Title
SPOTL: Some Programs for Ocean-Tide Loading

Permalink
https://escholarship.org/uc/item/954322pg

Author
Agnew, Duncan Carr

Publication Date
2012-03-28

SPOTL: **S**ome **P**rograms for **O**cean-**T**ide **L**oading

Duncan Carr Agnew
Institute of Geophysics and Planetary Physics
Scripps Institution of Oceanography
University of California
La Jolla CA 92093-0225 USA
`dagnew@ucsd.edu`

Program Version 3.3.0
—
Scripps Institution of Oceanography
Technical Report
March 28, 2012

1 Introduction

The increasing precision of geodetic measurements has made the effects of loading by ocean tides (or other sources) important to a wider range of researchers than just the earth-tide community. Computing such loading effects has, however, remained a rather specialized activity. This collection of programs aims to make it easy to compute load tides, or, with slight modifications, the effects of other loads.

Given that the most accurate representations of the ocean tides require both global and regional models, my aim has also been to make it easy to combine different tidal models, and to use different Earth models (though the method is restricted to spherically symmetric ones). Especially for the global ocean tide there are many models available; this package provides a set of current models found using different methods.

The package also includes programs to allow the computed loads (or the ocean tide) to be converted into harmonic constants, and to compute the tide in the time domain from these constants. For completeness a program for direct computation of the body tides is included; while its accuracy is not as high as that of some others (for example Merriam (1992)), it should be more than adequate for representing any but (perhaps) gravity-tide measurements with low-noise instruments.

This package can actually be used to find the surface effects of any load, so long as these effects are from elastic deformation, which is appropriate for any load with a time constant shorter than years: for example, changing reservoir water levels, seasonal groundwater changes, and non-tidal ocean loading. Such loads need to be put in the format used for the tides (Section 3.3) and (for loads on land) the Green-function files must be slightly modified. See Section 6 for details, and Section 2.6 for an example.

Most of the information on how to run individual programs, with simple examples, is given on the manual pages that accompany this document. Section 2 provides some examples of how the programs may be combined to carry out more complex tasks. For many users, the rest of this manual should be necessary only for reference.

Section 3 describes the file formats; the parts of interest to most users will be Section 3.1 (on “polygon files”) and Section 3.2, which describes the Green functions available. Section 4 provides details about how these Green functions were computed. Sections 5.1 and 5.2 describe, briefly, the global and local tidal models; some of this description has been taken directly from material written by the model developers.

1.1 Latest Changes

This distribution is labeled as Version 3.3.0. It has an almost completely new set of global and local models and polygon files; the examples in the manual have also been revised. It also includes bug fixes:

1. The interpolation from nearby cells was incorrect in the case that three cells on the north and east sides were used. (Thanks to Machiel Bos and Simon Williams)

2. An error in `oclook` which caused more complete dumps of the information than requested. (Thanks to Simon McCluskey).
3. Errors that caused compilation problems or warnings in `hartid` and `lodout`; also minor changes in the installation scripts. (Thanks to Simon McCluskey, Kathleen Hodgkinson, Mirko Scheinert, and Andrew Barbour).

and additions and changes to the programs:

1. A new program `polymake` has been added to make it easier to construct polygons that include or exclude particular models where they overlap.
2. The output file gives more details about which polygons are included or excluded.
3. The Green-function files have been extended in two ways. First, Green functions are available for two reference frames, one corresponding to the overall center of mass, the other to the surface of the solid Earth. Second, there are files that include smaller cells running closer to the origin. See Sections 3.2 and 4 for more information.
4. Options (input via the Green-function files) have been added to allow computation of loads with an arbitrary location (on land or not) and to restrict the load calculation to land-based loads only; see Section 6.
5. The seawater density is found using a global database; see Section 5.4.
6. The program `hartid`, for computing the time-domain tides, uses more harmonic constituents (79 long-period, 154 diurnal, 109 semidiurnal) for higher accuracy (better than 0.1%); this addition of harmonics was originally done for a program (`hardisp`) that is part of the IERS standards.
7. The Newtonian Green function for gravity has been modified to a single expression for all distances, instead of the two approximations used before – though the accuracy of these was high enough that this change should make only a very small difference. See Section 4.3 for details.

1.2 Development and History

Version 1, developed in 1981, and not distributed, was based very loosely on the integrated Green-function load program of Goad (1980). Many of the program structures were designed to fit within the limited memory available on a PDP-11/34. Since this program was developed for research, it was made as flexible as possible; for example, although most computations used the Schwiderski (1980) ocean models, it was capable of including other models, both global and local. Because of this flexibility and the memory restrictions, this implementation required three programs just to compute the loads. This version was not distributed.

I developed Version 2 for the National Geodetic Survey in 1987, and took advantage of a larger computer to combine the three programs into one, also hardwiring the choices available in the earlier version so that the only input required was the location of the place of interest.

With the appearance of the many new, Topex/Poseidon-based, ocean tide models, it was clear that it would be useful to update the programs in a way that retains the flexibility that proved useful in Version 1 while also allowing easy “automatic” use as in Version 2. So I wrote Version 3.0, and distributed it in June 1996. In Version 3.1 (distributed in 1999) I added the induced potential to the quantities computed. The changes from 3.1 to 3.2 were (A) the inclusion of two new global models (GOT00.2 and TPXO6.2), (B) the revision of the local models for Canadian waters using an improved land-sea database, and (C) an improved Antarctic coastline. Version 3.2.1 had an added regional model for the Hawaii area. Version 3.2.2 replaced TPXO6.2 with TPXO7.0.

Two of the programs included (for computing the time-domain tides) were not part of this development. The body-tide program, `ertid`, has a history (and includes some code) going back to the work of Munk and Cartwright (1966); the program comments summarize later developments. I developed `hartid`, the program for computing tides from harmonic constants (including spline interpolation of small constituents) in 1983; it was included (with fewer constituents) in Version 2.

1.3 Portability and Installation

The programs are written in standard Fortran 77. All files are read with Fortran reads and writes, either binary (for the ocean-model and land-sea files) or ASCII (for the others). One C routine is used to do bitwise AND’s for reading the bitmapped part of the land-sea database (described in more detail in Section 3.4).

The only required subroutine or function calls not standard to Fortran are to the routines `iargc` and `getarg` for reading arguments from the command line.[1] In addition, the function `fdate` is used in subroutine `lodout` to provide a time-stamp for the output of the loading program; this may be omitted if such a routine is not available. All these routines are available in most Unix-like or Linux implementations.

Options for various Fortran compilers are included in the file `Makefile` in the `src` directory; additions to this, from users who have access to other compilers, are welcome.

Installation of the programs requires the following steps

- Run `tar -xf spotl.tar` to create a directory `spotl`; all the files and directories will be put into this.
- Modify the `Makefile` in the `spotl/src` directory to have the appropriate flags for the compiler you are running.
- Move to the main (`spotl`) directory and run the `install.compile` script there. This script will compile all the routines and load them into the `bin` directory. It has been separated from the rest of the installation because this is where problems (from compiler flags not being set) are most likely.
- If the compile script runs correctly, run the `install.rest` script, again from the main (`spotl`) directory. This does the following:

---

[1] The statement `narg = iargc()` puts the number of command-line arguments in `narg`, while `call getarg(n,string)` places the characters of the `n`-th argument in character variable `string`.

