Practical Notes on Coordinate Transformation

PHILIP COLLIER
MARCH 2020

PRACTICAL NOTES ON COORDINATE TRANSFORMATION

Philip Collier
March 2020

Table of Contents

Glossary of terms

Term	Definition
Geodesy	The science that underpins the determination of the size and shape of the earth, the precise location of points and objects on and near the earth, and the 4D modelling of the earth's gravity field
Reference system	The theoretical and conceptual basis for the unique and repeatable description of location. The International Terrestrial Reference System (ITRS) is a well known example
Reference frame	The practical realisation of a reference system through the computation and publication of station coordinates expressed relative to the frame definition. The International Terrestrial Reference Frame 2014 (ITRF2014) is a contemporary example
Geodetic datum	A term used interchangeably with the term reference frame. It is nowadays more accepted to used the latter, while the term "datum" has fallen out of favour in the geodetic community
Coordinate conversion	Location can be expressed in different forms. Cartesian coordinates, geodetic (geographical) coordinates and map grid coordinates are common examples. Coordinate conversion is the mathematical processes whereby one form of coordinates is changed into another form within the same reference frame
Coordinate transformation	Location is always expressed relative to some pre-defined frame of reference and at some given epoch in time. Coordinate transformation is the generic term for the mathematical process whereby the reference frame and/or the reference epoch of a given location is changed to a different reference frame and/or epoch. The generic nature of the term "coordinate transformation" commonly causes confusion because it does not distinguish the spatial and temporal components
Spatial transformation	Spatial transformation refers to that component of the coordinate transformation process that deals with a change of reference frame at a common epoch
Temporal transformation	Temporal transformation refers to that component of the coordinate transformation process that deals with a change of reference epoch, while remaining in the same reference frame
Combined transformation	The explicit combination of both the spatial and temporal elements of the coordinate transformation process
Crustal motion	The surface of the earth is in constant motion due to the movement of the tectonic plates and, in some locations, due to local and regional crustal deformation. The latter may be natural (e.g. due to post-seismic deformation) or induced through human activity (anthropogenic) (e.g. due to ground water extraction or the impact of underground mining)
Deformation	In this paper, the term deformation is strictly taken to mean non-linear crustal motion. It refers to motion that cannot be modelled over time by fixed point velocities

Distortion	When a reference frame is realised, errors in the underlying measurements and shortcomings in data processing strategies will always impact the accuracy of the station coordinates. This is particularly true for older reference frames. The resultant errors in station coordinates are referred to as distortion
Accuracy	Closeness to the truth, quantified by the mean or average difference between the true and measured locations
Precision	Statistical repeatability of the solution, quantified by the standard deviation of the data sample
Map transformation	A process for shifting the user’s digital map data to spatially align it with the Swift navigation solution
Navigation transformation	A process for shifting the Swift solution to spatially align it with the user’s map data

Purpose

The purpose of this document is to provide a practical guide to the background, requirements and processes needed to temporally and spatially align geodetic reference frames. Emphasis is given to describing fundamental technical issues in succinct terms and on discussing important practical considerations.

The document begins by presenting basic concepts and definitions. It then distinguishes between coordinate conversion and coordinate transformation. Next follows the development of a logical process whereby the most appropriate transformation strategy for a given case can be determined and applied. Finally, a series of worked examples are provided to assist readers grasp and implement the preceding theoretical concepts. A series of appendices provide some of the essential mathematical background to coordinate conversion and transformation.

1. Background

1.1 Expressing Location

There are three common ways (forms) of expressing location on the earth. Consider Figure 1 which shows a typical geodetic ellipsoid and a 3D cartesian axes system, with its origin at the centre of the ellipsoid. We are interested in expressing the location of point P shown in Figure 1. A summary of the various ways of describing location and the advantages and disadvantages of each is provided in Table 1.

1.1.1 Option 1 – Cartesian coordinates

The simplest way to express the location of Point P is to use the 3D cartesian axes. In this case, we have: (X_P, Y_P, Z_P).

The benefit of the cartesian system is that the three components of the position vector are expressed in linear units (metres) and all calculations relating to position can be done using vector geometry. The disadvantage is the non-intuitive nature of cartesian coordinates in that it is challenging to conceptualise location when expressed in cartesian form as there are no logical "horizontal" and "vertical" components.

1.1.2 Option 2 – Geodetic coordinates

Alternatively, we can use the geodetic (or geographic) coordinate system of latitude, longitude and height. In this case we have: (ϕ_P, λ_P, h_P)

Geodetic coordinates are defined by the normal to the ellipsoid through P. Latitude is the angle formed between the normal and the equatorial plane, Longitude is the angle between the zero meridian (Greenwich) and the meridian containing the normal. The ellipsoid height is the distance along the ellipsoid normal from the surface of the ellipsoid to P.

The benefit of the geodetic system is the ease with which location can be conceptualised and visualised. The primary disadvantages are the use of curvilinear coordinates (degrees) and the complexity of spatial calculations on the surface of the reference ellipsoid. For example calculating the ellipsoid distance between two points expressed in latitude and longitude is not simple, though it is a more practically meaningful quantity than the vector distance derived from the corresponding cartesian coordinates.

1.1.3 Option 3 – Map grid coordinates

To escape the complexity of working on the curved surface of the ellipsoid and the non-intuitive nature of cartesian coordinates, planar or map grid coordinates are often used. Such coordinates are derived by application of the relevant map projection formulae to the corresponding geodetic coordinates. There are several types of map projection, but commonly, the Transverse Mercator (TM) or Universal Transverse Mercator (UTM) projections are adopted. Map grid coordinates are expressed as: (E_P, N_P, h_P).

Strictly speaking, the ellipsoidal height is not part of the map grid system, it simply follows from the geodetic coordinates. For practical reasons, the ellipsoidal height is often converted into a height above the geoid (an equipotential reference surface such as mean sea level) by subtracting the geoid undulation (the separation between the ellipsoid and the geoid). Map grid coordinates are expressed in linear units relative to the origin of a specified map grid zone covering the area of interest.

The benefit of the map grid system is the planar nature of the coordinates. However, every map projection has some form of distortion (e.g. scale, area, orientation) which must be accounted for in spatial calculations, introducing complexity and scope for error.

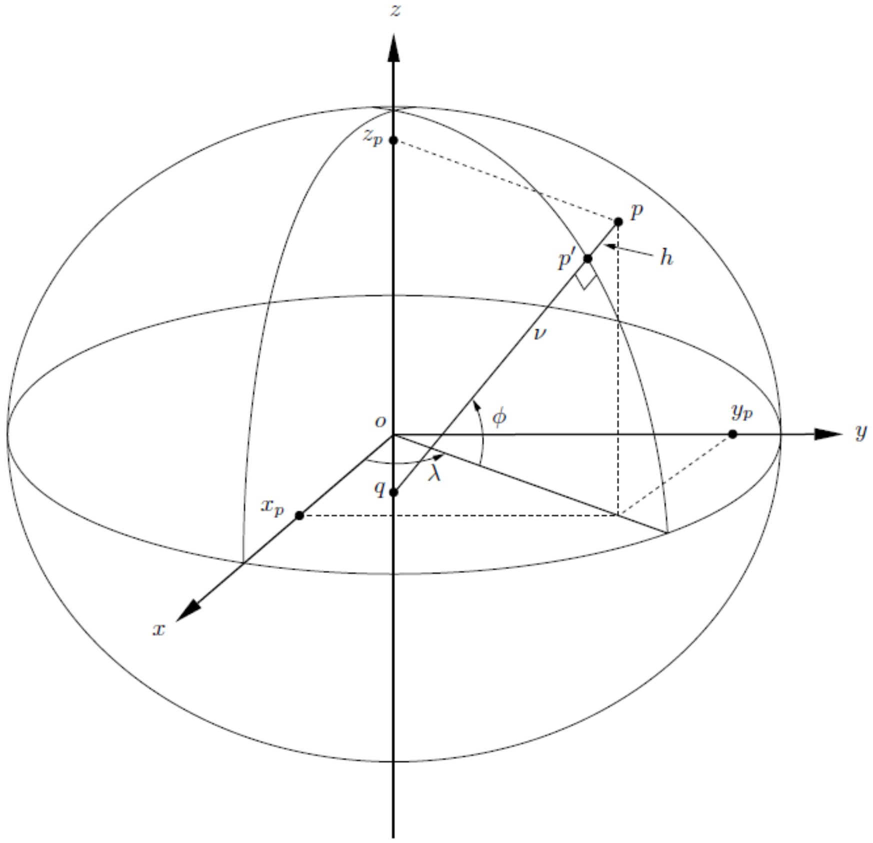

Figure 1 – The cartesian and geodetic coordinate systems

Name	Expression	Advantages	Disadvantages
Cartesian	(X_P, Y_P, Z_P)	• Expressed in linear units (metres) • Easy to work with (vector calculations)	• Difficult to humanly conceptualise and visualise location
Geodetic (Geographic)	(ϕ_P, λ_P, h_P)	• Conceptualisation of location is simple and *natural*	• Expressed in mixed units (sexagesimal/metres) • Calculations are complex due to the curvilinear form of latitude and longitude
Map Grid (planar)	(E_P, N_P, h_P)	• Planar • Expressed in linear units (metres) • Easy to visualise	• Complex calculations due to distortion introduced by projection • Map grid zones introduce ambiguity in location

Table 1 – Summary of different coordinate systems

1.2 Reference Systems and Reference Frames

1.2.1 What is a reference system?

In geodetic terms a reference *system* provides the theoretical or conceptual foundation for the realisation of station coordinates in an accessible reference *frame*. The design of a reference system involves four elements. Table 2 lists these elements and gives their definition for the well known International Terrestrial Reference System (ITRS).

Element	ITRS definition
Axes	A 3D, right-handed cartesian system
Location	The origin of the axes coincides with the centre of mass of the whole earth, including the atmosphere and the oceans. The system is thereby defined as being *geocentric*
Orientation	The Z-axis runs through the CTP (Conventional Terrestrial Pole) as defined by the BIH (Bureau Internationale de l'Heure) The Y-axis aligns with reference meridian for longitude (Greenwich) The XY-plane coincides with the reference parallel for latitude (Equator)
Scale	The international metre (SI system of units)

Table 2 – The defining elements of the International Terrestrial Reference System (ITRS)

1.2.2 What is a reference frame?

A geodetic reference *frame* provides a practical realisation of a reference system through the computation and publication of station coordinates expressed relative to the system definition. The most familiar and contemporary example is the International Terrestrial Reference Frame 2014 (ITRF2014). Figure 2 shows the stations and measurement techniques used in the realisation of ITRF2014. In total, ITRF2014 includes 3D station coordinates for 975 distinct global sites, and comprises inputs from four complementary geodetic observation techniques:

- Very Long Baseline Interferometry (VLBI)
- Satellite Laser Ranging (SLR)
- Doppler Oribitography and Radio-positioning Integrated by Satellite (DORIS)
- Global Navigation Satellite Systems (GNSS)

Data released as part of the ITRF2014 product suite includes:

- 3D stations coordinates and uncertainties at epoch 2010.0
- 3D station velocities and uncertainties at epoch 2010.0
- Earth orientation parameters (EOP)
- Post-seismic deformation (PSD) models

The International Earth Rotation and Reference Systems Service (IERS) is responsible for the establishment of the ITRS and the on-going realisation and maintenance of the ITRF. Substantial technical detail regarding the realisation of ITRF2014 can be found in Altamimi et al. (2016)[1].

