Cartographic Projection Procedures for the UNIX Environment—A User's Manual

Gerald I. Evenden

September 24, 1995

Abstract

A tutorial description of the general usage of the cartographic projection program *proj* (release 3) along with specific cartographic parameters and illustrations of the approximately 70 cartographic projections supported by the program is presented. The program is designed as a standard UNIX filter utility to be employed with other programs in the generation of maps and charts and, in many cases, used in map digitizing applications. Tables and shell scripts are also provided for conversion of State Plane Coordinate Systems to and from geographic coordinates.

Introduction

Program *proj* (release 3) is a standard UNIX filter function which converts geographic longitude and latitude coordinates into cartesian coordinates, $(\lambda, \phi) \rightarrow (x, y)$, by means of a wide variety of cartographic projection functions. For many of the projection functions the inverse conversion, $(x, y) \rightarrow (\lambda, \phi)$, can also be performed. The principle purpose of this user's manual is to provide complete details of the technical usage and execution of program *proj*, supplemented with examples and some practical applications. However, cartographic characteristics of the projections are given minimal coverage in this manual and the reader should refer to *Map Projections—A Working Manual* (Snyder, 1987) and *An Album of Map Projections* (Snyder and Voxland, 1989) for more complete discussions. Program *proj* supplements these publications in terms of computational applications in the UNIX environment.

This manual is divided into three main sections:

- command line parameters controlling *proj* operation related to input-output format and file control,
- specification and usage of general cartographic parameters, and
- description of specific projections and control parameter usage.

The style of the first two sections is tutorial and contains many examples of usage. The first section is also summarized in the UNIX *man*(1) style document available on the host system or in system documentation. The reader is expected to have basic familiarity with at least one of the UNIX shell programs such as Bourne (*sh*(1)), C (*csh*(1)) or Korn (*ksh*(1)).

History

The original *proj* was developed in the early 1980's as a RATFOR (a FORTRAN preprocessor) program (Evenden, 1983) with much of the code derived from the Geological Survey's General Cartographic Transformation Package (GCTP) (superceded by a more portable Version II by Elassal, 1987). Program *proj* was recoded in the C language when the MAPGEN package (Evenden and Botbol, 1985), of which *proj* is an integral part, was transfered to the UNIX operating system. Based upon the formularies in the aforementioned publications many new projections have been added to the program. Source code for program *proj* is written in the Kernighan and Ritchie (K&R) (1978) style and is available from the author.

Typological Conventions

As with mathematics texts, descriptions of computer software execution can use typology to emphasize or augment the text with various fonts for clarification. Examples of this usage have already been introduced with the slant font being employed for program names. In addition to the slant font, references to standard UNIX utility programs will be suffixed with a number in parenthesis refering to the chapter in standard UNIX manuals. File names mentioned in the text will also be in slant font.

For command line arguments, scripts or file contents a typewriter style font will be employed (i.e. `+proj=poly`). The typewriter font is always used for literal usage and explanations, but if part of the descriptive text is refering to an abstract value to be completed later, then italics are used. For example, `+proj=`*name* is used when discussing the argument *name* in the abstract sense and `+proj=poly` is used for an exact definition of usage.

To ensure the recognition of the importance of a blank character in text descriptions, the cup character, ␣, is used. For example, the cup character in `-f␣'%.6f'` emphasizes that there is a blank after the `-f` and not a spacing operation of the typesetter.

Basic Usage

In this section the basic execution of *proj* will be described with empasis on command line control and the nature of the input and output of the program. Since previous releases of *proj* may be installed on the user's host machine the user must check and verify that he is using the proper release. This is readily done by executing:

```
$ proj
```

which should result in the following message displayed on the user's terminal:

```
usage(3.0): proj [ -bcefiormstwW [args] ] [ +opts[=arg] ] [ files ]
```

(note: `$␣` is the shell prompt for interactive examples). The critical element of the response is that `3.x` appears. If the release 3 message does not appear or the message

```
proj: not found
```

appears (indicating the program is not in the user's search path) and consultation with the host system manager is required.

The `.x` in the release number represents versions within the release that will either contain corrections to the previous versions or new features that are compatible with previous versions of the same release. Some of the hyphened command line parameters of the previous release of *proj* (implicitly release 2, but never documented as such) are not compatible with the current version. However, for MAPGEN users the current release is fully compatible.

Before the input and output data format and the hyphen-control arguments can be discussed, a detail from the cartographic parameter section must be mentioned.

Except for the above execution, *proj* requires the selection of a cartographic projection which is performed by the `+proj=`*name* option where *name* is the abbreviated name of one of the supported projections. If a projection is not named, then the following message will appear:

```
<proj>: projection not selected
program abnormally terminated
```

For the following examples `+proj=poly` (p. 36) will be used.

Inputing Data

There are two ways data are input to program *proj*: through *stdin* or files named on the command line. The usage of *stdin* means that data can either be piped from the *stdout* of a previous process on the runline by means of the | shell control character, redirected from a file by <*filename*, or entered directly from the user's terminal if nothing is specified. For example:

```
cat file1 file2 | proj +proj=poly
proj +proj=poly file1 file2
```

will produce identical results where the files *file1* and *file2* are read and processed in left to right order. The difference is that *cat*(1) is employed to put the data in a *stdout* stream to be read by *proj*'s *stdin*. Sometimes it is convenient to execute *proj* interactively:

```
$ proj +proj=poly
10 45
786491.58       5033320.60
-4 30
-385874.54      3326668.76
^D
$ ...
```

where `^D` is the control D character for indicating the end of terminal input. The large numbers are *proj*'s cartesian results (in meters) to the input values of longitude and latitude values in degrees.

When file names are specified on a *proj* command line the data are only input from these files and *stdin* is never read. Conversely, if there are no files specified then input is assumed to be only from *stdin*. It is occasionally useful to process data from both specified files and *stdin*. This can be achieved by using a `␣-␣` placed in the command line where *stdin* is to be read by *proj*. For example:

```
cat file1 file2 | proj +proj=poly fileA - fileB
```

is equivalent to:

```
proj +proj=poly fileA file1 file2 fileB
```

Form of Input Data

As already demonstated, the input data consist of lines of two numeric values separated by "white space" (either blanks or tabs). The general case of specifying geographic coordinates is somewhat more complex than the simple integral degrees used in the example discussed so far and it is necessary to introduce the DMS (Degrees, Minutes and Seconds) system. As an example, a typical value for latitude would be $45^\circ 25' 15.22''$ N. But because there is no degree symbol in the ASCII character set and imbedded blanks are not allowed, the equivalent DMS value is expressed as `45d25'15.22"N` or as `45d25'15.22`. This example shows that intuitively obvious material at the end of the value may be dropped; viz., the positive sign associated with N and the seconds symbol. If the latitude value is in the southern hemisphere then either `-45d25'15.22` or `45d25'15.22S` is acceptable. Either a prefix sign (`+` or `-`) or suffix letter (upper or lower case `N, S, E, W`) may be used and western longitude and southern latitude are always considered negative values.

If the geographic data are in degrees and fractional minutes then typical DMS values might be `33d22.5s` and `-120d45.666`. Note that in this case the apostrophe or minute mark can be dropped. Similarly, fractional degrees are simply expressed by `77.5N` or `-33.33`. The `d` and `'` are only required when respective minute and second subfields are employed and act as subfield delimiters or when a preceeding subfield is missing (i.e. `15'33.5` could be used when degrees are zero).

One important element to note is that although the use of suffix letter might lead to the conclusion that *proj* would recognize the values as being longitude or latitude, **it does not**. The values for forward projection are always in longitude and latitude order on the record unless the `-r` command line option is used. For example, if the input data are in latitude-longitude order then the earlier example would be:

```
$ proj +proj=poly -r
45 10
786491.58       5033320.60
30 -4
-385874.54      3326668.76
^D
$ ...
```

To perform an inverse projection, either execute program *invproj* (which is just an alternate name for *proj*) or use the command line parameter `+inv`. Consequently,

```
invproj +proj=poly
proj +proj=poly +inv
```

are equivalent. Input data are now cartesian data in (x, y) order or, if the `-r` command line switch is used, in (y, x) order. Typically, the form of the cartesian data is in fixed format as shown in the output of the previous example, thus:

```
$ invproj +proj=poly
786491.58 5033320.60
10dE    45dN
-385874.54 3326668.76
4dW     30dN
^D
$ ...
```

An important feature of *proj* is its ability to pass through non-cartographic information in the data file. If an input record contains a `#` as the first character on the line it is simply copied to the output. Also, any information after the data fields is copied to the output after the converted data. For example, a data file *cities* contains:

```
# coordinates for a few cities
45d15n 71d07w Boston, United States
40d40n 73d58w New York, United States
48d51n 2d20e Paris, France
51d30n 7'w London, England
```

When *proj* processes this file the results are:

```
$ proj +proj=poly -r cities
# coordinates for a few cities
-4887590.49     7317961.48 Boston, United States
-5542524.55     6982689.05 New York, United States
171224.94       5415352.81 Paris, France
-8101.66        5707500.23 London, England
```

If required, the default `#` tag character can be changed by means of the `-t`*char* command line switch, where *char* will become the new tag character.

Situations will arise where input data cannot be directly processed by *proj* and one of the standard UNIX utilities will be required as a prefilter. A typical example is data stored in a "packed" format as shown in the file *packed.data*:

```
2205501202277
2215451201182
2216111201625
```

In this case assume that latitude and longitude are in degrees and hundredths of minutes, with an implied decimal point, and western longitude is positive. A solution with the UNIX utility *awk*(1) would be:

```
awk -f unpack packed.data | tee /dev/tty | proj ...
```

where the contents of file *unpack* is the *awk* control script:

```
{
    if (substr($0,1,1) == "#")
        print
    else {
        lnd=substr($0,7,3)+0
        lnm=substr($0,10,4)*.01
        ltd=substr($0,1,2)+0
        ltm=substr($0,3,4)*.01
        printf "-%dd%.2f %dd%.2f%s\n", \
            lnd,lnm,ltd,ltm,substr($0,14)
    }
}
```

The intermediate results of *awk*'s output displayed by *tee*(1) on the user's terminal will appear as:

```
-120d22.77 22d05.50
-120d11.82 22d15.45
-120d16.25 22d16.11
```

Data Output

The previous sections have already demonstrated default forms of the data output of *proj*: fixed format with two decimal digits of precision for forward projected cartesian output and DMS output of inverse projected geographic data. In the latter case there is an implied default precision of $\pm 0.0005''$ but trailing zeros and zero subfields are not output. That is, if the seconds field is zero, it is dropped and if both the seconds and minutes field are zero, both are dropped.

The output precision of the cartesian data can be controlled by the `-f` command line option which provides use of a C language *printf*(3) floating point format specification. But a change may be made without C software experience by simply changing the number 2 in the default format control `-f␣'%.2f'` to the new, desired precision. If four digits of precision are desired, then the command line would appear as:

```
proj +proj=poly -f '%.4f' ...
```

For other changes, it might be wise for the user to seek the advice of a C programmer.

There are several modifications possible for inverse projection output of geographic data. If the previously discussed `-f` option is used, the output is simply signed decimal degrees with specified fractional digits. DMS output precision can be altered with the `-w`*digits* command line option where *digits* is the number of significant fractional digits of the seconds output subfield. The range of *digits* can be from 0 (integral seconds) to 9.

A more formal DMS output may be generated with the `-W`*digits* command line option. As with the `-w` option, *digits* is the precision of the seconds subfield, but in this case all subfields are output and leading zeros are used in the seconds and minutes field. An example command line of:

```
invproj +proj=poly -W0
```

would produce the following output:

```
0d53'56"E       1d48'31"N
56d01'54"W      32d53'22"S
2d21'49"W       40d36'44"N
0d00'00"W       0d00'00"N
```

When neither a `-w`, `-W` nor `-f` command line option is specified, `-w3` is implicit for DMS output. Only one of these options should be selected.

For cartesian or geographic output the pair of output values are always separated by a tab character and appear in respective (x, y) or (λ, ϕ) order regardless of input data order. Output ordering can be reversed with the `-s` command line switch.

Output of error data

Input data values that cannot not be processed by either basic *proj* input checking procedures or transformation limitations of a particular projection will cause two tab-separated `*` characters to be output (ancillary data will still be printed). For example, a file *bad.data* contains the following geographic coordinates:

```
-10 45 point 1
-7.4 95 point 2
8.3 33 point 3
```

The output of a forward projection is:

```
-786491.58      5033320.60 point 1
*       * point 2
774874.19       3683140.58 point 3
```

A latitude value of 95° is not valid.

The user may change the `*` error flags with the `-e␣`*string* command line option where *string* may be any arbitrary message. Reprocessing the file *bad.data* with the command line:

```
proj +proj=poly -e '# -b #' bad.data
```

will produce as output:

```
-786491.58      5033320.60 point 1
# -b # point 2
774874.19       3683140.58 point 3
```

Binary Input–Output

In many cases, the conversion of input and output ASCII data to internal *proj* binary values may require up to 50 percent of the process execution time when using less computationally intensive cartographic projections. Binary data also require other cooperating programs capable of treating such input-output. In general, the average user is not going to be involved in using this feature of *proj*, but should be aware of it. Respective binary input and output is selected by the `-i` and `-o` command line switches, or the `-b` switch when both input and output are binary. One interesting binary operation, left to the user to try, is:

```
proj +proj=poly my.geog.data -o | invproj +proj=poly -i
```

Ancillary data are not passed through in binary operations, and data ordering and format switches related to binary input or output are ignored.

Cartesian Scaling

The units of cartesian data are determined by the units used to define the Earth's radius or ellipsoidal axes which are typically in meters. Thus, previous examples of cartesian data exhibit large values because of the default Earth parameters employed by *proj*. In some applications, especially map making, it may be more convenient to see the cartesian values in a more comprehendible range. This can be performed by the command line option `-m␣`*scale* where the value the output and input cartesian coordinates are respectively multiplied and divided by *scale*. In addition, if the first two characters of *scale* are `1/` or `1:` the reciprocal value of the remaining number is employed as the scaling factor. This is a convenient feature because most map scales are represented in this manner.

For example, the approximate size of a $7\frac{1}{2}'$ quadrangle at a scale of 1:25,000 at a latitude of 35° can be obtain by:

```
$ proj +proj=poly -m 1:25000 +lat_0=35 -f '%.3f'
-3.75' 35
-0.228  0.000
3.75' 35
0.228   0.000
-3.75' 35d7.5
-0.228  0.555
3.75' 35d7.5
0.228   0.555
```

A latitude origin of 35° was introduced with `+lat_0` to give clearer results. The results show the map to be about 45 by 56 cm.

Cartographic Parameters

In this section the `+` prefixed command line parameters employed for controlling the purely cartographic aspects of *proj* execution will be discussed. As has been shown in previous examples they are in the general form of `+param=`*value* for specifying a value to a projection parameter or `+param` when setting a switch or flag. *Value* may be a geographic or angular coordinate in general DMS format, a real number, an integer number or an ASCII string. In addition, *value* will be used in context with the formularies in the Snyder references.

Two additional aspects of the `+params` should be noted: a mispelled parameter name will be completely ignored and if a parameter is entered twice, only the first occurence will be employed by *proj*. The former condition usually leads to erroneous results and the user should always double check the control data. The second feature is often quite useful: MAPGEN's *mapdef*(1l) program determines the map's central meridian automatically and appends a `+lon_0` entry to the user's *proj* parameter list so that, if the user specifies `+lon_0`, it preceeds and is used in lieu of *mapdef*'s value.

The one remaining hyphen command line option, `-c␣`*confile*, permits entry of all the `+` parameters in an an ASCII file, *confile*, rather than on the command line. When cartographic parameters are both on the command line and in a *confile*, the contents of *confile* are processed after the command line arguments. For example, a file `proj.params` may contain:

```
+proj=tmerc +ellps=GRS80
+x_0=500000 +y_0=-4000000
+lon_0=90w
```

and where usage would be:

```
proj -c proj.params input.data >output.data
```

Alternately,

```
proj -c proj.params +ellps=clrk66 input.data >output.data
```

where the `+ellps` option on the command line preceeds the option in the file *proj.params*. The `+` before the parameters is not required in the *confile*, but is recommended to avoid confusion.

Selection of Projection

As noted before, the selection of the cartographic projection by means of the `+proj=`*name* parameter is required by *proj*. But because projections are continually being added and capablities of existing projections expanded (i.e. adding the inverse operation) it is difficult to keep documentation up to date or properly distributed to all users of a particular version of the program. Conversely, the user may have documentation of a version not yet installed on a particular host machine.

To solve partially the problem of current projection availability and status, the user can get a list of projections from the available version of *proj* by executing:

```
$ proj +proj=list | more
```

An extract of the list that will appear on the user's terminal is:

```
list of projections:

+proj=aea      -> Albers Egual Area FIE
+proj=aeqd     -> Azimuthal equidistant FI
+proj=aitoff   -> Aitoff F
   ...
+proj=vandg    -> Van der Grinten FI
+proj=wink1    -> Winkel I FI
+proj=wintri   -> Winkel Tripel F
```

The last one to three uppercase characters indicate F for forward (always available), I for inverse, and E for elliptical projection capability. This also provides a quick remedy for a memory lapse regarding the *name* of a particular projection.

Specifying the Earth's Figure

For many applications, especially in small scale mapping, the Earth can be considered a sphere and for appropriately scaled output the user need only specify its radius. But due to the rotation on its axis, the Earth is slightly flattened and it is better approximated by an ellipsoid of rotation about the polar axis, thus ellipsoidal projections are used for accurate, large scale maps and plane coordinate systems. To specify the Earth's elliptical figure constants to be used by *proj*, two parameters will be required. The first and required value `+a=`a where a is the semimajor axis of the ellipse or equitorial radius and the second parameter can be in any one of the following standard forms:

- semiminor axis or polar radius `+b=`b,
- flattening `+f=`f,
- reciprocal flattening, `+rf=`$1/f$,
- eccentricity `+e=`e, or
- eccentricity squared `+es=`e^2.

An example of specifying the Clark 1866 elliptical constants for the Polyconic projection would be:

```
proj +proj=poly +a=6378206.4 +es=.006768658
```

Program *proj* has a convenient method of specifying standard ellisoidal constants by means of the `+ellps=`*ellipse* parameter. A list of *proj ellipse* names currently available along with their exanded names and coefficients can be obtained by executing

```
$ proj +ellps=list
```

which will output the following extract on the user's terminal:

```
list of standard ellipsoid constants (+ellps=  )
       default selection:clrk66

     MERIT: a=6378137.0      rf=298.257       "MERIT 1983"
     GRS80: a=6378137.0      rf=298.257222    "GRS 1980(IUGG, 1980)"
     ...
    clrk66: a=6378206.4      b=6356583.8      "Clarke 1866"
     ...
     WGS72: a=6378135.0      b=6356750.519915 "WGS 72"
    sphere: a=6370997.0      es=0.0           "Sphere of 6370997 m"
```

When *proj* is executed without specification of elliptical constants (as per examples in the previous sections) a default ellipse, specified on the second line of the `+ellps=list` option output, will be used. This default is determined at system generation and Clarke's 1866 constants are currently selected for use in North American applications. GRS80 will eventually become a default setting.

In the case of spherical projections or usage of the elliptical projections in spherical form, specifying `+a` without one of the second ellipsoid parameters, such as `+rf` or `+es`, forces a spherical transformation. In this case, `+a` determines the Earth's radius R. For example, to use the spherical form of the Polyconic projection

```
proj +proj=poly +a=1
```

will create forward projection cartesian output for a unit sphere.

The user is responsible for maintaining consistence of the units of a (R) (and, if employed, b) with the expected cartesian forward projection output or cartesian input for inverse projections. Some options to be discussed later must also be in the same units.

Common Parameters

Three parameters are common to all projections: the central meridian `+lon_0=`λ_0, and the cartesian offsets for the respective x and y axes, `+x_0=`x_0, `+y_0=`y_0 (often refered to as false easting and northing). The central meridian is a simple translation of the longitude axis which is normaly used to center a projection at a particular longitude. Unless noted, the projection's x axis origin is at λ_0. The cartesian offsets are usually used in plane coordinate systems so that cartesian coordinates will be positive:

$$\begin{aligned} x &= x' + x_0 \\ y &= y' + y_0 \end{aligned}$$

where (x', y') are the internal projection cartesian coordinates and (x, y) are cartesian input–output of *proj*. A fourth parameter, `lat_0=`ϕ_0, is used to designate a central parallel and associated y axis origin for several projections. Unless the user specifies a value for these parameters, they all assume a zero value.

