadian Surveyor 26*(2), 136–145.

Krakiwsky, E. J. (1975). A synthesis of recent advances in the method of least squares, lecure notes 42. 1994 reprint.

Kreemer, C., Blewitt, G., and Klein, E. C. (2014). A geodetic plate motion and global strain rate model. *Geochemistry, Geophysics, Geosystems 15*(10), 3849–3889.

Krüger, L. (1905). *Über die Ausgleichung von bedingten Beobachtungen in zwei Gruppen*. Number 18 in 4. Neue Folge, Potsdam.

Lambeck, K. (1963). *The Adjustment of Triangulation with Particular Reference to Subdivision into Phases*. Bachelor's thesis, University of New South Wales, Sydney.

Langley, R. (1996). Propagation of the GPS Signals. In *GPS for Geodesy*, Kleusberg, A. and Teunissen, P. (Eds.). Springer–Verlag, Berlin.

Larson, R., Hostetler, R., and Edwards, B. (1994). *Calculus with Analytic Geometry* (5th ed.). D. C. Heath & Company, Lexington, MA.

Lawson, C. L. and Hanson, R. J. (1995). *Solving least squares problems*, Volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. Revised reprint of the 1974 original.

Leahy, F. (1983). *Phased Adjustment of Large Networks*. Research Paper. Department of Surveying, University of Melbourne.

Leahy, F. (1996). *Kalman Filtering — An Introduction*. Lecture Notes. Department of Geomatics, University of Melbourne.

Leahy, F. (1999). *The Automatic Segmenting and Sequential Phased Adjustment of Large Geodetic Networks*. Ph. D. thesis, University of Melbourne.

Leahy, F. and Collier, P. (1998). Dynamic network adjustment and the transition to GDA94. *Australian Surveyor 43*(4), 261–272.

Leahy, F., Fuansumruat, W., and Norton, T. (1986). VICNET — geodetic adjustment software. *Australian Surveyor 33*(2), 86–91.

Leick, A. (1995). *GPS Satellite Surveying* (2nd ed.). John Wiley & Sons, New York.

Lin, K.-C. and Wang, J. (1995). Transformation from geocentric to geodetic coordinates using newton"s iteration. *Bulletin Géodésique 69*(4), 300–303.

López-Espín, J. J., Vidal, A. M., and Giménez, D. (2012). Two–stage least squares and indirect least squares algorithms for simultaneous equations models. *Journal of Computational and Applied Mathematics 236*(15), 3676–3684. Proceedings of the Fifteenth International Congress on Computational and Applied Mathematics (ICCAM-2010), Leuven, Belgium, 5-9 July, 2010.

Ltaief, H., Tomov, S., Nath, R., Du, P., and Dongarra, J. (2011). A scalable high performant Cholesky factorization for multicore with GPU accelerators. In *High Performance Computing for Computational Science — VECPAR 2010*, Palma, J. M. L. M., Daydé, M., Marques, O., and Lopes, J. C. (Eds.), pp. 93–101. Springer Berlin Heidelberg.

Merriman, M. (1884). *A Text Book on the Method of Least Squares*. John Wiley & Sons, New York.

Mikhail, E. (1976). *Observations and Least Squares*. IEP — A Dun-Donnelley Publisher, New York.

Mikhail, E. and Gracie, G. (1987). *Analysis and Adjustment of Survey Observations*. Van Nostrand Reinhold, New York.

Pinch, M. and Peterson, A. (1974). A method for adjusting survey networks in sections. *The Canadian Surveyor 28*(1), 3–24.

Prawirodirdjo, L. and Bock, Y. (2004). Instantaneous global plate motion model from 12 years of continuous GPS observations. *Journal of Geophysical Research: Solid Earth 109*(B8), 1–15.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2002). *Numerical Recipes in C: The Art of Scientific Computing* (2nd ed.). Cambridge University Press, Cambridge.

Prószynski, W. (1994). Criteria for internal reliability of linear least squares models. *Bulletin Géodésique 68*(3), 162–167.

Pursell, D. G. and Potterfield, M. (2008). *NAD 83(NSRS2007) National Readjustment Final Report*. NOAA Technical Report NOS NGS 60.

Rainsford, H. F. (1957). *Survey Adjustment and Least Squares*. Constable & Company Ltd.

Ramiro, C., López-Espín, J. J., Giménez, D., and Vidal, A. M. (2012). Two–stage least squares algorithms with QR decomposition for simultaneous equations models on heterogeneous multi-core and multi–GPU systems. *Procedia Computer Science 9*, 2004–2007. Proceedings of the International Conference on Computational Science, ICCS 2012.

Rapp, R. H. (1961). The orthometric height. M.s. Thesis, Ohio State University, Columbus, USA.

Rebischung, P., Griffiths, J., Ray, J., Schmid, R., Collilieux, X., and Garayt, B. (2012). IGS08: the IGS realization of ITRF2008. *GPS Solutions 16*(4), 483–494.

Redfearn, J. (1948). Transverse Mercator formulae. *The Empire Survey Review 69*, 218–322.

Schäling, B. (2014). *The Boost C++ libraries*. XML Press. Available online: http://boost.org.

Schenk, O. and Gärtner, K. (2002). Two–level dynamic scheduling in PARDISO: Improved scalability on shared memory multiprocessing systems. *Parallel Computing 28*(2), 187–197.

Schmid, H. and Schmid, E. (1965). Estimation of geodetic network station coordinates via phased least squares adjustment — an overview. *Canadian Surveyor 19*(1), 27–41.

Shi, Y., Xu, P., and Peng, J. (2017). A computational complexity analysis of Tienstra's solution to equality–constrained adjustment. *Studia Geophysica et Geodaetica 61*, 601–615.

Snay, R. A. (1990). Accuracy analysis for the NAD 83 geodetic reference system. *Bulletin Géodésique 64*(1), 1–27.

Snyder, J. (1997). *Map Projections: A Working Manual*. United States Government Printing.

Stoughton, H. W. (1989). *The North American Datum of 1983. A Collection of Papers Describing the Planning and Implementation of the Readjustment of the North American Horizontal Network*. American Congress on Surveying and Mapping, Virginia.

Strang, G. and Borre, K. (1997). *Linear Algebra, Geodesy, and GPS*. Wellesley–Cambridge Press, MA.

Teunissen, P. J. G. (2003). *Adjustment theory*. VSSD.

Teunissen, P. J. G. (2006). *Testing theory* (2nd ed.). VSSD.

Tienstra, J. M. (1956). *Theory of Adjustment of Normally Distributed Observations*. Argus, Amsterdam.

TOP500 (2020). TOP500 LIST — JUNE 2020. https://www.top500.org/lists/top500/list/2020/06/. Accessed: 2020-07-10.

Van Huffel, S. and Vandewalle, J. (1988). The partial total least squares algorithm. *Journal of Computational and Applied Mathematics 21*, 333–341.

Vaniček, P. and Krakiwsky, K. (1986). *Geodesy: The Concepts*. North Holland, Amsterdam.

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review 12*(176), 88–93.

Wolf, H. (1978). The Helmert Block method — its origin and development. In *Annals of the 2nd International Symposium on Problems Related to the Redefinition of North American Geodetic Networks, Washington*, Verastegui, B. (Ed.), pp. 319–326. US Department of Commerce.

Ying Chung–Chi (1970). On phased adjustment. *Survey Review 20*(156, 157), 282–288, 298–309.

Index of subjects

Appendix A

Command line reference

A.1 dynadjust

Basic usage

To automate the processing of a geodetic network adjustment in a single step using a project file, type `dynadjust` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> dynadjust [options]
```

The options available to **dynadjust** are described below. The term `arg` denotes a required argument.

Standard options

Table A.1 lists the standard options.

Table A.1: **dynadjust** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. This file contains all user options for import, segment, geoid, reftran, adjust and plot.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.
`--import`	Causes **dynadjust** to invoke **Import** – DynAdjust file exchange software.
`--geoid`	Causes **dynadjust** to invoke **geoid** – geoid model interpolation software to determine geoid–ellipsoid separation and deflection values.
`--reftran`	Causes **dynadjust** to invoke **reftran** – reference frame transformation software to transform stations and measurements.
`--segment`	Causes **dynadjust** to invoke **segment** – automated segmentation software to partition a geodetic network into smaller sized blocks.
`--adjust`	Causes **dynadjust** to invoke **adjust** – geodetic network adjustment software.

Generic options

Table A.2 lists the generic options for **dynadjust**. These options are common to all DynAdjust programs and as such, reference will be made to this section from the command line reference for other programs.

Table A.2: **dynadjust** generic options

Option	Comments and Result
`--verbose-level arg`	Give detailed information about what **dynadjust** is doing, where `arg` produces the following results: 0. No information (default) 1. Helpful information 2. Extended information 3. Debug level information
`--quiet`	Suppress all explanation of what **dynadjust** is doing unless an error occurs.
`--version`	Display the version information for **dynadjust**.
`--help`	Show the **dynadjust** command line reference.
`--help-module`	Provide help for a specific help category.

A.2 import

Basic usage

To import station and measurement information into DynAdjust using **import**, type `import` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> import [options] [files]...
```

The options available to **import** are described below. The term `arg` denotes a required argument.

Standard options

Table A.3 lists the standard options. At least one option is mandatory, either `--project-file` or `--network-name`.

Table A.3: **import** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **import** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where `#` is incremented until the first available network is reached.
`--stn-msr-file arg`	`arg` is a station or measurement file name. Any number of station and measurement files can be passed to **import** in any order. This is the default option to **import** and as such, files may be passed to **import** without the option text **`--stn-msr-file`**.
`--geo-file arg`	`arg` is a geoid file formatted according to the DNA Version 1.0 file format.
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **import** assumes all input files are located in the folder from which **import** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **import** will create all output files in the folder from which **import** is executed.
`--binary-stn-file arg`	`arg` is the binary station output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--binary-msr-file arg`	`arg` is the binary measurement output file name. `arg` overrides the default file name formed from the argument to `--network-name`.

Reference frame options

Table A.4 lists the reference frame options.

Table A.4: **import** reference frame options

Option	Comments and Result
`--reference-frame arg`	When input files do not specify a reference frame, `arg` is the default reference frame to be assigned to all imported stations and measurements, and to be used for preliminary reductions on the ellipsoid. The default reference frame is `GDA94`.
`--override-input-ref-frame`	Replace the reference frame specified in the input files with the reference frame specified by `arg` in `--reference-frame` .

Data screening options

Table A.5 lists the data screening options.

Table A.5: **import** data screening options

Option	Comments and Result
`--bounding-box arg`	`arg` is a comma delimited string in the form `lat1,lon1,lat2,lon2` (i.e. latitude, longitude pairs in dd.mmss) defining the upper–left and lower–right limits of a bounding box within which to extract stations and measurements from the input files. A measurement must be wholly located within the bounding box for its successful inclusion.
`--get-msrs-transcending-box`	Include measurements which transcend the bounding box, including the associated stations which lie outside the bounding box.
`--include-stns-assoc-msrs arg`	`arg` is a comma delimited string `stn1,stn2,stn3,stn4` defining the stations and all associated measurements to include from the input files. All other stations are excluded.
`--exclude-stns-assoc-msrs arg`	`arg` is a comma delimited string `stn1,stn2,stn3,stn4` defining the stations and all associated measurements to exclude from the input files. All other stations are included.
`--split-gnss-cluster-msrs`	Split GNSS point and baseline cluster measurements, causing **import** to ignore points or baselines within the cluster which lie outside the user defined bounding box, or are not associated with the list of stations specified by `--include-stns-assoc-msrs arg`.
`--import-block-stn-msr arg`	`arg` is the segmented network block number from which to extract stations and all associated measurements. A corresponding `.seg` file must exist for this option.
`--seg-file arg`	`arg` is the name of the segmentation file from which to determine the segmentation information. `arg` overrides default file name formed from the argument to `--network-name`.
`--prefer-single-x-as-g`	Convert baseline cluster measurements (`X`) with only one baseline to a single baseline measurement (`G`).
`--include-msr-types arg`	`arg` is a non-delimited string (eg `GXY`) defining the list of measurement types to be included. All other measurement types will be excluded.
`--exclude-msr-types arg`	`arg` is a non-delimited string (eg `IJK`) defining the list of measurement types to be excluded. All other measurement types will be included.
`--stn-renaming-file`	`arg` is the name of the station renaming file containing the preferred station name and the aliases to be replaced.
`--discontinuity-file arg`	`arg` is the name of the station discontinuity file containing known discontinuities for named stations in the input files.
`--search-nearby-stn`	Invoke a search for nearby stations that are within a radius of `--nearby-stn-buffer arg`.
`--nearby-stn-buffer arg`	`arg` is the radius of the circle within which to search for nearby stations. Default is 0.3m.
`--search-similar-gnss-msr`	Invoke a search and provide warnings for GNSS baselines which appear to be similar to GNSS baseline clusters
`--search-similar-msr`	Invoke a search and provide warnings for similar measurements.

Option	Comments and Result
`--ignore-similar-msr`	Ignore similar measurements.
`--remove-ignored-msr`	Exclude ignored measurements from the import process.
`--flag-unused-stations`	Mark unused stations in binary file. Stations marked will be excluded from any further processing.
`--test-integrity`	Test the integrity of the association lists and binary files.

GNSS variance matrix scaling options

Table A.6 lists the GNSS variance matrix scaling options.

Table A.6: **import** GNSS variance matrix scaling options

Option	Comments and Result
`--v-scale arg`	`arg` is a global *variance* matrix scalar for all GNSS measurements. Replaces existing scalar. Default is 1.
`--p-scale arg`	`arg` is a *latitude* variance matrix scalar for all GNSS measurements. Replaces existing scalar. Default is 1.
`--l-scale arg`	`arg` is a *longitude* variance matrix scalar for all GNSS measurements. Replaces existing scalar. Default is 1.
`--h-scale arg`	`arg` is a *height* variance matrix scalar for all GNSS measurements. Replaces existing scalar. Default is 1.
`--baseline-scalar-file arg`	`arg` is a file containing global, latitude, longitude and height variance matrix scalars for GNSS baseline measurements between specific station pairs. Scalar file values do not apply to GNSS point or baseline clusters.

Network simulation options

Table A.7 lists the network simulation options.

Table A.7: **import** network simulation options

Option	Comments and Result
`--simulate-msr-file`	Simulate measurements corresponding to the input measurements using the coordinates in the station file. If a geoid file was imported using `--geo-file`, **import** will apply the geoid–ellipsoid separations and deflections of the vertical to the simulated measurements.

Output options

Table A.8 lists the output options.

Table A.8: **import** output options

Option	Comments and Result
`--output-msr-to-stn`	Print a summary of the measurements connected to each station.

Export options

Table A.9 lists the file export options.

Table A.9: **import** export options

Option	Comments and Result
`--export-xml-files`	Export stations and measurements to DynaML (DynAdjust XML) format.
`--single-xml-file`	Create a single DynaML file containing all stations and measurements.
`--export-dna-files`	Export stations and measurements to DNA STN and MSR format.
`--export-asl-file`	Export the ASL file as raw text.
`--export-aml-file`	Export the AML file as raw text.
`--export-map-file`	Export the MAP file as raw text.
`--export-discont-file`	Export discontinuity information as raw text.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **import**.

A.3 reftran

Basic usage

To transform imported station coordinates and GNSS measurement into a common reference frame using **reftran**, type `reftran` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> reftran network-name [options]
```

The options available to **reftran** are described below. The term `arg` denotes a required argument.

Standard options

Table A.10 lists the standard options.

Table A.10: **reftran** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **reftran** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **reftran** assumes all input files are located in the folder from which **reftran** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **reftran** will create all output files in the folder from which **reftran** is executed.
`--binary-stn-file arg`	`arg` is the binary station output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--binary-msr-file arg`	`arg` is the binary measurement output file name. `arg` overrides the default file name formed from the argument to `--network-name`.

Transformation options

Table A.11 lists the reference frame transformation options.

Table A.11: **reftran** transformation options

Option	Comments and Result
`--reference-frame arg`	`arg` is the target reference frame for all stations and datum-dependent measurements.
`--epoch arg`	`arg` is a dot delimited string in the form of `dd.mm.yyyy` defining the projected date for the transformed stations and measurements. If today's date is required, type `today`. If no date is supplied, the reference epoch of the supplied reference frame will be used.
`--plate-model-option`	Plate motion model option. 0. Assume all stations are on the Australian plate (default) 1. Interpolate plate motion model parameters from a defined set of global tectonic plates. For this option, a global tectonic plate boundary file and corresponding Euler pole parameters file must be provided.
`--plate-boundary-file`	Global tectonic plate boundaries.
`--plate-pole-file`	Euler pole parameters corresponding to the global tectonic plate boundaries supplied with option `--plate-boundary-file`.

Export options

Table A.12 lists the file export options.

Table A.12: **reftran** export options

Option	Comments and Result
`--export-xml-files`	Export stations and measurements to DynaML (DynAdjust XML) format.
`--single-xml-file`	Create a single DynaML file containing all stations and measurements.
`--export-dna-files`	Export stations and measurements to DNA STN and MSR format.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **reftran**.

A.4 geoid

Basic usage

To introduce geoid–ellipsoid separations and deflections of the vertical into a project from either an NTv2 formatted geoid model or ASCII text file using **geoid**, type `geoid` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> geoid [options]
```

The options available to **geoid** are described below. The term `arg` denotes a required argument.

Standard options

Table A.13 lists the standard options.

Table A.13: **geoid** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **geoid** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **geoid** assumes all input files are located in the folder from which **geoid** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **geoid** will create all output files in the folder from which **geoid** is executed.

Interpolation options

Table A.14 lists the geoid grid file interpolation options.

