Proceedings of the *Open source GIS - GRASS users conference 2002* - Trento, Italy, 11-13 September 2002

Reference frames: definition and management

Elena Carcano, Domenico Sguerso, Alfonso Vitti

Dipartimento di Ingegneria Civile e Ambientale, Università degli Studi di Trento,
Via Mesiano 77, 38050 Trento, ITALY, tel. ++39 0461882608, fax ++390461882672,
e-mail domenico.sguerso@ing.unitn.it

1 Introduction

Geographical Information Systems often require much heterogeneous kind of data, like cartography supports, GPS points, satellite images and so on, generally in different reference frames and/or in different cartography representations. The problem of different datum and representations is a classical problem in geodesy that software generally solves with dedicated tools. This problem has different sources, so different kinds of transformations have to be applied. Starting from the reference frame and cartography definitions, this paper would like point out the main relations with data-set and that the external inputs necessary for a correct planimetry and altimetry data management. Some issues related to the points positioning in a three dimensional space are also presented. The information needed to completely describe the position of a point in the space depend on the degrees of freedom of the point. An object, regarded as a rigid body in the 3D space, has 6 degrees of freedom: three translations and three rotations so 6 information are necessary and sufficient to completely fix position. These information must univocally describe that particular point in an absolutely way. Some reference structures (axes, surfaces, directions) have to be defined to permanently recognize the same points in the same positions. Therefore a reference frame has to be used to univocally describe the reality. As the Earth is a non-flat surface to correctly describe object lying on it a reference frame in accord to Earth shape and features has to be defined. Different choices are possible depending on the way the Earth and positioning on it are described, mainly ellipsoids are used for horizontal positioning while geoids, gravity related models, for referencing the elevation. The position and the orientation of a geodetic model with respect to the Earth are described through a datum. Different coordinates system can also be used to mathematically describe geographic locations. In each coordinate-reference system the geographic position of a point gets its own coordinate values. Moreover, the Earth is not a rigid body, its shape changes with time, therefore using an Earth reference frame requires to be update it with a certain frequency. Since many reference frames can be defined, it is necessary to develop relations and mathematical models to transfer information from one reference frame to another. Nevertheless, frequently the transformation parameters are not available, so some different procedures with different approximations have to be accepted. To plot a curved surface, like ellipsoid one, to a plane cartographic projection are used. In this way the objects features readable on the cartographic map are representative of the real dimension and position of the same objects on the Earth. GRASS uses and analyses data describing objects on the Earth surface. This GIS manages data in a large set of different reference frames and supports cartography and datum transformation, some examples with its dedicated tools are reported.

2 Coordinates systems

Taking measurements on the Earth surface leads to obtain the points positions of our survey, but for any taken measurements only relative positions among different points can be obtained. Therefore we need to determine the degrees of freedom of the system and "freeze" some possible directions in order to recognize, in an absolute way, the points and label them univocally.

CARTESIAN ORTHOGONAL COORDINATES

Let us consider a tern with axes named X,Y,Z connected in an orthogonal way as in figure 1:

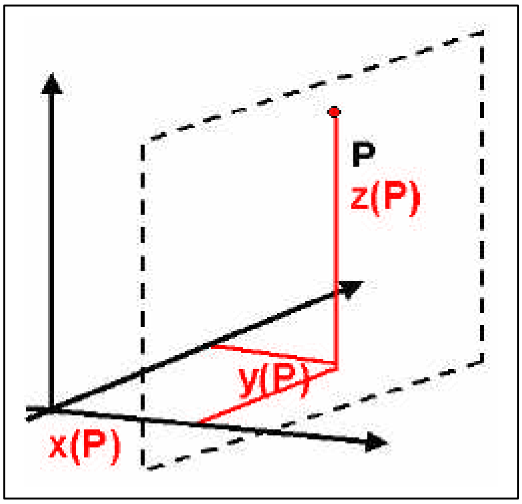


Figure 1: Cartesian orthogonal coordinates

Considering three geometric planes orthogonal to the axis on which the point P lies, we call X(P), Y(P) and Z(P) the Cartesian projections respectively on X,Y,Z axes.
This Cartesian system is often used and without singularities.

SPHERICAL COORDINATES

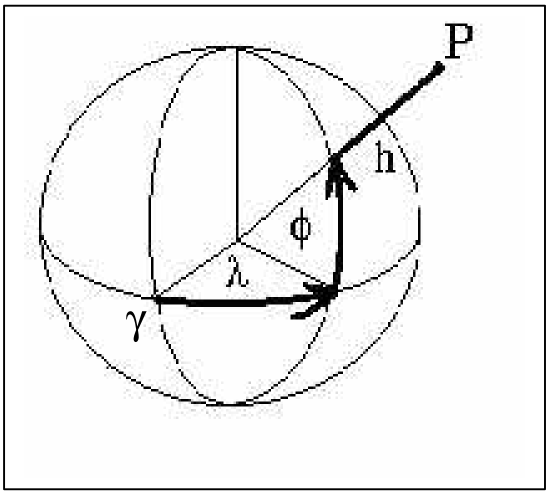


Figure 2: Spherical coordinates

In this second example, instead, we consider a sphere of a given radius, a geometric plane which crosses the center (called equatorial plane) and a point named γ on the equator.
The normal direction is the line from point P to the sphere center.

The latitude ϕ of a point is the angle from the equatorial plane to the normal direction, starting from P to the reference surface.

The longitude λ of a point is the angle between a reference plane per γ and a plane passing through the point, both planes being perpendicular to the equatorial plane whose intersection passes through the sphere's center.

ELLIPSOIDAL COORDINATES

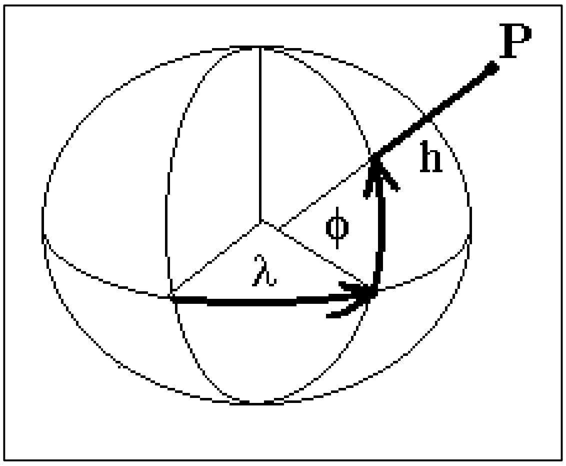


Figure 3: Ellipsoidal coordinates

Here an ellipsoid of rotation is used.
We consider the normal to the ellipsoid from point P. In this case the line does not, generally, cross the ellipsoid center. ϕ, λ are defined analogously to the spherical coordinates and are, respectively, called ellipsoidal latitude and longitude.
The ellipsoidal height of a point is the distance along the “normal” direction from the reference ellipsoid surface.

As figure 3 shows, we call:

- *meridians:* lines where λ is constant, the one with $\lambda=0$ is the Greenwich central meridian.
- *parallels:* lines where ϕ is constant, the one with $\phi=0$ is the equator.
- *geodetic lines:* lines of minimum distance between two points on the ellipsoid. It does not, generally, lie on a geometrical plane and it is a thought task trying to find an easy expression.

For any chosen point P there are two normal lines to the given ellipsoid; this singularity can be, obviously, avoided by choosing the nearest projection.

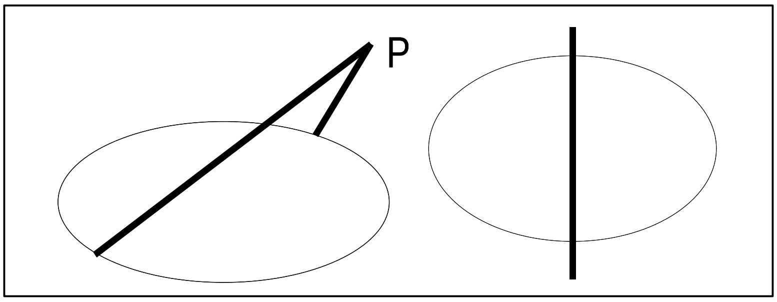


Figure 4: Singularities

The points on the rotation axis have no unique values of λ. This singularity can not be avoided, in this case a point may have different sets of coordinates but each set individuates one and only one point.

How can we choose the Coordinates Systems (C.S.) ?
It must be: -arbitrary
-conventional
and it must have a physical meaning (not only a geometrical one).

Arbitrary: it implies a personal choice.
Conventional: it permits to relate to different C.S. (for example: right hand rule).
It must have a physical meaning since we are dealing with topographical topics applied for engineering purposes.

The degrees of freedom can be determined by constraining an opportune numbers of coordinates of points which have been materialized and labeled before.

3 Degrees of freedom

A Reference Frame (R.F.) must constrain the degrees of freedom whose descriptions have left “unchained” before.
We know, from analytical mechanics, that a shape-retaining body has six degrees of freedom in the three dimensions space (3D): three translations and three rotations.
We must add to these another one referred to the unit of measurement of lengths.

Example of some opportune “bounds” (figure 5):

• Altimetry network	we need to constrain only one suitable parameter such as the height of a point P.
• Planimetry network	we need to constrain three parameters; for example let us consider X and Y as two bars joined in an orthogonal way. We consider that the origin lies on a point named P and the X axis follows the direction PQ: we have either Xp=0, Yp=0, Yq=0 or other three conditions Xp, Yp, α given referred to a certain direction.
• Spatial network	it is similar to the one given before only with six conditions instead of three. For example: one point totally constrained and the heights of other three fixed or, more practically, two totally and the height of another.

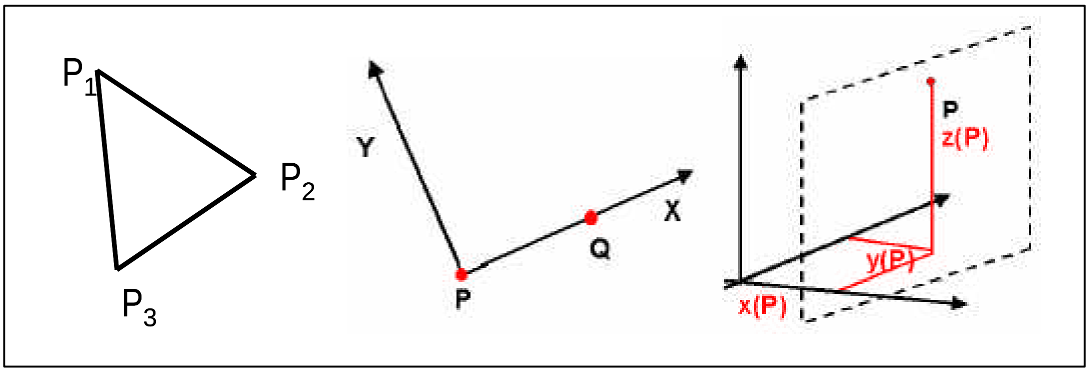


Figure 5: Degrees of freedom in 1, 2 and 3 dimension space

Among these points we are able to create a network of joined points in order to determine the peculiar elements which characterize the Reference Frame (R.F.). With this procedure we have just defined a R.F. in an implicit way.

