NAME

ertid − compute body tides on an elastic Earth, or Munk-Cartwright coefficients

DESCRIPTION

Ertid generates the theoretical earth tides on an elastic Earth (the ‘‘body tides’’) or the Munk-Cartwright coefficients which describe the different spherical harmonics of the tidal potential. (For a fuller description of these, see Munk and Cartwright [1966]). The tides are computed directly, by finding the positions of the Moon and Sun at each time and computing the resulting earth-tide at a given location.

After asking for the start time, end time, and sample interval (the times are given [in Greenwich time] as year, day-number, and hour), the program asks whether theoretical tides or Munk-Cartwright coefficients are wanted. In the first case, up to one gravity tide, two tilt tides, and three strain tides may be generated for a particular location. (As a special feature, if the number of gravity tides is −1, the potential height on a rigid Earth is generated instead of the gravity tide). The station coordinates needed for this option are given in degrees and decimals, and should be geographic (they are converted to geocentric internally to provide a partial correction for ellipticity). In the second case, Munk-Cartwright coefficients of order 2 or 3 (or both), and species 0 through 2 (or 3) may be computed and written out, the series for each coefficient to a separate file, with real and imaginary parts alternating. All outputs (except for some diagnostic messages) are to ASCII files. Each data series is written to a separate file, the filenames being read in one to a line. The data are written out to the file with a 5(g13.5) format.

Because the solar and lunar positions are calculated anew for each time, the run time goes as the number of points computed.

UNITS

The units and sign conventions are: for potential height, meters and positive up; for gravity, μgals (10^{-8} m s^{-2}) and positive for a decrease in g; for tilt, nanoradians and positive for motion of the plumb bob in the azimuth given (or a downwards ground tilt in that direction); for strain, nanostrain and positive for extension.

THEORY

The computation procedure roughly follows that given by Munk and Cartwright (1966): at each time point the positions of the Sun and Moon are found (more precisely, the latitude and longitude of the sublunar and subsolar points, and the ratio of true to mean distance); from these, the tides at the place of observation can be found directly. Munk and Cartwright give the derivation for the potential only; the extension to gravity, tilt, and strain is contained in Appendix C of Berger (1969). The two main sources of error are the imprecision of the solar and lunar ephemerides, and the neglect of the Earth’s ellipticity and rotation. The ephemerides used are those of Harrison (1971) for the Moon, with a simple theory for the Sun. The tides computed by this program have been checked against those gotten from a very complete ephemeris by Cartwright and Taylor (1971) and Cartwright and Edden (1973); the level of disagreement was less than 10^{-3}, which will be important only for the most precise observations of tidal gravity. The program assumes a spherical Earth, with second and third-degree parts of the potential only being taken into account.

It should be noted that this program computes, not the full Earth tide, but that tide less the amount contributed by the permanent deformation (and potential) contributed by the Sun and Moon. For gravity this quantity is known as the Honkasalo correction. The program, in computing this quantity, assumes that the elastic-Earth Love numbers apply. (These are the degree-two numbers, h_2 = 0.6114, k_2 = 0.304 and l_2 = 0.0832.)

REFERENCES

Berger, J. (1969) A Laser Earth Strain Meter, Ph.D. Thesis, University of California, San Diego

Cartwright, D. E., and R. J. Tayler (1971). New computations of the tide-generating potential, *Geophys. J. Roy. Astron. Soc.*, **23**, 45-73.

Cartwright, D. E., and A. C. Edden (1973). Corrected tables of tidal harmonics, *Geophys. J. Roy. Astron. Soc.*, **33**, 253-264.

Harrison, J. C. (1971). New computer programs for the calculation of Earth tides, Internal Report, Cooperative Institute for Research in Environmental Sciences, University of Colorado.

Munk, W. H., and D. E. Cartwright (1966), Tidal Spectroscopy and Prediction, *Phil. Trans. Roy. Soc., Ser. A.*, **259**, 533-581.

