List of Figures

List of Tables

Table of Contents

1. INTRODUCTION ABOUT PROJECT

1.1 Overview Of Project

Problem Statement :

Geographic Information System (GIS) is used to store, retrieve and render the Earth-relative spatial data. GIS deals with flat map projections. Geodetic datum uniquely defines each location on Earth’s surface with coordinates in latitude and longitude. Notable geodetic datum’s are NAD27, NAD83, Everest datum and WGS84. The coordinates for each point on Earth in one geodetic datum is different from other geodetic datum. For example, the latitude and longitude of a location in a Everest datum differs from NAD83 or WGS84. The transformation of one datum to other geodetic datum is known as a datum shift. GIS applications demand the support for more than one geodetic datum’s for both vector and raster maps, hence the datum transformations or datum shift are often necessary. Users of this system: The solution will help Geographic Information System (GIS) based applications which requires different geodetic datum. Technology that can help address the issue: Any high level Program language, Map data handling techniques.

Desired Solution :

The geodetic datum’s transformation tool should have a provision for input to take map data of one geodetic datum and convert into other geodetic datum for each NAD27, NAD83, Everest datum and WGS84.

What is Geodetic datum :

A geodetic datum is a tool used to define the shape and size of the earth, as well as the reference point for the various coordinate systems used in mapping the earth. Throughout time, hundreds of different datums have been used - each one changing with the earth views of the times.

True geodetic datums, however, are only those which appeared after the 1700s. Prior to that, the earth's ellipsoidal shape was not always taken into consideration, as many still believed it was flat.

Since most datums today are used for measuring and showing large portions of the earth, an ellipsoidal model is essent

Geodetic Co-ordinates :

In geodetic coordinates, the Earth's surface is approximated by an ellipsoid, and locations near the surface are described in terms of latitude(Φ), longitude(λ) and height(h).

Longitude lines are perpendicular to and latitude lines are parallel to the equator.

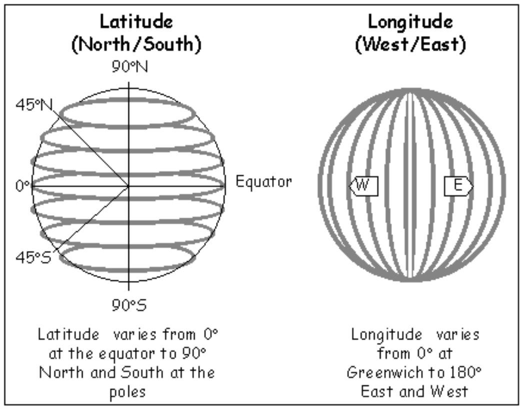


Figure 1 : latitude and longitude

Defining and derived parameters:
The ellipsoid is completely parameterised by the semi-major axis and the flattening .

Parameter	**Symbol**
Semi-major axis	a
Reciprocal of flattening	$1/f$

Table 1 : Defining parameters

From a and f it is possible to derive the semi-minor axis b, first eccentricity e and second eccentricity e' of the ellipsoid

Parameter	**Value**
semi-minor axis	$b = a(1-f)$
First eccentricity squared	$e^2 = 1-b^2/a^2 = 2f-f^2$
Second eccentricity squared	$e'^2 = a^2/b^2-1 = f(2-f)/(1-f)^2$

Table 2 : derived parameters

Datum Transformation

Transformations among datums can be accomplished in a number of ways. There are transformations that directly convert geodetic coordinates from one datum to another. There are more indirect transforms that convert from geodetic coordinates to ECEF coordinates, transform the ECEF coordinates from one datum to the another, then transform ECEF coordinates of the new datum back to geodetic coordinates. There are also grid-based transformations that directly transform from one (datum, map projection) pair to another (datum, map projection) pair.