UNIT OF MEASUREMENTS OF LENGTHS

Since the unit of measurements of lengths is an arbitrary choice the problem can be solved by choosing a sample unit of measurements of lengths or constraining a connection between two points.
Before the GPS and electrodistanziometer age, the topographers been able, in their surveys, to measure angles between the points of a network and only few distances to,

through onerous procedure, block similarly factor. The Italian cartography, for example, used eight connections (baselines) to cover the entire national territory and so introduced, in an implicit way, eight R.F.

4 Altimetry references

Which clues can we use to choose a correct C.S. on the Earth?
At the first go the concept of "height" reminds us to an intuitive idea of "tall" and "short" but since we are dealing with practical problems referred to gravity some definitions are needed. The gravitational field is preservative and allows potential W. In the gravity filed we have:

$$\underline{g} = \left(\frac{\partial W}{\partial X} \quad \frac{\partial W}{\partial Y} \quad \frac{\partial W}{\partial Z} \right) \tag{1}$$

The places where the potential has the same value are called equipotent surfaces, or level surfaces. The gradient vector represents the line of force, it is perpendicular to a level surface and its direction represents the "vertical" (or plumb line). The perpendicular to $\underline{g}$ define the horizontal plane.
Regarding the idea of "short" and "tall" we can try to associate it to the gravitational potential decreases as we move away from the surface. We can conventionally use a "origin surface" where the potential value is W_0 and determine the others as follows:

$$W_0 - W_p = \int_0^P \underline{g} \cdot dh \tag{2}$$

but we can also scaling the equation (2) as follows:

$$C = \frac{\Phi_0 - \Phi_P}{1000} \tag{3}$$

where 1000 is a suitable factor (we could also use other factors like $|\underline{g}|$ or something else connected to its value).
In most of the cases, the altimetry information is taken along the orthometric height; it is the distance from P to P_0 on the reference surface G, along the vertical direction as figure 6 shows.

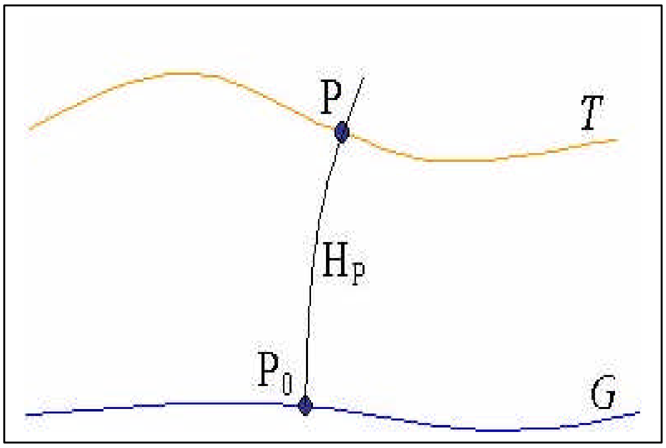


Figure 6: Orthometric height H =(PP_0) along vertical

We call “geoid” the level surface G, which crosses the medium level of seas (oceans). For example, in Italy the mean sea level is referred to the Genova mean sea level. Therefore, the geoid’s surface is the altimetry reference surface.

Can we use it also for the planimetry purposes?
No, because it doesn’t have an easy analytical expression and it will be very a difficult task to relate it for the observed quantities (angles, distances).

5 Planimetry reference

If we can’t use the geoid as a reference surface also for the planimetry purpose, we have to search a simpler analytical shape as similar as possible to the geoid this is the rotational ellipsoid.
Can we use it also for altimetry purposes?
No because we can never confuse ellipsoidal height h with orthometric height H; in other words, the distance from rotational ellipsoid and geoid (geoid undulations N) can not be neglected! The ellipsoid surface may approximate the geoid around ± 100 m (figure 7).
Since they are different R.F two references are needed: one for planimetry and one for altimetry.

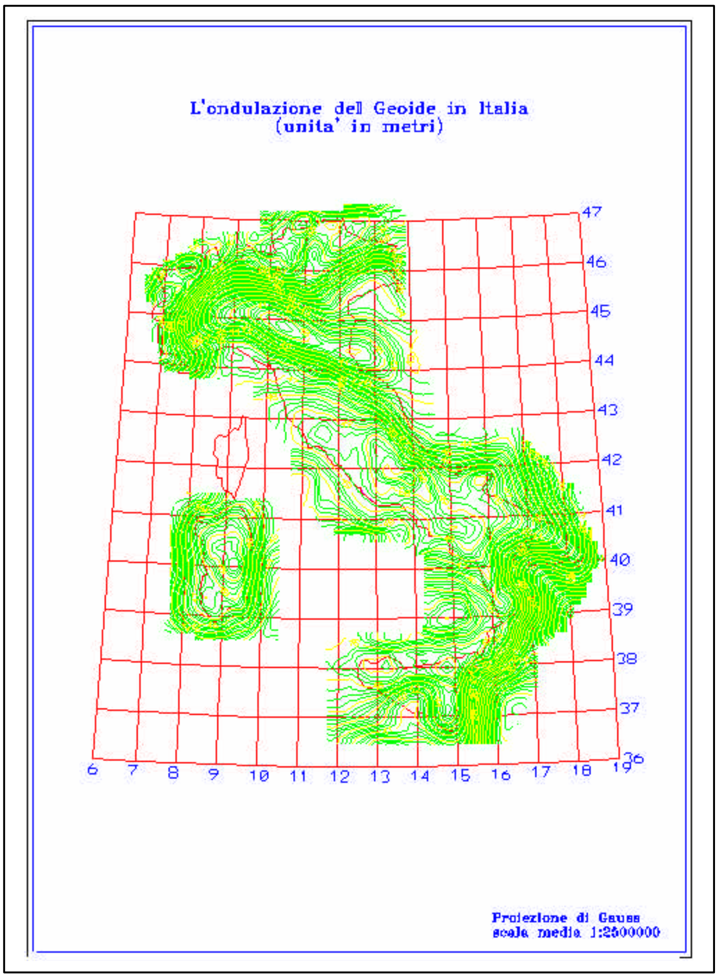


Figure 7: Geoid undulation in the Italian area

Therefore the coordinates we used in the past and still use now are ϕ,λ and H. The first two are referred to planimetry and the last is referred to altimetry.

6 Cartography systems

The cartography representation can be expressed in an analytical way. Choosing a cartography reference means to establish a bivalent relationship between these couples:

$$(f, l) \leftrightarrow (N, E) \tag{4}$$

Where ϕ, λ are connected to the ellipsoid and N and E (North and East) are the orthogonal cartography coordinates. These transformations must be: reversible, with only one value and C^1 (continuous with first derivative continuous).

INEVITABLE TRANSFORMATIONS

Since the ellipsoidal surface cannot be entirely developed on a geometrical plane, some inevitable deformations arise.

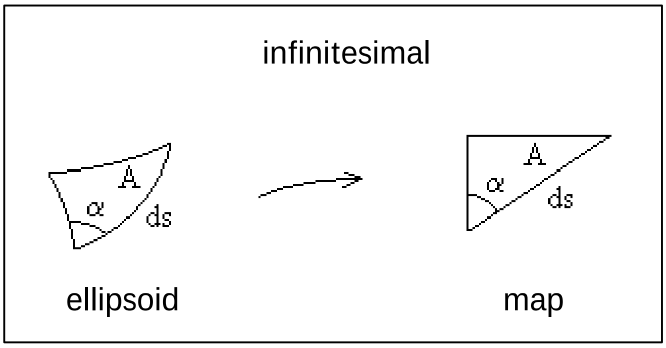


Figure 8: Infinitesimal projection between ellipsoid and map

We call:

- Modulus of linear deformation “m” $m = \dfrac{dS_{map}}{dS_{ellipsoid}}$

  m equals to 1 only on some particular lines on the map (cannot be for all the lines!).

- Modulus of angular deformation “δ” $d = a_{map} - a_{ellipsoid}$

  δ equals to 0 only on so called conformal maps.

- Modulus of areal deformation “μ” $m = \dfrac{dA_{map}}{dA_{ellipsoid}}$

  μ equals to 1 on equivalent maps.

In the conformal (δ=0) maps m=constant in any directions that we choose from point P, but is different in any points (this is true also for points very close to P).
Warning! Unfortunately a cartography representation can not be conformal and equivalent at the same time; in other words, the perfect projection cannot exist.

NOMINAL-SCALE-MAP

The maximal deformations we can tolerate must be smaller than the graphics error: in other words we are not interested in obtaining an accuracy we can not further estimate from the map. These limits of accuracy shall apply in all cases to positions of well-defined points only. Well-defined points are those that are easily visible or recoverable on the ground, such as the following: monuments or markers, such as bench marks, property boundary monuments, intersections of roads, railroads, corners of large buildings or structures (or center points of small buildings), etc. In general what is well defined will be determined by what is plottable on the scale of the map within ~ 0.2 mm. (thickness of a standard draw line). For example:

Maps 1:10.000 we have a planimetric error of 0,2 mm*10.000 = 2 m

Maps 1: 5.000 we have a planimetric error of 0,2 mm* 5.000 = 1 m

Apparently introducing a scale-map does not seem to be very useful while dealing with digital maps since zooms are possible; however we need to think to the accuracy of the results we want to obtain depending on the definition of the previous survey, that in numerical cartography defines the map nominal-scale.

UTM PROJECTION

One can choose in a large set of cartography representations (with corresponding analytical relationships); and between various conventional choices related to the national and/or international cartography organization. An example is: UTM.
UTM stands for Universal Transverse Mercator, it is a cartography system adopting a Gauss projection with a cylindrical confirm representation. That is assimilable to a projection from the ellipsoidal center to transverse cylinder. The main axis crosses the equatorial plane and the cylinder is tangent to a given meridian, called “central meridian”. We apply a correction factor (contraction factor of 0.9996) in order to limit the maps deformations as much as possible. Infact the modulus of linear deformations shows a maximal discrepancy of $4^0/_{000}$ from the unit value for a 6 degrees wide segment, less than the graphics error.
n.b. the cartographical representation of near zones of different segments, can not be joined side by side. However, we can always connect one point to another one which belongs to different sprindles by using the ellipsoidal coordinates ϕ, λ. If a joined representation is needed, it may always be possible to realize a new “personal projection” (using the same “tools” as the previous) with a new central meridian barycentric to the given area.

7 Different Reference Frames

We use different R.F.s mainly because some of them are for planimetrical purposes, others for altimetrical but also because they are applied for different types of observations.
These purposes may be: applications of the orbital motions to world-wide, continental, national or regional cartography.
Different realizations, instead, mean different choices of nets, different tools implemented and different period of observations.
Some examples of R.F.s can be: world-wide net, GPS check-stations, WGS84 a part of which has been augmented by ETRF89 (national map) and subsequently by IGM95. IGM95 is a campaign of measurements realized by IGM (Istituto Geografico Militare Italiano, Italian Military Geographic Institute) using almost every principal point (first order points) surveyed with the traditional procedure, with much better precision [14]. Infact the first method led to a 10-20 cm order of precision; the last, instead, to 5 cm.