Table A.14: **geoid** interpolation options

Option	Comments and Result
`--interactive`	Perform geoid interpolation from coordinates entered via the command line.
`--text-file arg`	Perform geoid interpolation from coordinates contained in a text file. `arg` is the full path to the text file to be transformed.
`--interpolation-method arg`	`arg` is an integer informing **geoid** which interpolation method to use: 0. Bi–linear 1. Bi–cubic (default)
`--decimal-degrees`	Informs **geoid** that all input coordinates are in decimal degrees. Default coordinate format is degrees, minutes and seconds.
`--create-ntv2`	Create an NTv2 binary grid file from the legacy Winter DAT file format.
`--ntv2-file arg`	`arg` is the full path to the NTv2 grid file.
`--summary`	Print a summary of the grid file.

NTv2 creation options

Table A.15 lists the National Transformation version 2.0 (NTv2) file creation options.

Table A.15: **geoid** NTv2 creation options

Option	Comments and Result
`--dat-file arg`	`arg` is the full path to the WINTER DAT grid file.
`--grid-shift-type arg`	`arg` is the grid shift type. Valid values include `SECONDS` or `RADIANS`.
`--grid-version arg`	`arg` is the grid file version.
`--system-from arg`	`arg` is the 'from' reference system (e.g. `GDA94`).
`--system-to arg`	`arg` is the 'to' reference system (e.g. `AHD_1971`).
`--semi-major-from arg`	`arg` is the semi major axis of the 'from' reference system.
`--semi-major-to arg`	`arg` is the semi major axis of the 'to' reference system.
`--semi-minor-from arg`	`arg` is the semi minor axis of the 'from' reference system.
`--semi-minor-to arg`	`arg` is the semi minor axis of the 'to' reference system.
`--sub-grid-name arg`	`arg` is the name of the subgrid.
`--creation-date arg`	`arg` is a dot delimited string in the form of `dd.mm.yyyy` defining the date of file creation. Default is today's date if no value is supplied.
`--update-date arg`	`arg` is a dot delimited string in the form of `dd.mm.yyyy` defining the date of the last file update. Default is today's date if no value is supplied.

Interactive interpolation options

Table A.16 lists the interactive interpolation options.

Table A.16: **geoid** interactive interpolation options

Option	Comments and Result
`--latitude arg`	`arg` is the latitude of the interpolant. Default is degrees, minutes and seconds in the form dd.mmssss.
`--longitude arg`	`arg` is the longitude of the interpolant. Default is degrees, minutes and seconds in the form dd.mmssss.

File interpolation options

Table A.17 lists the file interpolation options.

Table A.17: **geoid** file interpolation options

Option	Comments and Result
`--direction arg`	`arg` is an integer informing **geoid** which direction the conversion of user–supplied heights should be: 0. Orthometric to ellipsoid 1. Ellipsoid to orthometric (default)
`--convert-station-hts`	Informs **geoid** to convert orthometric heights in the binary station file to ellipsoidal heights.

Export options

Table A.18 lists the export options.

Table A.18: **geoid** export options

Option	Comments and Result
`--export-dna-geo-file`	Creates a DynAdjust version 1.0 geoid file from interpolated geoid information.
`--export-ntv2-asc-file`	Export a binary NTv2 geoid file to ASCII (`.asc`) format.
`--export-ntv2-gsb-file`	Export an ASCII NTv2 geoid file to binary (`.gsb`) format.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **geoid**.

A.5 segment

Basic usage

To segment a network into a series of blocks for use by **adjust** in phased adjustment mode, type `segment` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> segment network-name [options]
```

The options available to **segment** are described below. The term `arg` denotes a required argument.

Standard options

Table A.19 lists the standard options.

Table A.19: **segment** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **segment** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **segment** assumes all input files are located in the folder from which **segment** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **segment** will create all output files in the folder from which **segment** is executed.
`--binary-stn-file arg`	`arg` is the binary station output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--binary-msr-file arg`	`arg` is the binary measurement output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--seg-file arg`	`arg` is the segmentation output file name. `arg` overrides the default file name formed from the argument to `--network-name`.

Configuration options

Table A.20 lists the segmentation configuration options.

Table A.20: **segment** configuration options

Option	Comments and Result
`--net-file`	Look for a file named `network-name.net`, which contains the list of stations to be incorporated within the first block.
`--starting-stns arg`	`arg` is a comma delimited string in the form of "`stn1, stn 2,stn3 , stn 4`", which defines a list of additional stations to be incorporated within the first block.
`--min-inner-stns arg`	`arg` is the minimum number of inner stations that must appear within each block. Default is `5`.
`--max-block-stns arg`	`arg` is the threshold defining the maximum number of stations per block. Default is 65.
`--contiguous-blocks arg`	`arg` is an integer which informs **segment** how to treat isolated networks: 0. Isolated networks as individual blocks 1. Force production of contiguous blocks (default)
`--search-level arg`	`arg` is the level to which searches should be conducted to find stations with the lowest measurement count. Default is `0`.
`--test-integrity`	Test the integrity of the segmentation output file.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **segment**.

A.6 adjust

Basic usage

To perform a least squares adjustment using **adjust**, type `adjust` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> adjust network-name [options]
```

The options available to **adjust** are described below. The term `arg` denotes a required argument.

Standard options

Table A.28 lists the standard options.

Table A.21: **adjust** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **adjust** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **adjust** assumes all input files are located in the folder from which **adjust** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **adjust** will create all output files in the folder from which **adjust** is executed.
`--binary-stn-file arg`	`arg` is the binary station output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--binary-msr-file arg`	`arg` is the binary measurement output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--seg-file arg`	`arg` is the name of an alternative segmentation file. `arg` overrides the default file name formed from the argument to `--network-name`.
`--comments arg`	`arg` is a string of comments to be associated with this adjustment. If `arg` contains whitespace, ensure that `arg` is enclosed with double quotes, e.g. "`Constraints assumed from GNSS CORS network.`".

Adjustment mode options

Table A.22 lists the adjustment mode options.

Table A.22: **adjust** adjustment mode options

Option	Comments and Result
`--simultaneous-adjustment`	Adjust the network in simultaneous adjustment mode. The default mode.
`--phased-adjustment`	Adjust the network in sequential phased adjustment mode. Refer to Phased adjustment options for additional options.
`--simulation`	Perform a simulation adjustment, resulting in rigorous variance matrices for station coordinates and adjusted measurements. Rigorous station coordinates are not produced.
`--report-results`	Reproduce the adjustment output files without performing an adjustment.

Phased adjustment options

Table A.23 lists the phased adjustment mode options.

Table A.23: **adjust** phased adjustment options

Option	Comments and Result
`--staged-adjustment`	Store adjustment matrices in memory mapped files instead of retaining data in memory. This option decreases efficiency but may be required if there is insufficient RAM to hold an adjustment in memory. Refer to Staged adjustment options for additional options.
`--multi-thread`	Process forward, reverse and combination adjustments concurrently using all available CPU cores. This option yields maximum efficiency but requires a RAM footprint twice the size of a phased adjustment.
`--block1-phased`	Adjust the network in sequential phased adjustment mode, resulting in rigorous estimates for block 1 only.

Adjustment configuration options

Table A.24 lists the adjustment configuration options.

Table A.24: **adjust** adjustment configuration options

Option	Comments and Result
`--conf-interval arg`	`arg` is the confidence interval to be used for testing the least squares solution and measurement corrections. Default is `95.0%`.
`--iteration-threshold arg`	`arg` is the least squares iteration threshold. Default is `0.0005m`.
`--max-iterations arg`	`arg` is the maximum number of iterations. Default is `10`.
`--constraints arg`	`arg` is a comma delimited string in the form of `station,constraint` pairs (e.g. "`stn 1,CCC,stn 2,CCF`") defining specific station constraints. These values override those contained in the station file.
`--free-stn-sd arg`	`arg` is the a–priori standard deviation for free stations. Default is `10m`.
`--fixed-stn-sd arg`	`arg` is the a–priori standard deviation for fixed stations. Default is `0.000001m`.
`--scale-normals-to-unity`	Scale the normal matrices to unity prior to computing the inverse to minimise loss of precision caused by tight variances placed on constraint stations.

Staged adjustment options

Table A.25 lists the staged adjustment options.

Table A.25: **adjust** staged adjustment options

Option	Comments and Result
`--create-stage-files`	Recreate memory mapped files. If not specified, **adjust** will attempt to load all information from existing files.
`--purge-stage-files`	Purge memory mapped files from disk upon completing the adjustment.

Output options

Table A.26 lists the adjustment output options. All output options affect the `.adj` file only.

Table A.26: **adjust** adjustment output options

Option	Comments and Result
`--output-msr-to-stn`	Print a summary of the measurements connected to each station.
`--output-iter-adj-stn`	Print the adjusted station coordinates on each iteration.
`--output-iter-adj-stat`	Print a statistical summary on each iteration.
`--output-iter-adj-msr`	Print the adjusted measurements on each iteration.
`--output-iter-cmp-msr`	Print the computed measurements on each iteration.
`--output-adj-msr`	Print the final adjusted measurements.
`--output-adj-gnss-units arg`	arg is an integer defining the units for the adjusted GNSS measurements: 0. As measured (default) 1. Local [east, north, up] 2. Polar [azimuth, vert. angle, slope dist] 3. Polar [azimuth, slope dist, up]
`--output-tstat-adj-msr`	Print Student t statistics for adjusted measurements.
`--output-database-ids`	Print measurement ID and cluster ID for database mapping.
`--output-ignored-msrs`	Print the ignored measurements and the differences to the equivalent measurements computed from the adjusted station coordinates.
`--sort-adj-msr-field arg`	arg is an integer defining the sort order for the adjusted measurements: 0. Original input file order (default) 1. Measurement type 2. Station 1 3. Station 2 4. Measurement value 5. Correction 6. Adjusted std. dev. 7. N-statistic 8. Suspected outlier

Option	Comments and Result
`--output-stn-blocks`	For phased adjustments, print the adjusted coordinates according to each block.
`--output-msr-blocks`	For phased adjustments, print the adjusted measurements according to each block.
`--sort-stn-orig-order`	Print station information using the station order in the original station file. By default, stations are output in alpha–numeric order.
`--stn-coord-types arg`	`arg` is a case–sensitive string in the form of "`ENzPLHhXYZ`", defining the specific types and order for the output of adjusted station coordinates. Default is "`PLHhXYZ`".
`--stn-corrections`	Print the corrections (in the form of e, n, up) to the initial station coordinates with the adjusted station coordinates.
`--precision-stn-linear arg`	`arg` is an integer defining the precision for linear station coordinates. Default is `4`.
`--precision-stn-angular arg`	`arg` is an integer defining the precision for angular station coordinates. For values in degrees, minutes and seconds, `arg` relates to seconds. For values in decimal degrees, `arg` relates to degrees. Default is `5`.
`--precision-msr-linear arg`	`arg` is an integer defining the precision for linear measurements. Default is `4`.
`--precision-msr-angular arg`	`arg` is an integer defining the precision for angular measurements. For values in degrees, minutes and seconds, `arg` relates to seconds. For values in decimal degrees, `arg` relates to degrees. Default is `4`.
`--angular-msr-type arg`	`arg` is an integer defining the output type for adjusted angular measurements: 0. Degrees, minutes and seconds (default) 1. Decimal degrees
`--dms-msr-format arg`	`arg` is an integer defining the output format for adjusted angular measurements in degrees, minutes and seconds: 0. Separated fields (default) 1. Separated fields with symbols 2. HP notation

Export options

Table A.27 lists the adjustment export options.

Table A.27: **adjust** export options

Option	Comments and Result
`--output-pos-uncertainty`	Print positional uncertainty and variances of the adjusted station coordinates to the `.apu` file.
`--output-all-covariances`	Print covariances between adjusted station coordinates to the `.apu` file.

Option	Comments and Result
`--output-apu-vcv-units arg`	`arg` is an integer defining the variance matrix units in the `.apu` file: 0. Cartesian [X,Y,Z] (default) 1. Local [e,n,up]
`--output-corrections-file`	Print the corrections (in the form of azimuth, distance, e, n, up) to the initial station coordinates to the `.cor` file.
`--hz-corr-threshold arg`	`arg` is the minimum horizontal threshold by which to restrict output of station corrections to `.cor` file. Default is `0.0m`.
`--vt-corr-threshold arg`	`arg` is the minimum vertical threshold by which to restrict output of station corrections to `.cor` file. Default is `0.0m`.
`--export-xml-stn-file`	Export the estimated station coordinates to a DynaML (DynAdjust XML) station file.
`--export-dna-stn-file`	Export the estimated station coordinates to a DNA STN file.
`--export-sinex-file`	Export the estimated station cooridnates and full variance matrix to a SINEX file.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **adjust**.

A.7 plot

Basic usage

To generate a PDF or PNG image of the network using **plot**, type `plot` at the command prompt, followed by one or more *options* and, if required, option *arguments*:

```
> plot network-name [options]
```

The options available to **plot** are described below. The term `arg` denotes a required argument.

Standard options

Table A.28 lists the standard options.

Table A.28: **plot** standard options

Option	Comments and Result
`--project-file arg`	`arg` is the full path to the DynAdjust project file. Using this option alone, **plot** will configure all other options using the values contained in the DynAdjust project file. If no file is specified, a new file is created using default and user–specified options.
`--network-name arg`	`arg` is the network name for the DynAdjust project and is used to form the name of all input and output files. If no name is provided, a default name of "`network#`" will be adopted, where # is incremented until the first available network is reached.

Option	Comments and Result
`--input-folder arg`	`arg` is the path containing all input files. If not specified, **plot** assumes all input files are located in the folder from which **plot** is executed.
`--output-folder arg`	`arg` is the path for the creation of all output files. If not specified, **plot** will create all output files in the folder from which **plot** is executed.
`--binary-stn-file arg`	`arg` is the binary station output file name. `arg` overrides the default file name formed from the argument to `--network-name`.
`--binary-msr-file arg`	`arg` is the binary measurement output file name. `arg` overrides the default file name formed from the argument to `--network-name`.

Data configuration options

Table A.29 lists the data configuration options.

Table A.29: **plot** data configuration options

Option	Comments and Result
`--plot-msr-types arg`	`arg` is a non-delimited string (eg `"GXY"`) defining the specific measurement types to be printed.
`--plot-ignored-msrs`	Plot ignored measurements.
`--phased-block-view`	Plot the blocks of a segmented network in individual sheets. A corresponding `.seg` file must exist for this option.
`--seg-file arg`	`arg` is the name of an alternative segmentation file. `arg` overrides the default file name formed from the argument to `--network-name`.
`--block-number arg`	When plotting phased adjustments, `arg` is the number of the block to be printed. A value of zero (default) causes all blocks to be plotted. A corresponding `.seg` file must exist for this option.
`--label-stations`	Label each station with the station name.
`--alternate-name`	When station labels are required, label each station with the station description string, rather than the station name.
`--label-constraints`	Appends each station label with the station constraints. Applies to the `--label-stations` option.
`--correction-arrows`	Plot arrows representing the direction and magnitude of corrections to the original station coordinates.
`--label-corrections`	Plot correction labels.
`--omit-measurements`	Prevent measurements from being printed.
`--compute-corrections`	Compute the corrections to the original station coordinates from binary station file. Default is to use the values contained in the `.cor` file.
`--error-ellipses`	Plot error ellipses for all stations.
`--positional-uncertainty`	Plot positional uncertainties for all stations.
`--scale-arrows arg`	`arg` is the amount by which to scale the size of the correction arrows.
`--scale-ellipse-circles arg`	`arg` is the amount by which to scale the size of error ellipses positional uncertainty circles.

Option	Comments and Result
`--bounding-box arg`	`arg` is a comma delimited string in the form `lat1,lon1,lat2,lon2` (i.e. latitude, longitude pairs in dd.mmss) defining the upper–left and lower–right limits of the desired plot extents.
`--centre-latitude arg`	`arg` is the latitude, in the form of ±`dd.mmsssss`, upon which to centre the plot. The plot area is circumscribed by `--area-radius`.
`--centre-longitude arg`	`arg` is the longitude, in the form of ±`ddd.mmsssss`, upon which to centre the plot. The plot area is circumscribed by `--area-radius`.
`--centre-station arg`	`arg` is the station name upon which to centre the plot. The plot area is circumscribed by `--area-radius`.
`--area-radius arg`	`arg` is the radius (in metres) of a circular perimeter by which to limit plots that are to be centered upon a station or latitude/longitude pair. Default is `5000m`.
`--graph-stn-blocks`	Plot a graph of the block stations resulting from network segmentation.
`--graph-msr-blocks`	Plot a graph of the block measurements resulting from network segmentation.
`--keep-gen-files`	Keep command and data files used to generate EPS and PDF plots. Default behaviour is to delete these files once the PDF has been generated.

Mapping options

Table A.30 lists the mapping options.

Table A.30: **plot** mapping options

Option	Comments and Result
`--map-projection arg`	`arg` is an integer defining the map projection to use for the display of the network: 0. Let **plot** choose best projection (default) 1. World plot 2. Orthographic 3. Mercator 4. Transverse Mercator 5. Albers equal-area conic 6. Lambert conformal conic 7. General stereographic 8. Robinson
`--label-font-size arg`	`arg` is an integer defining the font size for station name and constraint labels. Default is `5`.
`--msr-line-width arg`	`arg` is an integer defining the measurement line width. Default is `0.15`.
`--supress-pdf-creation`	Don't create a pdf, just the command files.
`--export-png`	Export the GMT plot to png at 300 dpi.

Title block options

Table A.31 lists the title block options.

Table A.31: **plot** title block options

Option	Comments and Result
`--omit-title-block`	Prevent **plot** from printing a title block and measurements legend to the network map.
`--title arg`	The title of the plot. Supply quotation marks if spaces are required. Default is the network name.
`--org-unit-name arg`	The name of the organisational unit name. Default is Surveyor-General Victoria.
`--org-subunit-name arg`	The name of the organisational sub-unit unit name. Default is Geodesy.

