Some Conformal Mappings and Transformations for Geodesy and Topographic Cartography

Skrifter 4. række, bind 6
Publications 4. series, volume 6

by Knud Poder and Karsten Engsager
Geodetic Division, KMS
Denmark 1998

Kort & Matrikelstyrelsen

Some Conformal Mappings and Transformations for Geodesy and Topographic Cartography

Skrifter 4. række, bind 6
Publications 4. series, volume 6

by Knud Poder and Karsten Engsager
Geodetic Division, KMS
Denmark 1998

Kort & Matrikelstyrelsen

Foreword

The present publication comes from the daily work and environment of its authors: Practical geodetic computations for geodesy and topographic cartography. The target group of readers are people in our own profession, i.e. colleagues using coordinates for their work. The aim is therefore to give the background for simple implementations of algorithms with high and known numerical precision. The main topic is the promotion of the conformal transversal cylinder mapping (Gauss-Krüger, UTM, etc.) with safe and simple algorithms working even at the poles of the ellipsoid.

Two very different books have been our inspiration, (1) "Mathematische Grundlagen der höheren Geodäsie und Kartographie" by R. König and K.H. Weise (1951), and (2) "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie (1988). The first book is maybe "mathematisch altmodisch", but extremely precise and thorough (apparently almost void of misprints). The other is apparently simple, but it is an outstanding good tutorial introducing a complete programming language with examples which "make the reader think".

We have out of necessity maintained the rather old-fashioned mathematical form with much less thoroughness and in spite of the inspiration coming from the book on C in our daily work no direct C-algorithms are given here.

We shall with pleasure thank all colleagues, who have contributed with corrections and suggestions for improvement.

Technical Note

The manuscript has been written with Word Perfect (TM). The formulae are mostly given in frames numbered in parentheses, e.g. (2.4) = the fourth frame in Chapter 2. Individual lines in a frame are referred to with capital letters, e.g. (3.7.A).

Input/output parameters to algorithms in frames are often indicated by arrows → and ← as a help for the reader.

Angular units are assumed to be in radians, so that the rather clumsy conversion factors to angular units are not found in the formulae. All linear units are assumed to be in metres or dimensionless. The actual algorithms based upon the formulae are always using radians and metres, so that the conversion from or to any other unit, which may be convenient for the user, is done by the input/output functions.

The concept *scale* will refer to mapping in "natural size", so that one metre on the ellipsoid will mostly be mapped as nearly one metre. (The Mercator mapping and mapping to the Gaussian sphere are the exceptions). In order to get a *publication scale* of M, e.g. 1:25000, all coordinates must be multiplied with M, or the equatorial radius of the mapped ellipsoid should be multiplied with M.

Some of the transformation formulae are not so easily derived, but they are far more easy and precise in the use than some of the easily derived and most known formulae. We have tried to reduce the volume of the text by only giving some hints of the development of the formulae, which frequently requires substitution of one series into another, even in more than one level. All details may be found in the first volume of R. König und K.H. Weise: Mathematische Grundlagen der höheren Geodäsie und Kartographie (1951), in the sequel referred to as K&W. Unfortunately the second volume was never published. Another approach (leading to the same result) is given by L. Krüger, 1912.

National Survey and Cadastre, Denmark is in the sequel referred to as **KMS**.

CONTENTS

1. Introduction

The geometric geodesy uses many different coordinate systems - both for convenience and out of necessity. The safe and fast handling of position information - coordinates - is vital for professional geodetic and topographic users, but also other professional users requiring such information as a support for their field of work. The present work treats only some few conformal mappings, which are used in geodetic computation and topographic maps. The non-conformal mappings, genuine projections etc. are not interesting for such work, but have applications elsewhere.

1.1 Transformations and Predictions

From a geodetic point-of-view we have 3 classes of transformations and predictions.

1. Regular mapping: The mapping from the real 3-d world and/or a reference ellipsoid to and from simple (mostly plane) 2-d surfaces, all on the same datum. All formulae may be derived from mathematics alone. The name *mapping* is deliberately used instead of *projection*, because there is no general geometrical interpretation of the relation between the point on or near the ellipsoid and its position on the mapping surface.

2. Prediction: The mapping between two coordinate systems with lacking or incomplete datum definition for one or both. The formulae are developed empirically from known coordinates in both systems. This is in principle a pure numerical problem of finding numerical functional relations between corresponding numerical coordinate values. The name *prediction* is used instead of *transformation* in order to indicate that it is an approximation found from observations, not a mathematical exact derivation.

3. Datum shift: Mostly a linear transformation between two cartesian coordinate systems, based upon 3 to 7 parameters and - of course - observation data. (Boucher et al. (1989), Boucher et al. (1992), Ashkenazi et. al. (1996)). Differential formulae may be used, but they are not attractive from a numerical point-of-view and not convenient at high latitudes. A datum shift may be regarded as a prediction, but there is a more clear physical background for it than just a list of corresponding coordinates from different and maybe less clear sources. However, in some cases the methods used for prediction may even give a much better datum shift when used directly. The reduction of GPS data to/from various epochs may be carried out with the algorithms used for datum shifts, because it is in reality a datum shift.

The inverse mapping, i.e. from the mapping surface to the ellipsoid is just as useful as the direct one, not least when using the ideas of

(1) dual transformation functions, where the same function (= procedure or subroutine) can transform both to and from the mapping surface, and

(2) a parent and child structure of the transformation processes,

as described by Poder, (1989 and 1992).

1.2 Definitions and Names

The following four items are meant for a clarification of some concepts in connection with coordinate transformation, but admittedly more precision may be needed from a geodetic view on datums, reference frames and systems.

1. Datum and reference frame: Definition of the position, scale and orientation of a network or array of stations from observable quantities. Datum was the classical concept, where the geodetic ("horizontal") coordinates on the chosen ellipsoid referred to a horizontal datum, see e.g. Moritz (1978). A vertical datum was used for the heights, in principle independent of the horizontal datum. A reference frame is defined with 3-d coordinates, see e.g. Boucher et al. (1995), and could be independent of an ellipsoid. However, an ellipsoid associated with such a frame is needed if one wants to use the mappings described in the present work.

2. Reference Ellipsoid: A direct mapping from 3-d coordinates to a 2-d surface would not be convenient if we want the scale of the mapping to appear almost constant. Projecting the actual points in space on an ellipsoid maintains the actual scale almost constant, so the mapping to a Euclidean surface will now be easy. A reference ellipsoid for mapping and transformation applications requires only 2 parameters defining (1) the size and (2) the shape. General geodesy requires furthermore 2 parameters for the ellipsoid and the intrinsic definition that the ellipsoid is a rotation ellipsoid. Appendix II contains information of how to get the subset of (1) equatorial radius and (2) flattening from the otherwise preferable 4 parameter definition.

3. Mapping (or projection): A relation between the 3-dim coordinates and/or the geodetic coordinates and the coordinates in the mapping space (mostly a plane, but mapping on a sphere may also occur). The relation between the 3-d cartesian coordinates and geodetic coordinates (supplemented by a height) is in reality also a mapping.

4. Coordinate system: A coordinate system is defined by a datum (which includes the reference system and an ellipsoid) and a mapping. Some coordinate systems are (possibly out of ignorance) rather incompletely defined, so that e.g. the relations to geodetic coordinates do not exist.

As a general principle, names of variables related to the ellipsoid have small letters. In order to avoid confusion with concepts from physical geodesy, the word *geodetic* is used instead of *ellipsoidal*. In the mappings the north-going coordinate is N, Y, or y and the east-going coordinate is E, X, or x, although some x may be west-going. Geodetic coordinates and spherical ones are positive towards north and east. Some "equation engineering" is left to a user with other sign conventions.

It should be noted that subscripted variable names in series expansions are selected on a mnemonic basis. E.g. e^2 is the squared first eccentricity of the ellipsoid while e_2 is a coefficient in a trigonometric series expansion giving Gaussian latitude from geodetic latitude.
The corresponding spherical coordinates are named with corresponding capital letters, just opposite the principles found in K&W. However, the final linear coordinates in the mappings are called N and E. It is generally assumed that the origo of the coordinate system has rectangular coordinates (0, 0) and longitudes are relative to a central meridian, so that subtraction or addition of e.g. the 500 km added to UTM eastings and the Greenwich longitude of the central meridian is handled by the input/output process.

(1.1) Coordinate Designations

$$
\begin{aligned}
\phi &= \textit{Geodetic latitude}\\
p &= \pi/2 - \phi = \textit{Geodetic co-latitude or polar distance}\\
\lambda &= \textit{Geodetic longitude} \qquad ;\ \textit{difference from a central longitude}\\
\phi_c &= \phi_r + i\,\phi_i = \textit{Complex geodetic latitude}\\
p_c &= \pi/2 - \phi_r - i\,\lambda = \textit{Complex geodetic co-latitude}\\
\psi &= -\ln\left(\tan(p/2)\left(\frac{1+e\cos p}{1-e\cos p}\right)^{e/2}\right)\\
&= \ln\left(\tan(\pi/4 + \phi/2)\left(\frac{1-e\sin\phi}{1+e\sin\phi}\right)^{e/2}\right) = \textit{Isometric latitude}\\
\psi_c &= \psi + i\,\lambda = \textit{Complex isometric coordinates}\\
u &= y + i\,x = \textit{Complex (normalized) mapping coordinates}\\
N &+ i\,E = \textit{Complex (metric) mapping coordinates}\\
z &= \exp(i\,\phi_c) = i\exp(-i\,p_c) = \textit{Exponential coordinates}\\
dz &= iz\,d\phi_c = -iz\,dp_c = \textit{Differential of exponential coord.}
\end{aligned}
$$

The name *complex latitude* is taken from K&W, who even called the complex isometric coordinates *complex longitude* and the transversal conformal coordinates *complex meridian arc*. They also used the greek letters A(=M), B, and Γ for the mappings of Mercator (isometric coord.), Breite (=latitude), and Gauss-Krüger, which we consider as a form of mathematical humour.

1.3 Survey and Strategy

The mappings treated in the present work are a very little fraction of possible mappings, but they are nevertheless some of the most commonly used mappings, and in fact the conformal transversal ("cylinder") mapping is the only really practical and attractive one, if geodetic computations should be carried out in an Euclidean geometry.

It is our opinion that a very high computation precision, exceeding largely the physical precision of the coordinates, is very desirable when automatic data processing is used. Neglecting this would mean that the coordinate values from a transformation would depend heavily on the computing algorithm, so that "new" results could emerge from the pure manipulation of the representation system. The contemporary de facto standard of 14-16 decimal digit computation precision makes this intention feasible.

Chapter 2 introduces the transversal mapping and the co-axial mappings using the original differential equations to give normalized coordinates, i.e. coordinates without metric units. The first Gaussian fundamental form is presented with simplifications, because the coordinate systems are isometric and orthogonal.

Chapter 3 introduces the Gaussian sphere, which is used as a parametric sphere, which enables us to (1) do away with the complex latitude, (2) solve the inverse mapping problem (i.e. finding the latitude and longitude from rectangular coordinates) very easily, and (3) show a shortcut to transformation between the regular mapping coordinates. The name "Gaussian latitude" is possibly an idiom used at the Geodetic Institute, e.g. in the Geodetic Tables (Andersen, Krarup and Svejgaard, 1956). We have here generalized this concept to both the coordinates and the surface.

Chapter 4 deals with the transversal mapping and produces formulae which are reliable and robust, permitting transformations at arbitrarily high latitudes (including the poles) and a zone width of up to 9000 km with a precision of mostly less than 0.015 mm and never more than 0.1 mm on the ellipsoid.

Chapter 5 treats the co-axial mappings, which are simple complex functions of the complex isometric latitude.

Chapter 6 completes the regular mappings by dealing with 3-d coordinates and geodetic coordinates + ellipsoidal heights. This chapter and Chapter 7 are included in order to show how two regular coordinate systems on different datums or reference frames can be mapped, but no thorough discussion on datum shifts is given.

Chapter 7 describes predictions of coordinates by means of empirically determined general polynomials using a very large number of coordinate pairs.

Chapter 8 shows datum shifts via 3-d coordinates, which permits very consistent transformations, not obtainable by differential formulae. However, the formulae shown here should in practise be more specific for the sequence of rotation and scaling.

Chapter 9 describes some of the principles used in the implementation of transformation functions (subroutines) and transformation programmes.

The appendices contain (I) a description of the Clenshaw summation, (II) ellipsoid parameters, (III) general formulae for orientation, and (IV) the structure of a so-called coordinate label.

2. Conformal Mappings

The mappings treated in the present work are all conformal mappings. Consequently the word "conformal" (sometimes also called "orthomorphic") in tacitly understood, so that e.g. *transversal mapping* means *conformal transversal mapping*.

2.1 Introduction and Thesis

Thesis: A mapping of the ellipsoid for geodetic applications should be a rigorous representation of a part of the ellipsoid on a plane enabling computations with any desired accuracy when the observations also are mapped precisely on the space consisting of the mapping plane.

It is rather common to regard a mapping as an approximation of the nature, but it is a more fertile view to look at a mapping as a precise representation of the curved (non-Euclidean) surface of the ellipsoid on a plane, Euclidean surface.

A consequence is that the observations also must be mapped into the mapping space (i.e. the plane). The geodesic - satisfying Clairauts formula - on the ellipsoid is mapped into a curved line satisfying Schols's formula for the curvature. It is in fact the geodesic in the mapping space, when the (varying) scale is used in the metric. In practise corrections are applied to the observations of directions and distances, so that a simple Euclidean geometry can be used.

(2.1) Conformal Mapping

Geodetic coordinates:	ϕ, λ	*(geodetic latitude and longitude)*
Isometric coordinates:	ψ, λ	*(isometric latitude and longitude)*
Mapped coordinates:	y, x	*(y→north, x→east)*
General mapping functions:	$y = y(\psi, \lambda)$	$x = x(\psi, \lambda)$
Cauchy-Riemann:	$\frac{\partial y}{\partial \psi} = j\frac{\partial x}{\partial \lambda}$	$\frac{\partial x}{\partial \psi} = -j\frac{\partial y}{\partial \lambda}$
(Orientation parameter:	$j = +1$ *or* $-1)$	
Conformal mapping:	$y + ix = f(\psi + i\lambda)$	
Inverse conf. mapping:	$\psi + i\lambda = f^{-1}(y + ix)$	

A complex function of a complex variable satisfying the Cauchy-Riemann differential equations is called regular or analytic, meaning that it can be expanded in a convergent series. *It is a* *nearby thought, that any of such complex functions can be used for conformal mapping* - they are of course not necessarily useful in practise. But one can take advantage of the theory for complex functions when treating conformal mappings, instead of speaking of projection cylinders, cones, and planes in normal, transversal, or skew positions.

The orientation parameter *j* takes care of the orientation of the coordinate axes. Any odd number of changes of the orientation of the coordinate axes (ellipsoid and/or mapping) switches the value of *j*, but retains the analytic properties of the function. The coordinates y and x can be multiplied with a common constant without disturbing the Cauchy-Riemann equations.

2.2 Conformal Mappings

The number of conformal mapping kinds is here limited to just two:

1. The transverse conformal mapping ("transverse cylinder projection"), mostly known as "UTM - Universal Transversal Mercator" or "Gauss-Krüger transversal conformal cylinder projection". It is in fact the only really attractive mapping for geodetic computations.

2. The co-axial, conformal mappings, where the 3 types have the normal "Mercator projection" as the generating function. The mappings are known under the names of (A) "Mercator projection", (B) "Lambert conical projection", and (C) "Hipparchos or Stereographic projection". The polar stereographic mapping can in fact be treated as a special case of the Lambert conical mapping.

The Mercator mapping is a useful tool for the production of the formulae for the transverse conformal mapping.

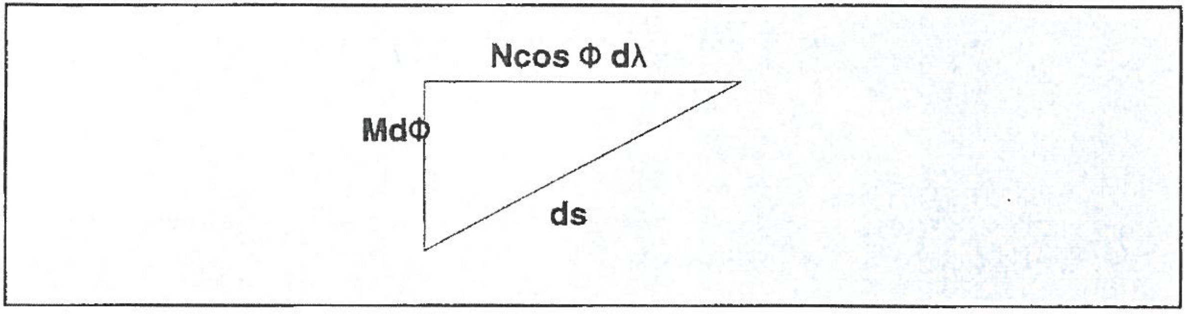
The conical mapping is not very interesting for geodetic computations, because corrections for direction and distance cannot easily be computed.

The stereographic mapping may be used near the poles of the ellipsoid, but the transversal mapping as presented here works perfect even at the poles. It turns out that the polar stereographic can be computed as just an ordinary conical mapping.

2.3 The First Gaussian Fundamental Form

A differential line element *ds* on the ellipsoid with its components of latitude and longitude shown in Fig. 2.1 is related to these components by the so-called first Gaussian fundamental form. See Appendix III for a general orientation of ellipsoid parameters.

Fig. 2.1 The Differential Line Element



The expression (2.2.A) for the ds^2 is a simplified version of the fundamental form, which here presupposes that the coordinate axes are orthogonal. The form can be further simplified by means of isometric coordinates (coordinates with the same, but not necessarily constant, metric in each point) instead of using geodetic latitude and longitude.

(2.2) The First Gaussian Fundamental Form

$$\begin{aligned} ds^2 &= M^2(\phi)\, d\phi^2 + N^2(\phi) \cos^2\phi\, d\lambda^2 \\ &= N^2(\phi) \cos^2\phi \left(\frac{M^2(\phi)}{N^2(\phi) \cos^2\phi}\, d\phi^2 + d\lambda^2 \right) \end{aligned}$$

$$d\psi = \frac{M(\phi)}{N(\phi) \cos\phi}\, d\phi \quad \textit{and} \quad r(\phi) = N(\phi) \cos\phi \quad \textit{gives:}$$

$$ds^2 = r^2(\phi)\, (d\psi^2 + d\lambda^2) \qquad (A)$$

$$ds = r(\phi)\, (d\psi + i\, d\lambda) \qquad (B)$$

$$\Re(ds) = r(\phi)\, d\psi \qquad \Im(ds) = r(\phi)\, d\lambda$$

The expression (2.2.B) with the *ds* as a complex number is in principle not the Gaussian fundamental form, but it is very convenient for the simple mappings in the present work. The product of *ds* with is complex conjugate is of course the Gaussian fundamental form as given in the beginning.

The isometric latitude and the longitude form the most fundamental mapping of the (rotation-) ellipsoid, and we shall use them as the theoretical gateway to the other mappings. The integration of the differential relation between the isometric latitude and the geodetic latitude provides a definition of the isometric latitude, but in practise a much more convenient approach is used.

2.4 The Mapping Derivative

The derivative of a mapping function can be found as the ratio of the complex differential with modulus m and argument g of the mapping function to the complex differential of isometric latitude and longitude. The simplification is possible due to the orthogonality and isometry of both coordinate systems.

(2.3) The Derivatives of the Mapping Function

$$du = dy + i\,dx \qquad \textit{(Mapping fundamental form)}$$

$$ds = r(\phi)(d\psi + i\,d\lambda) \qquad \textit{(Geodetic fundamental form)}$$

$$\mu = m \exp(ig) = \frac{du}{ds}$$

$$dy = \frac{\partial y}{\partial \psi}d\psi + \frac{\partial y}{\partial \lambda}d\lambda \qquad dx = \frac{\partial x}{\partial \psi}d\psi + \frac{\partial x}{\partial \lambda}d\lambda$$

$$\mu = \frac{\left(\frac{\partial y}{\partial \psi} + i\frac{\partial x}{\partial \psi}\right)d\psi + i\left(\frac{\partial x}{\partial \lambda} - i\frac{\partial y}{\partial \lambda}\right)d\lambda}{r(\phi)\,(d\psi + i\,d\lambda)}$$

Using the Cauchy-Riemann differential equations, four expressions for the complex scale μ can be found. Each of these can also be split into a modulus m and argument g (only one of the four is shown). Any of the four expressions may be used for finding the scale and meridian convergence, and the choice of scale for a single line on the ellipsoid in fact defines the mapping completely because of the Cauchy-Riemann partial differential equations.