DIFFERENT PERIODS OF OBSERVATIONS LEAD TO DIFFERENT REALIZATIONS

Earth rotational axis (I) is subjected to precession, nutation motions (which are due to moon and sun attractions) and also to other little perturbations which are due to the planets motions (figure 9).
The rotation angle is not constant but varies within a very little range and depends upon oscillations connected to the total angular moment of the atmosphere and the oceans.

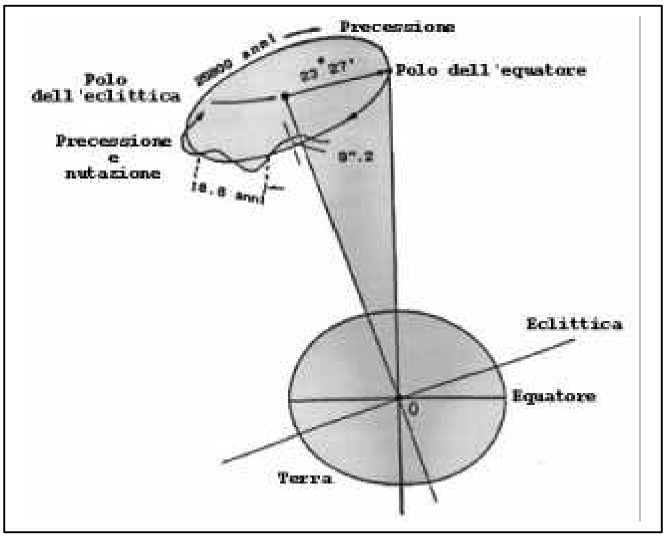


Figure 9: Precession and nutation, Earth motions

If we imagine to materialize axis I inside the Earth we will see that it won't stand still but it would be subjected to the Chandler Wobble factor which includes a gyroscopic component due to the interaction between Earth liquid nucleus and elastic mantle and also to some irregular components which deal with convective motions.

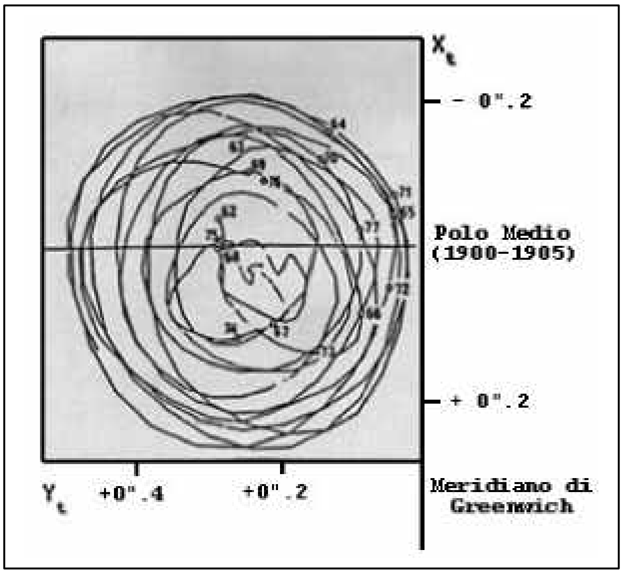


Figure 10: Polar motion (without precession and nutation motion)

Figure 10 shows the pole's motion already purified from precession and nutation effects, we can see that the Earth is not similar to a rigid body, therefore it is very important not to neglect the date of the study. Infact, since the pole moves in different periods of time, we cannot univocally fix the rotation axis but we need to update the R.F. periodically. For this purpose the labels of the coordinates of the permanent control stations include the movement speed of the points, and the acronym of Reference Frame include the reference year like WGS84, Rome40 and so on.

Thus, different realizations of Reference Frame implicitly define different Reference Frames. The coordinates values in different R.F. are obviously different as figure 12 shows.

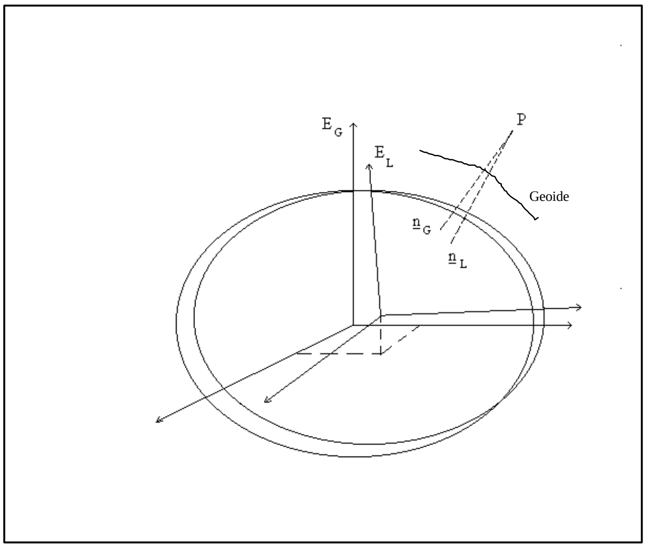


Figure 11: Different datum/different ellipsoidal coordinates

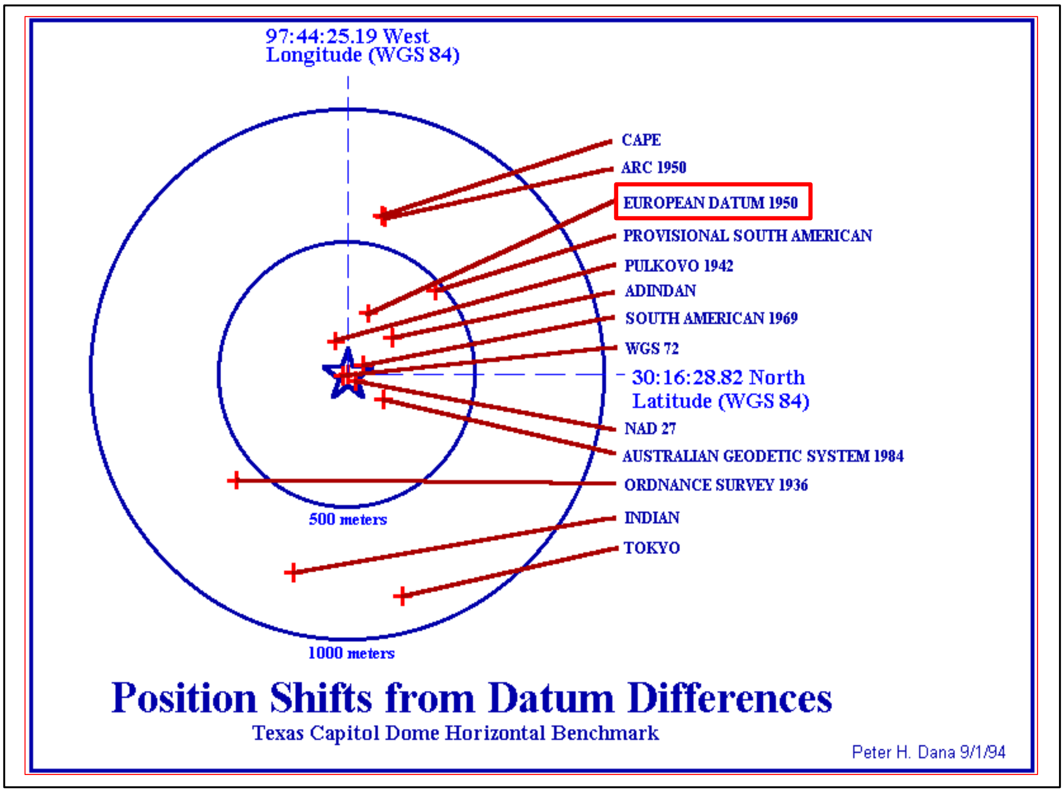


Figure 12: Possible different positions of a same point in different R.F.

8 Transformations

The transformations of coordinates in different R.F. happen in different way for altimetrical and planimetrical purposes.

ALTIMETRICAL DATUM

For the altimetrical case the GPS system allows us to know the “ellipsoidal heights” (typical geometric values), if the undulation N were known it could be possible to evaluate also the geoid height (see par.4).

$$H = h - N \tag{5}$$

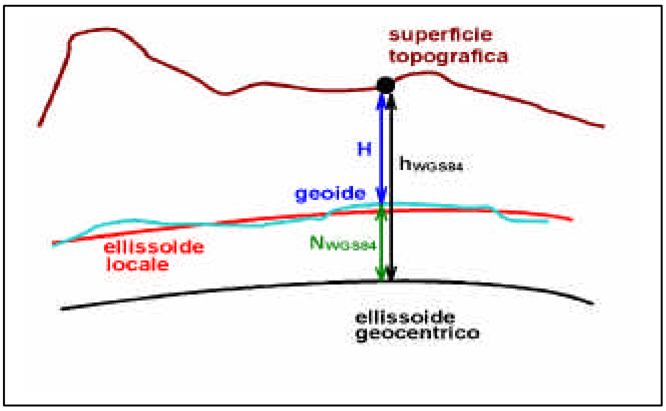


Figure 13: Orthometric and ellipsoidal eights, geoid undulation

For example, in 1999 in Italy IGM obtained a gravimetric geoid estimation called ITALGEO99 [1]. In this case we know the undulations N, therefore equation (5) permits the use of h or H allowing to choose the most convenient altimetrical datum each time.

We call:

H	othometric height as before;
h	ellipsoidal height which represents the distance (along the perpendicular line to the ellipsoid) from point P to the ellipsoid surface;
N	is never to be neglected and cannot be estimated but needs to be known!! Its values varies from –100 to +100 m.

For example, in Italy N values varies from +35m to +55m with differences of 60-80 cm in only 7 km in Pianura Padana (near Milano).

PLANIMETRICAL DATUM

We have made an introduction of different datum and its problems has been made we can deduce that the most important thing is the choice of the R.F. with different sets of data and the knowledge of the parameters of the transformation introduced.

If we don’t have the transformation parameters we first need to estimate them starting from the coordinates of known points in both the R.F.s.