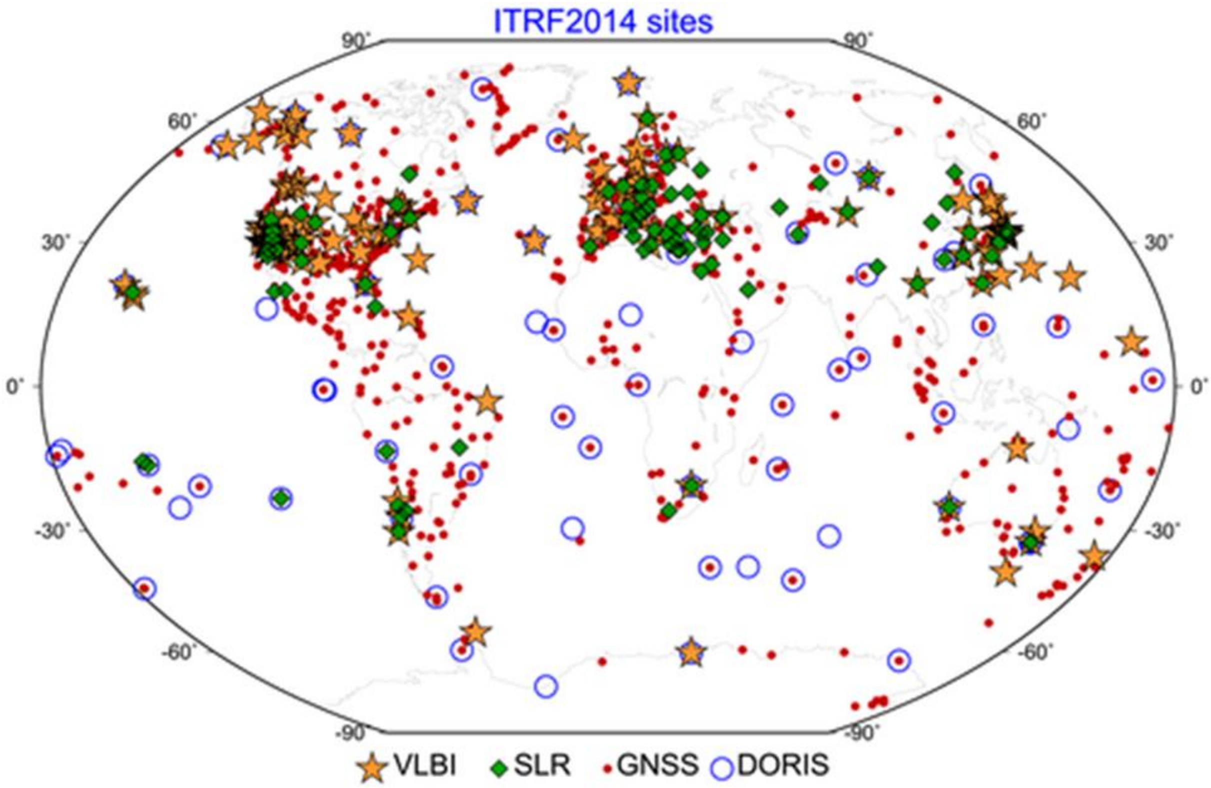

Figure 2 – Global station distribution and geodetic techniques used to realise ITRF 2014

[1] Altamimi, Z., P. Rebischung, L. Métivier, and X. Collilieux (2016) *"ITRF2014: A new release of the International Terrestrial Reference Frame modeling nonlinear station motions".* Journal of Geophysical Research Solid Earth 121:6109–6131, doi:10.1002/2016JB013098.

1.2.3 Why different reference frames?

While a geodetic reference *system* can be defined once and thereafter remains valid, the same is not true of a geodetic reference *frame*. The fact that the earth's crust is dynamic means that points on the surface of the earth move as tectonic plate motion and other sources of crustal movement (e.g. earthquakes) deform the crust (e.g. Figure 3). Practically, and relative to an earth-centred, earth-fixed (ECEF) reference frame such as ITRF2014, this means that coordinates of points on the earth's surface are constantly changing. It is for this reason that IERS publishes station velocities along with stations coordinates at a specified epoch for each realisation of the frame. These velocities allow station coordinates to be moved through time within the reference frame, so long as the assumption of linear motion remains valid. However, the longer the time between the reference epoch and the computation date, the less accurate the propagated coordinates will be. Additionally, the progress of time allows more stations to be incorporated into the reference frame solution, more and improved observations to be included, longer time series to be considered and more sophisticated error modelling and computational methods to be employed. Consequently, each realisation of ITRF is more robust and more accurate than its predecessor. To illustrate, ITRF2014 is the eighth realisation of the ITRS since ITRF92 was published. Planning and data processing are underway for the next realisation – ITRF2020 (http://itrf.ensg.ign.fr/ITRF_solutions/index.php).

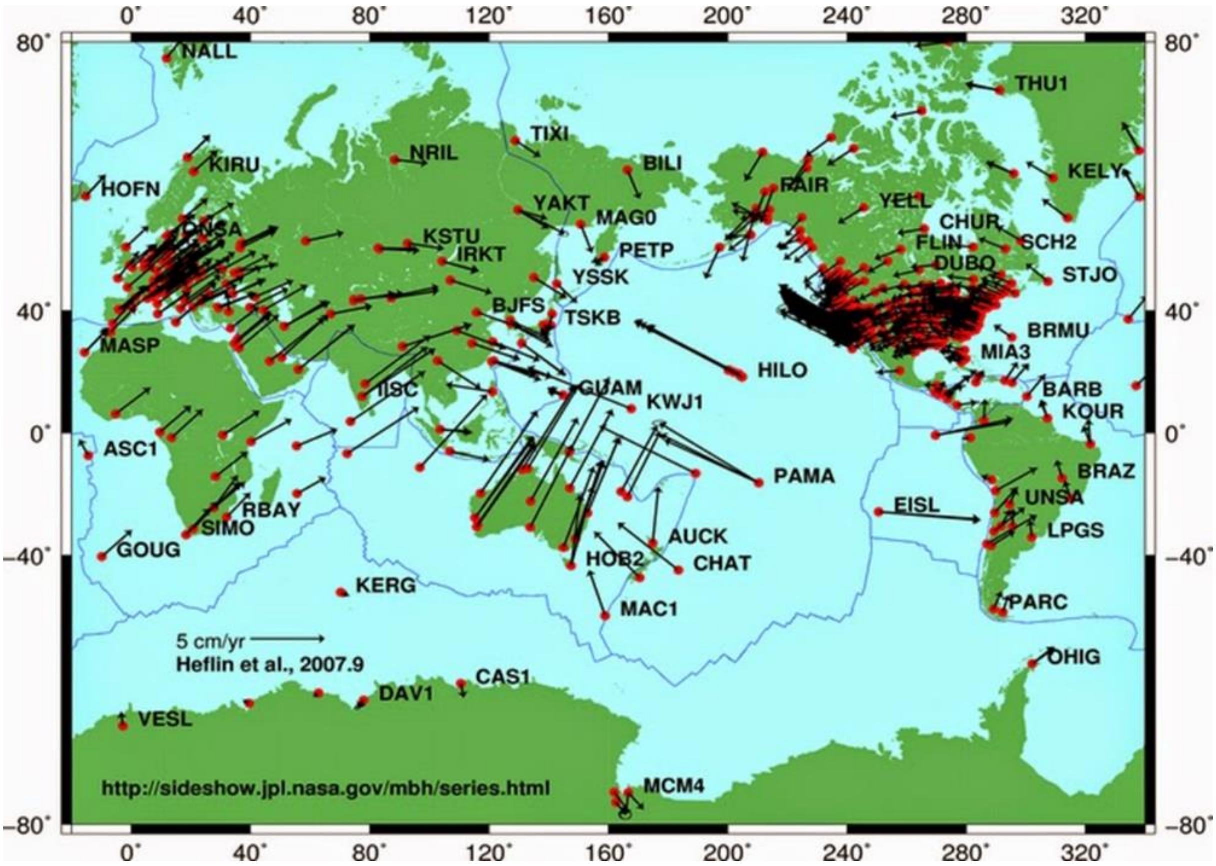

Figure 3 – Global map of tectonic plate motion

2. Coordinate Conversion & Transformation

The nomenclature around the processes of moving coordinates from one frame of reference to another and between specified epochs has not been standardised. In this paper an attempt is made to introduce and consistently use unique terms for the various elements of the overall transformation process in order to minimise confusion. Table 3 provides a summary of the processes of coordinate “conversion” and the various categories of “transformation”.

Process	Purpose	Change Form	Change Epoch	Change Reference Frame
Conversion (Section 2.1)	Change of coordinate form: $(E_P, N_P, h_P) \leftrightarrow (\phi_P, \lambda_P, h_P) \leftrightarrow (X_P, Y_P, Z_P)$	☑	☒	☒
Temporal transformation (Section 2.2-2.4)	Change of coordinate epoch: (X_P, Y_P, Z_P) @ $t_j \leftrightarrow (X_P, Y_P, Z_P)$ @ t_i	☒	☑	☒
Spatial transformation (Section 2.5-2.7)	Change of coordinate reference frame: $(X_P, Y_P, Z_P)_{REF1}$ @ $t_i \leftrightarrow (X_P, Y_P, Z_P)_{REF2}$ @ t_i	☒	☒	☑
Combined transformation (Section 2.8)	Change of both reference frame and epoch: $(X_P, Y_P, Z_P)_{REF1}$ @ $t_j \leftrightarrow (X_P, Y_P, Z_P)_{REF2}$ @ t_i	☒	☑	☑

Table 3 – Purpose and attributes of coordinate conversion and transformation

2.1 Coordinate conversion

Coordinate conversion is the process of changing from one form of coordinates to another, while staying in the same reference frame at the same epoch. The process of coordinate conversion is illustrated in Figure 4.

