ABSTRACT

Clenshaw's recurrence formula (with an associated sum) is an efficient way to evaluate a sum of coefficients multiplied by functions that obey a recurrence formula. It has been used extensively in physical geodesy in the evaluation of sums of high degree and order spherical harmonic series approximating the earth's gravitational potential. But less well known, and the subject of this paper, are applications in map projections where Clenshaw's method is used to develop compact formula for meridian distance and for computation of coordinates, grid convergence and point scale factor on the Transverse Mercator (TM) projection using the Karney-Krüger equations.

INTRODUCTION

Clenshaw’s recurrence formula, with an associated sum (Clenshaw 1955) is an efficient way to evaluate a sum of coefficients multiplied by functions that obey a recurrence formula. As an example consider the formula for meridian distance \( M \) on an ellipsoid of revolution (Deakin et al. 2010)

\[
M = \frac{a}{1+n} \left\{ c_0 \phi + c_2 \sin 2 \phi + c_4 \sin 4 \phi + \cdots + c_{2N} \sin 2N \phi \right\} = \frac{a}{1+n} \left\{ c_0 \phi + \sum_{k=1}^{N} c_{2k} \sin 2k \phi \right\}
\]

(1)

\( a \) and \( n \) are geometric constants of the ellipsoid, \( \phi \) is latitude and \( c_0, c_2, c_4, \ldots \) are known coefficients. \( k \) is an integer and \( N \) is the upper terminal of the summation and the function \( \sin 2k \phi \) satisfies the three-term recurrence formula

\[
\sin 2k \phi = 2 \cos 2 \phi \sin (k-1) \phi - \sin 2(k-2) \phi
\]

(2)
To compute the sum \( S = \sum_{k=1}^{N} c_{2k} \sin 2k\phi \) on the right-hand-side of (1) by usual means requires \( N \) evaluations of the trigonometric sine function, \( 2N \) multiplications and \( N \) additions.

As we will show later, using Clenshaw's recurrence formula:

\[
y_k = \begin{cases} 
0, & \text{for } k > N \\
2 \cos 2\phi y_{k+1} - y_{k+2} + c_{2k}, & \text{for } k = N, N-1, N-2, \ldots, 3, 2, 1
\end{cases}
\]

the sum \( S = y_1 \sin 2\phi \) where \( y_1 \) is obtained from the ‘reverse’ recurrence formula above and the computation of \( S \) requires only two evaluations of trigonometric functions, \( N + 4 \) multiplications and \( 2N \) additions. This is a significant reduction in multiplications and evaluations of trigonometric functions, both of which are expensive in computer CPU time, and Clenshaw’s technique will be more efficient. In addition, since any computer evaluations of trigonometric functions contain very small errors, this method (with a reduced number of trigonometric evaluations) will likely be more accurate.

Clenshaw’s recurrence formula (with the associated sum) is used extensively in physical geodesy, for example in the evaluation of finite sums of high degree and order spherical harmonic series expansions that approximate the earth’s external gravitational potential \( V \) (Tscherning & Poder 1981)

\[
V(r, \psi, \lambda) \approx \sum_{n=0}^{N} \frac{GM}{r} \left(\frac{a}{r}\right)^{n} \sum_{m=0}^{n} P_{n}^{m}(t) \left\{ C_{n}^{m} \cos m\lambda + S_{n}^{m} \sin m\lambda \right\}
\]

where \((r, \psi, \lambda)\) are polar coordinates (\(r\) geocentric radial distance, \(\psi\) geocentric latitude, \(\lambda\) longitude), \(t = \sin \psi\), \(GM\) is the product of the earth’s gravitational constant \(G\) and mass \(M\), \(a\) is the semi-major axis of the reference ellipsoid, \(n\) and \(m\) are positive integers or zero, \(C_{n}^{m}, S_{n}^{m}\) are geopotential coefficients of \(n\)th order and \(m\)th degree and where (i) \(C_{0}^{0} = 1\) enforces the conditions that the first term of the series \((n = 0)\) is \(GM/r\), the mean value of the spherical harmonic series; and (ii) \(C_{1}^{0} = C_{1}^{1} = S_{1}^{1} = 0\) ensures the coordinate origin is at the earth’s centre of mass. \(P_{n}^{m}(t)\) are Associated Legendre Functions and \(N\) is the maximum degree and order of the available coefficients.

The Associated Legendre Functions \(P_{n}^{m}(t)\) are orthogonal polynomials and can be evaluated from the recurrence formula (Gleeson 1985)

\[
P_{n}^{m}(t) = \frac{2n-1}{n-m} t P_{n-1}^{m}(t) - \frac{n+m-1}{n-m} P_{n-2}^{m}(t)
\]

\[
P_{n}^{m}(t) = (2m-1) u P_{n-1}^{m-1}(t)
\]
where by definition \( P_n^m(t) = 0 \) if \( n < 0 \) or \( m > n \), \( P_n^0(t) = P_n(t), \quad u = \sqrt{1 - t^2} = \cos \psi \) and the special values \( P_0^0(t) = 1, \quad P_1^0(t) = t, \quad P_1^1(t) = u \) act as initial values in (4) and (5).

Another form of the potential \( V \) can be obtained from (3) by letting \( q = a/r \) and re-arranging the finite double summation as (Tscherning, Rapp & Goad 1983)

\[
V(r, \psi, \lambda) \approx \frac{GM}{r} \sum_{m=0}^{N} \sum_{n=m}^{N} \left( C_n^m \cos m\lambda + S_n^m \sin m\lambda \right) q^n P_n^m(t)
\]

Equation (6) can now be written as

\[
V(r, \psi, \lambda) \approx \sum_{m=0}^{N} \frac{GM}{r} \left\{ \sum_{n=m}^{N} q^n C_n^m P_n^m(t) \cos m\lambda + \sum_{n=m}^{N} q^n S_n^m P_n^m(t) \sin m\lambda \right\}
\]

\[
= \sum_{m=0}^{N} \frac{GM}{r} \left\{ V_m^{(1)} \cos m\lambda + V_m^{(2)} \sin m\lambda \right\}
\]

where

\[
V_m^{(1)} = \sum_{n=m}^{N} q^n C_n^m P_n^m(t)
\]

\[
V_m^{(2)} = \sum_{n=m}^{N} q^n S_n^m P_n^m(t)
\]

Now, since the Associated Legendre Functions \( P_n^m(t) \) are evaluated from the recurrence relationships (4) and (5), and the initial values \( P_0^0(t) = 1, \quad P_1^0(t) = t, \quad P_1^1(t) = u \)

Clenshaw's recurrence formula can be used to evaluate (8) and (9), each as the product of two numbers, one obtained from a reverse recurrence involving the coefficients \( C_n^m \) or \( S_n^m \) and the coefficients of the recurrence formula (4), and the other being a single evaluation of a modified Legendre function \( P_n^m(t) = q^n P_n^m(t) \) (Deakin 1998). For geopotential models of high degree and order, Clenshaw's recurrence formula and the associated summation algorithm is the only practical method of evaluating the sums.

Clenshaw's recurrence formula can also be conveniently applied to problems in geometric geodesy and map projections. Tscherning & Poder (1981) outline its use, referring to the fundamental work on the topic by König & Weise (1951) and Poder & Engsager (1998) provide matrix formulae and computer code for map projection computations. This paper will provide a detailed explanation of Clenshaw's recurrence formula and the associated summation algorithm and how they can be applied to the computation of meridian distance \( M \) on the ellipsoid (outlined above) and also how they can be used with the TM projection using the Karney-Krueger equations.
CLENSHAW'S RECURRENCE FORMULA AND SUM

Following Press et al. (1992), Clenshaw (1955) showed that for a sum $S$ where

$$ S = u_0 F_0(x) + u_1 F_1(x) + u_2 F_2(x) + \cdots + u_N F_N(x) $$

(10)

$u_k$ are coefficients independent of $x$ and $F_k(x)$ obey the recurrence relation

$$ F_{k+1}(x) = a_k F_k(x) + b_k F_{k-1}(x) $$

(11)

where $a_k, b_k$ may be functions of $x$ as well as of $k$. The sum $S$ can be evaluated from

$$ S = b_1 F_0(x) y_2 + F_1(x) y_1 + F_0(x) u_0 $$

(12)

where the quantities $y_k$ are obtained from the 'reverse' recurrence formula

$$ y_k = \begin{cases} 0, & \text{for } k > N \\ a_k y_{k+1} + b_{k+1} y_{k+2} + u_k, & \text{for } k = N, N - 1, N - 2, \ldots, 3, 2, 1 \end{cases} $$

(13)

Equation (13) is Clenshaw's recurrence formula and (12) is the associated sum; equations (12) and (13) combined are called Clenshaw's summation.

Clenshaw's summation can be explained by writing out (10) as

$$ S = u_N F_N(x) + u_{N-1} F_{N-1}(x) + u_{N-2} F_{N-2}(x) + \cdots + u_8 F_8(x) + u_7 F_7(x) + u_6 F_6(x) + u_5 F_5(x) + \cdots + u_2 F_2(x) + u_1 F_1(x) + u_0 F_0(x) $$

(14)

and re-arranging (13) as

$$ u_k = y_k - a_k y_{k+1} - b_{k+1} y_{k+2} $$

(15)

Then substituting (15) into (14) gives
\[
S = [y_N - a_N y_{N+1} - b_{N+1} y_{N+2}] F_N (x) \\
+ [y_{N-1} - a_{N-1} y_N - b_N y_{N+1}] F_{N-1} (x) \\
+ [y_{N-2} - a_{N-2} y_{N-1} - b_{N-1} y_N] F_{N-2} (x) \\
+ \cdots \\
+ [y_0 - a_0 y_1 - b_1 y_2] F_0 (x) \\
+ y_0 - a_0 y_1 - b_0 y_2 F_1 (x) \\
+ [y_0 - a_0 y_1 - b_0 y_2] F_2 (x) \\
+ [y_0 - a_0 y_1 - b_0 y_2] F_3 (x) \\
+ \cdots \\
+ [y_2 - a_2 y_3 - b_3 y_4] F_2 (x) \\
+ [y_2 - a_2 y_3 - b_3 y_4] F_3 (x) \\
+ [u_0 + b_1 y_2 - b_1 y_2] F_0 (x)
\]

(16)

noting that in the last line \(b_i y_2\) has been added and subtracted. Examining the terms containing a factor of \(y_k\) in (16) involves

\[
\left\{F_s (x) - a_s F_{s+1} (x) - b_s F_{s+2} (x)\right\} y_s
\]

(17)

and as a consequence of the recurrence relation (11) the term in \(\{\\}\) will equal zero and similarly for all other \(y_k\) down through and including \(y_2\). The only surviving terms in (16) are \(u_0, y_1\) and \(b_1 y_2\); and so the sum \(S\) is given by (12).