AUTHORS

J. Berger, W. Farrell, J. C. Harrison, J. Levine, D. C. Agnew

NAME

harprp − extracts individual results from a tidal load file

SYNOPSIS

harprp o | g | p | z | v

harprp d | l | | s | t azimuth

DESCRIPTION

Harprp reads from standard input, a load file (or set of them), selects the numbers associated with a particular quantity, and writes out a file of harmonic constants suitable for tidal prediction by *hartid*. The designators (on the command line) are:

o ocean-tide height; units are meters. In this case only, the ‘‘load file’’ is assumed to have only one entry (per constituent), that being for the ocean tide; such a file is produced by *oclook*.

g gravity; units are μgal, positive for decreasing *g*.

p potential height, relative to the surface of the Earth; units are millimeters, positive for up.

z vertical displacement; units are millimeters, positive for up.

d horizontal displacement at azimuth *azimuth* (given in degrees, clockwise from North); units are millimeters.

l extensional strain at azimuth *azimuth* (given in degrees); units are nanostrain.

v volume strain (for Poisson’s ratio 0.25); units are nanostrain.

s shear strain at azimuth *azimuth* (given in degrees); units are nanostrain (tensor strain).

t tilt at azimuth *azimuth* (given in degrees); units are nanoradians.

The first two lines of the output file indicate that the phases are local, and give the longitude of the station. The subsequent lines give the Cartwright-Tayler version of the Doodson code for the tidal constituent, and the amplitude and phase. The final line gives an impossible Cartwright-Tayler code, to indicate EOF.

SEE ALSO

hartid

AUTHOR

D. C. Agnew

NAME

hartid − predicts tides from harmonic constants

SYNOPSIS

hartid year [day-of-year | month day] hr min sec nterms samp

DESCRIPTION

Given the ‘‘harmonic constants’’ for a tidal series, *hartid* computes the predicted tides for a specified time. The harmonic constants (in the format specified below) are read in from the standard input, and the predicted tides written to the standard output. The computation (described in more detail below) infers the value of small constituents from those of larger ones, so only a few are needed to give a good result. The arguments on the command line are:

year day hr min sec
: The time of the first output value, in Greenwich time (UTC); the date may be given either as day of the year, or as month and day (Gregorian calendar). There are no explicit restrictions on the range of admissable dates, but the relevant formulae for the fundamental tidal arguments will be increasingly inaccurate before 1700.

nterms The number of values to be written out

samp The sample interval, in seconds.

HARMONIC CONSTANTS

The harmonic constants are often prepared by *harprp*. The format of this file is:

1. The first line should contain either the letter ‘‘l’’ if the phases will be given relative to the local potential (as with the κ of traditional tidal analysis) or a ‘‘g’’ if the phases will be given relative to the Greenwich potential (as with the G of traditional tidal analysis). **NOTE,** however, that phases are to be given with lags regarded as **negative**; this is the reverse of the traditional usage.

2. If the first line contains an ‘‘l’’, the second should contain the East longitude in degrees of the locality; for Greenwich phases this entry is omitted.

3. The following lines (starting with the third for local phase, the second for Greenwich phase) contain the specification of the tidal constituent (its Cartwright code—see below), followed by the amplitude (in whatever units are appropriate) and phase (of whatever type, in degrees). These lines are read with a formatted read statement, the format being (6i2,2f9.0). The first character of each line is therefore a blank (since the first number is always a single-digit integer). The only exception is the last line, which should simply contain a ‘‘−1’’ at the beginning; this signals the end-of-list to the program. A typical (though short) file might look like:

```
l
-116.455
 1-1 0 0 0 0 7.41000 -78.0
 1 1 0 0 0 0 11.7900 -94.0
 2 0 0 0 0 0 16.0600 -287.0
 2 2-2 0 0 0 5.01000 -282.0
-1
```

The program expects to be provided with at least one harmonic constant for both the diurnal and semidiurnal bands. Long-period tides (excluding semiannual and longer periods) will be included if any harmonic constants are provided for long-period tides; similarly, if the constants are given for M_3, terdiurnal tides will be predicted.