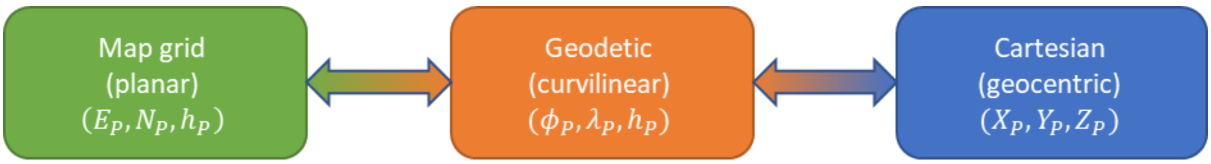

Figure 4 – The sequence of coordinate conversion

Standard formulae exist to derive map grid coordinates from geodetic coordinates and vice versa, but these formulae depend on the type of map projection being used (e.g. UTM). Formulae for this process can be readily sourced from the Internet or standard geodetic text books. Cartesian coordinates can likewise be derived from geodetic coordinates and vice versa. Formulae for this process, along with worked examples are provided in Appendix A. It is not possible to derive map grid coordinates directly from cartesian coordinates nor vice versa, these conversions must always be via the geodetic system.

2.2 Linear temporal transformation

Temporal transformation is the process of moving coordinates from one epoch to another, while remaining in the same reference frame. Typically, cartesian coordinates are used in temporal transformation calculations. Figure 5 shows the location of a point P changing linearly over time. Most commonly, such movement is due to linear tectonic plate motion. Non-linear movement through time can also arise and will be discussed in more detail in Section 2.3.

In the example shown in Figure 5, the position of P at the reference epoch t_0 is denoted by P_0, and expressed in cartesian form as (X_P, Y_P, Z_P) @ t_0. Over time, P will move relative to the reference frame so that at any later time t_i the position of P will be given by (X_P, Y_P, Z_P) @ t_i. The quantities $\Delta_{i,j}$ denote physical point displacement vectors between epochs i and j. The coloured arrows depict these displacements overlying the long-term linear trend of the tectonic motion, which can, in this case, be modelled by a known/given point velocity. Formulae for linear transformation of coordinates through time are given in Appendix B, along with worked examples.

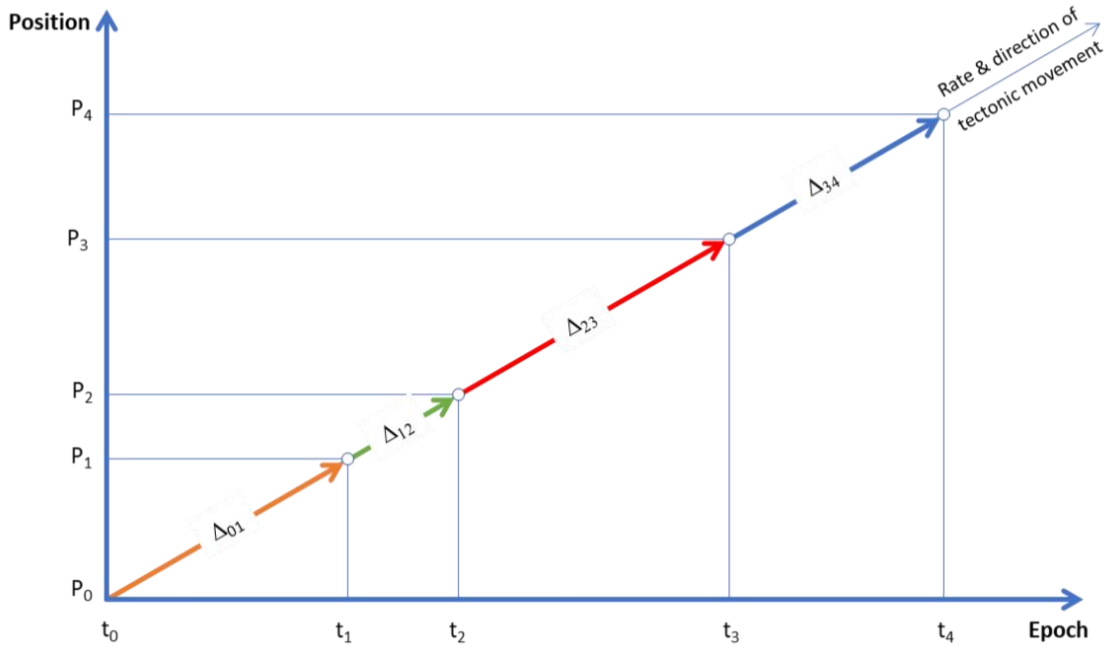

Figure 5 – Changing position over time as a result of linear tectonic plate movement

The assumption that underpins the description above is that tectonic plate motion is essentially linear and able to be modelled either by point velocities or rotation rates (see Equations B.1 and B.2 respectively). This assumption is generally valid (at least for relatively short periods of time). In Australia, for example, where tectonic motion is highly stable, empirical data suggests the assumption of linearity can be adopted for approximately 20 years. However, there are areas where the assumption fails much more quickly, most typically in regions of earthquake activity, such as Japan, New Zealand and along the San Andreas fault of the US[2]. In such cases, point velocities must be augmented with a model to account for the non-linear deformation. This case and how to deal with it will be discussed in more detail in Section 2.3

[2] For example, SmartNet North America update station coordinates twice per year in areas of "high dynamic" activity (e.g. California) to support NRTK positioning. While the bi-annual updates are typically only in the order of a few centimetres, such movements are unpredictable and cumulative and so must be accounted for to satisfy the accuracy requirements of users. (See https://support.smartnetna.com/hc/en-us/articles/115014592788-Fall-Update-Adjustment-Launch-11-2017)

2.3 Non-linear temporal transformation

While linear motion resulting from tectonic plate movement is widespread, well understood and routinely accounted for, the existence and modelling of non-linear crustal motion (referred to hereafter as *deformation*) is less common and more challenging to detect and model. Deformation may be due to natural causes such as post-seismic relaxation or induced by human activity (e.g. de-watering, oil and gas extraction, underground mining). Where relevant, a rigorous approach to dealing with time-dependent coordinates will require the creation of regional deformation models to capture non-linear crustal motion to augment linear tectonic plate motion. Work is beginning in Australia to address this issue, but an operational solution is some way off.

As in the case of linear plate motion, time-series data is needed to measure and model non-linear crustal motion. A much higher spatial resolution and temporal frequency of sampling are required because of the localised nature of the movement. Typical CORS station density of tens to hundreds of kilometres is not sufficient. For this reason, remote sensing techniques such as InSAR and LiDAR are being evaluated as the primary sources of information to measure non-linear, regional crustal deformation. Such motion must subsequently be modelled to augment linear plate motion and provide detail of *total* crustal motion on a continuous basis. Typically, a grid-based approach will be used to represent the on-going deformation between measurement epochs. A new grid can be created each time the crustal motion is measured, so that the total deformation can be determined by summation of the individual grids.

2.4 Total temporal transformation

A process for a full temporal transformation, accounting for both linear and non-linear components is presented in Table 4. The linear component (L_{0i}) represents the tectonic motion from the reference epoch (t_0) to the current epoch (t_i) as computed by one of the methods described in Appendix B. Periodically, local deformation is measured and represented by (m_i). These measured deformations are interpolated onto a grid (G_{ij}) to allow a continuous representation of the deformation field over the area of interest between adjacent measurement epochs ($t_i$and t_j). Adding these epoch-wise deformation grids allows the cumulative, non-linear deformation from the reference epoch to any other epoch (t_i) to be determined (G_{0i}). The total crustal motion at any epoch is then the sum of the linear motion and the non-linear deformation (L_{0i}+G_{0i}).

Epoch	t_0	t_1	t_2	t_3
Linear motion model	0	L_{01}	L_{02}	L_{03}
Measured deformation	m_0	m_1	m_2	m_3
Representation by grids		G_{01}	G_{12}	G_{23}
Cumulative deformation		G_{01}	$G_{02} = G_{01} + G_{12}$	$G_{03} = G_{01} + G_{12} + G_{23}$
Total crustal motion		L_{01}+G_{01}	L_{02}+G_{02}	L_{03}+G_{03}

Table 4 – A process for accounting for linear and non-linear crustal motion

An important issue in the context of this paper is the question of how the Swift solution responds to and accounts for *total* crustal motion. This question is addressed in Appendix E. Equally important is the impact of such motion on the user, which is addressed in Appendix F. Finally, Appendix G describes a process for dealing with the case of temporal misalignment between the Swift solution and the user's map.

2.5 Spatial conformal transformation

Spatial transformation is the process of moving coordinates from one frame of reference to another, at a common epoch. For example a user of the Swift solution may need to transform from ITRF2014 to ITRF2008, at the common epoch of 2010.0. A *spatial-conformal* transformation does not involve a time component and retains the *shape* of the transformed object. The parameters for such a transformation are known and can be applied without consideration of the reference epoch. The usual approach is to use a 7-parameter ("Similarity", "Conformal" or "Helmert") transformation. Appendix C provides the technical details for the 7-parameter transformation, In particular, Table C.1 defines the transformation parameters while Equations (C.1) and (C.2) present two alternative forms of the transformation equation. Worked examples for each equation are also provided.

2.6 Spatial non-conformal transformation

Historical (legacy) reference frames can contain significant spatial distortions. Typically, such distortions arise on account of measurement and computational errors occurring during the realisation of the frame. For example, in Australia, AGD66 was realised based on sparse, sinuous geodetic traversing, disparate measurements acquired using less accurate terrestrial surveying equipment and non-rigorous computational models and procedures. The result was a solution that was inferior to the newer GDA94 frame. To transform coordinates from AGD66 to GDA94 requires not just a conformal transformation, but a complex, spatially variable correction model that allows for the removal of distortions from the historic AGD66 coordinates.

Generally, a distortion model is represented by a grid of 2D distortion components $(\Delta\phi, \Delta\lambda)$, derived from spatial analysis and modelling of coordinates in both old and new reference frames. Such grids are typically provided in the *NTv2* format and commonly include both the conformal and the distortion components at each grid node, making the full transformation (conformal + distortion) possible in a single computational step.

2.7 Total spatial transformation

As mentioned in the preceding section, in the case of a spatial transformation comprising a conformal and a non-conformal (distortion) component, it is common practice to combine both components and to represent them in gridded form for the purposes of practical application. On occasions, there may be the need to spatially transform from one frame of reference to another, via an intermediate frame. For example, in Australia the transformation from AGD66 to GDA2020 can only be done via GDA94, simply because of the available transformation parameters and grids. Thus a two-stage transformation is required, as illustrated in Figure 6 for the AGD66 to GDA2020 case. Unlike the total temporal transformation (Section 2.4), it is not correct to simply add the distortion grids, they must be applied in sequence as they are reference frame specific.