**HORNER'S FORM FOR POLYNOMIALS: A SPECIAL CASE OF CLENSHAW SUMMATION**

Suppose that we wish to evaluate the polynomial \(p (x)\) where

\[
p (x) = u_0 + u_1 x + u_2 x^2 + u_3 x^3 + \cdots + u_N x^N
\]

(18)

The evaluation of \(p (x)\) in this ‘monomial’ form requires \(N (N+1)/2\) multiplications and \(N\) additions. A more efficient evaluation of \(p (x)\) is obtained by re-arrangement into recursive Horner form\(^\dagger\)

\[
p (x) = u_0 + x \left( u_1 + x u_2 + x \left( u_3 + \cdots + x \left( u_{N-1} + x (u_N) \right) \right) \right)
\]

(19)

\(^\dagger\) Named after the English mathematician William George Horner (1786–1837) who used this form of polynomial representation in the solution of algebraic equations, although the method was known to others prior to Horner's use in 1830 (see MacTutor History of mathematics at http://www-history.mcs.st-andrews.ac.uk).
or alternatively
\[ p(x) = x \left( x \left( \cdots x(u_N + u_{N-1}) + \cdots + u_2 \right) + u_1 \right) + u_0 \] \tag{20}

Horner's form of recursive polynomial evaluation requires only \( N \) multiplications and \( N \) additions and is a special case of Clenshaw summation. This can be shown by firstly noting the similarity between equations (18) and (10), and secondly noting that \( x^k \) can be obtained from the two-term recurrence relation [see (11) with \( F_k(x) = x^k, a_k = x \) and \( b_k = 0 \)]
\[ x^{k+1} = x(x^k) \] \tag{21}

Defining a reverse recurrence as
\[ y_k = \begin{cases} 0, & \text{for } k > N \text{ and } k < 0 \\ x(y_{k+1}) + u_k, & \text{for } k = N, N-1, \ldots, 2, 1, 0 \end{cases} \] \tag{22}

The polynomial can be written as
\[ p(x) = u_N x^N + u_{N-1} x^{N-1} + \cdots + u_2 x^2 + u_1 x + u_0 \] \tag{23}

and re-arranging (22) as
\[ u_k = y_k - x(y_{k-1}) \] \tag{24}

then substituting (24) into (23) gives
\[ p(x) = \left[ y_N - x y_{N+1} \right] x^N + \left[ y_{N-1} - x y_N \right] x^{N-1} + \cdots + \left[ y_2 - x y_3 \right] x^2 + \left[ y_1 - x y_2 \right] x^1 + \left[ y_0 - x y_1 \right] x^0 \] \tag{25}

As a consequence of the recurrence relation (21) and that \( y_{N+1} = 0 \), the only surviving term in (25) will be \( y_0 x^0 = y_0 \) and \( p(x) = y_0 \) is obtained from (22) as
\[ \begin{align*}
  y_N &= u_N \\
  y_{N-1} &= x y_N + u_{N-1} \\
  & \vdots \\
  y_2 &= x y_3 + u_2 \\
  y_1 &= x y_2 + u_1 \\
  y_0 &= x y_1 + u_0
\end{align*} \] \tag{26}

Now, substituting successively for \( y_1, y_2, \ldots \) gives the polynomial as either (19) or (20).
RECURRANCE RELATIONS

Many useful functions in geodesy and map projections satisfy recurrence relations, e.g.,

Associated Legendre Functions:

\[ P_n^m(t) = \frac{2n-1}{n-m} t P_{n-1}^m(t) - \frac{n+m-1}{n-m} P_{n-2}^m(t) \]  (27)

Legendre polynomials (special case of (27) with \( m = 0 \))

\[ P_n(t) = \frac{2n-1}{n} t P_{n-1}(t) - \frac{n-1}{n} P_{n-2}(t) \]  (28)

Trigonometric functions

\[ \sin k \phi = 2 \cos \phi \sin(k-1)\phi - \sin(k-2)\phi \]  (29)
\[ \cos k \phi = 2 \cos \phi \cos(k-1)\phi - \cos(k-2)\phi \]  (30)

[Note that (29) and (30) can be derived from the product of trigonometric functions]

\[ 2 \sin A \cos B = \sin(A + B) + \sin(A - B) \]
\[ 2 \cos A \cos B = \cos(A + B) + \cos(A - B) \]

Writing \( A = (k-1)\phi \), \( B = \phi \) and rearranging gives the recurrence relations for \( \sin k\phi \) and \( \cos k\phi \).

In each of these recurrence relations certain initial values are required to begin the recurrence; e.g., special values of the Legendre polynomials \( P_0(t) = 1 \) and \( P_1(t) = t \) can be used as initial values in (28) to obtain, successively,

\[ P_2(t) = \frac{1}{2}(3t^2 - 1) \]
\[ P_3(t) = \frac{1}{2}(5t^3 - 3t) \]
\[ P_4(t) = \frac{1}{8}(35t^4 - 30t^2 + 3) \]

... And using the initial values \( \sin(0) = 0 \), \( \cos(0) = 1 \) in (29) and (30) gives, successively

\[ \sin 2\phi = 2 \cos \phi \sin \phi \]
\[ \sin 3\phi = 2 \cos \phi \sin 2\phi - \sin \phi \]
\[ \sin 4\phi = 2 \cos \phi \sin 3\phi - \sin 2\phi \]
\[ \sin 5\phi = \cdots \]
\[ \cos 2\phi = 2 \cos^2 \phi - 1 \]
\[ \cos 3\phi = 2 \cos \phi \cos 2\phi - \cos \phi \]
\[ \cos 4\phi = 2 \cos \phi \cos 3\phi - \cos 2\phi \]
\[ \cos 5\phi = \cdots \]
Also, it is sometimes required to evaluate sines and cosines of even multiples of angles, e.g., \( \sin 2\phi, \sin 4\phi, \sin 6\phi, \ldots \) in equation (1) for meridian distance \( M \). Recurrence relations for these even multiples are obtained by replacing \( \phi \) with \( 2\phi \) in (29) and (30) to give

\[
\sin 2k\phi = 2\cos 2\phi \sin 2(k-1)\phi - \sin 2(k-2)\phi
\]

\[
\cos 2k\phi = 2\cos 2\phi \cos 2(k-1)\phi - \cos 2(k-2)\phi
\]

CLENSHAW SUMMATION FOR MERIDIAN DISTANCE

The meridian distance \( M \) on an ellipsoid of revolution is the distance along a meridian from the equator to the point having latitude \( \phi \). The ellipsoid whose semi-axes lengths are \( a \) and \( b \) and \( a > b \) has the following geometric constants

flattening

\[
f = \frac{a - b}{a}
\]

(33)

eccentricity

\[
\varepsilon = \sqrt{\frac{a^2 - b^2}{a^2}}
\]

(34)

3rd flattening

\[
n = \frac{a - b}{a + b}
\]

(35)

and these constants are inter-related as follows

\[
\frac{b}{a} = 1 - f = \sqrt{1 - \varepsilon^2} = \frac{1 - n}{1 + n}
\]

(36)

\[
\varepsilon^2 = \frac{a^2 - b^2}{a^2} = f(2 - f) = \frac{4n}{(1 + n)^2}
\]

(37)

\[
n = \frac{f}{2 - f} = \frac{1 - \sqrt{1 - \varepsilon^2}}{1 + \sqrt{1 - \varepsilon^2}}
\]

(38)

The following absolutely convergent series for \( \varepsilon^2 \) and \( n \) (since \( 0 < \varepsilon^2, n < 1 \)) are also useful

\[
\varepsilon^2 = 4n - 8n^2 + 12n^3 - 16n^4 + 20n^5 - \cdots
\]

(39)

\[
n = \frac{1}{4} \varepsilon^2 + \frac{1}{8} \varepsilon^4 + \frac{5}{64} \varepsilon^6 + \frac{7}{128} \varepsilon^8 + \frac{21}{512} \varepsilon^{10} + \cdots
\]

(40)

There are various formula for meridian distance \( M \) often given as truncated series containing functions of \( \phi \) and \( \varepsilon^2 \) (Snyder 1987, eqn 3-21) but series containing functions of \( n \) and \( \phi \) converge more rapidly (since \( n \approx \frac{1}{4} \varepsilon^2 \)) and for the same accuracy require fewer terms and so are preferred here. Deakin et al. (2010) give
\[ M = \frac{a}{1 + n} \left\{ c_0 \phi + c_2 \sin 2\phi + c_4 \sin 4\phi + \cdots + c_{2N} \sin 2N\phi \right\} = \frac{a}{1 + n} \left\{ c_0 \phi + \sum_{k=1}^{N} c_{2k} \sin 2k\phi \right\} \] (41)

and for \( N = 4 \) the five coefficients \( c_0, c_2, \ldots, c_8 \) to order \( n^4 \) are

\[
\begin{align*}
    c_0 &= 1 + \frac{1}{4} n^2 + \frac{1}{64} n^4 + \cdots \\
    c_2 &= \frac{3}{2} n + \frac{3}{16} n^3 + \cdots \\
    c_4 &= \frac{15}{16} n^2 - \frac{15}{64} n^4 - \cdots \\
    c_6 &= -\frac{35}{48} n^3 + \cdots \\
    c_8 &= \frac{315}{512} n^4 - \cdots
\end{align*}
\] (42)

Note: This formulation of the meridian distance where the coefficients are given to order \( n^4 \) will be accurate to a micrometre \( (10^{-6} \text{m}) \) for any meridian distance on an ellipsoid whose flattening \( f \) is approximately \( 1/300 \).

Now the sum on the right-hand-side of (41) when \( N = 4 \) is

\[ S = \sum_{k=1}^{4} c_{2k} \sin k2\phi \]

and functions \( \sin 2\phi, \ldots, \sin 8\phi \) obey the recurrence relation (31). So \( S \) can be evaluated using Clenshaw summation.