There has been recent need to employ geocentric latitude, ϕ_g, rather than geographic (or geodetic) latitude where the two latitudes are related by:

$$\tan \phi_g = (1 - e^2) \tan \phi$$

Usage of the switch `+geoc` will cause *proj* to treat input or output latitude as geocentric. This parameter only applies to ellipsoidal projection usage and is ignored for spherical usage ($e^2 = 0$).

The normal geographic range for longitude is from 180° W to 180° E and similarly *proj* reduces, both in forward and inverse mode, the internal value of $\Delta\lambda$ to this range. Use of the option `+over` suppresses this range reduction but care should be taken in applications. Generally, this option is employed in the construction of world maps.

Projection descriptions

The following description sections of *proj*'s projection functions are divided into the catagories of cylindrical, pseudocylindrical, conic, azimuthal and miscellaneous. Each projection is described as to its classification and subclassification, aliases, available computational forms (i.e. elliptical, spherical, forward and/or inverse) and summary of usage options. Most projections will also have an example plot of the projection with parenthetical entries in the captions specifying options used to generate the graphic.

In some cases the aliases apply to names given special forms of the projection. For example, the Werner projection which is a special case of the Bonne projection is listed as an alias of the Bonne projection. The usage description does not list the options common to all projections discussed earlier such as the Earth's figure parameters and cartesian offsets.

Cylindrical Projections

Cylindrical projections are based upon the various methods of projecting the Earth upon a cylinder that is either tangent to the equator (normal or equatorial form), a meridian (transverse) or obliquely aligned. Any of these classes are available in both conformal and equal area form. These projections are best used in mapping applications involving a zone near the line of tangency.

Mercator Projection

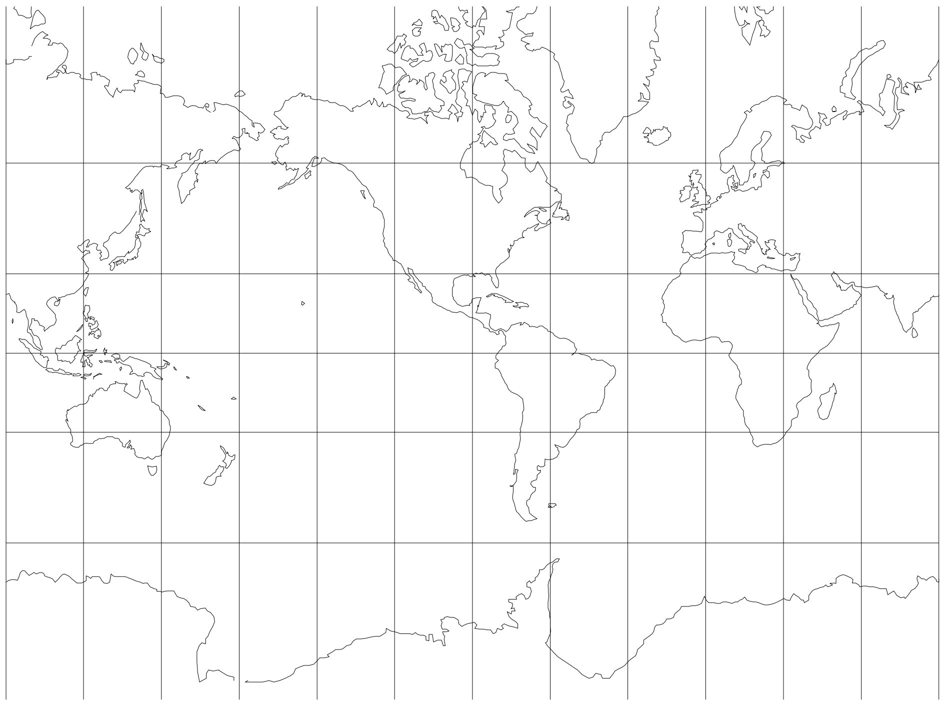

Figure 1: Mercator projection, with shorelines and 30° graticule. Central meridian 90° W (`+proj=merc +lon_0=90w`).

Classifications: Conformal cylindrical.

Aliases: Wright (rare).

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=merc +lat_ts=`ϕ_s

Applications should be limited to equatorial regions, but it is frequently used for navigational charts with latitude of true scale (ϕ_s) specified within or near the chart's boundaries. Often inappropriately used for world maps since the regions near the poles cannot be shown. When not specified, $\phi_s = 0°$.

Transverse Mercator Projection

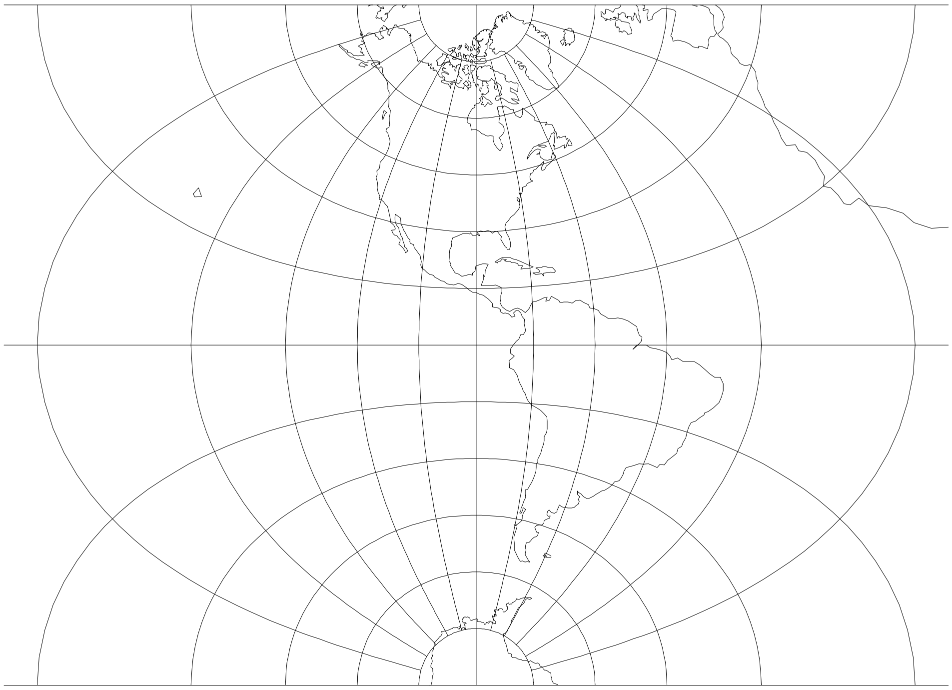

Figure 2: Transverse Mercator projection, Western hemisphere with shorelines and 15° graticule. Central meridian 90° W (`+proj=tmerc +lon_0=90w`).

Classifications: Transverse cylindrical. Conformal.

Aliases: Gauss Conformal (ellipsoidal form), Gauss-Krüger (ellipsoidal form), Transverse Cylindrical Orthomorphic

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=tmerc +lat_0=`ϕ_0 `+k=`k_0

This is a common projection for large scale maps of predominantly north-south extent. The parameter k_0 is the scale factor at the central meridian which has a value of 1. when `+k` is not specified.

Universal Transverse Mercator (UTM) Projection

Usage and options: `+proj=utm +south +zone=`*zone*

The UTM projection is a special ellipsoidal form of the general Transverse Mercator projection where $\lambda - \lambda_0 \leq \pm 4°$ and $84°\mathrm{S} \leq \phi \leq 80°\mathrm{N}$. The central meridian, λ_0, is constrained to 6° intervals starting at 3° and the projection automatically adjusts to the proper value nearest the user's entry of λ_0. An even multiple of 6° for λ_0 should be avoided since the intended zone is indeterminent. One of the 60 *zone* numbers may be specified in lieu of λ_0 where `+zone=1` specifies the region from 180°W to 174°W ($\lambda_0 = 177°$W) and proceeds easterly until `+zone=60` for the region from 174°E to 180°E ($\lambda_0 = 177°$E). If both `+zone` and `+lon_0` are used, `+zone` takes precedence.

For Southern hemisphere applications the option `+south` should be used which adds a false northing of 10,000,000m. In all cases, a false easting of 500,000m is used. Also see Universal Polar Stereographic (UPS), p. 40.

Oblique Mercator Projection

Classifications: Conformal cylindrical.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=omerc` `k=`k_0 `+lat_0=`ϕ_0 `+no_rot` and either `+lon_1=`λ_1 `+lat_1=`ϕ_1 `+lon_2=`λ_2 `+lat_2=`ϕ_2 or `+alpha=`α_c `+lonc=`λ_c

The Oblique Mercator (not illustrated) has two methods of specifying its control information:

1. by means of two points (λ_1, ϕ_1) and (λ_2, ϕ_2) which will determine a great circle, central line through each point or
2. by means of a point of origin at (λ_c, ϕ_0) and an azimuth α_c, measured clockwise from north, of the central line of the projection.

The presence of the `+alpha` option determines the method to be used. Cartesian origin of the projection will coincide with ϕ_0 and an internally determined value of longitude for the first method or λ_c in the second method (`+lon_0=`λ_0 is not used and is ignored if specified). The cartesian coordinates are rotated by $-\alpha_c$ (determined internally for the first method) unless the `+no_rot` options is specified. Scale factor at the central line can be modified by k_0 which is 1. if not specified with `+k`.

Initialization of the projection will fail if control parameters nearly define a transverse or normal (equatorial) Mercator projection.

Central Cylindrical Projection

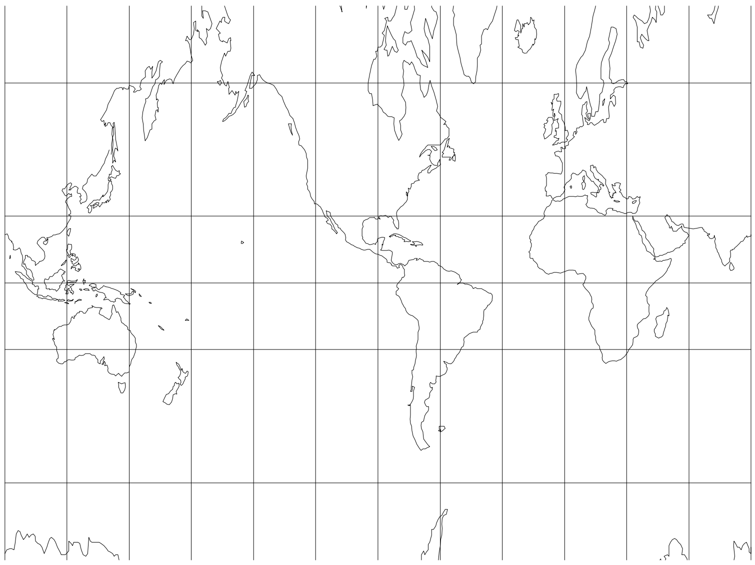

Figure 3: Central Cylindrical projection, with shorelines and 30° graticule. Central meridian 90° W (`+proj=cc +lon_0=90w`).

Classifications: Cylindrical. Perspective, neither conformal nor equal area.

Aliases: Simple Perspective Cylindrical

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=cc`

Transverse Central Cylindrical Projection

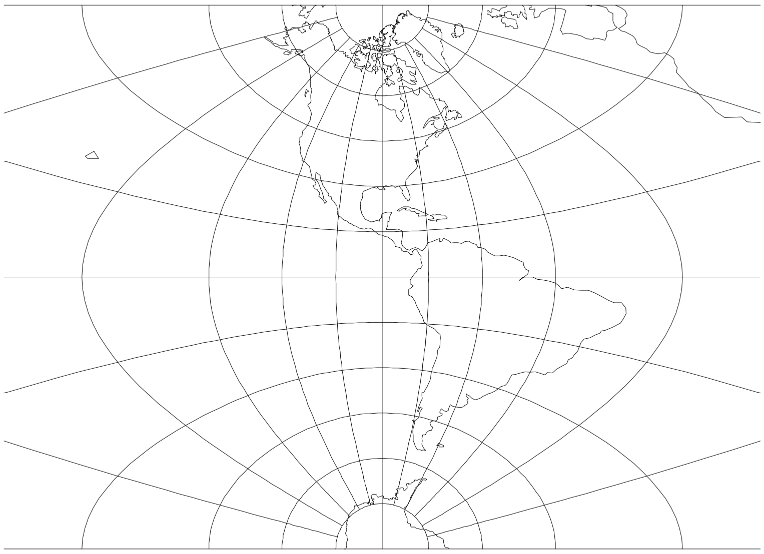

Figure 4: Transverse Central Cylindrical projection, Western hemisphere with shorelines and 15° graticule. Central meridian 90° W (`+proj=tcc +lon_0=90w`).

Classifications: Perspective cylindrical. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=tcc`

Miller Projection

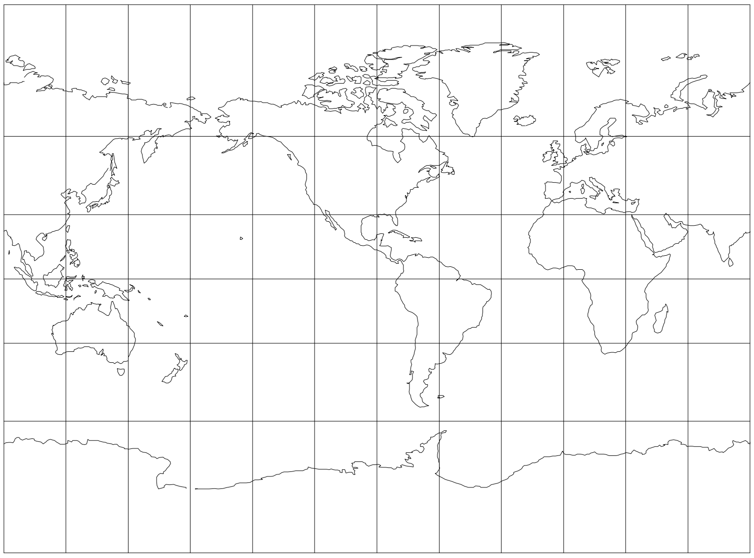

Figure 5: Miller projection, with shorelines and 30° graticule. Central meridian 90° W (`+proj=mill +lon_0=90w`).

Classifications: Cylindrical. Neither conformal nor equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=mill`

Lambert Cylindrical Equal Area Projection

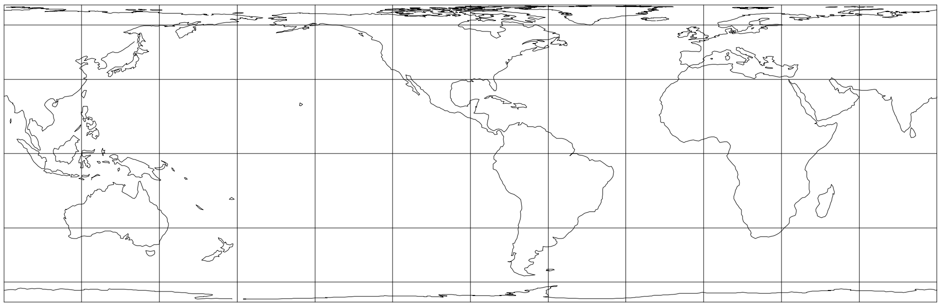

Figure 6: Lambert Cylindrical Equal Area projection, with shorelines and 30° graticule. Central meridian 90° W. Standard parallel 0° (`+proj=cea +lon_0=90w`).

Classifications: Cylindrical. Equal area.

Aliases: Cylindrical Equal Area, Behrmann, Gall Orthographic, Peters.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=cea +lat_ts=`ϕ_s

The latitude of true scale or standard parallels, ϕ_s, is assumed to be 0° if not specified by `lat_ts` (figure 6). Two alternate projections based upon the Cylindrical Equal area are Behrmann (figure 7) where $\phi_s = 30$ and Gall's Orthographic (figure 8) where $\phi_s = 45$. The latter is also an approximation of the Peters projection.

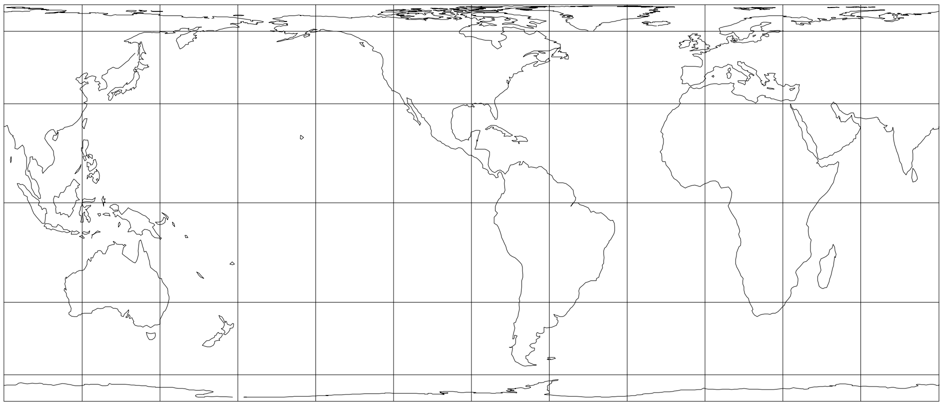

Figure 7: Behrmann Equal Area projection, with shorelines and 30° graticule. Central meridian 90° W. Standard parallels 30° (`+proj=cea +lon_0=90w +lat_ts=30`).

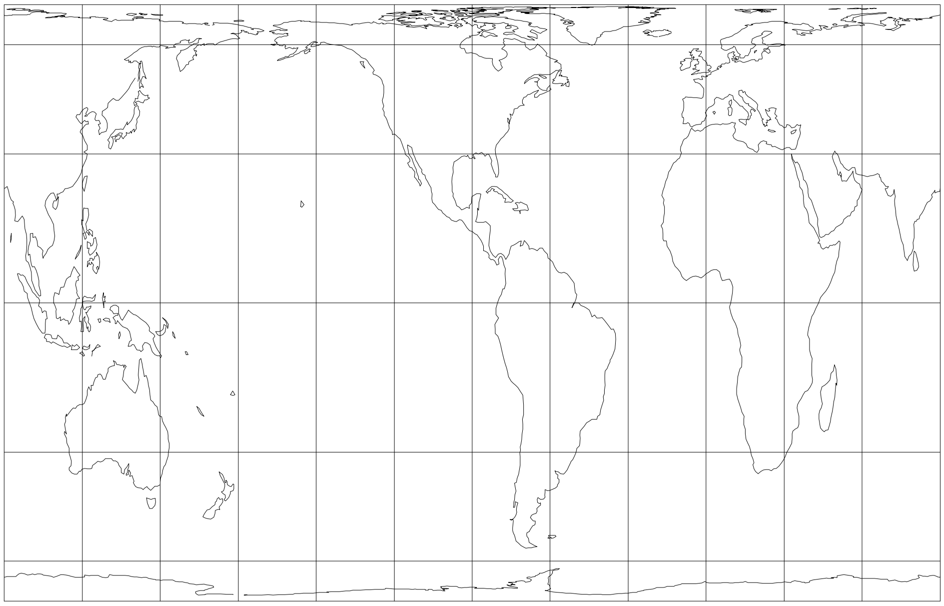

Figure 8: Gall's Orthographic or approximate Peters projection, with shorelines and 30° graticule. Central meridian 90° W. Standard parallels 45° (`+proj=cea +lon_0=90w +lat_ts=45`).

Gall (Stereographic) Projection

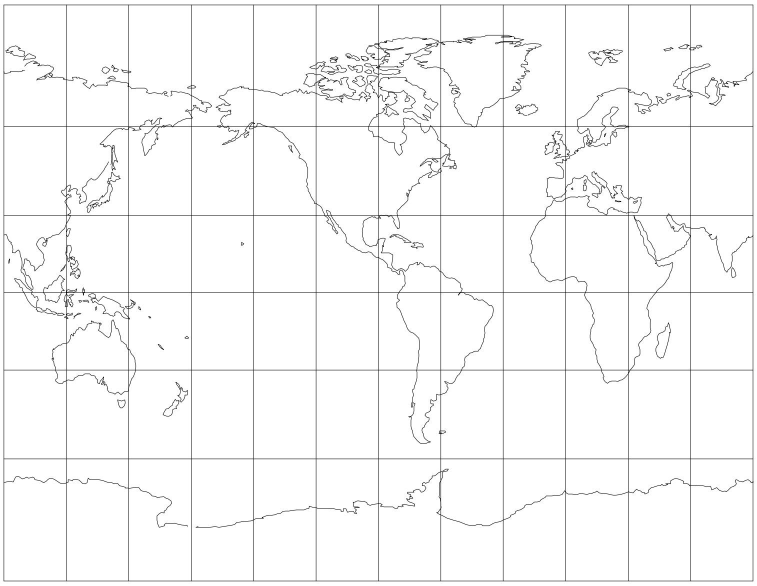

Figure 9: Gall (Stereographic) projection, with shorelines and 30° graticule. Central meridian 90° W (`+proj=gall +lon_0=90w`).