Generic options

Please refer to the Generic options for **dynadjust** on page 164 for a list of generic options for **plot**.

Appendix B

File format specification

B.1 Dynamic Network Adjustment (DNA) format

The DNA format provides information about geodetic network stations, measurements, and geoid–ellipsoid separations and deflections of the vertical. This information is contained in three separate files — the station file has a `stn` file extension, the measurement file has a `msr` extension and the geoid file has a geo extension. All three files must contain a single header line at the beginning of the file. As required, these files can contain one or more comment lines throughout the file. Comment lines must begin with an asterisk (*). Every record in the station, measurement and geoid files must contain data fields located in particular file positions (or columns).

Version 1.0 of the DNA format was originally defined for the first version of DynAdjust, formerly known as DNA. The following specifications relate to version 3.01.

B.1.1 Changes from Version 1 to Version 3

DNA Version 3 introduces the following changes:

- Version control information within the header record;
- Various changes in field locations and widths to support larger station names (up to 20 characters), and extra precision for GNSS measurements;
- A new coordinate format identifier to enable ellipsoid and orthometric heights in the station file to be distinguished;
- GNSS measurement variance matrix scalars, and;
- GNSS measurement reference frame and epoch.

B.1.2 Header line

DNA Version 3 introduces formatting to the header line to distinguish which DNA version a file has been formatted in, and to provide basic metadata relevant to the file and its contents. A single header line must be present at the beginning of every station file, measurement file and geoid file. The relevant fields and corresponding file positions for the header line are shown in Table B.1.

The reference frame/geoid model and epoch/version fields are non–mandatory and are used to indicate the default parameters for the file. The epoch field relates to the reference frame, and the version field relates to the geoid model. If this information is absent, DynAdjust will set the default reference frame for stations and measurements to `GDA94` and the default geoid model for geoid–ellipsoid separations and deflections of the vertical to `AusGeoid09`. For measurement files, the default reference frame and epoch will be overridden by the reference frame and epoch provided for each GNSS measurement. The measurement count specified in columns 58 – 67 relates to the number of measurement sets. Hence, a GNSS point cluster of four points is regarded as one measurement, not four points or twelve measurement components. The same applies to GNSS baseline clusters and direction sets.

Table B.1: Header line column locations and field widths

Field	**Columns**	**Width**	**Examples**
File format	1 – 6	6	`!#=DNA`
Version	7 – 12	6	`3.01`
File type	13 – 15	3	`STN`, `MSR`, `GEO`, `REN`
Creation date	16 – 29	14	`01.03.2013`
Default reference frame or geoid model	30 – 43	14	`GDA94`, `ITRF2005`, `AUSGEOID09`
Default epoch or version	44 – 57	14	`01.01.2010`, `1.0.0.0`
Number of stations or measurements	58 – 67	10	`100`

Figure B.1 shows some example header lines in DNA format for a station file, a measurement file and a geoid file. Note that all values can be positioned anywhere within the respective fields without the need for right or left justification.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
!#=DNA 3.00 STN    28.08.2013         GDA94                      43
```

```
!#=DNA 3.00 MSR    28.08.2013      ITRF2005    01.01.2010       121
```

```
!#=DNA 3.00 GEO    28.08.2013    AUSGEOID09       1.0.0.0        16
```

```
!#=DNA 3.01 REN                                                  16
```

Figure B.1: Example header lines

B.1.3 Station information

All information for a single station is formatted on one line. DynAdjust allows station coordinate information in a DNA station file to be stored in one of three different coordinate types:

`LLH`/`LLh` Geographic coordinates. Expected values are *latitude*, *longitude* and *height*. Latitude and longitude must be expressed in degrees, minutes and seconds using HP notation (e.g. dd.mmssssss). Height must be expressed in metres and can be provided as either an ellipsoidal or orthometric height. `LLH` is case sensitive — specify `LLH` for orthometric height and `LLh` for ellipsoidal height.

`UTM` Universal Transverse Mercator coordinates. Expected values are *Easting*, *Northing*, *zone* and *height*. Easting and Northing values must be expressed in metres. Zone is an integer value. Height is assumed to be orthometric.

`XYZ` Earth–centred cartesian coordinates. Expected values are x, y, z. All elements must be expressed in metres.

Information in a station file can be formatted using multiple coordinate types. That is, there is no requirement for all stations in a file to be formatted according to one coordinate type. The relevant fields and corresponding file positions for station information are shown in Table B.2.

Table B.2: Station information column locations and field widths

Field	Columns	Width	Comments
Station name	1 – 20	20	Alphanumeric string. Can contain spaces.
Constraints	21 – 23	3	Constraints must be supplied according to the coordinate order, where F = free and C = constrained, e.g. FFC.
Blank	24	1	Ignored.
Coordinate type	25 – 27	3	LLH/LLh, UTM or XYZ.
Latitude / Easting / x	28 – 47	20	Latitude is represented as dd.mmssss. Easting and x is in metres.
Longitude / northing / y	48 – 67	20	Longitude is represented as ddd.mmssss. Northing and y are in metres.
Height / z	68 – 87	20	Height and z are in metres.
Hemisphere and zone	88 – 90	3	Hemisphere (N or S) and zone (e.g. S55). This field is compulsory for UTM coordinates only. Hemisphere is optional (S is the default if left blank).
Blank	91	1	Ignored.
Description	92	128	Alphanumeric string. Not compulsory.

Figure B.2 shows some example stations in DNA format. Again, the values can be positioned anywhere within the respective fields without the need for right or left justification.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
!#=DNA 3.01 STN    16.10.2012         GDA94                      14
* Created 02 August 2013.
* Sample stations in UTM
* Level section 103
103-E               CCF UTM    554222.559   5760821.962      5.013    54 PTL
* Level section 748
748-16              CCF UTM    635671.48    5945852.24       137.515  54 BM 16
748-15              CCF UTM    634398.61    5945979.47       136.821  54 BM 15
* Level section 1002
1002-7              CCF UTM    340477.173   5913039.786      166.790  55 BM 7
1002-9              CCF UTM    343140.      5914120.         151.693  55 BM 9
1002-26             CCF UTM    372772.813   5931491.145      175.668  55 BM 26
1002-28             CCF UTM    350360.      5919700.         177.278  55 BM 28
* Sample stations in LLH
HOSP                CCF LLH-37.4801765569144.5717295509       83.9000    HOSP
PELH                FFF LLH-37.4813731558144.5734312921       31.0810    PELH
RMH-FP              CCC LLH-37.4801998629144.5717239117      100.0000 55 RMH Flag Pole
ramp                FFF LLH  -37.47053445  144.57103055        41.669 55 Zoo ramp
* Sample stations in XYZ
Zoo3                FFF XYZ -4131942.8194  2898817.3761 -3886309.6280    Zoo point 3
AAAAC               CCC XYZ -4373792.5877  3074983.9818 -3466230.4471
AAAAD               FFF XYZ -3227491.9637  4780697.6162 -2712698.5913
```

Figure B.2: Example station file with coordinates in UTM, LLH and XYZ formats

B.1.4 Measurement information

Depending on the type of measurement, information for a measurement (or cluster of measurements) can be contained on one or multiple lines. The measurement types supported by DynAdjust, Version 1.1 are listed in Table 3.2, on page 19.

The required information and formatting for a measurement will vary according to the type of measurement. However, there are certain fields which are common to all measurement types. Table B.3 shows the fields and file formatting common to all measurements.

Table B.3: General measurement information column locations and field widths

Field	Columns	Width	Comments
Measurement type	1	1	Alpha character.
Ignore flag	2	1	If a measurement is to be ignored, supply an asterisk ($*$).
First station name	3 – 22	20	Alphanumeric string. Can contain spaces. Must correspond to a station name in the Station file.
Measurement ID	143 – 152	10	Integer. Represents a unique database identifier.
Cluster ID	153 – 162	10	Integer. Represents a unique database identifier.

The following sections describe the additional fields and corresponding file formatting for the various measurement types.

Horizontal angles (A)

Table B.4 shows the information and file formatting that applies to horizontal angle measurements, and Figure B.3 shows some formatted examples.

Table B.4: Column locations and field widths for horizontal angles

Field	Columns	Width	Comments
Second station name	23 – 42	20	As per first station name in Table B.3.
Third station name	43 – 62	20	As per first station name in Table B.3.
Horizontal angle:			
Degrees	77 – 80	4	Integer (0 – 359).
Minutes	81 – 82	2	Integer (0 – 59).
Seconds	83 – 90	8	Decimal ($0 < s < 60$).
Standard deviation	91 – 99	9	Decimal value in units of seconds.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
A 102                 103                 104                                  47 32 31.716     0.010
A 104                 102                 103                                   8 34 49.259     0.010
```

Figure B.3: Example angle measurements

Geodetic (B) and astronomic (K) azimuths, zenith distances (V) and vertical angles (Z)

Table B.5 shows the information and file formatting that applies to geodetic azimuths, astronomic azimuths, zenith distances and vertical angles. Note that instrument and target heights are not required for geodetic or astronomic azimuths. Figure B.4 shows some formatted examples.

Table B.5: Column locations and field widths for geodetic azimuths, astronomic azimuths, zenith distances and vertical angles.

Field	**Columns**	**Width**	**Comments**
Second station name	23 – 42	20	As per first station name in Table B.3.
Azimuth / zenith distance / angle:			
Degrees	77 – 80	4	Integer (0 – 359 for azimuths, 0 – 179 for zenith distances and -89 – 89 for vertical angles).
Minutes	81 – 82	2	Integer (0 – 59).
Seconds	83 – 90	8	Decimal ($0 < s < 60$).
Standard deviation	91 – 99	9	Decimal value in units of seconds.
Instrument height	100 – 106	7	Compulsory for zenith distances and vertical angles only.
Target height	107 – 113	7	

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
B 101               202                                                137 42 08.500     0.010
K 301               502                                                 17 39 24.920     0.010
V 503               304                                                 89 32 59.147     0.010  1.591  1.438
Z 309               410                                                 -0 20 10.331     0.010  1.623  1.345
Z 503               307                                                  1 01 47.654     0.010  1.623  1.345
```

Figure B.4: Example geodetic azimuths, astronomic azimuths, zenith distances and vertical angles

Direction sets (D)

Information for direction sets is contained on at least two lines. The first record provides information relating to the reference direction and the number of directions in the set, and the remaining records provide information about each direction. Each direction must be provided in a clockwise fashion from the reference direction.

Table B.6 shows the information and file formatting that applies to direction sets, and Figure B.5 shows some formatted examples.

Table B.6: Column locations and field widths for direction sets.

Field	Columns	Width	Comments
Reference station name	23 – 42	20	As per first station name in Table B.3.
Reference direction:			
Number of directions	43 – 62	20	The number of directions that follow the reference direction.
Degrees	77 – 80	4	Integer (0 – 359).
Minutes	81 – 82	2	Integer (0 – 59).
Seconds	83 – 90	8	Decimal ($0 < s < 60$).
Standard deviation	91 – 99	9	Decimal value in units of seconds.
Directions (one per line):			
Station name	43 – 62	20	As per first station name in Table B.3.
Degrees	77 – 80	4	Integer (0 – 359).
Minutes	81 – 82	2	Integer (0 – 59).
Seconds	83 – 90	8	Decimal ($0 < s < 60$).
Standard deviation	91 – 99	9	Decimal value in units of seconds.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
D 322600020          322600010           2                                    0 00 00.000    1.769
D                                         213000050                          161 16 41.400    1.769
D                                         322600030                          342 09 25.400    1.769
D 322600010          305700010           4                                    0 00 00.000    1.769
D                                         213000040                           14 09 21.100    1.769
D                                         322600050                           76 21 16.600    1.769
D                                         409600180                          152 20 05.300    1.769
D                                         265600050                          170 48 53.400    1.769
```

Figure B.5: Example direction sets

Ellipsoid chords (C), ellipsoid arcs (E), Mean Sea Level arcs (M), slope distances (S) and height differences (L)

For ellipsoid chords, ellipsoid arcs, Mean Sea Level arcs, slope distances and height differences, the information and file formatting shown in Table B.7 applies. Note that instrument and target heights are only required for slope distances. Figure B.6 shows some formatted examples.

Table B.7: Column locations and field widths for ellipsoid chords, ellipsoid arcs, Mean Sea Level arcs, slope distances and height differences.

Field	Columns	Width	Comments
Second station name	23 – 42	20	As per first station name in Table B.3.
Distance / height difference	63 – 76	14	Decimal value in metres.
Standard deviation	91 – 99	9	Decimal value in metres.
Instrument height	100 – 106	7	Compulsory only for slope distances.
Target height	107 – 113	7	Compulsory only for slope distances.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
C 501                309                                         5476.1616                 0.031
E 503                307                                         8807.1030                 0.054
M 4000               13                                            53.9380                 0.005
S 4000               2012                                          76.9140                 0.005 1.6060 1.3370
L 108                1034                                          -0.2220                 0.010
```

Figure B.6: Example ellipsoid chords, ellipsoid arcs, Mean Sea Level arcs, slope distances and height differences

Single GNSS baselines (G), GNSS baseline clusters (X) and GNSS point clusters (Y)

Information for single GNSS baselines, GNSS baseline clusters, and GNSS point clusters is contained in multiple lines. Depending on the GNSS measurement, certain fields are compulsory whilst others are ignored. The first record for GNSS measurements is a general header record containing information about stations, variance matrix scaling, reference frame and epoch. This record varies depending on whether GNSS baselines or GNSS points are supplied. Table B.8 shows the required information and file formatting for the header record of a GNSS baseline (single or cluster) measurement.

Table B.8: Column locations and field widths for a GNSS baseline (single or cluster) header record.

Field	**Columns**	**Width**	**Comments**
Second station name	23 – 42	20	As per first station name in Table B.3.
Cluster count	43 – 62	20	Not required for single GNSS baselines.
Variance matrix scaling:			
V–scale	63 – 72	10	Whole of matrix scalar.
P–scale	73 – 82	10	North–south scalar.
L–scale	83 – 92	10	East–west scalar.
H–scale	93 – 102	10	Vertical scalar.
Reference frame	103 – 122	20	Abbreviated name (c.f. Table 4.1), e.g. `ITRF2008`
Epoch	123 – 142	20	Date in the form '`dd.mm.yyyy`', e.g. `01.03.2010`

GNSS point clusters have a slightly different header record. The only differences are the omission of the second station name and the inclusion of a coordinate type, whereby GNSS point clusters may be supplied in `XYZ` or `LLH` format as described in §B.1.3. GNSS measurements in `UTM` format are not supported. Table B.9 shows the different header information and file formatting for GNSS point cluster measurements.

Table B.9: Column locations and field widths for a GNSS point cluster header record.

Field	Columns	Width	Comments
Coordinate type	23 – 42	20	`XYZ` or `LLH`. `UTM` is not supported.
Cluster count	43 – 62	20	Integer.

...

Variance matrix scaling, reference frame and epoch as per Table B.8

...

For all GNSS measurements, the next three lines provide the three GNSS measurement components and the lower/upper–triangular elements of the accompanying variance matrix $\mathbf{V}_m$. For GNSS baselines, this information must be supplied in metres. For GNSS points, the coordinate format of this information must correspond to the coordinate type specified in the header record (c.f. Table B.9). Hence, the measurement and variance components can be provided in metres (corresponding to `XYZ`) or in latitude, longitude and height (corresponding to `XYZ`). Latitude and longitude values must be expressed in degrees, minutes and seconds using HP notation (e.g. dd.mmssssss), and variance components for latitude and longitude must be in radians. Table B.10 shows the required information and file formatting for the GNSS measurement and variance components. Figure B.7 shows an example GNSS baseline (G) measurement. In this example, note that the reference frame and epoch have not been provided. In this case DynAdjust will adopt the reference frame and epoch supplied in the header line (c.f. Table B.1).

Table B.10: Column locations and field widths for GNSS baseline and point measurements.