Write the recurrence relation (31) in another form by replacing \( k \) with \( k + 1 \) giving

\[ \sin 2(k + 1)\phi = 2 \cos 2\phi \sin 2k\phi - \sin 2(k - 1)\phi \] (43)

Equation (43) has the same form as (11) where \( F_k(x) = \sin 2k\phi, \ a_k = 2 \cos 2\phi, \ b_k = -1 \) and Clenshaw's recurrence formula (13) becomes

\[
y_k = \begin{cases}
    0, & k > 4 \text{ and } k \leq 0 \\
    2 \cos 2\phi y_{k+1} - y_{k+2} + c_{2k}, & k = 4, 3, 2, 1
\end{cases}
\] (44)

The associated sum (see equation (12) with a slight modification discussed below) is

\[ S = -F_0(x)y_2 + F_1(x)y_1 \] (45)

with initial values \( F_0(x) = \sin(0) = 0 \) and \( F_1(x) = \sin 2\phi \) giving the sum \( S \) as

\[ S = \sum_{k=1}^{4} c_{2k} \sin 2k\phi = y_1 \sin 2\phi \] (46)

The compact formula for meridian distance \( M \) is when \( N = 4 \)

\[ M = \frac{a}{1 + n} \left\{ c_0 \phi + y_1 \sin 2\phi \right\} \] (47)

where \( y_1 \) is obtained from (44) with the coefficients \( c_{2k} \ (k = 1, \ldots, 4) \) given by (42).
Note that the sum (45) is obtained in the following manner by first writing $S$ as

$$S = \sum_{k=1}^{N=4} c_{2k} \sin k 2\phi = c_8 \sin 8\phi + c_6 \sin 6\phi + c_4 \sin 4\phi + c_2 \sin 2\phi$$  \hspace{1cm} (48)$$

then re-arranging (44) as

$$c_{2k} = y_k - 2 \cos 2\phi y_{k+1} + y_{k+2}$$  \hspace{1cm} (49)$$

Substituting (49) into (48) gives the sum $S$ as

$$S = \left[ y_4 - 2 \cos 2\phi y_5 + y_6 \right] \sin 8\phi$$
$$+ \left[ y_3 - 2 \cos 2\phi y_4 + y_5 \right] \sin 6\phi$$
$$+ \left[ y_2 - 2 \cos 2\phi y_3 + y_4 \right] \sin 4\phi$$
$$+ \left[ y_1 - 2 \cos 2\phi y_2 + y_3 \right] \sin 2\phi$$
$$+ \left[ -y_2 + y_1 \right] \sin 0\phi$$  \hspace{1cm} (50)$$

Noting that $y_5 = y_6 = 0$ by definition, and that the last line of (50) is equal to zero, examination of the terms containing a factor $y_4$ in (50) involves

$$\{ \sin 8\phi - 2 \cos 2\phi \sin 6\phi + \sin 4\phi \} y_4$$  \hspace{1cm} (51)$$

and as a consequence of the recurrence relation (31) the terms in $\{ \}$ will equal zero and similarly for all other $y_k$ down to and including $y_2$. The remaining terms are $y_1 \sin 2\phi$ and $y_2 \sin 0\phi = 0$ giving $S = y_4 \sin 2\phi$.

A Matlab function `Clenshaw_mdist()` is given in the Appendix that shows how Clenshaw summation is used in the computation of meridian distance to an approximation where $N = 8$. 


The Transverse Mercator (TM) projection is widely used in the geospatial community and is also known as the Gauss-Krueger projection acknowledging C.F. Gauss's original development of the ellipsoidal form of the projection and the work of L. Krueger (1912) who re-evaluated both Gauss' work and also the contributions by Oscar Schreiber who used a simplified form of Gauss's projection for the Prussian Land Survey of 1876-1923. Krueger (1912) published two sets of equations for the transformations between the ellipsoid and the TM projection: one set (also known as Redfearn's or Thomas's equations, (Redfearn 1948, Thomas 1952)) only accurate within a narrow band of longitude about a central meridian and another set that offer micrometre accuracy anywhere within 30° of the central meridian. These latter equations which are far more useful to the geospatial community have been re-evaluated and improved by Poder & Engsager (1998), Engsager & Poder (2007) and Karney (2011) and are hereinafter described as the Karney-Krueger equations to avoid confusion with other sets of TM projection equations.

Deakin et al. (2010) also provide a development of the Karney-Krueger equations and show how, in the forward transformation \( \phi, \lambda \rightarrow X, Y \), they represent a triple projection in two parts: the first part is a conformal mapping from the ellipsoid to a sphere (the conformal sphere of radius \( a \)) followed by a conformal transformation from this sphere to the plane using the spherical TM projection equations with spherical latitude \( \phi \) replaced by conformal latitude \( \phi' \). This two-step process is also known as the Gauss-Schreiber projection and the scale along the central meridian is not constant. The second part is the conformal mapping from the Gauss-Schreiber to the TM projection where the scale factor along the central meridian is made constant.

The sequence of operations and the Karney-Krueger equations for the forward \( (\phi, \lambda \rightarrow X, Y) \) and reverse \( (X, Y \rightarrow \phi, \lambda) \) transformations are set out below (Deakin et al. 2010) where \( \lambda_0 \) and \( m_0 \) are the central meridian longitude and central meridian scale factor respectively.
Forward transformation: $\phi, \lambda \to X,Y$ given $a, f, \lambda, m_0$

1. Compute ellipsoid constants $\varepsilon^2 = f(2 - f)$ and $n = \frac{f}{2 - f}$

2. Compute the rectifying radius $A$ from

$$A = \frac{a}{1 + n} \left(1 + \frac{1}{4} n^2 + \frac{1}{64} n^4 + \frac{1}{256} n^6 + \frac{25}{16384} n^8 + \ldots \right)$$  \hspace{1cm} (52)

3. Compute conformal latitude $\phi'$ from

$$\sigma = \sinh \left( \frac{\varepsilon \tan \phi}{\sqrt{1 + \tan^2 \phi}} \right)$$

$$\tan \phi' = \tan \phi \sqrt{1 + \sigma^2} - \sigma \sqrt{1 + \tan^2 \phi}$$  \hspace{1cm} (53)

4. Compute longitude difference $\omega = \lambda - \lambda_0$

5. Compute the $u, v$ Gauss-Schreiber coordinates from

$$u = a \tan^{-1} \left( \frac{\tan \phi'}{\cos \omega} \right)$$

$$v = a \sinh^{-1} \left( \frac{\sin \omega}{\sqrt{\tan^2 \phi' + \cos^2 \omega}} \right)$$  \hspace{1cm} (55)

6. Compute the coefficients $\{\alpha_{2k}\}$ for $k = 1, 2, \ldots, 8$ from

$$\alpha_2 = \frac{1}{2} n - \frac{n^2}{3} + \frac{5 n^3}{16} + \frac{41 n^4}{180} - \frac{127 n^5}{288} + \frac{7891 n^6}{37800} + \frac{72161 n^7}{38072} - \frac{18975107 n^8}{50803200} + \ldots$$

$$\alpha_4 = \frac{13}{48} n^2 - \frac{3 n^3}{5} + \frac{557 n^4}{1440} + \frac{281 n^5}{630} - \frac{1983433 n^6}{1935360} + \frac{13769 n^7}{28800} - \frac{148003883 n^8}{174182400} + \ldots$$

$$\alpha_6 = \frac{315}{240} n^3 - \frac{103 n^4}{140} + \frac{15061 n^5}{26880} + \frac{167603 n^6}{181440} - \frac{67102379 n^7}{29030400} + \frac{79682431 n^8}{7983600} + \ldots$$

$$\alpha_8 = \frac{161280 n^4}{49561} - \frac{179 n^5}{168} + \frac{6601661 n^6}{7257600} + \frac{97445 n^7}{49896} - \frac{40176129013 n^8}{7664025600} + \ldots$$

$$\alpha_{10} = -\frac{34729}{80640} n^5 - \frac{3418889 n^6}{1995840} + \frac{14644087 n^7}{9123840} + \frac{2605413599 n^8}{622702080} + \ldots$$

$$\alpha_{12} = \frac{212378941}{319334400} n^6 - \frac{30705481 n^7}{10378368} + \frac{175214326799 n^8}{58118860800} + \ldots$$

$$\alpha_{14} = -\frac{3152256789}{1522256789} n^7 - \frac{16759934899 n^8}{313510400} + \ldots$$

$$\alpha_{16} = \frac{1424729850961}{74392148240} n^8 - \ldots$$
7. Compute $X,Y$ coordinates from

$$
X = A \sum_{k=1}^{\infty} \alpha_{2k} \cos 2k \left( \frac{u}{a} \right) \sinh 2k \left( \frac{v}{a} \right)
$$

$$
Y = A \sum_{k=1}^{\infty} \alpha_{2k} \sin 2k \left( \frac{u}{a} \right) \cosh 2k \left( \frac{v}{a} \right)
$$

(57)

8. Compute $q$ and $p$ from

$$
q = -\sum_{k=1}^{\infty} 2k \alpha_{2k} \sin 2k \left( \frac{u}{a} \right) \sinh 2k \left( \frac{v}{a} \right)
$$

$$
p = 1 + \sum_{k=1}^{\infty} 2k \alpha_{2k} \cos 2k \left( \frac{u}{a} \right) \cosh 2k \left( \frac{v}{a} \right)
$$

(58)

9. Compute scale factor $m$ from

$$
m = m_0 \left( \frac{A}{a} \right) \frac{\sqrt{q^2 + p^2} \left( \sqrt{1 + \tan^2 \phi} \sqrt{1 - \varepsilon^2 \sin^2 \phi} \right)}{\sqrt{\tan^2 \phi' + \cos^2 \omega}}
$$

(59)

10. Compute grid convergence $\gamma$ from

$$
\gamma = \tan^{-1} \left( \frac{q}{p} \right) + \tan^{-1} \left( \frac{\tan \phi' \tan \omega}{\sqrt{1 + \tan^2 \phi'}} \right)
$$

(60)

**Inverse transformation:** $X,Y \rightarrow \phi, \lambda$ given $a,f,\lambda_0,m_0$

1. Compute ellipsoid constants $\varepsilon^2, n$

2. Compute the rectifying radius $A$ from equation (52)

3. Compute the coefficients $\{ \beta_{2k} \}$ for $k = 1,2,...,8$ from

$$
\beta_2 = -\frac{1}{2} n + \frac{2}{3} n^2 - \frac{37}{96} n^3 + \frac{1}{360} n^4 + \frac{81}{512} n^5 - \frac{96199}{604800} n^6 + \frac{540647}{38707200} n^7 - \frac{7944359}{67737600} n^8 + \ldots
$$

$$
\beta_4 = -\frac{1}{48} n^2 + \frac{1}{15} n^3 + \frac{437}{1440} n^4 - \frac{46}{105} n^5 + \frac{1118711}{3870720} n^6 - \frac{51841}{1209600} n^7 - \frac{24749483}{348364800} n^8 + \ldots
$$

$$
\beta_6 = -\frac{17}{480} n^3 + \frac{37}{840} n^4 - \frac{209}{90720} n^5 - \frac{5569}{5806080} n^6 - \frac{9261899}{177408000} n^7 + \frac{6457463}{177408000} n^8 + \ldots
$$

$$
\beta_8 = -\frac{4397}{161280} n^4 + \frac{11}{504} n^5 + \frac{830251}{7257600} n^6 - \frac{466511}{2494800} n^7 - \frac{324154477}{7664025600} n^8 + \ldots
$$

$$
\beta_{10} = -\frac{4583}{161280} n^5 + \frac{108847}{3991680} n^6 + \frac{8005831}{63866880} n^7 - \frac{22894433}{1245404160} n^8 + \ldots
$$

$$
\beta_{12} = -\frac{20648693}{638668800} n^6 + \frac{16363163}{518918400} n^7 + \frac{2204645983}{12915302400} n^8 + \ldots
$$

$$
\beta_{14} = -\frac{219941297}{5535129600} n^7 + \frac{497323811}{12454041600} n^8 + \ldots
$$

$$
\beta_{16} = -\frac{191773887257}{3719607091200} n^8 + \ldots
$$

(61)
4. Compute the ratios \( \frac{u}{a}, \frac{v}{a} \) from

\[
\frac{u}{a} = \frac{Y}{A} + \sum_{k=1}^{\infty} \beta_{2k} \sin 2k \left( \frac{Y}{A} \right) \cosh 2k \left( \frac{X}{A} \right)
\]
\[
\frac{v}{a} = \frac{X}{A} + \sum_{k=1}^{\infty} \beta_{2k} \cos 2k \left( \frac{Y}{A} \right) \sinh 2k \left( \frac{X}{A} \right)
\]  
(62)