THEORY

In computing a series, *hartid* uses a larger number of tidal constituents than are input; the amplitudes of these additional constituents are found by spline interpolation of the admittance (the ratio of the potential amplitude to that observed, or in this case found from the loading computation). This automatically allows for the various nodal modulations. The fits are different for each species; note that if there are 3 or fewer harmonics for a species, a piecewise straight-line fit is used. A total of 79 long-period, 154 diurnal, and

109 semidiurnal constituents are used in the prediction (but no overtides). This corresponds to a cutoff 0.076% of the largest long-period tide (M_f), 0.013% of the largest diurnal tide (K_1), and 0.008% of the largest semidiurnal tide (M_2). The rms accuracy should be about this, but at some times smaller missing constituents could combine to give a larger error. It should be noted that a number of the constituents are in fact nodal ‘‘satellites’’ to larger ones. Also, the long-period constituents include the nodal tide, which can produce an apparent nonzero mean if any long-period tides are specified.

TIDAL CODES

The Cartwright codes and Darwin symbols for the largest tides are:

0 2 0 0 0 0 M_f	0 1 0−1 0 0 M_m

1 1 0 0 0 0 K_1	1−1 0 0 0 0 O_1	1 1−2 0 0 0 P_1	1−2 0 1 0 0 Q_1	1 2 0−1 0 0 J_1	1 0 0 1 0 0 M_1
1−2 2−1 0 0 ρ_1	1 3 0 0 0 0 OO_1	1−3 0 2 0 0 $2Q_1$			

2 0 0 0 0 0 M_2	2 2−2 0 0 0 S_2	2−1 0 1 0 0 N_2	2 2 0 0 0 0 K_2	2−1 2−1 0 0 ν_2	2−2 2 0 0 0 μ_2
2 1 0−1 0 0 L_2	2 2−3 0 0 1 T_2	2−2−0 2 0 0 $2N_2$	2−3 2 1 0 0 ε_2	2 3 0−1 0 0 ν_2	

AUTHOR

D. C. Agnew

NAME

loadcomb − combines and scales load tide files

SYNOPSIS

loadcomb c [scale (amp) scale (phase)] [rotation]

loadcomb b [rotation]

loadcomb t [scale (amp) scale (phase)] [rotation]

loadcomb r [scale (amp) scale (phase)] [rotation]

DESCRIPTION

Loadcomb performs operations on load-tide files (as written by *nloadf* and concatenated), read from the standard input, and writes the results to the standard output. The options are:

c takes two load files, concatenated together with the *cat* command, and adds the load values in them. The header lines from each file are included in the output, except that the ‘‘station’’ line is taken to be that from the first file. If two numerical arguments are given on the command line, they are taken to be the amplitude and phase of a (complex) scaling factor that is applied to the loads in the **second** file before doing the addition. If a single numerical argument is present, it is taken to be the amount by which the reference frame is rotated from the ‘‘standard’’ one; this is equivalent to giving the direction (clockwise from East) along which the loads are to be given. An argument of 0 would thus give the loads in the original system; for one of 90 [degrees] (for example) the displacements would be given, in order, as South, West, and Up. If three numerical arguments are given, they are taken to be the scaling (applied to the second file) followed by a rotation (applied to the sum). If a rotation or scaling is performed, additional ‘‘header lines’’ (with initial letter R) are written to the load file to indicate this.

r takes a single load file and applies the scaling (and/or rotation) as specified for the previous option; additional header lines are added.

b using the first two lines of the load file, gets the station location and tidal constituent, and from these computes the body tides at this location, assuming a spherical Earth with Love numbers h = 0.6114, k = 0.3040, and l = 0.0832, and the Cartwright-Tayler constituent amplitudes. The file output includes a ‘‘model’’ line which indicates that the tides given are the body tides. The results may be given rotated to be along any azimuth, as with the previous two cases; a header line is added to show this.

t operates as with the *b* option, except that the load tides read in are added to the body tides to give the total tides. The load tide may be scaled, or the sum may be rotated, as with the previous cases; header lines are added to show this.

