UNITED NATIONS GLOBAL GEODETIC CENTRE OF EXCELLENCE

MODERNISING GEOSPATIAL REFERENCE SYSTEM CAPACITY DEVELOPMENT WORKSHOP

EXAMPLE: Creating transformation parameters

Nicholas Brown
Head of Office, UN-GGCE

HOW

Day 2, Session 3 [2_3_2]

Acknowledgements: John Dawson (AUS); Guorong Hu (AUS).

Computing transformation parameters

Fiji's datum is WGS72

The Pacific Islands Applied Geoscience Commission (SOPAC) requested Geoscience Australia to compute International Terrestrial Reference Frame (ITRF) coordinates for 18 survey sites on islands in the northern Fiji archipelago from continuous geodetic Global Positioning System (GPS) measurements observed from 8th July to 5th August 2008 inclusive.

These coordinates provide the coordinate reference frame to be used to define Fiji's claim to extended continental shelf under the provisions of Article 76 of the United Nations Convention on the Law of the Sea

Computing transformation parameters

STRONGER.
TOGETHER.

Computing transformation parameters

1. Observe GNSS on marks on which WGS72 coordinates are available

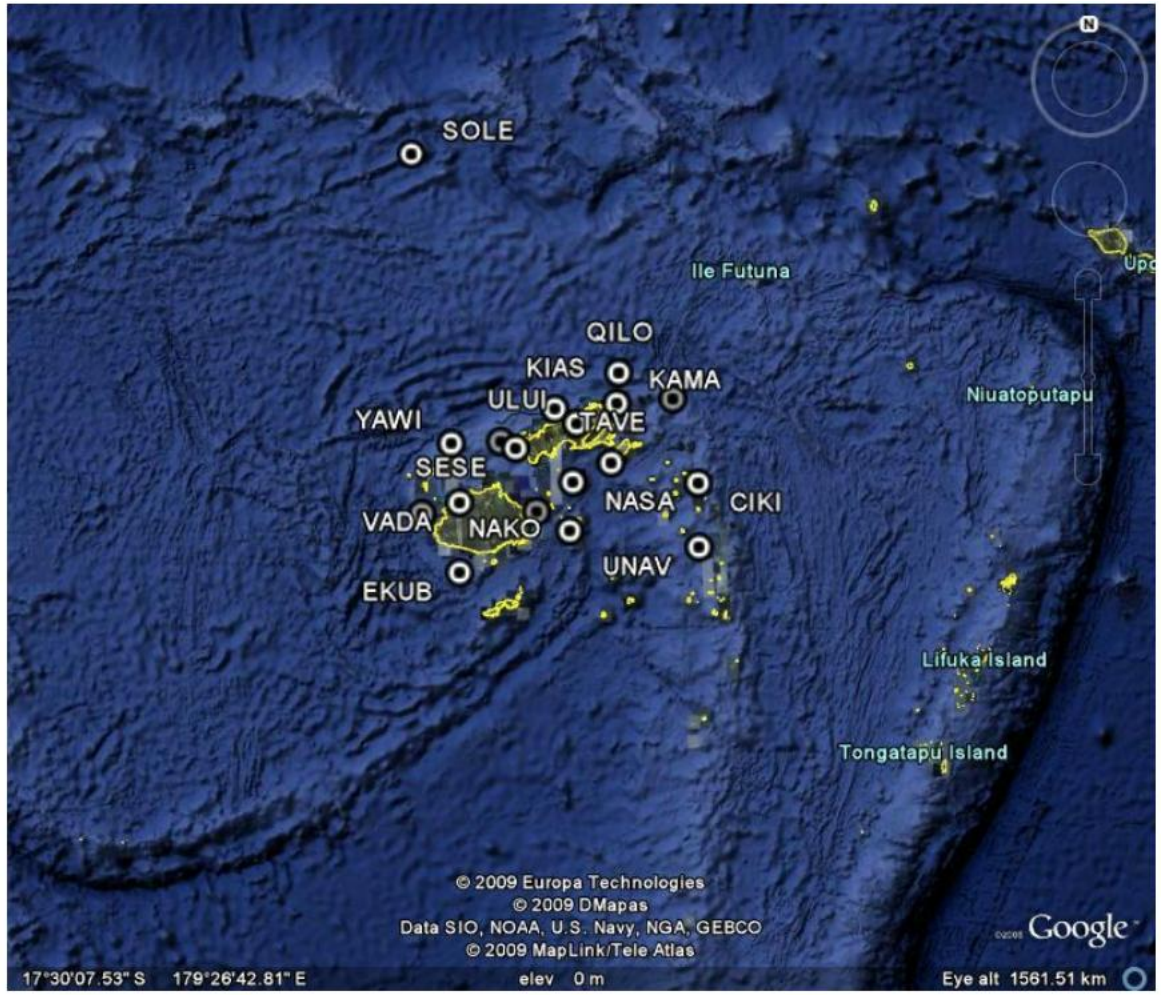

Figure 1: *The distribution of the stations of the 2008 Northern Fiji Islands GPS campaign.*

- 18 stations
- Dual freq. observations
- 28 days continuously

Dawson and Hu (AUS)