The line element differentials do not enter in the expressions for the complex scale, its modulus and argument, because the complex scale is a function of the position (the coordinates) only, but independent of the direction of the line element. This also means that the angle between two geodesics is mapped exactly, because both line elements at the apex of the two geodesics are rotated by the same angle (viz. g the local meridian convergence). The corrections for the scale and direction of a geodesic used for replacing observations of the length and/or direction of a geodesic by a straight line are functions of the local complex scale along the entire geodesic. The corrections are mapping the observations into a straight line in the mapping plane. The mapping of the geodesic itself is a curved line, which is of no special interest for the computations in the mapping space (the plane).

(2.4) Scale and Meridian Convergence

$$\mu = \frac{\left(\frac{\partial y}{\partial \psi} + i\frac{\partial x}{\partial \psi}\right)}{r(\phi)} = \frac{\left(\frac{\partial x}{\partial \lambda} - i\frac{\partial y}{\partial \lambda}\right)}{r(\phi)} = \frac{\left(\frac{\partial x}{\partial \lambda} + i\frac{\partial x}{\partial \psi}\right)}{r(\phi)} = \frac{\left(\frac{\partial y}{\partial \psi} - i\frac{\partial y}{\partial \lambda}\right)}{r(\phi)}$$

$$m = \frac{\mathbf{hypot}(\partial y/\partial \psi,\ \partial x/\partial \psi)}{r(\phi)} \qquad g = \mathbf{atan2}(\partial x/\partial \psi,\ \partial y/\partial \psi)$$

The function names "hypot" and "atan2" are used in the programming language C. Hypot gives the square root of the square sum of the arguments, and atan2 gives the angle (in proper quadrants) for arguments proportionate to sine and cosine of the angle.

2.5 The Exponential Latitude Function

The so-called complex latitude is introduced in the next section in order to generalize the well-known formula for the meridian arc length as a function of the latitude into a formula for the transversal coordinates as a function of the complex latitude. The formulae in (2.5) are simply valid for any complex number and shows just how to do when sine and cosine have complex arguments.

(2.5) Exponential Latitude Function

$$
\begin{aligned}
z &= \exp(i\,\phi_c) = \cos\phi_c + i\sin\phi_c \\
z^{\kappa} &= \exp(i\,\kappa\,\phi_c) = \cos\kappa\phi_c + i\sin\kappa\phi_c \\
\cos\kappa\phi_c &= \frac{z^{\kappa} + z^{-\kappa}}{2} = \cos\kappa\phi_r\,\cosh\kappa\phi_i - i\sin\kappa\phi_r\,\sinh\kappa\phi_i \\
\sin\kappa\phi_c &= \frac{z^{\kappa} - z^{-\kappa}}{2i} = \sin\kappa\phi_r\,\cosh\kappa\phi_i + i\cos\kappa\phi_r\,\sinh\kappa\phi_i
\end{aligned}
$$

The complex latitude will appear in differential coefficients used for the definition of the transversal mapping, and thus may be used in the expressions for the curvature radii of the ellipsoid, $V(\phi)$ and $W(\phi)$ (see (III.1)). The function $F(z)$, apparently introduced by K&W, is frequently a much better choice than V and W. $F(z)$ uses the third flattening n defined in Appendix III.

(2.6) Definition of $F(z)$

$$
\begin{aligned}
V(\phi_c) &= \sqrt{1 + e'^2\cos^2\phi_c} = \sqrt{1 + \frac{4n}{(1-n)^2}\left(\frac{z + z^{-1}}{2}\right)^2} \\
&= \frac{1}{1-n}\sqrt{1 + n\,(z^2 + z^{-2}) + n^2} = \frac{\sqrt{F(z)}}{1-n} \\
&= \frac{1}{1-n}\sqrt{1 + 2n\cos(2\phi_c) + n^2} \\
F(z) &= 1 + n\,(z^2 + z^{-2}) + n^2 = (1 + n z^2)\,(1 + n z^{-2})
\end{aligned}
$$

2.6 Fundamental Mapping Coordinates

The isometric latitude used in (2.2) can be derived from the geodetic latitude by simple integration. The corresponding isometric latitude for a sphere is found by putting the eccentricity $e = 0$.

(2.7) Geodetic or Spherical Latitude ⇒ Isometric Latitude

ELLIPSOID

$$d\psi = \frac{M(\phi)}{N(\phi)\cos\phi}\,d\phi \qquad p = \pi/2 - \phi$$

$$d\psi = -\frac{1-e^2}{(1-e^2\sin^2 p)\,\sin p}\,dp = -\left(\frac{dp}{\sin p} + \frac{e}{2}\left(\frac{d(1+e\cos p)}{1+\cos p} - \frac{d(1-e\cos p)}{1-\cos p}\right)\right)$$

$$\psi = -\ln\left(\tan(p/2)\left(\frac{1+e\cos p}{1-e\cos p}\right)^{e/2}\right) \qquad ;\ p = \pi/2 \to \psi = 0$$

$$\psi_c = \ln\left(\tan(\pi/4+\phi/2)\left(\frac{1-e\sin\phi}{1+e\sin\phi}\right)^{e/2}\right) + i\lambda \qquad ;\ \textit{fundamental mapping coord.}$$

SPHERE

$$\Psi_c = -\ln\tan(P/2) + i\Lambda = \ln\tan(\pi/4 + \Phi/2) + i\Lambda \qquad ;\ P = \pi/2 \to \Psi = 0$$

The isometric latitude as a real number and the longitude as an imaginary number form the complex isometric coordinates, from which the other mappings are derived.

2.7 The Complex Latitude

The differential coefficients from (2.2) and (2.7) for the geodetic latitude and the isometric latitude may be re-used with complex numbers, but now for finding a relation between the complex isometric coordinates and the complex geodetic latitude mentioned in the preceding Sec. 2.5.

(2.8) Differentials of isometric coordinates and complex latitude

$$d\psi_c = \frac{M(\phi_c)}{N(\phi_c)\cos\phi_c}\,d\phi_c = \frac{d\phi_c}{V^2(\phi_c)\,\cos\phi_c}, \qquad z = \exp(i\,\phi_c), \qquad d\phi_c = -i\,dz/z$$

$$d\psi_c = -2i\,\frac{(1-n)^2}{F(z)\,(z+z^{-1})}\,\frac{dz}{z} = -2i\left(\frac{dz}{z^2-1} - n\,\frac{z+z^{-1}}{F(z)\,z}\,dz\right)$$

$$d\psi_c = \frac{dz}{z+i} - \frac{dz}{z-i} + 2in\,(1+z^{-2})\,F^{-1}(z)\,dz$$

The integration is easy due to the convenient series expansion for $F^{-1}(z)$ given in (III.2).

(2.9) Complex Latitude ⇒ Complex Isometric coordinates

$$\psi_c = \int \left(d\ln\left(\frac{z+i}{z-i}\right) + 2in \left(n_0 + n_2 + \sum_{\kappa=1}^{\infty} (n_{2\kappa} + n_{2\kappa+2})\, z^{2\kappa} + \sum_{\kappa=1}^{\infty} (n_{2\kappa} + n_{2\kappa-2})\, z^{-2\kappa} \right) \right) dz$$

$$\psi_c = \ln\left(\frac{z+i}{z-i}\right) + \frac{1}{2i} \sum_{\kappa=1}^{\infty} \frac{4\,(-n)^{\kappa}}{(1+n)\,(2\kappa-1)} \left(z^{2\kappa-1} - z^{-2(\kappa-1)}\right) + \ln t$$

Integration constant: $\ln t$

$$\phi_c = 0 \Rightarrow z = 1 \Rightarrow \psi_c = 0 \Rightarrow \ln\left(\frac{1+i}{1-i}\right) + \ln t = 0 \Rightarrow t = -i$$

$$\psi_c = \ln\left(-i\,\frac{z+i}{z-i}\right) + \frac{1}{2i} \sum_{\kappa=1}^{\infty} \frac{4\,(-n)^{\kappa}}{(1+n)\,(2\kappa-1)} \left(z^{2(\kappa-1)} - z^{-2(\kappa-1)}\right)$$

The complex latitude is different from the complex number composed of the ordinary latitude as the real component and the longitude as the imaginary one, unless the longitude equals nought as can be seen by comparing the formulae for the isometric latitude and the complex isometric coordinates. Therefore the formulae for the complex coordinates based upon z, the exponential latitude, can also be used for finding the isometric latitude as a function of the latitude.

The inversion of the formulae for the isometric coordinates and the complex latitude will be shown in Chapter 4.

(2.10) Details of the Isometric Latitude Formulae

$$\ln\left(-i\,\frac{z+i}{z-i}\right) = \ln\left(-i\,\frac{\exp(i\,\phi_c) - \exp(-i\,\pi/2)}{\exp(i\,\phi_c) + \exp(-i\,\pi/2)}\right)$$

$$= \ln\left(\frac{1}{i}\,\frac{\exp(i(\phi_c/2 + \pi/4)) - \exp(-i(\phi_c/2 + \pi/4))}{\exp(i(\phi_c/2 + \pi/4)) + \exp(-i(\phi_c/2 + \pi/4))}\right) = \ln\tan(\phi_c/2 + \pi/4)$$

$$\frac{1}{2i}\left(z^{+(2\kappa-1)} - z^{-(2\kappa-1)}\right) = \sin((2\kappa-1)\,\phi_c)$$

ELLIPSOID: ψ_c = *Complex isometric coordinates,* ψ = *Isometric latitude*

$$\psi_c = \ln\tan(\pi/4 + \phi_c/2) + \sum_{\kappa=1}^{\infty} \frac{4\,(-n)^{\kappa}}{(1+n)\,(2\kappa-1)} \sin((2\kappa-1)\,\phi_c)$$

$$\psi = \ln\tan(\pi/4 + \phi/2) + \sum_{\kappa=1}^{\infty} \frac{4\,(-n)^{\kappa}}{(1+n)\,(2\kappa-1)} \sin((2\kappa-1)\,\phi)$$

SPHERE: Ψ_c = *Complex isometric coordinates,* Ψ = *Isometric latitude*

$$\Psi_c = \ln\left(-i\,\frac{Z+i}{Z-i}\right) = \ln\tan(\pi/4 + \Phi_c/2)$$

$$\Psi = \ln\tan(\pi/4 + \Phi/2) = -\ln\tan(P/2)$$

2.8 The Normalized Transversal Coordinates

The formula for the length of a meridian arc is derived in Appendix III so that it is also valid for complex values of the latitude. Therefore the central meridian is mapped precisely in the desired scale of the central meridian, e.g. 1 for Gauss-Krüger and 0.9996 for UTM. The value of the Northing coordinate thus becomes precisely the value of the meridian quadrant times the central scale. However, it is more convenient for the development of the formulae to use normalized coordinates at this stage, so that the normalized northing (y) at the North Pole becomes $\pi/2$. The metric coordinates (N, E) are then found from the normalized (y, z) by multiplication with the meridian arc unit Q and the central scale m_0. This will be dealt with in Chapter 4.

The complex latitude can be computed from the complex isometric coordinates which is the direct way used for defining the complex latitude, but in Chapter 3 a "back door" to the complex coordinates from the geodetic latitude and longitude without need for isometric coordinates will be found.

The complex latitude can of course be eliminated already in the differential equations in (2.8) and (2.10), so that a direct transformation from isometric coordinates to transversal coordinates. This method is frequently used, but has the drawbacks that the poles of the ellipsoid are singular points for the isometric coordinates and the coefficients of the power series expansions are more complicated that those of the trigonometric series expansions shown in Chapter 4.

(2.11) Normalized Transversal Coordinates

ELLIPSOID:

$$du = \frac{M(\phi_c)}{Q}\, d\,\phi_c \qquad \text{; differential normalized transversal coordinates}$$

$$u = y + ix = \frac{1}{Q}\int_0^{\phi_c} M(\phi_c)\, d\,\phi_c \qquad \text{; Normalized transversal coordinates}$$

$$u = \phi_c + \sum_{\kappa=1}^{\infty} p_{2\kappa} \sin(2\kappa\phi_c) \qquad \text{; trig. series for } u\text{, see (III.4.A)}$$

SPHERE (Radius = 1) :

$$dU = d\Phi_c \qquad \text{; differential transv. coord}$$

$$U = \Phi_c \qquad \text{; transv. coord = compl. sph. coord}$$

The normalized transversal coordinates for the ellipsoid are almost equal the complex latitude because the curvature radius of the meridian is varying. The corresponding quantities for the Gaussian sphere are precisely equal because the radius is constant (= unity).

The inverse transformation uses the inverse of the meridian arc formula shown in (III.5.A).

(2.12) Complex Latitude from Normalized Transversal Coordinates

$$\phi_c = u + \sum_{\kappa=1}^{4} q_{2\kappa} \sin(2\kappa u) \qquad ;\ \textit{see (III.5.A)}$$

2.9 The Normalized Co-axial Mappings

The co-axial mappings treated in the present work are very simple functions of the complex isometric coordinates, as indicated in (2.13). It is even so, that the stereographic mapping can be regarded as a special case of the Lambert (conical) mapping, just by putting k equal +1 or -1. Any factor k different from 0, +1, and -1 will in fact give a conical conformal mapping, but probably only the values in (2.13) will be of interest.

(2.13) Normalized Co-axial Mappings

$u = y + ix$. ; ***mapping coordinates***

$u = \psi_c$. ; ***Mercator***

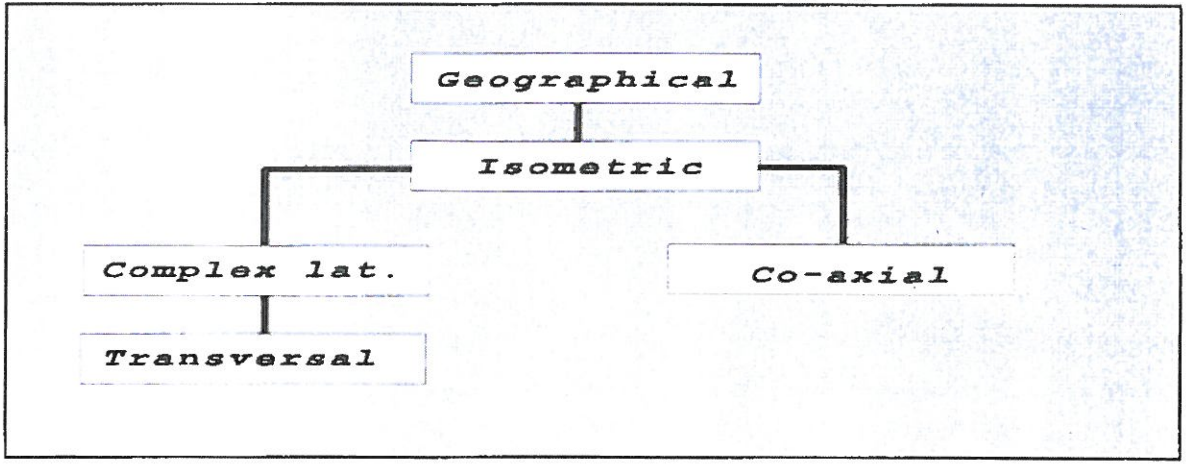
$u = \exp(-k\,\psi_c)$ **abs(k)≤1 ∧ k≠0** ; ***Lambert & Polar Stereographic***

The factors giving metric coordinates from the normalized coordinates will be found in Chapter 5.

2.10 A Diagram of Normalized Conformal Mappings

The diagram in Fig. 2.2 shows the relations end sequences of the conformal mappings treated in the present work. The formal entrance to the mappings goes via the isometric coordinates.

Fig. 2.2 Normalized Conformal Mappings



3. The Gaussian Sphere

The Gaussian sphere is virtually a Soldner sphere with the equator as the latitude of contact and a radius of unity (dimensionless). It is not meant to be an approximation sphere (the Gaussian osculation sphere would do such an approximation job much better), but it is a parameter sphere leading to very convenient formulae. It is of course permitted to regard it as an intermediate mapping sphere, but the mapping scale expressions will be complicated and uninteresting, because they will have to first introduce a very awkward mapping scale to the sphere and then restore this mapping scale when going to the intended mapping surface.

The Gauss-Schreiber mapping is using the Gaussian osculation sphere as a very good intermediate mapping surface, from which the transversal mapping is very simple, but the scale on the central meridian is not constant, because the scale of the mapping on the sphere varies with the third power of the latitude difference from the latitude of minimum scale.

3.1 Mapping Ellipsoid ⇐⇒ Sphere

The ellipsoid is mapped on the sphere with the simple and virtually unique mapping equations as shown below. Strictly speaking we should account for the Riemann leaves arising from the periodicity of the complex functions. The singularity at the poles is not serious, because the corresponding latitudes could be found by the atan2 function. We call the spherical latitudes defined in this way the Gaussian latitudes.

(3.1) Ellipsoid ⇐⇒ Sphere

Mapping equations

$$\Psi = \psi \qquad\qquad \Lambda = \lambda$$

$$\mu = \frac{\cos(\Phi)\,(d\,\Psi + i\,d\,\Lambda)}{N(\phi)\cos(\phi)\,(d\,\psi + i\,d\,\lambda)} = \frac{\cos(\Phi)}{N(\phi)\cos(\phi)} = m \qquad ;\, g = 0$$

Cauchy-Riemann

$$\frac{\partial\Psi}{\partial\psi} = \frac{\partial\Lambda}{\partial\lambda} = 1 \qquad\qquad \frac{\partial\Lambda}{\partial\psi} = -\frac{\partial\Psi}{\partial\lambda} = 0$$

$$\exp(-\psi) = \tan(p/2)\left(\frac{1 + e\cos p}{1 - e\cos p}\right)^{e/2} = \tan(P/2) \qquad (A)$$

3.2 Geodetic Coordinates ⇐⇒ Gaussian Coordinates

The Gaussian latitude plays a central rôle in the mappings treated here. In principle the relations in (3.1.A) to the (same) isometric latitude for ellipsoid and sphere provides a unique definition, but it is possible to give more useful formulae giving conversion both ways without iteration and with a precision better than 1 micro arcsecond.

If the complex isometric coordinates on the Gaussian sphere and the ellipsoid are put-equal, then we have defined a mapping between the two surfaces. The derivations in (3.2) produce the tools for creating the formulae giving the complex Gaussian coordinates from the complex latitude and vice versa.

(3.2) Ellipsoid ⟺ Gaussian Sphere Using Exponential Latitude

$$\Psi_c = \psi_c \quad ; \textit{Equal complex isometric coordinates, see } (2.10)$$

$$\ln\left(-i\frac{Z+i}{Z-i}\right) = \ln\left(-i\frac{z+i}{z-i}\right) + \frac{1}{2i}\sum_{\kappa=1}^{4}\frac{4(-n)^{\kappa}}{(1+n)(2\kappa-1)}\left(z^{+(2\kappa-1)} - z^{(-(2\kappa-1))}\right)$$

$$q = \sum_{\kappa=1}^{4}\frac{(-n)^{\kappa}}{(1+n)(2\kappa-1)}\left(z^{+(2\kappa-1)} - z^{-(2\kappa-1)}\right)$$

$$\ln\left(-i\frac{Z+i}{Z-i}\right) = \ln\left(-i\frac{z+i}{z-i}\right) - i\,2q \quad ; \quad \frac{Z+i}{Z-i} = \frac{z+i}{z-i}\exp(-i\,2q)$$

$$\begin{aligned} Z &= i\,\frac{z(1+\exp(i\,2q)) + i(1-\exp(i\,2q))}{z(1-\exp(i\,2q)) + i(1+\exp(i\,2q))} \\ &= i\,\frac{z(\exp(iq)+\exp(-iq)) - (\exp(iq)-\exp(-iq))}{-z(\exp(iq)-\exp(-iq)) + i(\exp(iq)+\exp(-iq))} \\ &= z\,\frac{1+z^{-1}\tan q}{1-z\tan q} \end{aligned}$$

$$\tan q = q + \tfrac{1}{3}q^3 + o(q^5)$$

Taking the logarithm:

$$\Phi_c = \phi_c + \ln(1+z^{-1}\tan q) - \ln(1-z\tan q) \qquad (A)$$

The components of the relation of Z and z are found from the expression for the isometric latitude. The additive non-spheric terms are included by the tan q.

(3.3) Expansion of tan q

$$\tan q = \sum_{-\infty}^{+\infty} t_{2\kappa-1}\, z^{2\kappa-1} \quad ; \quad t_{-(2\kappa-1)} = -t_{2\kappa-1}$$

$$\begin{aligned} t_1 &= -n + n^2 - \tfrac{5}{3}n^4 & t_3 &= +\tfrac{1}{3}n^2 - \tfrac{2}{3}n^3 + \tfrac{2}{3}n^4 \\ t_5 &= -\tfrac{1}{5}n^3 + \tfrac{8}{15}n^4 & t_7 &= +\tfrac{1}{7}n^4 \end{aligned}$$

The next step is the series expansion of the logarithms of the nominator and the denominator. As q is of the order of n, the two terms shown give a precision of 4th order. The substitution of tan q in the series expansions of the logarithms is comprehensive but trivial. The final formula for the Gaussian latitude as a function of the geodetic latitude is very simple and valid for the complex latitudes. However, we shall mostly use it for pure real values. The truncation after the order 4 produces an error of the order of $(1/600)^5$, ie. much less than one microsecond of arc. Note that the coefficients are real, which of course simplifies the summation, and that the coefficients depend only on the 3. flattening, n, but not on the latitude.