  - Use `modcon` (through the script `Tobinary`) to convert the ocean models from compressed ASCII to binary.
  - Use `mapcon` to convert the land-sea database from compressed ASCII to binary (two files).
  - Link the files into the `/working` directory. This includes the Green-function files in directory `green`, the ocean-model files in directory `tidmod`, and the land-sea database in directory `lndsea`.[2]

1.4 Referencing the Package

This package was developed with support from the University of California, the US National Science Foundation, and the National Aeronautics and Space Administration. It may be copied and used without charge. It may not be modified in a way which hides its origin or removes this message or any copyright messages. It may not be resold for more than the cost of reproduction and mailing. Scientific ethics, courtesy, and completeness require the program to be referenced in any publications that use results from it; an adequate reference would be to this document (Agnew, 2012), (Agnew, 1997), a brief journal article. or the earlier version of this document (Agnew, 1996) – though this is now not easily come by.

Similar restrictions apply to the various ocean-tide models, in varying degrees; in general, these should not be redistributed without contacting their makers. Sections 5.1 and 5.2 give additional information and references.

2 Examples

These examples are intended to show how the programs can be combined into simple scripts to do different tasks; consult the individual manual pages for the programs for an explanation of the command-line arguments. These scripts, and the files produced by them, are in subdirectory `working/Exampl` in the distribution; if these are rerun in directory `working`, the results they produce can be checked against these earlier ones.

2.1 Example 1

```
../bin/polymake << EOF > poly.tmp
- cortez.1976
EOF
../bin/nloadf PFO 33.609 -116.455 1280 m2.got4p7.2004 gr.gbaver.wef.p01.ce l poly.tmp > ex1.f1
../bin/nloadf PFO 33.609 -116.455 1280 m2.cortez.1976 gr.gbaver.wef.p01.ce l poly.tmp + > ex1.f2
cat ex1.f1 ex1.f2 | ../bin/loadcomb c > ex1.f3
```

This script computes the M_2 loads at station PFO, in southern California. The first step uses

---

[2] Previous versions also placed polygon files in directory `polys`; this directory still exists but is empty, since polygons can be created more flexibly using `polymake`.

`polymake` to create a polygon that excludes the Gulf of California[3] using a “here document” (common to most shell scripts). The next line computes the load from a global model (GOT04), with the Gulf excluded using the polygon file. The second line computes the load from a separate model for the Gulf, with only the inside of the polygon included, using the + command to over-ride the setting in the `tmp.poly` file. Using the polygon in this way prevents any overlap in the computation, even if the models themselves overlap. Figure 2.1 illustrates this process. In this case, the global model covers the Gulf, but it is better to use a local model because the global one is too coarse to represent the tides adequately: a resonance near the M_2 frequency causes the tidal amplitude to increase very steeply towards the head of the gulf. The final line uses `loadcomb` to combine the two files, adding up the loads to give the total load and copying all the header lines from each file.

2.2 Example 2

```
../bin/polymake << EOF > poly.tmp
+ osu.hudson.2010
EOF
../bin/nloadf CHUR 58.759 -94.089 10 m2.osu.hudson.2010 gr.gbcont.wef.p01.ce l poly.tmp > ex2.f1
../bin/polymake << EOF > poly.tmp
- osu.hudson.2010
+ osu.namereast.2010
EOF
../bin/nloadf CHUR 58.759 -94.089 10 m2.osu.namereast.2010 gr.gbcont.wef.p01.ce l poly.tmp > ex2.f2
../bin/polymake << EOF > poly.tmp
- osu.hudson.2010
- osu.namereast.2010
+ esr.aotim5.2004
EOF
../bin/nloadf CHUR 58.759 -94.089 10 m2.esr.aotim5.2004 gr.gbcont.wef.p01.ce l poly.tmp > ex2.f3
../bin/polymake << EOF > poly.tmp
- osu.hudson.2010
- osu.namereast.2010
- esr.aotim5.2004
EOF
../bin/nloadf CHUR 58.759 -94.089 10 m2.got4p7.2004 gr.gbcont.wef.p01.ce l poly.tmp > ex2.f4
cat ex2.f1 ex2.f2 | ../bin/loadcomb c > ex2.f6
cat ex2.f6 ex2.f3 | ../bin/loadcomb c > ex2.f7
cat ex2.f7 ex2.f4 | ../bin/loadcomb c > ex2.f8
```

This example is similar to the first one, but more complicated. It shows how to combine three local models and one global one. The location (Churchill, in the Canadian Arctic) can reasonably be assumed to be affected by tides in Hudson Bay[4] and the rest of the Arctic, as well as the large tides on the east cost of North America. So `nloadf` is run on these three models, and a global model (the DTU10 model), using polygons created by `polymake` to successively include a model, and then to exclude it from subsequent calculations. Because `loadcomb` only combines pairs of files, it needs to be run three times to produce the final result.

If two regions do not overlap (as is the case here for the `osu.hudson.2010` and `osu.namereast.2010` models), it is not actually necessary to exclude one when the other is used; for example, the second

---

[3] The Sea of Cortez is another name for the Gulf of California.

[4] This is one of the few places where large tides and lower densities combine to make the seawater-density correction relatively large.

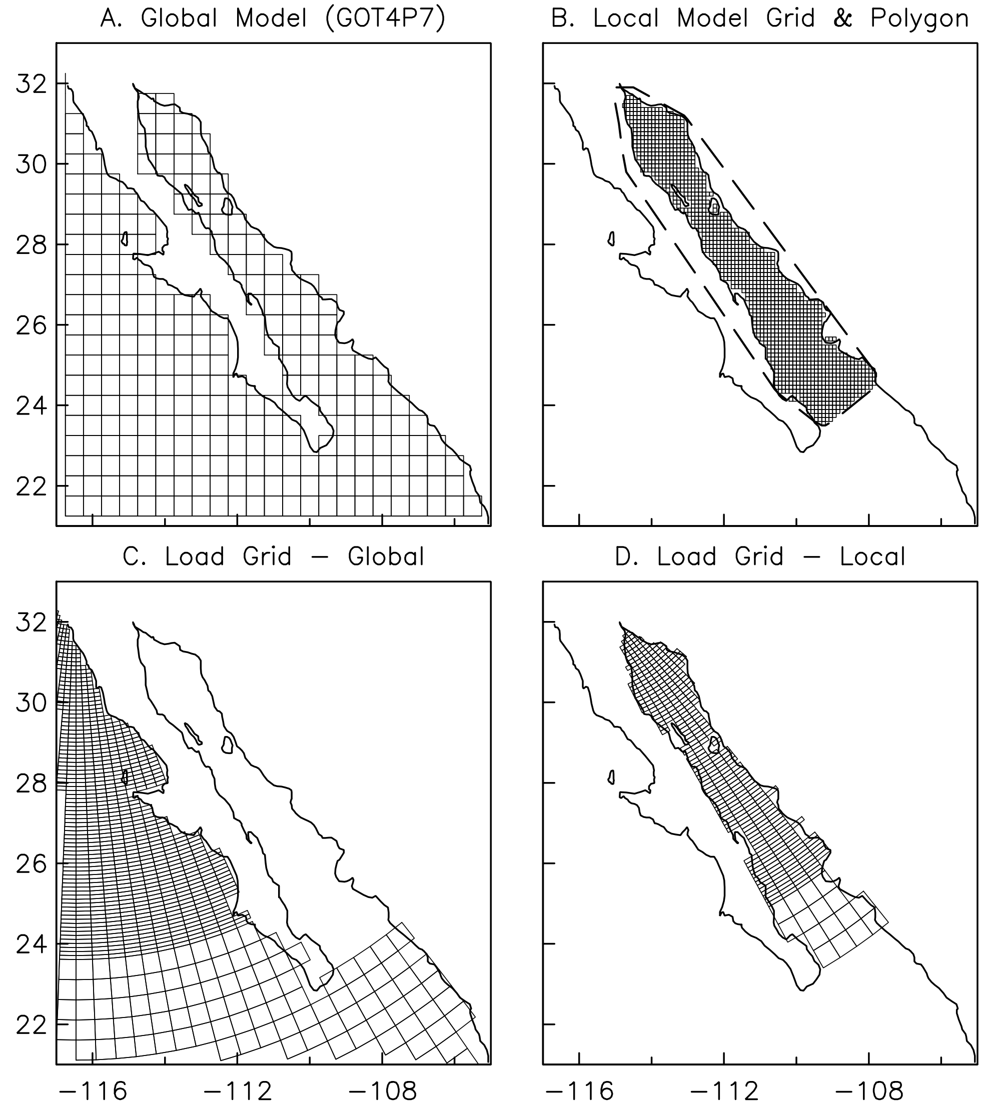

Figure 1: Frame A shows the global model around the Gulf of California; this model also covers that region. Frame B shows the local model and the polygon. Frames C and D show the grid of cells used in the loading computation: C for the global model (first use of `nloadf` in Example 2.1), and D for the local model (second use of `nloadf`).

run of `polymake` really needs only the one line `+ osu.namereast`. But always excluding other models keeps you from having to know whether or not they overlap with the one you are using.