Computing transformation parameters

2. Process GNSS data to compute ITRF2005@2008 coordinates

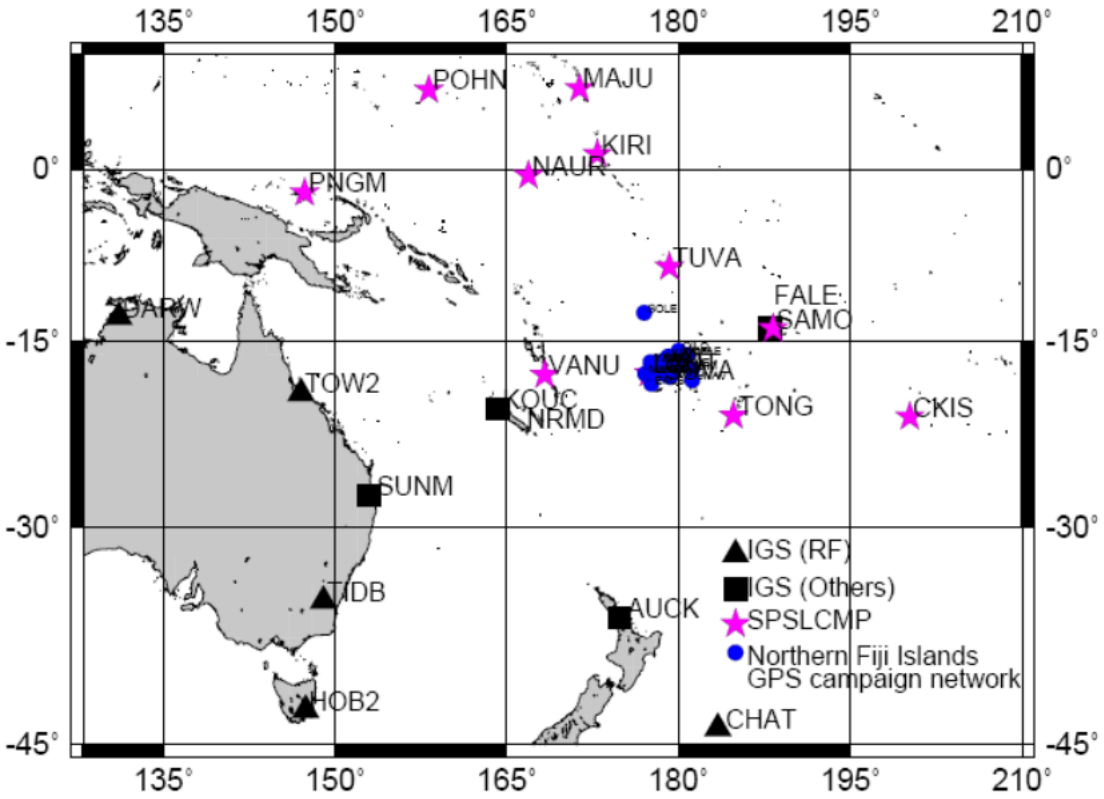

Figure 2: *The SPSLCMP/IGS stations included in the GPS data processing.*

- Included 5 IGS stations from the surrounding area in the processing (to align / constrain to ITRF)
- Also included Pacific Island regional reference sites to “aid ambiguity resolution and to reduce baseline lengths between the above IGS stations and the Fiji campaign stations”.

Dawson and Hu (AUS)

STRONGER.
TOGETHER.

Computing transformation parameters

WGS72

SITE	X (m)	Y (m)	Z (m)
CIKI	-6090790.884	-128354.367	-1882866.878
EKUB	-6045005.524	248397.533	-2012568.398
KADV	-6025612.829	211678.930	-2075994.466
KAMA	-6128128.427	-876.912	-1763105.787
KIAS	-6125126.204	96614.615	-1770429.211
MATU	-6026782.025	26320.369	-2081724.597
NAKO	-6069393.337	74317.387	-1952868.080
ODRI	-6023321.140	-167612.926	-2084345.816
ONOI	-5969814.367	-131365.080	-2234500.378
SAIL	-6108964.587	182329.478	-1818369.130
SESE	-6106871.289	158749.826	-1828962.411
SOLE	-6219742.471	324299.487	-1371185.634
TAVE	-6101703.123	8543.670	-1852334.121
UNAV	-6060199.677	-129057.382	-1978129.289
VADA	-6078939.965	247719.845	-1908646.928
VATO	-5999583.742	-183087.633	-2149906.544

ITRF2005@2008

SITE	X (m)	Y (m)	Z (m)