(3.4) Complex Geodetic Latitude → Complex Gaussian Coordinates

$$\ln(1 + \frac{1}{z}\tan(q)) = \sum_{\kappa=1}^{\infty} \frac{(-1)^{\kappa-1} z^{-\kappa}}{\kappa} \tan^{\kappa}(q)$$

$$\ln(1 - z\tan(q)) = -\sum_{\kappa=1}^{\infty} \frac{z^{\kappa}}{\kappa} \tan^{\kappa}(q)$$

$$\ln Z = \ln z + \sum_{\kappa=1}^{\infty} e_{2\kappa} \left(\frac{z^{2\kappa} - z^{-2\kappa}}{2}\right) \qquad ; (\ln z = \ln \exp(i\phi_c) = i\phi_c)$$

$$U = \Phi_c = \phi_c + \sum_{\kappa=1}^{4} e_{2\kappa} \sin 2\kappa\phi_c \qquad (A)$$

$$Array\ e[4] = \{e_2, e_4, e_6, e_8\} \qquad (B)$$

$$e_2 = -2n + \frac{2}{3}n^2 + \frac{4}{3}n^3 - \frac{82}{45}n^4 \qquad e_4 = +\frac{5}{3}n^2 - \frac{16}{15}n^3 - \frac{13}{9}n^4$$

$$e_6 = -\frac{26}{15}n^3 + \frac{34}{21}n^4 \qquad e_8 = +\frac{1237}{630}n^4$$

The notation $e[4]$ in (3.4.B) used for an array of the constants e_2, e_4, e_6, and e_8 marks $e[4]$ as different from e representing the eccentricity of the ellipsoid.

The series (3.4.A) can be inverted to give the complex geodetic latitude as a function of the complex Gaussian coordinates.

(3.5) Complex Gaussian Coordinates → Complex Geodetic Latitude

$$U = \Phi_c$$

$$\phi_c = U + \sum_{\kappa=1}^{4} G_{2\kappa} \sin 2\kappa U \qquad (A)$$

$$Array\ G[4] = \{G_2, G_4, G_6, G_8\} \qquad (B)$$

$$G_2 = +2n - \frac{2}{3}n^2 - 2n^3 + \frac{116}{45}n^4 \qquad G_4 = +\frac{7}{3}n^2 - \frac{8}{5}n^3 - \frac{227}{45}n^4$$

$$G_6 = +\frac{56}{15}n^3 - \frac{136}{35}n^4 \qquad G_8 = +\frac{4279}{630}n^4$$

The series in (3.4.A) and (3.5.A) are used for deriving the relations between the normalized transversal coordinates (u = y+ix) and the Gaussian complex coordinates (U = Y+iX), but they can also be used for transforming between geodetic latitude and Gaussian latitude. If the meridian of the actual point is considered to be a main meridian, then the imaginary part of the complex latitude is zero. The real part of the complex latitude for the formulae in (3.4) and (3.5) is then the same as the geodetic latitude or Gaussian latitude, and the formulae in (3.1) connecting the sphere is defined by putting the complex isometric coordinates equal for ellipsoid and Gaussian sphere. The actual summation is done as Clenshaw summation (see Appendix I).

(3.6) Function CS(e[4], 2ϕ): Geodetic Coordinates ⇒ Gaussian Coordinates

$$\begin{aligned}
\Psi_c &\equiv \psi_c && ;\ \textit{identic complex isometric coordinates}\\
\Phi &= \phi + \sum_{\kappa=1}^{4} e_{2\kappa} \sin 2\kappa\phi && ;\ \textit{Coefficients in } (3.4.B)\\
&= \phi + CS(e[4],\ 2\phi) && ;\ \textit{Clenshaw sine summation, see App. I}\\
\Lambda &= \lambda
\end{aligned}$$

(3.7) Function CS(G[4], 2Φ): Gaussian Coordinates ⇒ Geodetic Coordinates

$$\begin{aligned}
\psi_c &= \psi + i\lambda \equiv \Psi_c = \Psi + i\Lambda && ;\ \textit{identic complex isometric coordinates}\\
\phi &= \Phi + \sum_{\kappa=1}^{4} G_{2\kappa} \sin 2\kappa\Phi && ;\ \textit{Coefficients in } (3.5.B)\\
&= \Phi + CS(G[4],\ 2\Phi) && ;\ \textit{Clenshaw sine summation, see App. I}\\
\lambda &= \Lambda
\end{aligned}$$

The transformation shown in (3.6) and its inverse in (3.7) using only real latitudes are universal tools used for transformations between geodetic coordinates and Gaussian coordinates treated in the present work. The transformations in (3.4) and (3.5) are used in substitutions of the complex geodetic latitude in Clenshaw summations with complex arguments, see (4.3) and (4.4).

3.3 Gaussian Coordinates ⇒ Complex Gaussian Coordinates

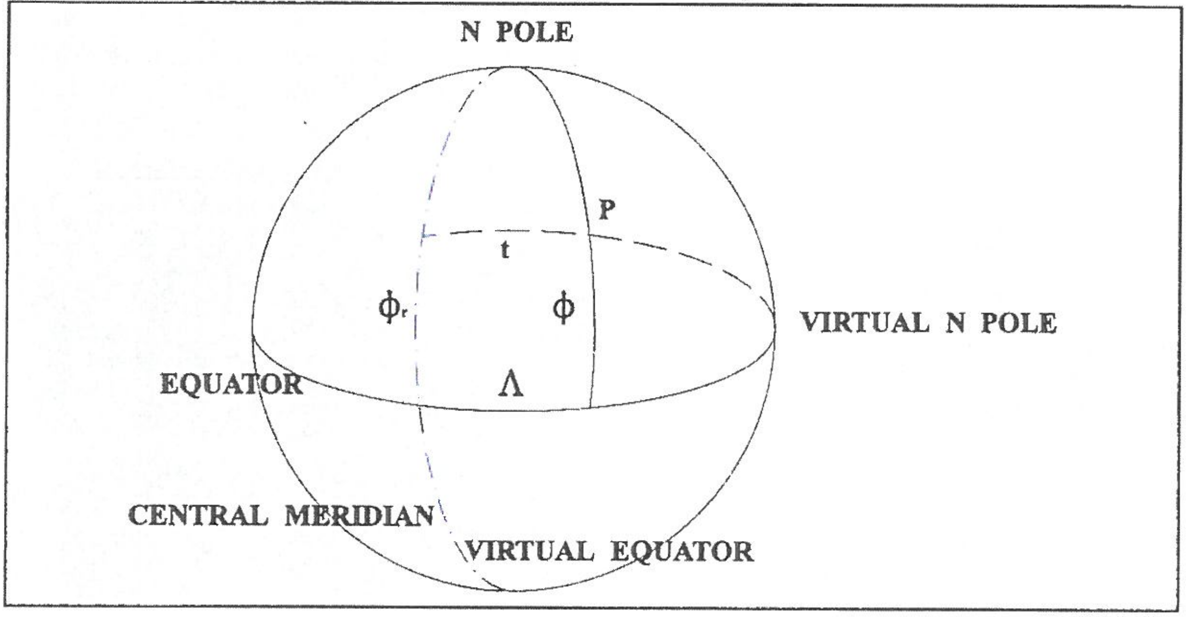
The complex Gaussian coordinates gives a transversal mapping, where the central meridian is mapped with unity scale, i.e. it behaves as if the central meridian is the "equator" and the normal to the central meridian plane passing through the centre of the sphere is the "rotation axis". A point P with Gaussian coordinates Φ, Λ should be mapped to the complex Gaussian coordinates $\Phi_c = \Phi_r + i\ \Phi_i$.

The Figure 3.1 shows the latitudes, the longitude (difference) and the auxiliary parameter t, which in fact is the virtual latitude, while Φ_r is the virtual longitude in a Mercator mapping, where the central meridian acts as the virtual equator and the vitual poles are found in the real equator 90 degrees from the central meridian.

The formulae are based upon spherical geometry for finding Φ_r and t. The latter is then mapped as a virtual y in a normalized Mercator mapping. The formulae use "atan2" and "hypot" and have all sign rules built in. The formulae are robust so that they will work even at the North Pole and the South Pole. The value of t should not exceed 40-50 degrees (corresponding to 4-5000 km off the central meridian on the earth).

The complex transversal coordinates $U = Y + iX$ and the complex Gaussian coordinates $\Phi_c = \Phi_r + i\Phi_i$ are identical as shown in (2.11).

Fig. 3.1 The Gaussian Coordinates and the Complex Gaussian Coordinates



(3.8) Function T_d: Gaussian Coordinates ⇒ Complex Gaussian Coordinates

```
→   INPUT: $\Phi, \Lambda$                ; Gaussian latitude & longitude
    $\tan \Phi_r = \dfrac{\tan \Phi}{\cos \Lambda}$
←   $Y = \Phi_r = \mathrm{atan2}(\sin \Phi, \cos \Phi * \cos \Lambda)$
    $t = \mathrm{atan2}(\cos \Phi * \sin \Lambda, \mathrm{hypot}(\sin \Phi, \cos \Phi * \cos \Lambda))$
←   $X = \Phi_i = \ln \tan(\pi/4 + t/2)$
←   $U = Y + iX = \Phi_r + i\,\Phi_i$

    $U = T_d(\Phi, \Lambda)$              ; FUNCTION NAME
```

The solution of the inverse problem of finding the Gaussian coordinates from the complex Gaussian coordinates $\Phi_c = U$ is likewise based upon spherical geometry.

(3.9) Function T_i: Complex Gaussian Coordinates ⇒ Gaussian Coordinates

```
→   $U = Y + iX = \Phi_r + i\,\Phi_i$            ; Compl. Gauss. coord.

    $t = 2\,\mathrm{atan}(\exp(X)) - \pi/2$
←   $\Lambda = \mathrm{atan2}(\sin t, \cos t \cos Y)$
←   $\Phi = \mathrm{atan2}(\sin Y \cos t, \mathrm{hypot}(\sin t, \cos t \cos Y))$

    $(\Phi, \Lambda) = T_i(Y, X)$                ; FUNCTION NAME
```

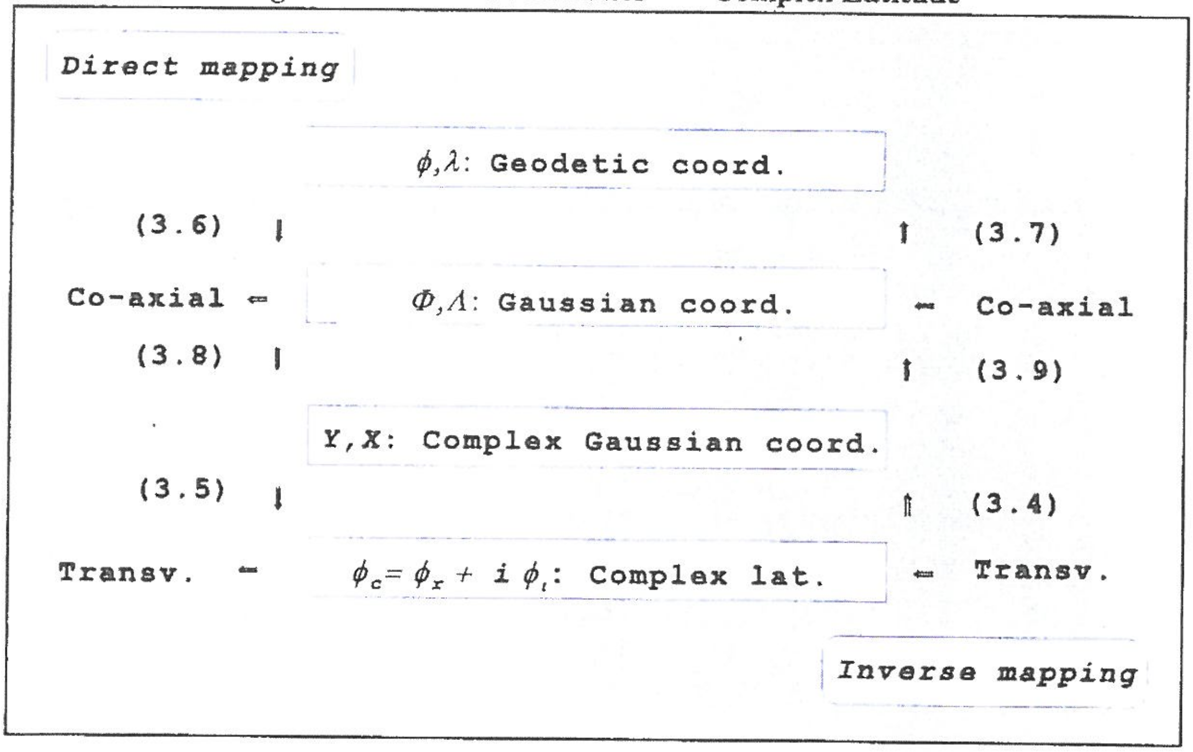
3.4 The Back Door Keys

The formulae in this chapter are the "keys to the back door" mentioned in Sec. 2.8. It is possible to compute the complex latitude from the geodetic coordinates without using the isometric coordinates and also to do the inverse computation. This is used in Chapter 4, where the complex latitude is eliminated.

One of the co-axial mappings (Mercator) has the isometric coordinates as its normalized coordinates, but the two other mappings can be computed without the isometric coordinates, but with the Gaussian latitude as a tool as will be shown in Chapter 5, and it is even so that the Gaussian latitude simplifies the computations to/from Mercator coordinates.

Fig. 3.2 should illustrate where the formulae are used in the sequence of the computations, and show the back doors labelled "Transv." and "Co-axial".

Fig. 3.2 Geodetic Coordinates ⇐⇒ Complex Latitude


Direct mapping
ϕ,λ: Geodetic coord.
(3.6)
(3.7)
Co-axial
Φ,Λ: Gaussian coord.
Co-axial
(3.8)
(3.9)
Y,X: Complex Gaussian coord.
(3.5)
(3.4)
Transv.
$\phi_c = \phi_r + i\,\phi_i$: Complex lat.
Transv.
Inverse mapping

4. Transversal Mapping

The basic idea behind this mapping is the generalisation of the meridian arc to the plane of complex numbers.

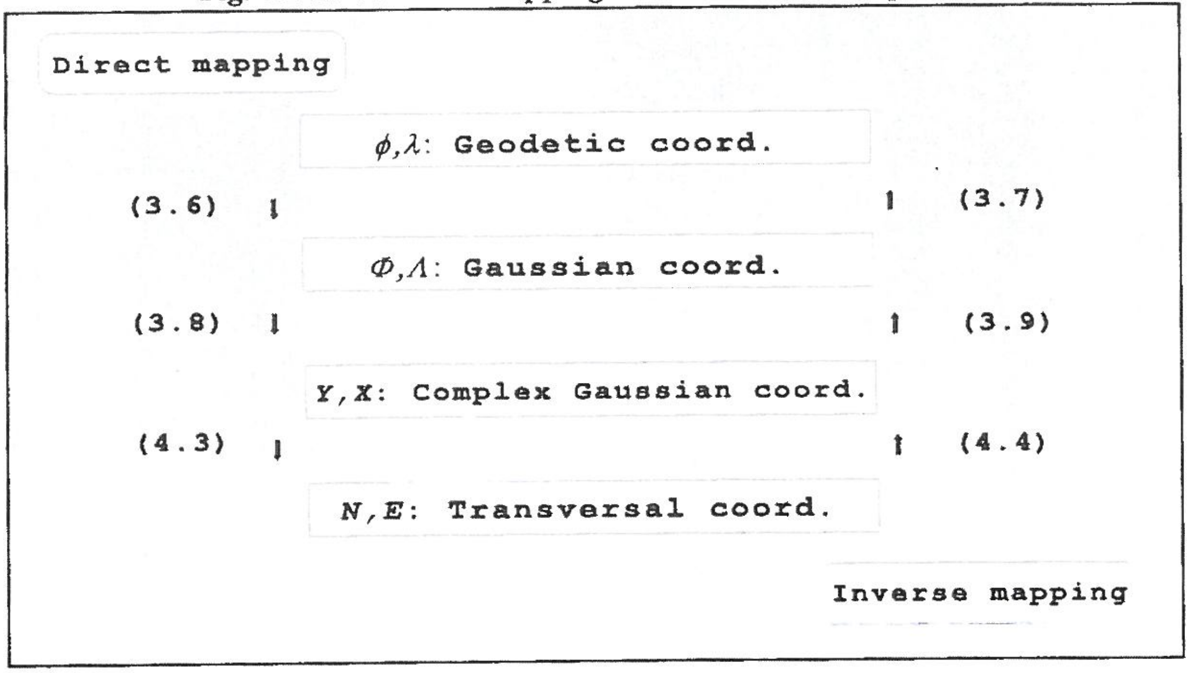
(4.1) Definitions for the Transversal Mapping

Point P with geodetic coordinates $(\phi, \lambda + \lambda_c)$
Genuine origo: $(N, E) = (0, 0)$
at $\phi = 0$, λ_c = *longitude of central meridian*
Isometric complex coord: $\psi_c = \psi + i\lambda = \psi_c(\phi, \lambda)$

ELLIPSOID	*SPHERE*
Cmpl. latitude: $\phi_c = \phi_r + i\,\phi_i$	$\Phi_c = \Phi_r + i\,\Phi_c$
$\dfrac{d\phi_c}{d\psi_c} = \dfrac{\cos(\phi_c)}{(1-n)^2}(1 + n^2 + 2\cos(2\phi_c))$	
Norm. transv. crd: $u = y + i\,x$	$U = Y + i\,X$
$\dfrac{du}{d\phi_c} = \dfrac{M(\phi_c)}{Q}$	$\dfrac{dU}{d\Phi_c} = 1$
$N + iE = Q_m u = Q_m(y + i\,x)$	$U \equiv \Phi_c$

Q, Q_m: *See* (4.2)

Fig. 4.1 Transversal Mapping: Transformation Sequenses



The direct and inverse transversal mapping follows the sequence of subprocesses shown in Fig. 4.1, where the complex latitude has been eliminated.

The formulae given in Chapters 2 and 3 uses normalized, dimensionless mapping coordinates, which should be converted to/from metric, scaled coordinates by means of the scale meridian arc unit.

(4.2) Scaling and Normalizing

$$
\begin{array}{lll}
Q & = \dfrac{a}{1+n}\left(1+\frac{1}{4}n^2+\frac{1}{64}n^4\right) & ;\ \textit{Meridian arc unit} \\
m_0 & = 0.9996 & ;\ \textit{UTM central scale} \\
m_0 & = 1.0 & ;\ \textit{Gauss-Krüger central scale} \\
Q_m & = m_0 Q & ;\ \textit{Scaled meridian arc unit} \\
N + iE & = (y+ix)\,Q_m & ;\ \textit{normalized coord} \rightarrow \textit{metric, scaled coord} \\
y + ix & = (N+iE)/Q_m & ;\ \textit{metric, scaled coord} \Rightarrow \textit{normalized coord}
\end{array}
$$

Two examples of the central scale (0.9996 and 1.0) are shown, but it is easy to use any other scale. The scaling may also be regarded as a mapping with unit scale for an ellipsoid with the equatorial radius scaled by the central scale m_0.

4.1 Geodetic Coordinates → Transversal Coordinates

The complex latitude can be computed from the geodetic latitude and longitude without using the isometric coordinates as shown in Chapter 3, and in Chapter 2 the normalized transversal coordinates could be found by using the meridian arc formula with the complex latitude. The metric, scaled transversal coordinates are then finally found by means of the scaled meridian arc unit.

The series for the complex latitude from complex Gaussian coordinates (3.5) can be substituted in the series for the normalized transversal coordinates from the complex latitude (2.11).