EXAMPLES

```
cat loadfile1 loadfile2 loadcomb c

cat loadfile1 loadfile2 | loadcomb c 1.3 −10

cat loadfile1 | loadcomb b

cat loadfile1 | loadcomb t 45

cat loadfile1 | loadcomb r 45

cat loadfile1 | loadcomb r 10 −5
```

The first line shows the most usual use: adding the results already computed in the two loadfiles. The second line shows the same, with the loads in the second loadfile being scaled by 1.3 in amplitude and shifted by −10° in phase. The third line will produce the body tides at the location given by loadfile1; the fourth line, the total (body and load) tides at the same location, in a reference frame rotated by 45° (so the 1-axis points SE). The fifth line will rotate the loads by 45°; the sixth line will scale the loads by a factor of 10, with a −5° phase shift.

SEE ALSO

harprp(1), which may be used to compute the strains and tilts at a particular azimuth.

AUTHOR

D. C. Agnew

NAME

mapcheck − examine fine-scale land-sea database

SYNOPSIS

mapcheck r lat long dist

mapcheck h lat long

mapcheck o lat long

DESCRIPTION

The program **mapcheck** is designed to read from the fine-scale land-sea data file to allow the user to examine what this file shows in the vicinity of a particular point. Since this file is used to determine the distribution of the tide close to a station for loading computations, the accuracy of it can be important. The file (which is hardwired into the program with the name *lndsea.dat*) must be present in the directory in which the program is being run. The program has three options, all of which require that station coordinates (latitude and longitude) be included on the command lines. The options are:

r Write to standard output a rough ‘‘printer-plot’’ of the land and sea distribution within a distance *dist* of the station; *dist* should be in kilometers. In this plot, - means land.

r Write to standard output a ‘‘printer-plot’’ of a 1° by 1° area to show the local structure of the land-sea file. The lower left corner of the area will be at the integer latitude and longitude closest to the values given on the command line. The file has a resolution of 1/64° and what is shown is a plot with 64 by 64 points; each point in the file thus corresponds to 1 ‘‘pixel’’ of the database.

o The program writes out the distance to the nearest ocean from the point specified on the command line.

DATA FILE

The data file is a binary representation of the land-sea distribution of the entire world, based on the version created from the World Vector Shoreline for Version 3 of the GMT mapping tools. See the supplementary documentation to the loading programs for more details.

AUTHOR

D. C. Agnew

NAME

nloadf − compute ocean loads at a site, or give the grid used to do so

SYNOPSIS

nloadf stname lat long ht modelfile greenfile **g | l** [polyfile [+ | −]]

nloadf stname lat long ht modelfile greenfile **m** [dist] [polyfile [+ | −]]

DESCRIPTION

Nloadf is a program for computing ocean tidal loads at a location, given (1) a model of the ocean tides specified on a latitude-longitude grid; (2) a Green function for the response of the earth (which also includes the information about the intervals of integration); (3) a file specifying the distribution of land and sea; and (4) a ‘‘polygon file’’, which specifies what regions are and are not to be included in the convolution. Items (1-3) are required, the first two being given on the command line and the third being hardwired into the program. Item (4) is optional, and when present is specified on the command line.

The arguments on the command line are (with R for required, O for optional):

stname (R) an ascii string givening the station name; it must not contain any blanks or be longer than 80 characters.

lat (R) the North latitude (in decimal degrees) of the station.

long (R) the East longitude (in decimal degrees) of the station.

ht (R) the elevation (in meters above or below sea level) of the station.

modelfile

(R) the name of a file specifying the tidal height for a global of regional model of a particular tidal constituent, in ‘‘standard form’’. (See the supplementary documentation for a fuller description).

greenfile (R) the name of a file specifying the Green function for a particular Earth model, along with the distance range(s) and step size(s) to be used in the convolution, and whether or not the land-sea file is to be used to determine if ocean is present or not. (See the supplementary documentation for a fuller description).