5. Compute conformal latitude \( \phi' \) and longitude difference \( \omega \) from

\[
\tan \phi' = \frac{\sin \left( \frac{u}{a} \right)}{\sqrt{\sinh^2 \left( \frac{v}{a} \right) + \cos^2 \left( \frac{u}{a} \right)}}
\]
\[
\tan \omega = \sinh \left( \frac{v}{a} \right) \cos \left( \frac{u}{a} \right)
\]  
(63)

6. Compute \( t = \tan \phi \) by Newton-Raphson iteration using

\[
t_{n+1} = t_n - \frac{f(t_n)}{f'(t_n)}
\]  
(64)

where

\[
f(t) = t \sqrt{1 + \sigma^2} - \sigma \sqrt{1 + t^2} - t'
\]
\[
f'(t) = \left( \sqrt{1 + \sigma^2} \sqrt{1 + t^2} - \sigma t \right) \frac{(1 - \varepsilon^2) \sqrt{1 + t^2}}{1 + (1 - \varepsilon^2) t^2}
\]

and noting that \( t' = \tan \phi' \) is obtained from (63) and \( \sigma \) is obtained from (53) and that an initial value for \( t_1 \) can be taken as \( t_1 = t' = \tan \phi' \).

7. Compute latitude \( \phi = \tan^{-1} t \) and longitude \( \lambda = \lambda_0 \pm \omega \)

8. Compute the coefficients \( \{ \alpha_n \} \) from equations (56)

9. Compute \( q \) and \( p \) from equations (58)

10. Compute scale factor \( m \) from equation (59)

11. Compute grid convergence \( \gamma \) from equation (60)
Development of the complex functions of the transformation

In the forward transformation $\phi, \lambda \rightarrow X, Y$ the conformal mapping from the $u, v$ plane of the Gauss-Schreiber projection to the $X, Y$ plane of the TM projection is given by the complex function (Deakin et al. 2010)

$$\frac{1}{A}(Y + iX) = \frac{u}{a} + i\frac{v}{a} + \sum_{k=1}^{\infty} \alpha_{2k} \sin 2k \left( \frac{u}{a} + i\frac{v}{a} \right)$$  \hspace{1cm} (65)

and by expanding the trigonometric function and equating real and imaginary parts we obtain equations (57) assuming $\alpha_{2k}$ is real valued. Equation (65) may be simplified by letting

$$\eta = \frac{X}{A}, \xi = \frac{Y}{A}, \eta' = \frac{v}{a}, \xi' = \frac{u}{a}$$  \hspace{1cm} (66)

and substituting into (65) to give

$$\xi + i\eta = \xi' + i\eta' + \sum_{k=1}^{\infty} \alpha_{2k} \sin 2k(\xi' + i\eta')$$  \hspace{1cm} (67)

A further simplification is made by letting

$$\zeta = \xi + i\eta \quad \text{and} \quad \zeta' = \xi' + i\eta'$$  \hspace{1cm} (68)

and substituting (68) into (67) to give (Karney 2011)

$$\zeta = \zeta' + \sum_{k=1}^{\infty} \alpha_{2k} \sin 2k\zeta'$$  \hspace{1cm} (69)

Similarly, in the inverse transformation $X, Y \rightarrow \phi, \lambda$ the conformal mapping from the $X, Y$ plane of the TM projection to the $u, v$ plane of the Gauss-Schreiber projection is given by the complex function (Deakin et al. 2010)

$$\frac{1}{a}(u + iv) = \frac{Y}{A} + i\frac{X}{A} + \sum_{k=1}^{\infty} \beta_{2k} \sin 2k \left( \frac{Y}{A} + i\frac{X}{A} \right)$$  \hspace{1cm} (70)

where $\beta_{2k}$ is real and by expanding the trigonometric function and equating real and imaginary parts we obtain equations (62). Using the substitutions in (66) we may write (70) as

$$\xi' + i\eta' = \xi + i\eta + \sum_{k=1}^{\infty} \beta_{2k} \sin 2k(\xi + i\eta)$$  \hspace{1cm} (71)

and with (68) we may write (71) as (Karney 2011, noting the change of sign of $\{\beta\}$)

$$\zeta' = \zeta + \sum_{k=1}^{\infty} \beta_{2k} \sin 2k\zeta$$  \hspace{1cm} (72)
Now, differentiate (69) with respect to \( \zeta' \) to give

\[
\frac{d\zeta}{d\zeta'} = 1 + \sum_{k=1}^{\infty} 2k\alpha_{2k} \cos 2k\zeta'
\]

(73)

and write the derivative as

\[
\frac{d\zeta}{d\zeta'} = p + iq
\]

(74)

Then with (73) and (74) we have the complex equation

\[
\boxed{p + iq = 1 + \sum_{k=1}^{\infty} 2k\alpha_{2k} \cos 2k\zeta'}
\]

(75)

Expanding the trigonometric function in (75) and equating real and imaginary parts will give equations (58)

Now inspection of equations (69), (72) and (75) reveals that the right-hand-sides of each contains a summation of coefficients multiplied by functions that obey recurrence relations; hence the summations can be efficiently evaluated using Clenshaw summation, but in these instances the functions are complex and the Clenshaw summation is slightly more difficult.

**Complex Clenshaw Summation**

Here we will consider two complex sums:

(i) \( S_s = \sum_{k=1}^{N} \zeta_{2k} \sin 2k\zeta \) and

(ii) \( S_c = \sum_{k=1}^{N} 2k\gamma_{2k} \cos 2k\zeta \)

where \( S_s, S_c \) and \( \zeta_{2k} \) and \( \zeta \) have real and imaginary parts (note: \( \zeta_{2k} = \zeta_{2k} + i0 \))

The trigonometric expansions of the complex functions \( \sin 2\zeta \) and \( \cos 2\zeta \) where \( \zeta = \xi + i\eta \) are useful (noting also the relationships \( \sin(ix) = i\sinh x \) and \( \cos(ix) = \cosh x \))

\[
\sin 2\zeta = \sin(2\zeta + i2\eta) = \sin 2\zeta \cos i2\eta + \cos 2\zeta \sin i2\eta = \sin 2\zeta \cosh 2\eta + i \cos 2\zeta \sinh 2\eta
\]

(76)

\[
\cos 2\zeta = \cos(2\zeta + i2\eta) = \cos 2\zeta \cos i2\eta - \sin 2\zeta \sin i2\eta = \cos 2\zeta \cosh 2\eta - i \sin 2\zeta \sinh 2\eta
\]

(77)
Now consider the first complex sum

\[ S_S = \sum_{k=1}^{N} c_{2k} \sin 2k\zeta \]  

(78)

This is identical in form to the sum in the formula for meridian distance (41) and following the methods in the section on meridian distance we define a reverse complex recurrence as

\[ g_k = \begin{cases} 
0, & k > N \text{ and } k < 1 \\
2 \cos 2\zeta (g_{k+1}) - g_{k+2} + c_{2k}, & k = N, N-1, N-2, \ldots, 3, 2, 1 
\end{cases} \]  

(79)

where \( \zeta \), \( g_k \) and \( c_{2k} \) are complex numbers having real and imaginary parts and we write \( g_k \) and \( c_{2k} \) as

\[ g_k = g_k^{\text{Re}} + ig_k^{\text{Im}} \]  

(80)

\[ c_{2k} = c_{2k} + i0 \]  

(81)

With (77), (80) and (81) we write the recurrence (79) as

\[ g_k = g_k^{\text{Re}} + ig_k^{\text{Im}} = 2\left(g_{k+1}^{\text{Re}} + ig_{k+1}^{\text{Im}}\right)\left(\cos 2\zeta \cosh 2\eta - i \sin 2\zeta \sinh 2\eta\right) - \left(g_{k+2}^{\text{Re}} + ig_{k+2}^{\text{Im}}\right) + (c_{2k} + i0) \]

Expanding and equating real and imaginary parts gives two recurrence relations

**Real**

\[ g_k^{\text{Re}} = \begin{cases} 
0, & k > N \text{ and } k < 1 \\
2 \cos 2\zeta \cosh 2\eta \left(g_{k+1}^{\text{Re}}\right) + 2 \sin 2\zeta \sinh 2\eta \left(g_{k+1}^{\text{Im}}\right) - g_{k+2}^{\text{Re}} + c_{2k}, & k = N, N-1, \ldots, 3, 2, 1 
\end{cases} \]  

(82)

**Imaginary**

\[ g_k^{\text{Im}} = \begin{cases} 
0, & k > N \text{ and } k < 1 \\
2 \cos 2\zeta \cosh 2\eta \left(g_{k+1}^{\text{Im}}\right) - 2 \sin 2\zeta \sinh 2\eta \left(g_{k+1}^{\text{Re}}\right) - g_{k+2}^{\text{Im}}, & k = N, N-1, \ldots, 3, 2, 1 
\end{cases} \]  

(83)

And the complex sum (78) is given as

\[ S_S = S_S^{\text{Re}} + iS_S^{\text{Im}} = \sum_{k=1}^{N} c_{2k} \sin 2k\zeta = g_1 \sin 2\zeta \]  

(84)

With the aid of (80) and (76) we have

\[ S_S^{\text{Re}} + iS_S^{\text{Im}} = \left(g_1^{\text{Re}} + ig_1^{\text{Im}}\right)\left(\sin 2\zeta \cosh 2\eta + i \cos 2\zeta \sinh 2\eta\right) \]

and expanding and equating real and imaginary parts gives

\[ S_S^{\text{Re}} = g_1^{\text{Re}} \sin 2\zeta \cosh 2\eta - g_1^{\text{Im}} \cos 2\zeta \sinh 2\eta \]  

(85)

\[ S_S^{\text{Im}} = g_1^{\text{Im}} \sin 2\zeta \cosh 2\eta + g_1^{\text{Re}} \cos 2\zeta \sinh 2\eta \]  

(86)
where \( g_1^{\text{Re}} \) is evaluated from (82) and \( g_1^{\text{Im}} \) is evaluated from (83).