Classifications: Cylindrical. Neither conformal nor equal area.

Aliases: Gall Stereographic.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=gall`

Transverse Cylindrical Equal Area Projection

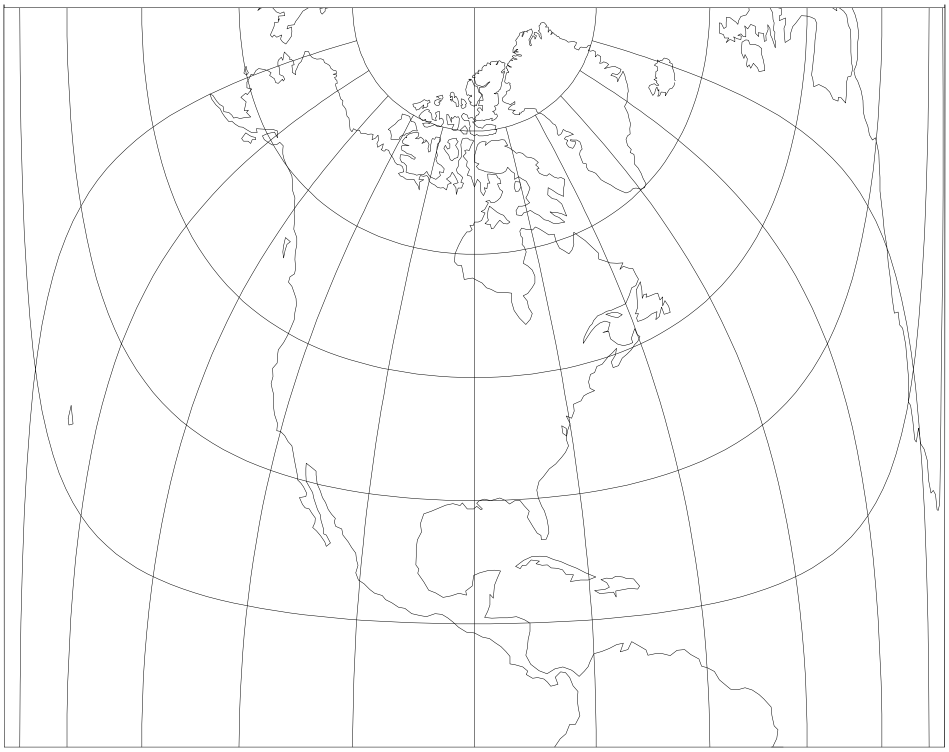

Figure 10: Transverse Cylindrical Equal Area projection, Western hemisphere with shorelines and 15° graticule. Central meridian 90° W (`+proj=tcea +lon_0=90w`).

Classifications: Cylindrical. Equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=tcea +lat_0=` ϕ_0 `+k=`k_0

The parameter k_0 is the scale factor at the central meridian which has a value of 1. when `+k` is not specified.

Equidistant Cylindrical Projection

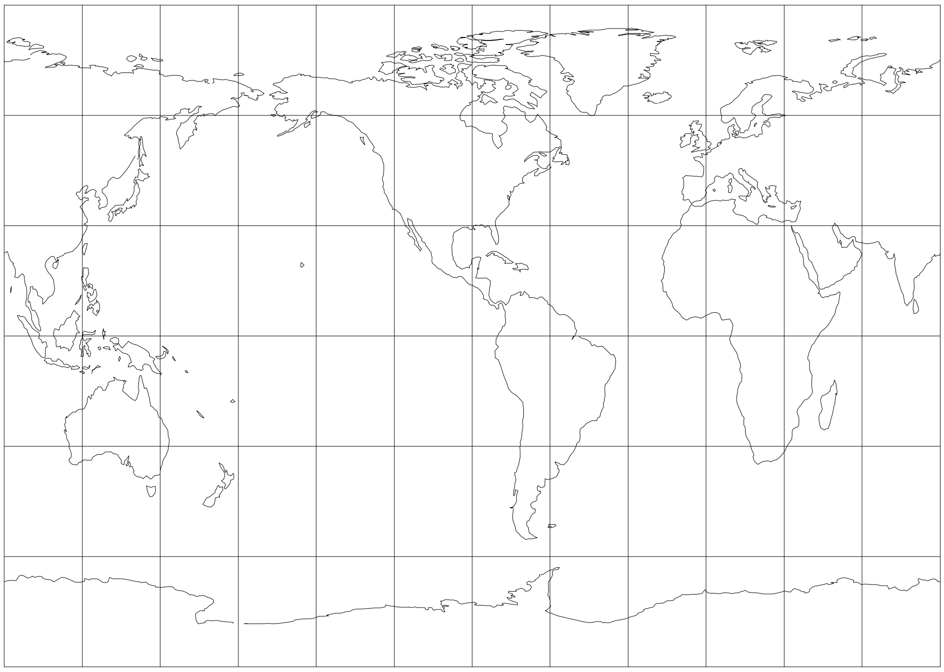

Figure 11: Equidistant Cylindrical projection, with shorelines and 30° graticule. Central meridian 90° W and $\phi_s = 45°$ (`+proj=eqc +lon_0=90w`).

Classifications: Cylindrical. Neither conformal nor equal area.
Aliases: Plate Carée.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=eqc +lat_ts=`ϕ_s

The latitude of true scale, ϕ_s, is assumed to be 0° if not specified by `lat_ts` (figure 11). The Plate Carée designation only applies when $\phi_s = 0$ (figure 12).

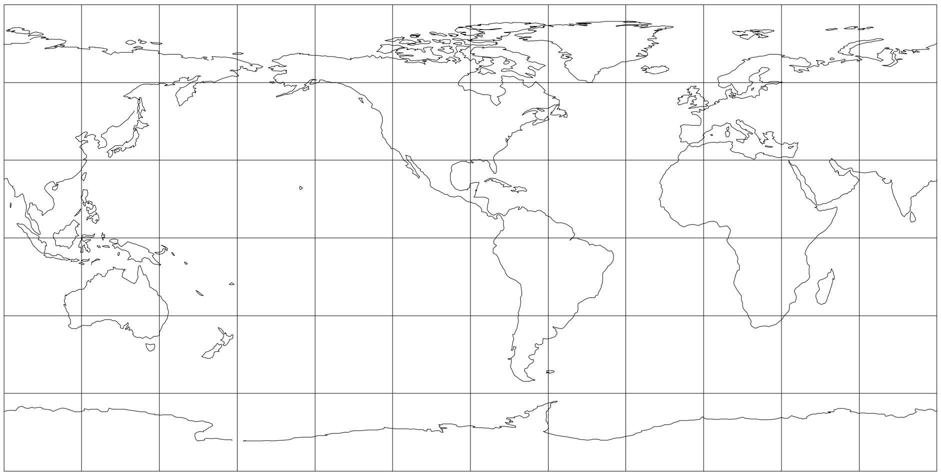

Figure 12: Plate Carée projection, with shorelines and 30° graticule. Central meridian 90° W (`+proj=eqc +lon_0=90w +lat_ts=30`).

Cassini Projection

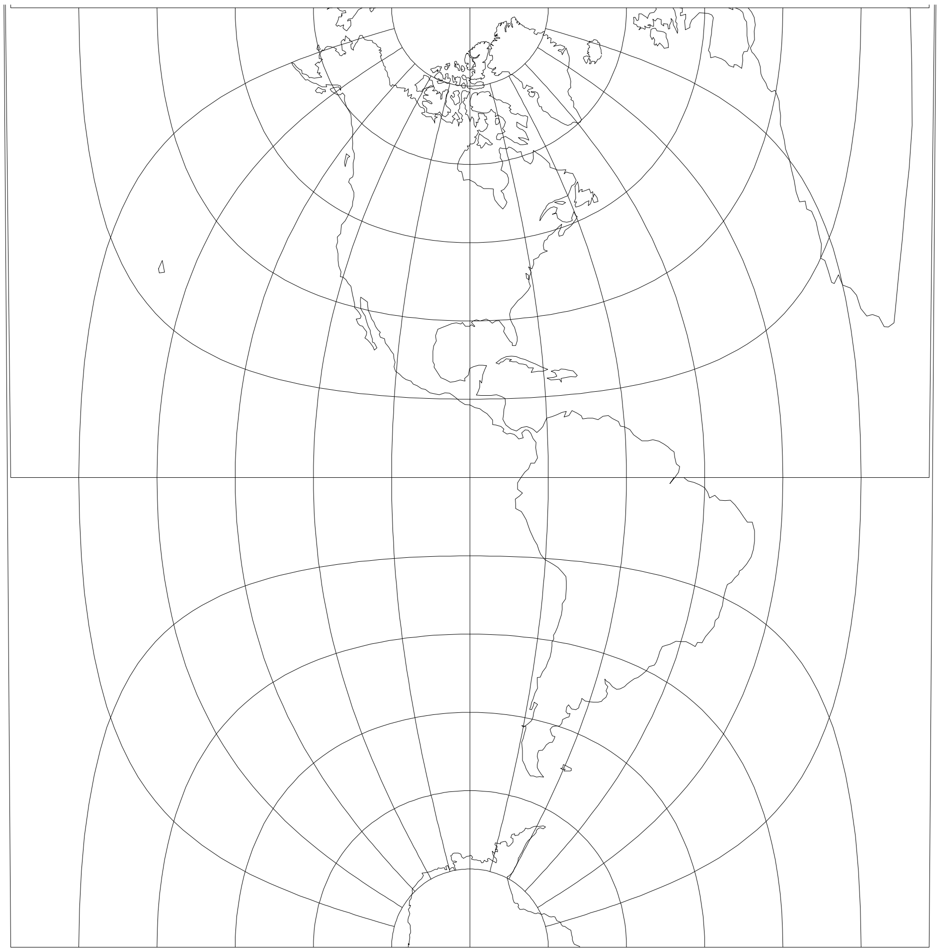

Figure 13: Cassini projection, Western hemisphere with shorelines and 15° graticule. Central meridian 90° W (`+proj=cass +lon_0=90w`).

Classifications: Transverse cylindrical. Neither conformal nor equal area.
Available forms: Forward and inverse, spherical and elliptical projection.
Usage and options: `+proj=cass +lat_0=`ϕ_0

Transverse form of the Plate Carée projection.

Pseudocylindrical Projections

The defining quality of the pseudocylindrical projections is that the parallels of latitude are parallel, as with the normal cylindrical projections, but meridians, other than the central meridian, always converge to the polar ends of the central meridian. Typically, the convergence of the meridians is a smooth arc but some novelty projections (Eckert I and II and Collignon) use straight lines. Several of the pseudocylindical projections equal area but none are conformal.

The principle usage of these projections is for small scale, global maps, and frequently in an interupted form.

Sinusoidal Projection

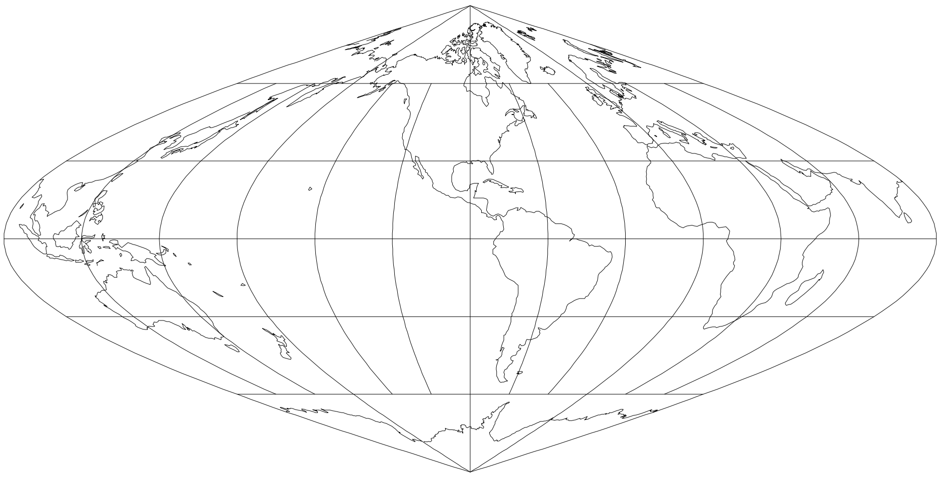

Figure 14: Sinusoidal projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=sinu +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Aliases: Sanson-Flamsteed, Mercator Equal-Area.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=sinu`

Because this projection is useful for large scale applications it is frequently employed in transverse and oblique aspects.

Mollweide Projection

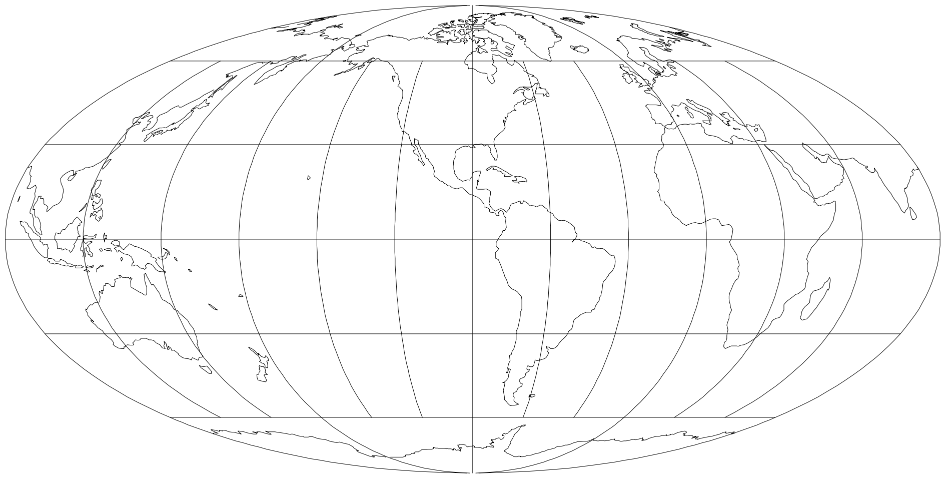

Figure 15: Mollweide projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=moll +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Aliases: Homolographic, Homalographic, Babinet, Elliptical.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=moll`

Robinson Projection

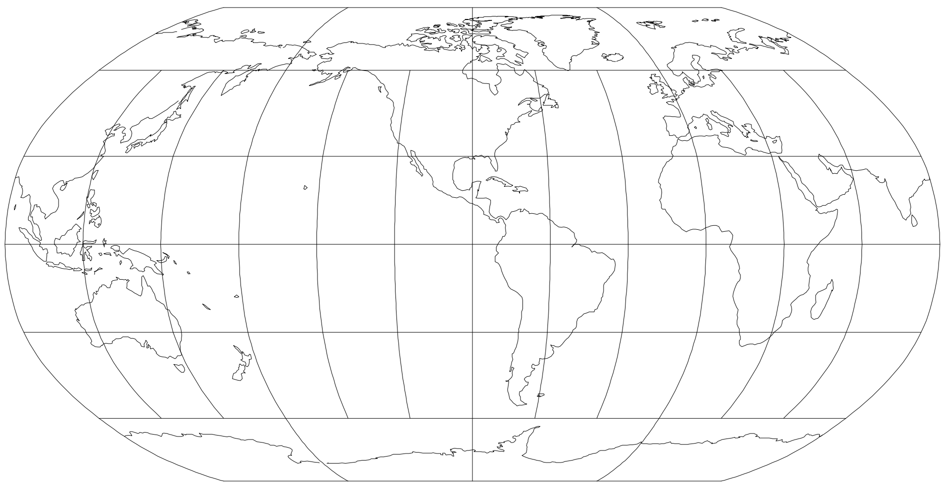

Figure 16: Robinson projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=robin +lon_0=90w`).

Classifications: Miscellaneous conformal.

Aliases: Orthophanic.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=robin`

Eckert I Projection

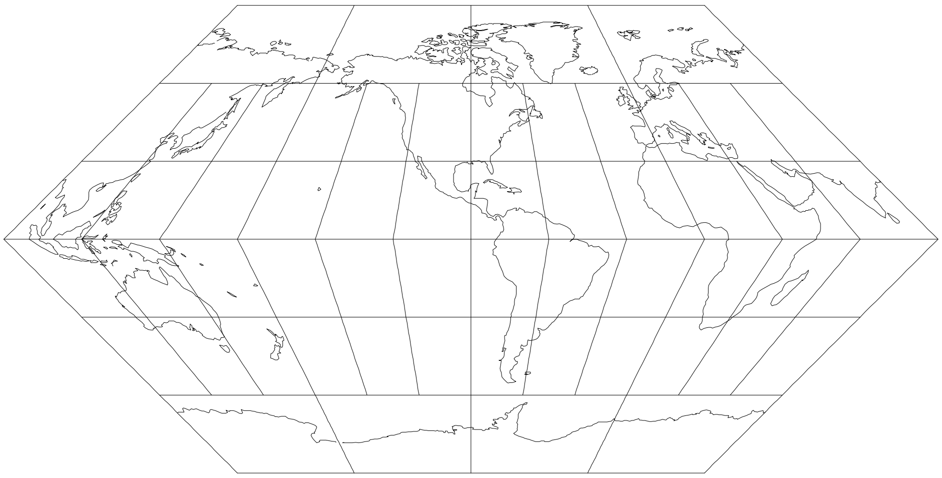

Figure 17: Eckert I projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck1 +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck1`

Eckert II Projection

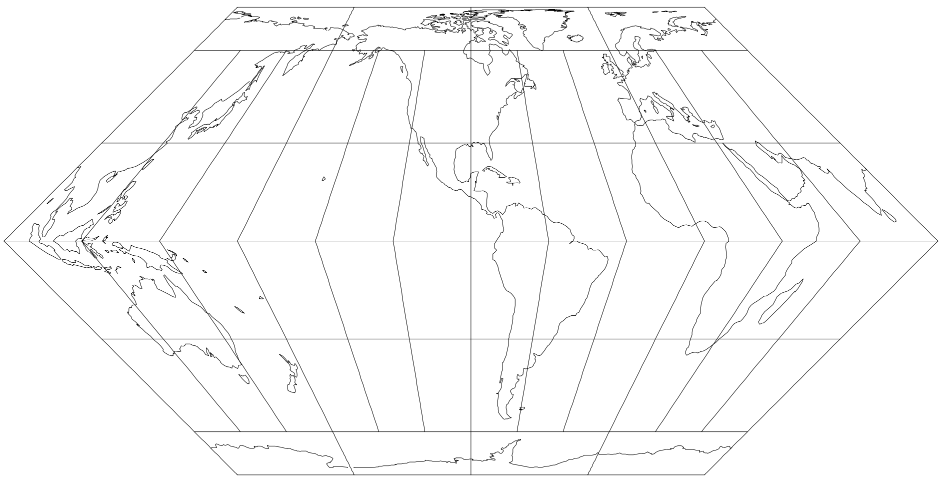

Figure 18: Eckert II projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck2 +lon_0=90w`).