Figure 6 – Spatial transformation in stages is sometimes required

As time goes by, the need for distortion models will diminish due to the phasing out of historical reference frames. For example, the distortions between AGD66 and GDA94 were in excess of ±1 m, whereas the distortions between GDA94 ad GDA2020 are typically less than ±0.1 m. As modern reference frames replace legacy systems around the world and the distortions characteristic of these older reference frames are eliminated, the need for distortion models will reduce.

The question of whether Swift needs to accommodate distortion models will need to be answered on a case-by-case basis and will depend on the magnitude of the distortions compared to the accuracy of the Swift solution, the accuracy requirements of the user and, most importantly, whether the digital map product has been influenced by the distortions present in the survey control network from which the distortion model has been derived.

2.8 Combined temporal and spatial transformation

With the advent of time-dependent reference frames, it is increasingly common to both change from one reference frame to another and to simultaneously propagate from one epoch to another. For example, it may be necessary to move coordinates from ITRF08@2005.0 to ITRF2014@2010.0. We refer to this process as a combined transformation and for such purposes, a 14-parameter transformation model is typically used.

The 14-parameter transformation adds time-dependence to the 7-parameter transformation introduced in Section 2.5 and detailed in Appendix C. This is achieved by including a rate of change for each of the seven "static" parameters described in Table C.1. The inclusion of these rates of change allows the prescribed translation, rotation and scale parameters to be modified to reflect the time difference between the two reference frame realisations. It should be noted that the 14-parameter transformation model used in geodesy presumes linear crustal motion. It does not allow for the case of non-linear crustal deformation. As described in Section 2.3, non-linear crustal motion cannot be accounted for in a single step transformation unless a gridded approach is used. Rather, its impact must be incorporated separately and subsequent to accounting for the linear component.

Appendix D provides technical details for the 14-parameter transformation, In particular, Equations (D.1) and (D.2) show alternative forms of the transformation equation, while Table D.1 describes the additional rate of change parameters.

With the release of ITRF2014, the IERS has, for the first time, introduced modelling of non-linear station motions, including seasonal signals of station positions and post-seismic deformation for sites subject to major earthquakes. Full details of the computation and application of the non-linear models can be found in Altamimi et al. (2016)[3]. In practice, such models augment the linear point velocity model and the 14-parameter transformation approach. The need to apply such models to the Swift solution is unlikely as they are site specific, applying only to stations used in the ITRF2014 solution. At this stage it is not possible to extrapolate/interpolate the models for use beyond the ITRF sites, as the sparse ITRF2014 station density does not allow such models to be readily developed. Rather, the deformation modelling approach described in Section 2.3 will be the most practical way of accommodating non-linear crustal motion in the Swift solution.

[3] Altamimi, Z., P. Rebischung, L. Métivier, and C. Xavier (2016), ITRF2014: A new release of the International Terrestrial Reference Frame modelling nonlinear station motions, J. Geophys. Res. Solid Earth, 121, 6109–6131, doi:10.1002/2016JB013098.

2.9 The complete transformation picture

Table 5 provides a summary of the full complement of transformation components. The picture is a complex one and care is needed to ensure reliability, repeatability and traceability of the transformation results.

Given the range of formulae required to effect the total transformation and the possibility of multiple layers of transformation, there may be computational benefits in gridding the individual transformation components and adding those grids as required to deliver the total transformation solution. Conceptually, this approach is illustrated in Figure 7.

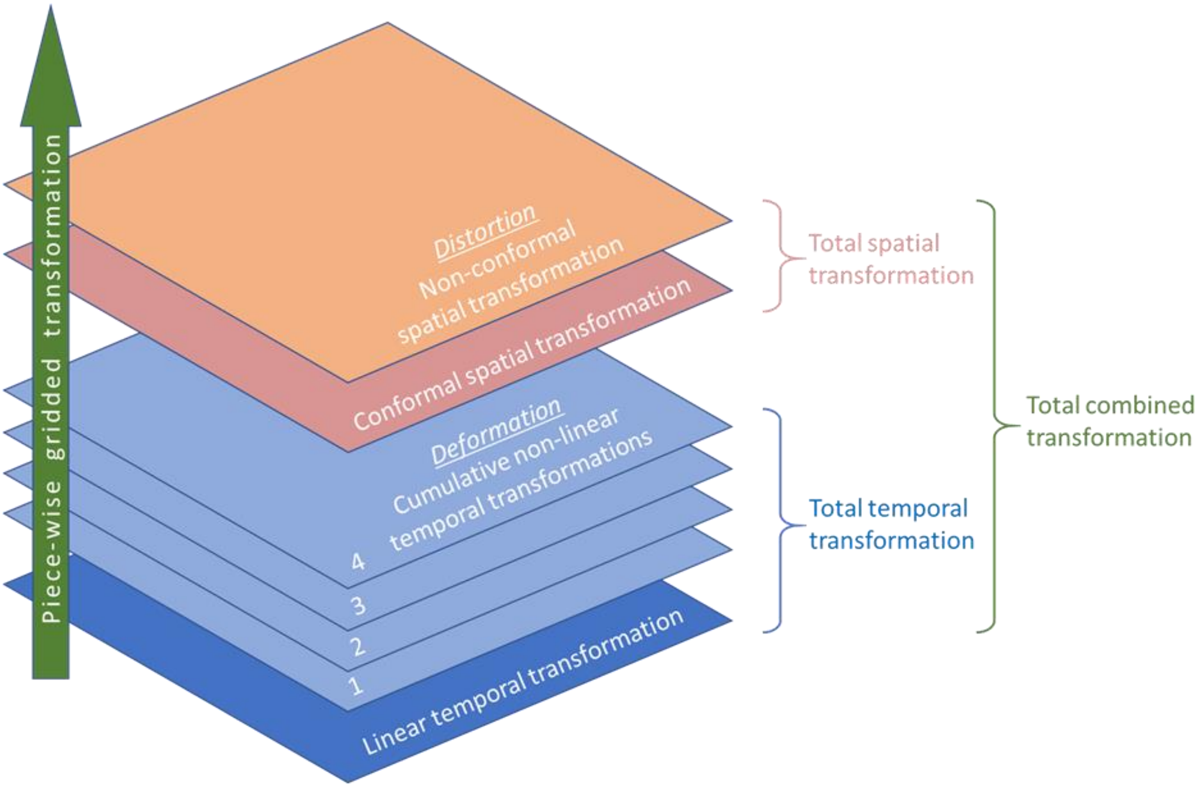

Figure 7 – The concept of a fully gridded transformation solution, with each component represented in gridded form and the total solution computed by sequential addition of the individual grids

TEMPORAL TRANSFORMATION (between epochs in a single reference frame)		SPATIAL TRANSFORMATION (between reference frames)		COMBINED TRANSFORMATION (between reference frames at different epochs)	
Linear (Section 2.2)	Rotation rates model (RRM) Equation (B.2)	**Conformal (Section 2.5)**	7 parameter model Equation (C.1) or (C.2)	**Linear + Conformal (Section 2.8)**	14 parameter model Equation (D.1) or (D.2)
	Point velocity model (PVM) Equation (B.1)				PVM + 7 parameter model Equation (B.1) + Equation (C.1) or (C.2)
Non-linear (deformation) (Section 2.3)	Σ Deformation grids	**Non-conformal (distortion) (Section 2.6)**	Distortion grid	**Non-linear + Non-conformal**	Σ Deformation grids + Distortion grid
Total temporal (Section 2.4)	Linear + Non-linear (deformation)	**Total spatial (Section 2.7)**	Conformal + Non-conformal (distortion)	**Total transformation**	Linear + Conformal + Non-linear (deformation) + Non-conformal (distortion)

<u>Table 5</u> – The full array of transformation components and their various combination and modelling options, leading to the "total transformation" solution

3. Practical Considerations

The aim of this section is to present a process for the practical application of the transformation options described above. Swift Navigation will routinely face the need to transform its solution into the reference frame and epoch required by the user. The question to be answered is: How to determine and apply the most appropriate transformation strategy? Before addressing this question, it is vital to be clear on the definitions and implications of *accuracy* and *precision*.

3.1 Accuracy

Accuracy is defined as *"closeness to the truth"* and is a fundamental consideration when endeavouring to align the Swift solution to the user's digital map.

<u>In the Swift solution</u>, accuracy will be represented by the delivery of coordinates reliably and consistently linked to the adopted reference frame. <u>For the user's map data</u>, accuracy will be a function of how the underlying spatial information has been acquired, processed and generalised for mapping and navigation purposes. Fundamentally though, the accuracy of the map data will be measured by its alignment to the user's reference frame.

The purpose of transformation is to bring the Swift solution into spatial agreement with the user's reference frame. This requires:

1. The reference frames and reference epochs for both data sets to be known

2. The respective reference frames to be reliably realised in the two data sets
3. The spatial and temporal transformation parameters to be known with confidence

If these conditions are satisfied, it will be possible to transform the Swift solution to *accurately* overlay the user's digital map. In practice however, fully satisfying the above conditions will be challenging and data alignment problems will routinely appear, typically in the form of offsets between the (transformed) Swift solution and the user's map data. The problem is conceptually illustrated in Figure 8, where the blue line shows a road centreline in the user's map data and the green line shows the transformed Swift solution consistently offset from the corresponding map data (as indicated by the red arrows).

Figure 8 – Inaccuracy manifests in the form of a consistent offset between the two data sets

The accuracy of the digital map data and of the Swift solution may vary spatially within each data set (e.g. data may be more accurate in urban areas). In such a case, the offset vectors will not be strictly consistent, tending to mask the nature of the problem and making it more difficult to resolve.

Notwithstanding this potentially complex and variable accuracy picture, user requirements will ultimately dictate Swift's response. For example, if the user needs agreement between the Swift solution and the incumbent digital map at the 0.1 m level, the transformation model, coupled with the inherent accuracy of the two data sets, must be capable of delivering this. If the required accuracy cannot be consistently satisfied, the quality of one or both of the data products must be upgraded and/or the transformation model refined and improved (see the three requirements for spatial agreement listed above).

3.1.1 Testing achievable accuracy

Achievable accuracy can be quantified by calculating coordinate differences between matching points from the digital map data and the transformed Swift solution. If the mean difference is non-zero, there is an accuracy problem, the cause of which must be investigated. Alternatively, the RMS and the standard deviation of the coordinate differences can be computed and compared. If the RMS is larger than the standard deviation, again this indicates the existence of a non-zero mean and therefore a shortcoming in aligning the two data sets.