Field	Columns	Width	Comments
x–component:			
Δx or x	63 – 82	20	For `XYZ`, measurement and variance in metres. For `LLH`, measurement in dd.mmssss and variance in units of radians.
$\mathbf{V}_{m_{11}}$ $(\sigma_x\sigma_x)$	83 – 102	20	
y–component:			
Δy or y	63 – 82	20	
$\mathbf{V}_{m_{21}}$ $(\sigma_y\sigma_x)$	83 – 102	20	
$\mathbf{V}_{m_{22}}$ $(\sigma_y\sigma_y)$	103 – 122	20	
z–component:			
Δz or z	63 – 82	20	Measurement variances in metres for both `XYZ` and `LLH`.
$\mathbf{V}_{m_{31}}$ $(\sigma_z\sigma_x)$	83 – 102	20	
$\mathbf{V}_{m_{32}}$ $(\sigma_z\sigma_y)$	103 – 122	20	
$\mathbf{V}_{m_{33}}$ $(\sigma_z\sigma_z)$	123 – 142	20	

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
G 202                 304                                          1.000     1.000     1.000     1.000
                                                                       -53817.2076          4.0220e-05
                                                                       -88430.4791         -1.3690e-05          1.4870e-05
                                                                       -50031.4264          3.9750e-05         -2.0350e-05          6.8030e-05
```

Figure B.7: Example GNSS baseline measurement

For GNSS baseline and point clusters, the measurement information shown in Table B.10 is repeated for the total number n of measurement vectors $(m_1, m_2, \ldots, m_n)$ in the cluster. Immediately after each vector, the row–wise covariance blocks $(\mathbf{C}_{m_1m_2}, \mathbf{C}_{m_1m_3}, \ldots, \mathbf{C}_{m_1m_n})$ between the respective GNSS measurement variance blocks $(\mathbf{V}_{m_1}, \mathbf{V}_{m_2}, \ldots, \mathbf{V}_{m_n})$ must be provided such that a rigorous upper variance matrix $\mathbf{V}_{1\ldots n}$ can be formed for the whole cluster, as per equation B.1.

$$\mathbf{V}_{1\ldots n} = \left[\begin{array}{ccccccccc}
\sigma_x\sigma_x & \sigma_x\sigma_y & \sigma_x\sigma_z & & & & & & \\
 & \sigma_y\sigma_y & \sigma_y\sigma_z & & \mathbf{C}_{m_1m_2} & & \cdots & \mathbf{C}_{m_1m_n} & \\
 & & \sigma_z\sigma_z & & & & & & \\
 & & & \sigma_x\sigma_x & \sigma_x\sigma_y & \sigma_x\sigma_z & & & \\
 & & & & \sigma_y\sigma_y & \sigma_y\sigma_z & \cdots & \mathbf{C}_{m_2m_n} & \\
 & & & & & \sigma_z\sigma_z & & & \\
 & & & & & & \ddots & \vdots & \\
 & & & & & & \sigma_x\sigma_x & \sigma_x\sigma_y & \sigma_x\sigma_z \\
 & & & & & & & \sigma_y\sigma_y & \sigma_y\sigma_z \\
 & & & & & & & & \sigma_z\sigma_z
\end{array}\right] \tag{B.1}$$

The information for a single covariance block $\mathbf{C}$ contains nine elements which are provided over three records. Table B.11 shows the required information and file formatting for the nine cluster covariance components.

Table B.11: Column locations and field widths for a GNSS cluster covariance block.

Field	Columns	Width	Comments
$\mathbf{C}_{11}$	83 – 102	20	First record
$\mathbf{C}_{12}$	103 – 122	20	
$\mathbf{C}_{13}$	123 – 142	20	
$\mathbf{C}_{21}$	83 – 102	20	Second record
$\mathbf{C}_{22}$	103 – 122	20	
$\mathbf{C}_{23}$	123 – 142	20	
$\mathbf{C}_{31}$	83 – 102	20	Third record
$\mathbf{C}_{32}$	103 – 122	20	
$\mathbf{C}_{33}$	123 – 142	20	

These blocks are repeated for as many covariance blocks required to complete each row of the upper–triangular matrix $\mathbf{V}_{1\ldots n}$. As with the variance matrix information supplied for the individual GNSS measurements, the covariance units must correspond to the coordinate type specified in the cluster header record.

As an example, Figure B.8 shows a GNSS baseline cluster comprised of three baselines. Note that there are two covariance blocks associated with the first GNSS baseline (`HOI_` – `MEI_`). These blocks represent the row–wise covariances between the first baseline and the second and third baselines (`HOI_` – `MSI_` and `HOI_` – `SBI_`) respectively. Similarly, the covariance block associated with the second GNSS baseline corresponds to the covariance between the second and third baselines. Figure B.9 shows an example of two GNSS point clusters — the first is comprised of three correlated points in `XYZ` format and the second has two points in `LLH` format.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
X HOI_               MEI_                 3                         1.000      1.000      1.000      1.000            itrf2005          17.09.2008
                                                                 -125691.6314         2.034048e-06
                                                                 -145519.2865        -6.968547e-07        1.474554e-06
                                                                   73663.5870         3.962077e-07       -2.984549e-07        1.537337e-07
                                                                                      5.871788e-07       -2.839605e-07        1.247021e-07
                                                                                     -2.808181e-07        4.298451e-07       -1.016602e-07
                                                                                      1.242805e-07       -1.025933e-07        5.222703e-08
                                                                                      7.646870e-07       -2.856280e-07        1.465497e-07
                                                                                     -2.926789e-07        5.238148e-07       -1.193196e-07
                                                                                      1.550035e-07       -1.182915e-07        6.196860e-08
X HOI_               MSI_
                                                                  -86797.3963         1.876158e-06
                                                                  -87459.1959        -7.301331e-07        1.217260e-06
                                                                   91306.3628         3.423311e-07       -2.596589e-07        1.278370e-07
                                                                                      5.709494e-07       -2.902187e-07        1.189632e-07
                                                                                     -2.869810e-07        4.243482e-07       -1.022121e-07
                                                                                      1.239490e-07       -1.010039e-07        5.159684e-08
X HOI_               SBI_
                                                                  -39640.6285         2.385609e-06
                                                                  -13533.0128        -6.219224e-07        1.637419e-06
                                                                  131721.2044         4.633211e-07       -2.878135e-07        1.732844e-07
```

Figure B.8: Example GNSS baseline clusters.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
Y YIEL               XYZ                  3                          1.00       1.00       1.00       1.00               IGb08          01.06.2013
                                                                -4247834.5782         1.288543e-05
                                                                 2948383.3671        -5.037133e-06        1.040270e-05
                                                                -3721788.4927         3.642770e-06       -2.406477e-06        9.898976e-06
                                                                                      1.065777e-05       -3.701601e-06        2.041076e-06
                                                                                     -3.579744e-06        9.246168e-06       -1.254975e-06
                                                                                      2.174192e-06       -1.449455e-06        8.465730e-06
                                                                                      8.159775e-06       -2.180570e-06        1.558934e-06
                                                                                     -1.892996e-06        7.311562e-06       -1.455025e-06
                                                                                      1.484222e-06       -4.435524e-07        5.804253e-06
Y YRRM
                                                                -4172492.9321         1.269096e-05
                                                                 2743402.2911        -4.907058e-06        1.027008e-05
                                                                -3954709.8313         3.570733e-06       -2.329364e-06        9.936777e-06
                                                                                      8.167171e-06       -2.118313e-06        1.598938e-06
                                                                                     -1.942259e-06        7.291434e-06       -1.513716e-06
                                                                                      1.394181e-06       -3.230191e-07        5.780071e-06
Y YSSK
                                                                -3465321.0856         9.840133e-06
                                                                 2638269.2861        -2.482731e-06        8.405404e-06
                                                                 4644085.3490         2.303389e-06       -1.473060e-06        7.531223e-06
Y 1004               LLH                            2             10.000      1.000      1.000      1.500               GDA94
                                                                 -37.480000000            9.402e-09
                                                                 144.573400000            5.876e-10           9.402e-09
                                                                        42.195            5.876e-10           5.876e-10           2.500e-01
                                                                                          5.876e-12           5.876e-12           5.876e-12
                                                                                          5.876e-12           5.876e-12           5.876e-12
                                                                                          5.876e-12           5.876e-12           5.876e-12
Y 9004
                                                                 -37.474800000            9.402e-09
                                                                 144.573600000            5.876e-10           9.402e-09
                                                                        44.324            5.876e-10           5.876e-10           2.500e-01
```

Figure B.9: Example GNSS point clusters.

Orthometric height (H) and ellipsoid height (R)

For orthometric height and ellipsoid height measurements, the information and file formatting shown in Table B.12 applies. Figure B.10 shows some formatted examples.

Table B.12: Column locations and field widths for orthometric and ellipsoid height measurements.

Field	Columns	Width	Comments
Height	63 – 76	14	Decimal value in metres.
Standard deviation	91 – 99	9	Decimal value in metres.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

H 387800830                                                         283.8600                  0.090
R 701219                                                             130.161                 0.0001
```

Figure B.10: Example orthometric height and ellipsoid height measurements.

Geodetic latitudes (P), geodetic longitudes (Q), astronomic latitudes (I) and astronomic longitudes (J)

For geodetic latitudes, geodetic longitudes, astronomic latitudes and astronomic longitudes, the information and file formatting shown in Table B.13 applies. Figure B.11 shows some formatted examples.

Table B.13: Column locations and field widths for geodetic latitudes and longitudes, and astronomic latitudes and longitudes.

Field	Columns	Width	Comments
Latitude / longitude	63 – 76	14	Decimal value in ddd.mmssss.
Standard deviation	91 – 99	9	Decimal value in units of seconds.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

P 105                                                         -24.2958384014              0.0001
Q 105                                                         136.3000088022              0.0001
I 106                                                         -24.2957481018              0.02
J 106                                                         137.0003182747              0.02
```

Figure B.11: Example geodetic latitude and longitude, and astronomic latitude and longitude measurements.

B.1.5 Geoid information

Following the single header line, all geoid information that can be introduced to an adjustment for a station is formatted on one line. The relevant fields and corresponding file positions for geoid information are shown in Table B.14.

Table B.14: Geoid information column locations and field widths

Field	Columns	Width	Comments
Station name	1 – 20	20	Alphanumeric string. Can contain spaces.
N–value	41 – 50	10	Geoid–ellipsoid separation (N). Decimal value in units of metres.
N–S deflection	60–69	10	Deflection in prime meridian (ξ). Decimal value in units of seconds.
E–W deflection	70 – 79	10	Deflection in prime vertical (η). Decimal value in units of seconds.

Figure B.12 shows some example geoid records in DNA format. As with station and measurement information, geoid information can be positioned anywhere within the respective fields without the need for right or left justification.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789
!#=DNA 3.01 GEO    16.10.2012        GDA94                    12
* Created 06 November 2016.
* Geoid information for level section 1596
1596-3H                                      3.177            -9.267    -0.816
1596-4                                       3.205            -9.354    -1.035
1596-7                                       2.897            -8.991    -1.351
1596-8                                       2.810            -8.669    -1.672
1596-9                                       2.728            -8.334    -1.316
1596-10                                      2.573            -8.112    -1.119
1596-11H                                     2.501            -7.819    -0.767
1596-12                                      2.430            -7.462    -0.606
1596-13                                      2.382            -7.189    -0.423
1596-14                                      2.289            -6.919    -0.269
1596-15H                                     2.232            -7.035    -0.365
1596-16H                                     2.184            -7.154    -0.457
```

Figure B.12: Example geoid information records

Note that whilst the header record can simply be in the form of a comment, it is helpful to record metadata pertaining to the records, including the date, geodetic datum (and hence, ellipsoid) to which the geoid–ellipsoid separations relate and the number of records. Any number of comments may be dispersed throughout the file.

B.2 Dynamic Network Adjustment Project (DNAPROJ) format

The DynAdjust project file is designed to capture default and user–specified program options and arguments. The first line in the project file is reserved for comments and commences with the hash character (#). The remainder of the file consists of sections which correspond directly to the various DynAdjust programs. The formatting used to capture the options for each program is given in Table B.15.

Table B.15: DynAdjust program options formatting.

Field	Columns	Width	Comments
Element	1 – 35	35	Program option
Value	36 –	Unlimited	Default or user–specified value

Using this formatting, each section contains a header record commencing with a hash character, a line of dashes, then one or more options and their corresponding values. Each section is separated by a blank line. The first section is a general section and contains the list of options which are common to all programs. The following sections contain the default and user–specified options corresponding to **import**, **reftran**, **geoid**, **segment**, **adjust** and the options for the output of information. Figure B.13 shows a formatted example for the `skye` project (see §1.4), including the comment line and the list of general options.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
# skye project file. Created by: import, DynAdjust file exchange software.

#general (35)                      VALUE
--------------------------------------------------------------------------------
network-name                       skye
input-folder                       C:\Data\proj
output-folder                      C:\Data\proj
verbose-level                      0
quiet                              no
project-file                       C:\Data\proj\skye.dnaproj
dynadjust-log-file                   C:\Data\proj\dynadjust.log
```

Figure B.13: Example general options.

Note that the general section commences with `#general`. As a project file is loaded, DynAdjust will look for these section headings before attempting to load the options and values which follow. In addition to `#general`, the valid section headings are `#import`, `#reftran`, `#geoid`, `#segment`, `#adjust`, `#output` and `#display`. Using the `skye` project (see §1.4), Figures B.14, B.15, B.16, B.17 and B.18 list the options and example arguments for **import**, **reftran**, **geoid**, **segment** and **adjust**. B.19 lists an example of the options and arguments for the output of information from DynAdjust.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
#import (35)                       VALUE
--------------------------------------------------------------------------------
stn-msr-file                       skye.stn
stn-msr-file                       skye.msr
geo-file
bounding-box
get-msrs-transcending-box          no
include-stns-assoc-msrs            no
exclude-stns-assoc-msrs            no
split-gnss-cluster-msrs            no
import-block-stn-msr               no
seg-file
prefer-single-x-as-g               no
import-msr-types
exclude-msr-types
stn-renaming-file
search-nearby-stn                  no
nearby-stn-buffer                  0.3
search-similar-msr                 no
ignore-similar-msr                 no
remove-ignored-msr                 no
flag-unused-stations               no
test-integrity                     no
v-scale                            1
p-scale                            1
l-scale                            1
h-scale                            1
baseline-scalar-file
export-xml-files                   no
single-xml-file                    no
export-dna-files                   no
export-asl-file                    no
export-aml-file                    no
export-map-file                    no
simulate-msr-file                  no
```

Figure B.14: Example **import** options.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
#reftran (35)                      VALUE
--------------------------------------------------------------------------------
reference-frame                    GDA94
epoch
```

Figure B.15: Example **reftran** options.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
#geoid (35)                        VALUE
--------------------------------------------------------------------------------
ntv2-file                          C:\Data\geoid\ausgeoid09.gsb
interpolation-method               1
decimal-degrees                    0
direction                          0
convert-stn-hts                    yes
export-dna-geo-file                no
```

Figure B.16: Example **geoid** options.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
#segment (35)                      VALUE
-------------------------------------------------------------------------------
net-file
seg-file
min-inner-stns                     5
max-block-stns                     65
contiguous-blocks                  yes
starting-stns
```

Figure B.17: Example **segment** options.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
#adjust (35)                       VALUE
--------------------------------------------------------------------------------
seg-file
comments
adjustment-mode                    simultaneous-adjustment
multi-thread                       no
staged-adjustment                  no
conf-interval                      95
iteration-threshold                0.0005
max-iterations                     10
constraints                        302513640,CCC
free-stn-sd                        10.000
fixed-stn-sd                       1.0000e-06
inversion-method                   0
scale-normals-to-unity             no
create-stage-files                 no
purge-stage-files                  no
```

Figure B.18: Example **adjust** options.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
#output (35)                       VALUE
--------------------------------------------------------------------------------
output-msr-to-stn                  no
output-iter-adj-stn                no
output-iter-adj-stat               no
output-iter-adj-msr                no
output-iter-cmp-msr                no
output-adj-msr                     yes
output-adj-gnss-units              0
output-tstat-adj-msr               no
sort-adj-msr-field                 0
output-database-ids                no
output-msr-blocks                  no
sort-stn-orig-order                no
stn-coord-types                    PLHhXYZ
stn-corrections                    no
precision-stn-linear               4
precision-stn-angular              5
precision-msr-linear               4
precision-msr-angular              4
angular-msr-type                   0
dms-msr-format                     0
output-pos-uncertainty             yes
output-all-covariances             no
output-apu-vcv-units               0
output-corrections-file            yes
hz-corr-threshold                  0.000
vt-corr-threshold                  0.000
export-xml-stn-file                no
export-dna-stn-file                no
export-sinex-file                  no
```

Figure B.19: Example output options.

B.3 DynAdjust Markup Language (DynaML) format

The DynaML format provides for the exchange of stations and measurements in eXtensible Markup Language (XML) using the DynAdjust 2.0 XML (DynaML) schema. The scope of information includes the station and measurement information handled by the DNA format (see §B.1). The DynaML schema definition permits this information to be stored in two files, corresponding to station and measurement files, or a single stations and measurements file. All DynaML files have an `xml` file extension. For convenience, DynAdjust names station files as `*stn.xml`, and measurement files as `*msr.xml`. DynaML files must be encoded using well–formed XML, and conform to the XML schema definition.

As required, these files can contain one or more comment lines throughout the file. Comment lines must conform with conventional XML encoding as follows.

```
<!-- Comments can appear on one line. -->
<!-- Alternatively, comments can be
     spread over many lines.
-->
```

DynaML files must contain a root element named `DnaXmlFormat`. The DynaML schema definition for `DnaXmlFormat` is given in Figure B.20.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!--W3C Schema definition for DynaML -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="DnaXmlFormat">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="DnaStation"/>
        <xs:element ref="DnaMeasurement"/>
      </xs:choice>
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Measurement File"/>
            <xs:enumeration value="Station File"/>
            <xs:enumeration value="Combined File"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  ...
  ...
</xs:schema>
```

Figure B.20: `DnaXMLFormat` schema definition

As shown above, the root element can be one of three types — `"Measurement File"`, `"Station File"` and `"Combined File"`. These types inform the parser whether the elements contained in the file station information, measurement information or both. Accordingly, the root element can contain an unlimited number of `DnaStation` elements, `DnaMeasurement` elements or both.

B.3.1 Station information

All station information is recorded in the `DnaStation` element. This element contains a sub–element `StationCoord` which is a `xs:complexType` containing the coordinates for the station. The DynaML schema definition is given in Figure B.21.

```
<xs:element name="DnaStation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Name"/>
      <xs:element ref="Constraints"/>
      <xs:element ref="Type"/>
      <xs:element ref="StationCoord"/>
      <xs:element ref="Description"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(a) DnaStation element

```
<xs:element name="StationCoord">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Name" minOccurs="0"/>
      <xs:element ref="XAxis"/>
      <xs:element ref="YAxis"/>
      <xs:element ref="Height"/>
      <xs:element ref="HemisphereZone"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(b) StationCoord element

Figure B.21: DnaStation schema definition

With respect to DnaStation and StationCoord, elements Name, Constraints, Type, Description, XAxis, YAxis, Height and HemisphereZone are of type xs:string. All elements correspond to the station information fields shown in Table B.2.