Classifications: Pseudocylindrical, equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck2`

Eckert III Projection

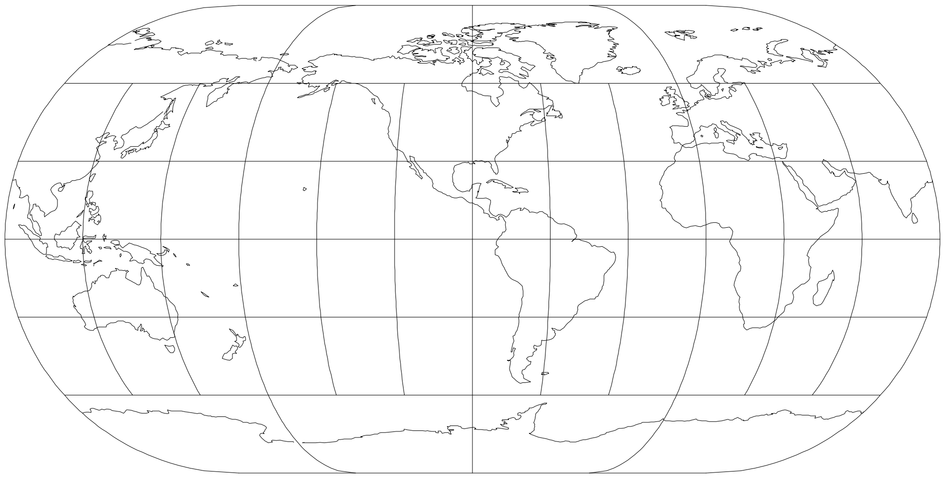

Figure 19: Eckert III projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck3 +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck3`

Eckert IV Projection

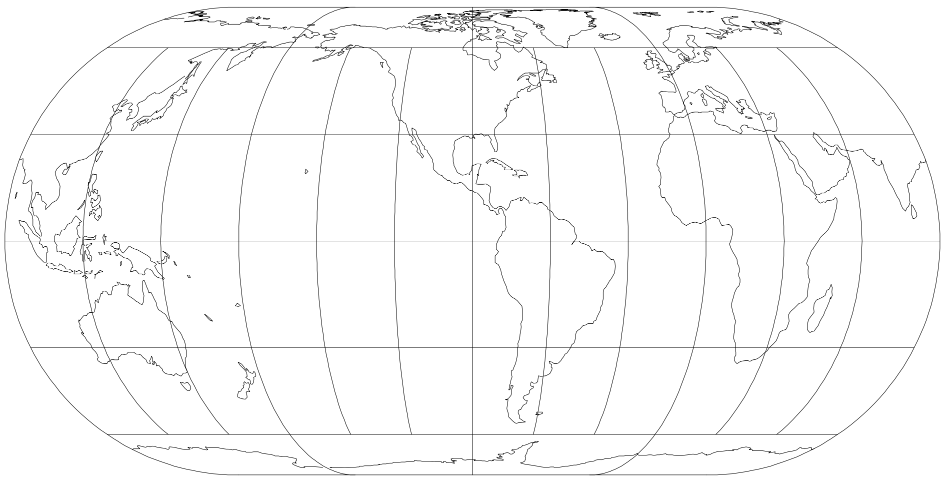

Figure 20: Eckert IV projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck4 +lon_0=90w`).

Classifications: Pseudocylindrical, equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck4`

Eckert V Projection

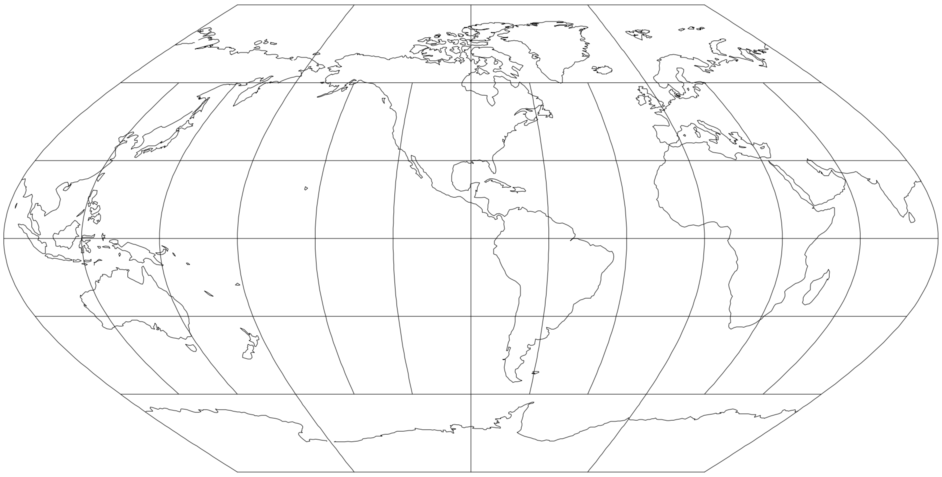

Figure 21: Eckert V projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck5 +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck5`

Eckert VI Projection

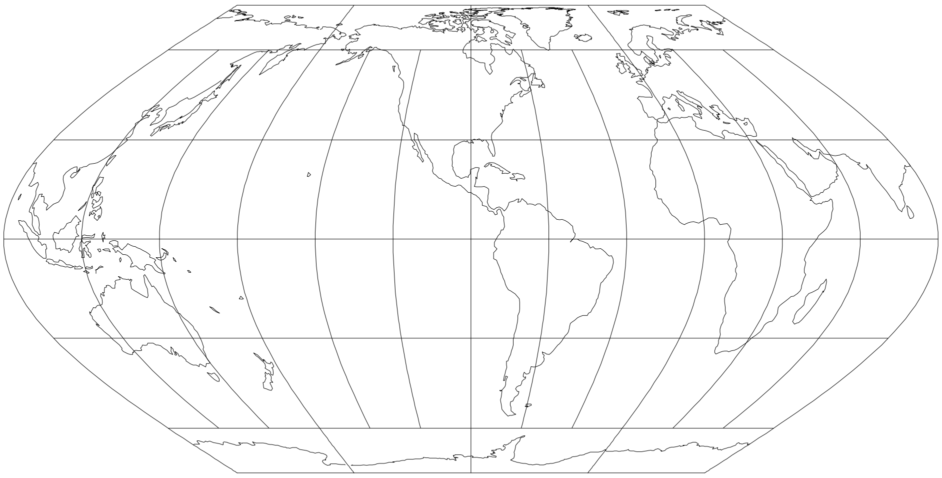

Figure 22: Eckert VI projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eck6 +lon_0=90w`).

Classifications: Pseudocylindrical, equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=eck6`

Goode Homolosine Projection

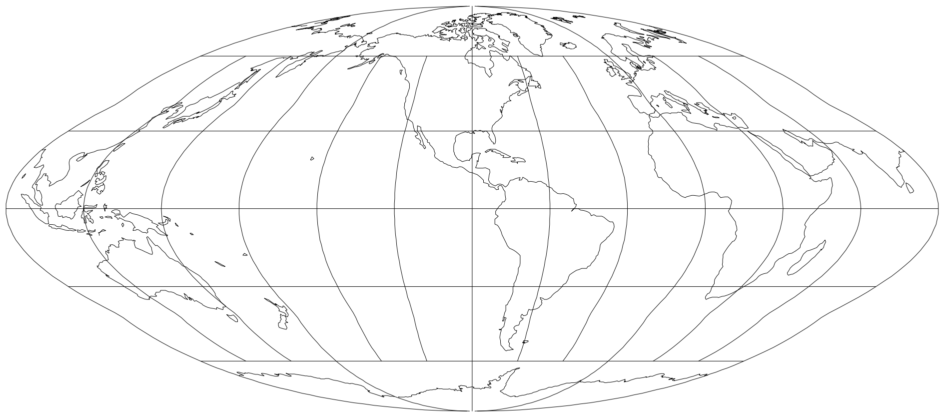

Figure 23: Goode Homolosine projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=goode +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Aliases: Homolosine.

Available forms: Forward spherical projection.

Usage and options: `+proj=goode`

This projection is often used in interupted form and it is a composite projection using Sinusoidal for $|\phi| < 40°44'$ and Mollweide for $|\phi| \geq 40°44'$.

Hatano Asymmetrical Equal-Area Projection

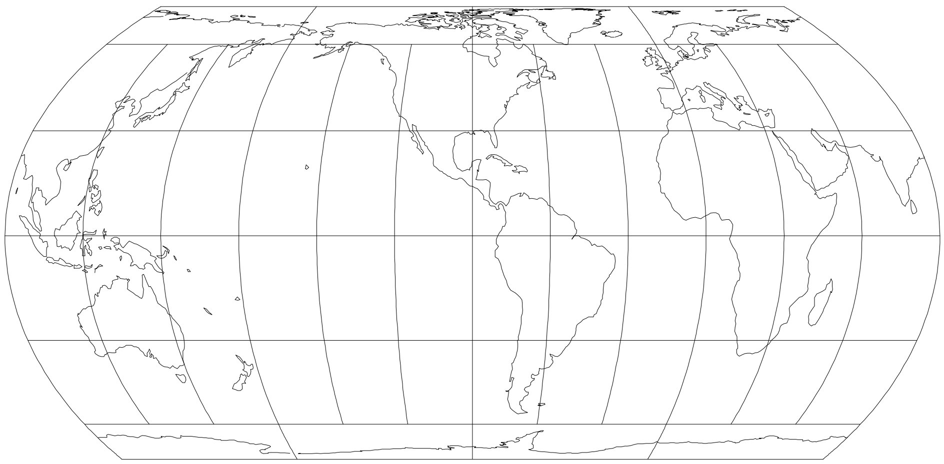

Figure 24: Hatano Asymmetrical Equal-Area projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=hataea +lon_0=90w`).

Classifications: Pseudocylindrical, equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=hataea`

Loximuthal Projection

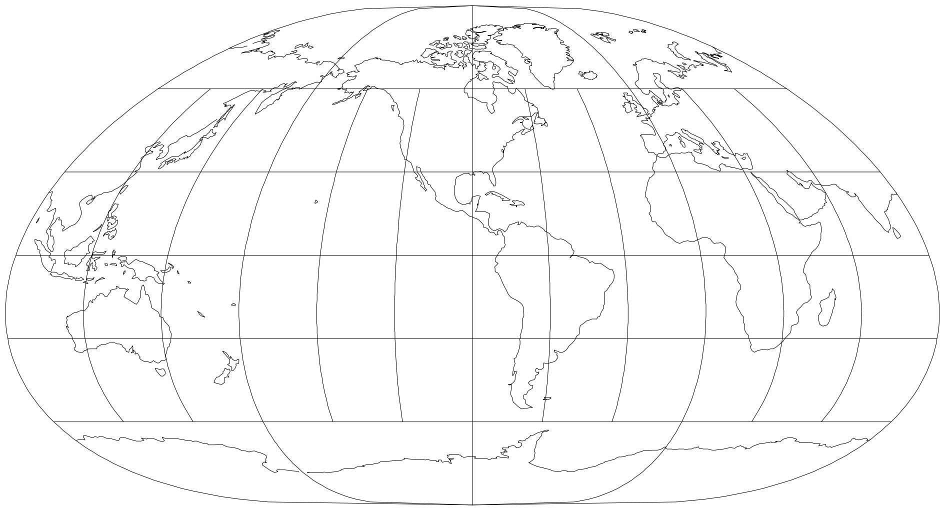

Figure 25: Loximuthal projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=loxim +lon_0=90w +lat_1=40n`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.

Available forms: Forward spherical projection.

Usage and options: `+proj=loxim +lat_1=`ϕ_1

Loxodromes (rhumb lines) from the central point, (λ_0, ϕ_1), are true to scale from this point and have proper azimuth at this point.

McBryde-Thomas Flat-Polar Parabolic Projection

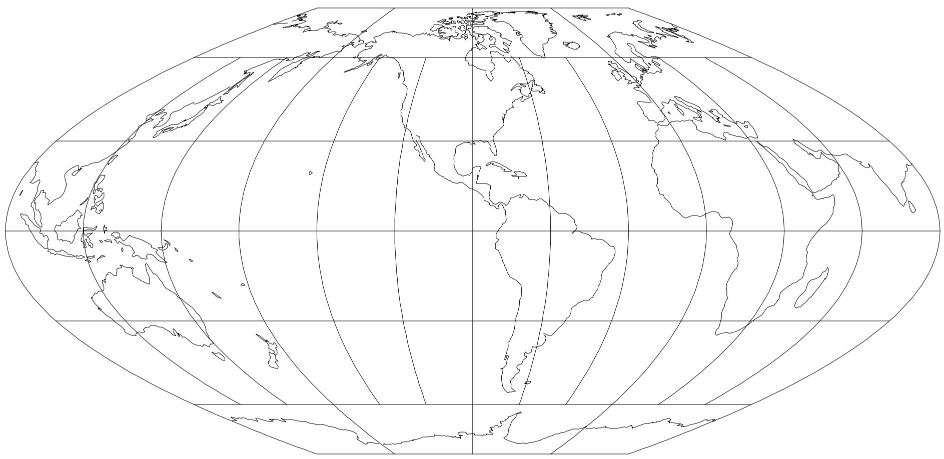

Figure 26: McBryde-Thomas Flat-Polar Parabolic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=mbtfpp +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=mbtfpp`

McBryde-Thomas Flat-Polar Quartic Projection

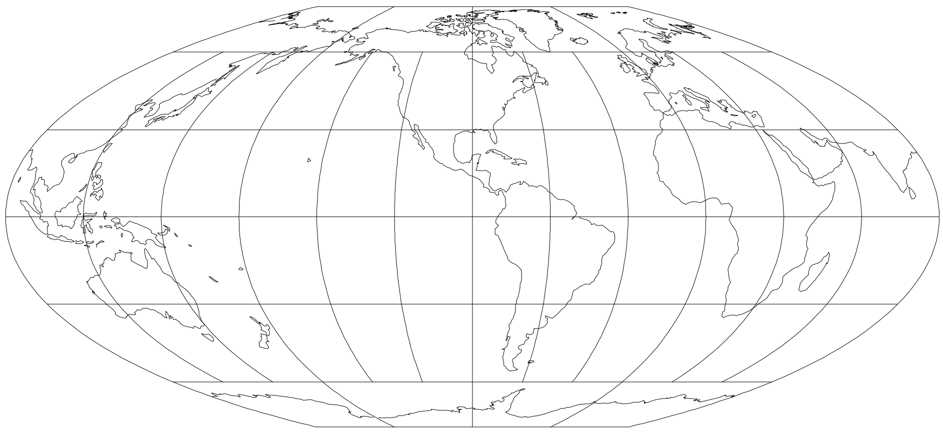

Figure 27: McBryde-Thomas Flat-Polar Quartic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=mbtfpq +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Aliases: Flat-Polar Quartic.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=mbtfpq`

McBryde-Thomas Flat-Polar Sinusoidal Projection

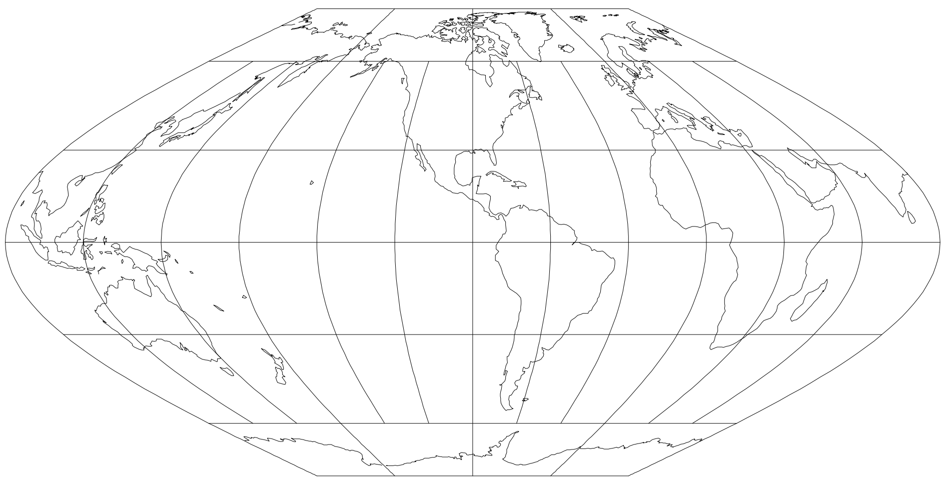

Figure 28: McBryde-Thomas Flat-Polar Sinusoidal projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=mbtfps +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=mbtfps`

Putniņš P_2' Projection

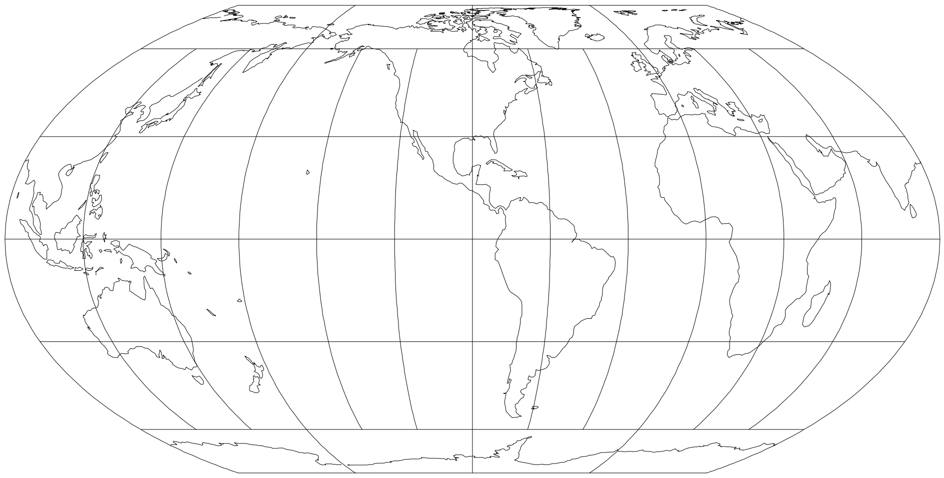

Figure 29: Putniņš P_2' projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=putp2 +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=putp2`

Putniņš P_5 Projection

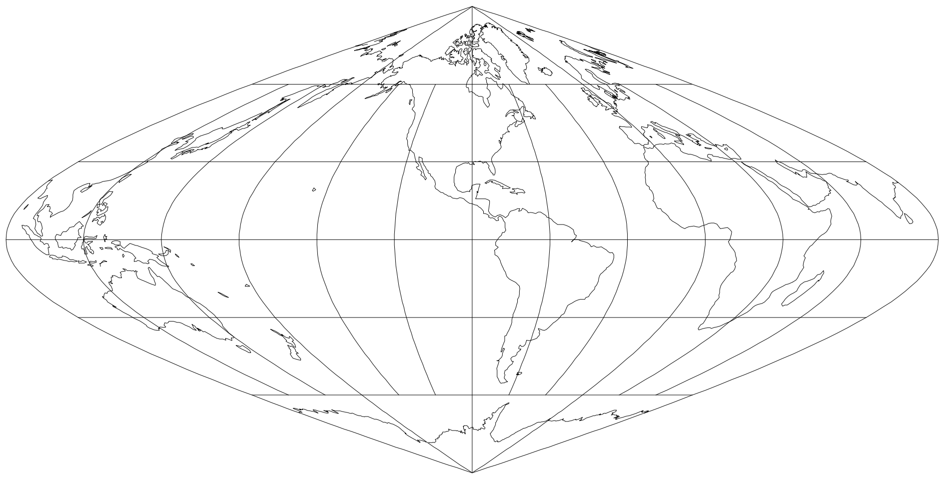

Figure 30: Putniņš P_5 projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=putp5 +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=putp5`

Quartic Authalic Projection

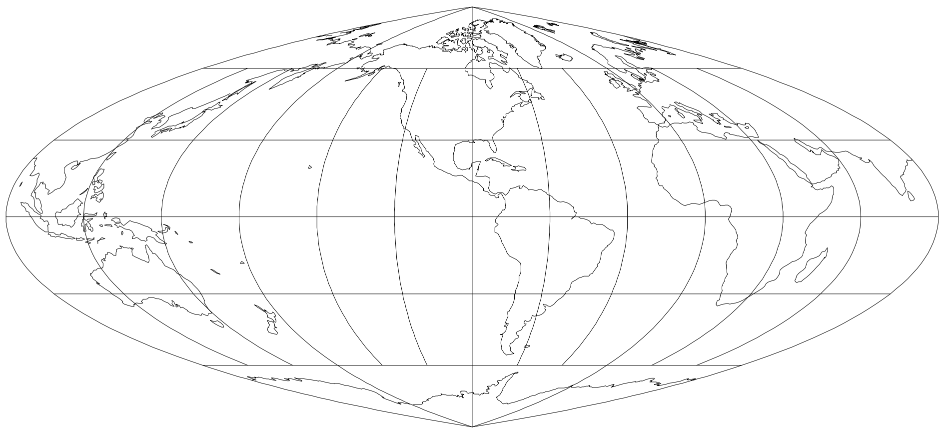

Figure 31: Quartic Authalic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=quau +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=quau`

Winkel I Projection

Figure 32: Winkel I projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=wink1 +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=wink1`

Boggs Eumorphic Projection

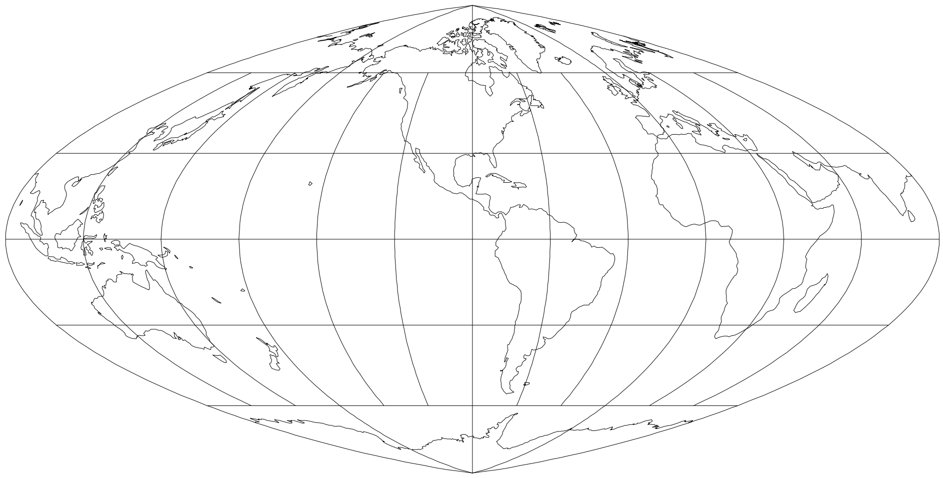

Figure 33: Boggs Eumorphic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=boggs +lon_0=90w`).