2.3 Example 3

```
\label{sec.exampl3}
 ../bin/oclook q1.osu.usawest.2010 32.867 -117.267 o > ex3.f1
 ../bin/oclook o1.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook p1.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook k1.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook n2.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook m2.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook s2.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 ../bin/oclook k2.osu.usawest.2010 32.867 -117.267 o >> ex3.f1
 cat ex3.f1 | ../bin/harprp o > ex3.f2
 cat ex3.f2 | ../bin/hartid 1995 246 0 0 0 145 1800 >> ex3.f3
```

The first eight lines extract the complex amplitude of the ocean tide at the specified location, for all the constituents of the OSU model for the west coast of the US. This file is then piped through `harprp`, with the option set to extract the ocean-tide amplitude and turn it into a file of constituent amplitudes and phases, which in turn is sent to `hartid` to compute the actual tide at this location. The last two lines could of course be combined.

2.4 Example 4

To save space the script for this example is not printed here. The script combines Example 2.2 and Example 2.3 to compute the load tide at Churchill, with the same combination of local and global models. This is done for seven tidal constituents, after which a final file is produced of the harmonic constants for vertical displacement, and a time series from that.

2.5 Example 5

This script is also not printed here. It does the same computation as the script in Example 2.4, but uses the capabilities of the shell for compactness. If there is one argument, the script is run assuming that argument is the constituent name; if run with no arguments, it calls itself with a set of constituent names. See the script for additional explanatory comments.

2.6 Example 6

This script demonstrates how to use SPOTL to compute a non-tidal load, from a non-ocean source: specifically, the load from the uniform filling (or draining) of a large lake, the Salton Sea, in Southern California, at a nearby GPS (and strainmeter) site.

```
cat gr.gbaver.wef.p01.ce | sed ’s/F\$/L/’ | sed ’s/C\$/L/’ > tmpgr
```

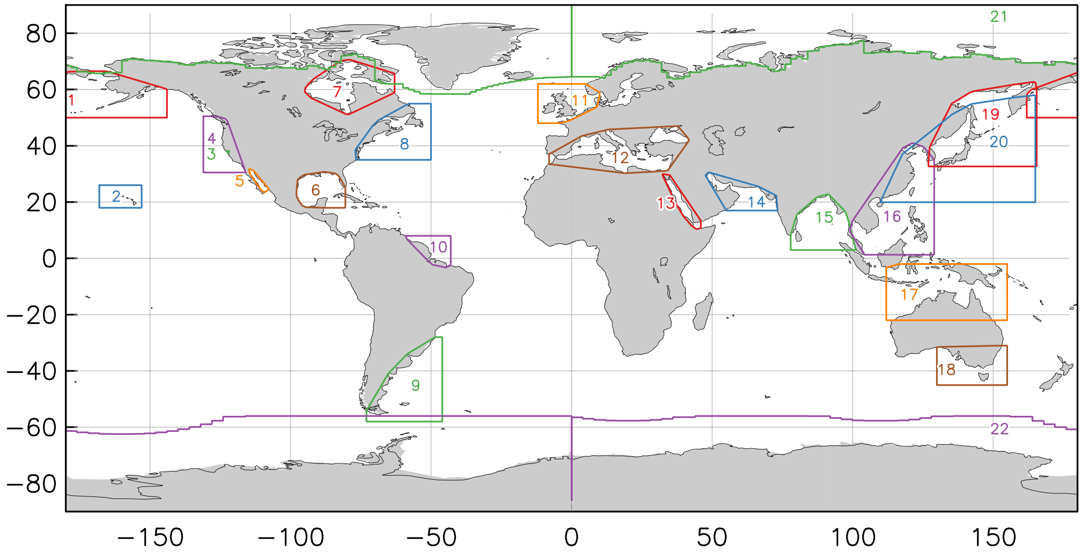

Figure 2: Polygons available using `polymake`. The numbers refer to Table 2, and are also used for the *man* page.

```
../bin/nloadf DHLG 33.3898 -115.7880 -83.0 z0.salton tmpgr g > ex6.f1
rm tmpgr
```

The script begins by creating a temporary Green-function file, `tmpgr`, which has been modified so that SPOTL will integrate only over regions on land, using the the land-sea database to make this decision (see Table 4 for the codes). Then `nloadf` is run with a model that has a uniform 1-meter load (Section 5.3). Note that we need to use the `g` option in this case to preserve the phase of the model (0°); the output file shows phases of 0 or 180 depending on the sign of the response.

3 File Formats and Information

This section summarizes the formats of the various files, and information for all but the tidal models, which are described in Section 5. With the possible exception of the “polygon files”, and the `fine` settings in the Green-function files (Table 4) these files should not require user modification,

3.1 Polygon Files

These files are designed to specify, relatively simply, a particular region, or set of regions, which is either to be the only one used in the convolution, or is to be excluded from it. As shown in Section 2, being able to include and exclude regions is useful in merging local and global models: if we specify a region that includes the local model, and which has a boundary (in part) along the overlap between local and global models, then by including and excluding this region in using the local and global models we may ensure that the total convolution has no overlap. These files are ASCII, and are formatted as shown in Table 1. The last four elements in the table (`polyname`, `npoints`, `use`, and the `xpoly, ypoly` arrays) are in the file `npoly` times.

Variable(s)	Format	Description
`filnam`	`a80`	Name appropriate to the whole file
`npoly`	`i`	Number of polygons in file (maximum of 30)
`polynam`	`a80`	Name of the polygon
`npoints`	`i`	Number of points in the polygon
`use`	`a1`	This is `+` if the polygon defines the region to be included, `-` if it defines the region to be excluded. Note that it can be overridden by use of the appropriate symbol in the command line of `nloadf`.
`xpoly,ypoly`	`*`	The vertices of the polygon, in order, given as longitude/latitude pairs (East and North positive). These coordinates may not cross a 360-degree discontinuity and must be in the range 180 to 180 or 0 to 360. It is only necessary to give each vertex once.

Table 1: Format of polygon file

Several choices are possible in deciding what to do if the file contains polygons with both excluded and included areas. The choice implemented here is that, first, if the point falls in any excluded area it is excluded; then if there are any included areas it must fall in one of them, though if there are not any included areas the point may fall anywhere outside the excluded areas.

Polygon data, in a table in the program `polymake`, is included for all the regional models; these polygons are shown in Figure 2, the numbers in which refer to Table 2; this table also gives the names to be used in the input file to `polymake`, to get these polygons included in its output.