Now consider the second complex sum

\[
S_c = \sum_{k=1}^{N} 2kc_{2k} \cos 2k\zeta
\]  

(87)

In a similar manner to before, but using the recurrence (32) we define a reverse complex recurrence

\[
d_k = \begin{cases} 
0, & k > N \text{ and } k < 1 \\
2 \cos 2\zeta (d_{k+1}) - d_{k+2} + 2kc_{2k}, & k = N, N-1, N-2, \ldots, 3, 2, 1
\end{cases}
\]  

(88)

where \( \zeta, d_k \) and \( 2kc_{2k} \) are complex having real and imaginary parts and we write \( d_k \) and \( 2kc_{2k} \) as

\[
d_k = d_k^{\text{Re}} + id_k^{\text{Im}}
\]  

(89)

\[
2kc_{2k} = 2kc_{2k} + i0
\]  

(90)

With (77) (89) and (90) we write the recurrence (88) as

\[
d_k = d_k^{\text{Re}} + id_k^{\text{Im}} = 2\left(d_{k+1}^{\text{Re}} + id_{k+1}^{\text{Im}}\right)\left(\cos 2\xi \cosh 2\eta - i \sin 2\xi \sinh 2\eta\right) - \left(d_{k+2}^{\text{Re}} + id_{k+2}^{\text{Im}}\right) + \left(2kc_{2k} + i0\right)
\]

Expanding and equating real and imaginary parts gives two recurrence relations

Real

\[
d_k^{\text{Re}} = \begin{cases} 
0, & \text{for } k > N \text{ and } k < 1 \\
2 \cos 2\xi \cosh 2\eta \left(d_{k+1}^{\text{Re}}\right) \\
+ 2 \sin 2\xi \sinh 2\eta \left(d_{k+1}^{\text{Im}}\right) \\
- d_{k+2}^{\text{Re}} + 2kc_{2k}, & \text{for } k = N, N-1, \ldots, 3, 2, 1
\end{cases}
\]  

(91)

Imaginary

\[
d_k^{\text{Im}} = \begin{cases} 
0, & \text{for } k > N \text{ and } k < 1 \\
2 \cos 2\xi \cosh 2\eta \left(d_{k+1}^{\text{Im}}\right) \\
- 2 \sin 2\xi \sinh 2\eta \left(d_{k+1}^{\text{Re}}\right) - d_{k+2}^{\text{Im}}, & \text{for } k = N, N-1, \ldots, 3, 2, 1
\end{cases}
\]  

(92)

And the complex sum (87) is given as

\[
S_c = S_c^{\text{Re}} + iS_c^{\text{Im}} = \sum_{k=1}^{N} 2kc_{2k} \cos 2k\zeta = -d_2 + d_1 \cos 2\zeta
\]  

(93)

With the aid of (89) and (77) we have

\[
S_c^{\text{Re}} + iS_c^{\text{Im}} = -\left(d_2^{\text{Re}} + id_2^{\text{Im}}\right) + \left(d_1^{\text{Re}} + id_1^{\text{Im}}\right)\left(\cos 2\xi \cosh 2\eta - i \sin 2\xi \sinh 2\eta\right)
\]

and expanding and equating real and imaginary parts gives

\[
S_c^{\text{Re}} = -d_2^{\text{Re}} + d_1^{\text{Re}} \cos 2\xi \cosh 2\eta + d_1^{\text{Im}} \sin 2\xi \sinh 2\eta
\]  

(94)
where $d_2^{\text{Re}}, d_1^{\text{Re}}$ are evaluated from (91) and $d_2^{\text{Im}}, d_1^{\text{Im}}$ are evaluated from (92).

### Clenshaw summation for $X, Y$ coordinates given $u, v$ Gauss-Schreiber coordinates

The $X, Y$ coordinates are evaluated from equations (57), but with the aid of complex Clenshaw summation [see equations (65) to (69) and equations (78), (85) and (86) with appropriate change of symbols], this evaluation may be expressed as

\[
X = A \left\{ \eta' + g_1^{\text{Im}} \sin 2\zeta' \cosh 2\eta' + g_1^{\text{Re}} \cos 2\zeta' \sinh 2\eta' \right\} \\
Y = A \left\{ \xi' + g_1^{\text{Re}} \sin 2\zeta' \cosh 2\eta' - g_1^{\text{Im}} \cos 2\zeta' \sinh 2\eta' \right\}
\]

where $\eta' = \frac{v}{a}, \xi' = \frac{u}{a}$ and $g_1^{\text{Re}}, g_1^{\text{Im}}$ are computed from the recurrence relations

\[
g_k^{\text{Re}} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( f_k^{\text{Re}} + f_{k+1}^{\text{Im}} \right) - g_{k+2}^{\text{Re}} + \alpha_{2k}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases}
\]

\[
g_k^{\text{Im}} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( f_k^{\text{Im}} - f_{k+1}^{\text{Re}} \right) - g_{k+2}^{\text{Im}}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases}
\]

where

\[
f_k^{\text{Re}} = \cos 2\zeta' \cosh 2\eta' \\
f_k^{\text{Im}} = \sin 2\zeta' \sinh 2\eta'
\]

and the coefficients $\{\alpha\}$ are given by (56).

### Clenshaw summation for point scale factor $m$ and grid convergence $\gamma$ given $u, v$ Gauss-Schreiber coordinates

$m$ and $\gamma$ are evaluated from equations (59) and (60). Compute $p$ and $q$ first from equations (58) then with the aid of complex Clenshaw summation [see equations (75), (87), (94) and (95) with appropriate change of symbols] the evaluation of $p$ and $q$ may be expressed as

\[
p = 1 - d_2^{\text{Re}} + d_1^{\text{Re}} f^{\text{Re}} + d_1^{\text{Im}} f^{\text{Im}} \\
q = - d_2^{\text{Im}} + d_1^{\text{Im}} f^{\text{Re}} - d_1^{\text{Re}} f^{\text{Im}}
\]

where $f^{\text{Re}}, f^{\text{Im}}$ are given by (99) and $d_2^{\text{Re}}, d_2^{\text{Im}}, d_1^{\text{Re}}, d_1^{\text{Im}}$ are computed from the recurrence relations
Real

\[ d_k^{Re} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( f_k^{Re} d_{k+1}^{Re} + f_k^{Im} d_{k+1}^{Im} \right) - d_k^{Re} + 2k\alpha_{2k}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases} \]  

(101)

Imaginary

\[ d_k^{Im} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( f_k^{Re} d_{k+1}^{Im} - f_k^{Im} d_{k+1}^{Re} \right) - d_k^{Im}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases} \]  

(102)

and the coefficients \( \{\alpha\} \) are given by (56).

Clenshaw summation for \( u,v \) Gauss-Schreiber coordinates given \( X,Y \) coordinates

The \( u,v \) Gauss-Schreiber coordinates are evaluated from equations (62), but with the aid of complex Clenshaw summation [see equations (70) to (72) and (78), (85) and (86) with appropriate change of symbols], this evaluation may be expressed as

\[
\begin{align*}
    u &= a \left\{ \xi + w_1^{Re} \sin 2\xi \cosh 2\eta - w_1^{Im} \cos 2\xi \sinh 2\eta \right\} \\
    v &= a \left\{ \eta + w_1^{Im} \sin 2\xi \cosh 2\eta + w_1^{Re} \cos 2\xi \sinh 2\eta \right\}
\end{align*}
\]

(103)

where \( \eta = \frac{X}{A}, \xi = \frac{Y}{A} \) and \( w_1^{Re}, w_1^{Im} \) are computed from the recurrence relations

Real

\[ w_k^{Re} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( F_k^{Re} w_{k+1}^{Re} + F_k^{Im} w_{k+1}^{Im} \right) - w_k^{Re} + \beta_{2k}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases} \]  

(104)

Imaginary

\[ w_k^{Im} = \begin{cases} 0, & \text{for } k > N \text{ and } k < 1 \\ 2 \left( F_k^{Re} w_{k+1}^{Im} - F_k^{Im} w_{k+1}^{Re} \right) - w_k^{Im}, & \text{for } k = N, N - 1, \ldots, 3, 2, 1 \end{cases} \]  

(105)

where

\[
\begin{align*}
    F_k^{Re} &= \cos 2\xi \cosh 2\eta \\
    F_k^{Im} &= \sin 2\xi \sinh 2\eta
\end{align*}
\]

(106)

and the coefficients \( \{\beta\} \) are given by (61).

Two Matlab functions \( \text{TM\_Cart.m} \) and \( \text{TM\_Geo.m} \) given in the Appendix show how Clenshaw summation is used in the forward and inverse transformations between the ellipsoid and the TM projection plane.
Matlab function for meridian distance using Clenshaw summation

function mdist = Clenshaw_mdist(a,flat,lat)

% mdist = Clenshaw_mdist(a,flat,lat) Function computes the meridian
distance on an ellipsoid defined by semi-major axis (a) and denominator
of flattening (flat) from the equator to a point having latitude (lat)
in radians. Note that if lat is an array of latitudes then mdist will
be an array also.

% For example, at the command prompt type:
% >> format long g;
% >> a = 6378137;
% >> flat = 298.257222101;
% >> lat = [0.2 0.5 0.8 1];
% >> mdist = Clenshaw_mdist(a,flat,lat')
% should return
% mdist =
% 1267256.19418549
% 3170243.93586599
% 5077926.68596087
% 6352852.63825197

% Function:  Clenshaw_mdist()
% Usage:     mdist = Clenshaw_mdist(a,flat,lat)
% Author:    R.E.Deakin,
% School of Mathematical & Geospatial Sciences, RMIT University
% GPO Box 2476V, MELBOURNE, VIC 3001, AUSTRALIA.
% email: rod.deakin@rmit.edu.au
% Version  1.0  12 January 2011

% Purpose:
% Function Clenshaw_mdist(a,f,lat) will compute the meridian distance on
% an ellipsoid defined by semi-major axis a and flat, the denominator of
% the flattening f where f = 1/flat. Latitude is given in radians.
% Functions required:
% Variables:
% a    - semi-major axis of ellipsoid
% A    - constant in Clenshaw's recurrence formula
% c    - (1,N) row vector of coefficients
% c0   - coefficient
% n2,nx - powers of n
% f    - f = 1/flat is the flattening of ellipsoid
% flat - denominator of flattening of ellipsoid
% lat  - latitude (radians)
% mdist - meridian distance
% N    - maximum order of sum
% S    - sum of products c(k)*sin(k*2*lat) = y1*sin(2*lat)
% y1,y2 - values from Clenshaw's recurrence
% Remarks:
% This function uses Helmert's formula for meridian distance with Clenshaw
% Summation to give a compact expression for the meridian distance. An
% explanation of Clenshaw Summation is given in [2]. The coefficients c0,
% c2,c4, ..., c16 are given as functions of n up to order n^8 and are given
% in [2]. The formula for meridian distance (mdist) is:
% mdist = a*/(1+n)*(c0*lat + y1*sin(2*lat))
% where y1 is obtained from Clenshaw's reverse recurrence relation.
% References:
Matlab function for TM projection coordinates given geographical coordinates using Clenshaw summation

```matlab
function TM_Cart(a,flat,lat,lon,lon0,E0,N0,m0)
    % TM_Cart(a,flat,lat,lon,lon0,E0,N0,m0) A function to calculate the Cartesian E and N coordinates on a Transverse Mercator projection given ellipsoid parameters a (semi-major axis in metres), flat (denominator of flattening), lat (latitude in radians), lon (longitude in radians), lon0 (central meridian longitude in radians), E0,N0 (false origin offsets in metres) and m0 (central meridian scale factor). This function uses the Karney-Krueger equations with Clenshaw summation.
    %
    % For example, at the command prompt type:
```