There are some different transformations that can be applied, depending on the accuracy needed and on the acceptable approximations. One is the 3D roto-translation and scale factor (Helmert) between Cartesian coordinates from datum 1 to datum 2, like equation (6) and figure 14:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_2 = l\ R \begin{bmatrix} x \\ y \\ z \end{bmatrix}_1 + \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \qquad (6)$$

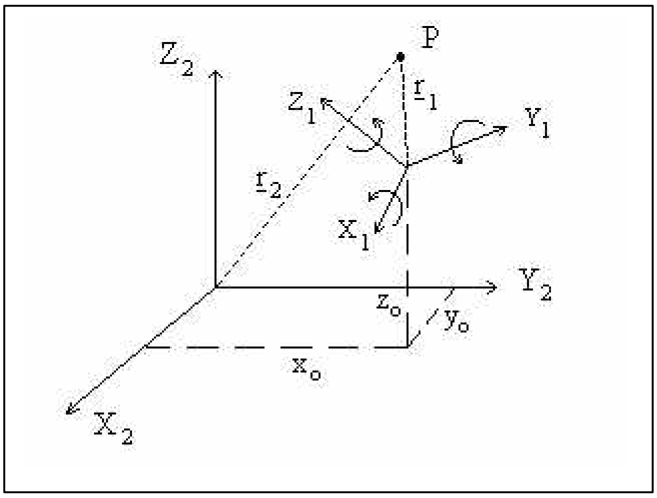


Figure 14: roto-translation from datum 1 to datum 2

The change between R.F. (datum) generally comes from the sequence below:

$$(N,E)_1\ \mathrm{h}_1 \rightarrow (f,l)_1\ \mathrm{h}_1 \rightarrow (x,y,z)_1 \rightarrow \text{HelmertTransf.} \rightarrow (x,y,z)_2 \rightarrow (f,l)_2\ \mathrm{h}_2 \rightarrow (N,E)_2\ \mathrm{h}_2 \quad (7)$$

Therefore we change representation $(N,E)_1 \rightarrow (f,l)_1$ and then we convert the Coordinate System to obtain Cartesian coordinates to use the very simpler Helmert transformation.

The same steps must be carry out to obtain the cartography coordinates $(N, E)_2$. Warning, if we go from a geocentric datum, like WGS84, to a local datum, like the Italian Rome40, the obtained ellipsoidal height h_2 has to be substitute with orthometric height H.

The parameters estimation, instead, is not an easy task because only with the WGS84 system supported by GPS observations we can know the coordinates X,Y,Z of points in a R.F. Infact, in a local survey system we can easily deal with the plane - altimetrical coordinates of points lying on the geodetic national network but it is not generally possible to transform the altimetric information (for example orthometric heights) in ellipsoidal heights of the local system considered.

The geodetic undulations are referred to a geocentric ellipsoid. Therefore, in order to move from ellipsoidal to Cartesian coordinates, some approximations are needed such as:

- Approximating H=h;
- Disregarding the altimetrical information and letting h=0.

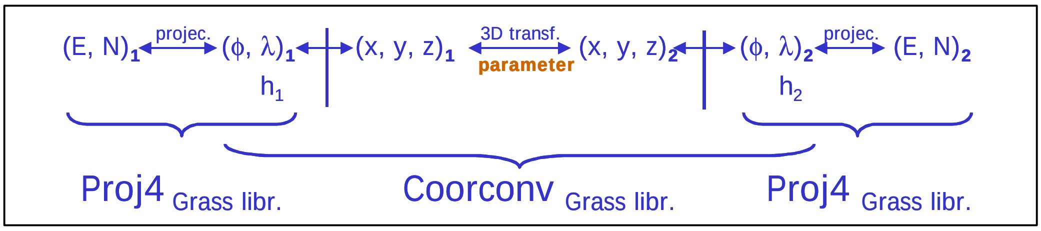
Similar hypotheses can be done by using the Molodensky theory which describes 3-D transformations starting from the ellipsoidal coordinates ϕ, λ, h.
Transformations among ellipsoids which only use planimetrical information in terms of latitude ϕ and longitude λ are possible, obviously with approximations. The number of parameters and the transformations depends on the dimensions of the area considered.

Generally GIS do not include the 3-D transformation and only some times other transformations formulas are available. Generally rasters' dimensions are limited, therefore some people apply a planar transformation directly modeled from $(N,E)_1 \rightarrow (N,E)_2$, for example with a similar or a polynomial transformation, involving much greater approximations.

9 About project and datum transformation in GRASS

GRASS supports a large set of reference frames to manage landscape data of different location on the word, it is also possible to specify a map datum to better describe the geographic information. When defining a new location the user will be prompted to specify the coordinate system (projection). The specification of any projection other than *Latitude-Longitude* and *State Plane* will generate a request of the ellipsoid to the user, (see Appendix A). Then the user will be asked to specify a map datum. If no map datum support is needed, the question should be answered with no. Some projections will generate a request to the user for the prime meridian and standard parallel for the output map. The projection of *State Plane* will generate a request to the user for the choice of the zone for the output map, the *UTM* projection requests to specify ellipsoid and zone information. When register data are not available it is however possible to use a x-y Cartesian coordinate system. GRASS supports databases in a longitude-latitude grid using the "*ll*" projection called Equidistant Cylindrical Projection (ECP) where the x coordinate is the longitude and the y coordinate is the latitude. This projection, also known as "Plate Carree", has the property that "where am I" and "row-column" calculations inside GRASS files are identical to those in planimetric grids, like UTM. This implies that normal GRASS registration and overlay functions work without any special considerations or modifications to existing code. However, the projection is not on a plain so distance and area calculations are no longed Euclidean.

Supported projections within GRASS are listed in Appendix A, as they appear at the GRASS command prompt when a new location is created. To manage them GRASS utilizes the PROJ.4 library developed by Gerald Evenden/USGS (Cartographic Projection Procedures for the UNIX Environment) [17]. PROJ.4 library converts state plane Cartesian coordinates systems to and from geographic longitude and latitude coordinates by means of a wide variety of cartographic projection functions. For many of the projection functions the inverse conversion can also be performed.
The following information are taken from the GRASS 5.0 programmer's manual [16] describing the Coordinate Conversion Library "coorconv". The Coordinate Conversion Library provides functions for latitude-longitude calculations, i.e. projecting latitude-longitude to universal transverse Mercator (UTM) and transverse Mercator (tm), for inverse projection to latitude-longitude, for calculating datum shifts on latitude-longitude coordinates, for scanning coordinates from strings and converting of latitude-longitude coordinates to geocentric coordinates and vice versa.



All map datum known within GRASS must be listed in the map datum database in the file `$GISBASE/etc/datum.table` and the ellipsoid-acronym is a reference to the ellipsoid database in the file `$GISBASE/etc/ellipse.table`. The format of the map datum file is as follows:

```
acronym "description" ellipsoid-acronym dx= dy= dz=
```

The acronym is a short name (single word) that specifies the datum, the description gives a long name and reference to the map datum (enclosed in double quotes). The ellipsoid-acronym is the short name of the ellipsoid used with this map datum and referenced in the `ellipse.table`. dx, dy and dz are the datum shift parameters, which are applied to convert from local map datum to WGS84 datum, only with the Block shift transformation formula [16; par. 10; Appendix B: `m.datum.shift`]. The reverse calculation can be done with signs reversed. Comments are signaled by a '#' at first position of the line, empty lines are discarded. A sample entry:

```
# World Geodetic System 1984
WGS84  "World  Geodetic  System  1984"  WGS84  dx=0.0  dy=0.0
dz=0.0
```

If you need additional map datum you can add them to the file.

The format of the ellipsoid file is as follows:

```
acronym "description" a= x=
```

The acronym is a short name (single word) that specifies the ellipsoid, the description gives a long name and reference to the ellipsoid (enclosed in double quotes). a is the equatorial radius and x is one of the following:

`b= polar radius;`
`f= flattening;`
e^2`= eccentricity squared.`

f, the ellipsoid flattening, is given as a reciprocal value (1/…) (even if this is not signaled by the variable name). The relations between flattening and eccentricity squared is:
$e^2 = [1 - (1 - f)^2]$ because f is defined as $f = (a - b)/a$ and $b = a\ \mathrm{sqrt}(1-e^2)$. A sample entry:

```
WGS84 "World Geodetic System 1984"    a=6378137.000
      f=1/298.257223563
```

Map datum shift on latitude-longitude coordinates can be evaluate with GRASS. Generally the accuracy depends on the transformation method used and the accuracy of the parameters supplied to the transformation function.

In GRASS three different transformation formulas are available giving different accuracy [16]:

Block shift transformation	10 m;
Molodensky transformation	5 m;
Bursa-Wolf transformation[1]	1 m.

All transformations need correct ellipsoid and mathematical parameter to obtain precise results. Parameters estimation it is not yet possible; if no parameters are available, it should be only use Block transformation, which uses the parameters in the GRASS database (`datum.table`).

Obviously parameters for the Molodensky datum shift formula can not be used with any other datum shift formula. The same thing holds for the Bursa-Wolf datum transformation which needs 7 parameters (3 xyz-shift, 3 xyz-rotational, 1 scale factor), as the parameters are not independent from another.

GRASS DISTANCE AND AREA CALCULATIONS

Distance between two points is measured along the geodesic line if the projection is latitude-longitude, through the Euclidean formula otherwise. In this case the area is measured using Euclidean geometry formulas, for maps in raster and vector format. For raster maps in latitude-longitude projection, the area is measured on the ellipsoid. Vector/polygon data is described as a series of x y coordinates. The lines connecting the points are not stored but inferred. This is a simple, straight-forward process for planimetric grids, but it is not simple for latitude-longitude since the shape of the line that connects two points on the surface of an ellipsoid is not rectilinear. GRASS uses a straight line on the grid. Another choice, among many, is the shortest path between the points, known as the geodesic, but this has not yet implemented within GRASS.

10 An example in GRASS

An application has been developed to extend the cartography and datum transformations principles to a real situation in GRASS 5.0.

In a local area of 2 by 2 degrees in the Italian region Trentino Alto Adige, we have used few points of the IGM95 campaign [14] (Istituto Geografico Militare Italiano, I.G.M.I.), and we have generated a regular grid in the cartography coordinates (E, N) with WGS84 datum (defined by one point every 5 km).

Now we suppose to have to transform our data set cartography coordinates from a local datum to a WGS84 reference frame (frequently requested).

These applications took advantages from fictitious DTM placed with height zero to obviate the lack of the ellipsoidal heights in the local system. To obtain more accurate results information on local transformations or on their estimations is needed

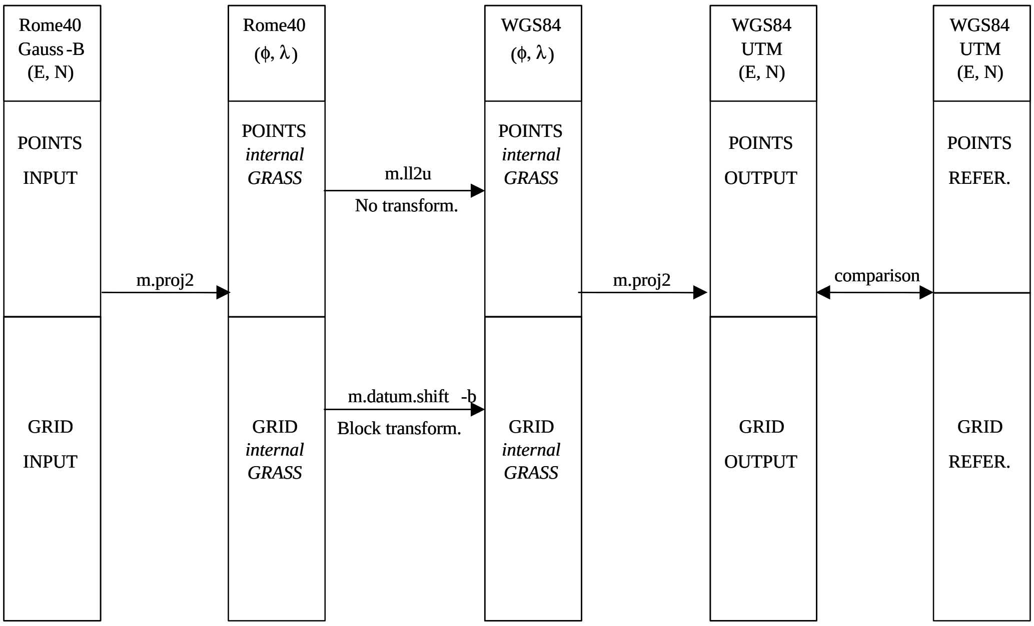
---

[1] This transformation formula is not yet implemented.