By way of example, Figure B.22 shows a station file encoded in XML according to the DynaML schema definition. In this example, the coordinates for HOSP are provided in terms of type LLH and 1002-26 in terms of UTM.

```
<?xml version="1.0"?>
<DnaXmlFormat type="Station File"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="DynaML.xsd">
<!-- Station file. -->
 <DnaStation>
    <Name>HOSP</Name>
    <Constraints>FFF</Constraints>
    <Type>LLH</Type>
    <StationCoord>
      <XAxis>-37.4801765569</XAxis>
      <YAxis>144.5717295509</YAxis>
      <Height>83.9</Height>
    </StationCoord>
    <Description>HOSP</Description>
  </DnaStation>
<DnaStation>
    <Name>1002-26</Name>
    <Constraints>CCF</Constraints>
    <Type>UTM</Type>
    <StationCoord>
      <XAxis>372772.813</XAxis>
      <YAxis>5931491.145</YAxis>
      <Height>175.668</Height>
      <HemisphereZone>S55</HemisphereZone>
    </StationCoord>
    <Description>BM 26</Description>
  </DnaStation>
</DnaXmlFormat>
```

Figure B.22: Sample station file encoded in DynaML format

In relation to the expected format of coordinates corresponding to the value in the Type element, §B.1.3 can be used as a guide. However, note that latitude and longitude values must be expressed in degrees, minutes and seconds using HP notation (e.g. dd.mmssssss). Since there is no restriction on column formatting, values can be provided to any level of precision.

B.3.2 Measurement information

All measurement information is recorded in the `DnaMeasurement` element. The DynaML schema definition is given in Figure B.23.

```
<xs:element name="DnaMeasurement">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="ClusterID" minOccurs="0"/>
      <xs:element ref="Clusterpoint"/>
      <xs:element ref="Coords"/>
      <xs:element ref="Directions"/>
      <xs:element ref="First"/>
      <xs:element ref="Epoch" minOccurs="0"/>
      <xs:element ref="GPSBaseline"/>
      <xs:element ref="Hscale" minOccurs="0"/>
      <xs:element ref="Ignore" minOccurs="0"/>
      <xs:element ref="InstHeight"/>
      <xs:element ref="Lscale" minOccurs="0"/>
      <xs:element ref="MeasurementID" minOccurs="0"/>
      <xs:element ref="Pscale" minOccurs="0"/>
      <xs:element ref="ReferenceFrame" minOccurs="0"/>
      <xs:element ref="Second"/>
      <xs:element ref="Source" minOccurs="0"/>
      <xs:element ref="StdDev"/>
      <xs:element ref="TargHeight"/>
      <xs:element ref="Third"/>
      <xs:element ref="Total"/>
      <xs:element ref="Type"/>
      <xs:element ref="Value"/>
      <xs:element ref="Vscale" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Figure B.23: `DnaMeasurement` schema definition

Elements `Type`, `Ignore`, `First`, `Second`, `Third`, `Value`, `StdDev`, `InstHeight`, `TargHeight`, `Total`, `Vscale`, `Hscale`, `Lscale`, `Pscale`, `Coords`, `MeasurementID`, `ClusterID` and `Source` are a string data type (`xs:string`). Not all elements are mandatory for each measurement. The mandatory elements and required format for the respective measurement types (c.f. `Type`) follow the definitions contained in the DNA format specification (see §B.1.4). To flag a measurement as ignored so that DynAdjust will exclude the measurement from all processing, provide an asterisk (*) for the `Ignore` element's value:

```
<DnaMeasurement>
  ...
  <Ignore>*</Ignore>
  ...
</DnaMeasurement>
```

Elements `Clusterpoint` and `GPSBaseline` are of type `xs:complexType` and are given in Figure B.24. `Directions` is also of type `xs:complexType` and is given in Figure B.25.

```
<xs:element name="Clusterpoint">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="X"/>
      <xs:element ref="Y"/>
      <xs:element ref="Z"/>
      <xs:element ref="MeasurementID"
        minOccurs="0"/>
      <xs:element ref="SigmaXX"/>
      <xs:element ref="SigmaXY"/>
      <xs:element ref="SigmaXZ"/>
      <xs:element ref="SigmaYY"/>
      <xs:element ref="SigmaYZ"/>
      <xs:element ref="SigmaZZ"/>
      <xs:element ref="PointCovariance"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(a) `Clusterpoint` element

```
<xs:element name="GPSBaseline">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="X"/>
      <xs:element ref="Y"/>
      <xs:element ref="Z"/>
      <xs:element ref="MeasurementID"
        minOccurs="0"/>
      <xs:element ref="SigmaXX"/>
      <xs:element ref="SigmaXY"/>
      <xs:element ref="SigmaXZ"/>
      <xs:element ref="SigmaYY"/>
      <xs:element ref="SigmaYZ"/>
      <xs:element ref="SigmaZZ"/>
      <xs:element ref="GPSCovariance"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(b) `GPSBaseline` element

Figure B.24: `Clusterpoint` and `GPSBaseline` schema definition

Elements `GPSCovariance` (for `GPSBaseline`) and `PointCovariance` (for `Clusterpoint`) are complex element types (`xs:complexType`) and relate to the covariance information between GNSS baseline and GNSS point variance matrices. The schema definition for `GPSCovariance` is given in Figure B.25. The element `PointCovariance` contains an identical set of elements to those in `GPSCovariance` and as such is not shown here.

```
<xs:element name="Directions">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Ignore"/>
      <xs:element ref="Target"/>
      <xs:element ref="Value"/>
      <xs:element ref="StdDev"/>
      <xs:element ref="MeasurementID"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(a) `Directions` element

```
<xs:element name="GPSCovariance">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="m11"/>
      <xs:element ref="m12"/>
      <xs:element ref="m13"/>
      <xs:element ref="m21"/>
      <xs:element ref="m22"/>
      <xs:element ref="m23"/>
      <xs:element ref="m31"/>
      <xs:element ref="m32"/>
      <xs:element ref="m33"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

(b) `GPSCovariance` element

Figure B.25: `Directions` and `GPSCovariance` schema definition

Elements `Target`, `X`, `Y`, `Z` , `SigmaXX` through to `SigmaZZ` and `m11` through to `m33` are of type `xs:string`.

Figure B.26 provides some sample measurements presented in §B.1.4 encoded in XML according to the DynaML schema definition. As with station information, §B.1.4 can be used as a guide for the expected format and precision of angular and linear measurement values. However, with angular measurements (such as latitude, angles, etc.), all measurement values must be expressed in degrees, minutes and seconds using HP notation (e.g. dd.mmssssss) with their precision in seconds.

```xml
<?xml version="1.0"?>
<DnaXmlFormat type="Measurement File"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="DynaML.xsd">
  <!-- Sample measurements file -->
  <!-- Type G GPS Baseline -->
  <DnaMeasurement>
    <Type>G</Type>
    <ReferenceFrame>ITRF2008</ReferenceFrame>
    <Epoch>01.01.2010</Epoch>
    <Vscale>1.000</Vscale>
    <Pscale>1.000</Pscale>
    <Lscale>1.000</Lscale>
    <Hscale>1.000</Hscale>
    <First>202</First>
    <Second>304</Second>
    <GPSBaseline>
      <X>-53817.2076</X>
      <Y>-88430.4791</Y>
      <Z>-50031.4264</Z>
      <SigmaXX>4.0220e-05</SigmaXX>
      <SigmaXY>-1.3690e-05</SigmaXY>
      <SigmaXZ>3.9750e-05</SigmaXZ>
      <SigmaYY>1.4870e-05</SigmaYY>
      <SigmaYZ>-2.0350e-05</SigmaYZ>
      <SigmaZZ>6.8030e-05</SigmaZZ>
    </GPSBaseline>
  </DnaMeasurement>
  <!-- Orthometric Height -->
  <DnaMeasurement>
    <Type>H</Type>
    <Ignore>*</Ignore>
    <First>202</First>
    <Value>43.086</Value>
    <StdDev>0.065</StdDev>
  </DnaMeasurement>
  <!-- Zenith Angle -->
  <!-- Instrument and target heights -->
  <DnaMeasurement>
    <Type>V</Type>
    <First>1013</First>
    <Second>1014</Second>
    <Value>90.243</Value>
    <StdDev>20</StdDev>
    <InstHeight>1.545</InstHeight>
    <TargHeight>0.125</TargHeight>
  </DnaMeasurement>
  <!-- Horizontal Angle -->
  <DnaMeasurement>
    <Type>A</Type>
    <First>2012</First>
    <Second>4000</Second>
    <Third>2013</Third>
    <Value>266.281</Value>
    <StdDev>20</StdDev>
  </DnaMeasurement>
  <!-- Geodetic Azimuth -->
  <DnaMeasurement>
    <Type>B</Type>
    <First>1046</First>
    <Second>4010</Second>
    <Value>91.2031</Value>
    <StdDev>20</StdDev>
  </DnaMeasurement>
  ...
```

```xml
  ...
  <!-- Slope Distance -->
  <!-- Instrument and target heights -->
  <DnaMeasurement>
    <Type>S</Type>
    <First>1037</First>
    <Second>4006</Second>
    <Value>44.06</Value>
    <StdDev>0.005</StdDev>
    <InstHeight>1.542</InstHeight>
    <TargHeight>1.559</TargHeight>
  </DnaMeasurement>
  <!-- Astronomic Azimuth -->
  <DnaMeasurement>
    <Type>K</Type>
    <First>1046</First>
    <Second>4010</Second>
    <Value>91.2001</Value>
    <StdDev>20</StdDev>
  </DnaMeasurement>
  <!-- Direction set -->
  <DnaMeasurement>
    <Type>D</Type>
    <First>365200180</First>
    <Second>TS5137</Second>
    <Value>52.025209</Value>
    <StdDev>0.707</StdDev>
    <Total>3</Total>
    <Directions>
      <Target>409700260</Target>
      <Value>238.545977</Value>
      <StdDev>0.707</StdDev>
    </Directions>
    <Directions>
      <Target>TS2761</Target>
      <Value>264.344190</Value>
      <StdDev>0.707</StdDev>
    </Directions>
    <Directions>
      <Target>TS3568</Target>
      <Value>305.285959</Value>
      <StdDev>0.707</StdDev>
    </Directions>
  </DnaMeasurement>
  <!-- Vertical Angle -->
  <!-- Instrument and target heights -->
  <DnaMeasurement>
    <Type>Z</Type>
    <First>4000</First>
    <Second>1050</Second>
    <Value>0.3711</Value>
    <StdDev>20</StdDev>
    <InstHeight>1.606</InstHeight>
    <TargHeight>1.715</TargHeight>
  </DnaMeasurement>
 <!-- Level Difference -->
  <DnaMeasurement>
    <Type>L</Type>
    <First>2217</First>
    <Second>2218</Second>
    <Value>0.018</Value>
    <StdDev>0.002</StdDev>
  </DnaMeasurement>
  ...
</DnaXmlFormat>
```

Figure B.26: Sample measurements encoded in DynaML format

B.4 GeodesyML format

GeodesyML is a comprehensive Geography Markup Language (GML) application schema defined by the Intergovernmental Committee on Surveying and Mapping (ICSM). GeodesyML provides both XML schema definition and XML format. The specification for GeodesyML can be found at http://geodesyml.org. The version of GeodesyML supported by DynAdjust is 0.1.1.

B.5 SINEX format

The Solution (software/technique) INdependent EXchange (SINEX) file format is maintained by the International Earth Rotation Service (IERS) and provides for the management of station coordinates, velocities and earth rotation parameters. The SINEX file format specification is beyond the scope of this user guide, however interested readers are referred to the following site for more information:

> http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html

The version of SINEX supported by DynAdjust is 2.02.

B.6 Geoid input text file format

To provide an efficient means for converting spot heights contained in both small and extremely large files from one height system to another, **geoid** supports Formatted Text files (e.g. `*.dat`, `*.prn`, `*.txt`) and Comma Separated Values files (`*.csv`). **geoid** expects all input coordinates in both file formats to be geographic coordinates in degrees, minutes and seconds or decimal degrees. In both cases, coordinates must be in HP Notation (i.e. dd.dddddddd or dd.mmssssss). In this form, the latitude and longitude fields should each contain only one numeric value. When working with formatted text files, the maximum number of significant digits the values can have is 15 significant figures[1]. For latitudes in the southern hemisphere and longitudes west of the zero meridian, the number of significant figures is further reduced by 1 to cater for the minus sign.

Formatted text files

Every line in a formatted text file must contain data fields in particular file positions (or columns). Certain fields may be omitted depending on what they are. Table B.16 lists the compulsory and non-compulsory fields in the required order. Figure B.27 shows an example formatted text file and illustrates the use of the non–compulsory fields. Column numbers shown for reference only.

Table B.16: Formatted text file fields

Field	Columns	Width	Compulsory?
Point ID	1 – 11	11	No
Latitude	12 –27	16	Yes
Longitude	28 – 43	16	Yes
Height	44 – 52	9	No

1. Since the maximum field width for both latitude and longitude is 16 characters, excluding the decimal point leaves a maximum of 15 characters.

```
12345678901234567890123456789012345678901234567890123456789
---------><-------------><-------------><-------><
Point (11) Latitude (16)   Longitude (16)  Hght (9)
MT HIGH       -27.498408428   153.001072611
              -27.498421786   150.001124192
              -29.086179181   151.966654878
              -29.073486997   151.805272886   4.23
62 / 54       -29.000294436   151.457723186
GBM16         -28.636707072   151.970252700
GBM34         -28.619868617   151.650131492
              -28.235994419   151.990397797  36.281
```

Figure B.27: Example formatted text file

Comma separated values files

Every line in a CSV file must contain data fields separated by commas. Non-compulsory fields may be empty, however a sufficient number of commas must be present to delineate the presence of compulsory fields. Table B.17 lists the compulsory and non-compulsory fields in the required order.

Table B.17: Comma separated values file fields

Field	Compulsory?
Point ID	No
Latitude	Yes
Longitude	Yes
Height	No

According to Table B.17, a minimum of two commas is sufficient to delineate Point ID, Latitude and Longitude. Figure B.28 shows an example CSV file. Note that a header line is not required.

```
MT HIGH,-10.498408428,153.001072611
,-20.498421786,140.001124192
,-40.086179181,121.966654878,
1596-4     ,   -37.593644101,   144.204321339
1596-5     ,   -37.593616320,   144.204318245
62 / 54,-40.000294436,141.457723186, 4.23
GBM16,-30.636707072,151.970252700,
GBM34,-20.619868617,141.650131492
4,-20.235994419,121.99039779,36.281
```

Figure B.28: Example comma separated values file

Appendix C

Output file format specification

C.1 Header block

For each of the output files that DynAdjust generates, a standard header block is printed to the file. The information printed in the header block will commence and end with a dashed line, and will contain basic to detailed information depending on the file type. As a minimum, the header block contains the file type, information about the version and build of the program that created the file, and the file's creation date and location. Following this, additional information relevant to the file type is printed. The following example shows the header block of a coordinate output file.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

```
--------------------------------------------------------------------------------
DYNADJUST COORDINATE OUTPUT FILE
Version:                           3.2.6, Release (64-bit)
Build:                             Oct 18 2016, 16:48:34 (MSVC++ 10.0)
File created:                      Tuesday, 18 October 2016, 4:53:59 PM
File name:                         c:\Data\uni_sqr.phased.xyz

Reference frame:                   GDA94
Epoch:                             01.01.1994
Geoid model:                       c:\Data\ausgeoid09.gsb
Stations printed in blocks:        No
--------------------------------------------------------------------------------
```

The width of the field names in the header block will always be 35 characters wide.

C.2 Import log file (IMP)

The import log (`.imp`) file is generated by **import** and contains information closely matching the information printed to the screen (c.f. Figure 3.1). In addition to the program version and log file metadata printed in the header block, the import log header will contain the command line arguments supplied to **import**, the file paths to the binary and ASCII files created by **import**, the input station and measurement files, and the options that have been supplied **import**. Apart from the default reference frame, only the options which have been modified on calling **import** (c.f. §3.3) will be printed to the header. The following example shows the information printed to the import log file upon calling **import** using the default options.

Any failures in the import of stations and measurements will be reported to this file and must be rectified before proceeding to other steps in the adjustment.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
--------------------------------------------------------------------------------
DYNADJUST IMPORT LOG FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Oct 19 2016, 07:30:53 (MSVC++ 10.0)
File created:                      Wednesday, 19 October 2016, 7:31:30 AM
File name:                         c:\Data\uni_sqr.imp

Command line arguments:            import -n uni_sqr uni_sqrstn.xml uni_sqrmsr.xml

Network name:                      uni_sqr
Input folder:                      c:\Data\
Output folder:                     c:\Data\
Stations file:                     c:\Data\uni_sqr.bst
Measurements file:                 c:\Data\uni_sqr.bms
Associated station file:           c:\Data\uni_sqr.asl
Associated measurement file:       c:\Data\uni_sqr.aml
Duplicate stations output file:    c:\Data\uni_sqr.dst
Similar measurement output file:   c:\Data\uni_sqr.dms
Input files:                       uni_sqrstn.xml
                                   uni_sqrmsr.xml
Default reference frame:           GDA94
--------------------------------------------------------------------------------

+ Parsing
  uni_sqrstn.xml...         Done. Loaded 149 stations in 0.035s
  uni_sqrmsr.xml...         Done. Loaded 1199 measurements in 0.04s
+ Reducing stations... Done.

+ File parsing summary:

  Read 149 stations:
 ------------------------------------------
  (CCC) 3D constrained:                  1
  (FFF) 3D free:                       145
  (CCF) 2D constrained, 1D free:         1
  (FFC) 2D free, 1D constrained:         1
  (CFF) custom 2D constraints:           1
 ------------------------------------------
  Total                                149

  Read 1199 measurements:
 ------------------------------------------
  (A) Horizontal angle:                251
  (B) Geodetic azimuth:                  1
  (C) Chord dist:                        0
  (D) Directions:                        0
  (E) Ellipsoid arc:                     0
  (G) GPS baseline:                    114
  (H) Orthometric height:                1
  (I) Astronomic latitude:               0
  (J) Astronomic longitude:              0
  (K) Astronomic azimuth:                1
  (L) Level difference:                 89
  (M) Mean sea level arc:                1
  (P) Geodetic latitude:                 0
  (Q) Geodetic longitude:                0
  (R) Ellipsoidal height:                0
  (S) Slope distance:                  428
  (V) Zenith angle:                    300
  (X) GPS baseline cluster:              0
  (Y) GPS point cluster:                12
  (Z) Vertical angle:                    1
 ------------------------------------------
  Total                               1199*
  * Includes 17 ignored measurements

+ Testing for duplicate stations... Done.
+ Sorting stations... Done.
+ Serialising station map... Done.
+ Mapping measurements to stations... Done. Mapped 1199 measurements to 149 stations.
+ Creating association lists... Done.
+ Serialising association lists... Done.
+ Serialising binary station file uni_sqr.bst... Done.
+ Serialising binary measurement file uni_sqr.bms... Done.

+ Total file handling process took 0.182s.
+ Binary station and measurement files are now ready for processing.
```