% set ellipsoid constants
f  = 1/flat;
n  = f/(2-f);
% set the order of the summation
N  = 8;
% set an array c() for the coefficients c2,c4,c6,...,c16
c = zeros(N,1);
n2 = n*n;
nx = 1;
c0   =  n2*(n2*(n2*(n2*25/16384 + 1/256) + 1/64) + 1/4) + 1;
nx   =  nx*n;
c(1) =  nx*(n2*(n2*15/2048 + 3/128) + 3/16) - 3/2;
nx   =  nx*n;
c(2) = -nx*(n2*(n2*105/8192 + 75/2048) + 15/64) - 15/16;
nx   =  nx*n;
c(3) =  nx*(n2*(n2*245/6144 + 175/768) - 35/48);
nx   =  nx*n;
c(4) = -nx*(n2*(n2*1323/32768 + 441/2048) - 315/512);
nx   =  nx*n;
c(5) =  nx*(n2*2079/10240 - 693/1280);
nx   =  nx*n;
c(6) = -nx*(n2*1573/8192 - 1001/2048);
nx   =  nx*n;
c(7) = -nx*6435/14336;
nx   =  nx*n;
c(8) =  nx*109395/262144;
% set up y1 and y2 for Clenshaw's reverse recurrence
y2 = 0;
y1 = 0;
% calculate y1 from Clenshaw's reverse recurrence
A = 2*cos(2*lat);
for k = N:-2:1
    y2 = A.*y1-y2+c(k);
    y1 = A.*y2-y1+c(k-1);
end
% compute the sum S
S = y1.*sin(2*lat);
% compute meridian distance mdist
mdist = a/(1+n)*(c0*lat + S);
return
```
%>> d2r = 180/pi;
%>> a = 6378137;
%>> flat = 298.257222101;
%>> lat = -37/d2r;
%>> lon = 144/d2r;
%>> lon0 = 147/d2r;
%>> E0 = 500000;
%>> N0 = 10000000;
%>> m0 = 0.9996;
%>> TM_Cart(a,flat,lat,lon,lon0,E0,N0,m0)
%
% Should return:
%
% Transverse Mercator coordinates
%===========================================
% Ellipsoid:
% semi-major axis a = 6378137.000
% flattening f = 1/298.257222101
% Geodetic coordinates:
% Latitude = -37 0 0.000000 (D M S)
% Longitude = 144 0 0.000000 (D M S)
% CM Long. = 147 0 0.000000 (D M S)
% dlon = -2 59 60.000000 (D M S)
% Meridian Distance:
% M = -4096510.974734
% Central Meridian Scale Factor:
% m0 = 0.999600000
% X,Y Cartesian coordinates:
% X = -267068.947750
% Y = -4100720.981730
% E,N Cartesian coordinates:
% E0 = 500000.000000
% N0 = 10000000.000000
% E = m0*X + E0 = 233037.879829
% N = m0*Y + N0 = 5900919.306662
% Point Scale Factor:
% m = 1.000478061
% Grid Convergence:
% gamma = 1 48 23.441616 (D M S)
% Conformal latitude:
% clat = -36 48 54.995213 (D M S)

===========================================================================
% Function: TM_Cart(a,flat,lat,lon,lon0,E0,N0,m0)
%
% Author: R.E.Deakin
% School of Mathematical & Geospatial Sciences
% RMIT University
% GPO Box 2476V, MELBOURNE VIC 3001
% email: rod.deakin@rmit.edu.au
% Version 1.0 12-Aug-2010
% Version 1.1 17-Jan-2011
%
% Functions Required:
% [D,M,S] = DMS(DecDeg);
% mdist = Clenshaw_mdist(a,flat,lat)
%
% Variables:
% A rectifying radius (metres)
% alpha N,1 vector of coefficients alpha
% clat conformal latitude (radians)
% dlon longitude difference from central meridian (radians)
% d2r degree to radian conversion factor d2r = 57.29577951...
% d1_Re,d1_Im Real and Imaginary parts of Clenshaw reverse recurrence
% d2_Re,d2_Im Real and Imaginary parts of Clenshaw reverse recurrence
% etap TM coord related to false origin
% Eta-primed, Etap = v/a
% E0 false origin offset west from true origin
% e1 eccentricity of ellipsoid
% e2 eccentricity-squared
% f flattening of ellipsoid f = 1/flat
% f_Re,f_Im Real and Imaginary parts of trigonometric function
% flat denominator of flattening
% g1_Re,g1_Im Real and Imaginary parts of Clenshaw reverse recurrence
% g2_Re,g2_Im Real and Imaginary parts of Clenshaw reverse recurrence
% lat latitude (radians)
% lon longitude 9radians)
% lon0 longitude of central meridian (radians)

23
% point scale factor
% mdist         meridian distance (metres)
% N0           central meridian scale factor
% N            maximum order of Clenshaw summation
% N_coord      TM coord related to false origin
% N0           false origin offset south from true origin
% n            3rd flattening of ellipsoid
% n2,nx        2nd and higher powers of n
% p,q          projection quantities for computation of point scale factor
% and grid convergence
% sigma        variable used in the computation of conformal latitude
% t            tan(lat)
% tp           t-primed = tan(clat)
% u,v          Gauss-Schreiber coordinates (metres)
% X,Y          TM coordinates related to true origin
% Xip          Xi-primed, Xip = u/a
%
% Remarks:
% This function uses Clenshaw Summation to compute Cartesian coordinates
% on a TM projection using the Karney-Krueger equations. An explanation
% of Clenshaw summation is given in [1] and the Karney-Krueger equations
% for the TM projection (forward transformation) are given in [1] and [2].
%
% References:
% [1] Deakin, R.E., 'Some Applications of Clenshaw’s Recurrence Formula in
%      Map Projections’, School of Mathematical and Geospatial
%      Sciences, RMIT University, Melbourne, Australia, January 2011.
%      Krueger Projection, Presented at the Victorian Regional Survey
%      Conference, Warrnambool, 10-12 September 2010.
%==========================================================================
%
% set degree to radian conversion factor
d2r   = 180/pi;
%==========================================================================
% 1. Compute ellipsoid constants e^2, f, n and powers of n and meridian
%    distance
%==========================================================================
f  = 1/flat;
e2 = f*(2-f);
n  = f/(2-f);
% compute meridian distance
mdist = Clenshaw_mdist(a,flat,lat);
%==========================================================================
% 2. Compute rectifying radius A
%==========================================================================
n2 = n*n;
A  = a/(1+n)*(n2*(n2*(n2*(n2*25/16384 + 1/256) + 1/64) + 1/4) + 1);
%==========================================================================
% 3. Compute conformal latitude clat
%==========================================================================
e1 = sqrt(e2);
t  = tan(lat);
sigma = sinh(e1*atanh(e1*t/sqrt(1+t^2)));
tp = t*sqrt(1+sigma^2)-sigma*sqrt(1+t^2);
clat = atan(tp);
%==========================================================================
% 4. Compute longitude difference from central meridian
%==========================================================================
dlon = lon-lon0;
%==========================================================================
% 5. Compute u,v coords of Gauss-Schreiber projection
%==========================================================================
s = sin(dlon);
c = cos(dlon);
u = a*atan(tp/c);
v = a*asinh(s/sqrt(tp^2+c^2));
%==========================================================================
% 6. Compute the coefficients alpha
%==========================================================================
N = 8;
% set a vector alpha[] for the coefficients alpha

24
alpha = zeros(N,1);
% compute the coefficients alpha (Horner form)

nx = n;
alpha(1) = -nx*(n*(n*(n*(n*(n*(n*(n*(n*18975107/50803200 - 72161/387072)... - 7891/37800) + 127/288) - 41/180) - 5/16) + 2/3) - 1/2);

nx = nx*n;
alpha(2) = nx*(n*(n*(n*(n*148003883/174182400 + 13769/28800) - 1983433/1935360) + 281/630) + 557/1440) - 3/5)+ 13/48);

nx = nx*n;
alpha(3) = nx*(n*(n*(n*(n*(n*79682431/79833600 - 67102379/29030400) + 167603/181440) + 15061/26880) - 103/140) + 61/240);

nx = nx*n;
alpha(4) = nx*(n*(n*(n*(n*(n*40176129013/7664025600 - 97445/49896) - 6601661/7257600) + 179/168) - 49561/161280); nx = nx*n;
alpha(5) = nx*(n*(n*(n*2605413599/622702080 + 14644087/91378240) - 3/5)+ 13/48);

nx = nx*n;
alpha(6) = nx*(n*(n*175214326799/58118860800 - 30705481/10378368) + 212378941/319334400); nx = nx*n;
alpha(7) = -nx*(n*16759934899/3113510400 - 1522256789/1383782400);

nx = nx*n;
alpha(8) = nx*1424729850961/74392148240;

%==========================================================================
% 7. Compute the X,Y Cartesian coords using Clenshaw summation
%==========================================================================
% Calculate Xi-prime and Eta-prime
Xip = u/a;
Etap = v/a;
% set up g1_Re, g2_Re and g1_Im, g2_Im for Clenshaw's reverse recurrence
g2_Re = 0;
g1_Re = 0;
g2_Im = 0;
g1_Im = 0;
% calculate g1_Re and g1_Im from Clenshaw's reverse recurrence.  See [1]
% equations (97), (98) and (99)
f_Re = cos(2*Xip)*cosh(2*Etap);
f_Im = sin(2*Xip)*sinh(2*Etap);
for k = N:-2:1
    g2_Re = 2*(f_Re*g1_Re + f_Im*g1_Im) - g2_Re + alpha(k);
g2_Im = 2*(f_Re*g1_Im - f_Im*g1_Re) - g2_Im;
g1_Re = 2*(f_Re*g2_Re + f_Im*g2_Im) - g1_Re + alpha(k-1);
g1_Im = 2*(f_Re*g2_Im - f_Im*g2_Re) - g1_Im;
end
% Calculate X and Y coords.  See [1], equations (96)
f1 = sin(2*Xip)*cosh(2*Etap);
f2 = cos(2*Xip)*sinh(2*Etap);
X = A*(Etap + g1_Im*f1 + g1_Re*f2);
Y = A*(Xip  + g1_Re*f1 - g1_Im*f2);
%==========================================================================
% 8. Compute q and p using Clenshaw summation
%==========================================================================
% set up d1_Re, d2_Re and d1_Im, d2_Im for Clenshaw's reverse recurrence
d2_Re = 0;
d1_Re = 0;
d2_Im = 0;
d1_Im = 0;
% calculate d1_Re, d1_Im and d2_Re, d2_Im from Clenshaw's reverse
% recurrence.  See [1], equations (99), (101) and (102)
for k = N:-2:1
d2_Re = 2*(f_Re*d1_Re + f_Im*d1_Im) - d2_Re + 2*k*alpha(k);
d2_Im = 2*(f_Re*d1_Im - f_Im*d1_Re) - d2_Im;
d1_Re = 2*(f_Re*d2_Re + f_Im*d2_Im) - d1_Re + 2*(k-1)*alpha(k-1);