Bursa-Wolf is a three dimensional 7-parameter-datum transformation (like Helmert or similarity transformation; different positive rotations definition may be possible), transform 3D Cartesian coordinates. It automatically converts input and output from/to ellipsoidal coordinates. The values for Rx, Ry, Rz and M should be given on the command line, as the coordinate conversion library has no support for this.

Bursa-Wolf is not directly available, the -w option in the `m.datum.shift` module does not work; it is not sufficient uncommented the #define BURSA-WOLF at the top of the `main.c` module source code.

First of all, for our test, we have to transform our cartography grid into the Rome40 datum, using the official transformation parameters of the Regione Autonoma Trentino Alto Adige (Regional Government).



Scheme: datum transformations and comparison

Starting from the location “rome40”:

```
LOCATION:       rome40              MAPSET: maps
CURRENT REGION: N=47N  S=45N  RES=0:00:05  ROWS=1440
                E=12E  W=10E  RES=0:00:05  COLS =1440
PROJECTION: 3 (Latitude-Longitude)
ZONE:      0
DATUM:     rome40
ELLIPSOID: international
```

we have projected the cartography coordinates (E, N) to the ellipsoidal coordinates (ϕ, λ) in Rome40 datum, defining the Italian cartography system Gauss-Boaga, with:

```
m.proj2   (Appendix B)
inproj="proj=tmerc,name=tmerc,a=6378388.000,es=0.0067226700,
unfact=1.0, \
lat_0=0.0,lon_0=9.0,k=0.9996,x_0=1500000.0,y_0=0.0" \
outproj="proj=ll,name=ll,a=6378388.000,es=0.0067226700" \
input=E_N_rome40-gb_IGM-RET.ascii \
output=L_F_rome40_IGM-RET.gout.ascii
```

The data set is formed by 5 IGM-points and by points describing the grid.
Block transformation (3D shift) using GRASS data base parameters, to transform the ellipsoidal coordinates (ϕ, λ) from Rome40 to WGS84 reference frame:

```
m.datum.shift -b   (Appendix B)
is=international id=rome40 os=wgs84 od=wgs84 \
lat=46:MM:SS.DDDDDDN \
lon=10: MM:SS.DDDDDDE >> L_F_wgs84_IGM.gout-3.ascii
```

the commands runs in a batch mode, using an iterative ad hoc routines.
Now the work pass in location “wgs84”:

```
LOCATION:       wgs84              MAPSET: maps
CURRENT REGION:  N=47N  S=45N  RES=0:00:05  ROWS=1440
                E=12E  W=10E  RES=0:00:05  COLS =1440
PROJECTION: 3 (Latitude-Longitude)
ZONE:      0
DATUM:     wgs84
ELLIPSOID: wgs84
```

projecting the ellipsoidal coordinates (ϕ, λ) in WGS84 datum to the standard UTM cartography coordinates (E, N), with:

```
m.ll2u (Appendix B)
spheroid=wgs84 zone=32 \
input=L_F_wgs84_IGM-RET.gout-3.ascii \
output=E_N_wgs84-utm_IGM-RET.gout-3.ascii
```

The input is the output file obtained by the Block transformation.
Now we have supposed not know any datum shift parameters, therefore we try to use our data set coordinates (defined in Rome40 reference frame) directly in the WGS84 datum:

```
m.ll2u
spheroid=wgs84 zone=32 \
input=L_F_rome40_IGM-RET.gout.ascii \
output=E_N_wgs84-utm_IGM-RET.gout-0.ascii
```

that we call “No transformation” solution.
The results have been compared to the official IGM values and the original regular grid both available in the WGS84 – utm reference.

Rome40 – GB [m]		WGS84 - utm No Transf. [m]		WGS84-utm Block Transf. [m]		WGS84 - utm Reference [m]		Differen. No Tr. [m]		Diffferen. Block Tr. [m]	
E	N	E	N	E	N	E	N	E	N	E	N
1619X00	5082X95	619395	5082800	619370	5082882	619X72	5082X74	-23	+74	+2	-8
1710X03	5188X22	710293	5188023	710273	5188107	710X74	5188X98	-19	+75	+2	-9
1646X31	5165X24	646424	5165625	646401	5165709	646X03	5165X00	-21	+75	+2	-9
1676X72	5145X80	676663	5145682	676641	5145765	676X43	5145X56	-21	+74	+2	-9
1678X11	5097X20	678403	5097224	678381	5097306	678X82	5097X97	-21	+74	+1	-9
						Points	**mean**	**-21**	**+74**	**+2**	**-9**
							rms	**1**	**1**	**0.3**	**0.6**
1615028	5190024	615022	5189924	614998	5190009	615000	5190000	-22	+76	+2	-9
1615028	5185024	615022	5184924	614998	5185009	615000	5185000	-22	+76	+2	-9
1615028	5180024	615022	5179925	614998	5180009	615000	5180000	-22	+75	+2	-9
…	…	…	…	…	…	…	…	…	…	…	…
1705030	5080023	705020	5079927	704999	5080009	705000	5080000	-20	+73	+1	-9
1710030	5080023	710020	5079927	709999	5080009	710000	5080000	-20	+73	+1	-9
1715030	5080023	715020	5079927	714999	5080009	715000	5080000	-20	+73	+1	-9
						Grid	**mean**	**-21**	**+74**	**+2**	**-9**
							rms	**1**	**1**	**0.5**	**0.5**

Table 1: Comparison of the transformation

The test shows differences lower then 10 m for Block transformation, like GRASS Programmer's Manual writes; whereas an order of magnitude more for the No transformation.
Obviously these results are not extensible for other areas. They are depending on ellipsoid and local datum considered, especially for the No transformation applications.

11 Conclusions

GRASS 5.0 supports a large set of Reference Frame (R.F.) and cartography systems to manage landscape data of different location on the word.
In different projections it allows the transformation between cartography systems (in other words, between different projections) using the relationships between ellipsoidal coordinates and projection parameters in Proj4 [17]. It also allows the transformation among different R.F. with different “formulas”. These transformations, however, require the previous knowledge of the parameters expected for the most simple transformations, the Block formula, that take in to account only the shift transformation between different datums. Parameters providing an accuracy of 10 meters are available in the GRASS database for a lot of R.F.
It has been made a representative example in a 2 degree by 2 degree area in the Italian region Trentino Alto Adige to transform from a local system Rome40 to WGS84 GPS system (par. 10). The test has pointed out the approximation limit of the Block transformation and, especially, the error magnitude we commit supposing that WGS84 and Rome40 are identical (No transformation). These applications do not consider the height transformation to obviate the lack of the ellipsoidal heights in the local system.
The test shows differences lower then 10 m for Block transformation, in accordance with GRASS Programmer's Manual while “No transformation” with errors higher of an order of magnitude, as expected. A value of 10 meters may be comparables with the map’s minimum definition (graphics error) at the 1:25.000 ÷ 1:50.000 scale.
Obviously the results obtained are not extensible for other areas. They are depending on ellipsoid and local datum, especially for the No transformation applications.

References

[1] Barzaghi R., Betti B., Borghi A., Sona G., Tornatore V. The Italian quasi-geoid ITALGEO99. In Bollettino di Geodesia e Scienze Affini, n.1, 2002.

[2] Betti B., Crespi M., Sansò F. GPS, geoide e sistemi di riferimento. In Ricerche di Geodesia, Topografia e Fotogrammetria, vol. 5, Clup, Milano, 1989.

[3] Birardi G. Corso di Topografia. Pitagora, Bologna, vol.3, 1982.

[4] Bitelli G., Unguendoli M. Inquadramento di reti GPS in un sistema locale. In Bollettino SIFET, n.4, 1994.

[5] Coticchia A., Surace L. La trasformazione delle coordinate dal sistema UTM (ED1950) al Sistema Cassini-Soldner (catastale). Laboratorio di Topografia e Fotogrammetria, Facoltà di Ingegneria, Università degli Studi di Firenze, 1984.

[6] DMA WGS84 Development Committee. Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems. DMA TR 8350.2, Defense Mapping Agency, Washington DC - USA, 1987.
[7] Inghilleri G. Topografia generale. UTET, Torino, 1974.

[8] Monti C., Sansò F. Esercizi di topografia, geodesia e cartografia. Clup, Milano, 1976.

[9] Moritz H. Geodetic reference system 1980. Bulletin Géodésique, vol.62, n.3, 1988.

[10] Pierozzi M. Alcune considerazioni sulla trasformazione dal sistema WGS84 ad un sistema geodetico locale. In Bollettino di Geodesia e Scienze Affini, n.1, 1989.

[11] Pierozzi M., Surace L. I parametri di trasformazione tra il sistema WGS84 ed il sistema geodetico nazionale Roma40. In Bollettino di Geodesia e Scienze Affini, n.1, 2000.
[12] Sacerdote F. Richiami di geodesia finalizzata al posizionamento satellitare. Collana di Geodesia e Cartografia CISM *"Il sistema di posizionamento globale satellitare"*, CISM, Udine, 1990.

[13] Sansò F. Annotazioni su GPS e sistemi di riferimento. Collana di Geodesia e Cartografia CISM *"Il sistema di posizionamento globale satellitare"*, CISM, Udine, 1990.

[14] Surace L. Il progetto IGM95. In Bollettino di Geodesia e Scienze Affini, n.3, 1993.

[15] Surace L. La georeferenziazione delle informazioni territoriali. In Bollettino di Geodesia e Scienze Affini, n.2, 1998.