g, **l**, **m** (R) determines the nature of the output. If **g** or **l**, the loading results are written to standard output. A **g** means that the phases are Greenwich phase, **BUT with lags negative** (the reverse of the usual phase convention in ocean tides). An **l** means that the the phases are in local phase at the site, again with lags negative. An **m** causes the program to write to standard output the coordinates of the corners of the cells used for the convolution, in a form (lat and long separated by impossible values) that is often suitable for plotting. This output is quite voluminous and should always be redirected to a file.

dist (O) If the **m** option is chosen, this is the maximum distance (in degrees) to which to cell coordinates will be given.

polyfile (O) The name of a ‘‘polygon file’’ which specifies regions into which points must (and/or must not) fall in order to be included in the convolution. By setting such files up correctly, it is easy to assure that a convolution done with a global model does not include a marginal sea for which a separate model is being used (and that the model of the marginal sea has no overlap with the global model). If a + or − is included following the name of this file, it overrides the instructions in that file for which polygons to include: if a + is given, all polygons in the file are included; if a −, all are excluded. (See the supplementary documentation for a fuller description).

OUTPUT

The program writes the results of its computations to standard output, along with a set of ‘‘header lines’’ which describe what went into the computation. These header lines are usually more voluminous than the results; experience has shown that it is easy to lose track of how a computation is done, so that it is worthwhile to make the results self-documenting. Both result and header lines are identified by a letter in the first column, lowercase for the result lines. The ones currently output are:

S The station name and coordinates; format (3x,a40,3x,2f10.4,f10.0).

O The tidal model: the Darwin symbol, Doodson number (in Cartwright-Tayler form), and a description of the model; format (2x,a4,5x,6i2,5x,a50).

G The Green function. The first line describes the Earth model used; subsequent lines describe what radial spacing was used, and whether the detailed land-sea model, or the ocean model, was used to determine the land-sea boundary.

P If a polygon file is used, the first line gives the description (from the polygon file) of what the file contains; subsequent lines show what polygons were included or excluded (that is, what areas within them were included or excluded).

C This code gives information about the convolution. One line gives the time at which the computation was done; a second indicates how far away, and where, the closest nonzero load was found; and a third summarizes information on how many cells (if any) were found to be ocean from the land-sea model but had an amplitude of zero from the ocean model, even after trying to interpolate from cells nearby.

L Describes the phase convention.

X A terminator line (used by *loadcomb*) to indicate the end of the header lines.

g The gravity load tide, amplitude and phase (as are all the loads), with the amplitude in microgal; format (10x,2f10.4). The convention is that positive accelerations are upwards; that is, they correspond to a decrease in the local acceleration of gravity.

p The induced potential load tides (actually, potential height), amplitude and phase, with the amplitude in millimeters; format (10x,2f10.4). The convention is that positive values correspond to upward motion of the equipotential surface. Note that this is potential height relative to the Earth's surface, not in a fixed reference frame: this is the potential which enters into the tide-generating equations.

d The displacement load tides, alternating amplitude and phase, with the amplitude in millimeters; format (10x,6f10.4). The displacements are given in order east, north, and up; note that the displacements given by some other programs (and used in the MERIT and successor standards) are south, west, and up, as well as having lags positive.

t The tilt load tides, alternating amplitude and phase, with the amplitude in nanoradians; format (10x,4f10.4). The tilts are given in order east and north, with positive tilt being that a plumb line would move in the direction given.

s The strain load tides, alternating amplitude and phase, with the amplitude in nanostrain; format (10x,6f10.4). The strains are given in order east extension, north extension, and tensor shear (e_{EN}).