d1_Im = 2*(f_Re*d2_Im - f_Im*d2_Re) - d1_Im;
end
% Compute p and q. See [1], equations (100)
p = 1 - d2_Re + d1_Re*f_Re + d1_Im*f_Im;
q = - d2_Im + d1_Im*f_Re - d1_Re*f_Im;

%==========================================================================
% 9. Compute point scale factor m
%==========================================================================
m = m0*(A/a)*sqrt(p^2+q^2)*sqrt((1+t^2)*sqrt(1-e2*sin(lat)^2)/sqrt(tp^2+cos(dlon)^2));

%==========================================================================
% 10. Compute grid convergence
%==========================================================================
gamma = atan(abs(q/p)) + atan(abs(tp*tan(dlon))/sqrt(1+tp^2));

%==========================================================================
% 11. Compute East and North coordinates
%==========================================================================
E_coord = m0*X + E0;
N_coord = m0*Y + N0;

% print result to screen
fprintf('%nTransverse Mercator coordinates');
fprintf('%n==================================');
fprintf('%nEllipsoid:');
fprintf('%n semi-major axis a = %13.3f',a);
fprintf('%nflattening f = 1/%13.9f',flat);
fprintf('%nGeodetic coordinates:');
[D,M,S] = DMS(lat*d2r);
if D == 0 && lat < 0
    fprintf('%nLatitude = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('%nLatitude = %4d %2d %9.6f (D M S)',D,M,S);
end;
[D,M,S] = DMS(lon*d2r);
if D == 0 && lon < 0
    fprintf('%nLongitude = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('%nLongitude = %4d %2d %9.6f (D M S)',D,M,S);
end;

% print Meridian Distance:
fprintf('%nMeridian Distance:');
fprintf('%nM = %15.6f',mdist);

% print Central Meridian Scale Factor:
fprintf('%nCentral Meridian Scale Factor:');
fprintf('%nm0 = %11.9f',m0);

% print X,Y Cartesian coordinates:
fprintf('%nX = %16.6f',X);
fprintf('%nY = %16.6f',Y);

% print N,E Cartesian coordinates:
fprintf('%nN = %16.6f',N);
fprintf('%nE = %16.6f',E);

% print Point Scale Factor:
fprintf('%nPoint Scale Factor:');
fprintf('%nm = %11.9f',m);

% print Grid Convergence:
fprintf('%nGrid Convergence:');
[D,M,S] = DMS(gamma*d2r);
if D == 0 && gamma < 0
    fprintf('%ngamma = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('%ngamma = %4d %2d %9.6f (D M S)',D,M,S);
end;

% print conformal latitude
[D,M,S] = DMS(clat*d2r);
if D == 0 && clat < 0
    fprintf('%nclat = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('%nclat = %4d %2d %9.6f (D M S)',D,M,S);
end;

26
end
fprintf('\n\n');
Matlab function for Geographical coordinates given TM projection coordinates using Clenshaw summation

function TM_Geo(a,flat,E_coord,N_coord,E0,N0,m0,lon0)
% TM_Geo(a,flat,E_coord,N_coord,E0,N0,m0,lon0) A function to calculate
% latitude and longitude given the ellipsoid parameters a, flat
% (denominator of flattening), Transverse Mercator (TM) E,N projection
% coordinates and the false origin offsets E0,N0, the central meridian
% scale factor m0 and the central meridian longitude lon0 (radians).
% For example, at the command prompt type:
% d2r = 180/pi;
% a = 6378137;
% flat = 298.257222101;
% E_coord = 123456;
% N_coord = 7654321;
% E0 = 500000;
% N0 = 10000000;
% m0 = 0.9996;
% lon0 = 147/d2r;
% TM_Geo(a,flat,E_coord,N_coord,E0,N0,m0,lon0)
% Should return
% Transverse Mercator geographical coordinates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ellipsoid:
% semi-major axis a = 6378137.000
% flattening f = 1/298.257222101
% Geodetic coordinates:
% Latitude = -21 10 25.329097 (D M S)
% iterations = 3
% Longitude = 143 22 28.173748 (D M S)
% CM Long. = 147 0 0.000000 (D M S)
% dlon = -3 37 31.826252 (D M S)
% Meridian Distance:
% M = -2342308.892951
% Central Meridian Scale Factor:
% m0 = 0.999600000
% X,Y Cartesian coordinates:
% X = -376694.677871
% Y = -2346617.647059
% E,N Cartesian coordinates:
% E0 = 500000.000000
% N0 = 10000000.000000
% E = m0*X + E0 = 123456.000000
% N = m0*Y + N0 = 7654321.000000
% Point Scale Factor:
% m = 1.001352560
% Grid Convergence:
% gamma = 1 18 39.850479 (D M S)
% Conformal latitude:
% clat = -21 2 39.915535 (D M S)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function: TM_Geo(a,flat,E_coord,N_coord,E0,N0,m0,lon0)
% Author: R.E.Deakin
% School of Mathematical & Geospatial Sciences
% RMIT University
% GPO Box 2476V, MELBOURNE VIC 3001
% email: rod.deakin@rmit.edu.au
% Version 1.0 19-Aug-2010
% Version 1.1 24-Jan-2011
% Functions Required:
% [D,M,S] = DMS(DecDeg);
% mdist = Clenshaw_mdist(a,flat,lat)
% Variables:
% A rectifying radius (metres)
% alpha N,1 row vector of coefficients alpha
% beta N,1 vector of coefficient alpha
% clat    conformal latitude (radians)
% dlon   longitude difference from central meridian (radians)
% d2r    degree to radian conversion factor d2r = 57.29577951...
% d1_Re,d1_Im Real and Imaginary parts of Clenshaw reverse recurrence
% d2_Re,d2_Im Real and Imaginary parts of Clenshaw reverse recurrence
% E_coord TM coord related to false origin
% Eta    Eta = X/A
% Etap   Eta-primed, Etap = v/a
% E0     false origin offset west from true origin
% e1     eccentricity of ellipsoid
% e2     eccentricity-squared
% f      flattening of ellipsoid f = 1/flat
% f1,f2  Real and Imaginary parts of trigonometric function
% flat   denominator of flattening
% f_Re,f_Im Real and Imaginary parts of trigonometric function
% F_Re,F_Im Real and Imaginary parts of trigonometric function
% gamma  grid convergence (radians)
% lat    latitude (radians)
% lon    longitude 9radians)
% lon0   longitude of central meridian (radians)
% m      point scale factor
% mdist  meridian distance (metres)
% m0     central meridian scale factor
% N      maximum order of Clenshaw summation
% N_coord TM coord related to false origin
% N0     false origin offset south from true origin
% n2,nx  2nd and higher powers of n
% p,q    projection quantities for computation of point scale factor
% sigma  variable used in the computation of conformal latitude
% t      tan(lat)
% tp     t-primed = tan(clat)
% u,v    Gauss-Schreiber coordinates (metres)
% w1_Re,w1_Im Real and Imaginary parts of Clenshaw reverse recurrence
% w2_Re,w2_Im Real and Imaginary parts of Clenshaw reverse recurrence
% X,Y    TM coordinates related to true origin
% Xi     Xi = Y/A
% Xip    Xi-primed, Xip = u/a

% Remarks:
% This function uses Clenshaw Summation to compute geographical
% coordinates given Cartesian coordinates on a TM projection using the
% Karney-Krueger equations. An explanation of Clenshaw summation is given
% in [1] and the Karney-Krueger equations for the TM projection (inverse
% transformation) are given in [1] and [2].
% References:
% [1] Deakin, R.E., ‘Some Applications of Clenshaw’s Recurrence Formula in
% Map Projections’, School of Mathematical and Geospatial
% Science, RMIT University, Melbourne, Australia, January 2011.
% Krueger Projection, Presented at the Victorian Regional Survey
% Conference, Warrnambool, 10-12 September 2010.
%==========================================================================
% set values of constants
% d2r   = 180/pi;
%==========================================================================
% 1. Compute ellipsoid constants e^2, f, n and powers of n and meridian
distance
%f   = 1/flat;
e2  = f*(2-f);
n   = f/(2-f);

%==========================================================================
% 2. Compute rectifying radius A
%n2   = n*n;
%A   = a/(1+n)*n2*(n2*(n2*(n2*25/16384 + 1/256) + 1/64) + 1/4) + 1);

%==========================================================================
% 3. Compute the coefficients beta
%set a vector beta[] for the coefficients alpha
%beta = zeros(N,1);

%cclat = conformal latitude (radians)
%dlon  = longitude difference from central meridian (radians)
%d2r   = degree to radian conversion factor d2r = 57.29577951...
%d1_Re,d1_Im  = Real and Imaginary parts of Clenshaw reverse recurrence
%d2_Re,d2_Im  = Real and Imaginary parts of Clenshaw reverse recurrence
%E_coord = TM coord related to false origin
%Eta    = Eta = X/A
%Etap   = Etap = v/a
%E0     = false origin offset west from true origin
%e1     = eccentricity of ellipsoid
%e2     = eccentricity-squared
%f      = flattening of ellipsoid f = 1/flat
%f1,f2  = Real and Imaginary parts of trigonometric function
%flat   = denominator of flattening
%f_Re,f_Im  = Real and Imaginary parts of trigonometric function
%F_Re,F_Im  = Real and Imaginary parts of trigonometric function
%gamma  = grid convergence (radians)
%lat     = latitude (radians)
%lon     = longitude 9radians)
%lon0    = longitude of central meridian (radians)
%m      = point scale factor
%mdist  = meridian distance (metres)
m0     = central meridian scale factor
%N      = maximum order of Clenshaw summation
%N_coord = TM coord related to false origin
%N0     = false origin offset south from true origin
%n2,nx  = 2nd and higher powers of n
%p,q    = projection quantities for computation of point scale factor
%sigma = variable used in the computation of conformal latitude
%t      = tan(lat)
%tp     = t-primed = tan(clat)
%u,v    = Gauss-Schreiber coordinates (metres)