[16] GRASS Programmer's Manual
http://grass.itc.it/grass5/progmangrass50.pdf

[17] PROJ4 (developed until 1995 by USGS)
http://www.remotesensing.org/proj/

[18] For a description of the Bursa-Wolf transformation see:
http://www.posc.org/Epicentre.2_2/DataModel/ExamplesofUsage/

[19] A full description of the Molodensky transformation is found at:
http://www.anzlic.org.au/icsm/gdatm/gdav2.2.pdf

Appendix A

```
To create a new LOCATION starting GRASS 5.0, you will need
the following information:

1.  The coordinate system for the database
   x,y (for imagery and other unreferenced data)
   Latitude-Longitude
   UTM
   Other Projection
2.  The zone for the UTM database and all the necessary
   parameters for projections other than Latitude-Longitude,
   x,y, and UTM
3.  The coordinates of the area to become the default region
   and the grid resolution of this region
4.  A short, one-line description or title for the location


Do you have all this information for location <tst_domenico>
? (y/n)


Please specify the coordinate system for location
<tst_domenico>

A   x,y
B   Latitude-Longitude
C   UTM
D   Other Projection
RETURN to cancel


Please specify projection name
ll -- Lat/Lon
```

[n.b.: ll – Lat/Lon is not a projection but coordinates defined on the ellipsoid!]

```
utm -- Universe Transverse Mercator
stp -- State Plane
aea -- Albers Equal Area
lcc -- Lambert Conformal Conic
merc -- Mercator
tmerc -- Transverse Mercator
leac -- Lambert Equal Area Conic
laea -- Lambert Azimuthal Equal Area
aeqd -- Azimuthal Equidistant
airy -- Airy
aitoff -- Aitoff
alsk -- Mod. Stererographics of Alaska
apian -- Apian Globular I
august -- August Epicycloidal
bacon -- Bacon Globular
bipc -- Bipolar conic of western hemisphere
boggs -- Boggs Eumorphic
bonne -- Bonne (Werner lat_1=90)
cass -- Cassini
cc -- Central Cylindrical
cea -- Equal Area Cylindrical
chamb -- Chamberlin Trimetric
collg -- Collignon
crast -- Craster Parabolic (Putnins P4)
```

denoy -- Denoyer Semi-Elliptical
eck1 -- Eckert I
eck2 -- Eckert II
eck3 -- Eckert III
eck4 -- Eckert IV
eck5 -- Eckert V
eck6 -- Eckert VI
eqc -- Equidistant Cylindrical (Plate Caree)
eqdc -- Equidistant Conic
euler -- Euler
fahey -- Fahey
fouc -- Foucaut
fouc_s -- Foucaut Sinusoidal
gall -- Gall (Gall Stereographic)
gins8 -- Ginsburg VIII (TsNIIGAiK)
gn_sinu -- General Sinusoidal Series
gnom -- Gnomonic
goode -- Goode Homolosine
gs48 -- Mod. Stererographics of 48 U.S.
gs50 -- Mod. Stererographics of 50 U.S.
hammer -- Hammer & Eckert-Greifendorff
hatano -- Hatano Asymmetrical Equal Area
imw_p -- International Map of the World Polyconic
kav5 -- Kavraisky V
kav7 -- Kavraisky VII
labrd -- Laborde
lagrng -- Lagrange
larr -- Larrivee
lask -- Laskowski
lee_os -- Lee Oblated Stereographic
loxim -- Loximuthal
lsat -- Space oblique for LANDSAT
mbt_s -- McBryde-Thomas Flat-Polar Sine (No. 1)
mbt_fps -- McBryde-Thomas Flat-Pole Sine (No. 2)
mbtfpp -- McBryde-Thomas Flat-Polar Parabolic
mbtfpq -- McBryde-Thomas Flat-Polar Quartic
mbtfps -- McBryde-Thomas Flat-Polar Sinusoidal
mil_os -- Miller Oblated Stereographic
mill -- Miller Cylindrical
mpoly -- Modified Polyconic
moll -- Mollweide
murd1 -- Murdoch I
murd2 -- Murdoch II
murd3 -- Murdoch III
nell -- Nell
nell_h -- Nell-Hammer
nicol -- Nicolosi Globular
nsper -- Near-sided perspective
nzmg -- New Zealand Map Grid
ob_tran -- General Oblique Transformation
ocea -- Oblique Cylindrical Equal Area
oea -- Oblated Equal Area
omerc -- Oblique Mercator
ortel -- Ortelius Oval
ortho -- Orthographic
pconic -- Perspective Conic
poly -- Polyconic (American)
putp1 -- Putnins P1
putp2 -- Putnins P2
putp3 -- Putnins P3
putp3p -- Putnins P3'
putp4p -- Putnins P4'
putp5 -- Putnins P5
putp5p -- Putnins P5'
putp6 -- Putnins P6
putp6p -- Putnins P6'
qua_aut -- Quartic Authalic
robin -- Robinson
rpoly -- Rectangular Polyconic
sinu -- Sinusoidal (Sanson-Flamsteed)
somerc -- Swiss. Obl. Mercator
stere -- Stereographic
tcc -- Transverse Central Cylindrical
tcea -- Transverse Cylindrical Equal Area
tissot -- Tissot
tpeqd -- Two Point Equidistant
tpers -- Tilted perspective
ups -- Universal Polar Stereographic
urm5 -- Urmaev V

urmfps -- Urmaev Flat-Polar Sinusoidal
vandg -- van der Grinten (I)
vandg2 -- van der Grinten II
vandg3 -- van der Grinten III
vandg4 -- van der Grinten IV
vitk1 -- Vitkovsky I
wag1 -- Wagner I (Kavraisky VI)
wag2 -- Wagner II
wag3 -- Wagner III
wag4 -- Wagner IV
wag5 -- Wagner V
wag6 -- Wagner VI
wag7 -- Wagner VII
weren -- Werenskiold I
wink1 -- Winkel I
wink2 -- Winkel II
wintri -- Winkel Tripel

```
Please specify ellipsoid name
sphere
airy
andrae
apl4.9
aust_sa
australian
bess_nam
bessel
clark66
clark80
cpm
delmbr
engelis
everest
everest_m
everest_p
evrst30
evrst48
evrst56
evrst69
evrstss
fschr60
fschr60m
fschr68
grs67
grs75
grs80
hayford
helmert
hough
iau76
indonesian
international
intl
kaula
krassovsky
lerch
merit
modif_airy
mprts
new_intl
nwl9d
plessis
SAD-69
sam69
seasia
sgs85
walbeck
wgs60
wgs66
wgs72
wgs84
```

```
Do you want to specify a map datum for this location?(y/n) [n]
Please specify datum name
datum
a-can
agd66
agd84
aus
carthage
eur
eur50
eur79
grs80
nad27
nad83
osgb36
potsdam
pulkovo
rome40
S-42
SAD-69
sam69
Sasia
tokyo
wgs72
wgs84
```

ELLIPSOID PARAMETERS IN GRASS 5.0:

```
# this file contains ellipsoid parameters
# format of the file is
#  name  "description"     a=value   x=value
#
# where name is a single word elliposid name,
# description is a text enclosed in double quotes,
# a is the equatorial radius
# and x is one of
#   b polar radius, f flattening, or e eccentricity squared
#
# x should be a defining parameter (as opposed to a derived
parameter)
#  note:  accuracy  should  be  sufficient  with  a  derived
parameter.
# All newer sources state only f as a parameter, even if it
is derived.
# b / a = 1 - f = sqrt( 1 - e2 ) = 1 / sqrt ( 1 + e'2 )
#
# no spaces before or after the =


# the following parameters are from NIMA document TR8350.2:
# old values commented
# Airy 1830
# airy                                   a=6377563.396
      e=.006670540
airy        "Airy 1830"            a=6377563.396
      f=1/299.3249646
# Australian National
australian  "Australian National"        a=6378160.000
      f=1/298.25
# Bessel 1841, Ethiopia, Indonesia, Japan and Korea
# bessel                                 a=6377397.155
      e=.006674372
bessel            "Bessel 1841"                a=6377397.155
      f=1/299.1528128
# Bessel 1841, Namibia
bess_nam    "Bessel 1841, Namibia"       a=6377483.865
      f=1/299.1528128
# Clarke 1866
# clark66                          a=6378206.4 b=6356583.8
clark66           "Clarke 1866"                a=6378206.400
      f=1/294.9786982
# Clarke 1880
clark80           "Clarke 1880"                a=6378249.145
      f=1/293.465
# Everest 1830, India
# everest                          a=6377276.345     e=.0066378466
everest           "Everest 1830, India"        a=6377276.345
      f=1/300.8017
evrst30     "Everest 1830, India"              a=6377276.345
      f=1/300.8017
# Everest 1948, W. Malaysia and Singapore
evrst48     "Everest 1948"                     a=6377304.063
      f=1/300.8017
# Everest 1956, India
```

```
evrst56           "Everest 1956, India"         a=6377301.243
      f=1/300.8017
# Everest 1969, W. Malaysia
evrst69           "Everest 1969, W. Malaysia"   a=6377295.664
      f=1/300.8017
# Everest (Sabah & Sarawak), Brunei and E. Malaysia
evrstss           "Everest Sabah and Sarawak"   a=6377298.556
      f=1/300.8017
# Geodetic Reference System 1980 (IUGG, 1980)
grs80       "Geodetic   Reference   System   1980"   a=6378137.000
      f=1/298.257222101
# Helmert 1906
helmert           "Helmert 1906"                a=6378200.000
      f=1/298.3
# Hough 1960
# hough                                   a=6378270.0
      b=6356794.343479
hough       "Hough 1960"                  a=6378270.000
      f=1/297.0
# Indonesian 1974
indonesian  "Indonesian 1974"       a=6378160.000     f=1/298.247
# International 1924
international     "International 1924"          a=6378388.000
      f=1/297.0
# Krassovsky 1940
krassovsky  "Krassovsky 1940"       a=6378245.000     f=1/298.3
# Airy 1830 Modified
# modif_airy                              a=6377340.189
      b=6356036.143
modif_airy  "Airy 1830 Modified"          a=6377340.189
      f=1/299.3249646
# South American 1969
sam69       "South American 1969"         a=6378160.000
      f=1/298.25
# Brazil South American 1969
SAD-69      "South American 1969"               a=6378160.000
      f=1/298.25
# WGS 1972
wgs72       "World Geodetic System 1972"  a=6378135.000
      f=1/298.26
# WGS84 as defined by NIMA document TR8350.2
wgs84       "World Geodetic System 1984"  a=6378137.000
      f=1/298.257223563



# from original file:
# GRS75
grs75       "GRS75"                 a=6378140.000     b=6356755.288
# Hayford (same as International 1909)
hayford           "Hayford"               a=6378388.000
      f=1/297.000


# additional ellipsoids from PROJ4.4.1, file pj_ellps.c
# Appl. Physics. 1965
apl4.9            "Appl. Phys. 1965"            a=6378137.0
      f=1/298.25
# Australian Natl & S. Amer. 1969
aust_sa           "Australian Natl & S. Amer. 1969"   a=6378160.000
      f=1/298.25
# Andrae 1876 (Den., Iclnd.)
```

```
andrae            "Andrae 1876"                 a=6377104.43
      f=1/300.0
# Comm. des Poids et Mesures 1799
cpm         "Comm. des Poids et Mesures 1799"
                                    a=6375738.7 f=1/334.29
# Delambre 1810 (Belgium)
delmbr            "Delambre 1810"               a=6376428.0
      f=1/311.5
# Engelis 1985
engelis           "Engelis 1985"                a=6378136.05
      f=1/298.2566
# Fischer (Mercury Datum) 1960
fschr60           "Fischer (Mercury Datum) 1960"     a=6378166.000
      f=1/298.3
# Mercury (same as Fischer 1960). Included only for reference.
# mercury   "Mercury"               a=6378166.0 b=6356784.283666
# modif_merc is a synonym to Modified Fischer 1960 (fschr60m).
Included only for reference.
# modif_merc      "Modified Mercury"            a=6378150.0
      b=6356768.337303
# Modified Fischer 1960
fschr60m    "Modified Fischer 1960"       a=6378155.000
      f=1/298.30
# Fischer 1968
fschr68           "Fischer 1968"                a=6378150.000
      f=1/298.3
# GRS 67 (IUGG 1967)
# GRS67                             a=6378160.000    b=6356774.516
grs67       "GRS67"                 a=6378160.000
      f=1/298.2471674270
# IAU 1976
iau76       "IAU 1976"              a=6378140.000    f=1/298.257
# International 1909 (Hayford)
intl        "International 1909 (Hayford)"      a=6378388.000
      f=1/297.000
# Kaula 1961
kaula       "Kaula 1961"                  a=6378163.0 f=1/298.24
# Lerch 1976
lerch       "Lerch 1976"                  a=6378139.0 f=1/298.257
# Maupertius 1738
mprts       "Maupertius 1738"       a=6397300.0 f=1/191.0
# MERIT 1983
merit       "MERIT 1983"                  a=6378137.000
      f=1/298.257
# New International 1967
new_intl    "New International 1967"                 a=6378157.500
      b=6356772.2
# This has same values as as WGS66
# Naval Weapons Lab., 1965
nwl9d       "Naval Weapons Lab., 1965"    a=6378145.000
      f=1/298.25
# Plessis 1817 (France)
plessis           "Plessis 1817"                a=6376523.0
      b=6355863.0
# Southeast Asia
seasia            "South East Asia"       a=6378155.000
      b=6356773.3205
# Soviet Geodetic System 85
sgs85       "Soviet Geodetic System 1985" a=6378136.000
      f=1/298.257
# Walbeck
```

```
walbeck     "Walbeck"                   a=6376896.000
      b=6355834.8467
# WGS60
wgs60       "WGS60"               a=6378165.000    f=1/298.3
# WGS66
wgs66       "WGS66"               a=6378145.000    f=1/298.25

# Modified Everest (from elist.html from Peter H. Dana, cited from
DMA 1987
# everest_m "Modified Everest"          a=6377304.063
      b=6356103.039
everest_m   "Modified Everest"          a=6377304.063
      f=1/300.8017

# new from ?
# Everest (Pakistan)
everest_p   "Everest (Pakistan)"        a=6377309.613
      f=1/300.8017
# Earth as a sphere:
# There is a bug in libes/gis/get_ellipse.c, where a function can
not
# return e=0.0 in any case. So do not use sphere!
# sphere    "Spherical Earth"           a=6370997.000    e=0.0
```