3.2 Green-function files

The Green-function files contain the values of the integrated Green functions, specified over a given grid of radial distances. The integrated Green function is defined as:

$$G_i = a^2 \int_{\Delta_i}^{\Delta_{i+1}} G(\Delta) \sin\Delta \, d\Delta$$

where Δ is the distance from the station and G is the mass-loading Green function for the quantity of interest, as defined by Farrell (1972). (For this definition, the effect of a constant load of material of thickness A and density ρ over this distance range is then [azimuthal effects aside] $\rho A G_i$.) In principle we could choose the Δ_i's arbitrarily; in this implementation they have been chosen to be spaced at equal intervals within different distance ranges. Suppose there are M such ranges, the j-th one of which has N_j intervals, each with width (radially from the station) of $\delta_j = \Delta_i - \Delta_{i-1}$. Further define (omitting the subscript on N_j) $\Delta_L = \Delta_1 + \frac{1}{2}\delta_j = \frac{1}{2}(\Delta_1 + \Delta_2)$ and $\Delta_H = \Delta_{N+1} - \frac{1}{2}\delta_j = \frac{1}{2}(\Delta_N + \Delta_N + 1)$; the total distance coverage is thus from Δ_1 through Δ_{N+1}, with the centers of the intervals running from Δ_L though Δ_H. The overall format of the Green-function file is given in Table 3; Table 4 describes the options for one of the variables in this file, which sets the interaction with the land-sea database: something that becomes important when computing nontidal loads (Section 6).

SPOTL includes several Green-function files; the naming convention has been altered in Version

1	`osu.bering.2010`	Bering Sea
2	`osu.hawaii.2010`	Pacific Ocean around Hawaii
3	`sfbay.1984`	San Francisco Bay
4	`osu.usawest.2010`	West coast of United States and British Columbia
5	`cortez.1976`	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 Sea 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	`naoregional.1999`	Sea of Japan area (used in GOTIC2 package)
21	`esr.aotim5.2004`	Arctic Ocean, and part of the North Atlantic
22	`esr.cats.2008`	Southern Ocean and Antarctic waters

Table 2: Polygons available with `polymake`

	Variable(s)	Format	Description
A	`title`	`a70`	Identification of Green function, usually by Earth model
B	n_{gr}, j, M, N_j, Δ_L, Δ_H, δ, `fine`	`i1, i3, 2i4, 3f10.4, 5x, a1`	The variables j through δ are as defined in the text; `fine` is a character variable described in Table 4. The number of Green functions, n_{gr}, was added starting with Version 3.1,
C	$G_i(j)$	`7e13.6`	The integrated Green function (see below for the normalization), for the six load types defined in Farrell (1972): vertical and radial[5] displacement, gravitational acceleration, radial tilt, and the strains $e_{\theta\theta}$ and $e_{\lambda\lambda}$. The induced potential height was added in Version 3.1; **note** that this height is relative to the (moving) surface of the Earth (Farrell, 1973), not relative to the geocenter (Francis and Mazzega, 1990). There are N_j lines of type C, followed by another line of type B for the next range, and so on.

Table 3: Format of integrated Green-function file

Value	Water density	Land-sea interaction
`F`	Oceanic	Land-sea database determines if point on ocean or not; if not, assumes no load. Invokes bilinear interpolation (Section 3.3)
`C`	Oceanic	Source grid determines if point on land or not; if not, assumes no load. No interpolation, load is that of ocean cell.
`L`	1000	Land-sea database determines if point on ocean; if so assumes no load. Invokes bilinear interpolation (Section 3.3)
`G`	1000	Source grid determines scope of integration; if no cell, assumes no load. No interpolation, load is that of grid cell.

Table 4: Settings for the variable `fine` in integrated Green-function files. "Oceanic" densities are shown in Figure 6.

Pattern	M	j	N_j	δ	Δ_L	Δ_H	fine
`01`	4	1	98	0.01	0.025	0.995	`F`
		2	90	0.10	1.050	9.950	`F`
		3	160	0.50	10.25	89.750	`C`
		4	90	1.00	90.50	179.500	`C`
`02`	6	1	95	0.0002	0.0011	0.0199	`F`
		2	30	0.0010	0.0205	0.0495	`F`
		3	95	0.0100	0.0550	0.9950	`F`
		4	90	0.1000	1.0500	9.9500	`F`
		5	160	0.5000	10.2500	89.7500	`C`
		6	90	1.0000	90.5000	179.5000	`C`

Table 5: Cell patterns of integrated Green functions

3.3 to be `gr.mmmmmm.www.pnn,c[e|m]`.[6] The `mmmmmm` string gives the Earth model. the string `www` gives the source (who computed the function), and the number `nn` corresponds to a "pattern" of δ. The Green functions used in Version 3 are now designated pattern 01; for these, the value of δ was chosen to be comparable to the size of the land-sea grid for very close distances, and to have a spacing adequate to represent the global tides for the farther ones. Pattern 02 has a much finer grid (about 20 m for the innermost range) running to much closer to the center (about 100 m), for computations of local loads in which the land-sea database might not be used (see Section 6) – though this Green function can be used as the default with little loss of speed (it has 20% more cells than pattern 01). Table 5 describes these patterns.

The Earth model `gbaver` is the Gutenberg-Bullen Model A average Earth; `gbcont` is the same Earth model with the top 1000 km replaced by the continental shield crust and mantle structure of Harkrider (1970); `gbocen` is the same Earth model with the top 1000 km replaced by the ocean model from the same paper. The source code is `wef` for W. E. Farrell, who computed and tabulated all of these Green functions in Farrell (1972). For the `gbaver` the numbers come from the original card deck from 10/19/1971; for `gbcont` the numbers also come from cards, and for `gbocen` from the published paper.

---

[6] The installation script will create links with the old names so that existing scripts can be used. The old name `green.gbavap.std` is linked to to `gr.gbaver.wef.p01.ce`; `green.contap.std` is linked to to `gr.gbcont.wef.p01.ce`; and `green.ocenap.std` is linked to to `gr.gbocen.wef.p01.ce`.

Variable(s)	Format	Description
`dsym`	`a4`	Darwin symbol of constituent
`icte(6)`	`6i3`	Doodson number of constituent, in Cartwright-Tayler form
`latt1,latt2`	`2i8`	Integer part (degrees) and fractional part (0.001°). of the latitude of the top edge of the grid
`latb1,latb2`	`2i8`	Integer part (degrees) and fractional part (0.001°). of the latitude of the bottom edge of the grid
`lonr1,lonr2`	`2i8`	Integer part (degrees) and fractional part (0.001°). of the longitude of the right (east) edge of the grid
`lonl1,lonl2`	`2i8`	Integer part (degrees) and fractional part (0.001°). of the longitude of the left (west) edge of the grid
`latc,longc`	`2i8`	The number of cells in latitude and longitude
`modname`	`a50`	A name for the model used
`ireal()`	`10i7`	The real part of the tidal height, in integer mm, the phase being Greenwich phase with lags positive.
`imagi()`	`10i7`	The imaginary part of the tidal height, in the same conventions.

Table 6: Format of ocean tide model file.

In all these files the Green functions for the induced potential are for the Harkrider ocean model; they are as described by Farrell (1973), and taken from a listing provided by him. Again, this is not the function tabulated by Francis and Mazzega (1990).

Finally, there is the suffix, which is either `cm` or `ce`. These differ by the value of the degree-one Love numbers according to the development described in Section 4.2 and in Blewitt (2003). The `ce` suffix corresponds to choosing a reference frame coincident with the center of mass of the solid Earth; this was the definition used by Farrell (1972). The `cm` suffix corresponds to choosing a reference frame coincident with the center of mass of the load and the Earth, combined; Blewitt (2003) and others suggested that this is the more appropriate frame to use for GPS analysis, something supported by the study by Fu *et al.* (2012).