Classifications: Pseudocylindrical, equal area.
Available forms: Forward spherical projection.
Usage and options: `+proj=boggs`

Collignon Projection

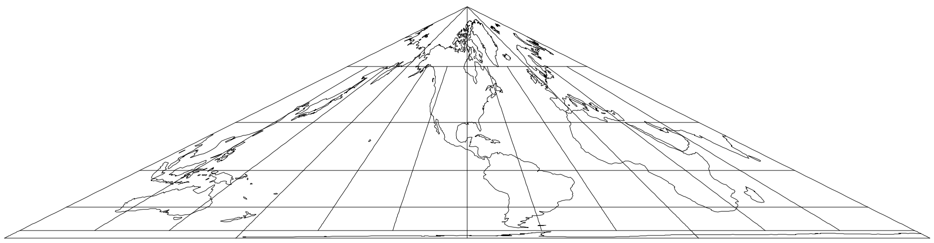

Figure 34: Collignon projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=collg +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=collg`

Denoyer Semi-Elliptical Projection

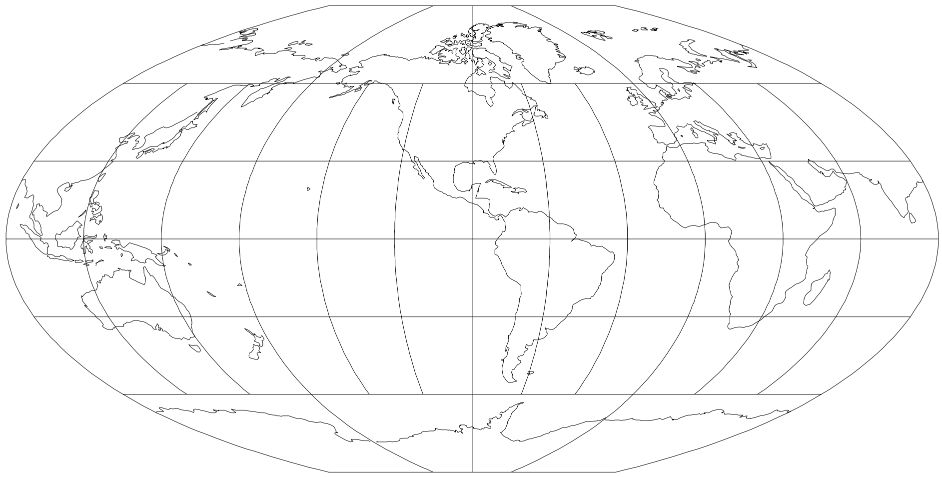

Figure 35: Denoyer Semi-Elliptical projection approximation, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=dense +lon_0=90w`).

Classifications: Pseudocylindrical. Neither conformal nor equal area.

Available forms: Forward spherical projection.

Usage and options: `+proj=dense`

Craster Parabolic Projection

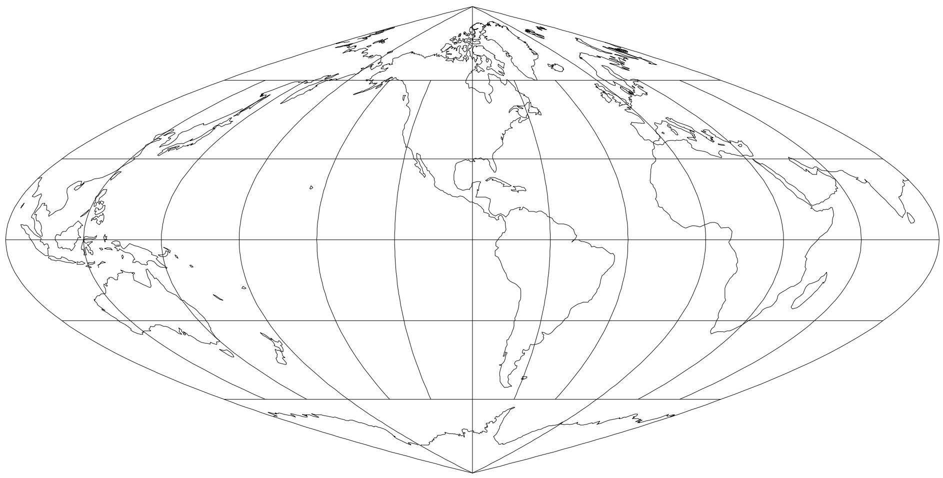

Figure 36: Craster Parabolic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=parab +lon_0=90w`).

Classifications: Pseudocylindrical. Equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=parab`

Conic Projections

Basic conic projections involve the transformations to a cone either secant or tangent to the Earth's surface. Specification of the latitudes of secant intersection are made with the `lat_1=`ϕ_1 and `lat_2=`ϕ_2 parameters ($\phi_1 = \phi_2$ for tangent form). Cylindrical ($\phi_1 = -\phi_2$) or azimuthal ($\phi_1 = \phi_2 = 90°$) limiting forms should not be performed by these projections.

Lambert Conformal Conic Projection

Figure 37: Lambert Conformal Conic projection, with shorelines and 30° graticule. Central Meridian 90° W. and standard parallels at 20° N and 60° N (`+proj=lcc +lon_0=90w +lat_1=20n +lat_2=60n`).

Classifications: Conformal conic.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=lcc +lat_1=`ϕ_1 `+lat_2=`ϕ_2 `+lat_0=`ϕ_0

Default values for ϕ_1 and ϕ_2 are respectively 33° N and 45° N (values normally used for maps of the conterminous United States). Limiting forms are Polar Stereographic and Mercator.

Equidistant Conic Projection

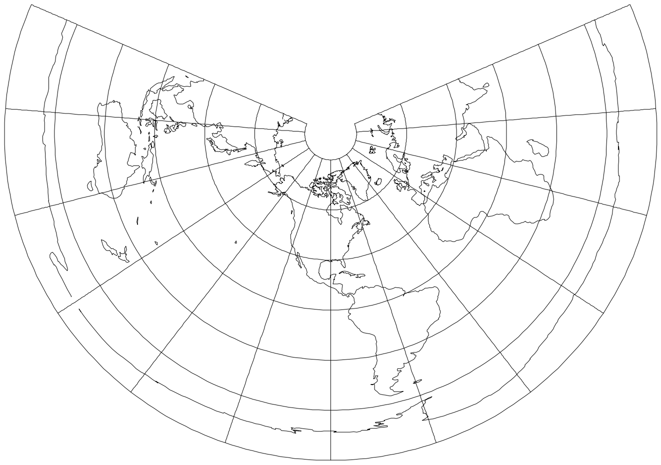

Figure 38: Equidistant Conic projection, with shorelines and 30° graticule. Central Meridian 90° W. and standard parallels at 20° N and 60° N (`+proj=eqdc +lon_0=90w +lat_1=20n +lat_2=60n`).

Classifications: Conic. Neither conformal nor equal area.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=eqdc +lat_1=`ϕ_1 `+lat_2=`ϕ_2 `+lat_0=`ϕ_0

Default values for ϕ_1 and ϕ_2 are respectively $29\frac{1}{2}^\circ$ N and $45\frac{1}{2}^\circ$ N. Limiting forms are Polar Azimuthal Equidistant and Equirectangular.

Perspective Conic Projection

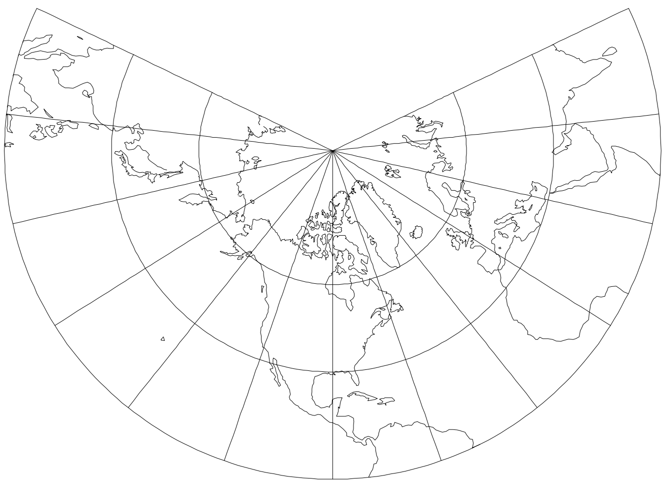

Figure 39: Perspective Conic projection, with shorelines and 30° graticule. Central Meridian 90° W and standard parallels at 20° N and 60° N (`+proj=pconic +lon_0=90w +lat_1=20n +lat_2=60n`).

Classifications: Conic. Neither conformal nor equal area.

Available forms: Forward, spherical projection.

Usage and options: `+proj=pconic +lat_1=`ϕ_1 `+lat_2=`ϕ_2

Default values for ϕ_1 and ϕ_2 are respectively 33° N and 45° N Limiting forms are Polar Gnomonic and Central Cylindrical.

Albers Equal Area Projection

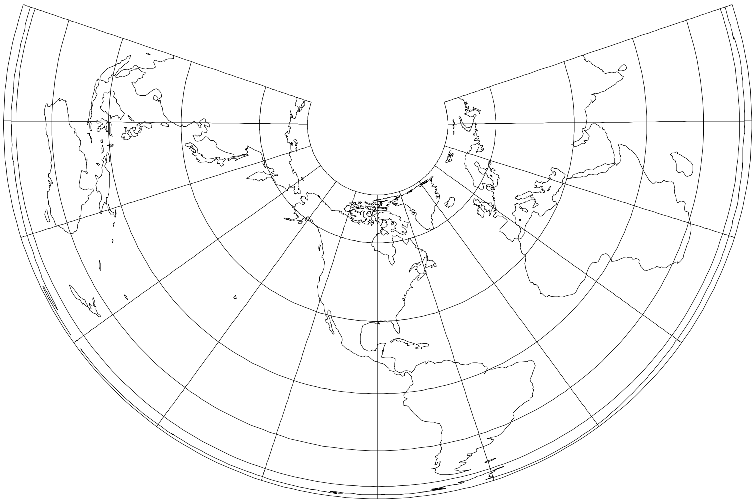

Figure 40: Albers equal area projection, with shorelines and 30° graticule. Central Meridian 90° W. and standard parallels at 20° N and 60° N (`+proj=aea +lon_0=90w +lat_1=20n +lat_2=60n`).

Classifications: Conic. Equal area.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=aea` `+lat_0=`ϕ_0 `+lat_1=`ϕ_1 `+lat_2=`ϕ_2

Default values for ϕ_1 and ϕ_2 are respectively $29\frac{1}{2}^\circ$ N and $45\frac{1}{2}^\circ$ N (values normally used for maps of the conterminous United States).

Lambert Equal Area Projection

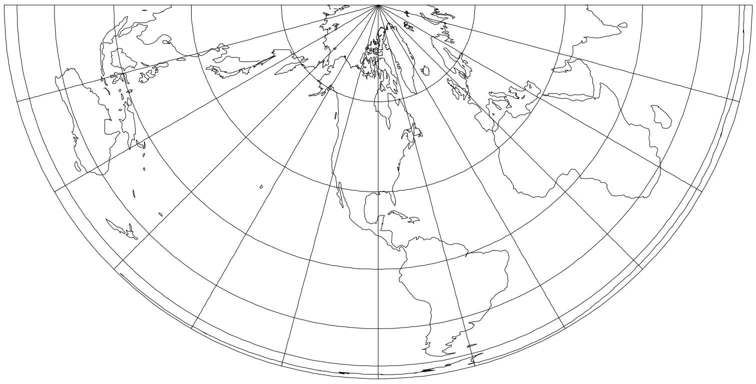

Figure 41: Lambert Equal Area Projection, with shorelines and 30° graticule. Central Meridian 90° W and standard parallel at 0° (`+proj=leac +lon_0=90w`).

Classifications: Conic. Equal area.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=leac` `+lat_1=`ϕ_1 `+lat_0=`ϕ_0 `+south`

Default value for ϕ_1 is 0°. Select parameter `+south` for south polar aspect. Limiting form is Polar Lambert Azimuthal Equal Area ($\phi_1 = \pm 90°$).

Bipolar Oblique Conic Conformal Projection

Classifications: Conformal conic.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=bipc +ns`

Not illustrated. For non-skewed cartesian coordinates select option `+ns`.

Polyconic (American) Projection

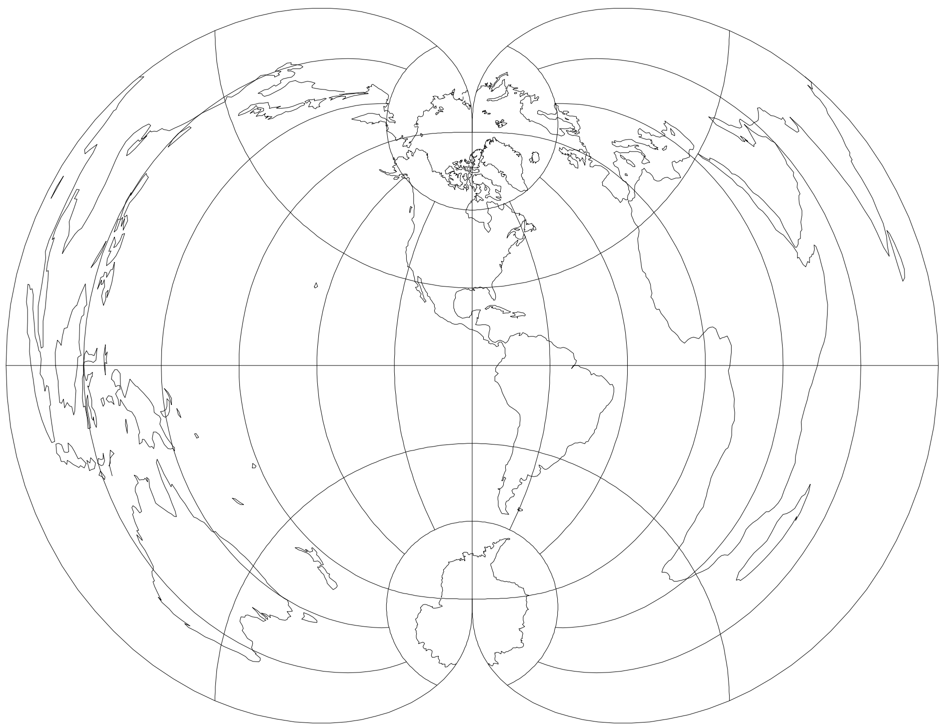

Figure 42: Polyconic (American) projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=poly +lon_0=90w`).

Classifications: Neither conformal nor equal area.
Available forms: Forward and inverse, spherical and elliptical projection.
Usage and options: `+proj=poly +lat_0=`ϕ_0

Inverse projection of cartesian coordinates representing λ greater than 90° from λ_0 cannot be performed.

Rectangular Polyconic Projection

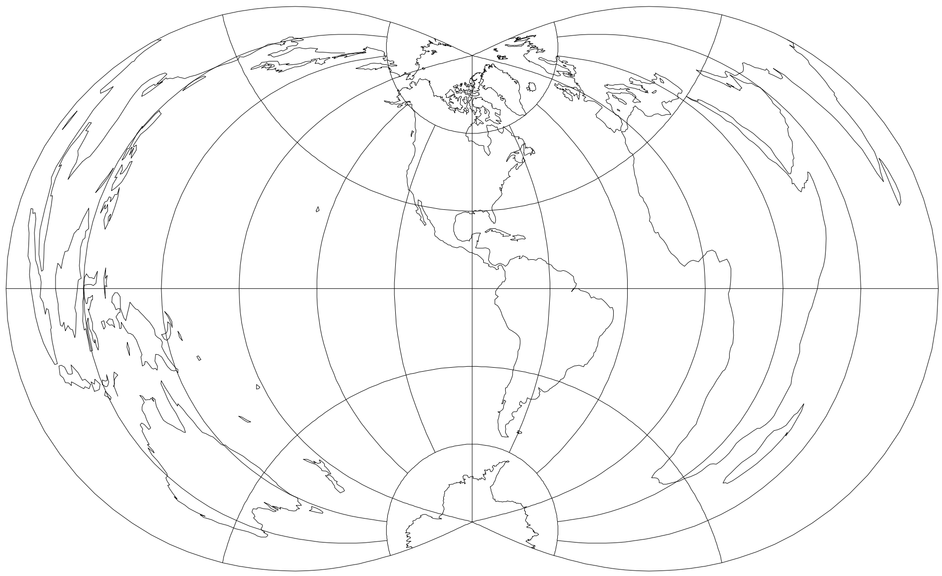

Figure 43: Rectangular Polyconic projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=rpoly +lon_0=90w`).

Classifications: Neither conformal nor equal area.

Aliases: War Office (British)

Available forms: Forward, spherical projection.

Usage and options: `+proj=rpoly +lat_ts=`ϕ_s

A latitude of true scale may be specified with `lat_ts=`ϕ_s as shown in figure 44 If not specified, $\phi_s = 0$ is assumed as shown in figure 43.

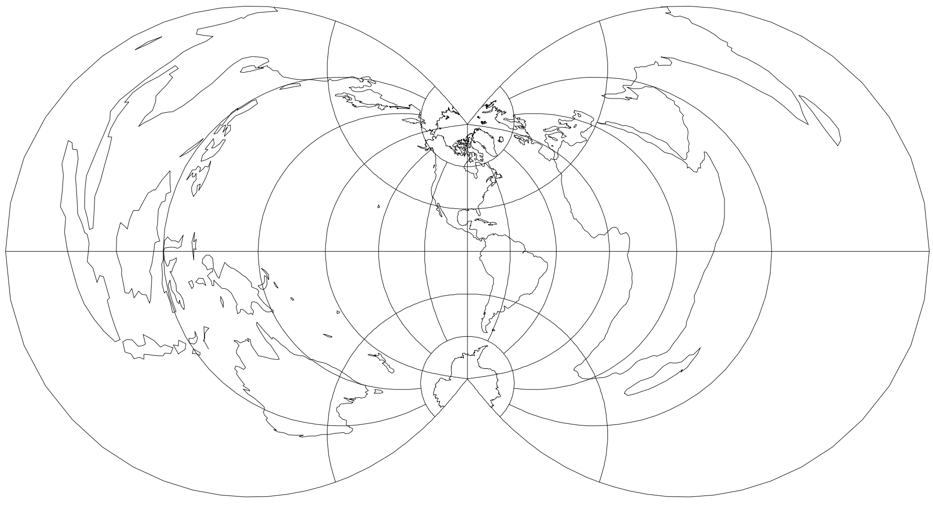

Figure 44: Rectangular Polyconic projection, with shorelines and 30° graticule. Central Meridian 90° W. North and south latitudes of 45° are at true scale. (`+proj=rpoly +lon_0=90w +lat_ts=45`).

Bonne Projection

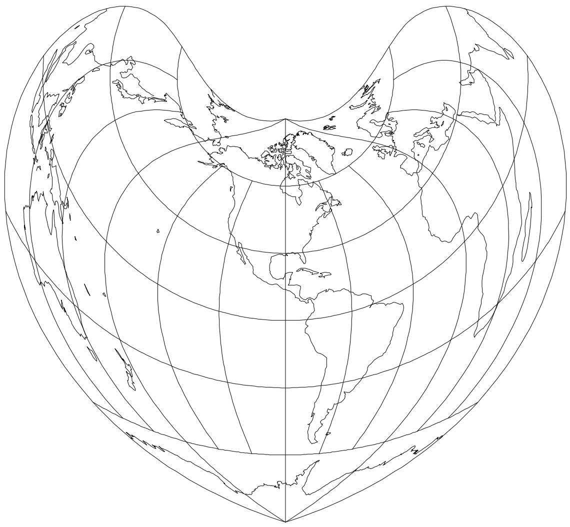

Figure 45: Bonne projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=bonne +lon_0=90w`).