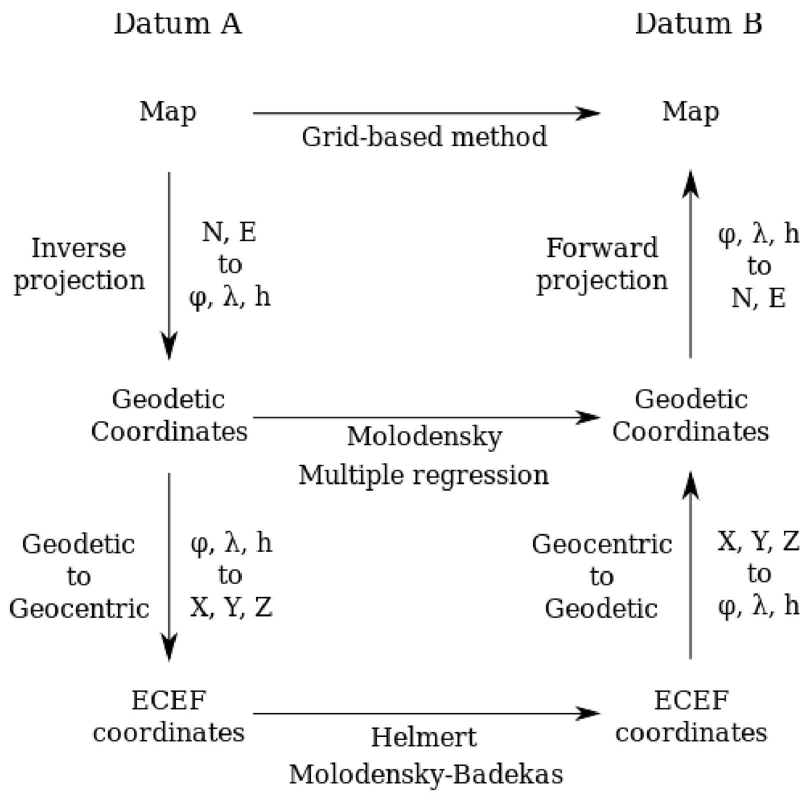


Figure 2 : datum transformation techniques

Molodensky Transformation

The Molodensky transformation converts directly between geodetic coordinate systems of different datums without the intermediate step of converting to geocentric ECEF coordinates. It requires the three shifts between the datum centers and the differences between the reference ellipsoid semi-major axes and flattening parameters.

The Molodensky transform is used by the National Geospatial-Intelligence Agency (NGA) in their standard TR8350.2 and the NGA supported GEOTRANS program. The

Molodensky method was popular before the advent of modern computers and the method is part of many geodetic programs.

Multiple regression equations

Datum transformations through the use of empirical multiple regression methods were created to achieve higher accuracy results over small geographic regions than the standard Molodensky transformations. MRE transforms are used to transform local datums over continent-sized or smaller regions to global datums, such as WGS 84.

1.2 Scope Of Project

this tool can be used by GIS(Geographic information systems) applications that demand multiple datums. Example of various GIS applications are

Uses of GIS range from indigenous people, communities, research institutions, environmental scientists, health organizations, land use planners, businesses, and government agencies at all levels.

Some examples include:

- Crime mapping
- Economic development and investment promotion
- Historical geographic information systems
- GIS and hydrology
- Remote sensing applications
- Traditional knowledge GIS
- Public Participation GIS
- Road networking
- Wastewater and stormwater systems
- Waste management