3.3 Ocean-model File Format

The ocean models are all specified on an array of cells, bounded by parallels of latitude and meridians of longitude, and all of equal size in degrees of each (though not always the same North-South as East-West). The files used are in binary, and read using Fortran sequential direct-access. For distribution the files are given in as compressed ASCII, being converted to binary through program `modcon`; this avoids any issues with byte order. The format of the ASCII file is given in Table 6.

For example, here is the ASCII version of the first lines of a file for one of the Schwiderski (1980) models, which had a 1° cell size and did not include anything south of 78°S:

```
M2
2 0 0 0 0 0
90 0
-78 0
```

```
360 0
0 0
168 360
Schwiderski 1980
```

The ordering of the cells is first from west to east, then north to south: for example, in the case given, the first cell is centered at 89.5°N, 0.5°E, the second one at 89.5°N, 1.5°E, number 361 at 88.5°N, 0.5°E, and so on.

If the setting in the Green-function file for some distance is set to `F`, the tides will be interpolated; for this purpose the values for the tides are assumed to apply to the center of each cell. The interpolated value is found by bilinear interpolation from the four cell centers closest to the point of interest. If some number of these cells have a zero value, then the values at these points are set (for the purpose of interpolation) so that bilinear interpolation is equivalent to interpolation along a plane surface (as usual, for the complex-valued amplitude).

3.4 Land-sea Database

The land-sea database shows for the whole world, where there is land and ocean, at a resolution of 1/64 of a degree (1.7 km at the Equator). This database is based on the World Vector Shoreline data, as converted to land-sea polygons for version 3.0 of the GMT (Global Mapping Tools) package (Wessel and Smith, 1996). The original form of these data set the Antarctic ice shelves to be land. To improve the representation of the Antarctic coast, coastal and grounding-line data were obtained from the Antarctic Digital Database (ADD), which is maintained as a public database for the Scientific Committee on Antarctic Research. This database has been digitized from the best available maps and photographic coverage, and covers all points south of 60°S. The coastal and grounding-line program were converted from the ADD ArcInfo format to geographic coordinates For the Antarctic ice shelves the determination of the true grounding-line is a difficult task with conventional coverage. The ADD grounding lines for these regions have been updated from recent determinations using local deformation measurements from InSAR: for the Amery Ice Shelf from H. Fricker (pers. commun.) and for the Ross Ice shelf (Siple coast) from I. Joughin (pers. commun.).

If stored as a single bit-mapped array this database would require 33.1 Mb; so save on memory this is instead stored as two arrays, each stored in a separate disk file, and read into memory at run time. The first array (stored on disk as file `landsea.ind`) covers the world at a coarse spacing (0.5°), and each element contains one of three values:

- -2 for ocean (meaning that the cell is all ocean)
- -1 for land (meaning that the cell is all land)
- A positive number if the cell has both land and ocean ("mixed"), in which case this number is the index of the cell in the second array.

The second array (stored on disk as file `landsea.bin`) contains only the "mixed" cells, at full resolution, stored in the order that they are indexed in the first array. These store the land-sea information as bits. The cell size in the coarse array was chosen to (roughly) minimize the overall storage

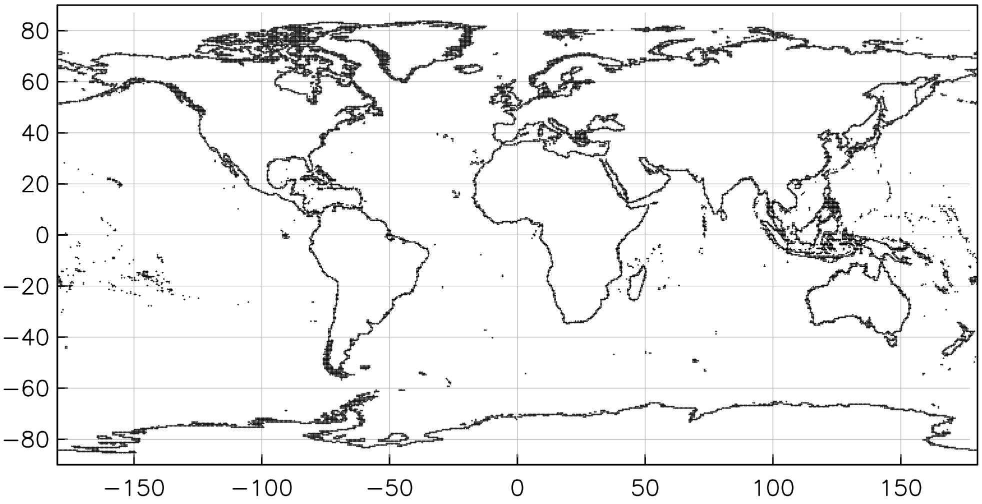

Figure 3: Locations of 0.5° cells that contain both land and ocean.

needed. As it turns out, only 16,423 (6% of the total of 259,200) of the cells are "mixed"; Figure 3 shows their locations. The two files needed are stored in directory `lndsea`. They are generated, using the program `mapcon`, from a file (`landsea.ascii.Z`) which contains both in compressed ASCII.

4 Integrated Green Function Computation

This section gives the details of how the integrated Green functions are computed. For the "Newtonian" (direct attraction) part this is done within the loading program; the analytical expressions needed are given later in this section. I first discuss the method by which the files of integrated Green functions (for the "elastic" part) are computed before distribution.

4.1 Elastic Green Functions

Goad (1980) showed how to find the integrated Green functions for gravity and displacement by forming suitable sums of Love numbers. I have instead started with the point-load Green functions computed by forming sums, since these were tabulated by Farrell (1972), who gave the Green functions for displacement, gravity, tilt, and radial strain ($e_{\theta\theta}$), normalized in the following way:

$$G_t(\Delta) = Ka\Delta G(\Delta) \qquad G_t(\Delta) = Ka^2\Delta^2 G(\Delta) \tag{1}$$

where the first equation applies to displacement, gravity, and induced potential, and the second one to tilt and strain; K is 10^{12} (SI units; 10^{18} for gravity) and a is the mean radius of the Earth, 6.371×10^6 m. The quantity we wish to compute is

$$a^2 \int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2} G(x) \sin x dx$$

We do this by renormalizing the tabulated Green functions, G_t, in the following way (again, the left is for displacement and gravity, the right for strain and tilt).

$$G'_t(\Delta) = a^2 G_t(\Delta)[2\sin(\Delta/2)/\Delta]/Ka \qquad G'_t(\Delta) = a^2 G_t(\Delta)[2\sin(\Delta/2)/\Delta]^2/Ka^2 \tag{2}$$

which for small values of Δ retains the feature of the earlier normalization of removing the singularities in G.