Classifications: Pseudoconic. Equal area.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=bonne +lat_1=`ϕ_1

Default value for ϕ_1 is 40° N. The Werner projection (figure 46) is a variant of the Bonne where $\phi_1 = \pm 90°$. Parameter `+lat_0` is not employed in this projection and the cartesian origin is at λ_0 and ϕ_1.

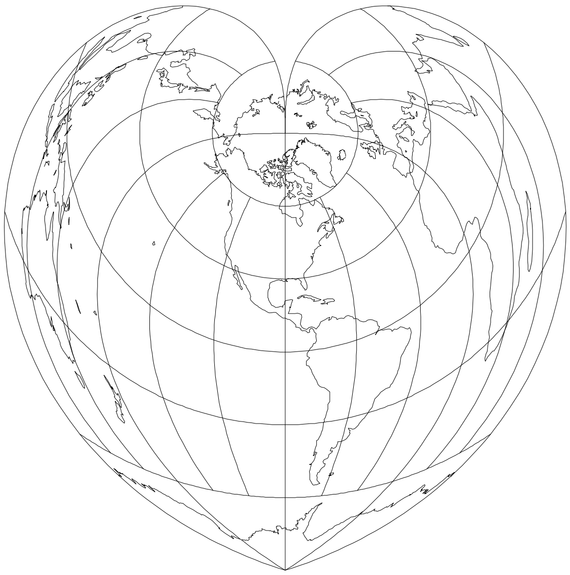

Figure 46: Werner projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=bonne +lon_0=90w +lat_1=90n`).

Azimuthal Projections

Azimuthal projections are based upon projections to a plane tangent to the Earth's surface at a point specified by `lon_0=`λ_0 and `lat_0=`ϕ_0. When ϕ is $\pm 90°$ or ϕ is $0°$ the projections are respectively termed polar and equatorial, otherwise they are termed oblique. In all cases, the x–y origin coincides with λ_0–ϕ_0.

Stereographic Projection

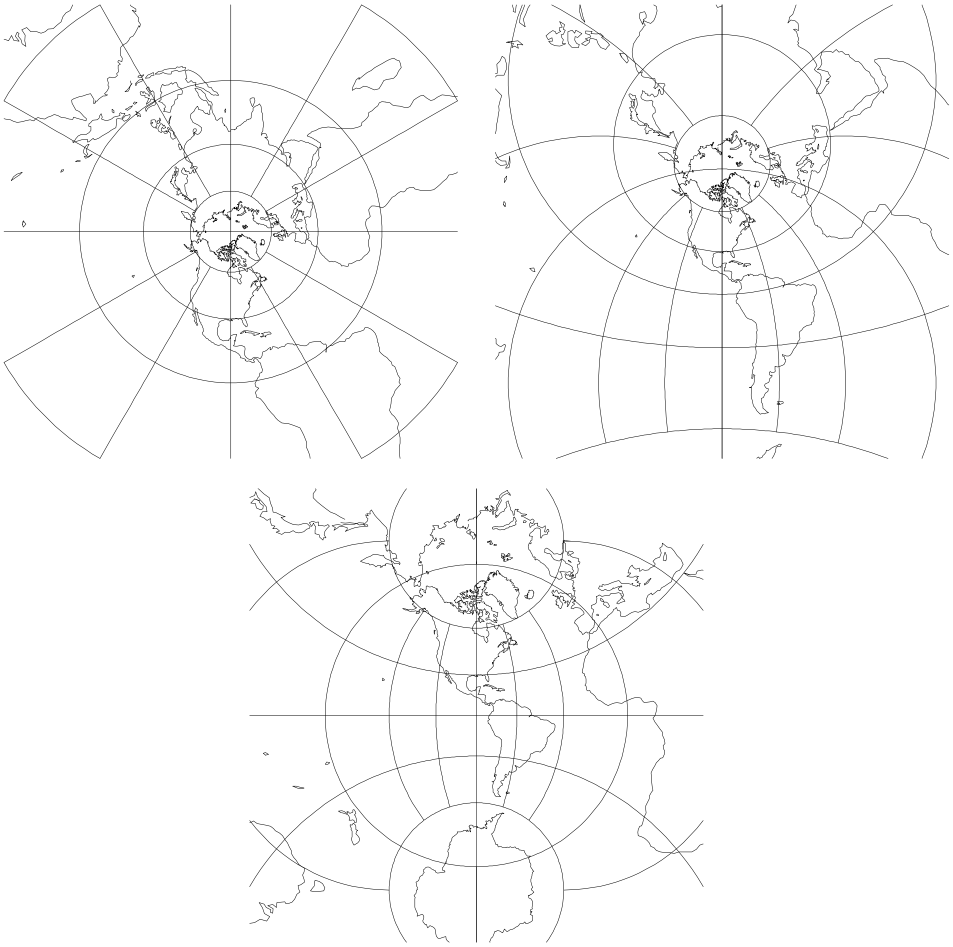

Figure 47: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Stereographic projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=stere +lon_0=90w`)

Classifications: Azimuthal. Conformal.

Available forms: Forward and inverse, spherical and elliptical projection.

Usage and options: `+proj=stere +lat_0=`ϕ_0 `+k=`k_0 `+lat_ts=` ϕ_0

The parameter k_0 is the scaling factor at the tangency point of the projection (λ_0, ϕ_0) and is 1. if `+k` is not specified. For the polar aspects, the latitude of true scale, ϕ_s, may be employed in lieu of k_0 when $\phi_s \neq \pm 90°$ (the default value).

Universal Polar Stereographic (UPS) Projection

Usage and options: `+proj=ups +south`

The UPS projection is a special case polar aspect of the Stereographic projection designed to cover the regions where $\phi \geq 84°$ N or $\phi \leq 80°$ S. The internal Stereographic parameters are fixed at $k_0 = 0.994$, $\lambda_0 = 0°$, $x_0 = y_0 = 2,000,000$m, and ϕ_0 is either 90° N when `+south` is not specified or 90° S when `+south` is specified. Elliptical parameters must be used (either *proj* default or entered as option). Also see the Universal Transverse Mercator (UTM) projection (p. 12).

Gnomonic Projection

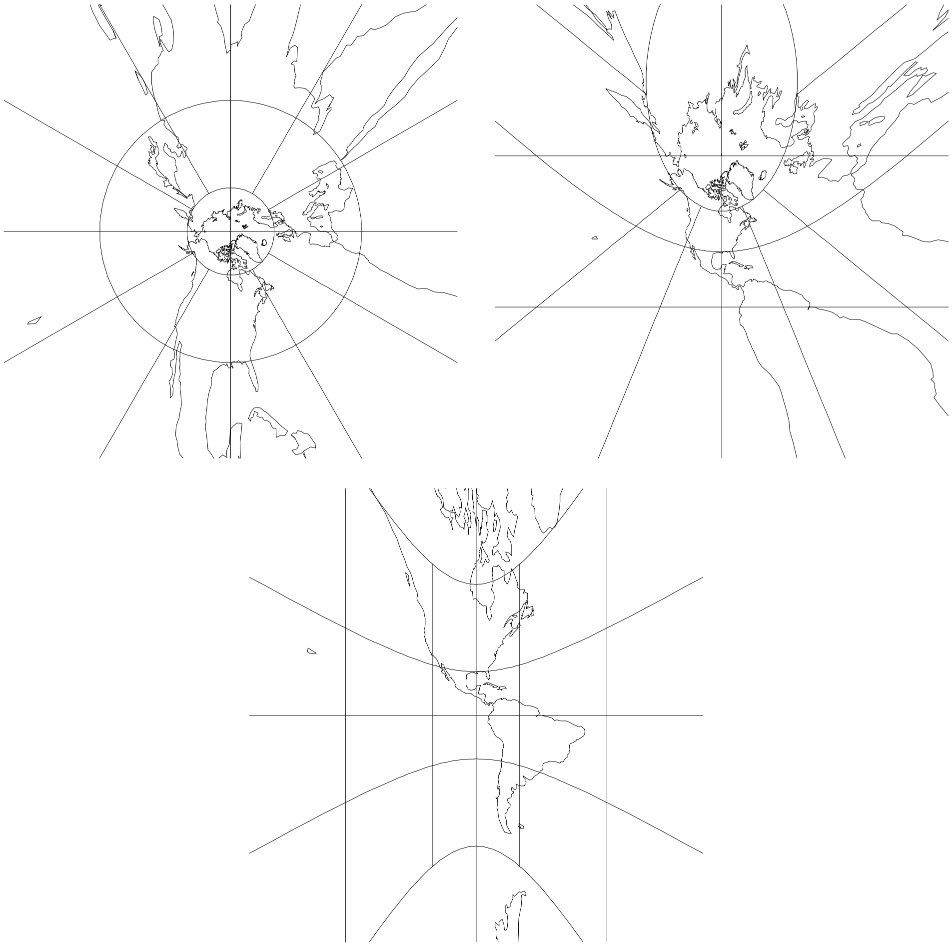

Figure 48: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Gnomonic projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=gnom +lon_0=90w`)

Classifications: Azimuthal. Neither confomal nor equal area.

Aliases: Gnomic, Central.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=gnom`

Orthographic Projection

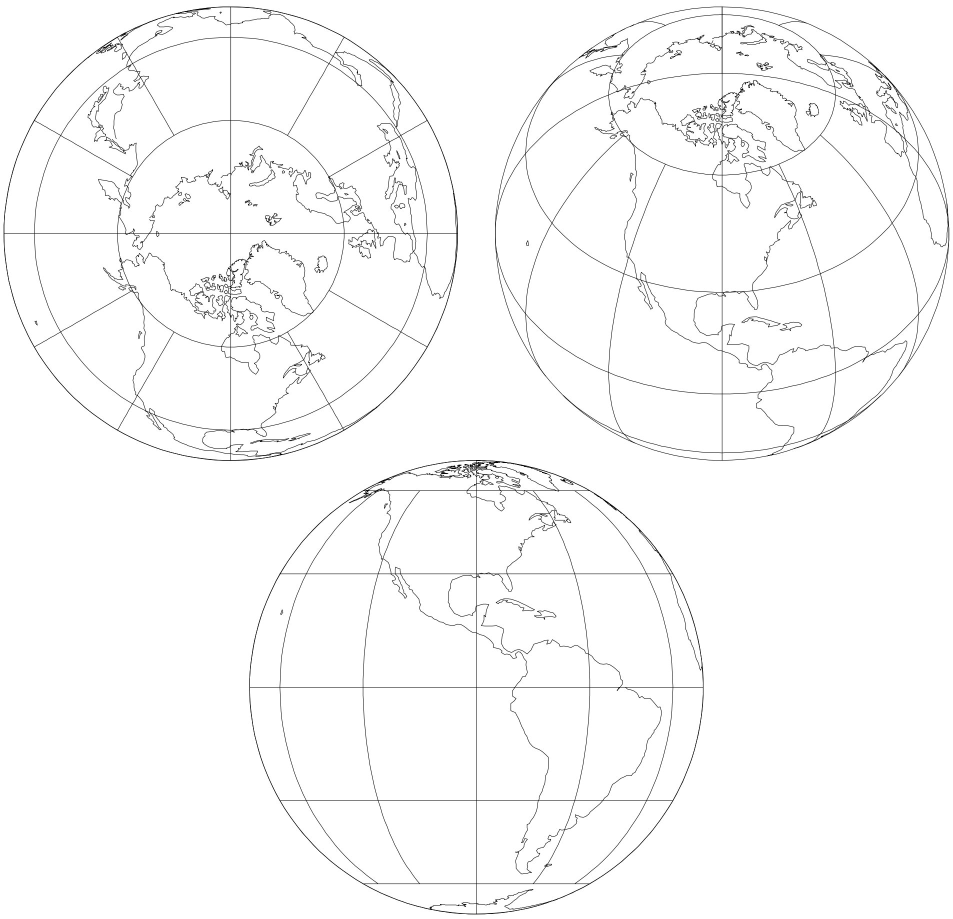

Figure 49: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Orthographic projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=ortho +lon_0=90w`)

Classifications: Azimuthal. Neither confomal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=ortho`

Airy Projection

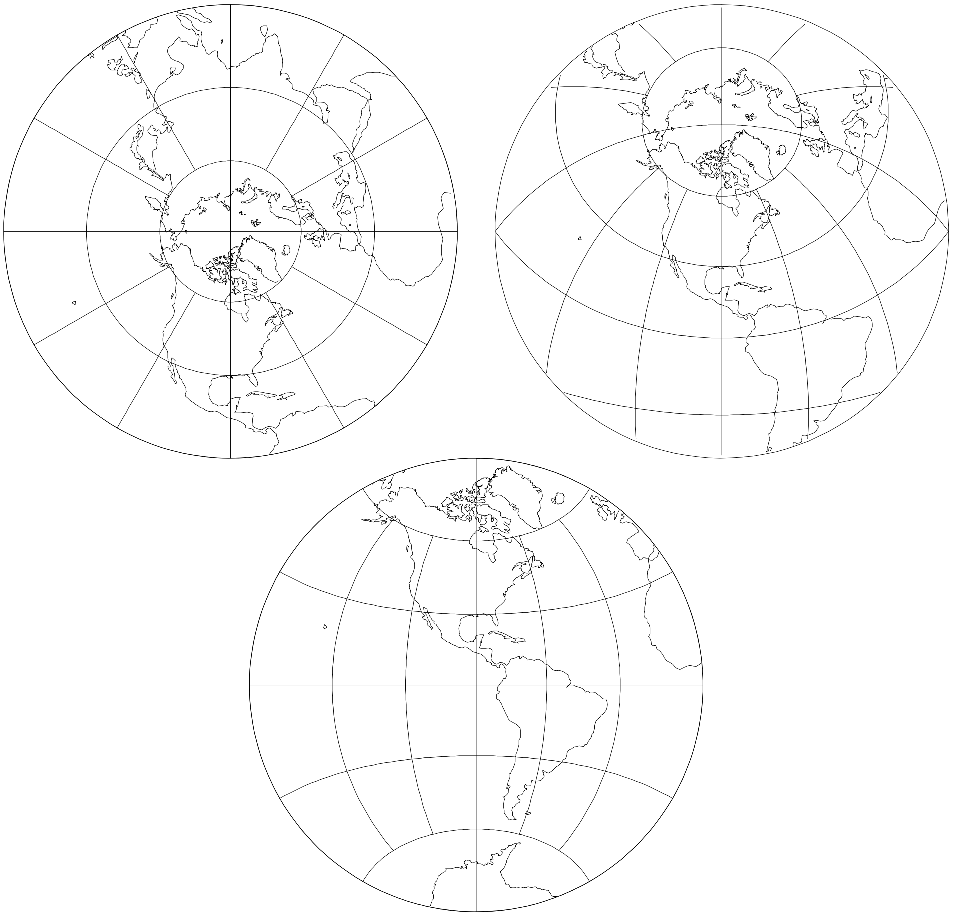

Figure 50: Polar (`+lat_0=90`), oblique (`+lat_0=45`) and equatorial aspects of the Airy projection, with shorelines and 30° graticule. Central Meridian 90° W and $\phi_b = 90°$. (`+proj=airy +lon_0=90w`)

Classifications: Azimuthal. Neither confomal nor equal area.

Available forms: Forward spherical projection.

Usage and options: `+proj=airy +lat_b=`ϕ_b `+no_cut`

The Airy projection is an azimuthal minimum error projection for the region within the small or great circle defined by an angular distance, ϕ_b, from the tangency point of the plane (λ_0, ϕ_0). The default value for ϕ_b is 90° that is suitable for hemispherical maps. Extent of projection is limited to the hemisphere unless `+no_cut` is specified.

Near-Sided Perspective Projection

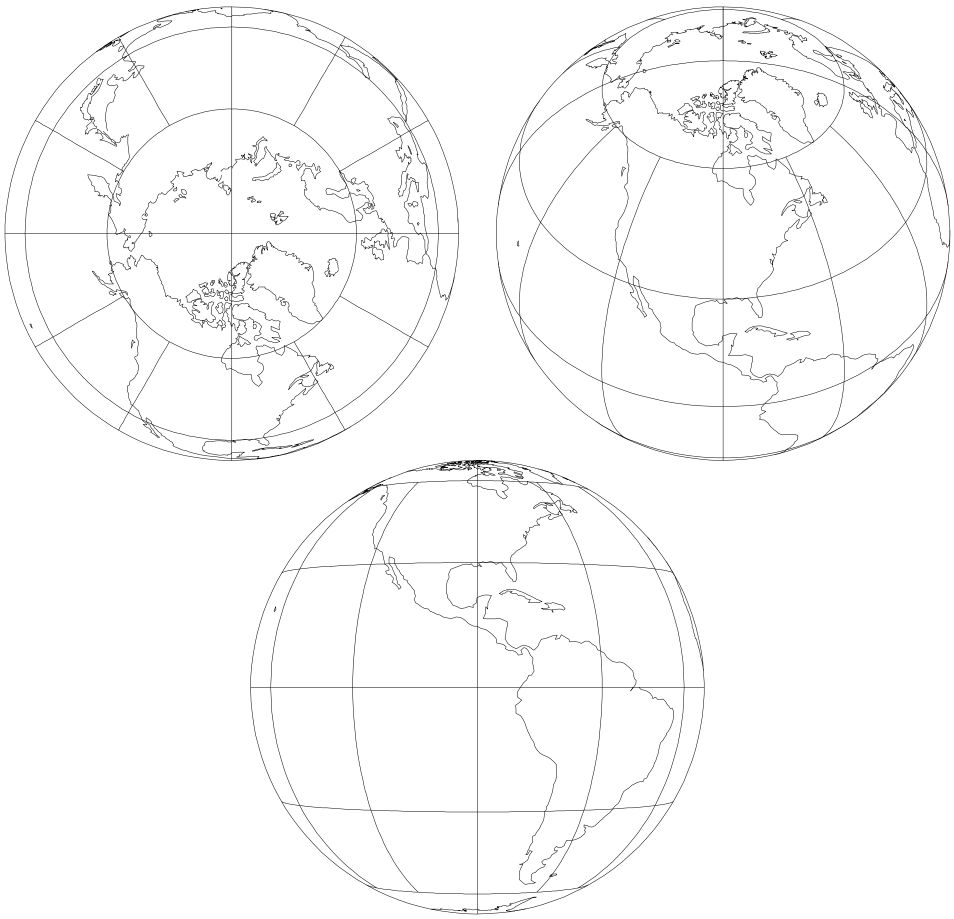

Figure 51: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Near-Sided Perspective projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=nsper +lon_0=90w +a=1 +h=8`)

Classifications: Azimuthal. Neither confomal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=nsper +h=`h

Parameter h is the height of the view point above the Earth and must be in the same units as a.