Possible causes of misalignment include:

- The Swift solution is not accurately aligned to its stated reference frame
- The user's data is not accurately aligned to its stated reference frame
- The transformation parameters are not reliable

- The transformation parameters are wrong
- The transformation model (temporal and/or spatial) is flawed

3.2 Precision

Precision is defined as "statistical repeatability" and is quantified by the standard deviation of a data sample. For example, the precision of the Swift solution could be expressed as ±0.1 m, which means that about 68% of repeated solutions will fall within a circle of that radius. It is important to understand that precision is distinct from accuracy and says nothing about how well the data aligns to datum. In other words, the data can be precise (repeatable) but inaccurate (biased and not close to the truth). In aligning the Swift solution to the user's map, imprecision will manifest as random (and generally small) positional differences as shown in Figure 9. For the purposes of this example, both solutions are deemed to be *accurate* in their own right and have been aligned through an appropriate transformation process.

It is important to note that the transformation process that aligns two solutions *will not* of itself introduce imprecision, though it can introduce inaccuracy.

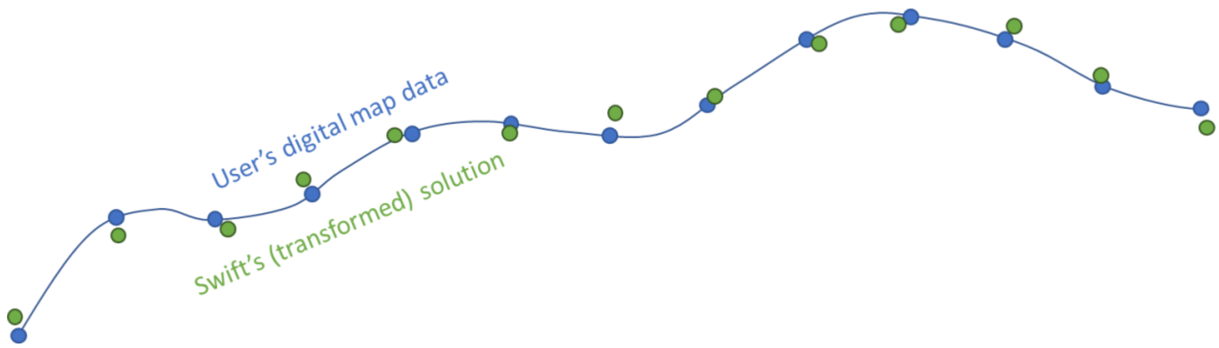

Figure 9 – Imprecision manifests in the form of random offsets at individual points

3.3 Determining the most appropriate transformation strategy

The following are essential pieces of information needed to answer the question posed at the beginning of this section: How can Swift determine and apply the most appropriate transformation strategy to align with the user's map?

1. Reference frame and epoch of the Swift solution $(REF_{swift}, EPOCH_{swift})$
   At present this is presumed to be ITRF2014@2010.0, but confirmation is required.
2. Reference frame and epoch of the user's data $(REF_{user}, EPOCH_{user})$
   It is assumed that the user will want to align the Swift solution to an existing digital map product or other spatial data set. Most likely, the user's data will not be in the same reference frame and epoch as the Swift solution, thus requiring both spatial and temporal transformation.
3. Required accuracy $(\sigma_{required})$
   The user will require a certain level of accuracy for the application at hand. While the complete accuracy picture is made up of several inputs, users will generally characterise "required accuracy" as the degree of alignment between the Swift solution and the map. The important question is whether the required accuracy can be achieved given the impact of the contributing variables.

Based on these three critical pieces of information, the following rules of thumb can be applied in determining the most appropriate transformation approach. These rules are also summarised in graphical form in Figure 10.

Spatial transformation

1. $REF_{user} = REF_{swift}$ — spatial transformation is not required
2. $REF_{user} \neq REF_{swift}$ — spatial transformation is required, so determine:
   - $\sigma_{required} \geq 1.0\ m$ — use conformal transformation only
   - $1.0\ m < \sigma_{required} \leq 0.1\ m$ — add distortion model (if available and relevant)
   - $\sigma_{required} < 0.1\ m$ — add distortion model (if available and relevant)

Temporal transformation

1. $EPOCH_{user} = EPOCH_{swift}$ — temporal transformation is not required
2. $EPOCH_{user} \neq EPOCH_{swift}$ — temporal transformation is required, so determine:
   - $\sigma_{required} \geq 1.0\ m$ — use linear (rate of change) model only
   - $1.0\ m < \sigma_{required} \leq 0.1\ m$ — use linear (rate of change) model only
   - $\sigma_{required} < 0.1\ m$ — add non-linear (deformation) model (if available)

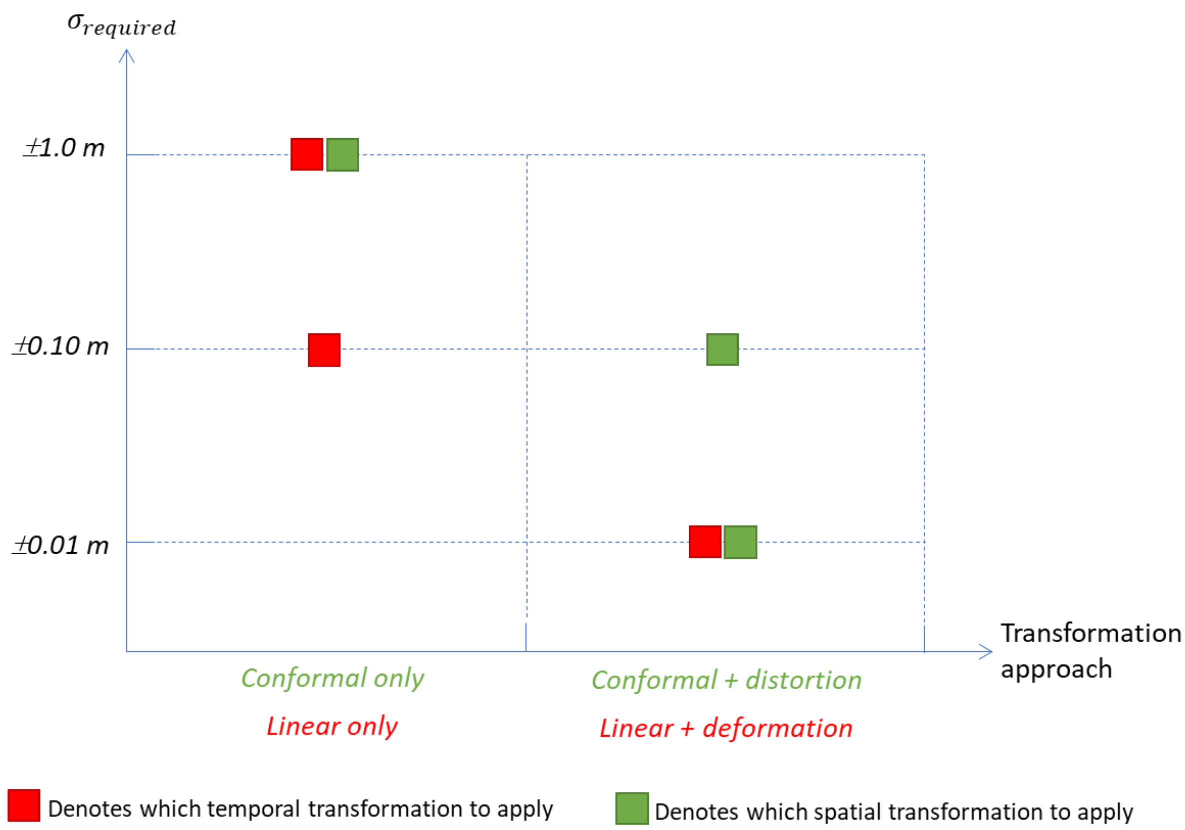

Figure 10 – Logic rules for combining spatial and temporal transformation models depending on the user's required accuracy

Appendix A – Coordinate Conversion

Geodetic to cartesian conversion $(\phi_P, \lambda_P, h_P) \rightarrow (X_P, Y_P, Z_P)$

Preliminary quantities

a – semi-major axis of the reference ellipsoid
b – semi-minor axis of the reference ellipsoid
e^2 – first eccentricity squared
e'^2 – second eccentricity squared
ν - radius of curvature in the prime vertical

$$\nu = \frac{a}{(1-e^2 \sin^2 \phi)^{1/2}} \qquad \text{...(A.1)}$$

Conversion formulae

$$X = (\nu + h) \cos \phi \cos \lambda \qquad \text{...(A.2)}$$

$$Y = (\nu + h) \cos \phi \sin \lambda \qquad \text{...(A.3)}$$

$$Z = ((\nu(1 - e^2) + h) \sin \phi \qquad \text{...(A.4)}$$

Cartesian to geodetic conversion $(X_P, Y_P, Z_P) \rightarrow (\phi_P, \lambda_P, h_P)$

Preliminary quantities

$$p = (X^2 + Y^2)^{1/2} \qquad \text{...(A.5)}$$
$$\tan \beta = \frac{aZ}{bp} \qquad \text{...(A.6)}$$

Conversion Formulae

$$\phi = \tan^{-1}(\frac{Z + e\prime^2 b \sin^3 \beta}{p - ae^2 \cos^3 \beta}) \qquad \text{...(A.7)}$$

$$\lambda = \tan^{-1}(\frac{Y}{X}) \qquad \text{...(A.8)}$$

$$h = p \cos \phi + Z \sin \phi - \frac{a^2}{\nu} \qquad \text{...(A.9)}$$

The geodetic to cartesian and the cartesian to geodetic conversion formulae are rigorous and do not result in a loss of positional accuracy through the coordinate conversion process. Sample calculations are provided on the following page for both conversions described above.

Worked examples

Geographic to Cartesian

Enter ellipsoid parameters :

Ellipsoid	GRS80
Semi-major axis (a)	6378137
Inverse flattening (1/f)	298.257222101

Enter latitude, longitude and height :

	°	′	″
Latitude	37	46	15.120000
Longitude	-122	24	11.970000
Ellipsoidal height	10.0000	m	

Solution :

X-coordinate	-2,705,130.4295	m
Y-coordinate	-4,262,056.7605	m
Z-coordinate	3,885,377.7577	m

Cartesian to Geographic

Enter ellipsoidal parameters:

Ellipsoid	GRS80
Semi-major axis (a)	6378137
Inverse flattening (1/f)	298.257222101

Enter cartesian coordinates:

X-coordinate	-4,130,791.3127	m
Y-coordinate	2,899,592.9037	m
Z-coordinate	-3,888,881.7742	m

Solution:

	°	′	″
Latitude	-37	48	8.12340
Longitude	144	55	59.56780
Ellipsoidal height	1234.5678	m	

Appendix B – Linear Temporal Transformation

Method 1 – Point Velocity Model (PVM)

In most, though not all, regions of the world, tectonic plate motion is linear with respect to time. Even though the rate of motion can be several centimetres per year, modelling linear plate motion by means of point velocities is robust and reliable.