For displacement, gravity, or the potential, the integral is then

$$\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2} G'_t(x)\frac{sin(x)}{2\sin(x/2)}dx = G'_t(\Delta_c)\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2}\cos(x/2)\,dx = 4G'_t(\Delta_c)\cos(\Delta_c/2)\sin(\delta/4)$$

where in the first equation we have assumed that $G'_t(x)$ is constant over the interval of integration. In practice the intervals δ have been chosen sufficiently small that halving them produces essentially equivalent results; the value of G'_t at Δ_c is evaluated from the tabulated values using Lagrange interpolation.[7]

In the case of the strains and tilt, the integral is

$$\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2} G'_t(x)\frac{\sin(x)}{4\sin^2(x/2)}dx = \frac{1}{2}G'_t(\Delta_c)\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2}\cot(x/2)\,dx =$$
$$G'_t(\Delta_c)\ln\left[\frac{\sin(\Delta_c/2)\cos(\delta/4)+\cos(\Delta_c/2)\sin(\delta/4)}{\sin(\Delta_c/2)\cos(\delta/4)-\cos(\Delta_c/2)\sin(\delta/4)}\right]$$

4.2 Green Functions for Different Reference Frames

With the improvement of space-geodetic methods it became important that the reference frame for the motions induced by loading be the same one that the observations are made in. Farrell (1972) used a frame in which the center of mass of the solid Earth was held fixed (CE frame), as is appropriate for observations on the Earth. As Farrell (1972) noted, the difference between this and keeping fixed the center of mass of the solid Earth and the load, combined, (the CM frame) was given by the value for the degree-one spherical harmonic in the Love-number expressions for the Green functions. This issue was explored more thoroughly by Blewitt (2003), who introduced a variety of other reference frames depending on different choices for the degree-one Love numbers, and pointed out that a rigid-body motion was not the correct model.[8]

In most cases we can write the Green function as

$$G(\Delta) = \sum_{n=0}^{\infty}\beta_n f_n(\Delta) \tag{3}$$

where the β's are combinations of Love numbers and other constants, and the f_n's are Legendre polynomials in $\cos\Delta$ or their derivative. We can then write the different between the Green function

---

[7] In the actual program used to compute the integrated Green functions, the Green functions are read from the unnormalized form used by Farrell in his programs, and immediately normalized to the form (equation 1) given in Farrell (1972), using a subroutine provided by him, before renormalizing (equation 2) for subsequent interpolation. In this initial subroutine all three strain Green functions are also computed, using the relations in Farrell (1972).

[8] That the degree-one motion involved deformation is implied by the mention, by Farrell (1972), of strains associated with the difference between CM and CE loading – though this is easy to overlook.

for the CM and CE frames as

$$G^{CM}(\Delta) - G^{CE}(\Delta) = (\beta_1^{CM} - \beta_1^{CE}) f_1(\Delta) \tag{4}$$

because all but the degree-one terms are the same. In finding these differences for the particular case of the CM and CE frames, we can use the fact established by Blewitt (2003), that

$$h_1^{CM} - h_1^{CE} = k_1^{CM} - k_1^{CE} = l_1^{CM} - l_1^{CE} = -1 \tag{5}$$

For the vertical displacement, $\beta_1 = \frac{ah_1}{m_E}$ and $f_1 = \cos\Delta$; for the horizontal displacement, $\beta_1 = \frac{al_1}{m_E}$ and $f_1 = \frac{d\cos\Delta}{d\Delta} = -\sin\Delta$. Applied to equations (4) and (5), this means that we need to add to the CE functions the quantities $\frac{-a}{m_E}\cos\Delta$ and $\frac{a}{m_E}\sin\Delta$ respectively. For gravity $\beta_1 = \frac{g}{m_E}(1+2h_1-2k_1)$ and for tilt and surface potential $\beta_1 = \frac{g}{m_E}(1+h_1-k_1)$; so in both cases the difference between CE and CM functions is zero.

Radial strain $e_{\theta\theta}$ cannot be expressed by an expression like equation (3), but is

$$G(\Delta) = \frac{1}{m_E}\sum_{n=0}^{\infty} h_n P_n(\cos\Delta) - l_n \frac{d^2 P_n(\cos\Delta)}{d\Delta^2} \tag{6}$$

which has a degree-one term

$$\frac{1}{m_E}(h_1 + l_1)\cos\Delta$$

and so we add $\frac{-2\cos\Delta}{m_E}$ to the CE strain function for $e_{\theta\theta}$ to get the CM version.

Figure 4 shows the ratio of typical CM and CE functions. As noted by Farrell (1972) the differences can be surprisingly large for $\Delta > 10°$ – though the ratios farthest from one occur when the CE function passes through zero at a different distance than the CM one does.

4.3 Newtonian Green Functions

For a point at elevation h above sea level, the vertical gravitational attraction from a mass m an angular distance Δ away is

$$\frac{-G_N m}{a^2}\frac{\varepsilon + 2\sin^2(\Delta/2)}{[4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2]^{\frac{3}{2}}}$$

where G_N is the Newtonian constant of gravitation; ε is h/a, and we reckon positive acceleration upwards. From this expression, the Green function is

$$a^2 G = -G_N \frac{\varepsilon + 2\sin^2(\Delta/2)}{[4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2]^{\frac{3}{2}}} \tag{7}$$

This may be put into the form given by Farrell (1972) if we set $\varepsilon = 0$ and realize that the gravitational acceleration on a spherical Earth, g, is given by $G_N m_E/a^2$, which makes the Green function

$$\frac{g}{m_E}\frac{-1}{4\sin(\Delta/2)}$$

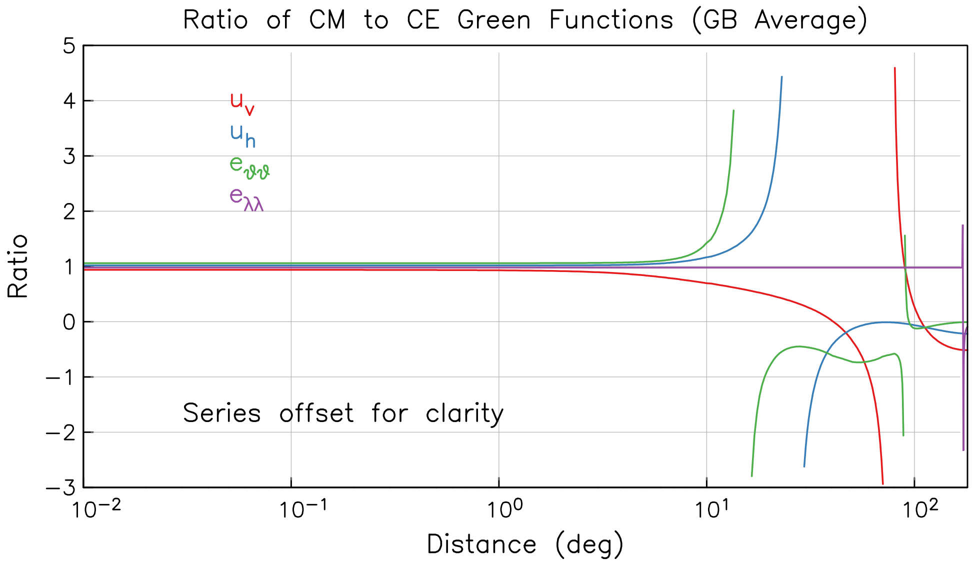

Figure 4: Ratio of the CM and CE Green functions for the Gutenberg-Bullen average Earth model. The ratios have been offset slightly to keep the lines from overlapping when the ratio is close to one.

Through version 3.2 of the software, the integral of equation (7) was found using two approximations, one for ε small and the other for Δ small enough that, to an adequate approximation, $\sin\Delta = \Delta$. It turns out that equation (7) can in fact be integrated exactly. If we change variables by introducing $u = \sin\Delta/2$, the integral of (7) becomes

$$\begin{aligned} G_N \int \frac{\varepsilon + 2\sin^2(\Delta/2)}{[4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2]^{\frac{3}{2}}} \sin\Delta\, d\Delta &= 4G_N \int \frac{2u^3+\varepsilon u}{[4(1+\varepsilon)u^2+\varepsilon^2]^{\frac{3}{2}}} du \\ &= \frac{G_N}{(1+\varepsilon)^2}\left[\frac{2(1+\varepsilon)\sin^2(\Delta/2)-\varepsilon}{\sqrt{4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2}}\right] \end{aligned} \tag{8}$$

and the integrated Green function is just the difference between the last expression evaluated at $\Delta_c - \delta/2$ and at $\Delta_c + \delta/2$.