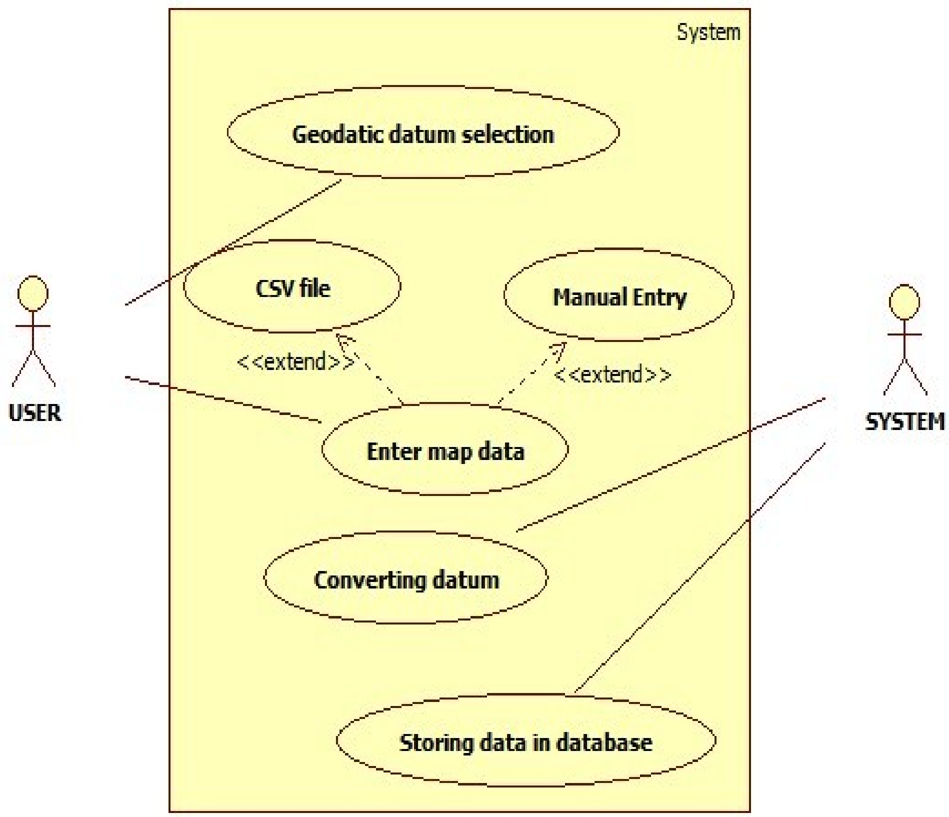
1.3 Purpose Of Project

Purpose of this project is to allow GIS applications to work on maps of multiple Geodetic datum. Purpose of this project is to work as a tool that GIS applications use as a additional function.

2. HARDWARE AND SOFTWARE REQUIREMENT

- minimum requirement for this tool is to have python 3.x installed in your computer.
- This software can work on any type of platform ans any type of hardware.

3. SYSTEM FLOW CHART OF IMPLEMENTATION



4. MAIN MODULES OF PROJECT

4.1 Standard Molodensky Transformation

$$\phi_{WGS84} = \phi_{Local} + \Delta\phi$$
$$\lambda_{WGS84} = \lambda_{Local} + \Delta\lambda \qquad (4.1)$$
$$h_{WGS84} = h_{Local} + \Delta h$$

where :

$\Delta\phi, \Delta\lambda, \Delta h$ *are provided by the Standard Molodensky transformation formulas*

$$\Delta\phi'' = \left\{ -\Delta X \sin\phi\cos\lambda - \Delta Y \sin\phi\sin\lambda + \Delta Z\cos\phi + \Delta a \frac{\left(R_n e^2 \sin\phi\cos\phi\right)}{a} + \Delta f\left[R_M\left(a/b\right) + R_N\left(b/a\right)\right]\sin\phi\cos\phi \right\}\cdot\left[\left(R_M + h\right)\sin 1''\right]^{-1}$$

$$\Delta\lambda'' = \left[-\Delta X\sin\lambda + \Delta Y\cos\lambda\right]\cdot\left[\left(R_N + h\right)\cos\phi\sin 1''\right]^{-1} \qquad (4.2)$$

$$\Delta h = \Delta X\cos\phi\cos\lambda + \Delta Y\cos\phi\sin\lambda + \Delta Z\sin\phi - \Delta a\left(\frac{a}{R_N}\right) + \Delta f\left(\frac{b}{a}\right)R_N\sin^2\phi$$

where :

ϕ, λ, h = *geodetic coordinates (old ellipsoid)*

ϕ = *geodetic latitude*

λ = *geodetic longitude*

$h = N + H$

where :

h = *geodetic height (height relative to the ellipsoid)*

N = *geoid height*

H = *orthometric height (height relative to the geoid)*

$\Delta\phi, \Delta\lambda, \Delta h$ = *corrections to transform local geodetic datum coordinates to WGS84*
ϕ, λ, h *values. The units of* $\Delta\phi$ *and* $\Delta\lambda$ *are arc seconds (″); the units of* Δh *are meters (m).*

Figure 3 : standard molodensky transformation formulae-1

$\Delta X, \Delta Y, \Delta Z = shifts\ between\ centers\ of\ the\ local\ geodetic\ datum\ and\ WGS84\ ellipsoid;$

$corrections\ to\ transform\ local\ geodetic\ system-related\ rectangular\ coordinates\ (X, Y, Z)\ to$

$WGS84-related\ X, Y, Z\ values.$

$a = semi-major\ axis\ of\ the\ local\ geodetic\ datum\ ellipsoid$

$b = semi-minor\ axis\ of\ the\ local\ geodetic\ datum\ ellipsoid$

$b/a = 1 - f$

$f = flattening\ of\ the\ local\ geodetic\ datum\ ellipsoid$

$\Delta a, \Delta f = differences\ between\ the\ semi-major\ axis\ and\ flattening\ of\ the\ local\ geodetic\ datum\ ellipsoid$

$and\ the\ WGS\ 84\ ellipsoid,\ respectively\ (WGS84 - Local).$

$e = first\ eccentricity$

$e^2 = 2f - f^2$

$R_N = radius\ of\ curvature\ in\ the\ prime\ vertical$

$$R_N = \frac{a}{\left(1 - e^2 \sin^2 \phi\right)^{1/2}}$$

$R_M = radius\ of\ curvature\ in\ the\ meridian$

$$R_M = \frac{a\left(1 - e^2\right)}{\left(1 - e^2 \sin^2 \phi\right)^{3/2}}$$

$Note: All\ \Delta - quantities\ are\ formed\ by\ subtracting\ local\ geodetic\ datum\ ellipsoid\ value\ from\ WGS\ 84\ Ellipsoid\ values.$