w1_Re,w1_Im  = Real and Imaginary parts of Clenshaw reverse recurrence
w2_Re,w2_Im  = Real and Imaginary parts of Clenshaw reverse recurrence
%X,Y   = TM coordinates related to true origin
%Xi    = Xi = Y/A
%Xip   = Xi-primed, Xip = u/a

% Remarks:
% This function uses Clenshaw Summation to compute geographical
% coordinates given Cartesian coordinates on a TM projection using the
% Karney-Krueger equations. An explanation of Clenshaw summation is given
% in [1] and the Karney-Krueger equations for the TM projection (inverse
% transformation) are given in [1] and [2].
% References:
% [1] Deakin, R.E., ‘Some Applications of Clenshaw’s Recurrence Formula in
% Map Projections’, School of Mathematical and Geospatial
% Science, RMIT University, Melbourne, Australia, January 2011.
% Krueger Projection, Presented at the Victorian Regional Survey
% Conference, Warrnambool, 10-12 September 2010.
%==========================================================================
% set values of constants
d2r   = 180/pi;
%==========================================================================
% 1. Compute ellipsoid constants e^2, f, n and powers of n and meridian
distance
%f   = 1/flat;
e2  = f*(2-f);
n   = f/(2-f);
%==========================================================================
% 2. Compute rectifying radius A
%n2   = n*n;
%A   = a/(1+n)*n2*(n2*(n2*(n2*25/16384 + 1/256) + 1/64) + 1/4) + 1);
%==========================================================================
% 3. Compute the coefficients beta
%set a vector beta[] for the coefficients alpha
%beta = zeros(N,1);
% compute the coefficients alpha (Horner form)
% nx = n;
beta(1) = -nx*(n*(n*(n*(n*(n*(n*(n*(n*7944359/67737600 - 5406467/38707200)...
    + 96199/604800) - 1/360) + 37/840) - 17/480))
% nx = nx*n;
beta(2) = -nx*(n*(n*(n*(n*(n*(n*24749483/348364800 + 51841/1209600)...
    - 1118711/38707200) + 46/105) - 437/1440) + 1/15) + 1/48);% nx = nx*n;
beta(3) =  nx*(n*(n*(n*(n*(n*6457463/17740800 - 9261899/58060800)...
    - 5569/90720) + 209/4480) + 37/840) - 17/480));% nx = nx*n;
beta(4) = -nx*(n*(n*(n*(n*324154477/7664025600 + 466511/2494800)...
    - 830251/7257600) - 11/504) - 4397/161280);% nx = nx*n;
beta(5) = -nx*(n*(n*22894433/124540416 - 8005831/63866800)...
    - 108847/3991680) - 4583/161280);% nx = nx*n;
beta(6) = nx*(n*(n*2204645983/12915302400 + 16363163/518918400)...
    - 20648693/638668800);% nx = nx*n;
beta(7) = nx*(n*497323811/12454041600 - 219941297/5535129600);% nx = nx*n;
beta(8) = -nx*191773887257/3719607091200;

%==========================================================================
% 4. Compute the ratios Etap = u/a, Xip = v/a
%==========================================================================
% compute the TM X,Y coords
X = (E_coord-E0)/m0;
Y = (N_coord-N0)/m0;
% compute Eta and Xi
Eta = X/A;
Xi  = Y/A;
% set up w1_Re, w2_Re and w1_Im, w2_Im for Clenshaw's reverse recurrence
w2_Re = 0;
w1_Re = 0;
w2_Im = 0;
w1_Im = 0;
% calculate w1_Re and w1_Im from Clenshaw's reverse recurrence.  See [1]
f1 = sin(2*Xi)*cosh(2*Eta);
f2 = cos(2*Xi)*sinh(2*Eta);
for k = N:-2:1
    w2_Re = 2*(F_Re*w1_Re + F_Im*w1_Im) - w2_Re + beta(k);
w2_Im = 2*(F_Re*w1_Im - F_Im*w1_Re) - w2_Im;
w1_Re = 2*(F_Re*w2_Re + F_Im*w2_Im) - w1_Re + beta(k-1);
w1_Im = 2*(F_Re*w2_Im - F_Im*w2_Re) - w1_Im;
end
% Calculate Etap = u/a and Xip = v/a.  See [1], equations (103)
f1 = sin(2*Xi)*cosh(2*Eta);
f2 = cos(2*Xi)*sinh(2*Eta);
Xip = Xi + w1_Re*f1 - w1_Im*f2;
Etap = Eta + w1_Im*f1 + w1_Re*f2;
% Calculate the u,v Gauss-Schreiber coords
u = Xip*a;
v = Etap*a;
%==========================================================================
% 5. Compute the conformal latitude clat and longitude difference dlon
%==========================================================================
clat = atan(tp);
dlon = atan(sinh(Etap)/cos(Xip));

%==========================================================================
% 6. Compute the t = tan(lat) using Newton-Raphson iteration
%==========================================================================
e1 = sqrt(e2);
t = tp;
iter = 1;
while 1
    t2 = t*t;
    s = sinh(e1*atanh(e1*t/sqrt(1+t^2)));
    s2 = s*s;
    corrn = (tp-t*sqrt(1+s2)+s*sqrt(1+t2))/((sqrt(1+s2)*sqrt(1+t2)-s*t);
    if abs(corrn)<1e-15
        break
    end
    t = t+corr;
    iter = iter+1;
    if iter>10
        break
    end
end
%
% 7. Compute the latitude and longitude and meridian distance
%
%==========================================================================
% 7. Compute the latitude and longitude and meridian distance
%
%==========================================================================
lat = atan(t);
lon = lon0+dlon;

% compute meridian distance
mdist = Clenshaw_mdist(a,flat,lat);

%==========================================================================
% 8. Compute the coefficients alpha
%==========================================================================
% set a vector alpha[] for the coefficients alpha
alpha = zeros(N,1);

% compute the coefficients alpha (Horner form)

nx = n;
alpha(1) = nx*(n*(n*(n*(n*(n*(n*(n*18975107/50803200 - 72161/387072)...
    - 7891/37800) + 127/288) - 41/180) + 2/3) - 1/2);

nx = nx*n;
alpha(2) = nx*(n*(n*(n*(n*(n*(n*148003883/174182400 + 13769/28800)...
    - 198343/1935360) + 281/630) + 557/1440) - 3/5)+ 13/48);

nx = nx*n;
alpha(3) = nx*(n*(n*(n*(n*(n*79682431/798336000 - 67102379/290304000)...
    + 167603/181440) + 15061/26880) - 3/5)+ 13/48);

nx = nx*n;
alpha(4) = nx*(n*(n*(n*40176129013/7664025600 - 97445/49896)...
    - 6601661/725600) + 179/168) - 49561/161280);

nx = nx*n;
alpha(5) = nx*(n*(n*2605413599/622702080 + 14644087/9123840)...
    - 3418889/1995840) + 34729/80640);

nx = nx*n;
alpha(6) = nx*(n*(n*175214326799/58118860800 - 30705481/10378368)...
    + 212378941/319334400);

nx = nx*n;
alpha(7) = nx*(n*167599934899/3113510400 - 1522256789/1383782400));

nx = nx*n;
alpha(8) = nx*1424729850961/743921418240;

%==========================================================================
% 9. Compute q and p using Clenshaw summation
%==========================================================================
% set up d1_Re, d2_Re and d1_Im, d2_Im for Clenshaw's reverse recurrence
d2_Re = 0;
d1_Re = 0;
d2_Im = 0;
d1_Im = 0;

% calculate d1_Re, d1_Im and d2_Re, d2_Im from Clenshaw's reverse
% recurrence.  See [1], equations (99), (101) and (102)
for k = N:-2:1
    d2_Re = 2*(f*Re*Re + f*Im*Im) - 2*k*alpha(k);
    d2_Im = 2*(f*Re*Im - f*Im*Re) - 2*k*alpha(k);
end

%==========================================================================
% 10. Compute the series coefficients
%==========================================================================

\[ d_1_{\text{Im}} = 2*(f_{\text{Re}}*d_2_{\text{Im}} - f_{\text{Im}}*d_2_{\text{Re}}) - d_1_{\text{Im}}; \]

end

% Compute p and q. See [1], equations (100)
p = 1 - d_2_{\text{Re}} + d_1_{\text{Re}}*f_{\text{Re}} + d_1_{\text{Im}}*f_{\text{Im}};
q = -(d_2_{\text{Im}} + d_1_{\text{Im}}*f_{\text{Re}} - d_1_{\text{Re}}*f_{\text{Im}});

%==========================================================================
% 10. Compute point scale factor m
%==========================================================================
m = m^0*(A/a)*sqrt(p^2+q^2)*(sqrt(1+t^2)*sqrt(1-e^2*sin(lat)^2))/sqrt(tp^2+cos(dlon)^2));

%==========================================================================
% 11. Compute grid convergence
%==========================================================================
gamma = atan(abs(q/p)) + atan(abs(tp*tan(dlon))/sqrt(1+t^2));

% print results to screen
fprintf('Transverse Mercator geographical coordinates');
fprintf('Ellipsoid:');
[f,flat] = (13.9f',lat);
fprintf('Geodetic coordinates:');
[D,M,S] = DMS(lat*d2r);

fprintf('
Latitude = %4d %2d %9.6f (D M S)',D,M,S);
if D == 0 && lat < 0
    fprintf('
Latitude = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('
Latitude = %4d %2d %9.6f (D M S)',D,M,S);
end;

fprintf('
Iterations = %2d',iter);

[D,M,S] = DMS(lon*d2r);
if D == 0 && lon < 0
    fprintf('
Longitude = -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('
Longitude = %4d %2d %9.6f (D M S)',D,M,S);
end;

[D,M,S] = DMS(dlon*d2r);
if D == 0 && dlon < 0
    fprintf('

Meridian Distance = %15.6f',mdist);

fprintf('
Central Meridian Scale Factor = %11.9f',m0);

fprintf('
E,N Cartesian coordinates:');
[E0,N0] = m0*X + E0 = %16.6f',N0);

fprintf('
E = m0*X + E0 = %16.6f',E_coord);

fprintf('
N = m0*Y + N0 = %16.6f',N_coord);

fprintf('
Point Scale Factor = %11.9f',m);

fprintf('
Grid Convergence = %4d %2d %9.6f (D M S)',D,M,S);
if D == 0 && gamma < 0
    fprintf('

Conformal latitude = %4d %2d %9.6f (D M S)',D,M,S);
else
    fprintf('
Conformal latitude = %4d %2d %9.6f (D M S)',D,M,S);
end

% print conformal latitude
fprintf('Conformal latitude:');
[D,M,S] = DMS(clat*d2r);
if D == 0 && clat < 0
    fprintf('

Conformal latitude = %4d %2d %9.6f (D M S)',D,M,S);
else
    fprintf('

Conformal latitude = %4d %2d %9.6f (D M S)',D,M,S);
end

32
REFERENCES


URL: http://docs.lib.noaa.gov/rescue/cgs_specpubs/QB27U35no2511952.pdf