CEVA	-5901432.3013	554458.5449	-2347334.7909
CIKI	-6090792.0436	-128369.4471	-1882863.3918
EKUB	-6045006.4702	248381.1969	-2012567.0641
KADV	-6025613.9456	211663.5349	-2075992.0838
KAMA	-6128129.1449	-891.7874	-1763104.8897
KIAS	-6125127.8089	96598.2963	-1770426.0192
LAUT	-6075194.5732	270923.9137	-1917189.4371
MAJU	-6257572.3044	950332.6831	785215.2375
MANF	-6071340.2395	307355.4745	-1923860.4408
MATU	-6026782.9119	26305.2446	-2081722.4555
NAKO	-6069393.3645	74300.7507	-1952865.8078
NASA	-6092446.1776	69059.2460	-1880371.3209
NAUR	-6212555.1027	1442786.8956	-61006.6725
ODRI	-6023324.3914	-167626.9300	-2084343.2937
ONOI	-5969817.8125	-131380.5993	-2234496.5488
OVAL	-6078392.5714	126360.6143	-1923939.7659
QELE	-6129224.7546	-89303.1288	-1756463.9017
QILO	-6141289.6922	-4172.9920	-1717003.0966
SAIL	-6106871.9025	158732.3759	-1828960.5555
SESE	-6108966.7088	182311.7462	-1818366.9382
SOLE	-6219743.2094	324285.6012	-1371182.6871
SUVA	-6060677.2218	166617.2074	-1973761.3461
TAVE	-6101703.0397	8527.7474	-1852331.7513
TUCO	-5955264.9601	-129584.0062	-2272571.3656
TURA	-5954650.3967	-119772.0561	-2274702.3597
ULUI	-6119100.7845	62728.2252	-1792699.4820
UNAV	-6060202.2155	-129071.4774	-1978127.1784
VADA	-6078942.2205	247703.3363	-1908644.9425
VATO	-5999585.5068	-183103.6074	-2149904.6272
YAWI	-6105774.5980	260113.9199	-1819818.9082