(4.3) Gaussian Complex Coordinates ⇒ Transversal Coordinates

$$
\begin{array}{lll}
\rightarrow\ U & & ;\ \textit{input: Gaussian compl. coord.} \equiv \textit{Gaussian transv. coord} \\
\phi_c & = U + \sum_{\nu=1}^{4} G_{2\nu}\sin 2\nu U & ;\ (3.5.A) : \rightarrow \textit{complex lat.} \\
u & = \phi_c + \sum_{\kappa=1}^{4} p_{2\kappa}\sin 2\kappa\phi_c & ;\ (2.11) : \rightarrow \textit{nrmlz. transv. crd.} \\
 & = \left(U + \sum_{\nu=1}^{4} G_{2\nu}\sin 2\nu U\right) + \sum_{\kappa=1}^{4} p_{2\kappa}\sin 2\kappa\left(U + \sum_{\nu=1}^{4} G_{2\nu}\sin 2\nu U\right) & \\
u & = U + \sum_{\kappa=1}^{4} U_{2\kappa}\sin 2\kappa U & ;\qquad (A)
\end{array}
$$

$$
\begin{array}{ll}
U_2 = +\frac{1}{2}n - \frac{2}{3}n^2 + \frac{5}{16}n^3 + \frac{41}{180}n^4 & U_4 = +\frac{13}{48}n^2 - \frac{3}{5}n^3 + \frac{557}{1440}n^4 \\
U_6 = \qquad\qquad + \frac{61}{240}n^3 - \frac{103}{140}n^4 & U_8 = \qquad\qquad + \frac{49561}{161280}n^4
\end{array}
$$

$$
\leftarrow\ N + iE = uQ \qquad ;\ \textit{Nrmlzed. transv. coord} \rightarrow \textit{metric scaled transv.}
$$

4.2 Transversal Coordinates ⇒ Geodetic Coordinates

All tools for the inverse problem of finding the geodetic coordinates from the transversal coordinates are also found in Chapters 2 and 3. The only remaining formula is the series for the complex Gaussian coordinates from the transversal coordinates. This series is found by substituting the series for the complex latitude from the normalized transversal coordinates into the series for the complex Gaussian complex coordinates from the complex latitude.

(4.4) Transversal Coordinates ⇒ Complex Gaussian Coordinates

$$\rightarrow \quad N,\ E \qquad \textit{; Input: Transversal coordinates}$$

$$u = (N + iE)/Q_m \qquad \textit{; normalized transversal coord}$$

$$\phi_c = u + \sum_{\nu=1}^{4} q_{2\nu} \sin(2\nu u) \qquad ;\ (2.12)\ :\ \Rightarrow \textit{complex latitude}$$

$$U = \Phi_c = \phi_c + \sum_{\kappa=1}^{4} e_{2\kappa} \sin 2\kappa\phi_c \qquad ;\ (3.4.A)\ :\ \Rightarrow \textit{complex Gaussian coord.}$$

$$= \left(u + \sum_{\nu=1}^{4} q_{2\nu} \sin 2\nu u\right) + \sum_{\kappa=1}^{4} e_{2\kappa} \sin 2\kappa \left(u + \sum_{\nu=1}^{4} q_{2\nu} \sin 2\nu u\right)$$

$$= u + \sum_{\kappa=1}^{4} u_{2\kappa} \sin 2\kappa u \qquad (A)$$

$$u_2 = -\frac{1}{2}n + \frac{2}{3}n^2 - \frac{37}{96}n^3 + \frac{1}{360}n^4 \qquad u_4 = -\frac{1}{48}n^2 - \frac{1}{15}n^3 + \frac{437}{1440}n^4$$

$$u_6 = -\frac{17}{480}n^3 + \frac{37}{840}n^4 \qquad u_8 = -\frac{4397}{161280}n^4$$

$$\leftarrow \quad \Phi_c = \Phi_r + i\,\Phi_i = U = Y + iX \qquad \textit{; Complex Gaussian coordinates}$$

The formulae will have no problems at the poles of the ellipsoid, and it is even so that coordinates on "the other side of the pole" may be computed. The poles of singularity of this mapping are situated on equator 90 degrees of longitude from the central meridian.

The last coefficient of the series expansion is of the order of 10^{-11}, but eventually the hyperbolic functions in the series expansion will be large. This will occur at distances of the order of 4-5000 km from the central meridian. The scale will here be rather large (1.2-1.5) for a central scale near unity.

It is customary to add 500 km to the Easting coordinate, and in the UTM standard, Northings south of the equator must have an addition of 10 000 km. This is somewhat inconvenient, because the coordinates are not unique, so 20 000 km would make more sense. This 10 000 km ambiguity is furthermore inconvenient if one wants to use transversal conformal mappings for polar maps.

(4.5) Parameters for Transversal Mapping

ϕ_0 = *Reference value for the latitude, mostly* 0°
N_0 = *Northing at ϕ_0, mostly* 0 *km or* 10 000 *km*
λ_0 = *Longitude of the central meridian*
E_0 = *Easting at λ_0, mostly* 500 *km*
Q_m = *Meridian arc length unit , scaled* = $m_0 Q$
m_0 = *Scale on central meridian*
$e[4]$ = *Coeff. for geodetic latitude* ⇒ *Gaussian latitude*: (3.4) *and* (3.5)
$G[4]$ = *Coeff. for Gaussian latitude* ⇒ *geodetic latitude*: (3.6) *and* (3.7)
$u[4]$ = *Coeff. for Normalz. transv. crd.* ⇒ *Gaussian compl. crd.*: (4.3)
$U[4]$ = *Coeff. for Gaussian compl. crd.* ⇒ *Normalz. transv. crd.*: (4.4)

(4.6) Geodetic Coordinates ⇒ Transversal Coordinates

```
→  INPUT  $\phi, \lambda$                                            ; UNIT: radians
   $\lambda = \lambda - \lambda_0$                                   ; Subtract central longitude
   $\Phi = \phi + \sum_{\kappa=1}^{4} e_{2\kappa} \sin(2\kappa\phi)$ ; see (3.6)
   $\Lambda = \lambda$
   $U = \Phi_c = T_d(\Phi, \Lambda)$                                 ; see (3.8)
   $u = U + \sum_{\kappa=1}^{4} U_{2\kappa} \sin(2\kappa U)$         ; see (4.3.A)
←  $N + iE = u Q_m + N_0 + i E_0$                                    ; UNIT: metres
```

(4.7) Transversal Coordinates ⇒ Geodetic Coordinates

```
→  INPUT:  $N, E$                                                    ; UNIT: metres
   $N = N - N_0$                                                     ; Subtract origo value
   $E = E - E_0$                                                     ; Subtract origo value
   $u = (N + iE)/Q_m$                                                ; Normalized coord.
   $\Phi_c = U = u + \sum_{\kappa=1}^{4} u_{2\kappa} \sin(2\kappa u)$ ; see (4.4.A)
   $(\Phi, \Lambda) = T_i(\Phi_c)$                                   ; see (3.9)
←  $\phi = \Phi + \sum_{\kappa=1}^{4} G_{2\kappa} \sin(2\kappa\Phi)$ ; see (3.7)
←  $\lambda = \Lambda + \lambda_0$                                   ; UNIT for $\phi$ and $\lambda$: radians
```

4.3 Mapping of Observations

The corrections for mapping the observations of directions and distances on the ellipsoid into the surface of the coordinate system give the possibility of computing in an Euclidean geometry. The formulae in (IV.2) are reasonably precise, and the extension of the formulae to a higher precision requires so many more terms, that it might be worthwhile to compute directly on the ellipsoid, and later transform the resulting coordinates to the actual mapping.

(4.8) Variables for Mapping of Observations

Indices:

stn = *station*
obj = *object*

$$N_m = \tfrac{1}{2}(N_{stn} + N_{obj})$$
$$E_m = \tfrac{1}{2}(E_{stn} + E_{obj})$$
$$\Delta N = N_{obj} - N_{stn}$$
$$\Delta E = E_{obj} - E_{stn}$$

ϕ = *Geodetic latitude*

See (III.1) and (III.4) in Appendix III

$G(\phi)$ = *Scaled central meridian arc length*
$\phi(G/Q_m)$ = *Latitude for scaled meridian arc*

$$K(\phi) = \frac{(1 + e'^2 \cos^2(\phi))^2}{2\,(m_0\,c)^2} = \tfrac{1}{2}\text{ Gaussian curvature gauge (scaled)}$$

$$\frac{dK}{d\phi} = -\frac{(1 + e'^2 \cos^2(\phi))\; e'^2 \sin(2\,\phi)}{(m_0\,c)^2} \approx o(2\,e'^2\,K(\phi)\sin(2\,\phi))$$

t = *Observed direction on the ellipsoid*
T = *Mapped direction*
s = *Observed distance on the ellipsoid*
S = *Mapped distance*

m = *Local scale in a point*
g = *Meridian convergence in a point*

The formulae give the mapped values corresponding to the geodetic values, i.e. the observed values on the ellipsoid contingently reduced for deflections of the vertical, height, slope etc. if needed and desired.

The use of "station" and "object" for distances is quite reasonable, because distance observations are frequently made in sets like directions from one standpoint to several other points.

(4.9) Corrections: Geodetic Observations to Mapped Observations

DIRECTIONS:

$$T = t - \Delta N\, E_3\, K(\phi_m)\left(1 - \tfrac{2}{3}\, E_3^2\, K(\phi_m)\right)$$

$$E_3 = E_m - \tfrac{1}{6}\,\Delta E$$

$$\phi_m = \phi(N_m) \qquad ;\ \text{see (4.8) and (III.4)}$$

DISTANCES:

$$S = s\, m_0 \left(1 + E_\mu^2\, K(\phi_m)\ (1 + \tfrac{1}{6}\, E_\mu^2\, K(\phi_m))\right)$$

$$E_\mu^2 = (E_m^2 + \tfrac{1}{12}\,(\Delta E)^2)$$

$$\phi_m = \phi(N_m/Q_m) \qquad ;\ \text{see (4.8) and (III.4)}$$

The scaled meridian arc length unit:

$$Q_m = m_0 \frac{a}{1+n}\, M_0 = m_0 \frac{a}{1+n}\, (1 + \tfrac{1}{4} n^2 + \tfrac{1}{64} n^4) \qquad \text{(A)}$$

$$m_0 = 1.0 \qquad \text{(Gauss -Krüger)}$$

$$m_0 = 0.9996 \qquad \text{(UTM)}$$

The corrections are functions of the transversal coordinates, but a parametric latitude appears in the formulae, so the half Gaussian curvature gauge $K(\phi)$ apparently needs a computation of a latitude, viz. the latitude of a point on the central meridian with the northing N equal to the northing parameter (and the easting equal to the easting value at the central meridian). The variation of K is rather small, so a common (mean) value in many cases will be precise enough. The latitude and the corresponding N for the directions should be evaluated for the 1/3 -point, but N_m can be used instead. The N_m can also be replaced by N_{stn} and it is even possible to use a common value for some hundreds of km, if the desired precision is modest.

The meridian convergence, defined as the angle from the mapping of the meridian in a point to the direction of the northing axis, can be computed by the formulae in (4.10). If the geodetic azimuth for a given direction T is computed from the coordinates then both the meridian convergence and the direction correction must be subtracted. The astronomical azimuth differs from the geodetic one by the effect of the local plumb-line deflection.

The trigonometric series in (4.10) can also be used for finding the local scale. Unfortunately the latitude of the point is required for finding the arc length on the central meridian corresponding to that latitude. The local scale can be found much easier from the formula for scale correction of a finite line as shown in (4.9) by putting the eastings of station and object equal and assuming a geodetic side length s of unity. The mapped length S will then be the local scale.

The summation of the trigonometric series as a complex cosine Clenshaw summation is more convenient than a direct computation of the imaginary part alone.

Note that N is counted from the equator and E is counted from the central meridian.

The formulae in (4.7) and (4.8) may be simplified further, if the width of the zone (span of eastings) is sufficiently small, say, 200 km, and if the maximum length of the lines is modest, say, 10 km. The terms of the fourth order in the formulae for the directions and distances may be removed, and the trigonometric series used for the meridian convergence may be reduced to the term with C_2 or fully dropped. A mean value of K can be used for a span of northings of 400 km.

(4.10) Meridian Convergence (and Local Scale)

$y = N/Q_m$; *Normalized northing*

$x = E/Q_m$; *Normalized easting*

ϕ = *Latitude of the point (N, E)*

$A = G(\phi)/Q_m$; *Normalized scaled central meridian arc length*

MERIDIAN CONVERGENCE:

$$\mu = \operatorname{atan2}(\sin(y)\tanh(x),\ \cos(y)) - \Im\left(\sum_{\kappa=1}^{4} C_{2\kappa}\cos(2\kappa(y+ix))\right)$$

GEODETIC AZIMUTH (from $T = \operatorname{atan2}(E_{obj} - E_{stn},\ N_{obj} - N_{stn})$ *) :*

$$\alpha = T - \mu - \left(-\Delta N\ E_3\ K(\phi_3)\left(1 - \frac{2}{3}\ E_3^2\ K(\phi_3)\right)\right) \quad ; \quad \textit{see (4.8) and (III.4)}$$

LOCAL SCALE:

$$\sigma = m_0 \frac{\sqrt{\cos^2(y) + \sinh^2(x)}}{\cos(A)} \exp\left(\Re\left(\sum_{\kappa=1}^{4} C_{2\kappa}\cos(2\kappa(y+ix))\right) - \sum_{\kappa=1}^{4}\cos(2\kappa A)\right)$$

$$C_2 = +\frac{1}{2}n - \frac{3}{8}n^2 + \frac{3}{32}n^3 - \frac{55}{1152}n^4 \qquad C_4 = +\frac{5}{16}n^2 - \frac{1}{3}n^3 + \frac{2421}{772}n^4$$

$$C_6 = +\frac{83}{480}n^3 - \frac{173}{899}n^4 \qquad C_8 = +\frac{1531}{336}n^4$$

LOCAL SCALE (alternative):

$$\sigma = m0\left(1 + E_\mu^2 K(\phi_m)\ (1 + \frac{1}{6}E_\mu^2 K(\phi_m))\right)$$

$$E_\mu^2 = (E_m^2 + \frac{1}{12}(\Delta E)^2)$$

$$\phi_m = \phi(N_m/Q_m) \quad . \quad ; \quad \textit{see (4.8) and (III.4)}$$

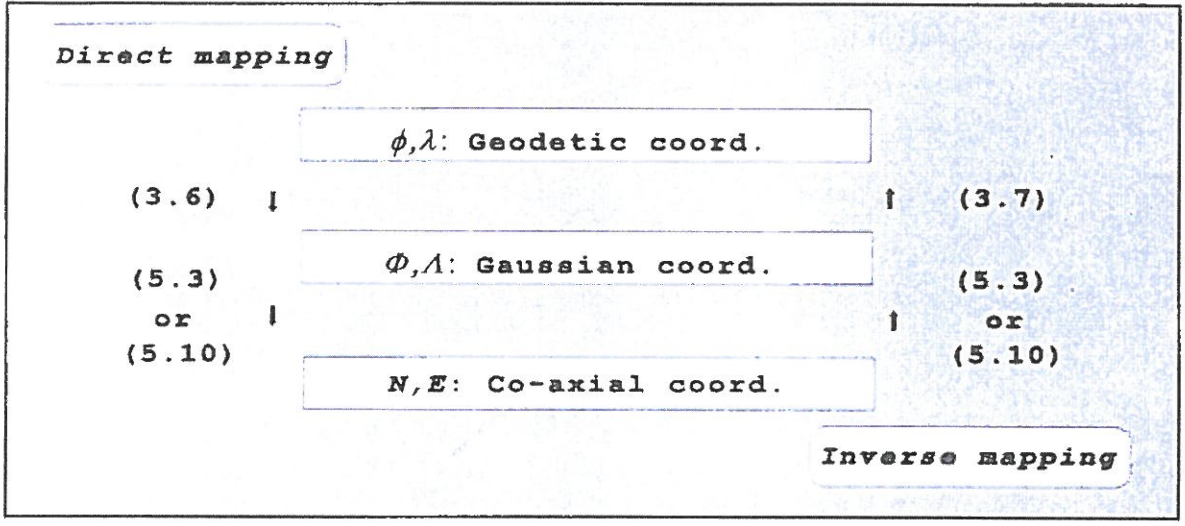
5. Co-axial Mapping

The co-axial mappings treated here are based upon

1. The complex isometric coordinates, giving the Mercator mapping, or

2. Exponential functions of the complex isometric coordinates, giving the Lambert conformal conical mapping and the polar stereographic mapping.

Fig. 5.1 Co-axial Mapping: Transformation Sequences



The diagram shows the sequence of dual subtransformations composing a regular transformation leading to a direct co-axial transformation or a reverse one.

The co-axial mappings are defined by differential expressions to the geodetic coordinates just as with the transversal mapping in Chapter 4, and the Gaussian sphere is likewise used as a parametric sphere for convenient formulae.

(5.1) Constants for Co-axial Mapping

ϕ_m = *Central latitude (of minimal scale)*

λ_0 = *Longitude of central meridian*

ϕ_0 = *Latitude for* N_0 *(optional)*

ϕ_1, ϕ_2 = *Stand. parallels (Lambert only)*

S = *Scaled, metric length unit*

m_0 = *Scale at central latitude*

k = *Cone-constant (Lambert and polar stereogr.)*

N_0, E_0 = *Convenience constants for coordinates*

The constants for the co-axial mappings should be computed once before the actual transformation is executed and stored in structure, which becomes a common parameter for that particular transformation, see Chapter 9.

5.1 Mercator Mapping

The complex isometric coordinates are defined in (2.7).

(5.2) Mercator mapping, Scale and Metric Length Unit

$$\psi_c = \ln\left(\tan(\pi/4 + \phi/2)\left(\frac{1 - e\sin\phi}{1 + e\sin\phi}\right)^{e/2}\right) + i\,\lambda$$

$$\mu = \frac{S\,(d\,\psi + i\,d\,\lambda)}{r(\phi)\,(d\,\psi + i d\,\lambda)} = \frac{S}{r(\phi)} = m \qquad \text{; complex and real scale}$$

$$g = \Im(\mu) = 0 \qquad \text{; meridian convergence}$$

$$S = r(\phi_m) = N(\phi_m)\cos(\phi_m) \qquad \text{; unit scale at latitude } \phi_m$$

$$N + i\,E = N_0 + i\,E_0 + S\,\psi_c \qquad \text{; Mercator mapping}$$

$$m(\phi) = V(\phi)\,\cos(\phi_c)/(V(\phi_m)\,\cos(\phi)) \qquad \text{; scale at } \phi$$

The meridians and the latitude parallels are mapped as straight lines, and the scale depends only on the latitude, being 1 on the latitude ϕ_c selected for the actual mapping. The minimum of scale is always on the equator, and the scale varies inversely proportionate to the cosine of the actual latitude. As the meridian convergence is zero, the loxodrome (or rhumb line), which intersect all meridians at the same angle will be a straight line. The constants N_0 and E_0 may be selected for convenience, e.g. to avoid having negative coordinates. The longitude is relative to a central meridian, the value of which must be taken into account in the computation.

The formulae for direct and reverse transformation uses the Gaussian sphere as a parameter surface for convenient formulae.

(5.3) Geodetic Coordinates ⟺ Mercator Coordinates

→ ***Direct mapping:*** $(\phi, \lambda) \rightarrow (N, E)$

$$\lambda = \lambda - \lambda_0 \qquad \text{; subtract the central longitude}$$

$$\Phi = \phi + CS(e[4], 2\,\phi)$$

← $N + i\,E = N_0 + i\,E_0 + S(\ln\,\tan(\pi/4 + \Phi/2) + i\,\lambda)$

→ ***Inverse mapping:*** $(N, E) \rightarrow (\phi, \lambda)$

$$\Phi = 2\arctan(\exp((N - N_0)/S)) - \pi/2$$

← $\phi = \Phi + CS(G[4], 2\,\Phi)$

← $\lambda = (E - E_0)/S + \lambda_0$

5.2 Conical Mapping

The conformal conical mapping (Lamberts conformal conical projection) is the exponential function of the complex isometric latitude multiplied with a constant. The complex function values are finally converted to metric units by multiplication with a constant in metric units.

(5.4) Conical Mapping. Definition

$$\psi_c = \psi + i\lambda = -\ln(\tan(P/2)) + i\lambda$$

$$-S\exp(-k\psi_c) = -S\exp(-k\psi)\exp(-ik\lambda)$$

$$= -S\tan^k(P/2)\exp(-ik\lambda) = N + iE \qquad \textit{; Polar and rectang. coord.}$$

$$\textit{Complex scale:}\quad \mu = \frac{dN + i\,dE}{r(\phi)\,(d\psi + i\,d\lambda)} = \frac{kS\tan^k(P/2)\exp(-ik\lambda)}{r(\phi)}$$

$$\textit{Scale:}\quad m = \frac{kS\tan^k(P/2)}{r(\phi)} \qquad (A)$$

$$\textit{Meridian convergence:}\quad g = arg(\mu) = -k\lambda$$

$$\ln m = \ln(kS) - k\psi - \ln r(\phi)$$

$$\frac{d\ln m}{d\phi} = -k\frac{d\psi}{d\phi} - \frac{-M(\phi)\sin(\phi)}{r(\phi)} = (\sin(\phi) - k)\,\frac{M(\phi)}{r(\phi)}$$

$$\frac{d\ln m}{d\phi} = 0 \qquad \textit{for } k = \sin(\phi), \quad \textit{where } \frac{d^2\ln m}{d\phi^2} = \frac{1}{V^2(\phi)} > 0$$

The mapping coordinates may also be regarded as polar coordinates with the radius $S\tan^k(P/2)$ and argument $-k\lambda$, so that latitude parallels are mapped as circles and meridians as straight lines. $(+\pi/2, \lambda_0)$ or $(-\pi/2, \lambda_0)$ with λ_0 as the central longitude of the mapping. The specific longitude even on the pole gives the definition of the coordinate axes.