It is somewhat informative to look at the difference between equation (8) evaluated at a distance Δ and the same evaluated at zero. If we multiply this by $2\pi\rho d$ this corresponds to the attraction from a spherical cap of density ρ and thickness d, extending from the station to a distance Δ. The results, in Figure 5, can be viewed as showing the effects of mass at two different distances. For distances from (roughly) the height of the measurement out to 20 km, the attraction is constant, up or down depending on whether the measurement is above or below the mass. To a very good approximation, this is just the attraction from an infinite flat layer. At larger distances, the curvature of the Earth means that the attraction is always down. If the measurement was made below sea level, this reduces the upward attraction from the nearby mass; if the layer covers the sphere, the attraction is reduced to zero, as it should for a measurement made inside a spherical shell. If the measurement was made above sea level, the more distant masses augment the effect of the nearby mass: for a spherical shell, the attraction (on the surface) is twice that of an infinite sheet.

The one height for which the nearby mass has no effect is if the measurement is made exactly at zero elevation. This is not ever realistic; the elevation should always be included unless there is no mass

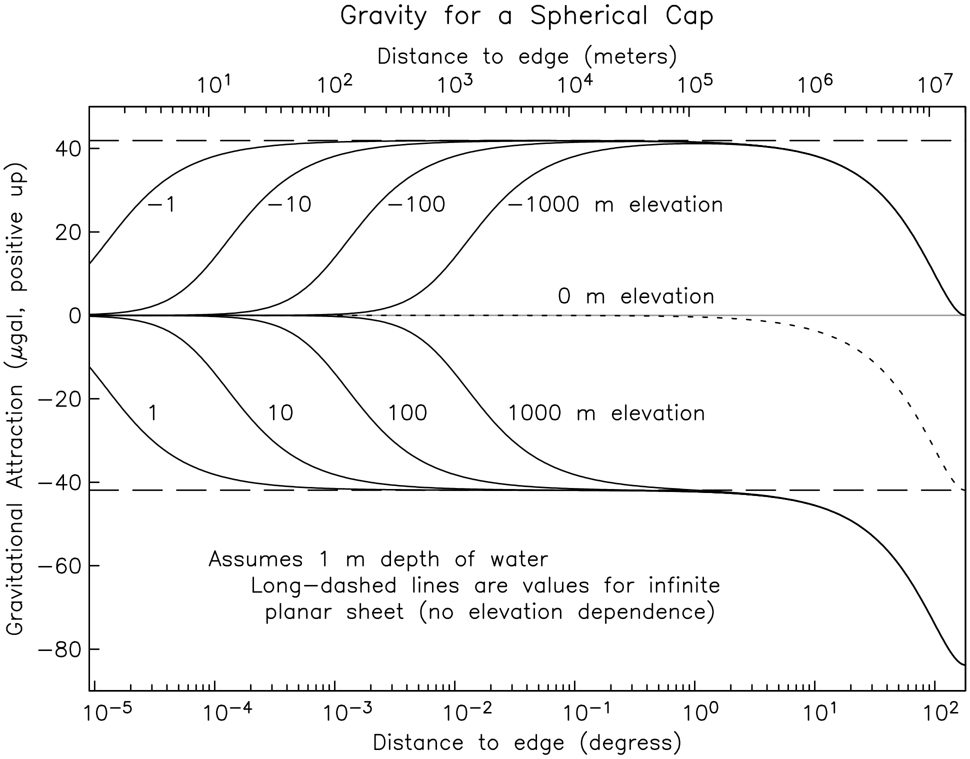

Figure 5: Newtonian gravitational attraction from a spherical cap, for measurements at various elevations; this extends the plot in Olsson *et al.* (2009) to negative (undersea) elevations.

within tens of kilometers. However, that the response is flat over a large distance range means that the Green function is small over that interval: the height effect is important only for loads within distances that are a few times the height.

The Green function for tilt is derived from the exact expression for the horizontal attraction, which is

$$\frac{G_N m}{a^2}\frac{\sin(\Delta/2)\cos(\Delta/2)}{[4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2]^{\frac{3}{2}}}$$

from which the Green function (scaling by a^2/mg, where g is now the local gravitational acceleration) is

$$\frac{2G_N}{g}\frac{\sin(\Delta/2)\cos(\Delta/2)}{[4(1+\varepsilon)\sin^2(\Delta/2)+\varepsilon^2]^{\frac{3}{2}}}$$

The primary difference between this and the usual expression with ε zero is that this expression goes to zero for $\Delta < \varepsilon$; for $\Delta > \varepsilon$ we may take $\varepsilon = 0$ with little error. In that case, the integrated Green function is

$$\frac{G_N}{g}\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2}\frac{\cos^2(x/2)}{\sin(x/2)}dx = \frac{G_N}{g}\left[-2\sin(\Delta_c/2)\sin(\delta/4)+\ln\left(\frac{\tan(0.25(\Delta_c+\delta/2))}{\tan(0.25(\Delta_c-\delta/2))}\right)\right] \quad (9)$$

For small distances we may make the approximation that $\sin(\Delta) = \Delta$, in which case the integrated Green function is

$$\frac{G_N}{g}\int_{\Delta_c-\delta/2}^{\Delta_c+\delta/2}\frac{x^2}{(\varepsilon^2+(1+\varepsilon)x^2)^{\frac{3}{2}}}dx$$

which we may integrate exactly, although it is sufficient to set $1+\varepsilon=1$, in which case the integrated Green function is

$$\frac{G_N}{g}\left[\frac{-x}{(x^2+\varepsilon^2)^{\frac{1}{2}}}+\ln(x+\sqrt{x^2+\varepsilon^2})\right]_{\Delta_c-\delta/2}^{\Delta_c+\delta/2} \qquad (10)$$

We use (9) for $\Delta > 3°$, and (10) for smaller distances.

For the induced potential (expressed as height by dividing by g), the Newtonian Green function (for height zero) is

$$\frac{a}{m_E}\frac{1}{2\sin(\Delta/2)}$$

which makes the integrated Green function

$$\frac{4a^3}{m_E}\cos(\Delta/2)\sin(\delta/4)$$

Since this is needed only at sea level, there is no correction for elevation.

5 Ocean Models

5.1 Global Models

All of the global models are presented with the same latitude/longitude grid spacing as originally provided, and have simply been reformatted to present the data in the form described in Section 3.3 above. For some models based on satellite altimetry a number of the nonzero cells are mostly or all on land. While this is convenient for processing altimetric data, it is undesirable for loading computations, so cells have been set to zero if they are more than 50% land, as determined from the land-sea database.

As was true in previous releases, not all the global models available have been included; my aim is to provide a range of current models. Many older global models are available on request. For this release the oldest models are the 1999 NAO models, included to allow comparisons with the loading program GOTIC2 (Matsumoto *et al.*, 2001), which uses them. Table 5.1 summarizes the global models included in the distribution.

In general, the global models may not well match areas which may have large local tides that will be important for the loads nearby; see Ray *et al.* (2011) for a discussion of the current state of the art in local tidal modeling.

5.1.1 NAO99b

This model (see Section 7 for link) uses a combination of hydrodynamics and data assimilated from about 5 years (191 cycles) of Topex/Poseidon data; see Matsumoto *et al.* (2000). The long-period models (NAO99L) are purely hydrodynamic.