Computing transformation parameters

$$\begin{bmatrix} X_{ITRF05} \\ Y_{ITRF05} \\ Z_{ITRF05} \end{bmatrix} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} + (1+S) \times \begin{bmatrix} 1 & R_Z & -R_Y \\ -R_Z & 1 & R_X \\ R_Y & -R_X & 1 \end{bmatrix} \begin{bmatrix} X_{WGS72} \\ Y_{WGS72} \\ Z_{WGS72} \end{bmatrix}$$

$(X_{ITRF05}, Y_{ITRF05}, Z_{ITRF05})$	are the transformed Cartesian coordinates (m) consistent with the ITRF05@2008.0 coordinates listed in this report.
$(X_{WGS72}, Y_{WGS72}, Z_{WGS72})$	are the Cartesian Fiji WGS72 geodetic datum coordinates (m).
(T_X, T_Y, T_Z)	are the coordinate origin translation parameters (m).
(RX, RY, RZ)	are the coordinate axis rotations (radians).
S	is the scale change between both coordinate systems.

STRONGER.
TOGETHER.

Computing transformation parameters

- Need at least 3 common points; but the more you have the better
- The transformation parameters are computed using a least squares adjustment process

- **Translation Parameters (Tx,Ty,Tz):**
  - Compute the mean difference between corresponding coordinates of Datum 1 and Datum 2 to get initial translation values.

- **Scale Factor (s):**
  - Estimate the scale change as the ratio of the distances between points in Datum 2 and Datum 1.

- **Rotation Parameters (Rx,Ry,Rz):**
  - Compute the rotation angles that best align Datum 1 with Datum 2. This involves solving for the rotation matrix R using a linearized form of the transformation equations.

Computing transformation parameters

Solve the Linearized System

- Set up a system of equations:
- $\Delta = A \cdot P$
  - Δ: The differences between the observed and transformed coordinates.
  - A: The design matrix (based on partial derivatives of the transformation equations).
  - P: The unknown parameters ($Tx, Ty, Tz, Rx, Ry, Rz, s$).

Iteratively Refine the Parameters

- Solve for P using matrix operations (e.g., $P = (A^T A)^{-1} A^T \Delta$
- Apply the estimated parameters to the input data.
- Refine the estimates until residuals are minimized.

Verify the Transformation

- Use the computed parameters to transform the original coordinates.
- Compare the transformed coordinates with the target datum to ensure accuracy.

- This can be done in GNSS Combination Software (e.g. CATREF), proprietary software … or with assistance from AI

Computing transformation parameters

WGS72			ITRF2005@2008		
X	Y	Z	X	Y	Z
-6090790.9	-128354.37	-1882866.9	-6090792	-128369.45	-1882863.4
-6045005.5	248397.533	-2012568.4	-6045006.5	248381.197	-2012567.1
-6025612.8	211678.93	-2075994.5	-6025613.9	211663.535	-2075992.1
-6128128.4	-876.912	-1763105.8	-6128129.1	-891.7874	-1763104.9
-6125126.2	96614.615	-1770429.2	-6125127.8	96598.2963	-1770426
-6026782	26320.369	-2081724.6	-6026782.9	26305.2446	-2081722.5
-6069393.3	74317.387	-1952868.1	-6069393.4	74300.7507	-1952865.8
-6023321.1	-167612.93	-2084345.8	-6023324.4	-167626.93	-2084343.3
-5969814.4	-131365.08	-2234500.4	-5969817.8	-131380.6	-2234496.5
-6108964.6	182329.478	-1818369.1	-6108966.7	182311.746	-1818366.9
-6106871.3	158749.826	-1828962.4	-6106871.9	158732.376	-1828960.6
-6219742.5	324299.487	-1371185.6	-6219743.2	324285.601	-1371182.7
-6101703.1	8543.67	-1852334.1	-6101703	8527.7474	-1852331.8
-6060199.7	-129057.38	-1978129.3	-6060202.2	-129071.48	-1978127.2
-6078940	247719.845	-1908646.9	-6078942.2	247703.336	-1908644.9
-5999583.7	-183087.63	-2149906.5	-5999585.5	-183103.61	-2149904.6

STRONGER.
TOGETHER.