(5.5) Rules for P and $\tan^k(P/2)$

$$\Phi = \phi + CS(e[4], 2\phi) \qquad \textit{; The Gaussian latitude}$$

$$P_n = \pi/2 - \Phi \qquad \textit{; Polar distance from the North Pole}$$

$$P_s = \pi/2 + \Phi \qquad \textit{; Polar distance from the South Pole}$$

$$\textit{Assume } k < 0\textit{: i.e. } -k = abs(k)$$

$$\tan^k(P_n/2) = \cot^k(P_s/2) = \tan^{abs(k)}(P_s/2)$$

$$\textit{Rule for P:}\quad P = \pi/2 - \Phi;\ \textit{if } (k<0)\ P = \pi - P$$

The polar distance P should be counted from the South Pole, when the mapping is on the southern part of the ellipsoid. The sign of k can be used to do this simply as shown in (5.5). The values of k, the cone constant, and S, the metric scale unit, determine the final formulae for the mapping. In principle any value of k may be used, but 0 gives no meaning, and +1 or -1 will actually give a stereographic mapping. Here only -1 ≤ k ≤ +1 will be considered. Minimum of the scale is found at a latitude where the sine of that latitude equals *k*. The value of *S* is negative for the southern hemisphere.

(5.6) Scale and Metric Length Unit Defined by 1 Parallel

$$\rightarrow \quad \textit{Standard latitude parallel: } \phi_m$$

$$\leftarrow \quad k = \sin(\phi_m)$$

$$P_m = \pi/2 - (\phi_m + \mathrm{CS}(e[4],\, 2\phi_m)) \qquad \textit{; and } P_m \textit{ by } (5.5)$$

$$\rightarrow \quad m = k\,S\,\frac{\tan^{abs(k)}(P_m/2)}{r(\phi_m)} \qquad (\textit{mostly } m = m_0 = 1)$$

$$\leftarrow \quad S = m\,\frac{N(\phi_m)}{\tan(\phi_m)\,\tan^{abs(k)}(P_m/2)} \qquad (\textit{NB}: k < 0 \rightarrow S < 0)$$

It is seen that the scales for varying latitudes are almost symmetric with respect to the minimum scale. It is therefore possible to find pairs of latitudes with the same scale, or inversely for a given pair of latitudes with the same scale to find the latitude of minimum scale, provided that the two latitudes are sufficiently different.

The constants *k* and *S* needed for a conical mapping defined by two latitude parallels are found by setting the scale at the two selected parallels equal m_e.

(5.7) Scale and Metric Length Unit Defined by 2 Parallels

$$\rightarrow \quad \textit{Latitudes with scale } m_e\textit{: } \phi_1 \textit{ and } \phi_2 \qquad \textit{; } P_1 \textit{ and } P_2 \textit{ by } (5.5)$$

$$\rightarrow \quad m_e = k\,S\,\frac{\tan^{abs(k)}(P_1/2)}{r(\phi_1)} = k\,S\,\frac{\tan^{abs(k)}(P_2/2)}{r(\phi_2)}$$

$$= k\,S\,\frac{\tan^{abs(k)}(P_1/2) + \tan^{abs(k)}(P_2/2)}{r(\phi_1) + r(\phi_2)}$$

$$\leftarrow \quad k = \frac{\ln\,(r(\phi_2)/r(\phi_1))}{\ln\,(\tan(P_2/2)/\tan(P_1/2))}$$

$$S = \frac{m_e}{k}\,\frac{r(\phi_1) + r(\phi_2)}{\tan^{abs(k)}(P_1/2) + \tan^{abs(k)}(P_2/2)}$$

$$\leftarrow \quad \phi_m = \arcsin(k)$$

$$\leftarrow \quad m_0 = k\,S\,\frac{\tan^{abs(k)}(P_m/2)}{r(\phi_m)}$$

It is evident from (5.7) that the case of two parallels always can be reduced to one standard parallel with a minimum scale less than unity.

The most used case with two standard parallels appears to be based upon $m_e = 1$.

(5.8) Unit Scale and Metric Length Unit Defined by 2 Standard Parallels

$$\begin{aligned}
\leftarrow \quad k &= \frac{\ln (r(\phi_2)/r(\phi_1)}{\ln (\tan(P_2/2)/\tan(P_1/2))} \\
\leftarrow \quad \phi_m &= \arcsin(k) \\
\leftarrow \quad S &= \frac{1}{k}\,\frac{r(\phi_1) + r(\phi_2)}{\tan^{abs(k)}(P_1/2) + \tan^{abs(k)}(P_2/2)} \\
\leftarrow \quad m_0 &= k\,S\,\frac{\tan^{abs(k)}(P_m/2)}{r(\phi_m)} = \frac{r(\phi_1) + r(\phi_2)}{\tan^{abs(k)}(P_1/2) + \tan^{abs(k)}(P_2/2)}\,\frac{\tan^{abs(k)}(P_m/2)}{r(\phi_m)}
\end{aligned}$$

The scale is equalized over a certain interval of latitude, where the two standard latitudes are selected to this aim, so that the full latitude interval is equalized. However, it is possible to use two limiting latitudes for the mapped area, so that the equalizing effect over the interval is directly determined. Conditions like $m_e\, m_0 = 1$ or $m_e - 1 = 1 - m_0$ may be used in (5.7) to give the constants k and S, which now will be based upon the geometric mean or the arithmetic mean of the maximum and minimum scale.

It may be convenient to add constants to N and E, e.g. to avoid negative values. The constants may be selected freely, but may also be related to the actual mapping, so that there is a computable relation between geographical coordinates and the added constants. An example of defining such constants is shown below.

(5.9) Choice of N_0 and E_0

$$\begin{aligned}
\rightarrow \quad & (\phi_z, 0) \Leftrightarrow (N_z, E_z) \\
& \Phi_z = \phi_z + \mathrm{CS}(e[4], 2\phi_z) \qquad ; P_z \text{ by } (5.5) \\
\leftarrow \quad & N_0 = N_z + S \tan^{abs(k)}(P_z/2) \\
\leftarrow \quad & E_0 = E_z
\end{aligned}$$

The Lambert conformal conical mapping has earlier been much used for topographic mapping and is still in use for large international map series as, e.g. the ICAO map sheets. The advantage is (maybe) that the map sheets can be limited by circles for the N and S borders and straight lines for the E and W borders.

The Lambert mapping has also been used for geodetic computations of local control networks, but the projection corrections for directions and distances are generally rather complicated or alternatively not precise. See e.g. (Bomford, 1962), for the complicated ones.

The algorithms for practical computation shown in (5.10) are valid both for positive and negative latitudes, using the rule in (5.5). The algorithm uses the (presumably precomputed) constants k and S so that it is independent of the mode used for standard parallels.

(5.10) Geodetic Coordinates ⟺ Conical Coordinates

→ *Direct mapping:* $(\phi, \lambda) \rightarrow (N, E)$

$\lambda = \lambda - \lambda_0$; *subtract central longitude*

$\Phi = \phi + \mathrm{CS}(e[4], 2\phi)$

$P = \pi/2 - \Phi$

if $(k < 0)$ $P = \pi - P$; *take supplementary angle*

← $N = N_0 - S \tan^{abs(k)}(P/2) \cos(k\lambda)$

← $E = E_0 + S \tan^{abs(k)}(P/2) \sin(k\lambda)$

→ *Inverse mapping:* $(N, E) \rightarrow (\phi, \lambda)$

$P = 2\arctan\left((hypot((N_0 - N)/S, (E - E_0)/S))^{1/abs(k)}\right)$

if $(k < 0)$ $P = \pi - P$; *take supplementary angle*

$\Phi = \pi/2 - P$

← $\phi = \Phi + \mathrm{CS}(G[4], 2\Phi)$

← $\lambda = atan2((E - E_0)/S, (N_0 - N)/S)/k + \lambda_0$

5.3 Stereographic Mapping

The polar stereographic mappings may be regarded as special cases of the conical mapping with $k = +1$ for mapping at the North Pole and $k = -1$ for mapping at the South Pole. The stereographic mapping with an arbitrary central point different from the Poles is not co-axial and is not treated here.

(5.11) Scale and Metric Length Unit for Polar Stereographic Mapping

Selected central latitude: $\phi_m \neq \pm\pi/2$, *initially*

$abs(k) = +1$; $p_m = \pi/2 - abs(\phi_m)$

$P_m = \pi/2 - \left(abs(\phi_m) + \mathrm{CS}(e[4], 2abs(\phi_m))\right)$

$$m_0 = k\,S\,\frac{\tan(P_m/2)}{r(\phi_m)} \qquad ;\ see\ (5.6)$$

$$= k\,S\,\frac{\tan(p_m/2)\left(\dfrac{1 + e\cos(p_m)}{1 - e\cos(p_m)}\right)^{e/2}}{N(\phi_m)\sin(p_m)}$$

$$= k\,S\,\frac{1}{2N(\phi_m)\cos^2(p_m/2)}\left(\frac{1 + e\cos(p_m)}{1 - e\cos(p_m)}\right)^{e/2}$$

$\phi_m = \pm\pi/2$ *now permitted*

$$m_0 = k\,S\,\frac{1}{2c}\left(\frac{1 + e}{1 - e}\right)^{e/2} \qquad ;\ c = N(\pm\pi/2)\ i.e.$$

– $$S = \frac{2}{k}\,m_0\,c\left(\frac{1 - e}{1 + e}\right)^{e/2} \qquad ;\ c = The\ polar\ curvature\ radius$$

(5.12) Scale, N_0, and E_0

$m_0 = 0.994$; *Universal Polar Stereogr. (UPS)*
$N_0 = E_0 = 2000\ km$; *UPS*

The algorithm in (5.10) for conical mapping may also be used for the polar stereographic mapping, but the call of a power-function for raising to the power of abs(k) and 1/abs(k) may of course be skipped. The central longitude is presumably zero, but if needed then it may be included as in (5.10).

(5.13) Geodetic Coordinates ⟺ Stereographic Coordinates

```
→   Direct mapping:  $(\phi, \lambda) \Rightarrow (N, E)$
      $\Phi = \phi + \mathrm{CS}(e[4], 2\phi)$
      $P = \pi/2 - \Phi$
          if $(S < 0)$ $P = \pi - P$
←     $N + iE = N_0 + iE_0 - S\tan(P/2)\,(\cos(k\lambda) - i\sin(k\lambda))$

→   Inverse mapping:  $(N, E) \Rightarrow (\phi, \lambda)$
      $\Phi = \pi/2 - 2\,\mathrm{atan}(\mathrm{hypot}((N_0 - N)/S,\ (E - E_0)/S))$
          if $(S < 0)$ $\Phi = -\Phi$
←     $\phi = \Phi + \mathrm{CS}(G[4], 2\Phi)$
←     $\lambda = \mathrm{atan2}((N_0 - N)/S,\ (E - E_0)/S)/k$
```

The stereographic mapping with an arbitrary central point may present some problems, when the ellipsoid is mapped, because the curvature varies with the azimuth. The Gauss-Krüger stereographic mapping is a double mapping to a Soldner sphere followed by a polar stereographic mapping of the sphere with the central point as the pole.

It has also been tried in some countries to improve the mapping by replacing the Soldner sphere with a Gaussian osculation sphere. It appears that the factor of 1/4 instead of the normal 1/2 in the scale variation has influenced people, who regard the scale variation as a distortion, to select the stereographic mapping in preference to the transversal Mercator mapping for national coordinate systems.

6. Cartesian 3-d Coordinates

The transformation from geodetic coordinates to 3-d cartesian coordinates is quite straightforward. A problem may turn up if the (ellipsoidal) height is not known. The automatic action is to assume that the point under transformation is on the ellipsoid, i.e. the height is zero. Alternatively an available "physical" height (i.e. a geopotential number, a normal height, or an orthometric height) may be updated to an ellipsoidal height by some kind of geoid height or height anomaly. If the main intention is to preserve the horizontal position information, then rather modest precision is required for the height component. The algorithm is:

(6.1) Geodetic Coordinates + Ellipsoidal Height → 3-d Cartesian Coordinates

$$
\begin{aligned}
Z_1 &= (N(\phi) + h) \cos\phi \cos\lambda \\
Z_2 &= (N(\phi) + h) \cos\phi \sin\lambda \\
Z_3 &= (N(\phi)(1-f)^2 + h) \sin\phi \\
&= (N(\phi) + h - e^2 N(\phi)) \sin\phi
\end{aligned}
$$

The inverse algorithm presented here uses iteration. The quantity $N(\phi) + h$ is considered to be one unknown. It is therefore necessary to compute the value of $N(\phi)$ after each iteration step to help the computation of Z_3. The present algorithm requires 3 or 4 iterations, but performs better the larger the h is.

(6.2) 3-d Cartesian Coordinates → Geodetic Coordinates + Ellipsoidal Height

```
$N_1 = \frac{1}{2} e^2 a$
$Z_h = \sqrt{Z_1^2 + Z_2^2}$
$\lambda = atan2(Z_2, Z_1)$

        Iterate:   (3-4 iterations needed)
{
        $\phi = atan2(Z_3 + N_1, Z_h)$
        $N_1 = e^2 N(\phi) \sin\phi$
}

$h = \sqrt{(Z_3 + N_1)^2 + Z_h^2} - N(\phi)$        (sic!)
```

The computation of geodetic coordinates and ellipsoidal heights from 3-d cartesian coordinates is iterative. Note that N+h is treated as one unknown, giving a fast convergence even if the ellipsoidal height h is of the order of a satellite height. The only blemish is the floating point subtraction of N from N+h. The de facto standard of 16 digits computing precision makes this blemish almost invisible.

7. Predictions

Predictions are used as substitutes for transformations in situations where a regular transformation is not feasible, e.g. because a proper datum definition of one or both systems does not exist. The applications may e.g. be (1) in an adjustment of a network in a system as the Danish System 1934 with GPS data (Engsager, 1997) or (2) providing preliminary coordinates for a network adjustment.

The name *predictions* is used instead of *transformations* in order to underline, that the coordinates resulting from such a process are based upon empirically determined relations found from corresponding coordinates in the two coordinate systems and not by a rigorous system of algebraic formulae. There is a certain analogy with the prediction of the gravity in a point from known gravity values in a net or array of points, but in this case a background of physical reality is found.

Problem: For two geodetic coordinate systems, A and B, where a number of stations have coordinates in both systems, a method is wanted for prediction of unknown coordinates in one of the systems from known coordinates in the other system.

The prediction of coordinates is a mapping of coordinates (N_A, E_A) in a System A to coordinates (N_B, E_B) in a System B based on an empirical transfer structure T_{AB}. The transfer structure may be a vector, a matrix, a table, or a file, depending on the type of the prediction function. The aim is a reasonable precise and consistent prediction of coordinates covering a large area as a whole national network. It is evident, that the attainable accuracy cannot be expected to be better than the accuracy of neighbour stations in each system, because the source data has a finite noise.

A subdivision in smaller areas in order to obtain higher precision and/or more simple functions may be used, but then special precautions may be required in order to avoid inconsistencies or ambiguities near the borders of the areas. See e.g. (Dinter, Illner and Jäger, 1996). However, the examples in Sec. 7.5 show what can be obtained for areas of different sizes.

A prediction should be dual, i.e. predict both from A to B and from B to A. It should be noted that the computation accuracy may pretty well be much higher than the accuracy of the coordinates. This is useful for the automatic computation checking, because it is then possible to see if a prediction is attempted for a station situated outside the area where the prediction function is valid.

7.1 Prediction Methods

The Helmert transformation and the affine transformation are two well-known methods for prediction of coordinates based upon simple formulae with constants determined from corresponding coordinate values in the two systems, where a prediction is wanted.

We have used generalization of these methods by means of polynomials of a suitably high degree - complex polynomials for Helmert transformation and pairs of general polynomials for affine transformation.

It is assumed that both systems are in rectangular coordinates. Geographical coordinates are not isometric, and a multiplication of longitudes with the cosine of a mean latitude will create an unnecessary discrepancy, which the method would have to cope with. Furthermore, the geometry on the ellipsoid is non-Euclidean, so that translation in longitude corresponds to a varying linear size and rotations refer to the Christoffel length ("the reduced length"). The remedy is to use rectangular coordinates.

The earlier systems for predictions between UTM coordinates and the Danish cadastral System 1934 used complex polynomials of third degree, (Andersson, 1981), followed by interpolation from the residual differences tabulated in 80 first order stations, (Poder, 1989). The method gave unique answers, but the consistency was not satisfactory.

Collocation methods have been used as reported by (Ehlert and Strauss, 1990), based upon 300 stations. As we shall produce predictions based upon several thousands of stations, it is assumed that collocation was not so well fitted for in this case. However, collocation will ensure that all coordinates used for creating the prediction function will retain their original values.

The methods based upon polynomials will in general give a certain noise in the predicted values for the coordinates used for the determination of the polynomials. This is the price to be paid for the relatively simple method. The last implementations were made several years ago at a time where we were not aware of the non-linearity of the problem as demonstrated by (Teunissen, 1985) and (Borre, 1990). The very much increased number of unknowns - in our case of the order of 50000 - is no serious problem, because an elimination technique in analogy with that is used for orientation unknowns and photogrammetric model parameters.

The names of the variables in this chapter are slightly different from the names used otherwise in the present work.

(7.1) Names of Variables

$P(T\ ,\ N,\ E)$ = *Prediction function*

$N_A,\ E_A$ = *Coordinates in system A*

$N_B,\ E_B$ = *Coordinates in system B*

$N,\ E$ = *Coordinates with center values* $N_0,\ E_0$ *subtracted*

T_{AB} = *Transfer structure for* $(N_A,\ E_A) \rightarrow (N_B,\ E_B)$

T_{BA} = *Transfer structure for* $(N_B,\ E_B) \Rightarrow (N_A,\ E_A)$

7.2 Predictions with Complex Polynomials

A conformal coordinate system can be mapped on another conformal system by means of a complex polynomial, at least if they are on the same datum in the sense defined in Sec. 1.2, because they are always derived from the same geographical coordinates. This would so far not be of interest here because predictions are intended as an emergency when the datum definition is unknown or uncertain. Nevertheless, if both systems are reasonably homogeneous, then complex polynomials should be tried, because the number of parameters is considerably smaller than for general polynomials.

(7.2) Complex Polynomials

$$N_B + i\,E_B = P(T_{AB}, N_A, E_B) \qquad ; \textit{General prediction function}$$

$$N_B + i\,E_B = \sum_{k=0}^{d}\big((N_A - N_0) + i\,(E_A - E_0)\big)^k (R_k + i\,I_k)$$

$$T_{AB} = \{R_0, I_0, R_1, I_1, R_2, I_2, \ldots . R_d, I_d\}$$

$$N_0,\ E_0 = \textit{Center ("mean") value of coordinates}$$

The prediction function will in this case be a computation of a complex polynomial, and the T-structure will be the coefficients of a complex function and possibly (N_0, E_0), some centre coordinates to be subtracted from the input coordinates for numerical reasons.

The summation of a polynomial may give a floating point overflow if the input coordinates are widely outside of the area covered by the prediction. Therefore some limit for permitted coordinates may also be included in T_{AB} to be used for trapping a floating point overflow and calling an alarm routine instead. The prediction functions used in practise are dual, just like the regular transformation functions.

The summation algorithm, a Horner scheme, is shown as a piece of C-code taken from the KMS transformation system. The T structure may be declared as "double T[2*(d+1)]". The input coordinates y, x is relative to a centre. The values of N are also returned as function return values, so that the function call may be used in arithmetic statements.

TXT 7.2 Functiom cpol

```
double cpol(double *TC, int g, double y, double x, double *N, double *E)
/*____________________________________________________________________*/
{
  double     *tcp;
  double     R = 0.0, I = 0.0, Z;

  for (tcp = TC + 2*g + 2; tcp > TC;  ) {
    Z = x*R + y*I + *--tcp;
    R = y*R - x*I + *--tcp;
    I = Z;
  }
  *N = R;
  *E = I;
  return (R);
}
```

7.3 Predictions with General Polynomials

Using general polynomials instead of complex polynomials means that the mapping is no longer assumed to be conformal. The polynomials in two variables used for prediction have only coefficients up to a maximum degree as shown in the example in (7.3), where the maximum degree is 3.