Figure 4 : standard molodensky transformation formulae-2

5. IMPLEMENTATION DETAILS WITH SNAP SHOT

5.1 Standard Molodensky Transformations

Formulae to python code conversion

sample code for above formulae in python

```
from math import *


# input

# co-ordinates in local geodetic datums

lat_local = 0  # value not available yet

lon_local = 0  # value not available yet

h_local = 0  # value not available yet


# delta x, y, z - shifts between local geodetic datum and wgs84
ellipsoid center

delta_x = 0  # value not available yet

delta_y = 0  # value not available yet

delta_z = 0  # value not available yet


# a - semi-major axis of local geodetic datum ellipsoid

# b - semi-minor axis of local geodetic datum ellipsoid

# f - flattening of local geodetic datum ellipsoid

# e - first eccentricity

a = 1  # value not available yet

b = 1  # value not available yet

f = 1 - (float(b) / a)

e = sqrt(2 * f - f ** 2)


# f_wgs , a_wgs - flattening and semi-major axis of wgs84 ellipsoid

f_wgs = 1  # value not available yet

a_wgs = 1  # value not available yet
```

```
# delta a, f - difference between the semi-major and flattening of local geodetic datum ellipsoid and wgs84 ellipsoid

#  (wgs84 - local)

delta_f = f_wgs - f

delta_a = a_wgs - a


# Rn - radius of curvature in prime vertical

# Rm - radius of curvature in meridian

# Rn = a / sqrt(1 - pow(e*sin(radians(lat_local)),2))

Rn = a / (1 - (e * sin(lat_local)) ** 2) ** (1 / 2)

Rm = a * (1 - e ** 2) / (1 - (e * sin(lat_local)) ** 2) ** (3 / 2)


# lattitude correction

delta_lat = (-delta_x * sin(radians(lat_local)) * cos(radians(lon_local)) - delta_y * sin(radians(lat_local)) * sin(

    radians(lon_local)) + delta_z * cos(radians(

    lat_local)) + delta_a * (Rn * e ** 2 * sin(radians(lat_local)) * cos(radians(lat_local))) / a + delta_f * (

                     Rm * (a / b) + Rn * (b / a)) * sin(radians(lat_local)) * cos(radians(lat_local))) / (

                        (Rm + h_local) * sin(radians(1 / 3600)))


# longitude correction

delta_lon = (-delta_x * sin(radians(lon_local)) + delta_y * cos(radians(lon_local))) / (

            (Rn + h_local) * cos(radians(lat_local)) * sin(radians(1 / 3600)))


# height correction

delta_h = delta_x * cos(radians(lat_local)) * cos(radians(lon_local)) + delta_y * cos(radians(lat_local)) * sin(

    radians(lon_local)) + delta_z * sin(radians(

    lat_local)) - delta_a * (a / Rn) + delta_f * (b / a) * Rn * (sin(radians(lat_local)) ** 2)
```

```
# final step of conversion
lat_wgs = lat_local + delta_lat
lon_wgs = lon_local + delta_lon
h_wgs = h_local + delta_h
```

6. DATA DICTIONARY

Column Name	**Datatype**	**Description**
a	FLOAT	STORES THE SEMI-MAJOR AXIS FOR THEGEODETIC DATUM ELLIPSIOD
f	FLOAT	STORES THE FLETTENING OF LOCAL GEODETIC DATUM ELLIPSIOD

Table 3 : Data Dictionary

7. FURTHER ENHANCEMENT

- Furthermore we can implement user interface to this tool
- we can also add functionality of multiple geodetic datums
- we can add functionality to input data-set file and convert the whole file and output converted data to another dataset file.

8. Bibliography

- Geodetic datum - https://en.wikipedia.org/wiki/Geodetic_datum
- Geodetic Datum - https://www.thoughtco.com/geodetic-datums-overview-1434909
- Geographic coordinate system - https://en.wikipedia.org/wiki/Geographic_coordinate_system
- World Geodetic System - https://en.wikipedia.org/wiki/World_Geodetic_System
- Geographic coordinate conversion - https://en.wikipedia.org/wiki/Geographic_coordinate_conversion
- National Geospatial-Intelligence Agency - https://en.wikipedia.org/wiki/National_Geospatial-Intelligence_Agency
- (NGA) Datum Transformations - http://earth-info.nga.mil/GandG/coordsys/datums/index.html
- (NGA) Standard Molodensky Transformations - http://earth-info.nga.mil/GandG/coordsys/datums/standardmolodensky.html