C.3 Measurement to station output file (M2S)

The measurement to station (`.m2s`) file is generated by **import** when the option `--output-msr-to-stn` has been provided. The following example shows the information printed to the measurement to station file (several records have been omitted). By default, the measurement to station table is sorted according to station. To sort this table according to the number of measurements connected to each station, pass to **import** the option `--sort-msr-to-stn-field` followed by 1.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------
DYNADJUST MEASUREMENT TO STATION OUTPUT FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Oct 19 2016, 07:30:17 (MSVC++ 10.0)
File created:                      Wednesday, 19 October 2016, 7:51:13 AM
File name:                         c:\Data\uni_sqr.m2s

Associated measurement file:       c:\Data\uni_sqr.aml
Stations file:                     c:\Data\uni_sqr.bst
Measurements file:                 c:\Data\uni_sqr.bms

No. stations:                      149
--------------------------------------------------------------------------------


MEASUREMENT TO STATIONS
------------------------------------------
```

Station	A	B	C	D	E	G	H	I	J	K	L	M	P	Q	R	S	V	X	Y	Z	Total
0102	2															4	4				10
1002	9										2					5	4				20
1003	10										1					2	3				16
1004	11					2										5	5		1		24
1005	6															4	4				14
1006	3															2	2				7
...																					...
1042	12						1									3	3		1		20
1044	10															4	3				17
1046	5	1								1						5	4				16
1047	5															7	5				17
1049	6															5	6				17
1050	6															7	6			1	20
1051	4															3	4				11
1052	3															6	4				13
108	5										2					1					8
13	5											1				4	3				13
...																					...
9003	20					2										10	7				39
9004	4					4										4	4		1		17
BARR	3					4					2					4	4				17
BOWL	3					3					2					8	8				24
CLUB						3															3
HOSP						7															7
NEWP	6										4					13	12				35
PELH	3					8					2					10	9				32
RMH-FP	12																				12
Totals	251	1				114	1			1	89	1				428	300		12	1	1199

```
WARNING: 1 station was found to have GNSS measurements and absolute terrestrial measurements:

Station             Measurement types            Count
------------------------------------------------------------
1042                AHSVY                        20
```

Table C.1 shows the formatting of the measurement to station table printed to the `.m2s` file. The table will be sorted row–wise by the station name, and column–wise by the measurement type (c.f. §7.2). In addition to the row–wise measurement count for each station, total counts for each measurement type are provided at the bottom of the table (column–wise), with a final count at the end of the table.

Table C.1: Measurement to station connections table

Heading	Columns	Width	Comments
`Station name`	1–20	20	Alphanumeric string. Can contain spaces. Corresponds to the station names in the Station file.
`A ... Z`	21–180	20 x 8	20 columns (8 characters wide) with header fields representing the supported measurement types.
`Total`	181–191	11	Integer. Represents the total number of measurements connected to the station in this row.

If a station has been observed by GNSS, and that station is also connected by terrestrial measurements of an absolute nature (e.g. latitude, longitude and/or height), conflicts between those measurements may arise. In some scenarios, such conflicts can cause bias in the estimation of coordinates and uncertainties. To assist with identifying the stations which may be the result of such conflicts, a warning summary will be printed at the end of the file. In the example shown above, a warning is produced for station `1042` because it is connected to a height measurement (`H`) and a GNSS point cluster (`Y`).

C.4 Reftran log file (RFT)

The reference frame transformation log (`.rft`) file is generated by **reftran** and contains information closely matching the information printed to the screen (c.f. Figure 4.8). In addition to the program version and log file metadata printed in the header block, the log file header will contain the command line arguments supplied to **reftran**, the file paths to the binary files modified by **reftran**, the target reference frame, the number of stations and measurements transformed, and the options that have been supplied **reftran** (c.f. §4.7.2). The example on the next page shows the information printed to the reference frame transformation log file upon calling **reftran** with options to:

- transform a 3 station GNSS cluster from ITRF2008@08.03.2007 to ITRF2014@19.11.2020;
- use a global plate motion model (PB2002);
- export the results to DNA format;
- print the calculated cartesian rotations from the Euler pole parameters (verbose level 2). Here, several records have been omitted;
- print the transformation steps (verbose level 2), and;
- print the reduced transformation parameters used in the calculations (verbose level 3).

If a global plate motion model has been loaded, and `--verbose-level` has been supplied with an argument of 2, a table of the imported Euler pole rotation parameters and the calculated cartesian elements will be printed to the `.rft` file. Table C.2 shows the formatting of the Euler pole rotation parameters table. The table will be sorted row–wise by the tectonic plate identifier.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

```
--------------------------------------------------------------------------------
DYNADJUST IMPORT LOG FILE

Version:                           1.0.2, Release (64-bit)
Build:                             Nov 13 2020, 17:28:09 (MSVC++ 1916)
File created:                      Thursday, 19 November 2020, 21:03:43
File name:                         c:\Data\uni_sqr.imp

Command line arguments:            reftran apr14177_sub -r itrf2014 -e today --export-dna --verb 3 --plate-model-option 1 -b ./tectonic-plates/PB2002_plates.dig -m ./tectonic-plates/PB2002_poles.dat

Network name:                      apr14177_sub
Input folder:                      c:\Data\
Output folder:                     c:\Data\
Stations file:                     c:\Data\apr14177_sub.bst
Measurements file:                 c:\Data\apr14177_sub.bms
Target reference frame:            ITRF2014
Target Epoch:                      Thursday, 19 November 2020
Plate boundaries file:             c:\Data\tectonic-plates\PB2002_plates.dig
Plate pole parameter file:         c:\Data\tectonic-plates\PB2002_poles.dat
DNA station file:                  c:\Data\apr14177_sub.ITRF2014.19.11.2020.stn
DNA measurement file:              c:\Data\apr14177_sub.ITRF2014.19.11.2020.msr
--------------------------------------------------------------------------------

+ Loading global tectonic plate boundaries and plate motion information...

Euler pole rotation parameters
-------------------------------------------

Plate     Pole Latitude    Pole Longitude   Euler Rot. Rate       X Rot. Rate       Y Rot. Rate       Z Rot. Rate     Reference
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
AF             59.1600          -73.1740          -73.1740          0.495216         -1.637548          2.865327     DeMets et al. [1994]
AM             57.6450          -83.7360          -83.7360          0.195684         -1.782754          2.830955     Heki et al. [1999]
AN             64.3150          -83.9840          -83.9840          0.142191         -1.349229          2.820907     DeMets et al. [1994]
AP             33.6390          -81.1770          -81.1770          0.421096         -2.712912          1.826733     Lamb [2000]
AR             59.6580          -33.1930          -33.1930          1.767772         -1.156489          3.608965     DeMets et al. [1994]
AS             74.2750          -87.2370          -87.2370          0.030557         -0.633158          2.251383     McClusky et al. [2000]
AT             56.2830            8.9320            8.9320          3.237516          0.508834          4.910885     McClusky et al. [2000]
AU             32.2447           38.2022           38.2022          1.503814          1.183478          1.207179     Geoscience Australia [2016]
...
YA             69.0670          -97.7180          -97.7180         -0.172439         -1.272375          3.356680     Heki et al. [1999]

+ Done.

+ Transforming stations and measurements...

Station coordinate transformations
-------------------------------------------

ID Station                    Frame       Epoch   Plate              X               Y               Z            dX            dY            dZ            Sc            rX            rY            rZ            dt
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
FR ALIC                   ITRF2008  08.03.2007   AU      -4052052.2343    4212836.0582   -2545105.2820
DD ALIC                   ITRF2014  08.03.2007   AU      -4052052.2363    4212836.0568   -2545105.2849       -0.0016       -0.0019       -0.0027    1.0453e-10    0.0000e+00    0.0000e+00    0.0000e+00       -2.8178
PM ALIC                   ITRF2014  19.11.2020   AU      -4052052.7742    4212835.9861   -2545104.5455        0.0000        0.0000        0.0000    0.0000e+00   -9.9895e-08   -7.8616e-08   -8.0190e-08      -13.7017
TO ALIC                   ITRF2014  19.11.2020   AU      -4052052.7742    4212835.9861   -2545104.5455
FR DAV1                   ITRF2008  08.03.2007   AN        486854.5590    2285099.2376   -5914955.6773
DD DAV1                   ITRF2014  08.03.2007   AN        486854.5574    2285099.2360   -5914955.6806       -0.0016       -0.0019       -0.0027    1.0453e-10    0.0000e+00    0.0000e+00    0.0000e+00       -2.8178
PM DAV1                   ITRF2014  19.11.2020   AN        486854.6593    2285099.3831   -5914955.6154        0.0000        0.0000        0.0000    0.0000e+00   -9.4454e-09    8.9626e-08   -1.8739e-07      -13.7017
TO DAV1                   ITRF2014  19.11.2020   AN        486854.6593    2285099.3831   -5914955.6154
FR DUND                   ITRF2008  08.03.2007   PA      -4388120.5794     726671.5295   -4556533.4219
DD DUND                   ITRF2014  08.03.2007   PA      -4388120.5814     726671.5277   -4556533.4250       -0.0016       -0.0019       -0.0027    1.0453e-10    0.0000e+00    0.0000e+00    0.0000e+00       -2.8178
PM DUND                   ITRF2014  19.11.2020   PA      -4388120.5814     726671.5277   -4556533.4250        0.0000        0.0000        0.0000    0.0000e+00    0.0000e+00    0.0000e+00    0.0000e+00      -13.7017
TO DUND                   ITRF2014  19.11.2020   PA      -4388120.5814     726671.5277   -4556533.4250

+ Done.

+ Transformed 3 stations.
+ Transformed 9 measurements.

+ Exporting stations and measurements to apr14177_sub.ITRF2014.19.11.2020.stn and apr14177_sub.ITRF2014.19.11.2020.msr... Done.
```

Table C.2: Euler pole rotation parameters

Heading	Columns	Width	Comments
`Plate`	1–5	5	Alphanumeric string. Tectonic plate identifier.
`Pole Latitude`	6–23	18	Euler pole latitude in decimal degrees.
`Pole Longitude`	24–41	18	Euler pole longitude in decimal degrees.
`Euler Rot. Rate`	42–59	18	Euler rotation rate in decimal degrees per million years
`X Rot. Rate`	60–77	18	X rotation rate ($\dot{r_x}$) in metres.
`Y Rot. Rate`	78–95	18	Y rotation rate ($\dot{r_y}$) in metres.
`Z Rot. Rate`	96–113	18	Z rotation rate ($\dot{r_z}$) in metres.
	114–118	5	Space.
`Reference`	119–	*	Author citation for the solution of Euler pole parameters.

If `--verbose-level` has been supplied with an argument of 2, a table of station coordinate transformations will be printed to the `.rft` file. Table C.3 shows the formatting of the station coordinate transformations table. The table will be sorted row–wise by the station name in the binary station file. If `--verbose-level` has been supplied with an argument of 3, additional columns tabulating the reduced transformation parameters will be printed.

Table C.3: Station coordinate transformations

Heading	Columns	Width	Comments
`ID`	1–3	3	Two–character identifier for the transformation type.
`Station`	4 – 23	20	Station name (corresponding to the input station file).
`Frame`	24–35	12	Reference frame name.
`Epoch`	36–47	12	Epoch in the form of `dd.mm.yyyy`.
	48–50	3	Space.
`Plate`	51–55	5	Alphanumeric string. Tectonic plate identifier.
`X`	56–71	16	Computed X coordinate.
`Y`	72–87	16	Computed Y coordinate.
`Z`	88–103	16	Computed Z coordinate.
`dX`	104–117	14	Calculated X translation (Δx). Optional output and is disabled by default. To enable, add the `--verbose-level` option with 3 to the list of **reftran** commands.
`dY`	118–131	14	Calculated Y translation (Δy). Optional output as above.
`dZ`	132–145	14	Calculated Z translation (Δz). Optional output as above.
`Sc`	146–159	14	Calculated scale (δ). Optional output as above.
`rX`	160–173	14	Calculated X rotation (r_x). Optional output as above.
`rY`	174–187	14	Calculated Y rotation (r_y). Optional output as above.
`rZ`	188–201	14	Calculated Z rotation (r_z). Optional output as above.
`dt`	202–215	14	Elapsed time ($t_2 - t_1$). Optional output as above.

The two–character identifier for the transformation type will be one of the following: `FR` (From), `TO` (To), `SS` (static to static), `DD` (dynamic to dynamic), `DS` (dynamic to static), `SD` (static to dynamic), `PM` (plate motion), and `JN` (joined parameters via ITRF2014, see §4.6.1).

C.5 Segmentation output file (SEG)

The segmentation output (`.seg`) file is generated by **segment** and contains block–wise information about a segmented network. Following the header block, there are two primary sections in the file. Firstly, there is a segmentation summary section, titled '`SEGMENTATION SUMMARY`', and secondly, there is a section for all the individual block data, titled '`INDIVIDUAL BLOCK DATA`'. The example on the following page shows the segmentation output file for the trivial GNSS network shown in Figure 6.2.

Table C.4 shows the formatting of the segmentation summary table. In the summary table, each row summarises the station and measurement counts for a block (sequentially numbered in the `Block` column). If the network contains isolated networks, the values in the `Network ID` column will increment if the `--contiguous-blocks` switch has been passed to **segment**.

The `INDIVIDUAL BLOCK DATA` section contains sub–sections for each block, entitled '`Block #`' where `#` is the respective block number. Following the sub–section heading is a re–iteration of the summary statistics for the subject block, including the junction station count, inner station count, measurement count and total station count. After this comes the block data. Table C.5 shows the formatting of the individual block data. Note that the data in this table is presented column–wise.

Table C.4: Segmentation summary table

Heading	Columns	Width	Comments
`Block`	2–14	12	Unique block number.
`Network ID`	14–27	14	Unique network number. By default, all non–contiguous networks will be incorporated within a single network unless the `--contiguous-blocks` is passed to **segment** with a value of 0, in which case the values in this column will increment.
`Junction stns`	28–43	16	The number of junction stations in the block.
`Inner stns`	44–59	16	The number of inner stations in the block.
`Measurements`	60–75	16	The number of measurements in the block.
`Total stns`	76–89	14	Total number of stations (i.e. inner + junction).

Table C.5: Individual block data table

Heading	Columns	Width	Comments
`Inner stns`	1–15	16	Index of the inner station in the binary station file.
`Junction stns`	16–31	16	Index of the junction station in the binary station file.
`Measurements`	32–47	11	Index of the measurement in the binary measurement file.
`Type`	48–52	5	Measurement type.