(7.3) Polynomium with max. Degree = 3

$$\begin{aligned} Q(x,y) = \; & (p_{0,0} + Ep_{0,1} + E^2p_{0,2} + E^3p_{0,3}) \\ & + N\,(p_{1,0} + Ep_{1,1} + E^2p_{1,2}) \\ & + N^2(p_{2,0} + Ep_{2,1}) \\ & + N^3(p_{3,0}) \end{aligned}$$

The common, complex prediction function is replaced by one function for each coordinate, so that the prediction functions become general polynomials in two variables truncated to a maximum sum of the powers of the variables as shown in (7.4), where the matrices have zeroes when the sum of the powers of the variables exceeds the selected maximum degree.

(7.4) Prediction with General Polynomials

$$V_N^t = \{1, N, N^2, N^3, \ldots N^d\} \qquad V_E^t = \{1, E, E^2, E^3, \ldots E^d\}$$

$$N_B = P(T_{AB}^{(N)}, N, E); \qquad E_B = P(T_{AB}^{(E)}, N, E);$$

$$DIM(T_{AB}^{N}) = (d+1, d+1) \qquad DIM(T_{AB}^{E}) = (d+1, d+1)$$

$$T_{AB}^{(N)} = \left\{\begin{array}{ccccccc} * & * & * & \ldots & * & * & * \\ * & * & * & \ldots & * & * & 0 \\ * & * & * & \ldots & * & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ * & * & * & \ldots & 0 & 0 & 0 \\ * & * & 0 & \ldots & 0 & 0 & 0 \\ * & 0 & 0 & \ldots & 0 & 0 & 0 \end{array}\right\} \qquad T_{AB}^{(E)} = \left\{\begin{array}{ccccccc} * & * & * & \ldots & * & * & * \\ * & * & * & \ldots & * & * & 0 \\ * & * & * & \ldots & * & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ * & * & * & \ldots & 0 & 0 & 0 \\ * & * & 0 & \ldots & 0 & 0 & 0 \\ * & 0 & 0 & \ldots & 0 & 0 & 0 \end{array}\right\}$$

$$N_B = V_N^t\, T_{AB}^{(N)}\, V_E = \sum_{j=0,\,k=0}^{j+k \le d} n_{jk}\, N^j E^k \qquad E_B = V_N^t\, T_{AB}^{(E)}\, V_E = \sum_{j=0,\,k=0}^{j+k \le d} e_{jk}\, N^j E^k$$

The transfer structures T are quadratic matrices, but all elements below the bidiagonal are zero as indicated in (7.4). The elements with coefficients are marked by a *. Each row in the matrices contains the coefficients of a polynomial in one of the coordinates, and the values of these polynomials are the coefficients of a polynomial in the other coordinate. The maximum degree of the sub-polynomials is thus less or equal *d*. The total number of coefficients is *(d+1+1)(d+1)/2* for each coordinate, i.e. a total of *(d+2)(d+1)* for a prediction. The matrix is in practise placed in a one-dimesional array without the given zeroes. The summation can be made by a double Horner scheme as shown in the piece of C-code.

The coefficients of each polynomial are arranged in a linear array, and as each coefficient only is addressed once, then a simple backward counting will point at the proper element when it is needed.

TXT 7.3 General Polinimial Summation

Function call:
N_B = gpol($T_{AB}^{(N)}$, d, N, E); /* Prediction of N_B */
E_B = gpol($T_{AB}^{(E)}$, d, N, E); /* Prediction of E_B */

Function gpol:

```
double gpol(double *TC, int g, double y, double x)
/*_______________________________________________*/
{
  double   *ptc, hsum, psum;
  int      r, c;

  ptc = TC + (g + 2)*(g + 1)/2;
  for ( psum = *--ptc, r = g; r > 0;  r--) {
    for (hsum = *--ptc, c = g; c >= r; c--)
      hsum = y*hsum + *--ptc;
    psum = x*psum + hsum;
  }
  return (psum);
}
```

7.4 Determination of the Coefficients

The determination of the coefficients of the polynomials uses pairs of coordinates for the same station in the two systems in a least squares adjustment. Each pair gives two equations, which may be immediately transferred to the normal equations, so that they do not have to be stored. The equations are linear, but it may be an advantage to repeat the solution in order to test for errors and monitor the normal equations for computing noise. The right-hand side of the equations in (7.3) therefore contains the prediction function, so that the residuals after the first pass through the normal equations can be compared with the standard deviation from the adjustment.

The C-function cpow will produce the successive powers of $y + ix$, separate the real part and the imaginary part and place them in the observation equation matrix.

The degree of the polynomials may be started with a low value, and then increased in steps of 1 until the standard deviation no longer decreases, i.e. when the natural noise of the coordinates and the lacking conformity of mapping is apparent. Clearly only a reasonable degree is realistic. It is found in practise that no more than degree of 5 for complex polynomials is realistic. The inverse prediction should of course also be found, so that an active control always is available. The inverse polynomial may be found by repeating the adjustment with swapping of the two sets of coordinates, or by direct inversion of the polynomials.

(7.5) Observation Equations and Recursion Formulae

$$R_k^{(i+1)} = R_k^{(i)} + \Delta R_k^{(i)} \qquad R_k^{(0)} = 0$$

$$I_k^{(i+1)} = I_k^{(i)} + \Delta I_k^{(i)} \qquad I_k^{(0)} = 0$$

$$\sum_{k=0}^{d}\left(\Re((y+ix)^k)\,\Delta R_k - \Im((y+ix)^k)\,\Delta I_k\right) = N_B - cpol(T,\ d,\ y,\ x,\ \&N_r,\ \&E_r)$$

$$\sum_{k=0}^{d}\left(\Im((y+ix)^k)\,\Delta R_k + \Re((y+ix)^k)\,\Delta I_k\right) = E_B - E_r$$

Recursion for $(y+ix)^k$: $\quad (y+ix)^k = (y+ix)\,(y+ix)^{k-1}$

$$\Re((y+ix)^k) = y\ \Re((y+ix)^{k-1}) - x\ \Im((y+ix)^{k-1})$$

$$\Im((y+ix)^k) = y\ \Im((y+ix)^{k-1}) + x\ \Re((y+ix)^{k-1})$$

The coefficients of the general polynomials can be found by a simple least squares adjustment. The observation equations are linear, but it is useful to iterate the solutions, not least if some kind of error snooping or blunder defence is used. Therefore the word "centre value" instead of "mean value" is used, because some of the coordinate data may be down weighted during the blunder defence actions.

The coefficients of the "observation equations" are y and x in the powers and combinations found in the polynomials. The normal equation matrix is the same for the finding of the two polynomials, so only the right-hand sides and the solutions are belonging to the polynomials for y and x, resp. The standard deviations of unit weight delivered will provide an indication of the quality of the prediction, and the inverted matrix will give information of how well the coefficients are determined.

The approach is to start with finding the coefficients for polynomials with a relatively low degree, say, 3-4, and by and by increase the degree until the standard deviation of unit weight becomes stationary. This should happen not later than at a degree of 4-6 for complex polynomials and 10-12 for general polynomials. The condition number of the normal equations increases with the degree, and finally the matrices become singular. Using the IEEE standard double floating points with 15-16 decimal digits precision this will occur at a degree of 15. The redundancy of the material should be very high, and certainly not less than 90%.

Both algorithms shown in TXT 7.4 and 7.5 will map the coefficients on a one-dimensional array. The mapping for the general polynomials neglects all elements below the bidiagonal of the matrices shown in (7.4).

TXT 7.4 Coefficients for Complex Polinimial Determination

```
void cpow(double y, double x, double *cy, double *cx, int g)
/*______________________________________________________________*/
{
 int    i;
 double  R = 1.0, I = 0.0, Z;
 *(cy++) =  R;  *(cy++) = -I;
 *(cx++) =  I;  *(cx++) =  R;

 for (i = 1; i <= g; i++) {
  Z = R*y - I*x;
  I = R*x + I*y;
  R = Z;

  *(cy++) =  R;  *(cy++) = -I;
  *(cx++) =  I;  *(cx++) =  R;
 }
}
```

TXT 7.5 Coefficients in General Polinimial Determination

```
void   gpow(double y, double x, double *cy, int g)
/*__________________________________________________*/
{
 int     r, c;
 double   R, C;
 for (R = 1.0, r = g; r >= 0; R *= x, r--)
  for (C = R,   c = r; c >= 0; C *= y, c--)
   *(cy++) = C;
}
```

7.5 Examples

Predictions have been used in practise for relating the coordinates in the cadastral Danish System 1934 and the UTM ED50 system used in the topographical mapping. The areas Jylland-Fyn, Sjælland, and Bornholm have each a set of general polynomials for both ways. The internal computing consistency is here better than 1 mm, but the standard deviation of a prediction is about 2 cm.

A similar result for prediction between ED87 and EUREF89 for the whole SCAN block (Denmark, Norway, Finland, and Sweden) produces a standard deviation of 5 cm.

The last example 3 is the prediction for a special coordinate system dedicated to the Store Bælt Bridge construction.

1. ED50, utm32 <==> s34j: utm zone 32, area 400*250km, general pol., degree 11, 2*78 coeff, 24000 stations, redundancy: 99.69 %, standard deviation: 2 cm

2. ED87 <==> EUREF89: utm zone 33, area 2000*2000km, general pol., degree 5, 2*21 coeff, 5000 stations, redundancy 99.58 %, standard deviation 5 cm

3. ED50, utm32 <==> SB-projection: area 30*15 km, complex pol. degree 3, 8 coeff, 300 stations, redundancy 98.7 %, standard dev. 2 cm.

7.6 Symmetric Prediction Functions

The observation equations shown above assume that only one of the two sets of coordinates is treated as observations, while the other is assumed to produce the "exact" coefficients. As two sets of coefficients are determined independently ("direct" and "inverse"), both sets of coordinates in turn play the rôle as observations and coefficient source. It turns out that the standard deviation for both is almost the same. However, it is rather simple to implement the ideas of symmetry put forward by Teunissen (1985).

(7.6) Non-linear Observation Equations

$$
\begin{aligned}
D_{n,y} &= \frac{\partial POL_n(....)}{\partial y} \\
D_{n,x} &= \frac{\partial POL_n(....)}{\partial x} \\
D_{e,y} &= \frac{\partial POL_e(....)}{\partial y} \\
D_{e,x} &= \frac{\partial POL_e(....)}{\partial x}
\end{aligned}
$$

$$
\begin{aligned}
D_{n,y}\,\Delta y + D_{n,x}\,\Delta x + \Sigma C_n\,\Delta c &= N_B - POL_n(....) \\
D_{e,y}\,\Delta y + D_{e,x}\,\Delta x + \Sigma C_e\,\Delta c &= E_B - POL_e(....) \\
\Delta y &= N_A - y \\
\Delta x &= E_A - x
\end{aligned}
$$

The quantities y and x are then considered as elements in the adjustments. Accordingly there will be 2 observation equations and unknowns more for each pair of coordinates. In one of the examples shown about 50000 more unknowns will appear, and for general polynomials both systems must be solved simultaneously in a common set of normals. The observation equations now must include the partial derivatives of the coordinates with respect to the y and x. It is rather easy to find the numerical values of the derivatives simultaneously with the Horner summation of the polinimial.

The $POL_n(....)$ and $POL_e(....)$ in (7.6) represents the real part and the imaginary part of the value of the polynomial in the complex case and the N-part and the E-part for the general polynomials. The summations of $C_n \Delta c$ and $C_e \Delta c$ represent the observation equations used in the linear case, and terms with the partial derivatives is the part corresponding to the elements y and x. Due to the non-linearity some preliminary value of the polynomial coefficients should be found from the linear approach described in the preceding sections.

The redundancy is not changed, because the increased number of unknowns is followed by exactly the same increase of the number of observations.

The normal equations can be handled rather easily by a very simple artifice. When the first observation equations for each pair of coordinates are formed, the two unknowns corresponding to y and x can be eliminated, because no further contributions from these observation equations will appear. This is a well-known artifice for handling unknowns like orientation unknowns and model parameters in network adjustment. The effective dimension of the normals is therefore only some few hundreds in the worst case, but it should be noted that the general case can no longer be treated as two independent sets of normals.

7.7 Concluding Remarks on Predictions

Prediction may be regarded as an emergency method, when more regular methods fail. It may have been used earlier for producing coordinates from an existing network adjustment in order to save a re-adjustment in the desired system. The available software for network adjustment today makes a re-adjustment of some thousands of stations to be a fairly easy and almost automatic running job, so if the observation data is available then a total re-adjustment will replace the prediction with production of coordinates from their natural source: observations.

One may in fact regard the observations as the original coordinates - they do actually coordinate two or more stations - and (in order to increase the confusion) regard the computed coordinates as a kind of derived observations giving the position of a point relative to an origo.

The most likely application of an established prediction function now and in the future are:

1. Production of preliminary coordinates for the start of a network adjustment in a desired system when (preliminary) coordinates only are available in different system(s). The required precision for preliminary coordinates is modest, and prediction is a much faster method for preliminary coordinates than, e.g. using preliminary, linear observation equations. Coordinates which should remain fixed in the adjustment should never be predicted.

2. Producing coordinates in a - possibly obsolete - system when the original observation data is not directly available for a re-computation. A good prediction between EUREF89 and ED50 could give reasonable ED50 coordinates for a new station determined in EUREF89.

8. Datum Shifts

Regular datum shifts can be implemented in several ways, many of them based upon differential formulae due to Vening Meinez, de Graff-Hunter, and Molodensky (HEISKANEN and MORITZ, 1967), (DOD, 1993). However, the direct method of computing 3-d cartesian coordinates, which can be transformed by a 3 to 7 parameter linear transformation, has the advantage of rigorous formulae void of high latitude problems and with an easily obtainable computing precision better than 50 microns. On the other hand, a datum shift is a "prediction", where the physical precision never will be better than the precision of the observation data.

If a datum shift is small then the sequence of translation, rotation, and scaling is irrelevant, but otherwise one must decide the sequences to be used for "direct" and for "inverse". The formulae in (8.1) are simple illustrating approximations of a more precise approach with 3 consecutive rotations, a scale change and a parallel shift. The practical solution is of course to use the rigorous 7-parameter formulae giving a safe numerical precision in all cases and with contingently unused parameters kept fixed to the value of nought.

It is recommended that all communications concerning datum shifts are given not only with the numerical values of the parameters but also with the numerical values of the formulae, because the sequences of translation, rotation, scaling, and sign conventions are critical for the numerical consistency.

The subscripts 1,2, and 3 in (8.1) correspond to the conventional X, Y, and Z- axes.

(8.1) 3-d Coordinate Shifts

Given system: $X = (X_1, X_2, X_3)$

Wanted system: $Y = (Y_1, Y_2, Y_3)$

Translations: $z = (z_1, z_2, z_3)$

Rotations: $\alpha_1, \alpha_2, \alpha_3$

Scale (change): m

Direct datum shift:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 1+m & \alpha_3 & \alpha_2 \\ -\alpha_3 & 1+m & \alpha_1 \\ -\alpha_2 & -\alpha_1 & 1+m \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} + \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

Inverse datum shift:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 1-m & -\alpha_3 & -\alpha_2 \\ \alpha_3 & 1-m & -\alpha_1 \\ \alpha_2 & \alpha_1 & 1-m \end{pmatrix} \begin{pmatrix} Y_1 - z_1 \\ Y_2 - z_2 \\ Y_3 - z_3 \end{pmatrix}$$

The datum shifts presented in Table 2 are based upon the logical concepts of parent datum and child datums. The datums are arranged in a parent level, where any datum shift is (virtually) possible, and a child level, where only datum shifts to or from the parent level are possible. Using a chain starting from child level, passing the parent level, and ending in a child level, any shift is feasible with relatively few definitions of shifts.

As an example of a block of 4 parent datums as used in earlier the KMS-systems we have

1. WGS84	(= W)
2. ED50	(= E)
3. NWL9D	(= N)
4. EUREF89	(= U)

The parent datums can be reached from any datum or reach any other datum. In principle any parent datum goes to any of the other 3 parent datums, but in order to avoid inconsistencies a strict sequence control is used - hidden to the ordinary user. The implementation is the usual state/action technique as shown in table 2.

STATE ⇒	**0**	**1**	**2**	**3**
INPUT ⇒	**WGS84**	**ED50**	**NWL9D**	**EUREF89**
OUTPUT ⇓	**(W)**	**(E)**	**(N)**	**(U)**
WGS84	**OUT/0**	**E→W/0**	**N→W/0**	**U→W/0**
ED50	**W→E/1**	**OUT/1**	**N→W/0**	**U→W/0**
NWL9D	**W→N/2**	**E→W/0**	**OUT/2**	**U→W/0**
EUREF89	**W→U/3**	**E→W/0**	**N→W/0**	**OUT/3**

Table 2: State/Action Table for a parent datum system.

As an example W→E/1 means do a datum shift from WGS84 to ED50 and go to state 1. OUT and unchanged state means that the desired datum is reached. It is seen that only 3 dual datum shifts are needed, wiz. the 3 dual relations to WGS84.

All other datums in the implemented systems have one of the four as a parent datum, so that the datum chain at most has three steps: (1) the input datum, (2) its parent datums, and (3) the output datum. The datum shifts operate in 3-d cartesian coordinates, so that any system must be transformed to that before the datum shift takes place, giving the result in 3-d coordinates, which if needed should be transformed to the desired output system.

The datum shifts as presented here or by the differential formulae is only a linear approach. A common datum shift for a whole continent may therefore need local "corrections", i.e. some kind of prediction, but not necessarily by the simple methods outlined in Chapter 7.

9. Implementation

The actual implementation of transformations (since 1988) at the KMS is based upon the programming language C, but it is assumed that almost any language may be used. The earlier implementations used algol, but certain parts have also been programmed in FORTRAN, Pascal, and PL1. Various compilers have been used, for UNIX the ANSI standard versions from SUN, for Linux the GNU C and for DOS, windows etc. the Turbo-C.

The complete system consists of more than 50 functions with a total of more than 20000 lines of code, including dedicated input/output functions. The production and maintenance with UNIX and Linux uses the make-facility supported by awk-code for automatic inclusion of all needed modules.

The main components of the system are:

1. Coordinate descriptors and the input/output modules.

2. The transformation and prediction modules.

3. Service modules for initialization and error reports.

4. The transformation programme.

The underlying idea is to join the transformation modules as an independent function available for programmes requiring transformation and to compose the transformation programme with this same function plus what is needed for input/output and service functions.

9.1 Coordinate System Descriptors

The coordinate systems are described in so-called *coordinate labels*. A coordinate label contains a description of all parameters needed for the handling of the coordinates:

1. The name of the coordinate system in short form (the "minilabel"), e. g. utm32_euref89 for UTM zone 32 coordinates in the EUREF89 system.

2. The internal enumeration of the coordinate system, the ellipsoid, the datum, and the parent datum. (Users and most programmers will refer to the system names and will not have to know the enumeration).

3. Parameter values for the ellipsoid, transformation constants, and datum shift constants, etc.

The size of a coordinate label is about 500 bytes, being defined as a structure in C. The Appendix IV contains a listing of the structure.

The coordinate labels are initialized at run time, e.g. from forerunners of a data stream or by user commands, and an output stream of coordinates is always preceded by a forerunner describing the coordinates. Similar labels exist for other data. The label system comprises:

1. Coordinate labels.
2. Height labels.
3. Geoid labels.
4. Adjustment result labels.
5. Job definition labels.
6. Observation labels.
7. Identity labels.
8. Stop label.

A label is preceded by a # (except item 4 above) for identification. Only item 1 is used for transformation, and all other data types may be skipped if they occur in the data stream for the transformation.

A transformation may thus be defined by a pair of coordinate labels defining the input system and the output system. It is then the task of the transformation system to find a sequence of transformations needed or contingently to reject the transformation as impossible or illegal.

9.2 Internal Transformation Functions

The major parts of the transformations are made by means of three functions:

1. ptg: Mapping Coordinates ⇔ Geodetic Coordinates.
2. gtc: Geodetic Coordinates ⇔ 3-d Cartesian Coordinates.
3. ctc: 3-d Cartesian Coord. ⇔ 3-d Cartesian Coord. with Datum Shift.

The first two has no built-in strategy and will simply transform the input coordinates to the output coordinates, assuming that the coordinates are on the same datum. The functions are dual, meaning that they will transform both ways as controlled by a direction parameter. The ptg is called thus:

```
res = ptg(c_lab, direction, CN_in, CE_in, &CNout, &CEout, text, err_file);
```

where c_lab is a coordinate label for the actual mapping, and the direction parameter is +1 for mapping ⇒ geodetic, and -1 for the opposite. The input coordinates are (CN_in, CE_in), and (CNout, CEout) is the output. The text is a user defined text which will be included in a contingent transformation error report in the err_file. The value of the transformation is 0 for a successful transformation and an error number in case of error. The coordinate units are metres or radians corresponding to the type of the coordinates.