Lambert Azimuthal Equal Area Projection

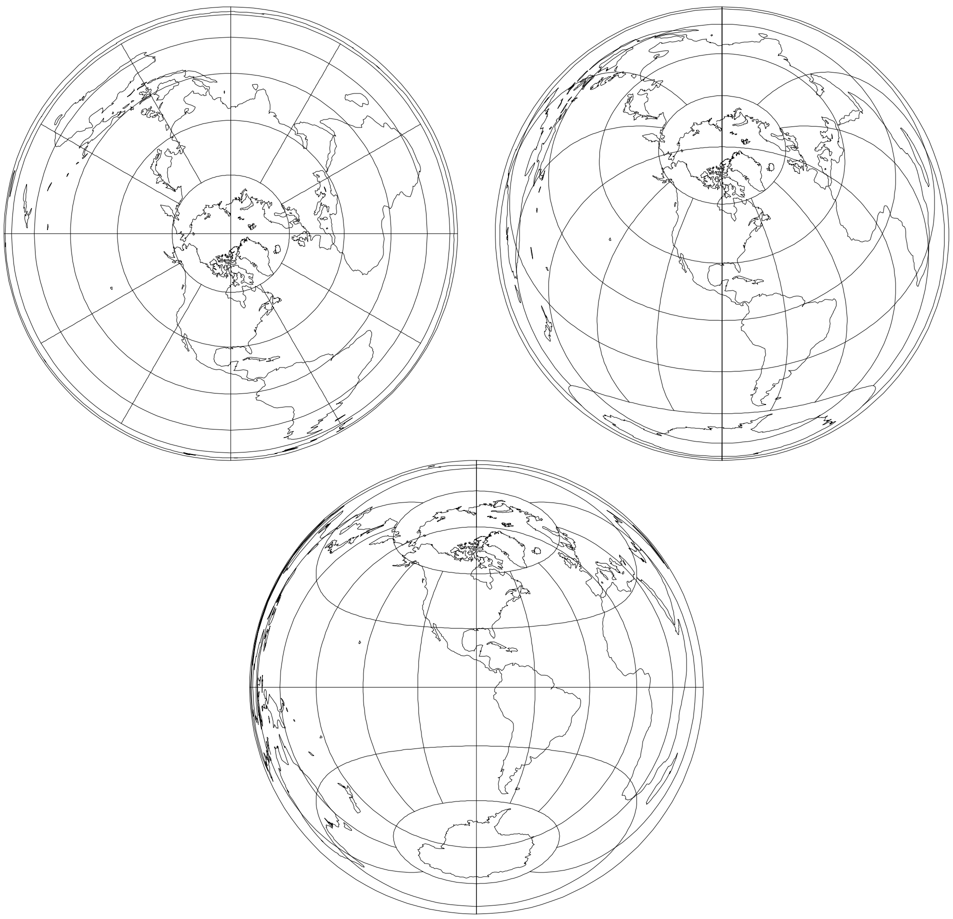

Figure 52: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Lambert Azimuthal Equal Area projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=laea +lon_0=90w`)

Classifications: Azimuthal. Equal area.

Aliases: Lorgna, Zenithal Equal-Area, Zenithal Equivalent.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=laea`

Azimuthal Equidistant Projection

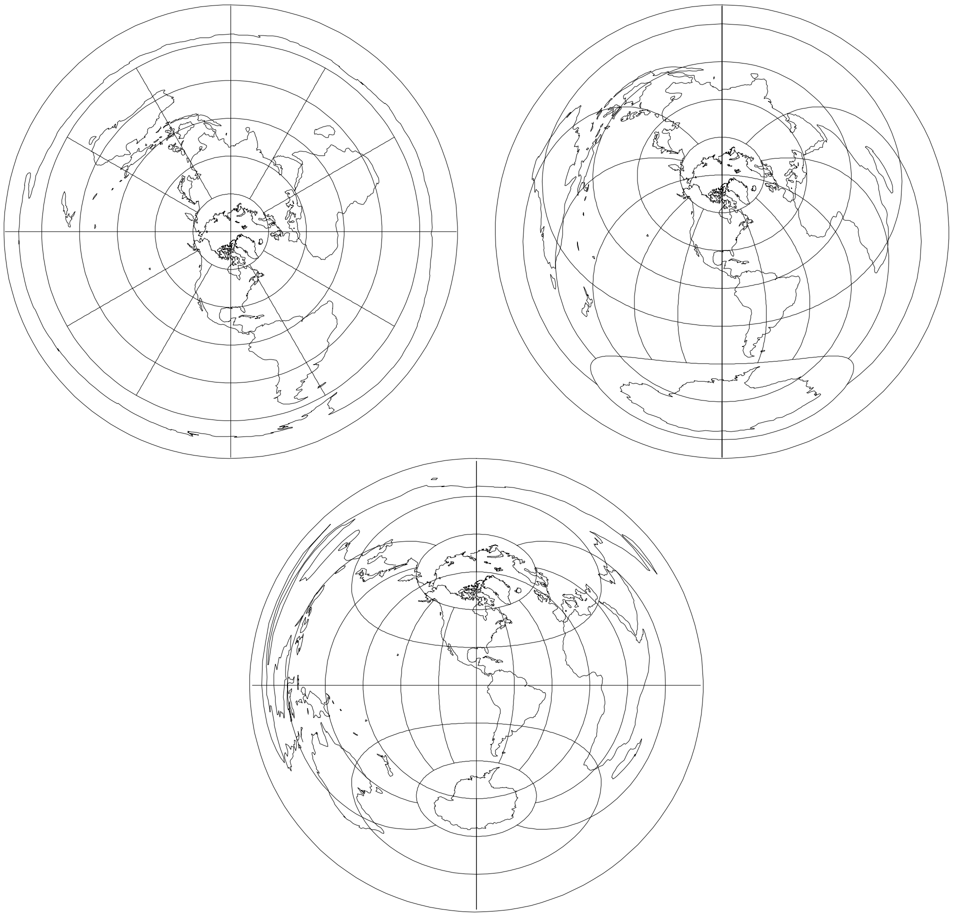

Figure 53: Polar (`lat_0=90`), oblique (`lat_0=45`) and equatorial aspects of the Azimuthal Equidistant projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=aeqd +lon_0=90w`)

Classifications: Azimuthal. Neither confomal nor equal area.

Aliases: Postel, Zenithal Equidistant.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=aeqd`

Hammer Projection

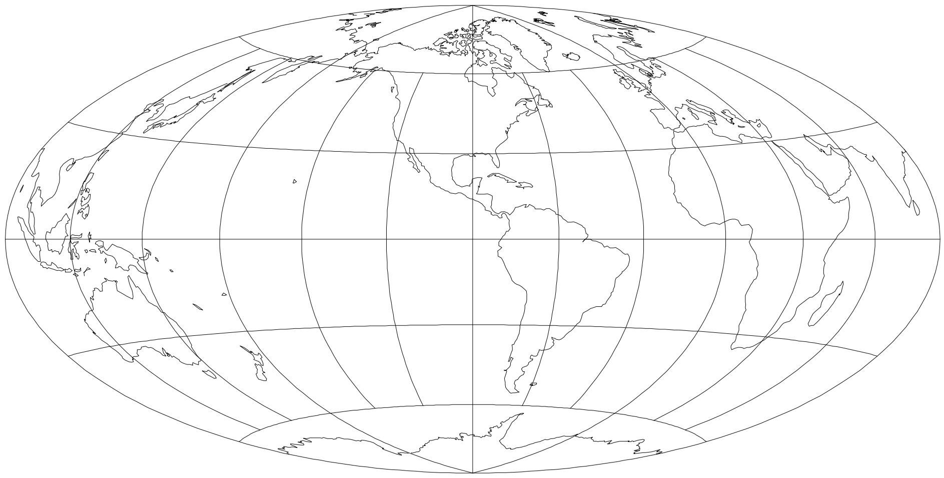

Figure 54: Hammer projection, with shorelines and 30°graticule. Central Meridian 90° W. (`+proj=hammer +lon_0=90w`)

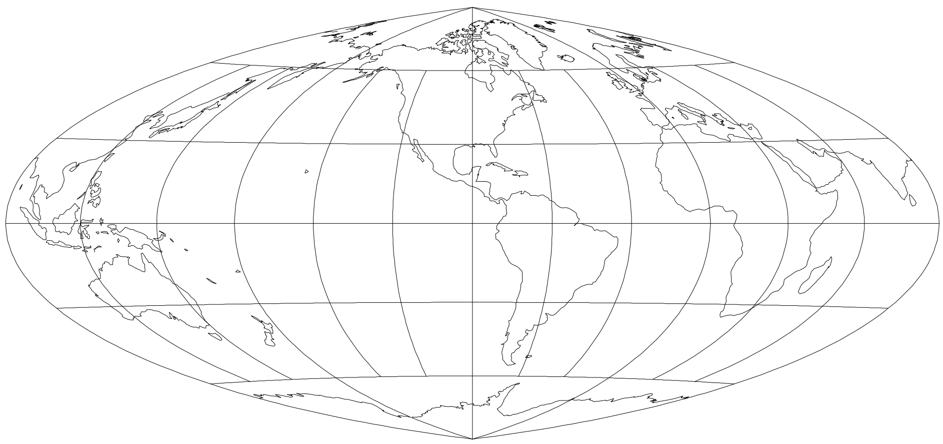

Figure 55: Eckert-Greifendorff modified Hammer projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=hammer +lon_0=90w +W=.25`)

Classifications: Modified azimuthal, equal area.

Aliases: Hammer-Aitoff (also erroneously called Aitoff).

Available forms: Forward, spherical projection.

Usage and options: `+proj=hammer [ +W=`n` ]`

The real value n assigned to the `+W` option determines the normal Hammer projection, $n = 0.5$ or `+W` option omitted, shown in figure 54, the Eckert-Greifendorff, $n = 0.25$, projection shown in figure 55. A value of $n = 0$ is a terminal error.

Wagner VII Projection

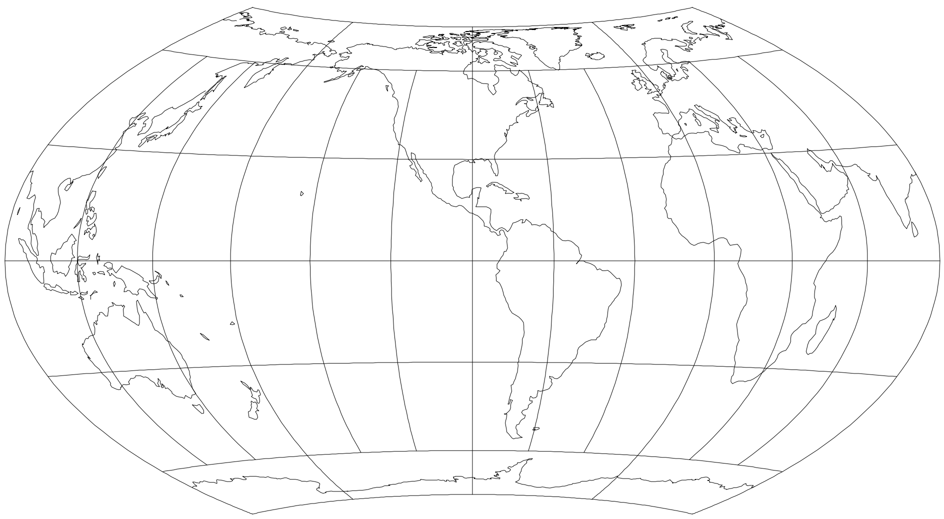

Figure 56: Wagner VII projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=wag7 +lon_0=90w`)

Classifications: Modified azimuthal. Equal area.

Aliases: Hammer-Wagner.

Available forms: Forward, spherical projection.

Usage and options: `+proj=wag7`

Aitoff Projection

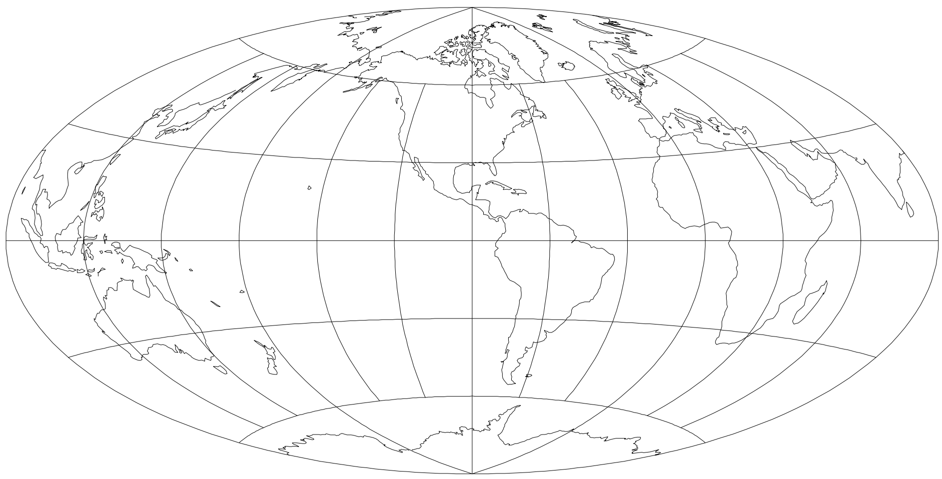

Figure 57: Aitoff projection, with shorelines and 30°graticule. Central Meridian 90° W. (`+proj=aitoff +lon_0=90w`)

Classifications: Modified azimuthal. Neither conformal nor equal area.

Available forms: Forward, spherical projection.

Usage and options: `+proj=aitoff`

Winkel Tripel Projection

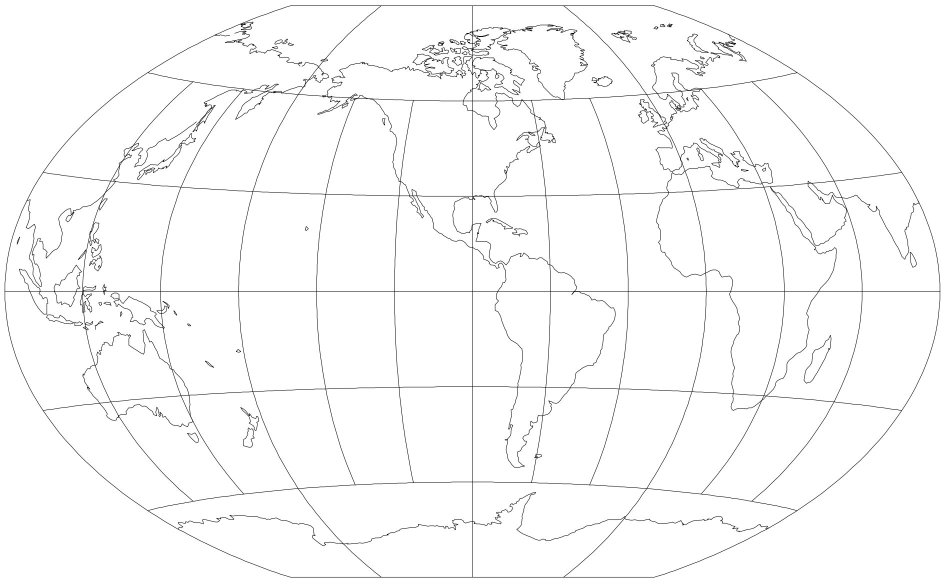

Figure 58: Winkel Tripel projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=wintri +lon_0=90w`)

Classifications: Modified azimuthal. Neither conformal nor equal area.

Available forms: Forward, spherical projection.

Usage and options: `+proj=wintri`

Miscellaneous Projections

The miscellaneous classification is employed for projections not mathematically developed as projections onto a conic surface and which are primarily used for global or hemispherical small scale maps. Except for certain cases with the Lagrange projection, the cartesian origin coincides with λ_0 and $\phi = 0°$, and `+lat_0` is not used with any of the miscellaneous projections.

August Epicycloidal Projection

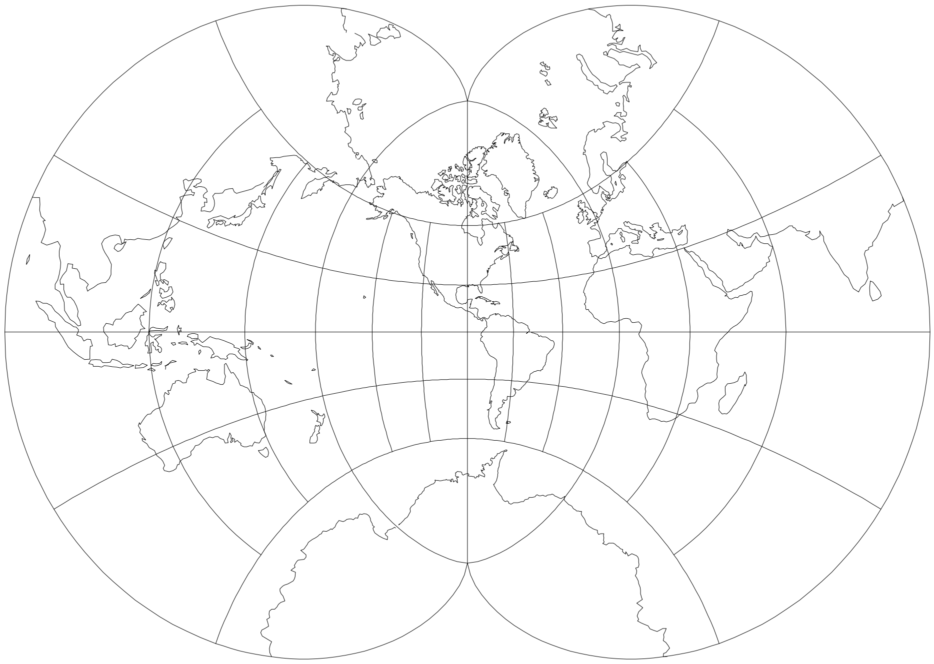

Figure 59: August Epicycloidal projection, with shorelines and 30°graticule. Central Meridian 90° W (`+proj=august +lon_0=90w`).

Classifications: Miscellaneous conformal.

Aliases: August.

Available forms: Forward, spherical projection.

Usage and options: `+proj=august`

Bacon Globular Projection

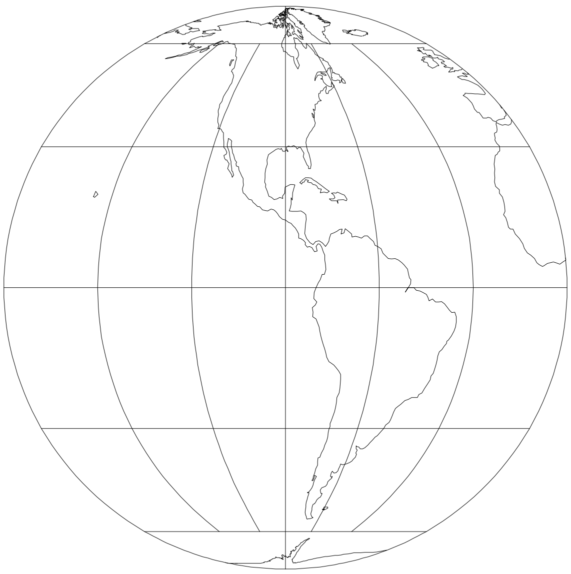

Figure 60: Bacon Globular projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=bacon +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=bacon`

Nicolosi Globular Projection

Figure 61: Nicolosi Globular projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=nicol +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=nicol`

Fournier Globular I Projection

Figure 62: Fournier Globular I projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=fourn +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=fourn`

Apian Globular I Projection

Figure 63: Apian Globular I projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=apian +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=apian`

Eisenlohr Projection

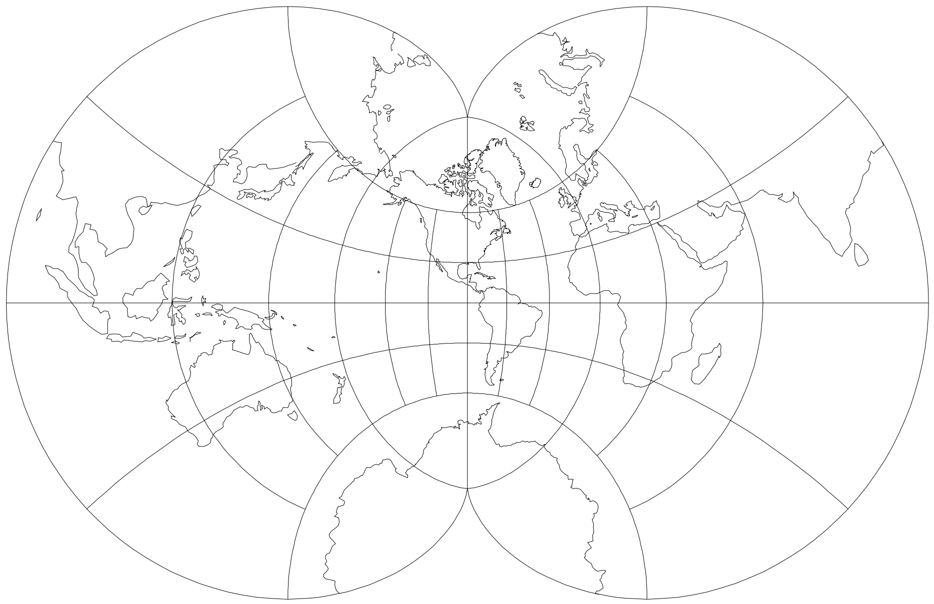

Figure 64: Eisenlohr projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=eisen +lon_0=90w`).