```
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------
DYNADJUST SEGMENTATION OUTPUT FILE

Version:                           3.2.1, Release (64-bit)
Build:                             Mar  2 2016, 10:55:54 (MSVC++ 10.0)
File created                       Wednesday, 02 March 2016, 10:56:44 AM
File name                          c:\data\gnss_example.seg

Command line arguments             segment gnss_example --min 2 --max 5 --start 409704930

Stations file                      c:\Data\gnss_example.bst
Measurements file                  c:\Data\gnss_example.bms

Minimum inner stations             2
Block size threshold               5
Starting stations                  409704930
--------------------------------------------------------------------------------

SEGMENTATION SUMMARY

No. blocks produced                2
------------------------------------------------------------------------------------------
  Block       Network ID    Junction stns   Inner stns      Measurements    Total stns
  1           0             2               5               13              7
  2           0             0               8               17              8
------------------------------------------------------------------------------------------

INDIVIDUAL BLOCK DATA
------------------------------------------------------------------------------------------

Block 1
-----------------------------------------------------
Junction stns:  2
Inner stns:     5
Measurements:   13
Total stns:     7

Inner stns      Junction stns   Measurements    Type
-----------------------------------------------------
0               7               0               G
1               8               3               G
3                               6               G
4                               9               G
11                              12              G
                                15              G
                                18              G
                                21              G
                                24              G
                                27              G
                                30              G
                                33              G
                                36              G
-----------------------------------------------------

Block 2
-----------------------------------------------------
Junction stns:  0
Inner stns:     8
Measurements:   17
Total stns:     8

Inner stns      Junction stns   Measurements    Type
-----------------------------------------------------
2                               39              G
5                               42              G
6                               45              G
7                               48              G
8                               51              G
9                               54              G
10                              57              G
12                              60              G
                                63              G
                                66              G
                                69              G
                                72              G
                                75              G
                                78              G
                                81              G
                                84              G
                                87              G
-----------------------------------------------------
```

C.6 Coordinate output file (XYZ)

The coordinate output (`.xyz`) file is produced by **adjust** after an adjustment and contains a list of the rigorous station coordinate estimates and uncertainties. The coordinate information will commence after the section '`Adjusted Coordinates`' and dashed line following it. Table C.6 describes the structure and formatting of the coordinate information printed to this file.

Since the user may select the type(s) and order of the coordinates to be printed, and whether or not station corrections in east, north and up should be printed (c.f. §8.3.4, page 128), Table C.6 lists the formatting for *all* possible data types. Hence, the columns to which each heading and value are printed may vary, however the width will remain as shown in the Table. The default precision for units in metres is 4. The default format for angular units is degrees, minutes and seconds using HP notation (e.g. dd.mmssssss) with a precision of 4 decimal places of a second.

If a phased adjustment has been undertaken and the option `--output-stn-blocks` has been provided to **adjust** to print station coordinates in sections according to the segmented blocks, the header field `Stations printed in blocks` will be set to `Yes` and the coordinates will be separated into sections beginning with the heading '`Block #`' where # is the respective block number.

Table C.6: Adjusted station coordinates

Heading	Columns	Width	Comments
`Station name`	1–20	20	Alphanumeric string. Can contain spaces. Corresponds to the station names in the Station file.
`Const`	21–25	5	Three character string representing the station constraints loaded from the station file, including any additional constraints applied upon executing **adjust**.
`X`	26–40	15	X coordinate
`Y`	41–55	15	Y coordinate
`Z`	56–70	15	Z coordinate
`Latitude`	71–84	14	Latitude
`Longitude`	85–99	15	Longitude
`H(Ortho)`	100–110	11	Orthometric height
`h(Ellipse)`	111–121	11	Ellipsoid height
`Easting`	122–135	14	UTM Easting
`Northing`	136–150	15	UTM Northing
`Zone`	151–158	8	UTM zone
		2	Blank space
`SD(e)`	161–170	10	Standard deviation (1 sigma, $k = 0.683$) in east–west direction
`SD(n)`	171–180	10	Standard deviation (1 sigma, $k = 0.683$) in north–south direction
`SD(up)`	181–190	10	Standard deviation (1 sigma, $k = 0.683$) in up direction
`Corr(e)`	193–203	11	Correction (m) in east–west direction. Optional output.
`Corr(n)`	204–214	11	Correction (m) in north–south direction. Optional output.
`Corr(up)`	215–225	11	Correction (m) in up direction. Optional output.
`Description`	228–283	56	Description obtained from the input station file

The example on the next page shows a coordinate output file for a ten–station GNSS point cluster produced from the following command:

```
adjust auscope --stn-coord PLhHENz --stn-cor
```

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------
DYNADJUST COORDINATE OUTPUT FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Oct 18 2016, 21:15:12 (MSVC++ 10.0)
File created:                      Thursday, 20 October 2016, 8:44:59 PM
File name:                         c:\Data\auscope.2016.281.simult.adj

Reference frame:                   GDA94
Epoch:                             01.01.1994
Geoid model:                       c:\Data\ausgeoid09.gsb
Station coordinate types:          PLHhXYZ
Stations printed in blocks:        No
Station coordinate corrections:    Yes
--------------------------------------------------------------------------------

Adjusted Coordinates
------------------------------------------

Station             Const      Latitude      Longitude h(Ellipse)   H(Ortho)      Easting      Northing    Zone      SD(e)     SD(n)    SD(up)      Corr(e)    Corr(n)   Corr(up)  Description
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
BDLE                FFF   -37.453101249  147.392168354   126.0462   117.5732  557785.6828  5820764.0674      55     0.0004    0.0004    0.0015      -0.0000    -0.0000     0.0000  Bairnsdale
BEEC                FFF   -36.204720955  146.392785392   443.0460   431.7983  469288.0989  5977570.0200      55     0.0003    0.0004    0.0014       0.0000     0.0000     0.0000  Beechworth
BROC                FFF   -36.015307858  144.121443042   131.5443   124.8616  248072.8417  6008950.1585      55     0.0002    0.0003    0.0009       0.0000     0.0000     0.0000  Bald Rock
GABO                FFF   -37.340528406  149.545470658    24.0108    14.2482  757468.8589  5838104.0853      55     0.0004    0.0004    0.0015       0.0000    -0.0000     0.0000  Gabo Island
MNGO                FFF   -38.464726709  143.390618768    62.6922    62.7174  730344.8964  5704319.3171      54     0.0004    0.0005    0.0020       0.0000     0.0000     0.0000  Marengo
MOBS                FFF   -37.494589896  144.583120660    40.6783    35.9030  321819.5907  5811180.0342      55     0.0002    0.0003    0.0009       0.0000     0.0000     0.0000  Melbourne Observatory
MTEM                FFF   -37.351546510  143.265604555   518.0798   514.4954  716222.3846  5837117.2130      54     0.0003    0.0004    0.0015       0.0000     0.0000     0.0000  Mt Emu
NHIL                FFF   -36.183034611  141.384561955   139.0387   136.8019  557995.7461  5981647.7258      54     0.0003    0.0004    0.0013       0.0000     0.0000     0.0000  Nhil
PTLD                FFF   -38.203985552  141.364848580     0.9651     4.4363  553608.2465  5755793.4327      54     0.0003    0.0004    0.0013       0.0000    -0.0000    -0.0000  Portland
STNY                FFF   -38.223067510  145.125052090    29.3115    25.7223  343993.3077  5751046.0773      55     0.0003    0.0003    0.0010       0.0000     0.0000     0.0000  Stony Point
YANK                FFF   -38.484419898  146.122480626    29.8992    26.6257  431141.6910  5703756.1314      55     0.0005    0.0007    0.0024       0.0000     0.0000     0.0000  Yanakie
```

C.7 Adjustment output file (ADJ)

The adjustment output (`.adj`) file is produced by **adjust** and, depending on which options have been provided, may contain basic or detailed information relating to an adjustment. Again, depending on what options have been provided, a high amount of variability may exist in the formatting of the results.

C.7.1 Adjustment statistics

By default, **adjust** will print to the adjustment output file a summary of the adjustment statistics and a listing of the estimated station coordinates. The adjustment statistics summary commences with a dashed line, then the following information:

Table C.7: Adjustment statistics summary table

SOLUTION	Converged or Failed
Total time	The total wall time to undertake the adjustment (not including the time taken to load the input files or to print the output files).
Number of unknown parameters	This value corresponds to the number (u) of parameters to be estimated, excluding those station coordinates which have been fixed. A station is comprised of three parameters (i.e. x, y, z).
Number of measurements	This value reports the number (n) of measurement components. Note, that each GNSS baseline and GNSS point is regarded as having three measurement components (i.e. x, y, z).
Degrees of freedom	This value (r) is calculated directly from u and n (c.f. equation 8.3), excluding all fixed station coordinates, and is the parameter against which the global rigorous sigma zero value is tested.
Chi squared	This quantity (w) is calculated from the least squares adjustment using equation 8.12.
Rigorous sigma zero	This value ($\hat{\sigma}^2$) is derived from w and r and is used to test the least squares solution as a whole (c.f. §9.3.1).
Global (Pelzer) Reliability	This quantity (T^2) estimates the global measurement reliability criterion using equation 7.67.
Chi–Square test (95.0%)	This statement reports the result of the global test of $\hat{\sigma}^2$ using the default or–user specified confidence interval (c.f. §8.3.3, page 124). Full details on this test will be explained in §9.3.1.

C.7.2 Measurement to station connections

The table of measurement to station connections commences with the section heading '`Measurements to Station`' and shows the frequency of each measurement type and the total number of measurements associated with each station. This table is identical to that which is in the measurement to station `.m2s` file produced by **import** (c.f. §C.3). Table C.1 shows the formatting of this table.

C.7.3 Adjusted measurements

The table of adjusted measurements and their associated statistics commences with the section heading '`Adjusted Measurements`' and, when the full range of options are selected, will contain the columns shown in §C.8.

Table C.8: Adjusted measurements and associated statistics table

Heading	Columns	Width	Comments
`M`	1–2	2	Alpha character corresponding to the measurement type.
`Station 1`	3 – 22	20	Alphanumeric string. Can contain spaces. Must correspond to a station name in the Station file.
`Station 2`	23 – 42	20	As per first station name.
`Station 3`	43 – 62	20	As per first station name.
`*`	63–65	3	This column reports whether the measurement was ignored or not.
`C`	66–67	2	Coordinate cardinal (e.g. `e`, `n`, `u`; `P`, `L`, `H`; `X`, `Y`, `Z`).
`Measured`	68–86	19	The original measurement obtained from the input file.
`Adjusted`	87–105	19	The adjusted measurement.
`Correction`	106–117	12	The estimated random error (or residual) v in the original measurement. Linear corrections are in metres, whereas angular corrections are in seconds.
`Meas. SD`	118–130	13	The standard deviation of the original measurement. Linear values are in metres, whereas angular values are in seconds.
`Adj. SD`	131–143	13	The standard deviation of the adjusted measurement. Linear values are in metres, whereas angular values are in seconds.
`Corr. SD`	144–156	13	The standard deviation of the measurement corrections, computed as the difference between the precisions of the original and adjusted measurements. Linear / angular units apply.
`N-stat`	157–167	11	The standardised Normal statistic, calculated relative to the unit Normal distribution. See §9.3.2.1
`T-stat`	168–178	11	The standardised Student's t statistic, calculated relative to the unit Student's t distribution. Optional output and is disabled by default. To enable, add the `--output-tstat-adj-msr` option to the list of **adjust** commands. See §9.3.2.2.
`Pelzer Rel`	179–190	12	Pelzer's reliability criterion. See §7.3.3.3.
`Pre Adj Corr`	191–204	14	The total (systematic) correction applicable to this measurement applied prior to least squares adjustment, such as geoid–ellipsoid separation (N), deflection of the vertical (c.f. equations 7.30 and 7.37) or Laplace correction (c.f. equation 7.42).
`Outlier?`	205–216	12	An asterisk denoting whether this measurement has failed the local test. See §9.3.2.1.
`Msr ID`	217–226	10	The original measurement ID obtained from the input file. Optional output and is disabled by default. To enable, add the `--output-database-ids` option to the list of **adjust** commands.
`Cluster ID`	227–236	10	Original cluster ID obtained from the input file (associated with Measurement IDs).

The example on the following page shows some of the adjusted measurements arising from a simultaneous adjustment of the `uni_sqr` network with the `--output-tstat-adj-msr` option set. See §8.3.4 on page 127 for a range of options for configuring the output of adjusted measurements, such as GNSS baseline units, sort order and output precision.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