EXAMPLES

```
nloadf MDO1 30.6805 255.9849 2095 m2.csr3tr green.gbav.std l

nloadf HAYS 42.617 −71.483 0 m2.bfund green.gbav.std l poly.bfund +

nloadf HAYS 42.617 −71.483 0 m2.csr3tr green.gbav.std l poly.bfund −

nloadf PFO 33.609 −116.455 1280 m2.csr3tr green.gbav.std m 20 poly.cortez −

nloadf PFO 33.609 −116.455 1280 m2.cortez green.gbav.std m 20 poly.cortez +
```

The first example shows the simplest (and usual) case where the convolution is done for a single tidal model. The next two lines show how a marginal sea (in this case the Gulf of Maine and Bay fo Fundy) would be included. On the second line the convolution is done for a model for this sea, with a polygon file being used to exclude any overlap with the global model; on the third a global model is used, with the polygon file set to exclude any overlap with the model for this sea. To get the final results, the outputs would have to be redirected to separate files and then added with *loadcom*. The fourth and fifth lines output the load-cell grids for a similar case; these are the lines used to produce the grids in Figures 1c and 1d of the manual.

SEE ALSO

loadcomb

AUTHOR

D. C. Agnew

NAME

oclook − provides information from an ocean-tide file

SYNOPSIS

oclook modelfile lat long **d | o** [polyfile]

oclook modelfile slat nlat wlong elong **g** [polyfile]

oclook modelfile slat nlat wlong elong **r | i | a | p** [polyfile]

DESCRIPTION

Oclook is writes to the standard output information about what the ocean-tide value is for a given ocean-tide file, whose name is given on the command line as *modelfile*. A file which excludes or includes regions through predefined polygons may also be included, as the last argument.

If a single position is given (as north latitude and east longitude), the output may be in one of two forms:

d In this case the program writes out complete information about the tidal data for this location. The first line of output looks like (for the M2 Schwiderski model, at 33°N, 120°W):

```
Cell indices are   241   57 (  20761) with fractional locations  −.500  .500
```

which gives the cell of the model, indexed both as a matrix (by column and row), and as a 1-d vector. The ‘‘fractional location’’ is the coordinates within the cell, relative to the center, with the cell dimensions being 1 by 1. The next lines give the tidal amplitudes in several forms:

```
Cell amps (R&I,[amp&ph]) are     −.3800     .2350
 [     .4468  148.2665 ]
Fine-grid (interpolated) amps (R&I,[amp&ph]) are     −.3973     .2105
 [     .4496  152.0813 ]
```

where the first of each pair of lines gives the real and imaginary part, and the second the amplitude (in meters) and Greenwich phase. **NOTE** that in this, alone amongst these programs, the phase is **positive** for a lag, thus corresponding to the phase convention usually used for ocean tides: this is thus, in the usual notation, *G*. The amplitude is given first for the value in the cell, and secondly for the result of bilinear interpolation to the point given.

o In this case the program writes out information in the style of a ‘‘load file’’ produced by *nloadf*, but with the only ‘‘load’’ being the tidal height in meters; for the example given above this line of output would be

```
          .4496 −152.0813
```

Note that the phase is now given in the other convention (lags negative), and that this is the interpolated value.

If a range of latitudes and longitudes are given the program lists parameters related to the cells which fall within the specified range. The options are:

g Write to standard output the coordinates of the corners of the cells (ordered as longitude and latitude), with each set of corner coordinates (five in all) separated by a geographically impossible value. When plotted, these values will thus draw the grid of cells. This output is quite lengthy and should be sent to a file.

r Write to standard output the coordinates of the centers of the cells, each with the real value of the tide (in meters) for that point.

i Write to standard output the coordinates of the centers of the cells, each with the imaginary value of the tide for that point (in meters, Greenwich phase, lag **positive**).

a Write to standard output the coordinates of the centers of the cells, each with the amplitude of the tide (in meters) for that point.

p Write to standard output the coordinates of the centers of the cells, each with the phase of the tide for that point (in degrees, Greenwich phase, lag **positive**).