DATUM PARAMETERS IN GRASS 5.0:

```
# this file contains datum shift parameters,
# it supplies a datum database table for GRASS 5.0
# format:
# shortname "description" ellipsoid dx= dy= dz=
#
# where short name is a single word datum specifier,
# description is a long name (enclosed in double qoutes),
# ellipsoid is the acronym of the ellipsoid used with this
map datum,
# dx, dy and dz are the datum shift parameters.
# no spaces allowed before or after the =.
#
# Please comment and cite the source if you add new
parameters
# to this list.


# cited from NIMA document TR8350.2
# World Geodetic System 1984
wgs84  "World Geodetic System 1984"     wgs84   dx=0.0  dy=0.0
                             dz=0.0
# World Geodetic System 1972
wgs72  "World Geodetic System 1972"     wgs72   dx=0.0  dy=0.0
                             dz=5.0
# North American 1927
nad27  "North American 1927" clark66    dx=-22.0        dy=157.0
                             dz=176.0
# North American 1983 (CONUS)
nad83  "North American 1983" grs80      dx=0.0  dy=0.0  dz=0.0
# Alaska and Canada
a-can  "Alaska and Canada NAD27"        clark66 dx=-9.0 dy=151.0
                             dz=185.0
# European datum
```

```
eur    "European"           international      dx=-84.0   dy=-
103.0                       dz=-127.0
# Tokyo datum mean
tokyo  "Tokyo mean"         Bessel      dx=-128.0     dy=481.0
                            dz=664.0
# Australian Geodetic
aus    "Australian Geodetic" australian  dx=-122.0     dy=-41.0
                            dz=146.0
# Ordnance Survey of Great Britain (1936)
osgb36 "Ordnance Survey of Great Britain" airy  dx=368.0   dy=-
120.0                       dz=425.0
# South American 1969 (= SAD-69/Brazil)
sam69 "South American 1969"  sam69       dx=-77.0      dy=3.0
                            dz=-45.0
SAD-69 "SAD-69/Brasil"      SAD-69      dx=-60.0      dy=-2.0
                            dz=-41.0
# Pulkovo 1942 ( = S-42)
pulkovo "Pulkovo 1942"      krassovsky  dx=28.0 dy=-130.0  dz=-
95.0
# European 1950 mean
eur50 "European 1950 mean"  international      dx=-87.0   dy=-98
                            dz=-121
# European 1979 mean
eur79 "European 1979 mean"  international      dx=-86  dy=-98
                            dz=-119
# AGD66 Australian Geodetic 1966 [from NIMA TR8350.2]
agd66 "Australian Geodetic 1966"        Australian     dx=-133
                            dy=-48      dz=148
# AGD84 Australian Geodetic 1984 [from NIMA TR8350.2]
agd84 "Australian Geodetic 1984"        Australian     dx=-134
                            dy=-48      dz=149
# GDA94 (nearly the same as wgs84)
#  International  Union  of  Geodesy  and  Geophysics  Geographic
Reference System 1980
grs80 "Geographic Reference System 1980" wgs84  dx=0.0  dy=0.0
                            dz=0.0
# Rome 1940
rome40 "Rome 1940"          international      dx=-225 dy=-65
                            dz=9
# from Geodetic Datum List, Peter H. Dana, 07/12/97, dlist.html
# South Asia
Sasia "South Asia"          fschr60m    dx=7.0  dy=-10.0   dz=-
26.0
# S-42 Pulkovo 1942
S-42 "S-42"                 krassovsky  dx=28.0 dy=-121.0  dz=-
77.0
# Potsdam Rauenberg 1950 DHDN
potsdam "Potsdam Rauenberg 1950 DHDN"  bessel   dx=606.0   dy=23.0
                            dz=413.0
# Carthage 1934 Tunisia
carthage "Carthage 1934 Tunisia"        clark80 dx=-263.0  dy=6.0
                            dz=431.0
```

Appendix B

A briefly describe of GRASS modules related to projection and map datum follows.

See also GRASS Programmer's Manual
http://grass.itc.it/grass5/progmangrass50.pdf

m.datum.shift

datum shift program

```
      m.datum.shift  lat=dd.mm.ss{n|s}  lon=dd.mm.ss{e|w}  h=height
above ellipsoid
      id=input_datum od=output_datum
      is=input_spheroid   os=output_spheroid   dx=xshift   dy=yshift
dz=zshift

      -b use block shift method
      -m use Molodensky formula
      -w use Bursa Wolf 7 parameter transformation
```

It returns geographic coordinates based on a different datum spheroids/ellipsoid than the one used to obtain the original coordinates. The input and output spheroids, -is- and -os-, are the spheroids for the two different datums.

The input spheroid is the one on which the original coordinates are based. The output spheroid is that on which the resultant coordinates will be based. The "shifting" occurs between the two datums. The shift values, -dx-, -dy-, and -dz-, are constants. They indicate the mean differences between points in the second datum versus the first as measured in meters. If both input and output datum id and od are listed in the system datum table, it is sufficient to provide input and output datum for the datum shift. The shift values, -dx-, -dy- and -dz- are read from the datum table. The list of datums and spheroids available is somewhat dynamic. The height above the ellipsoid is usually not known in GRASS. You should approximate this by zero (default for h). Obviously the resulting height is not a reasonable value.

Essentially, the program follows these steps for the block shift method. The original point, as defined by a latitude and a longitude, is converted to geocentric coordinates. The shift values are added to the geocentric coordinates. The summed values are then converted to latitude and longitude based on the output spheroid. The Molodensky method uses a one-step calculation without converting to and from geocentric coordinates. The Molodensky formula may be inaccurate for latitudes near the poles. The coordinate conversion library will take this into account and use the block shift formula for those latitudes. The Bursa-Wolf formula converts original point latitude and a longitude to geocentric coordinates, applying a Helmert 7-parameter transformation to geocentric coordinates and then converted to latitude and longitude based on the output spheroid

m.geo

calculates conversion coordinates for geographic positions

```
      m.geo
      m.geo help
```

It allows the user to interactively: convert projection coordinates Northings and Eastings to Latitude and Longitude values, or to interactively: convert Latitude and Longitude values to projection coordinate Northings and Eastings. It allows the user to do all of the above: reading from a file, writing to the screen, or reading from the keyboard, writing to a file, or reading from a file, writing to a file.

The program does not transform GRASS files, it is designed to determine coordinate values on an individual position.

When reading from a file of LATITUDE/LONGITUDE data the file will contain three columns of information:

the first column - latitude - in degrees minutes seconds,
the second column - longitude - in degrees minutes seconds,
the third column - zone - zero(0) if not required.

For example:

```
+40     36  31.4563     -87      2    7.8193    16
  40n   36  31.4563       87w   2    7.8193     16
```

When reading from a file of PROJECTION COORDINATES data the file will contain three (3) columns of information:

the first column - easting - ground coordinates
the second column - northing - ground coordinates
the third column - zone - zero(0) if not required.

For example:

```
500000.00   4496918.64  16    <- utm
-424489.11  1908736.13   0   <- lambert
```

No column headings are required, just the numbers.

m.proj

calculates conversion coordinates for geographic positions

```
m.proj
m.proj help
```

It allows the user to interactively: convert coordinates from one projection to another.
It allows a user to do all of the above:
reading from a file, writing to the screen, or
reading from the keyboard, writing to a file, or
reading from a file, writing to a file,

The program does not transform GRASS files, it is designed to determine coordinate values on an individual position.

When reading from a file of LATITUDE/LONGITUDE data the file will contain 2 columns of information:

the first column - latitude - in degrees minutes seconds,
the second column - longitude - in degrees minutes seconds,

For example:

```
+40     36   31.4563    -87      2   7.8193
  40n   36   31.4563          87w   2   7.8193
```

When reading from a file of PROJECTION COORDINATES data the file will contain 2 columns of information:

the first column - easting - ground coordinates
the second column - northing - ground coordinates

For example:

```
  500000.00      4496918.64
 -424489.11      1908736.13
```

No column headings are required, just the numbers.

m.proj2

calculates conversion coordinates for geographic positions

```
m.proj2 [inproj=name[,name,...]] [outproj=name[,name,...]]
        [input=name] [output=name]
```

It allows the user to convert coordinates from one projection to another. It allows a user to convert coordinates from a file, writing to a file. The program does not transform GRASS files, it is designed to determine coordinate values on an individual position.