Adjusted Measurements
------------------------------------------

M Station 1          Station 2           Station 3           *  C          Measured          Adjusted  Correction    Meas. SD     Adj. SD    Corr. SD    N-stat    T-stat  Pelzer Rel  Pre Adj Corr    Outlier?
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
H 1042                                                                      43.0859           43.0869      0.0010      0.0650      0.0062      0.0647      0.02      0.02        1.00        4.8659
Y 1042                                                              P  -37 47 52.0000    -37 47 52.4755     -0.4755     20.0003      0.0002     20.0003     -0.02      0.01        1.00        0.0000
Y 1042                                                              L  144 57 37.0000    144 57 37.2520      0.2520     20.0003      0.0002     20.0003      0.01     -0.03        1.00        0.0000
Y 1042                                                              H           43.1640           43.0869     -0.0771      0.5000      0.0053      0.5000     -0.15     -0.03        1.00        4.8659
Y 2215                                                              P  -37 48 01.0000    -37 48 05.1554     -4.1554     20.0003      0.0000     20.0003     -0.21      0.03        1.00        0.0000
Y 2215                                                              L  144 57 32.0000    144 57 35.6306      3.6306     20.0003      0.0001     20.0003      0.18     -0.29        1.00        0.0000
Y 2215                                                              H           57.0640           57.0650      0.0010      0.5000      0.0015      0.5000      0.00     -0.23        1.00        4.8521
Y 1004                                                              P  -37 48 00.0000    -37 48 00.5702     -0.5702     20.0003      0.0001     20.0003     -0.03      0.02        1.00        0.0000
Y 1004                                                              L  144 57 34.0000    144 57 34.0939      0.0939     20.0003      0.0001     20.0003      0.00     -0.02        1.00        0.0000
Y 1004                                                              H           42.1950           42.2181      0.0231      0.5000      0.0034      0.5000      0.05     -0.03        1.00        4.8561
Y 9004                                                              P  -37 47 48.0000    -37 47 48.6456     -0.6456     20.0003      0.0001     20.0003     -0.03      0.01        1.00        0.0000
Y 9004                                                              L  144 57 36.0000    144 57 36.3873      0.3873     20.0003      0.0001     20.0003      0.02     -0.04        1.00        0.0000
Y 9004                                                              H           44.3240           44.3168     -0.0072      0.5000      0.0028      0.5000     -0.01     -0.04        1.00        4.8694
A 2013               2012                1032                           91 41 49.5000     91 41 49.5075      0.0011     20.0000     20.0000      0.0148      0.07      0.08      999.99       -0.0064
G 2206               1039                                           X         -217.5180         -217.5272     -0.0092      0.0316      0.0076      0.0307     -0.30     -0.33        1.03        0.0000
G 2206               1039                                           Y         -166.5620         -166.5477      0.0143      0.0316      0.0077      0.0307      0.47      0.51        1.03        0.0000
G 2206               1039                                           Z          129.7860          129.7739     -0.0121      0.0316      0.0087      0.0304     -0.40     -0.43        1.04        0.0000
G 2215               BARR                                           X          242.0880          242.0793     -0.0087      0.0063      0.0025      0.0058     -1.49     -1.62        1.08        0.0000
G 2215               BARR                                           Y           28.7460           28.7462      0.0002      0.0039      0.0019      0.0034      0.06      0.06        1.15        0.0000
G 2215               BARR                                           Z         -192.4643         -192.4677     -0.0034      0.0082      0.0028      0.0078     -0.44     -0.48        1.06        0.0000
K 1046               4010                                                   91 20 01.0000     91 20 10.7885     12.7820     20.0000     10.0371     17.2990      0.74      0.80        1.16        2.9935
B 1046               4010                                                   91 20 31.0000     91 20 10.7885    -20.2115     20.0000     10.0371     17.2990     -1.17     -1.27        1.16        0.0000
V 1013               1014                                                   90 24 31.0000     90 24 35.6599     -0.6425     20.0000     18.8328      6.7324     -0.10     -0.10        2.97       -5.3024
V 1013               1017                                                   91 47 53.0000     91 47 47.7279     -1.0274     20.0000     19.6744      3.5944     -0.29     -0.31        5.56        4.2447
M 4000               13                                                             53.9380           53.9307     -0.0073      0.0050      0.0028      0.0041     -1.77     -1.92        1.21        0.0000
A 4000               1050                13                                 84 10 09.5000     84 10 04.9638     -4.5990     20.0000     15.7313     12.3502     -0.37     -0.41        1.62       -0.0628
A 4000               1050                2012                              171 55 24.5000    171 55 19.6089     -4.7430     20.0000     14.9844     13.2464     -0.36     -0.39        1.51        0.1481
...
S 4000               13                                                             53.9280           53.9311      0.0031      0.0050      0.0028      0.0041      0.76      0.82        1.21        0.0000
S 4000               2012                                                           76.9140           76.9145      0.0005      0.0050      0.0031      0.0039      0.12      0.13        1.27        0.0000
S 2012               4000                                                           76.9080           76.9088      0.0008      0.0050      0.0031      0.0039      0.20      0.22        1.27        0.0000
S 2012               2014                                                           62.9750           62.9759      0.0009      0.0050      0.0046      0.0019      0.49      0.53        2.66        0.0000
S 2012               2013                                                           28.4890           28.4901      0.0011      0.0050      0.0034      0.0037      0.30      0.33        1.37        0.0000
...
Z 4000               1050                                                    0 37 11.0000      0 37 08.3998     -0.9140     20.0000     10.6038     16.9575     -0.05     -0.06        1.18        1.6862
V 4000               1050                                                   89 22 59.0000     89 22 51.6002     -5.7135     20.0000     10.6038     16.9575     -0.34     -0.37        1.18        1.6862
V 4000               13                                                     89 51 51.0000     89 52 08.6295      9.8936     20.0000     13.3083     14.9294      0.66      0.72        1.34       -7.7359
V 4000               2012                                                   91 50 46.5000     91 50 42.4656     -6.8202     20.0000     12.0257     15.9807     -0.43     -0.46        1.25       -2.7858
...
L 108                1034                                                           -0.2220           -0.2126      0.0094      0.0100      0.0073      0.0068      1.38      1.50        1.47        0.0017
L 108                1002                                                            2.2700            2.2620     -0.0080      0.0100      0.0073      0.0068     -1.18     -1.28        1.47        0.0025
L 1002               1003                                                            0.3420            0.3417     -0.0003      0.0100      0.0044      0.0090     -0.03     -0.04        1.11       -0.0027
L PELH               NEWP                                                            0.0790            0.0890      0.0100      0.0100      0.0015      0.0099      1.01      1.10        1.01        0.0000
...
G BARR               PELH                                           X          -74.0779          -74.0646      0.0133      0.0125      0.0026      0.0122      1.09      1.19        1.02        0.0000
G BARR               PELH                                           Y         -107.1686         -107.1708     -0.0022      0.0076      0.0022      0.0072     -0.30     -0.33        1.05        0.0000
G BARR               PELH                                           Z           -0.5302           -0.5201      0.0101      0.0079      0.0026      0.0074      1.36      1.48        1.06        0.0000
G BARR               BOWL                                           X         -102.6500         -102.6460      0.0040      0.0079      0.0024      0.0075      0.53      0.57        1.05        0.0000
G BARR               BOWL                                           Y           27.3013           27.2897     -0.0116      0.0072      0.0024      0.0068     -1.70     -1.85        1.06        0.0000
G BARR               BOWL                                           Z          121.3314          121.3360      0.0046      0.0106      0.0027      0.0102      0.45      0.49        1.03        0.0000
G BOWL               PELH                                           X           28.5858           28.5814     -0.0044      0.0075      0.0025      0.0070     -0.62     -0.68        1.06        0.0000
G BOWL               PELH                                           Y         -134.4629         -134.4605      0.0024      0.0066      0.0023      0.0062      0.39      0.42        1.06        0.0000
G BOWL               PELH                                           Z         -121.8631         -121.8561      0.0070      0.0060      0.0021      0.0056      1.25      1.36        1.07        0.0000
...
```

C.7.4 Estimated station coordinates

The output of the estimated station coordinates commences with the heading '`Adjusted Coordinates`' and is identical to the output contained in the coordinate output `.xyz` file. Table C.6 shows the formatting of the coordinate listing.

The format and structure of the information in this section will vary depending on the options provided to **adjust**. For instance, by default, all estimated station coordinates will appear in a single block, sorted alphabetically, with coordinate types latitude, longitude, orthometric height, ellipsoidal height, X, Y, Z. Latitude and longitude values will be presented with a precision of 5 decimal places of a second, and values in metres will be shown to 4 decimal places. As discussed in §C.6, the order of the station coordinates can be sorted to the original order as found in the input station file using `--sort-stn-orig-order`; station coordinates produced from phased adjustment can be presented in blocks by `--output-stn-blocks`; the type of station coordinates can be altered by `--stn-coord-types`; and the precision of the coordinates can be altered by `--precision-stn-linear` and `--precision-stn-angular`.

C.7.5 Output of ignored measurements

Optionally, **adjust** is able to print ignored measurements (c.f. §8.3.4, page 129). If this option is provided to **adjust**, the adjustment output file will contain a section heading titled '`Ignored Measurements (a-posteriori)`'. Table C.9 shows the structure and format of the ignored measurements table.

Table C.9: Ignored measurements (a–posteriori)

Heading	**Columns**	**Width**	**Comments**
`M`	1–2	2	Alpha character corresponding to the measurement type.
`Station 1`	3 – 22	20	Alphanumeric string. Can contain spaces. Corresponds to a station name in the Station file.
`Station 2`	23 – 42	20	As per first station name.
`Station 3`	43 – 62	20	As per first station name.
`*`	63–65	3	Indicates whether a measurement is ignored or not.
`C`	66–67	2	Coordinate cardinal (e.g. `e`, `n`, `u`; `P`, `L`, `H`; `X`, `Y`, `Z`).
`Measured`	68–86	19	The original measurement obtained from the input file.
`Computed`	87–105	19	The measurement computed from the latest station coordinate estimates.
`Difference`	106–117	12	The difference between the original and computed measurement. Linear corrections are in metres, whereas angular corrections are in seconds.
`Meas. SD`	118–130	13	The standard deviation of the original measurement. Linear values are in metres, whereas angular values are in seconds.
`Pre Adj Corr`	131–144	14	The total (systematic) correction applicable to this measurement applied prior to least squares adjustment.

C.7.6 Output of results on each iteration

Optionally, **adjust** is able to print upon each iteration the current station coordinate estimates, the summary of adjustment statistics, the measurements computed prior to adjustment and the adjusted measurements (c.f. §8.3.4, page 129). If any of these options are provided to **adjust**, the adjustment output file will contain section headings titled 'ITERATION #' where # refers to the subject iteration.

The output of adjustment statistics, adjusted measurements and adjusted station coordinates will follow the same format as described in §C.7.1, §C.7.3 and §C.7.4 respectively. The a–priori computed measurements will be preceded with the heading 'Computed Measurements (a-priori)' and will follow a similar format and structure to the ignored measurements listed in Table C.9.

C.8 Station coordinate corrections file (COR)

The coordinate corrections output (.cor) file is produced by **adjust** and lists the three–dimensional shifts to the a–priori coordinates supplied in the input station file. Table C.10 shows the structure and format of the coordinate corrections table.

Table C.10: Coordinate corrections table

Heading	Columns	Width	Comments
Station	1 – 20	20	Station name (corresponding to the input station file)
Azimuth	23–41	19	Horizontal azimuth (from true north)
V.Angle	42–60	19	Vertical azimuth (from the horizontal plane)
S.Distance	61–79	19	Slope distance correction
H.Distance	80–98	19	Horizontal distance correction
east	99–109	11	East–ward correction (in the local reference frame)
north	110–120	11	North–ward correction
up	121–131	11	Up–ward correction

If either --hz-corr-threshold or --vt-corr-threshold options are supplied upon executing **adjust** (c.f. §8.3.5, page 130), the coordinate corrections will be limited to those corrections which exceed the respective horizontal and vertical thresholds.

The example on the facing page shows a sample of the coordinate corrections output for the uni_sqr network.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
--------------------------------------------------------------------------------
DYNADJUST CORRECTIONS OUTPUT FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Oct 18 2016, 21:15:12 (MSVC++ 10.0)
File created:                      Thursday, 20 October 2016, 5:55:49 PM
File name:                         c:\Data\dist\uni_sqr.simult.cor

Stations printed in blocks         No
--------------------------------------------------------------------------------

Corrections to stations
------------------------------------------

Station                           Azimuth           V. Angle        S. Distance        H. Distance       east      north         up
-----------------------------------------------------------------------------------------------------------------------------------
0102                            159 37 11          -82 20 18             0.0380             0.0051     0.0018    -0.0047    -0.0377
1002                             78 33 17          -26 13 15             0.0186             0.0167     0.0163     0.0033    -0.0082
1003                             70 48 05          -10 55 44             0.0396             0.0388     0.0367     0.0128    -0.0075
1004                            134 10 10          -20 12 05             0.0084             0.0078     0.0056    -0.0055    -0.0029
1005                            190 16 02          -85 41 35             0.0078             0.0006    -0.0001    -0.0006    -0.0078
1006                            146 34 51          -52 38 46             0.0089             0.0054     0.0030    -0.0045    -0.0070
1007                            141 12 57          -12 07 46             0.0094             0.0092     0.0058    -0.0072    -0.0020
1008                            135 53 54           -6 16 32             0.0345             0.0343     0.0239    -0.0246    -0.0038
101                             152 04 35           -2 56 37             0.2509             0.2506     0.1174    -0.2214    -0.0129
1010                            239 36 08           -2 00 57             0.0801             0.0801    -0.0691    -0.0405    -0.0028
1011                            143 29 55            9 41 38             0.0466             0.0460     0.0273    -0.0369     0.0079
1012                            144 17 17           20 19 55             0.0316             0.0296     0.0173    -0.0240     0.0110
1013                            130 34 02           20 43 20             0.0274             0.0257     0.0195    -0.0167     0.0097
1014                            138 08 53           18 25 01             0.0238             0.0225     0.0150    -0.0168     0.0075
1015                            124 56 37            7 29 55             0.0110             0.0109     0.0090    -0.0063     0.0014
1016                            171 11 57           -0 44 58             0.0182             0.0182     0.0028    -0.0180    -0.0002
1017                            147 35 40           29 42 37             0.0233             0.0202     0.0108    -0.0171     0.0115
1018                            123 23 53            0 52 43             0.0112             0.0112     0.0094    -0.0062     0.0002
1019                            128 12 06            0 35 33             0.0092             0.0092     0.0072    -0.0057     0.0001
1022                            117 53 03           44 50 07             0.0138             0.0098     0.0086    -0.0046     0.0097
1023                             77 30 02          -37 39 23             0.0163             0.0129     0.0126     0.0028    -0.0099
1025                            125 20 52           51 29 01             0.0175             0.0109     0.0089    -0.0063     0.0137
1027                            123 45 14           44 13 43             0.0161             0.0115     0.0096    -0.0064     0.0112
1029                            268 16 47          -43 55 10             0.0485             0.0350    -0.0349    -0.0010    -0.0337
1030                            266 39 57          -29 32 12             0.0511             0.0445    -0.0444    -0.0026    -0.0252
1032                            195 55 06           -1 11 37             0.4920             0.4919    -0.1349    -0.4730    -0.0102
1033                             22 46 24            0 15 14             0.0560             0.0560     0.0217     0.0516     0.0002
1034                             68 10 37           10 23 21             0.0232             0.0228     0.0212     0.0085     0.0042
1037                             91 09 34          -55 10 05             0.0139             0.0079     0.0079    -0.0002    -0.0114
1039                            119 16 32          -25 26 12             0.0104             0.0094     0.0082    -0.0046    -0.0045
...
```

C.9 Adjusted positional uncertainty file (APU)

The adjusted positional uncertainty output (`.apu`) file is produced by **adjust** and lists the rigorous positional uncertainty (horizontal and vertical), standard error ellipse elements and upper–triangular variance matrix elements for each station arising from the least squares adjustment. Table C.11 shows the structure and format of the adjusted positional uncertainty table.

Table C.11: Adjusted positional uncertainty table

Heading	Columns	Width	Comments
`Station`	1 – 20	20	Station name (corresponding to the input station file)
`Latitude`	23–36	14	Estimated latitude
`Longitude`	37–51	15	Estimated longitude
`Hz PosU`	52–62	11	Horizontal radius (95% confidence interval)
`Vt PosU`	63–73	11	Vertical uncertainty (95% confidence interval)
`Semi-major`	74–86	13	Semi–major axis of the standard error ellipse
`Semi-minor`	87–99	13	Semi–minor axis of the standard error ellipse
`Orientation`	100–112	13	Orientation of the standard error ellipse
`Variance (X/e)`	113–131	19	$\mathbf{V}_{s_{11}}$ ($\sigma_x\sigma_x$ or $\sigma_e\sigma_e$)
`Variance (Y/n)`	132–150	19	$\mathbf{V}_{s_{12}}$ ($\sigma_x\sigma_y$ or $\sigma_e\sigma_n$) $\mathbf{V}_{s_{22}}$ ($\sigma_y\sigma_y$ or $\sigma_n\sigma_n$)
`Variance (Z/up)`	151–169	19	$\mathbf{V}_{s_{13}}$ ($\sigma_x\sigma_z$ or $\sigma_e\sigma_{up}$) $\mathbf{V}_{s_{23}}$ ($\sigma_y\sigma_z$ or $\sigma_n\sigma_{up}$) $\mathbf{V}_{s_{33}}$ ($\sigma_z\sigma_z$ or $\sigma_{up}\sigma_{up}$)

In accordance with the *Guideline for the Adjustment and Evaluation of Survey Control* ICSM (2014), the positional uncertainty elements are computed at 95% (c.f. §7.3.3.2) irrespective of the chosen confidence interval. The error ellipse terms and variance matrix elements are expressed as one–sigma. This enables the output variance matrix elements to be used directly as input constraints for subsequent (or subsidiary) survey control projects.

By default, all variance matrix elements are expressed in the cartesian coordinate system. If the `--output-apu-vcv-units` option is provided to **adjust** with an argument of 1, all variance matrix elements will be propagated to the local reference frame and expressed in terms of e, n and up. In this case, the header field `Variance matrix units` will be set to `ENU`, and the column headers for the variance matrix elements will be `Variance (e)`, `Variance (n)` and `Variance (up)`.

If a phased adjustment has been undertaken and the option `--output-stn-blocks` has been provided to **adjust**, the header field `Stations printed in blocks` will be set to `Yes` and the variances will be separated into sections beginning with the heading '`Block #`' where `#` is the respective block number.

If the `--output-all-covariances` option is provided to **adjust**, covariances for the upper–triangular component of the full variance matrix will be printed in rows between the variances for the respective stations. Information for a single covariance block $\mathbf{C}_{s_i s_j}$ (between stations s_i and s_j) contains nine elements positioned over three records. These blocks are repeated for as many covariance blocks required to complete each row of the upper–triangular matrix $\mathbf{V}_{s_{ij}}$. When this option is selected, the header field `Full covariance matrix` will be set to `Yes`. Table C.12 shows the structure and format of the covariance blocks.

Table C.12: Positional uncertainty covariance block

Heading	Columns	Width	Comments
`Variance(X/e)`	113–131	19	$\mathbf{C}_{11}$ (first row) $\mathbf{C}_{21}$ (second row) $\mathbf{C}_{31}$ (third row)
`Variance(Y/n)`	132–150	19	$\mathbf{C}_{12}$ (first row) $\mathbf{C}_{22}$ (second row) $\mathbf{C}_{32}$ (third row)
`Variance(Z/up)`	151–169	19	$\mathbf{C}_{13}$ (first row) $\mathbf{C}_{23}$ (second row) $\mathbf{C}_{33}$ (third row)

The example on the next page shows the formatting of the positional uncertainty output for five GNSS CORS sites (`MTEM`, `NHIL`, `PTLD`, `STNY` and `YANK`), with variance matrix elements printed in the (default) cartesian system, and with all covariance blocks.

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
--------------------------------------------------------------------------------
DYNADJUST POSITIONAL UNCERTAINTY OUTPUT FILE

Version:                           3.2.6, Release (64-bit)
Build:                             Oct 21 2016, 08:21:55 (MSVC++ 10.0)
File created:                      Wednesday, 02 November 2016, 1:23:13 PM
File name:                         C:\Data\dist\2016_GDA2020\auscope.2016.281.simult.apu

PU confidence interval             95.0%
Stations printed in blocks         No
Variance matrix units              XYZ
Full covariance matrix             Yes
--------------------------------------------------------------------------------

Positional uncertainty of adjusted station coordinates
------------------------------------------------------

Station                    Latitude      Longitude    Hz PosU    Vt PosU   Semi-major   Semi-minor  Orientation       Variance(X)       Variance(Y)       Variance(Z)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
MTEM                   -37.351543592  143.265605967     0.0211     0.0616       0.0094       0.0077     179.1009   4.846809052e-04  -3.054932865e-04   3.432502058e-04
                                                                                                                                   2.784526132e-04  -2.443643972e-04
                                                                                                                                                     3.712110788e-04
NHIL                                                                                                                 1.978086552e-05  -6.652906981e-06   7.108005259e-06
                                                                                                                    -6.417476502e-06   1.590022835e-05  -4.826451197e-06
                                                                                                                     7.240680472e-06  -5.175665729e-06   1.779002888e-05
PTLD                                                                                                                 1.423033977e-05  -2.914252624e-06   3.310305848e-06
                                                                                                                    -2.828090458e-06   1.250344141e-05  -2.151283789e-06
                                                                                                                     3.187831757e-06  -2.133807649e-06   1.362578436e-05
STNY                                                                                                                 3.982497312e-05  -1.809900497e-05   2.193434324e-05
                                                                                                                    -1.857501340e-05   2.559121232e-05  -1.480062889e-05
                                                                                                                     2.160627581e-05  -1.416263970e-05   3.331940004e-05
YANK                                                                                                                 1.186492250e-05  -1.294983466e-06   1.532820860e-06
                                                                                                                    -1.572900826e-06   1.128156818e-05  -1.115428788e-06
                                                                                                                     1.455995371e-06  -7.386412641e-07   1.176504605e-05
NHIL                   -36.183031660  141.384563456     0.0119     0.0286       0.0052       0.0045     174.1112   1.033775111e-04  -6.539928097e-05   6.836658670e-05
                                                                                                                                       7.237999713e-05  -5.309352587e-05
                                                                                                                                                         8.437149226e-05
PTLD                                                                                                                 8.477705259e-06   9.252335698e-07  -9.143204033e-07
                                                                                                                     9.199795153e-07   9.203596111e-06   8.732055712e-07
                                                                                                                    -1.035811291e-06   1.000480253e-06   9.300828583e-06
STNY                                                                                                                 1.595900972e-05  -3.818155445e-06   4.567432495e-06
                                                                                                                    -4.202071526e-06   1.364772566e-05  -3.301540864e-06
                                                                                                                     4.379726299e-06  -2.766092178e-06   1.512126315e-05
YANK                                                                                                                 1.158084155e-05  -1.063747954e-06   1.366753222e-06
                                                                                                                    -1.499979436e-06   1.115547005e-05  -1.087739799e-06
                                                                                                                     1.202427044e-06  -5.073664549e-07   1.151681076e-05
PTLD                   -38.203982588  141.364850004     0.0302     0.0929       0.0136       0.0109     179.3113   1.034536689e-03  -7.095825589e-04   7.704937836e-04
                                                                                                                                       6.669820273e-04  -5.927929204e-04
                                                                                                                                                         8.465184087e-04
STNY                                                                                                                 1.899834165e-05  -5.542236095e-06   6.558160273e-06
                                                                                                                    -5.852001001e-06   1.482967563e-05  -4.445744643e-06
                                                                                                                     6.639141911e-06  -4.208603650e-06   1.702382773e-05
YANK                                                                                                                 1.256279844e-05  -1.626720489e-06   1.861596484e-06
                                                                                                                    -2.047692692e-06   1.161458709e-05  -1.393744103e-06
                                                                                                                     1.999596373e-06  -1.004998101e-06   1.218399481e-05
STNY                   -38.223064609  145.125053409     0.0136     0.0340       0.0060       0.0050     175.1517   1.487928854e-04  -8.641608728e-05   1.043654831e-04
                                                                                                                                       8.600511767e-05  -7.183574891e-05
                                                                                                                                                         1.280843718e-04
YANK                                                                                                                 1.180914377e-05  -1.338168295e-06   1.523055571e-06
                                                                                                                    -1.433162306e-06   1.122194317e-05  -9.833189549e-07
                                                                                                                     1.435418279e-06  -8.103754064e-07   1.176084216e-05
YANK                   -38.484417008  146.122481907     0.0384     0.0975       0.0168       0.0145     173.2844   1.225903848e-03  -7.196501509e-04   8.562354669e-04
                                                                                                                                       7.213269628e-04  -6.006868514e-04
                                                                                                                                                         1.021421363e-03
```

C.10 SINEX output warning file (SNX.ERR)

Upon running **adjust**, users may export the adjustment results to a SINEX file. Since the SINEX specification has strict formatting rules which may not accommodate the data that DynAdjust supports, warnings may be produced at the time of SINEX export. One such inconsistency is the four–character station name restriction on station names enforced by the SINEX standard. In this context, each time a station is found with more than four characters, a warning file named `*.snx.err` will be produced. Each warning file will contain a record '`Station name ##### exceeds four characters`', where '`#####`' refers to the station in question. If a phased adjustment has been undertaken, warning files will be produced for each block and each file will be named according to the block number.