For example, running the program with option **p**, and the range being from 32°N to 34°N, −122°E to −120°E, for the Schwiderski M2 model, gives:

```
−121.5000   33.5000   165.141
−121.5000   32.5000   159.497
−121.5000   31.5000   153.919
−120.5000   33.5000   159.507
−120.5000   32.5000   150.027
−120.5000   31.5000   149.477
−119.5000   33.5000   151.013
−119.5000   32.5000   148.267
−119.5000   31.5000   142.160
```

The next to last line gives the phase shown for the **d** option above.

The next two lines show the commands used to generate the model grids shown in Figures 1a and 1b of the supplementary documentation.

```
oclook m2.csr3tr 20 32 −117 −105 g
```

```
oclook m2.cortez 20 32 −117 −105 g
```

AUTHOR

D. C. Agnew

NAME

polymake − makes polygon files from a specified set

SYNOPSIS

polymake

DESCRIPTION

Polymake reads, from standard input, a file with names of polygons and a code to indicate if they are areas to be included or excluded from the loading calculation. The program writes, to standard output, a ‘‘polygon file’’ with coordinates, for use by *nloadf*. If the input file contains names not in the standard set (see below), or if the include/exclude code is incorrect, the program writes a message to the error output but does not stop.

Each line of the input file must begin with a `+` (for inclusion), `-` (for inclusion), or `#` (for a comment line), In non-comment lines this must be followed by a space, and then the name of the polygon.

If polygons overlap, the one or ones that are to be excluded take priority; this is is done in *nloadf*.

The ‘‘standard set’’ of these has names that match various regional models:

1	`osu.bering.2010`	Bering Sea
2	`osu.hawaii.2010`	Pacific Ocean around Hawaii
3	`sfbay`	San Francisco Bay
4	`osu.usawest.2010`	West coast of United States and British Columbia
5	`cortez`	Sea of Cortez/Gulf of California
6	`osu.gulfmex.2010`	Gulf of Mexico
7	`osu.hudson.2010`	Hudson Bay and surrounding waters
8	`osu.namereast.2010`	East coast of North America, Maryland to Labrador
9	`osu.patagonia.2010`	Patagonian shelf
10	`osu.amazon.2010`	Off the mouth of the Amazon
11	`osu.europeshelf.2008`	NW European shelf
12	`osu.mediterranean.2011`	Mediterranean and Black Seas
13	`osu.redsea.2010`	Red Sea
14	`osu.persian.2010`	Arabian Sae and Persian Gulf
15	`osu.bengal.2010`	Bay of Bengal
16	`osu.chinasea.2010`	East China Sea, South China Sea
17	`osu.northaustral.2009`	North of Australia, Indian Ocean to Tasman Sea
18	`osu.tasmania.2010`	Bass Strait and parts of the Tasman Sea and Great Australian Bight
19	`osu.okhotsk.2010`	Seas of Okhotsk and Japan, NE Pacific
20	`naoreg`	Sea of Japan area (used in GOTIC package)
21	`esr.aotim5.2004`	Arctic Ocean, and part of the North Atlantic
22	`esr.cats.2008`	Southern Ocean and Anarctic waters

EXAMPLE

The following uses the ‘‘here document’’ formulation available for shell scripts:

```
polymake << EOF > tmp.poly
+ esr.aotim5.2004
- osu.hudson.2010
- osu.bering.2010
EOF
```

This will include all points inside the Arctic Ocean polygon, **except** those that also fall into the Bering Sea or Hudson polygons; so we would use this when running *nloadf* on the Arctic Ocean model. To use the the Arctic Ocean model where it overlaps with the Hudson’s Bay model, we would simply omit the line for the Hudson’s Bay polygon in the above; and when running *nloadf* on the Hudson’s Bay model, would create a polygon with:

```
polymake << EOF > tmp.poly
- esr.aotim5.2004
+ osu.hudson.2010
EOF
```

to exclude the overlapping area. It might appear that there is no reason to include the polygon for a model (with a `+`) since only that model will be used by *nloadf*. However, since in some cases the ocean boundaries of the polygon and the model are not the same, always including the polygons makes sure that no cells are skipped.

AUTHOR

D. C. Agnew