Supplementary PROJ.4 Notes—
Swiss Oblique Mercator Projection

January 1, 2003

Release 4.3.3

The Swiss Oblique Mercator Projection (a tentative name based upon the Swiss usage in their CH1903 grid system) is based upon a three step process:

1. conformal transformation of ellipsoid coordinates to a sphere,
2. rotational translation of the spherical system so that the specified projection origin will lie on the equator, and
3. application of the Mercator projection to transform to the cartesian system.

The projection is conformal with no scale error at the projection origin ($k_0 = 1$) and increasing symmetrically as a function of Northing (y) distance from the origin.

This projection is selected in the PROJ.4 system by `+proj=somerc` with parameters `+lon_0=` and `+lat_0` required to specify projection origin. Optionally, false easting and northing, `+x_0=` and `+y_0`, may be used as well as origin scale factor `k_0`. For computing the Swiss CH1903 grid coordinates with the proj.4 system, `+init=world:CH1903` may be used.

Forward Projection

The first step is the conformal conversion of geodetic coordinates, ϕ–λ, to coordinates on a sphere, ϕ'–λ', with the following equations:

$$\ln\tan\left(\frac{\pi}{4}+\frac{\phi'}{2}\right) = c\left[\ln\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) - \frac{e}{2}\ln\left(\frac{1+e\sin\phi}{1-e\sin\phi}\right)\right] + K \quad (1)$$

$$\lambda' = c(\lambda - \lambda_0) \quad (2)$$

where λ' is designated longitude of the origin and e is ellipsoid eccentricity. Constant c is determined from:

$$c = \left(1 + \frac{e^2\cos^4\phi_0}{1-e^2}\right)^{1/2} \quad (3)$$

where ϕ_0 is the latitude of the projection origin. The equivalent origin latitude on the sphere, ϕ'_0, is obtained by:

$$\sin\phi'_0 = \frac{\sin\phi_0}{c} \quad (4)$$

which, along with c and ϕ_0, substituted into Eqn. 1 and solved for K.

Next the spherical coordinates are rotated by ϕ'_0 about an axis on the equatorial plane and perpendicular to the plane of the central meridian (Wray's *simple oblique*).

$$\sin\phi'' = \cos\phi'_0\sin\phi' - \sin\phi'_0\cos\phi'\cos\lambda' \quad (5)$$

$$\sin\lambda'' = \cos\phi'\sin\lambda'/\cos\phi'' \quad (6)$$

Finally, transform the rotated coordinates to cartesian by the spherical form of the Mercator projection:

$$y = R\ln\tan\left(\frac{\pi}{4}+\frac{\phi''}{2}\right) + y_0 \quad (7)$$

$$x = R\lambda'' + x_0 \quad (8)$$

where R is the geometric mean of the merdinal and parallel radius at the projection origin:

$$R = k_0 a \frac{\sqrt{1-e^2}}{1 - e^2 \sin^2 \phi_0} \tag{9}$$

where a is the major axis of the ellipsoid and k_0 is scaling a factor. The scaling factor is not used (= 1) in the Swiss version but it can reduce overall scale error if appropriately applied.

Inverse Projection

To get the cartesian coordinates back into the rotated spherical system:

$$\phi'' = 2\left[\tan^{-1}\exp\left(\frac{y-y_0}{R}\right) - \frac{\pi}{4}\right] \tag{10}$$

$$\lambda'' = \frac{x - x_0}{R} \tag{11}$$

Rotate the spherical coordinates back to the original position:

$$\sin\phi' = \cos\phi'_0 \sin\phi'' + \sin\phi'_0 \cos\phi'' \cos\lambda'' \tag{12}$$

$$\sin\lambda' = \cos\phi'' \sin\lambda'' / \cos\phi' \tag{13}$$

The ellipsoid longitude is obtained from Eqn. 2 but latitude requires application of Newton-Raphson method for a solution of Eqn. 1: $x_{n+1} = x_n - f(x_n)/f'(x_n)$. The correction ratio is:

$$\left[C + \ln\tan\left(\frac{\pi}{4} + \frac{\phi}{2}\right) - \frac{e}{2}\ln\left(\frac{1 + e\sin\phi}{1 - e\sin\phi}\right)\right] \frac{1 - e^2\sin^2\phi}{1 - e^2} \tag{14}$$

where:

$$C = \frac{1}{c}\left[K - \ln\tan\left(\frac{\pi}{4} + \frac{\phi'}{2}\right)\right] \tag{15}$$

Use ϕ' as the initial estimate of ϕ and iterate until the correction ratio has reached sufficient tolerance. The function converges rapidly.

Standard Definition

The following entry has been placed in the `world` file:

```
<CH1903> # Swiss Coordinate System
    +proj=somerc +lat_0=46d57'8.660"N +lon_0=7d26'22.500"E
    +ellps=bessel +x_0=600000 +y_0=200000
    +k_0=1.  no_defs <>
```

Note, the original are "reverse engineered" from decimal values as I suspect original specifications were in DMS format—needs verification.