Parameters:
inproj
Comma separated input projection parameters
outproj
Comma separated output projection parameters
input
Input coordinate file
output
Output coordinate file

The input format is the same ofS m.proj module.
Projection parameters have to be provided: "proj" (projection type), "name" (projection name), "a" (ellipsoid: equatorial radius), "es" (ellipsoid: eccentricity squared), "zone" (zone for the area), "unfact" (conversion factor from meters to other units, e.g. feet), "lat_0" (standard parallel), "lon_0" (central meridian), "k" (scale factor) and "x_0" (false easting). Sometimes false northing is needed which is coded as "y_0". Internally, the underlying PROJ 4 projection library performs an inverse projection to latitude-longitude and then projects the coordinate list to the target projection.

m.region.ll

converts Universal Transverse Mercator (UTM) coordinates falling within the current geographic region from UTM coordinates to geographic (latitude-longitude) coordinates.

```
m.region.ll
m.region.ll help
m.region.ll spheroid=name
```

It takes current geographic region settings in UTM coordinates, and converts them to geographic coordinates (i.e., latitudes and longitudes). It also prints the length (in meters) of one arc-second at each of the four edges of the geographic region. The user must enter the spheroid upon which to base the geographic coordinates.

m.setproj

allows the user to create the PROJ_INFO and the PROJ_UNITS file to record the projection information associated with a specified mapset

```
m.setproj
m.setproj help
m.setproj [set=name] proj=name
```

It allows the user to create a new PROJ_INFO file in the specified mapset. The file is used to record the projection information associated with the specified mapset which must not contain a PROJ_INFO or PROJ_UNITS file.

Parameters:
set=name
Mapset in which the projection information file is to be stored.
proj=name
Map projection name.
Options: utm, aea, stp, ll, lcc, merc, tmerc, xxx

g.setproj

allows the user to create the PROJ_INFO and the PROJ_UNITS file to record the projection information associated with a current location.

It allows the user to create a PROJ_INFO file in the PERMANENT mapset of the current location. PROJ_INFO file is used to record the projection information associated with the specified mapset.

g.projinfo

displays the PROJ INFO and UNITS file of current location.

```
g.projinfo
g.projinfo help
```

It reports information about the projection, coordinate system ellipsoid, datum, zone of the current location as well as the projection units.

m.gc2ll

converts geocentric to geographic coordinates

```
m.gc2ll x=# y=# z=# s=spheroid
```

It returns geographic coordinates for geocentric ones supplied by the user. It performs the reverse operation of the GRASS program m.ll2gc. The x, y and z values are the three dimensions needed to locate a point in three-dimensional space. The values that are printed include the latitude, the longitude and the height above (or distance below) the spheroid. If the spheroid desired is not on the list used by GRASS, the values for the semi-major axis and the eccentricity squared for the spheroid may be entered in place of a spheroid name in the following format:

```
s=a=semi-major_axis,e=eccentricity_squared
```

m.ll2gc

converts geographic coordinates to geocentric

```
m.ll2gc lat=dd.mm.ss{n|s} lon=dd.mm.ss{e|w} [h=height]
       s=spheroid
```

It returns geocentric coordinates for geographic coordinates (latitude and longitude). Geographic coordinates are in degrees, minutes, and seconds and must include designation of north or south {n|s} and east or west {e|w}. The spheroid on which they are based must also be entered. The height (in meters) above the spheroid is optional.

If the spheroid desired is not on the list used within GRASS, the values for the semi-major axis and the eccentricity squared for the spheroid may be entered in place of a spheroid name in the following format:

```
s=a=semi-major_axis,e=eccentricity_squared
```

This is an experimental program. It is part of initial efforts to incorporate geographic coordinates into GRASS.

m.ll2u

converts geographic coordinates to Universal Transverse Mercator (UTM) coordinates.

```
m.ll2u
m.ll2u help
m.ll2u [-rwoz] spheroid=name [zone=value] [input=name]
       [output=name]

m.ll2u converts geographic coordinates (i.e., latitudes and
longitudes)  to  Universal  Transverse  Mercator  (UTM)
      eastings and northings.
```

The user must specify the spheroid on which to base the UTM conversion. The user may optionally specify the UTM zone; however, the program can determine this from the geographic coordinates submitted.

Input can be entered from the keyboard or from an input file. In either case, input should be entered with one longitude and latitude pair per line, in either of the below forms:

```
degrees:minutes:seconds{E|W} degrees:minutes:seconds{N|S}
degrees:minutes:seconds{E|W} degrees:minutes:seconds{N|S}
degrees:minutes:seconds{E|W} degrees:minutes:seconds{N|S}
degrees:minutes:seconds{E|W} degrees:minutes:seconds{N|S}
.
.
end

degrees.decimal{E|W} degrees.decimal{N|S}
degrees.decimal{E|W} degrees.decimal{N|S}
degrees.decimal{E|W} degrees.decimal{N|S}
degrees.decimal{E|W} degrees.decimal{N|S}
.
.
end
```

If the user sets the -r flag, m.ll2u will expect the order of the coordinates to be reversed, and stated as latitude, longitude pairs, rather than as longitude, latitude pairs. Similarly, the user can elect to send output to an output file or (by default) to standard output (the user's terminal screen). If the user sets the -w flag, output will be printed in a format suitable for input to programs like d.points.

Flags:
-r
The order of coordinates is reversed in the input, and entered as: lat lon
-w
Do not flag invalid lon,lat input lines as errors.
-o
Flag other invalid input lines as errors.
-z
Suppress printing of the UTM zone in the output. (Note. This will produce output in a format suitable for direct input to programs like d.points.)

Parameters:
spheroid=name
Reference spheroid (ellipsoid).
Options: airy, australian, bessel, clark66, everest, grs80, hayford, international, krasovsky, wgs66, wgs72, wgs84
zone=value
UTM zone number (results will be forced into this UTM zone).
Options: 1-60
input=name

Name of an existing input file containing longitude, latitude coordinates to be converted. Input lines may either be input in the form of degrees:minutes:seconds, or as decimal degrees. If input as decimal degrees, m.ll2u recognizes negative numbers.
output=name
Name to be assigned to the output file containing UTM coordinates and zone designations.

m.u2ll

converts Universal Transverse Mercator (UTM) coordinates to geographic (latitude-longitude) coordinates

```
m.u2ll
m.u2ll help
m.u2ll [-srwod] spheroid=name [zone=value] [input=name]
       [output=name]
```

It converts Universal Transverse Mercator (UTM) northings and eastings to geographic coordinates (i.e., latitudes and longitudes). The user must specify the UTM coordinates to be converted and the spheroid on which the geographic coordinates will be based. The program also needs to know the UTM zone in which the input coordinates are located.
However, if the user is running GRASS from a UTM data-base LOCATION, m.u2ll will use this data-base's UTM zone designation, if no zone is specified by the user. Input can be entered from the keyboard or from an input file. In either case, input should be entered with one UTM easting and northing pair per line, in the format shown below:

easting northing easting northing easting northing easting northing...

Flags:
-s specified UTM zone is in the southern hemisphere.
-r he order of coordinates is reversed in the input, and entered as: north east. This option allows the user to pass an ASCII vector file through m.u2ll.
-w Do not flag invalid east, north input lines as errors.
-o Flag other invalid input lines as errors.
-d Output latitude/longitude values in decimal degrees, rather than in the form dd:mm:ss.

Parameters:
spheroid=name
Reference spheroid (ellipsoid).
Options: airy, australian, bessel, clark66, everest, grs80, hayford, international, krasovsky, wgs66, wgs72, wgs84
zone=value
UTM zone in which UTM coordinates are located.
Options: 1-60
input=name
Name of input file containing UTM values to be converted.
output=name
Name to be assigned to output file containing longitude and latitude values.

r.in.ll

converts raster data referenced using latitude and longitude coordinates to a UTM-referenced map layer in GRASS raster format

```
r.in.ll
r.in.ll help
r.in.ll    [-s]    input=name    output=name    bpc=value
corner=corner,lat,lon  dimension=rows,cols  res=latres,lonres
spheroid=name
```

It converts raster data referenced using latitude and longitude coordinates to a UTM-referenced map layer in GRASS raster format. r.in.ll is primarily used as the final program in converting DTED and DEM digital elevation data to GRASS raster format, but is not limited to this use.
r.in.ll uses the user's current geographic region settings.
Only data that falls within the current geographic region will appear in the final raster map layer.

Flags:
-s
Signed data (high bit means negative value).

Parameters:
input=name
Name of an existing input raster map layer.
output=name
Name to be assigned to the output raster map layer.
bpc=value
Number of bytes per cell.
corner=corner,lat,lon

One corner latitude and longitude of the input.
Format: {nw|ne|sw|se},dd:mm:ss{N|S},ddd:mm:ss{E|W}

The latitude and longitude are specified as dd.mm.ssH where dd is degrees, mm is minutes, ss is seconds, and H is the hemisphere (N or S for latitudes, E or W for longitudes).
For example, to specify the southwest corner: corner=sw,46N,120W
Note: the latitude and longitude specified are for the center of the corner cell.
dimension=rows,cols
Number of rows and columns in the input file.
res=latres,lonres
Resolution of the input (in arc seconds).
spheroid=name Name of spheroid to be used for coordinate conversion.
Options: airy, australian, bessel, clark66, everest, grs80, hayford, international, krasovsky, wgs66, wgs72, wgs84

The user should also note that the raster map layer imported into GRASS will be based on the current geographic region settings. The boundaries of this geographic region should therefore be checked before importing the raster map layer.
Data outside of the geographic region will not be imported and missing data will be assigned the category value "no data".

r.proj

allows the user to re-project a raster map from one location to the current location

```
r.proj [-l] input=name location=name [output=name] [mapset=name] [dbase=name]
        [method=name] [resolution=value]
```

It projects a raster map in a specified mapset of a specified location from the projection of the input location to a raster map in the current location. The projection information is taken from the momentary PROJ_INFO files. In the module man page more related information are provided.

Flags:
-l
List raster files in input location and exit

Parameters:
input
input raster map from source location
location
source location of input map
output

output raster map for current location
mapset
mapset of input map
dbase
path to GRASS database of input location
method
interpolation method to use
options: nearest,bilinear,cubic
default: nearest
resolution
resolution of output map

v.proj

allows projection conversion of vector files

```
v.proj
v.proj help
v.proj [-sl] input=name location=name [dbase=name]
[mapset=name] [out=name]
```

It allows the user to convert a vector map in a specified mapset of a specified location (different from current) with projection of input location to the vector map in a current mapset of current location with projection of current location (both projections are defined by corresponding PROJ_INFO files). The module will create dig, dig_att, and dig_cats directories in the output mapset, if they do NOT exist. Map files for dig, dig_att, and dig_cats are also created for the new map layer.

Flags:
-s
Automatically run "v.support" on newly created vector file.
-l
List vector files in input location and exit

Parameters:
input=name
Name of the input vector map.
location=name
Name of the location containing input vector map
dbase=name
path to GRASS database of input location
mapset=name
Name of the mapset containing input vector map
output=name
Name of the output vector map.