The gtc is similar, but has 3 coordinates both in input and output. A value of direction > 0 gives transformation from geodetic coordinates supplemented with an ellipsoidal height to 3-d cartesian coordinates. A value < 0 gives the reverse transformation. The label need not be a descriptor of a geodetic coordinate system or a 3-d system. A label describing the mapping (on the same datum) suffices, because only the equatorial radius and the flattening are needed. The default action in case of a missing input height is to assume that the point is on the ellipsoid, i.e. the ellipsoidal height is zero. If only orthometric heights or normal heights are available, a conversion to ellipsoidal heights by means of geoid tables is automatically invoked (if possible).

The ctc function has a certain built-in strategy using the datum shift parameters of the coordinate label, which will transform to/from the parent datum defined in the label. The call is:

```
res = ctc(in_lab, outlab, X_in, Y_in, Z_in, &Xout, &Yout, &Zout, text, err_file);
```

The ctc has a strategy and the needed constants for transformation between the parent datums of the input and the output systems. The function can hardly be called dual, but it will nevertheless always reverse the transformation so that the reverse result can be compared with the input.

The mappings handled in this way comprise:

1. Transversal Mercator (UTM, Gauss-Krüger)

2. Co-axial Conformal Mappings (Mercator, Lambert, and Polar stereographic)

3. The Gauss-Krüger stereographic mapping (with an arbitrary central point)

4. Equivalent mappings (Sansom-Flamsteed, Mollweide, Lambert cylindric)

A transformation from a given mapping on a given datum to a wanted mapping on a wanted datum can be performed in this way:

```
res  = ptg(in_lab, +1, ....     /* mapping  ⇒ geodetic  */
res |= gtc(in_lab, +1, ....     /* geodetic ⇒ cartesian */
res |= ctc(in_lab, outlab, ...  /* datum shift          */
res |= gtc(outlab, -1, ....     /* cartesian ⇒ geodetic */
res |= ptg(outlab, -1, ....     /* geodetic ⇒ mapping   */
```

The very simple transformation between two conformal mappings on the same datum (e.g. from one UTM zone to another one) can utilize the fact that the internal transformation between Gaussian coordinates and geodetic coordinates can be skipped. This is signalled by using 2 instead of 1 for the direction parameter.

```
res  = ptg(in_lab, +2  ....     /* mapping  ⇒ Gaussian  */
res |= ptg(outlab, -2, ....     /* Gaussian ⇒  mapping  */
```

The input coordinates are always called by value and the output coordinates are called with reference (therefore the &-operator), so that the same coordinate variable names may be used for input and output in a call.

9.3 Local Subsystems (dk_trans etc.)

The predictions between the non-regular systems are carried out in local subsystems, which are implemented with a state/action table for the selection of legal sequences of transformations and predictions (and alarms for illegal sequences). One specific coordinate system for each subsystem is the gateway to the regular systems, and the subsystem function may also be used as a freestanding transformation system, but they are of course also included in the unitrans function.

The local subsystems comprise:

1. dk_trans
   - (1) utm32_ed50 (Gateway)
   - (2) utm33_ed50
   - (3) geo_ed50
   - (a) s34j = System 1934 Jylland
   - (b) s34s = System 1934 Sjælland
   - (c) gs = General staff conical mapping (Jyll. + Sjæll.)
   - (d) geogs = General staff geographical crd. (J + S)
   - (e) kk = Local system for Copenhagen
   - (f) os = Old cadastral system in Sønderjylland

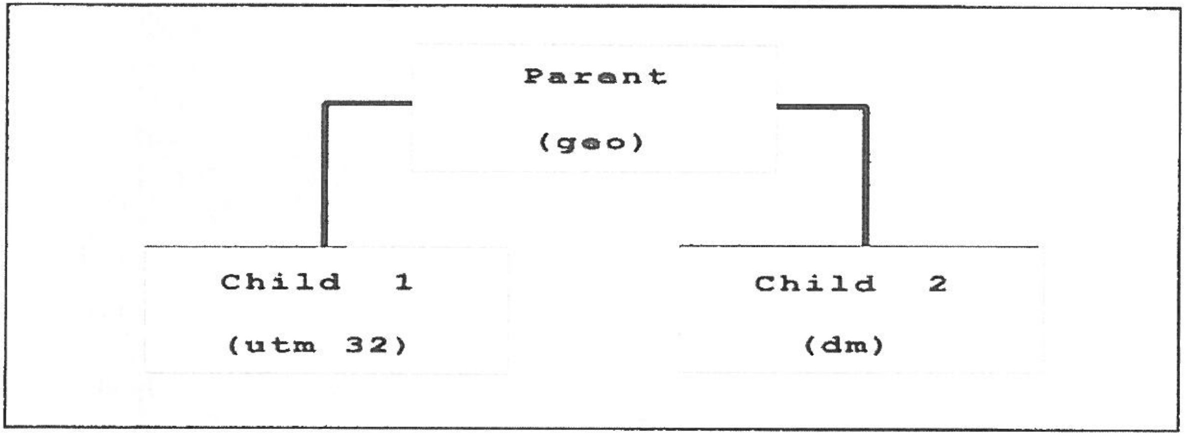
2. bo_trans
   - (1) utm33_ed50
   - (2) utm32_ed50 (Gateway)
   - (3) geo_ed50
   - (a) s45b = System 1945 Bornholm
   - (b) gsb = General staff conical mapping (Bornh.)
   - (c) geogsb = General staff geograph. crd. (Bornh.)

3. fe_trans
   - (1) geo_euref89
   - (2) utm29_euref89 (Gateway)
   - (a) fk54 = Conformal Conical, Færø Datum 1954
   - (b) fu50 = UTM zone 29, European Datum 1950

9.4 The Universal Transformation Function

The individual transformation and/or prediction functions are modules in a common function *unitrans*. This function may be called in the various programmes needing transformations, and of course also in a transformation programme as the *kmstr[x]*, where x is the version mumber, at present 4, going to be replaced by version 5.

Fig. 9.1 Parent and Child Structure



The unitrans and all the subsystems are based upon a tree structure for the transformations. The background for this is that any transformation must have only one common specific path of transformation sequence connecting the two coordinate systems for both directions of transformation. Predictions will generally not have a physical precision better than 2 cm (in spite of an internal precision between direct and reverse prediction better than 1 mm), so if two different paths were used, then coordinates moved around in the transformation system could accumulate the systematic errors in the prediction functions (Poder, 1992).

Fig. 9.1 shows in a very simple example a detail of the tree structure used for two regular systems on the same datum (EUREF89). It is seen that a transformation from UTM zone 32 to a Mercator mapping (dm) is done in two steps:

1. UTM zone 32 ⇒ Geodetic coord.

2. Geodetic coord ⇒ dm.

The input label is here #utm32_euref89 and the output label is #dm_euref89. As each label contains all constants needed for transformation to/from geodetic coordinates; a call for transformation with the input label as a parameter may give geodetic coordinates. A subsequent call with the output label as parameter will produce the dm coordinates.

If the input and the output are on different datums then a datum shift is needed. Therefore 3-d coordinates (X, Y, Z) are computed from the geodetic coordinates and the actual height if available (or a default value of zero).

Unitrans has a table of the regular systems used as the gateway to the subsystems enabling it to find a path leading to the proper prediction function in the subsystem. Unitrans also includes service functions for computing the geoid height (or more precisely the height anomaly), so that normal heights or orthometric heights may be used or computed instead of ellipsoidal heights, which also may be used or computed. If no height is given but nevertheless needed then an input ellipsoidal height of zero is used as default. The output will then be coordinates of a point which in general has a finite height over the ellipsoid of the output, but the effect on the horizontal coordinates (latitude and longitude) is generally less than 1 mm.

The unitrans is thus the general purpose transformation function, which may be used both in a general transformation programme and also may be used in programmes for network adjustment, data base queries, plotting programmes, etc. The unitrans requires initialized coordinates labels describing the input and output. All other internal tables are initialized and maintained by unitrans itself.

9.5 The Transformation Programme

The transformation programme is based upon the unitrans function. It includes input/output functions for conversion between the text form of the station numbers and coordinates and their internal representation. The list of input data must be preceded by a coordinate label describing the input coordinate system. The desired output system is defined by its coordinate label given as an option in the programme call. Other options include (1) a report of the input coordinates (placed in "reading brackets"), (2) geoid heights, (3) names of non-standard geoids, (4) number of decimals (other than the default), etc. Transformation errors are reported in a special file, and option errors or omissions will release some short examples of how to use the programme, so a call without options will be a short introduction for the user.

Appendix I. Clenshaw Summation

I.1 Recurrence Relations

Clenshaw summation is used for the computation of the trigonometric series expansions of the Gaussian latitude and of the transversal conformal mappings. Clenshaw summation is based upon the existence of a recurrence relation of the functions occurring in series expansions.

(I.1) Recurrence Relations

$$\textit{Series expansion:}\ \sum_{k=K}^{\infty} c_k\, f_k(z)$$

$$\textit{Recurr. relation:}\ a_n f_n(z) + a_{n-1} f_{n-1}(z) + \ldots. = 0$$

Note that the recurrence relation coefficients a_n has no relation to the series expansion coefficients c_k in form or quantity.

The sine function has the recurrence relation

(I.2) Real Recurrence Relations

$$\sin(n+1)z - 2\cos z\ \sin nz + \sin(n-1)z = 0$$

which also is valid for a complex argument z.

(I.3) Complex Recurrence Relations

$$\sin((2k+2)(y+ix)) - 2\cos(2(y+ix))\sin((2k(y+ix)) + \sin((2k-2)(y+ix))$$

$$= R(k+1) + i\,I(k+1) - (R_0 + i\,I_0)(R(k) + i\,I(k)) + R(k-1) + i\,I(k-1)$$

$$= \quad 0$$

$$R(k) = \sin(2ky)\ \cosh(2kx) \qquad I(k) = \cos(2ky)\ \sinh(2kx)$$

$$R_0 = +2\cos 2y\ \cosh 2x \qquad I_0 = -2\sin 2y\ \sinh 2x$$

The treatment of the complex numbers requires a simple extension of the recursion matrix and the solution vector. The real part and the imaginary one must be collected at each step of their summation.

I.2 Geodetic Latitude ⟺ Gaussian Latitude

The summation is carried out by solving a simple set of equations with the coefficients of the series as right-hand sides and a coefficient matrix with the recurrence relation coefficients placed strategically in the columns.

(I.4) The Clenshaw Equations

$$\sum_{\kappa=1}^{4} e_{2\kappa}\sin 2\kappa\phi = S'E = S'(TQ) = (S'T)Q \qquad (\,\textit{using } E = TQ\,)$$

$$S = \begin{pmatrix} \sin 8\phi \\ \sin 6\phi \\ \sin 4\phi \\ \sin 2\phi \end{pmatrix} \quad E = \begin{pmatrix} e_8 \\ e_6 \\ e_4 \\ e_2 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2\cos 2\phi & 1 & 0 & 0 \\ 1 & -2\cos 2\phi & 1 & 0 \\ 0 & 1 & -2\cos 2\phi & 1 \end{pmatrix} \quad Q = \begin{pmatrix} q_8 \\ q_6 \\ q_4 \\ q_2 \end{pmatrix}$$

The product vector $S'T$ has only the last element different from zero, so the whole sum is found very simply.

The matrix T is a lower triangular matrix with columns containing the recurrence relation coefficients, always starting with the diagonal element. The determinant is therefore non-zero, and the solution vector Q is found by the artifice of the two additional zero-valued q's.

(I.5) The Clenshaw Sine Solutions

$$\begin{aligned} S'T &= (0,\ 0,\ \sin 4\phi\ -\ 2\cos 2\phi\ \sin 2\phi,\ \sin 2\phi) \\ &= (0,\ 0,\ 0,\ \sin 2\phi) \\[6pt] S'TQ &= q_2\ \sin 2\phi \end{aligned}$$

A piece of C-code shows the simple algorithm for the summation, requiring only the sine and cosine of 2ϕ.

(I.6) The Clenshaw Sine Algorithm

```
Initialize:   $q_{10} = q_{12} = 0$
Loop:
for ($\kappa$ = 4; $\kappa$ > 0; $\kappa$--) {
    $q_{2\kappa} = e_{2\kappa} + 2\cos(2\phi)\, q_{2\kappa+2} - q_{2\kappa+4}$
}
Result:   $q_2 \sin 2\phi$
```

The formulae for the transformation geodetic latitude <=> Gaussian latitude then is:

(I.7) Geodetic Latitude ⇒ Gaussian Latitude

$$\Phi_c = \phi_c + \sum_{\kappa=1}^{\infty} e_{2\kappa} \sin 2\kappa\phi_c$$

$$\begin{array}{llll}
e_2 = -2n + \frac{2}{3}n^2 + \frac{4}{3}n^3 - \frac{82}{45}n^4 & & e_4 = +\frac{5}{3}n^2 - \frac{16}{15}n^3 - \frac{13}{9}n^4 \\
e_6 = \qquad\qquad -\frac{26}{15}n^3 + \frac{34}{21}n^4 & & e_8 = \qquad\qquad +\frac{1237}{630}n^4
\end{array}$$

(I.8) Gaussian latitude ⇒ Geodetic Latitude

$$\phi_c = \Phi_c + \sum_{\kappa=1}^{\infty} G_{2\kappa} \sin 2\kappa\Phi_c$$

$$\begin{array}{llll}
G_2 = +2n - \frac{2}{3}n^2 - 2n^3 + \frac{116}{45}n^4 & & G_4 = +\frac{7}{3}n^2 - \frac{8}{5}n^3 - \frac{227}{45}n^4 \\
G_6 = \qquad\qquad +\frac{56}{15}n^3 - \frac{136}{35}n^4 & & G_8 = \qquad\qquad +\frac{4279}{630}n^4
\end{array}$$

I.3 Complex Gaussian Coordinates <=> Transversal Coordinates

The treatment of the complex numbers requires a simple extension of the recursion matrix and the solution vector. The real part and the imaginary one must be collected at each step of their summation.

(I.9) Clenshaw Complex Sine Equations

$$2\cos(2(y+ix)) = R_0 + iI_0; \quad R_0 = +2\cos(2y)\cosh(2x); \quad I_0 = -2\sin(2y)\sinh(2x)$$

$$\sum_{\kappa=1}^{4} u_{2\kappa} \sin 2\kappa(y + ix) = S^t C = S^t(TQ) = (S^t T)Q \qquad (\,using\ C = TQ\,)$$

$$S = \begin{pmatrix} \sin 8(y+ix) \\ \sin 6(y+ix) \\ \sin 4(y+ix) \\ \sin 2(y+ix) \end{pmatrix} \quad C = \begin{pmatrix} C_8 \\ C_6 \\ C_4 \\ C_2 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -(R_0+iI_0) & 1 & 0 & 0 \\ 1 & -(R_0+iI_0) & 1 & 0 \\ 0 & 1 & -(R_0+iI_0) & 1 \end{pmatrix} \quad Q = \begin{pmatrix} p_8+iq_8 \\ p_6+iq_6 \\ p_4+iq_4 \\ p_2+iq_2 \end{pmatrix}$$

(I.10) Clenshaw Complex Solutions

$$S^t T = \{\, 0,\ 0,\ \sin(4(y+ix)) - 2\cos(2(y+ix))\sin(2(y+ix)),\ \sin(2(y+ix)) \,\}$$
$$= \{\, 0,\ 0,\ 0,\ \sin(2(y+ix)) \,\}$$

Using the names in the recursion relation, one gets the algorithm as follows:

(I.11) The Clenshaw Complex Algorithm

```
Initialize: $p_{10} = p_{12} = q_{10} = q_{12} = 0$
            $R_0 = +2\cos(2y)\cosh(2x);$        $I_0 = -2\sin(2y)\sinh(2x)$
            $R(1) = \sin(2y)\cosh(2x);$        $I(1) = \cos(2y)\sinh(2x)$
Coefficients: $C_8, C_6, C_4, C_2;$      (All real)

Loop:
    for ($\kappa = 4;\ \kappa > 0;\ \kappa--$) {
        $p_{2\kappa} = C_{2\kappa} + R_0 p_{2\kappa+2} - I_0 q_{2\kappa+2} - p_{2\kappa+4}$
        $q_{2\kappa} = R_0 q_{2\kappa+2} + I_0 p_{2\kappa+2} - q_{2\kappa+4}$
    }

Result:
    Realpart:  $R(1)\,p_2 - I(1)\,q_2$
    Imagpart:  $R(1)\,q_2 + I(1)\,p_2$
```

(I.12) Complex Gaussian Coordinates ⇒ Normalized Transversal Coordinates

$$u = U + \sum_{\kappa=1}^{\infty} U_{2\kappa} \sin 2\kappa U$$

$$U_2 = +\frac{1}{2}n - \frac{2}{3}n^2 + \frac{5}{16}n^3 + \frac{41}{180}n^4 \qquad U_4 = +\frac{13}{48}n^2 - \frac{3}{5}n^3 + \frac{557}{1440}n^4$$

$$U_6 = +\frac{61}{240}n^3 - \frac{103}{140}n^4 \qquad U_8 = +\frac{49561}{161280}n^4$$

(I.13) Normalized Transversal Coordinates ⇒ Complex Gaussian Coordinates

$$U = u + \sum_{\kappa=1}^{\infty} u_{2\kappa} \sin 2\kappa u$$

$$u_2 = -\frac{1}{2}n + \frac{2}{3}n^2 - \frac{37}{96}n^3 + \frac{1}{360}n^4 \qquad u_4 \doteq -\frac{1}{48}n^2 - \frac{1}{15}n^3 + \frac{437}{1440}n^4$$

$$u_6 = -\frac{17}{480}n^3 + \frac{37}{840}n^4 \qquad u_8 = -\frac{4397}{161280}n^4$$

Appendix II: Geodetic Parameters for Mappings

Mapping needs only two parameters giving the size and the shape of the biaxial ellipsoid. The more recent ellipsoids are rather a complete reference system giving four necessary and sufficient parameters. The equatorial radius and the flattening as given for the Hayford 1924 Ellipsoid (actually published by Hayford much earlier) were so far a convenient standard. The two supplementary physical parameters were added later.

The definition for GRS80 (and GRS67) no longer gives the flattening directly, but is a consistent system. The claimed system for WGS84 is almost GRS80, but the dynamic form factor J_2 was derived with too few digits from a coefficient in a series expansion for the gravity field. The effect is negligible (less than 0.1 mm for the meridian quadrant), but should of course have been avoided.

The constants of the ellipsoid given by Bessel in 1841 were certainly not uniquely defined by Bessel, who gave redundant parameters, with a reasonable consistency at that time (about 8 decimal digits). The dimension was given indirectly as the average length (in Toises) of 1 degree of latitude and the shape could be found in several ways from the published results. It is therefore very understandable that the values for Bessels ellipsoid are different, but eventually a reasonable consensus seems to exist for the equatorial radius and the flattening. It is not likely that Bessels ellipsoid will be used for Physical Geodesy, but for completeness the *GM*-values also used in GRS80 could be used, because the mass of the earth is assumed to be almost constant.

The rotation velocity of the earth is needed for completeness, but the value is known much better that strictly needed.

(II.1) Examples of Ellipsoids

	Geodetic Reference System (GRS80)		
a	$= 6\ 378\ 137$	m	; *equatorial radius*
J_2	$= 108\ 263 \times 10^{-8}$		; *dynamic form factor*
GM	$= 3\ 986\ 005 \times 10^{8}$	$m^3\ s^{-2}$	; *geoc. gravt. const.*
ω	$= 7\ 292\ 115 \times 10^{-11}$	$rad\ s^{-1}$	; *angular velocity*
(f	$= 1\ /\ 298.257\ 222\ 100\ 883$		; *derived flattening*)
	Hayford 1924 Ellipsoid (International 1924)		
a	$= 6\ 378\ 388$	m	; *equatorial radius*
f	$= 1\ /\ 297$		; *flattening*
γ_a	$= 978.049\ 000$	gal	; *equatorial gravity*
ω	$= 7\ 292\ 115 \times 10^{-11}$	$rad\ s^{-1}$	; *angular velocity*
	Bessel 1841 Ellipsoid, (Conventional and added constants)		
a	$= 6\ 377\ 397.1550$	m	; *equatorial radius*
f	$= 1\ /\ 299.1528128$		; *flattening*
GM	$= 3\ 986\ 005 \times 10^{8}$	$m^3\ s^{-2}$	; *geoc. gravt. const.*
ω	$= 7\ 292\ 115 \times 10^{-11}$	$rad\ s^{-1}$	; *angular velocity*

The convenient initial parameters for mapping are the equatorial radius a and the flattening f, although the polar curvature radius $c = a/(1-f)$, the third flattening n, and the square of the second eccentricity e'^2 frequently give more convenient formulae. However, the recent definitions of reference ellipsoids no longer give the flattening as a direct defining parameter. The rational way out of this problem is for any desired ellipsoid to start with its original definition and derive the desired parameters from that definition.