On the assumption of consistent, linear movement, a point velocity model provides a simple way of propagating coordinates through time. If the station coordinates (X, Y, Z) and point velocities (V_X, V_Y, V_Z) are known at epoch t_i, the coordinates at a later epoch t_j can be calculated using the following equation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{t_j} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{t_i} + (t_j - t_i) \begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix}_{t_i} \qquad \text{...(B.1)}$$

Method 2 – Rotation rates model (RRM)

Alternatively, for the purposes of coordinate propagation through time, a reduced/simplified form of the standard 14-parameter geodetic transformation model (see Appendix D) can be applied. This is particularly useful in the case that point velocities are not known and therefore the point velocity model cannot be used.

The simplification of the 14-parameter model comes from the fact that all parameters, except the rotation rates, are zero since the reference frame is not changing. Thus the translations, rotations, scale, translation rates and scale rate are all eliminated from the calculation.

The reduced form of the 14-parameter model used for coordinate propagation is shown in Equation (B.2).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{t_j} = \begin{bmatrix} 1 & \dot{r_z}(t_i - t_j) & -\dot{r_y}(t_i - t_j) \\ -\dot{r_z}(t_i - t_j) & 1 & \dot{r_x}(t_i - t_j) \\ \dot{r_y}(t_i - t_j) & -\dot{r_x}(t_i - t_j) & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{t_i} \qquad \text{...(B.2)}$$

Note that the time difference in Equation (B.2) has the opposite sense to that used in Equation (B.1).

Worked examples

Point Velocity Model								
From							**To**	
Cartesian coordinates (metres)		Velocities (m/year)		Epochs (yyyy.yyyyy)			Cartesian coordinates (metres)	
X	-3,753,473.1960	V_X	-0.0421	t_i	2020.000		X	-3,753,473.2381
Y	3,912,741.0310	V_Y	0.0024	t_j	2021.000		Y	3,912,741.0334
Z	-3,347,959.6998	V_Z	0.0501	Δt	1.000		Z	-3,347,959.6497

Rotation Rates Model								
From							**To**	
Cartesian coordinates (metres)		Rotation Rates ("/year)		Epochs (yyyy.yyyyy)			Cartesian coordinates (metres)	
X	-3,753,473.1960	$\dot{r}_X$	0.00150379	t_i	2020.000		X	-3,753,473.2381
Y	3,912,741.0310	$\dot{r}_Y$	0.00118346	t_j	2021.000		Y	3,912,741.0334
Z	-3,347,959.6998	$\dot{r}_Z$	0.00120716	Δt	-1.000		Z	-3,347,959.6497

The examples above are drawn from an Australian case where the rotation rates have been derived from a series of points with known and very stable velocities, spread uniformly across the Australian tectonic plate. The uniformity of the velocities gives rise to the two approaches yielding identical results at the sub-millimetre level. In the case of less uniform velocities across a large geographic area, differences between the two approaches would be expected.

Note also the different sign of Δt in each case, due to the way the relevant equations (B.1) and (B.2) are formulated.

Appendix C – Spatial Transformation

Parameter definitions

The seven parameters that underpin a conformal transformation are defined and presented in Table C.1.

Parameters	Description	Units
(t_x, t_y, t_z)	Three *translations* between the origins of the two reference frames	Metres (m)
(r_x, r_y, r_z)	Three *rotation* angles to align the cartesian axes	Radians (rad)
s	One *scale* parameter to account for differences in linear scale	Parts-per-million (ppm)

Table C.1 – The seven parameters of a standard conformal transformation

The Helmert transformation model

On the basis of the parameters defined in Table C.1 and given the cartesian coordinates of a point in Reference Frame 1 (REF_1), the transformed coordinates in Reference Frame 2 (REF_2), can be derived from the 3D Helmert transformation shown in Equation (C.1). This form of the 3D conformal transformation is the one most commonly used in geodetic applications and is premised on the assumption that the rotation angles are small (less than 10”).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF2} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + (1+s)\begin{bmatrix} 1 & r_z & -r_y \\ -r_z & 1 & r_x \\ r_y & -r_x & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1} \qquad \text{...(C.1)}$$

The IERS transformation model

The IERS formulates the above transformation equation in a slightly different, but effectively equivalent form, as shown in Equation (C.2):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF2} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + s\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1} + \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1} \qquad \text{...(C.2)}$$

A note on sign conventions

The only material difference between equations (C.1) and (C.2) is that the rotation angles are given opposite signs. This arises due to the existence of two *conventions* for the specification of these rotations. In both conventions a positive rotation occurs in an anti-clockwise direction, when looking along the positive axis towards the origin. However, the IERS convention deems the rotations to be of the points relative to the axes while the alternative approach deems the rotations to be of the axes relative to the points. Errors will occur if a user applies parameters that follow the IERS convention in Equation (C.1) or if rotations based on non-IERS convention are applied in Equation (C.2).

Angles following the IERS convention can be used in Equation (C.1) only if their signs are changed. Similarly, angles provided in the non-IERS convention can be used in Equation (C.2) following sign reversal. Finally, identical results will be achieved via either convention and either equation when applied *correctly*.

The sign convention being followed (and therefore the equation to be used) is not always obvious to the user, especially when encoded in software. In this case, sample data, where transformation results are also provided, can be used to confirm which convention is being employed. The worked examples provided on the following page may assist in answering this question.

Worked examples

Helmert Transformation

Reference frames				
From (REF1)	GDA94			
To (REF2)	GDA2020			
Input coordinates				
X(GDA94)	-4,130,791.313	m		
Y(GDA94)	2,899,592.904	m		
Z(GDA94)	-3,888,881.774	m		
Transformation parameters				
Translation X (t_x)	0.06155	m		
Translation Y (t_y)	-0.01087	m		
Translation Z (t_z)	-0.04019	m		
Scale (s)	-0.009994	ppm		
Rotation X (r_x)	-0.0394924	"	-1.91465E-07	rad
Rotation Y (r_y)	-0.0327221	"	-1.58641E-07	rad
Rotation Z (r_z)	-0.0328979	"	-1.59494E-07	rad
Rotation matrix				
	1.000	-1.5949E-07	1.5864E-07	
	1.5949E-07	1.000	-1.9146E-07	
	-1.5864E-07	1.9146E-07	1.000	
Transformed coordinates				
X(GDA2020)	-4,130,792.289	m		
Y(GDA2020)	2,899,592.950	m		
Z(GDA2020)	-3,888,880.565	m		

IERS Transformation

Reference frames				
From (REF1)	GDA94			
To (REF2)	GDA2020			
Input coordinates				
X(GDA94)	-4,130,791.313	m		
Y(GDA94)	2,899,592.904	m		
Z(GDA94)	-3,888,881.774	m		
Transformation parameters				
Translation X (t_x)	0.06155	m		
Translation Y (t_y)	-0.01087	m		
Translation Z (t_z)	-0.04019	m		
Scale (s)	-0.009994	ppm		
Rotation X (r_x)	0.0394924	"	1.91465E-07	rad
Rotation Y (r_y)	0.0327221	"	1.58641E-07	rad
Rotation Z (r_z)	0.0328979	"	1.59494E-07	rad
Rotation matrix				
	0.000	-1.5949E-07	1.5864E-07	
	1.5949E-07	0.000	-1.9146E-07	
	-1.5864E-07	1.9146E-07	0.000	
Transformed coordinates				
X(GDA2020)	-4,130,792.289	m		
Y(GDA2020)	2,899,592.950	m		
Z(GDA2020)	-3,888,880.565	m		

Appendix D – Simultaneous Transformation

Parameter definitions

In addition to the seven parameters defined in Appendix C for a spatial transformation, the remaining parameters required for the 14-parameter transformation are described in Table D.1. The 14-parameter transformation allows a simultaneous change of epoch and reference frame, on the assumption of linear crustal motion.

Parameters	Description	Units
$(\dot{t}_x, \dot{t}_y, \dot{t}_z)$	Rates of change of the three *translations* parameters	Metres per year (m/yr)
$(\dot{r}_x, \dot{r}_y, \dot{r}_z)$	Rates of change of the three *rotation* angles	Radians/year (rad/yr)
$\dot{s}$	Rate of change of the *scale* parameter	Parts-per-million/year (ppm/yr)
t_i *and* t_j	Dates of the REF1 and REF2 epochs respectively	Decimal years (yyyy.yyy)

Table D.1 – The additional parameters needed for the 14-parameter transformation

The 14-parameter geodetic transformation model

On the basis of the parameters defined in Tables C.1 and D.1 and given the cartesian coordinates in Reference Frame 1 (REF_1) at Epoch t_i, the transformed coordinates in Reference Frame 2 (REF_2) at Epoch t_j, can be derived from Equation (D.1):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF2\,@\,t_j} = \begin{bmatrix} t_x + \dot{t}_x(t_i - t_j) \\ t_y + \dot{t}_y(t_i - t_j) \\ t_z + \dot{t}_z(t_i - t_j) \end{bmatrix} + \left(1 + s + \dot{s}(t_i - t_j)\right)$$

$$\begin{bmatrix} 1 & r_z + \dot{r}_z(t_i - t_j) & -r_y - \dot{r}_y(t_i - t_j) \\ -r_z - \dot{r}_z(t_i - t_j) & 1 & r_x + \dot{r}_x(t_i - t_j) \\ r_y + \dot{r}_y(t_i - t_j) & -r_x - \dot{r}_x(t_i - t_j) & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1\,@\,t_i} \quad \text{...(D.1)}$$

The 14-parameter IERS transformation model

Once again, the IERS formulates the above 14-parameter transformation equation in a slightly different but effectively equivalent form, as shown in Equation (D.2):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF2\,@\,t_j} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1\,@\,t_i} + \begin{bmatrix} t_x + \dot{t}_x(t_i - t_j) \\ t_y + \dot{t}_y(t_i - t_j) \\ t_z + \dot{t}_z(t_i - t_j) \end{bmatrix} + \left(s + \dot{s}(t_i - t_j)\right)$$

$$\begin{bmatrix} 0 & -r_z - \dot{r}_z(t_i - t_j) & r_y + \dot{r}_y(t_i - t_j) \\ r_z + \dot{r}_z(t_i - t_j) & 0 & -r_x - \dot{r}_x(t_i - t_j) \\ -r_y - \dot{r}_y(t_i - t_j) & r_x + \dot{r}_x(t_i - t_j) & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{REF1\,@\,t_i} \quad \text{...(D.2)}$$

A note on sign conventions

The only material difference between equations (D.1) and (D.2) is that the rotation angles are given opposite signs. This arises due to the existence of two *conventions* for the specification of these rotations. In both conventions a positive rotation occurs in an anti-clockwise direction, when looking along the positive axis towards the origin. However, the IERS convention deems the rotations to be of the <u>points relative to the axes</u> while the alternative approach deems the rotations to be of the <u>axes relative to the points</u>. Errors will occur if a user applies parameters that follow the IERS convention in Equation (D.1) or if rotations based on non-IERS convention are applied in Equation (D.2).