Classifications: Miscellaneous. Conformal.
Available forms: Forward and inverse, spherical projection.
Usage and options: `+proj=eisen`

Ortelius Oval Projection

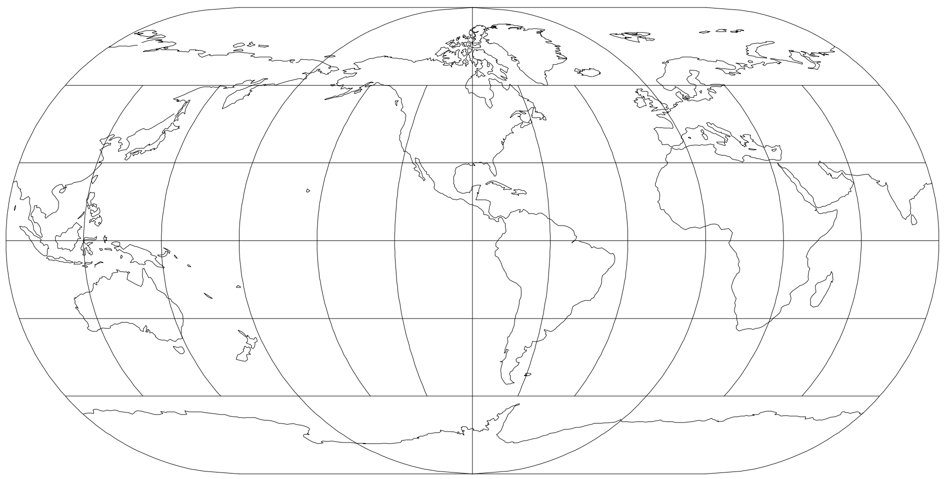

Figure 65: Ortelius Oval projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=ortel +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=ortel`

Van der Grinten (I) Projection

Figure 66: Van der Grinten I projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=vandg +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.

Available forms: Forward and inverse, spherical projection.

Usage and options: `+proj=vandg`

Van der Grinten II Projection

Figure 67: Van der Grinten II projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=vandg2 +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.

Available forms: Forward, spherical projection.

Usage and options: `+proj=vandg2`

Van der Grinten III Projection

Figure 68: Van der Grinten III projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=vandg3 +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=vandg3`

Van der Grinten IV Projection

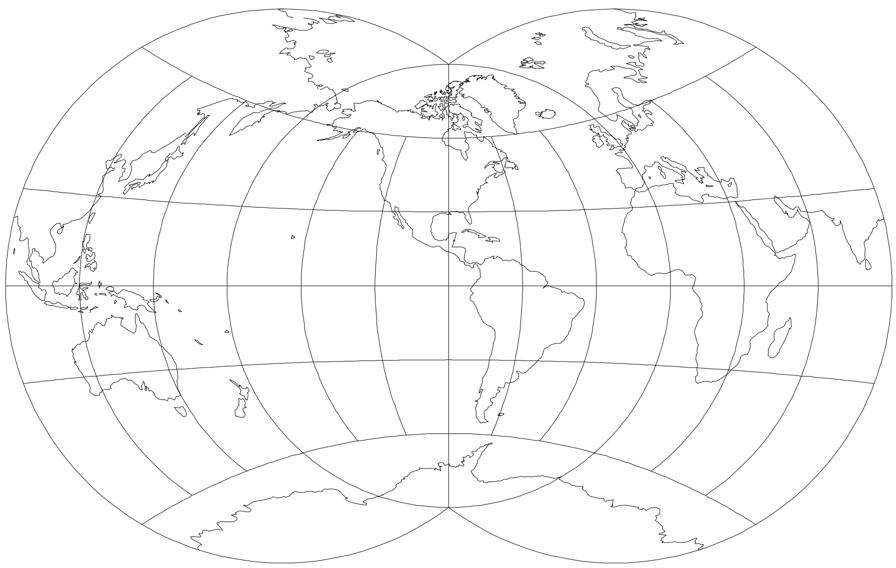

Figure 69: Van der Grinten IV projection, with shorelines and 30° graticule. Central Meridian 90° W (`+proj=vandg4 +lon_0=90w`).

Classifications: Miscellaneous. Neither conformal nor equal area.
Available forms: Forward, spherical projection.
Usage and options: `+proj=vandg4`

Lagrange Projection

Figure 70: Lagrange projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=lagrng +lon_0=90w`)

Classifications: Miscellaneous. Conformal.

Available forms: Forward, spherical projection.

Usage and options: `+proj=lagrng +W=`r `+lat_1=`ϕ_1

The `+W` option provides for specifying the ratio $r = \lambda'/180°$ that defines a circular meridian passing through the poles and $\lambda_0 \pm \lambda'$. Figure 70 shows the default value of $r = 2$, and figures 71 and 72 show respective values of $r = 1$ and $r = 1.4$. Option `+lat_1=`ϕ_1 specifies a parallel that will be a straight line ($\phi_1 = 0$ when not specified) as well as the origin of the y axis. Figure 73 is an example for $\phi_1 = 30°$S.

Figure 71: Lagrange projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=lagrng +lon_0=90w +W=1`)

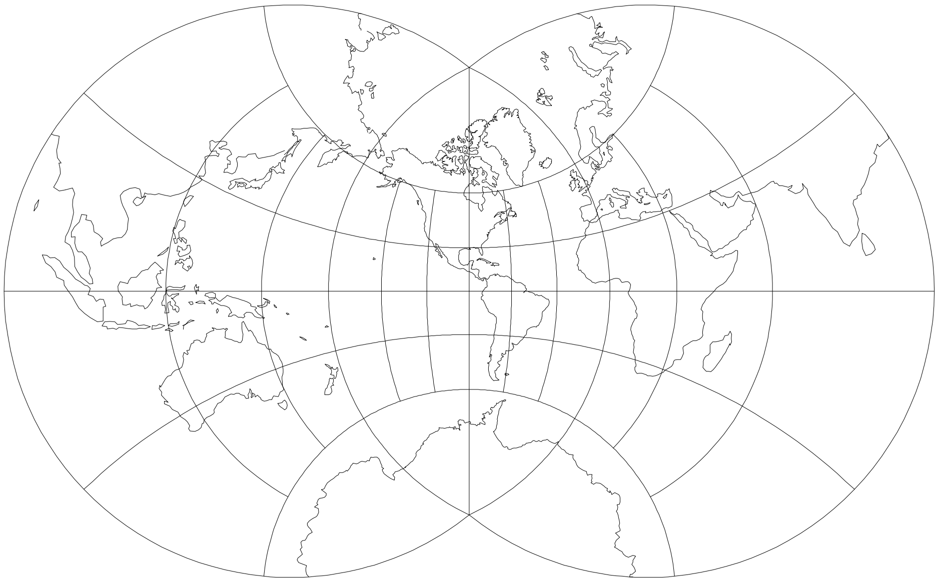

Figure 72: Lagrange projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=lagrng +lon_0=90w +W=1.4`)

Figure 73: Lagrange projection, with shorelines and 30° graticule. Central Meridian 90° W. (`+proj=lagrng +lon_0=90w +lat_1=30s`)

U.S. State Plane Coordinate Systems (SPCS)

Since 1935 each of the United States have defined by State legislative action one or more SPCS zones in terms of datum, geographical extent and cartographic projection parameters relating geographic coordinates to the cartesian coordinates used in land surveying. Mitchell and Simmons (1945, p. 45–47) and Snyder (1988, p. 52–56) provide tables of the cartographic parameters for each of these zones. The North American Datum of 1927 (NAD27) and Clarke 1866 ellipsoid are used for SPCS27 and cartesian units are in "U.S. surveyor feet" with a metric conversion of 1 m $\equiv$ 39.37 in. (versus the International conversion of 1 in. $\equiv$ 2.54 cm).

Subsequent to the use of modern surveying equipment and satellite data, improved positioning information has resulted in establishing a new datum, NAD83, SPCS83 and the GRS80 ellipsoid (Stem, 1989). Several States have also altered their zones and units of the cartesian coordinate system. It is unclear as to the status of NAD83 in each States because of the current transitional period (see Stem, 1989, p. 9) and users should check with local State authorities as to the exact system in current usage.

SPCS Conversion Procedures

Procedures are available to perform forward, `for27` or `for83`, or inverse, `inv27` or `inv83`, conversions for respective SPCS27 or SPCS83. If any of the procedures are executed without run-line arguments the following usage line will be printed:

```
usage: xxxxx state_code [ -M ] [ proj_parameters ] [ file ... ]
```

where `xxxxx` is the procedure name. The `state_code` is a National Geodetic Survey reference number for each of the SPCS zones. If the user does not know the code number, an anotated list can be obtained by executing `nad27` or `nad83` followed by the state's name. For example, to obtain Colorado's SPCS27 zone codes:

```
$ nad27 colo
```

and the following results will appear:

```
# 501:colorado north datum:NAD27
# 502:colorado central datum:NAD27
# 503:colorado south datum:NAD27
```

Colorado's respective North, Central and South zone codes are 501, 502 and 503. A complete list is obtained by executing `nad27` or `nad83` without any run-line argument.

The `-M` options allows cartesian coordinates to be in meters rather than the default surveyor's feet and the `proj_parameters` provides for entering any of program *proj's* hyphen type runline parameters. To select forward transformation to meters with 4 fractional digits then:

```
for27 502 -M -f '%.4f' my.data | ...
```

As an example, using real numbers, the SPCS27 cartesian coordinates for two sites near Boston are to be obtained from their latitude-longitude locations in a file *Boston.harbor*:

```
42d19'40.180"N  70d53'26.214"W # Boston Lighthouse
42d30'19.187"N  70d50'03.052"W # Marblehead Lighthouse
```

Execution of `nad27` determines the zone number:

```
$ nad27 mass
# 2001:massachusetts mainland datum:NAD27
# 2002:massachusetts island datum:NAD27
$ ...
```

And the following determine the SPCS27 coordinates in feet:

```
$ for27 2001 -r -f '%.3f' Boston.harbor
764787.570      484445.491 # Boston Lighthouse
779542.978      549243.661 # Marblehead Lighthouse
$ ...
```

Caveats

A few elements related to SPCS should be emphasized:

- The user must be warned that SPCS coordinates in one datum cannot be directly converted to the alternate datum. For example:

  ```
  inv27 2001 -o my.data.27 | for83 2001 -i >my.data.83
  ```

  is executionally correct, **but it will not produce the correct result!** The change in the datum represents shifts in the coordinates that are not compensated for by *proj* conversions.

- The tables employed by the software discussed in this appendix were adapted from tabular material in the GCTP II distribution tape (Elassal, 1987).

- The author has encountered tabulations of geographic-SPCS values which do not exactly agree with *proj* conversions (differences are in the centimeter range and limited to zones employing the Transverse Mercator projection). But exhaustive testing and improvement of *proj*'s accuracy and the fact that GCTP II agrees with *proj* to within 10^{-3} m or 3×10^{-3} feet suggests that either such tabulations were performed with insufficient precision or alternate algorithms were employed.

Appendix 1—Summary of basic commands

This is a summary of the material contained in the *man*(1) file available from host system as an on-line file or as hard copy from the system administrator.

The following run line control parameters can appear in any order:

`-b` — Special option for binary coordinate data input and output through standard input and standard output. Data is assumed to be in system type *double* floating point words. This option is useful when **proj** is a *son* process and allows bypassing formatting operations.

`-i` — Selects binary input only (see `-b` option).

`-o` — Selects binary output only (see `-b` option).

`-t`*a* — *A* specifies a character employed as the first character to denote a control line to be passed through without processing. This option applicable to ASCII input only. (`#` is the default value).

`-e␣`*string* — *String* is an arbitrary string to be output if an error is detected during data transformations. The default value is the string: `*\t*`. Note that if the options `-b`, `-i` or `-o` are employed, an error is returned as a HUGE value for both return values.

`-c␣`*file* — *File* is the source file name of additional `+` option control data for the projection procedures.

`-r` — This options reverses the order of the expected input from longitude-latitude or x-y to latitude-longitude or y-x.

`-s` — This options reverses the order of the output from x–y or longitude-latitude to y–x or latitude-longitude.

`-m␣`*mult* — The cartesian data may be scaled by the *mult* parameter. When processing data in a forward projection mode the cartesian output values are multiplied by *mult* otherwise the input cartesian values are divided by *mult* before inverse projection. If the first two characters of *mult* are `1/` or `1:` then the reciprocal value of *mult* is employed.

`-f␣`*format* — *Format* is a *printf*(3) format string to control the form of the output values. For inverse projections, the output will be in degrees when this option is employed. If a format is specified for inverse projection the output data will be in decimal degrees. The default format is `%.2f` for forward projection and DMS for inverse.

`-[w|W]`*n* — *N* is the number of significant fractional digits to employ for seconds output (when the option is not specified, `-w3` is assumed). When `-W` is employed the fields will be constant width with leading zeroes.

The `+args` runline arguments are associated with cartographic parameters and usage varies with projection selected and only two are discussed here They may be either entered on the run line where they must be preceeded by the `+` symbol or specified in the file reference by the `-c` option where the `+` prefix is optional. The options are processed in left to right order from the run line followed by processing the entries in the `-c` file. Reentry of an option is ignored with the first occurance assumed to be the desired value.

`+proj=`*name* — is **always** required for selection of the transformation function and where *name* is an acronym for the desired projection. A list of currently available projections and their *names* can be obtained by: `proj +proj=list | more`.

`+ellps=`*acronym* — The `+ellps` option allows selection of standard, predefined ellipsoid figures. For a list of currently available ellipsoid figures and their *acronyms* execute: `proj +ellps=list | more`.

`+inv` — The presence of this option selects the inverse projection mode of operation. It is equivalent to executing program `invproj`.

`+geoc` — Geographic coordinates are to be treated as geocentric.

`+over` — Inhibit reduction of longitude range to within $\pm 180°$ of the central meridian.

References

Elassal, A.A., 1987, General Cartographic Transformation Package (GCTP), Version II: NOAA Technical Report NOS 124 CGS 9, 24 p.

Evenden, G.I., 1983, Forward and inverse cartographic projection procedures: U.S. Geological Survey Open-File Report 83-623, 85 p.

__________, Botbol, J.M., 1985, User's manual for MAPGEN (UNIX version): a method to transform digital cartographic data to a map: U.S. Geological Survey Open-File Report 85-706, 134 p.

Kernighan, B.W., Ritchie, D.M., 1978, The C Programming Language: Englewood Cliffs, New Jersey, Prentice-Hall

Mitchell, H.C., Simmons, L.G., 1945, The State Coordinate Systems (A Manual for Surveyors): NOAA Special Publication No. 235, reprinted 1987, 62 p.

Snyder, J.P., 1987, Map projections—A working manual: U.S. Geological Survey Professional Paper 1395, 383 p.

__________,Voxland, R.M., 1989, An album of map projections: U.S. Geological Survey Professional Paper 1453, 249 p.

Stem, J.E., 1989, State Plane Coordinate System of 1983: NOAA Manual NOS NGS 5, 119 p.

Index

- placement of processing *stdin*, 3

Azimuthal Projections, 39–49

basic program usage, 2
binary I/O, 7

cartesian scaling, 7
cartographic parameters, 7
  `+a`, 9
  `+e`, 9
  `+ellps`, 8
  `+es`, 9
  `+f`, 9
  `+geoc`, 10
  `+inv`, 4
  `+lat_0`, 10
  `+lon_0`, 10
  `+over`, 10
  `+proj`, 3, 8
  `+rf`, 9
  `+x_0`, 10
  `+y_0`, 10
  effect of order, 8
comments in data input, 4
Conic Projections, 33–39
controlling output precision
  $\lambda - \phi$, 6
  $x - y$, 5
cup character, 2
Cylindrical Projections, 11–19

data input, 3
data output, 5
DMS degree, minute, second format, 3

elliptical figure, 9
elliptical listing, 9

history of *proj*, 1

input data format, 3
inverse projection
  by executing *invproj*, 4

longitude over-range, 10

MAPGEN, 2
Miscellaneous Projections, 49–56

order of processing input files, 3

projection list, 8
Projections
Airy, 42
Aitoff, 47
Albers Equal Area, 35
Apian Globular I, 51
August Epicycloidal, 49
Azimuthal Equidistant, 45
Babinet, 22
Bacon Globular, 50
Behrmann, 15
Bipolar Oblique Conic Conformal, 36
Boggs Eumorphic, 31
Bonne, 38
Cassini, 19
Central, 40
Central Cylindrical, 13
Collignon, 31
Craster Parabolic, 32
Cylindrical Equal Area, 15
Denoyer Semi-Elliptical, 32
Eckert I, 23
Eckert II, 23
Eckert III, 24
Eckert IV, 24
Eckert V, 25
Eckert VI, 25
Eckert-Greifendorff, 46
Eisenlohr, 52
Elliptical, 22
Equidistant Conic, 34
Equidistant Cylindrical, 18
Flat-Polar Quartic, 28
Fournier Globular I, 51
Gall (Stereographic), 16
Gall Orthographic, 15
Gauss Conformal, 12
Gauss-Krüger, 12
Gnomic, 40
Gnomonic, 40
Goode Homolosine, 26
Hammer, 46
Hammer-Aitoff, 46
Hammer-Wagner, 47
Hatano Asymmetrical Equal-Area, 26
Homalographic, 22
Homolographic, 22
Homolosine, 26
Lagrange, 55
Lambert Azimuthal Equal Area, 44
Lambert Conformal Conic, 33

Lambert Cylindrical Equal Area, 15
Lambert Equal Area, 35
Lorgna, 44
Loximuthal, 27
McBryde-Thomas Flat-Polar Parabolic, 27
McBryde-Thomas Flat-Polar Quartic, 28
McBryde-Thomas Flat-Polar Sinusoidal, 28
Mercator, 11
Mercator Equal-Area, 21
Mollweide, 22
Near-Sided Perspective, 43
Nicolosi Globular, 50
Oblique Mercator, 13
Ortelius Oval, 52
Orthographic, 41
Orthophanic, 22
Perspective Conic, 34
Peters, 15
Plate Carée, 18
Polyconic (American), 36
Postel, 45
Putniņš P_2', 29
Putniņš P_5, 29
Quartic Authalic, 30
Rectangular Polyconic, 37
Robinson, 22
Sanson-Flamsteed, 21
Simple Perspective Cylindrical, 13
Sinusoidal, 21
Stereographic, 39
Transverse Central Cylindrical, 14
Transverse Cylindrical Equal Area, 17
Transverse Cylindrical Orthomorphic, 12
Transverse Mercator, 12
Universal Polar Stereographic (UPS), 40
Universal Transverse Mercator (UTM), 12
Van der Grinten (I), 53
Van der Grinten II, 53
Van der Grinten III, 54
Van der Grinten IV, 54
Wagner VII, 47
War Office, 37
Werner, 38
Winkel I, 30
Winkel Tripel, 48
Wright, 11
Zenithal Equal-Area, 44
Zenithal Equidistant, 45
Zenithal Equivalent, 44
Pseudo Cylindrical Projections, 21–33

release
determination, 2
previous to release 3, 2
version number, 2
reversing input data order, `-r`, 4
reversing order of output, 6
run line parameters
`-b` both input and output binary, 59
`-c` specify control file, 8, 59
`-e` error output string, 6
`-f` output format, 5, 59
`-i` input binary, 7, 59
`-m` specify cartesian scaling, 7, 59
`-o` output binary, 7, 59
`-r` revese order of input, 4, 59
`-s` revese order of output, 6, 59
`-t` select comment tag character, 5, 59
`-w` specify DMS significant figures, 6, 59

selection of projection, 8
Snyder, J.P., 1
State Plane Coordinate Systems (SPCS), 57
stdin, 3
stdout, 3

typological conventions, 2

untransformable data, 6

Voxland, P.M., 1

white space emphasis, 2

UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Cartographic Projection Procedures for the UNIX Environment—A User's Manual

by

Gerald I. Evenden[1]

Open-File Report 90–284

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of tradenames is for purposes of identification only and does not constitute endorsement by the U.S. Geological Survey.

May 13, 1990

(Revised September 24, 1995)

[1]Woods Hole, MA 02543

Contents

Introduction **1**
- History . . . 1
- Typological Conventions . . . 2

Basic Usage **2**
- Inputing Data . . . 3
- Form of Input Data . . . 3

Data Output **5**
- Output of error data . . . 6
- Binary Input–Output . . . 7
- Cartesian Scaling . . . 7

Cartographic Parameters **7**
- Selection of Projection . . . 8
- Specifying the Earth's Figure . . . 9
- Common Parameters . . . 10

Projection descriptions **10**

Cylindrical Projections **11**

Pseudocylindrical Projections **21**

Conic Projections **33**

Azimuthal Projections **39**

Miscellaneous Projections **49**

U.S. State Plane Coordinate Systems (SPCS) **57**
- SPCS Conversion Procedures . . . 57
- Caveats . . . 58

Appendix 1—Summary of basic commands **59**

References **61**