(II.2) Determination of the Flattening from J_2

GEODESIST'S HANDBOOK on GRS80:

$$J_2 = \frac{e^2}{3}\left(1 - \frac{2}{15}\,\frac{m\,e'}{q_0}\right)$$

$$m = \frac{\omega^2\,a^2\,b}{GM}$$

$$2q_0 = \sum_{\kappa=1}^{\infty} \frac{4\,(-1)^{\kappa+1}\,\kappa}{(2\kappa+1)\,(2\kappa+3)}\,e'^{2\kappa+1}$$

$$ae = be' \quad ; \quad b = a\,(1-f) \quad ; \quad e^2 = f\,(2-f)$$

ALGORITHM:

Initialize: $m_a = \dfrac{\omega^2 a^3}{GM}$, $f = e'^2 = 0$, $q_e = \frac{2}{15}$; $(q_e = \dfrac{q_0}{e'^3})$

Iteration:

$$f = \tfrac{3}{2}J_2 + \tfrac{1}{2}f^2 + \tfrac{1}{15}m_a\,(1-f)^3/q_e$$

$$e'^2 = \frac{f\,(2-f)}{(1-f)^2}$$

$$q_e = \sum_{\kappa=0}^{8} \frac{2\kappa+2}{(2\kappa+3)\,(2\kappa+5)}\,\left(-(e'^2)\right)^{\kappa}$$

The computation of the eccentricity e for the GRS80 ellipsoid is shown in the Geodesists Handbook, (I.A.G. 1980, 1984, 1988, or 1992). The iteration algorithm shown gives directly the flattening f and the second eccentricity e'.

Appendix III: Ellipsoid Formulae and Parameters

Some formulae for the ellipsoid are included here for orientation. They can of course also be found in any geodetic textbook.

Three much used ellipsoids and an algorithm for finding the flattening f from J_2 are found in appendix II.

(III.1) Ellipsoid Formulae and Parameters

$$a = \textit{Equatorial radius} \qquad ; \textit{Defining ellipsoid parameter for mapping}$$

$$f = \textit{Flattening} \qquad ; \textit{Defining ellipsoid parameter for mapping}$$

$$c = a/(1-f) = a\frac{1+n}{1-n} \qquad ; \textit{Polar curvature radius}$$

$$b = a\,(1-f) = a\frac{1-n}{1+n} \qquad ; \textit{Minor semi-axis}$$

$$\frac{a+b}{2} = \frac{a}{1+n} \qquad ; \textit{For K\&W-fans}$$

$$n = \frac{a-b}{a+b} = f/(2-f) \qquad ; \textit{Third flattening}$$

$$e^2 = \frac{a^2-b^2}{a^2} = f(2-f) = \frac{4n}{(1+n)^2}$$

$$e'^2 = \frac{a^2-b^2}{b^2} = \frac{f(2-f)}{(1-f)^2} = \frac{4n}{(1-n)^2}$$

$$W = \sqrt{1-e^2\sin^2\phi} = F^{1/2}/(1+n)$$

$$V = \sqrt{1+e'^2\cos^2\phi} = F^{1/2}/(1-n)$$

$$F = 1 + 2n\cos2\phi + n^2 = (1-n)^2V^2 = (1+n)^2W^2$$

$$= 1 + n(z^2+z^{-2}) + n^2 = (1+nz^2)(1+nz^{-2})$$

$$M(\phi) = c/V^3 = a\,(1-n)^2\,(1+n)\,F^{-3/2} \qquad ; \textit{Meridian curvature radius}$$

$$N(\phi) = c/V = a\,(1+n)\;F^{-1/2} \qquad ; \textit{Prime vertical curvature radius}$$

$$r(\phi) = N(\phi)\cos\phi \qquad ; \textit{Latitude parallel radius}$$

$$Q = \frac{a}{1+n}\left(1 + \frac{1}{4}n^2 + \frac{1}{64}n^4\right) \qquad ; \textit{Meridian arc unit}$$

$$G(\phi) = \textit{The meridian arc from equator to } \phi \qquad ; \textit{see } (III,4)$$

$$\phi(G) = \textit{The latitude for a meridian arc of } G \qquad ; \textit{see } (III,4)$$

(III.2) Series Expansion of Powers of F(z)

$$
\begin{aligned}
z &= \exp(i\,\phi)\\
F^{\alpha}(z) &= (1 + n z^{2})^{\alpha}\,(1 + n z^{-2})^{\alpha} = c_0^{(\alpha)} + \sum_{\kappa=1}^{\infty} c_{2\kappa}^{(\alpha)}(z^{2k} + z^{-2\kappa})\\
c_0^{(\alpha)} &= 1 + \alpha^2 n^2 + \frac{\alpha^2(\alpha-1)^2}{4} n^4 + \dots\\
c_2^{(\alpha)} &= \alpha n + \frac{\alpha^2(\alpha-1)}{2} n^3 + \dots\\
c_4^{(\alpha)} &= \frac{\alpha(\alpha-1)}{2} n^2 + \frac{\alpha^2(\alpha-1)(\alpha-2)}{6} n^4 + \dots\\
c_6^{(\alpha)} &= \frac{\alpha(\alpha-1)(\alpha-2)}{6} n^3 + \dots\\
c_8^{(\alpha)} &= \frac{\alpha(\alpha-1)(\alpha-2)(\alpha-3)}{24} n^4 + \dots
\end{aligned}
$$

Series expansion of F^{α} are very useful tools. They are used here to give formulae for $M(\phi)$ and F^{-1}. The formulae are taken from K & W, where the linear quantity mostly was half the sum of the major semi axis and the minor semi axis, which gave all the relations a uniform appearance. We shall here prefer the expression $a/(1+n)$ instead.

(III.3) Meridian Curvature Radius and 1/F

$$
\begin{aligned}
F^{-3/2}(z) &= (1 + nz^{2})^{-3/2}\,(1 + nz^{-2})^{-3/2}\\
M(\phi_c) &= \frac{a}{1+n}(1 - n^{2})^{2} F^{-3/2}(z) = \frac{a}{1+n}\left(M_0 + \sum_{\kappa=1}^{4} M_{2\kappa}\frac{z^{2\kappa} + z^{-2\kappa}}{2}\right)\\
&= \frac{a}{1+n}\left(M_0 + \sum_{\kappa=1}^{4} M_{2\kappa}\cos(2\kappa\phi)\right) \qquad . \qquad (A)\\
F^{-1}(z) &= (1 + nz^{2})^{-1}\;(1 + nz^{-2})^{-1} = n_0 + \sum_{\kappa=1}^{\infty} n_{2\kappa}(z^{2\kappa} + z^{-2\kappa})
\end{aligned}
$$

$$
\begin{aligned}
M_0 &= 1 + \tfrac{1}{4}n^{2} + \tfrac{1}{64}n^{4} & n_0 &= +1/(1 - n^{2})\\
M_2 &= -3n + \tfrac{3}{8}n^{3} & n_2 &= -n/(1 - n^{2})\\
M_4 &= \tfrac{15}{4}n^{2} - \tfrac{15}{16}n^{4} & n_4 &= +n^{2}/(1 - n^{2})\\
M_6 &= -\tfrac{35}{8}n^{3} & n_6 &= -n^{3}/(1 - n^{2})\\
M_8 &= \tfrac{315}{64}n^{4} & n_8 &= +n^{4}/(1 - n^{2})
\end{aligned}
$$

The formulae for the meridian arc length as a function of the latitude and the inverse problem are derived from the formulae for the curvature radius of the meridian.

(III.4) The Meridian Arc and its Inverse

The scaled meridian arc length unit:

$$Q_m = m_0 \frac{a}{1+n} M_0 = m_0 \frac{a}{1+n} \left(1 + \frac{1}{4}n^2 + \frac{1}{64}n^4\right) \qquad (A)$$

$$m_0 = 1.0 \qquad \textit{(Ellipsoid and Gauss-Krüger)}$$

$$m_0 = 0.9996 \qquad \textit{(UTM)}$$

The scaled meridian arc from equator to the latitude ϕ:

$$G(\phi) = Q_m \left(\phi + \sum_{\kappa=1}^{4} p_{2\kappa} \sin(2\kappa\phi) \right) \quad . \qquad (B)$$

$$p_2 = -\frac{3}{2}n + \frac{9}{16}n^3 \qquad p_4 = +\frac{15}{16}n^2 - \frac{15}{32}n^4$$

$$p_6 = -\frac{35}{48}n^3 \qquad p_8 = +\frac{315}{512}n^4$$

The latitude at the normalized arc length $A = G/Q_m$

$$\phi(A) = A + \sum_{\kappa=1}^{4} q_{2\kappa} \sin(2\kappa A) \quad . \qquad (C)$$

$$q_2 = +\frac{3}{2}n - \frac{27}{32}n^3 \qquad q_4 = +\frac{21}{16}n^2 - \frac{55}{32}n^4$$

$$q_6 = +\frac{151}{96}n^3 \qquad q_8 = +\frac{1097}{512}n^4$$

The two series expansions (III.4.B) and (III.4.C) are in principle the formulae for the transversal mapping as given in Chapter 2 and elaborated in Chapter 4. The formulae may be used both for the ellipsoid and for the transversal mappings by proper choice of m_0.

Appendix IV. The Coordinate Label

A coordinate label contains the parameters for the description of a coordinate system and for the transformation to other coordinate systems.

IV.1 A Coordinate Label

```
struct  crd_lab {
  short      lab_type;     /* Type of label */
  short      version;      /* Label version */
  char       mlb[16];      /* Minilabel of the system */
  short      sepch;        /* Separator-char in label */
  short      compl;        /* geo_lab definitions compl.*/
  short      cstm;         /* Coordinate system */
  short      mode;         /* Coordinate system mode */
  short      region;       /* Region, see conv_lab.h */
  short      ncoord;       /* Number of coordinates */
  short      S_lat;        /* Lat. sgn pos N: 0, pos S: 1 */
  short      S_crd;        /* Crd. sgn pos N: 0, pos S: 1 */
  short      W_lng;        /* Lng. sgn pos E: 0, pos W: 1 */
  short      W_crd;        /* Crd. sgn pos E: 0, pos W: 1 */
  short      p_seq;        /* seq. of crd, 0=>N,E, 1=>E,N */
  short      ellipsoid;    /* Ellipsoid */
  short      datum;        /* Datum */
  short      p_dtm;        /* Parent datum */
  double     a;            /* Semi major axis of ellipsoid */
  double     f;            /* Flattening of ellipsoid */
  double     B0;           /* Origin latitude */
  double     N0;           /* Origin northing */
  double     L0;           /* Origin longitude */
  double     E0;           /* Origin easting */
  double     scale;        /* Central scale of the mapping */
  double     B1;           /* Contact lat (lmb), P0 (stg) */
  double     B2;           /* Lat. of intersect. for lmb */
  double     tol;          /* Tolerance of check transf. */
  short      zone;         /* utm zone no.                */
  short      imit;         /* mask for non-reg sys. */
  short      init;         /* init tr. const. 0=>init. */
  double     Qn;           /* Merid. quad., scaled */
  double     Zb;           /* Radius vector in p. crd. sys */
  double     cP;           /* Lambert exponent = cos P0 */
  double     tcgg[8];      /* Const. Gauss <-> Geo lat.   */
  double     utg[4];       /* Const. transv. merc. -> geo */
  double     gtu[4];       /* Const. geo -> transv. merc. */
  struct dsh_str dsh_con;/* Struct of datum shift const.*/
  long       ch_sum;       /* Checksum of label */
};
```

References and Literature

Andersen, Einar, Torben Krarup and Bjarner Svejgaard: Geodetic Tables, International Ellipsoid. Geodætisk Instituts Skrifter, 3. Række Bind XXIV. København 1956.

Ashkenazi, V., R. Bingley, B. Codd, and M. Cory: Results and Analysis of the EUREF EIR/GB 95 GPS Campaign. (In Report on the Symposion of the EUREF in Ankara 1996. Ed. by Erich Gubler and Helmut Hornik. Bayerischen Komm. f. Int. Erdmess. München 1996).

Bomford, G.: Geodesy, second edition, Oxford 1962.

Borre, Kai: Landmåling. Institut for Samfundsudvikling og Planlægning. Aalborg 1990.

Boucher, C., P. Willis, and Z. Altamimi: Contribution of Space Data to RETrig and EUREF. (Report on the IAG RETrig Subcommission Symposium in Lisbon May 1988, ed. by Poder and Hornik, RETrig Publication No. 18, München. 1989).

Boucher, C. and Z. Altamimi: The EUREF Terrestrial Reference System and its First Realizations. (In Report of the RETRIG Symposium in Florence May 1990 of the EUREF subcommission, edited by Gubler, Poder and Hornik). Bayerische Kommission für die Internationale Erdmessung. A.-G. Arb. Heft 52. München 1992.

Boucher, C. and Z. Altamimi: Specifications for Reference Frame Fixing in the Analysis of a EUREF GPS Campaign. (In Report on the Symposium of the EUREF subcommission in Helsinki May 1995 ed. by Erich Gubler and Helmut Hornik). Veröff. der Bayerischen Komm. für die Internationale Erdmessung der Bayerischen AdW. A.-G. Arbeiten, Heft Nr. 56. München 1995.

Bugayevskiy, Lev M. and John P. Snyder: Map Projections. A Reference Manual. Taylor and Francis, London, 1995.

Dinter, G., M. Illner and R. Jäger: A Synergetic Approach for the Transformation of Ellipsoidal Heights into a Standard Height Reference System (HRS). (In Report on the Symposion of the EUREF in Ankara 1996. Ed. by Erich Gubler and Helmut Hornik. Bayerischen Komm. f. Int. Erdmess. München 1996).

Ehlert, Dieter and Robert Strauss: Die Diagnoseausgleichung 1980 des Deutschen Haupdreiecknetzes, vol. V, DGK Reihe B. Heft Nr. 272, Teil I, IFAG, Frankfurt am Main, 1990.

Engsager, Karsten: Integration of Satellite Data in Local geodetic Networks, Implementations and Numerical Analysis. Publications 4. series, volume 5, National Survey and Cadastre, Denmark 1997.

Grossmann, Walter: Geodätische Rechnungen und Abbildungen in der Landesvermessung, Konrad Wittwer, Stuttgart, 1964.

Heiskanen, Weikko A. and Helmut Moritz: Physical Geodesy. W.H. Freeman and Company, 1967.

Kernighan, Brian W. and Dennis M. Ritchie: The C Programming Language, Second Edition Prentice-Hall, 1988. (First edition 1978).

Krüger, L.: Konforme Abbildung des Erdellipsoids in der Ebene. Veröffentlichung des Königlich Preuszischen Geodätischen Institutes. Neue Folge No. 52, Potsdam, 1912.

König, R. and K.H. Weise: Mathematische Grundlagen der Höheren Geodäsie und Kartographie, Erster Band, Springer, Berlin/Göttingen/Heidelberg, 1951.

Lee, L.P.: Conformal Projections Based on Elliptic Functions, Monograph No. 16/1976, Supplement NO. 1 to Canadian Cartographer, Vol. 13, 1976, CARTOGRAPHIA, B.V. Gutsell, Dep. of Geography, York University, Toronto, Canada, 1976.

Moritz, H.: The Definition of a Geodetic Datum. (In Proc. of the Second International Symposium for the North American Geodetic Networks, ed. by John G. Gergen). U.S. Dep. of Commerce, NOAA, NOS. 1978.

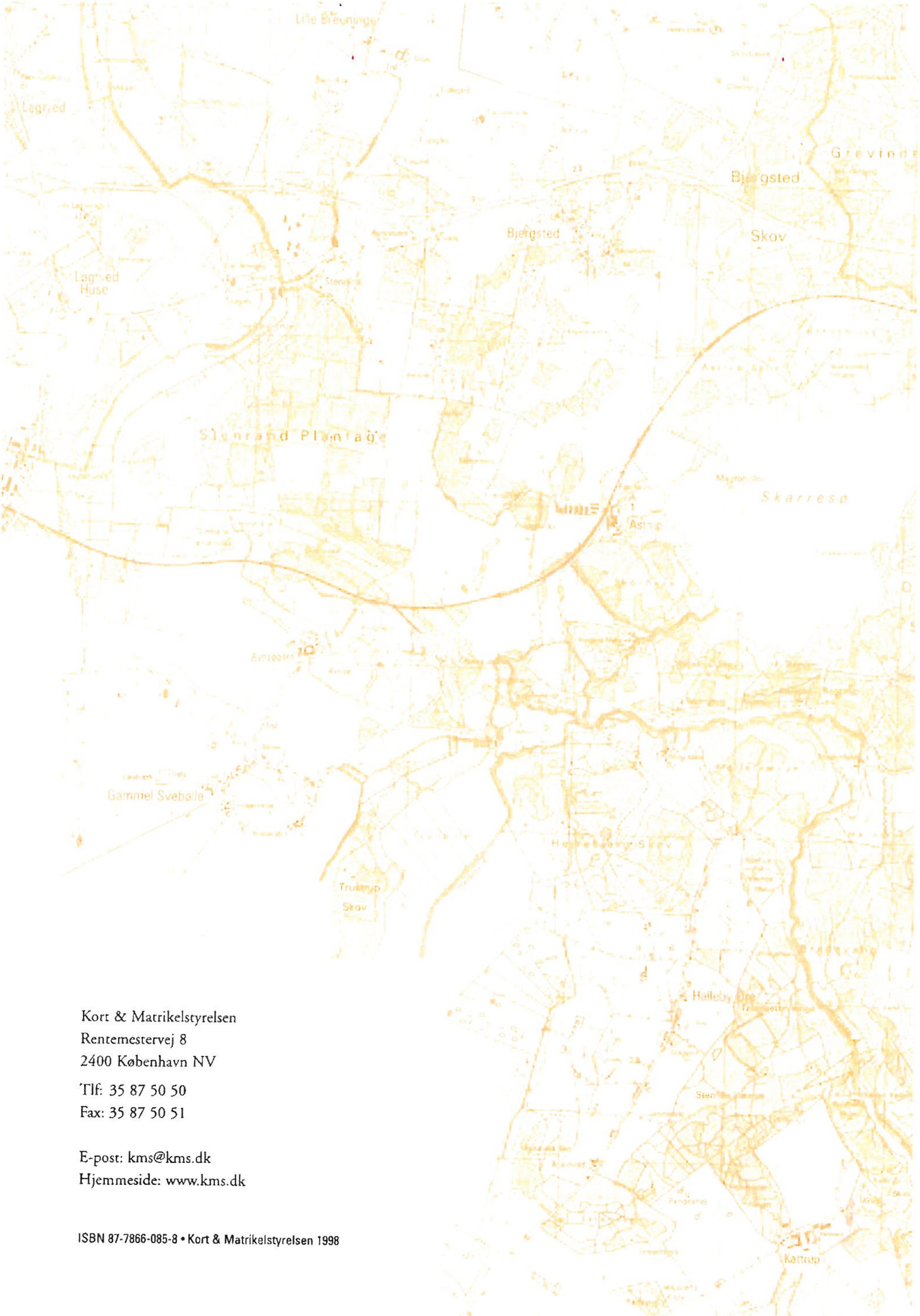
Moritz, H.: Geodetic Reference System 1989. The Geodesist's Handbook 1992, ed. by C.C. Tscherning. Bull. Geod. Vol. 66 no. 2, 1992.

Poder, Knud: Implementation of Coordinate Transformations, NKG Nordiska Forskarkurser, in: Modern Techniques in Geodesy and Surveying, ed. by Ole Bedsted Andersen, Kort- og Matrikelstyrelsen, Publications 4. Series vol 1. Copenhagen 1989.

Poder, Knud: A Strategy for Coordinate Transformations and Predictions. (In Report of the symposium in Florence 1990 of the EUREF subcommission, edited by Gubler, Poder and Hornik). Bayerische Kommission für die Internationale Erdmessung. A.-G. Arb. Heft 52 München 1992.

Richardus, Peter and Ron K. Adler: Map Projections, North Holland, Amsterdam, 1972

Teunissen, P.J.G: The Geometry of Geodetic Inverse Mapping and Non-linear Adjustment. Netherlands Geodetic Commission. Publications on Geodesy. New Series 8: 1. 1985.



Kort & Matrikelstyrelsen
Rentemestervej 8
2400 København NV

Tlf: 35 87 50 50
Fax: 35 87 50 51

E-post: kms@kms.dk
Hjemmeside: www.kms.dk

ISBN 87-7866-085-8 • Kort & Matrikelstyrelsen 1998