Computing transformation parameters

AI results

Translations (meters):

Tx= -6.9344
Ty= -21.2037
Tz= -10.4443

Scale factor (unitless):

S= -1.42 ppm

Rotations (arcseconds):

Rx= -0.1225
Ry= 0.3425
Rz= -0.2289

ChatGPT RMSE residuals

X: 0.887 m
Y: 1.038 m
Z: 0.745 m

Dawson and Hu results

Translations (meters):

Tx= -7.0295
Ty= -22.1185
Tz= -10.1505

Scale factor (unitless):

s= -1.4227 ppm

Rotations (arcseconds):

Rx= -0.1139
Ry= +0.3325
Rz= -0.2573

Dawson and Hu RMSE residuals

X: 0.983 m
Y: 0.863 m
Z: 0.808 m

STRONGER.
TOGETHER.

Python code

```
import numpy as np

def compute_transformation_params(datum1, datum2):
    """
    Compute the 7-parameter transformation parameters using least squares.

    Parameters:
    - datum1: numpy array of shape (n, 3), coordinates in the first datum
    - datum2: numpy array of shape (n, 3), coordinates in the second datum

    Returns:
    - parameters: dict containing translation, rotation, and scale factor
    """
    # Ensure data is in numpy array format
    datum1 = np.array(datum1)
    datum2 = np.array(datum2)

    # Compute centroids
    centroid1 = np.mean(datum1, axis=0)
    centroid2 = np.mean(datum2, axis=0)

    # Center the coordinates
    centered1 = datum1 - centroid1
    centered2 = datum2 - centroid2

    # Scale factor (s)
    scale = np.sqrt(np.sum(centered2**2) / np.sum(centered1**2))

    # Compute the rotation matrix using Singular Value Decomposition (SVD)
    H = np.dot(centered2.T, centered1)
    U, _, Vt = np.linalg.svd(H)
    rotation_matrix = np.dot(U, Vt)

    # Translation vector (Tx, Ty, Tz)
    translation = centroid2 - scale * np.dot(rotation_matrix, centroid1)

    # Extract rotation angles (in radians)
    rotation_angles = {
        'Rx': np.arctan2(rotation_matrix[2, 1], rotation_matrix[2, 2]),
        'Ry': np.arctan2(-rotation_matrix[2, 0], np.sqrt(rotation_matrix[2, 1]**2 + rotation_matrix[2, 2]**2)),
        'Rz': np.arctan2(rotation_matrix[1, 0], rotation_matrix[0, 0])
    }

    # Return parameters
    return {
        'Translation': translation,
        'Rotation (radians)': rotation_angles,
        'Scale Factor': scale
    }

# Example usage:
# Replace `datum1_coords` and `datum2_coords` with your actual data
datum1_coords = [
    [-6090790.884, -128354.367, -1882866.878],
    [-6045005.524, 248397.533, -2012568.398],
    # Add more points as needed
]

datum2_coords = [
    [-6090792.0436, -128369.4471, -1882863.3918],
    [-6045006.4702, 248381.1969, -2012567.0641],
    # Add more points as needed
]

# Compute transformation parameters
params = compute_transformation_params(datum1_coords, datum2_coords)

# Display results
print("Translation Parameters (Tx, Ty, Tz):", params['Translation'])
print("Rotation Angles (Rx, Ry, Rz) in radians:", params['Rotation (radians)'])
print("Scale Factor:", params['Scale Factor'])
```

- Replace the placeholder data (datum1_coords and datum2_coords) with your actual coordinates from the two datums.
- The function will compute the translation, rotation (in radians), and scale factor.

Discussion

- What should you consider before using AI?
- How could you validate the method?
- Who could you ask for assistance?
- How to deal with legacy geodetic data (e.g. no ellipsoidal heights)