Angles following the IERS convention can be used in Equation (D.1) only if their signs are changed. Similarly angles provided in the non-IERS convention can be used in Equation (D.2) following sign reversal. Finally, identical results will be achieved via either convention and either equation when applied *correctly*.

The sign convention being followed (and therefore the equation to be used) is not always obvious to the user, especially when encoded in software. In this case, sample data, where transformation results are also provided, can be used to confirm which convention is being employed in the software. The worked examples provided on the following pages may assist in answering this question.

14-Parameter Helmert Transformation

Reference frames

		Epochs (y.ddd)	Δt (y.yyy)
From (REF1)	ITRF2005	2010.167	
To (REF2)	GDA94	1994.000	16.4572

Input coordinates

X(ITRF2005)	-4,052,052.368	m
Y(ITRF2005)	4,212,836.041	m
Z(ITRF2005)	-2,545,105.109	m

Transformation parameters / **Updated parameters**

	Transformation parameters		Updated parameters			
Translation X (t_x)	-0.079730	m	-0.042701	m		
Translation Y (t_y)	-0.006860	m	-0.017063	m		
Translation Z (t_z)	0.038030	m	0.028814	m		
Scale (s)	0.006636	ppm	0.011474	ppm		
Rotation X (r_x)	-0.0000351	"	0.0241685	"	1.17172E-07	rad
Rotation Y (r_y)	0.0021211	"	0.0209531	"	1.01583E-07	rad
Rotation Z (r_z)	0.0021411	"	0.0213977	"	1.03739E-07	rad
Translation rate X ($\dot{t}_x$)	0.002250	m/year				
Translation rate Y ($\dot{t}_y$)	-0.000620	m/year				
Translation rate Z ($\dot{t}_z$)	-0.000560	m/year				
Scale rate ($\dot{s}$)	0.000294	ppm/year				
Rotation rate X ($\dot{r}_x$)	0.0014707	"/year				
Rotation rate Y ($\dot{r}_y$)	0.0011443	"/year				
Rotation rate Z ($\dot{r}_z$)	0.0011701	"/year				

Rotation matrix

1.000	1.0374E-07	-1.0158E-07
-1.0374E-07	1.000	1.1717E-07
1.0158E-07	-1.1717E-07	1.000

Transformed coordinates

X(GDA94)	-4,052,051.761	m
Y(GDA94)	4,212,836.195	m
Z(GDA94)	-2,545,106.015	m

14-Parameter IERS Transformation

Reference frames

		Epochs (y.ddd)	Δt (y.yyy)
From (REF1)	ITRF2005	2010.167	
To (REF2)	GDA94	1994.000	16.4572

Input coordinates

X(ITRF2005)	-4,052,052.368	m
Y(ITRF2005)	4,212,836.041	m
Z(ITRF2005)	-2,545,105.109	m

Transformation parameters / **Updated parameters**

Transformation parameters			Updated parameters			
Translation X (t_x)	-0.079730	m	-0.042701	m		
Translation Y (t_y)	-0.006860	m	-0.017063	m		
Translation Z (t_z)	0.038030	m	0.028814	m		
Scale (s)	0.006636	ppm	0.011474	ppm		
Rotation X (r_x)	0.0000351	"	-0.0241685	"	-1.17172E-07	rad
Rotation Y (r_y)	-0.0021211	"	-0.0209531	"	-1.01583E-07	rad
Rotation Z (r_z)	-0.0021411	"	-0.0213977	"	-1.03739E-07	rad
Translation rate X ($\dot{t}_x$)	0.002250	m/year				
Translation rate Y ($\dot{t}_y$)	-0.000620	m/year				
Translation rate Z ($\dot{t}_z$)	-0.000560	m/year				
Scale rate ($\dot{s}$)	0.000294	ppm/year				
Rotation rate X ($\dot{r}_x$)	-0.0014707	"/year				
Rotation rate Y ($\dot{r}_y$)	-0.0011443	"/year				
Rotation rate Z ($\dot{r}_z$)	-0.0011701	"/year				

Rotation matrix

0.000	1.0374E-07	-1.0158E-07
-1.0374E-07	0.000	1.1717E-07
1.0158E-07	-1.1717E-07	0.000

Transformed coordinates

X(GDA94)	-4,052,051.761	m
Y(GDA94)	4,212,836.195	m
Z(GDA94)	-2,545,106.015	m

Appendix E – How the Swift System Responds to Crustal Motion

Background

The Swift PPP-RTK solution is a *global* solution, designed and implemented to deliver real-time coordinates anywhere in the world in a consistently defined, global reference frame (currently ITRF2014@2010.0). For the user, that reference frame is realised through the satellite orbit and clock products delivered as part of the Swift real-time correction message. The reference frame for these products is in turn established through the coordinates assigned to the global reference stations used in orbit and clock generation (see Figure E.1).

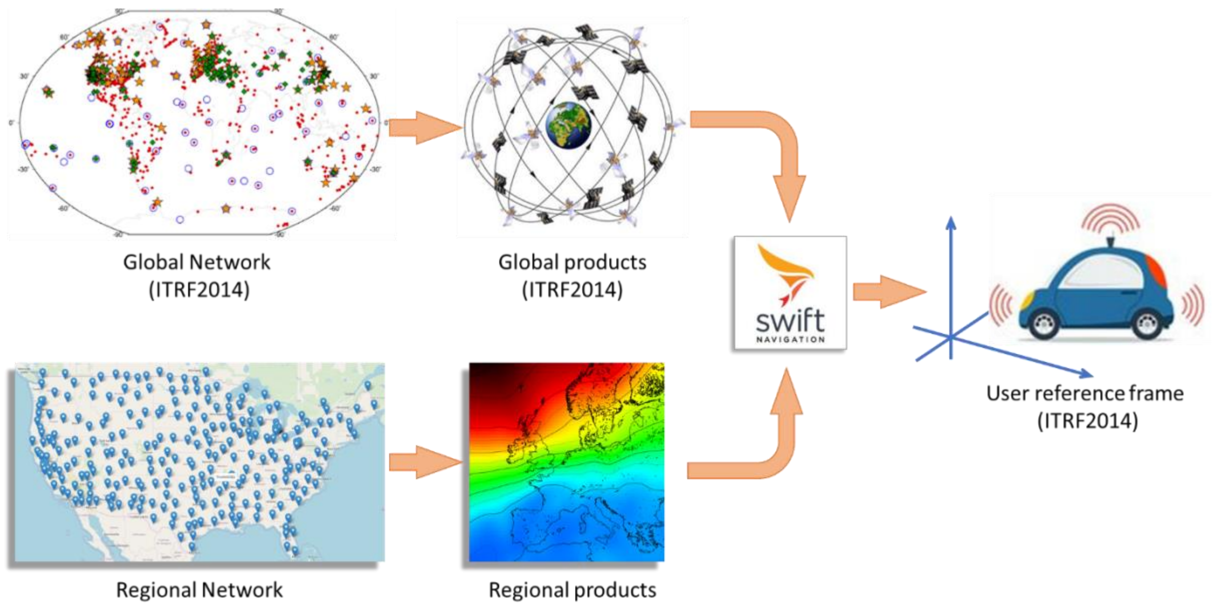

Figure E.1 – Establishing the user's reference frame through Swift's PPP-RTK solution

The important point to be made in relation to the user's real-time coordinates derived from the Swift approach is that they are not directly linked to the coordinates of reference stations in the *regional* network. This situation is in contrast to RTK-style solutions which deliver coordinates in a locally realised reference frame through the assignment of coordinates to the local RTK reference stations. In these RTK solutions, the local reference station coordinates explicitly dictate the final coordinates realised by the user and therefore the user's reference frame.

The reason for drawing this distinction is to highlight the role of and dependence on local reference stations and their coordinates in the two most common approaches to real-time GNSS positioning. It is important to understand that, for the Swift solution, local/regional reference station coordinates do not play a role in realising the reference frame for the user.

So what then is the impact of crustal motion on the global network and the regional network used in the Swift approach and how is the Swift solution impacted?

The global network

The coordinates of reference stations in the global network realise the reference frame for the satellite products (orbits, clock and biases), which in turn realise the reference frame for the user. For global network processing purposes and satellite orbit, clock and bias estimation, it is recommended that reference station coordinates in the latest ITRF (currently ITRF2014) at the current (i.e. *processing*) epoch[4] be used. It is also recommended that only stations in tectonically stable areas (i.e. stations with linear crustal motion reliably modelled through station velocities) be used in the global product estimation process. If stations in areas with non-linear crustal motion are used, current epoch coordinates will need to be estimated (not modelled), adding complexity and computational overhead to the Swift process. It should be noted that there is no advantage to the solution of including reference stations in areas of non-linear tectonic motion. It should also be noted that the described process rigorously accounts for tectonic motion on the global reference station coordinates up to the time of computation and propagates this motion into the global products. The process is illustrated in Figure E.2.

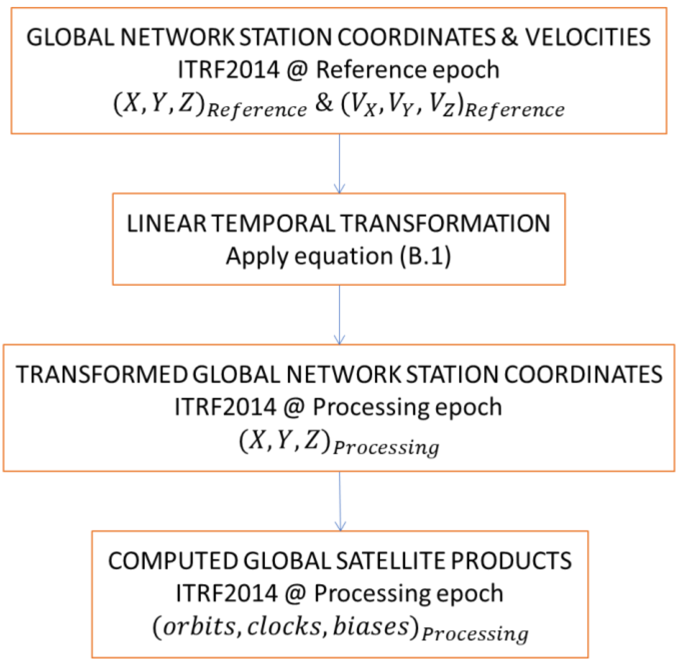

Figure E.2 – Derivation of "processing epoch" satellite products from the global network stations

As a consequence of the above approach to defining the processing epoch and computing the global satellite products, the user's position will be realised at the processing epoch and any crustal motion up to that time will be reflected in the final coordinates. Transformation into the user's frame of reference and epoch can be performed as part of delivering the solution to the user (Appendix F).

[4] The reference epoch used for processing global data products should be the date of computation. By this means the impact of crustal motion up to that date on the derived data products will be fully accounted for. Presuming point velocities are available and reliable (i.e. the stations undergo linear motion only) the computation of station coordinates at the processing epoch can be simply done using a simple point velocity model Equation (B.1).

The regional network

The coordinates of reference stations in the *regional* network will not have an impact on establishing the reference frame for the user. They will however impact on the reliable and efficient estimation of the other parameters in the Swift PPP-RTK solution (e.g. ambiguity resolution) and other PPP-RTK data products (e.g. ionosphere, troposphere). From this perspective, it is recommended that the coordinates of stations in the regional network be maintained in close relation to the global reference frame. As in the case of the global network, this can be done by developing and applying linear velocity models to these station coordinates if there is sufficient evidence of linear motion. Such confidence can only be established through long term monitoring and analysis of station behaviour. In regions of non-linear crustal motion, regular re-processing of the station coordinates will be required and consideration could also be given to developing a more sophisticated (non-linear) modelling approach. The process described here will rigorously account for tectonic motion on the regional reference station coordinates, simultaneously aligning them with the reference frame for the global network at the processing epoch and delivering a reference frame to the user that aligns with the same epoch.

Appendix F – How Swift Users are Impacted by Crustal Motion

Scenario

Suppose a particular region is subject to highly variable (non-linear) crustal motion, the cumulative effect of which can reach several centimetres over a few months. Due to its non-linear nature, the crustal motion cannot be modelled by point velocities or rotation rates, except over very short time periods. Further, the area in question is serviced by the Swift real-time PPP-RTK solution and hosts several users, mainly supporting autonomous vehicle operations.

Illustrating the problem

Consider a single point in the area described. Point movement over time is shown in Figure F.1. It is clear that a fixed, linear model (shown as - - -) applied over a period of anything more than a few months quickly becomes invalid. So what are the implications of this scenario for Swift and its users?

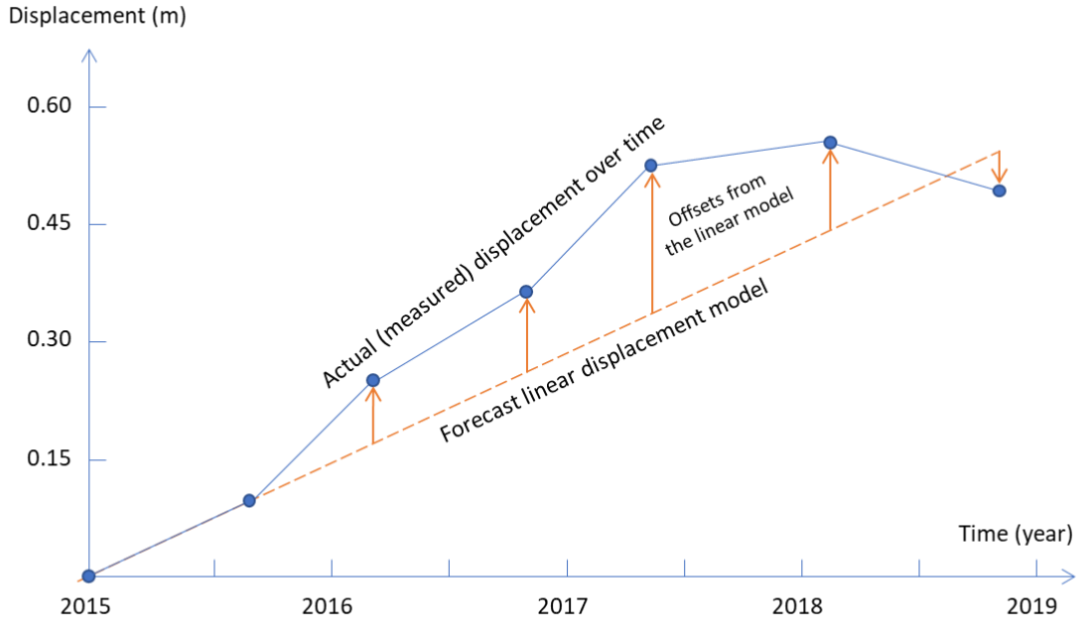

Figure F1 – Non-linear crustal motion, not able to be modelled by a simple velocity model

Will non-linear crustal motion in the deformation zone affect the Swift solution?

If the process for global network processing and satellite product generation described in Appendix E is followed and regional network station coordinates are monitored and upgraded to account for localised deformation as also described in Appendix E, non-linear, localised crustal motion will not negatively affect the Swift solution. In fact, the Swift solution will seamlessly respond to all crustal motion so that coordinates delivered to the user will reflect the reality of changes in the physical environment up to the time of processing.

Will non-linear crustal motion in the deformation zone affect the user's map data?

(a) When the map data is not being regularly updated over time
In this case, the map data will not reflect changes in the location of physical features in the user's environment caused by crustal motion. Given the Swift solution will track these changes, the two solutions will not align *a priori*. To overcome this misalignment (if it deemed significant), the Swift solution will need to be temporally transformed from the processing epoch of the map data – thus removing the crustal motion from the Swift solution. However, if the *relative* locations of features have changed over time due to non-linear crustal motion, navigation and guidance decisions based on out of date map data could be affected, potentially putting user safety at risk. Changes in relative location caused by crustal motion significant enough to cause a problem for autonomous vehicle guidance will be rare[5]. In such cases, an update of the map data is the only remedy. See Appendix G for further discussion of the question of dealing with out of date map data.

(b) When the map data is being updated over time
Presuming map updates are of sufficient frequency to capture the impacts of non-linear crustal deformation, or, as an alternative, the map data is being time transformed via some form of deformation model, the map data will hold the up to date (absolute and relative) location of all relevant features. For Swift, this is the ideal scenario as the updated map will be compatible with the Swift solution. Any reference frame differences (temporal and/or spatial) between the map data and the Swift solution can simply be accommodated through an appropriate transformation.

Conclusion

In autonomous driving applications, the dangers posed by incomplete, inaccurate and out of date map data – whatever the cause – are potentially life threatening. Building a navigation solution for fully autonomous vehicles that relies upon anything other than a complete, accurate and up to date digital map product is mandatory. Of course the practical difficulties and financial implications of regular and routine upgrades to map data should not be underestimated, but fully autonomous vehicle operation demands nothing less. Clearly then, cooperation between the provider of the navigation solution and the provider of the map data is required to deliver a safe and robust navigation outcome to the user.

[5] The situation described above is analogous to the inclusion of new physical features or the removal of existing features in the real-world, with such changes not being reflected in the outdated map data. This would in fact be a much more common and more significant problem for autonomous vehicle operation and one which cannot be solved by transformation. Only updated map data can reflect such changes.

Appendix G – A Process to Deal with Crustal Motion

If the user's map data is not being updated over time to ensure the epoch of the map data aligns with the epoch of the Swift solution, one of the following temporal transformation strategies must be followed. A spatial transformation to align reference frames may also be required:

- Option 1 – Transform the map data forward through time to match the epoch of the Swift solution (i.e. a *map* transformation)
- Option 2 – Transform the Swift solution backward through time to match the epoch of the map data (i.e. a *navigation* transformation)

Option 1 will update the map data to the processing epoch of the Swift solution by cumulatively adding the linear and non-linear crustal motion that has taken place since the map data was acquired. While providing a comprehensive approach, the computational requirements and practical challenges of performing this type transformation should not be underestimated. It may prove impractical for Swift to routinely undertake this task.

Option 2 will remove the impact of crustal deformation from the Swift solution by shifting it backward through time to align it to the epoch of the map data. From a practical perspective, and bearing in mind the need for computational efficiency, Option 2 will be easier to implement and will require less computational effort.

The various elements of either transformation process described above are shown in Figure G.1. The only distinction is whether the crustal deformation is being added to the map data or removed from the navigation solution.

The uppermost box in Figure G.1 represents the *linear* component of crustal motion. This component can be routinely modelled and accounted for, either by point velocities (Equation B.1) or rotation rates (Equation B.2), typically provided by the IGS or a local geodetic agency. The middle box shows the *non-linear* component which must be measured and modelled regionally or locally depending on the scale of motion. Usually this would be the responsibility of a government geodetic authority. Such models would typically take the form of cumulative deformation grids fitted to the measured deformation. Refer to Sections 2.3 and 2.4.

It is important to note that the non-linear motion will in many cases be the sum of separate and unrelated motions from a diverse array of sources. Some will be natural (e.g. tectonic), while others will be anthropogenic (e.g. induced by mining or ground water removal). The practical challenge of implementing the approach shown in Figure G.1 will be the measurement, monitoring and modelling of the non-linear crustal motion in a manner that is timely and sufficiently accurate to support the needs of the Swift solution and the Swift user.

LINEAR MOTION (Δx_{linear})
Long-term, wide-area, predictable (e.g. tectonic plate movement)

Computed from:

- Station velocities: V_x, V_y, V_z ...or...
- Rotation rates: $\dot{r}_x, \dot{r}_y, \dot{r}_z$

$+$

NON-LINEAR MOTION $(\Delta \boldsymbol{x}_{non-linear})$
Short-term, localised, unpredictable (e.g. post-seismic deformation, anthropogenic movement)

Computed from:

- Deformation grids: $\Delta x_{non-linear} = \sum_{i=1}^{n} G_i$

$=$

TOTAL MOTION $(\Delta \boldsymbol{x}_{total})$

- $\Delta \boldsymbol{x}_{total} = \Delta \boldsymbol{x}_{linear} + \Delta \boldsymbol{x}_{non-linear}$

<u>Figure G.1</u> – Elements of the temporal transformation strategy