Suffix	Cell Size	Constituents												
		M_2	N_2	S_2	K_2	K_1	P_1	O_1	Q_1	S_1	M_4	M_f	M_m	S_{sa}
naoglobal.1999	0.5	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾			☾	☾	
fes.2004	0.5	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾	⊙	☾	☾	☾	
got4p7.2004	0.5	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾	⊙				
ray.equil.2004	0.5											☾	☾	⊙
osu.tpxo72.2010	0.125	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾		☾	☾	☾	
dtu10.tr.2010	0.125	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾	⊙	☾			
osu.tpxo72atlas.2011	0.125	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾		☾	☾	☾	
eot11a.2011	0.125	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾	⊙	☾	☾	☾	
hamtide11a.2011	0.125	☾	☾	⊙	☾⊙	☾⊙	☾	☾	☾					

Table 7: Global tide models. Cell size is in degrees. Astronomical symbols show if the tide is lunar (☾), solar (⊙), or both.

5.1.2 FES04

This model (see Section 7 for link) is the most recent version (FES 2004, version 1.0.2) of the models produced by the “French Tidal Group”: originally Christian Le Provost and his collaborators at the Laboratoire des Ecoulements Geophysiques et Industriels, Institut de Mecanique de Grenoble. The first model was FES 94.1, computed using a finite-element hydrodynamic model with variable mesh size Le Provost *et al.* (1994); the FES95.2 model (Le Provost *et al.*, 1998) adjusted this using the CSR2.0 solution Eanes (1994), which was derived from the first year of Topex/Poseidon data. This was followed by a refined hydrodynamic model, FES98 (Lefevre *et al.*, 2000), another combined model, FES99 (Lefevre *et al.*, 2002), which assimilated tide gauge data, and finally FES04 (Lyard *et al.*, 2006), which uses a refined mesh and assimilates tide gauge, Topex/Poseidon, and ERS data. The long-period and M_4 tides are from a hydrodynamic model. These data are for scientific use only; for commercial use, contact T. Letellier (`tletelli@notos.cst.cnes.fr`) or Laurent Roblou (`laurent.roblou@legos.obs-mip.fr`).

5.1.3 GOT04 and Equilibrium Models

GOT4.7 is the 2004 version of the Goddard Ocean Tide Model, produced by Richard Ray based on 364 cycles of Topex/Poseidon, plus 114 cycles along the T/P interleaved groundtrack, plus also ERS-1/2 and GFO data in shallow and polar seas, and a very small amount of Icesat data. See Ray (1999) for the methods used.

The model for the S_1 tide comes from a combination of altimetric data and a hydrodynamic model; see Ray and Egbert (2004) for details. Note that loads computed using this tide **will not be meaningful**, since any such computation will not include the effect of the atmospheric S_1 pressure tide; see Ponte and Ray (2002).

The models for the equilibrium long period tides were computed by Ray using a spherical-harmonic method that includes self-attraction and loading (Egbert and Ray, 2003) all these models are of course just scaled versions of the same elevation field. All models were provided by Dr. Ray (pers. commun.).

5.1.4 TPXO7.2 and ATLAS

This model, the latest of a long series, was produced at Oregon State University by S. Y. Erofeeva and G. Egbert; see Section 7 for their website. As the description there states, this is "a global model of ocean tides, which best-fits, in a least-squares sense, the Laplace Tidal Equations and along track averaged data from TOPEX/Poseidon and Jason (on TOPEX/POSEIDON tracks since 2002)." That is, this is a hydrodynamic model with altimetry data assimilated. For a description of the methodology, see Egbert *et al.* (1994) and Egbert and Erofeeva (2002).

The ATLAS version of this model uses the same methods, and combines three basin-wide models (one each for the Atlantic, Pacific, and Indian Oceans, blended with higher-resolution local models (many of them included in SPOTL) produced by the same group.

5.1.5 DTU10, trimmed

This model (see Section 7 for link) is a shallow-water extension and adjustment of FES 2004 using 17 years of Topex/Poseidon, Jason-1, and Jason-2 data; see Cheng and Andersen (2010). In its original form this model had nonzero cells over land areas; any cell with more than 50% land (according to the land-sea database) was set to zero.

5.1.6 EOT11A

This model (see Section 7 for link) is altimetry-based, with data from Topex/Poseidon, Jason1, Jason-2, ENVISAT, and ERS 1 and 2. Data were combined and a harmonic analysis was used to determine corrections to FES 2004. See Savcenko and Bosch (2008) for a description of the method as applied to produce an earlier model.

5.1.7 HAMTIDE11A

This model (see Section 7 for link) from the Institut für Meereskunde of Hamburg University assimilates the multimission altimetry data used in the EOT models into an inverse model that also includes hydrodynamics. See Taguchi *et al.* (2011).

5.2 Local Models

Table 5.2 summarizes the local models (for their locations see Table 2 and Figure 2). Most of these models (the ones starting with `osu`) come from S. Y. Erofeeva and G. Egbert at Oregon State, who have produced them by assimilating altimetry data, and checked them against tide-gauge data. See the section on the TPXO global model for general references and a web address; an additional reference, for the model for the region north of Australia, is Ray *et al.* (2011).

The two models for the polar regions come from L. Padman and his collaborators at ESR, Inc, Oregon State, and Scripps; see Section 7 for their website. The model for the Arctic comes from

Suffix	Cell Size	Constituents										
		M_2	N_2	S_2	K_2	K_1	O_1	P_1	Q_1	M_4	M_m	M_f
`osu.bering.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.hawaii.2010`	1	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾		☾	☾
`osu.usawest.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`cortez.1976`	2	☾	☾	⊙	☾ ⊙							
`sfbay.1984`	2	☾		⊙		☾ ⊙	☾					
`osu.gulfmex.2010`	1.33	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.hudson.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.namereast.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.patagonia.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾	☾		
`osu.amazon.2010`	1	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾	☾		
`osu.europeshelf.2008`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾	☾		
`osu.mediterranean.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.redsea.2010`	1	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.persian.2010`	1	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.bengal.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.chinasea.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾	☾		
`osu.northaustral.2009`	2.5	☾		⊙		☾ ⊙	☾			☾		
`osu.tasmania.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`osu.okhotsk.2010`	2	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾	☾		
`naoregional.1999`	30	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`esr.aotim5.2004`	6×3	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾			
`esr.cats.2008`	6×3	☾	☾	⊙	☾ ⊙	☾ ⊙	☾	☾	☾		☾	☾

Table 8: Local and regional ocean-tide models. Cell size is in minutes of arc, and if two dimensions are given, is EW by NS.

their model AOTIM-5 (Padman and Erofeeva, 2004), for which the M_2, S_2, K_1 and O_1 tides used data assimilation from tide-gauge and altimetry data; the other tides are based on a forward model. The model for the Antarctic, CATS2008a, uses data assimilation applied to altimetry data (including ICESAT data from the Ross and Filchner-Ronne ice shelves) and selected data from bottom pressure measurements; see Padman *et al.* (2002) and Padman and Fricker (2005) for more information.

Both of these models were computed on a rectangular grid (5 km for AOTIM-5 and 4 km for CATS2008a. For use in SPOTL the models were interpolated, using bilinear interpolation, to the centers of latitude/longitude cells of the sizes shown.

The NAO regional model is NAO99Jb, used with GOTIC2, and was produced using the same methods as NAO99b; see Matsumoto *et al.* (2000).

For San Francisco Bay the tides were interpolated between observed values around the Bay, taken from Cheng and Gartner (1984); a comparison with the model results of Cheng *et al.* (1993) shows that the M_2 amplitude is within 5 cm almost everywhere, and often better: 10% error would probably be conservative. The boundaries of the Bay were defined using a detailed coastline file.

For the Gulf of California (Sea of Cortez), the original grid for the three-dimensional hydrodynamic model of Stock (1976) was spline interpolated onto a latitude-longitude grid. (The original grid was aligned along the axis of the Gulf of California). This model has been included because it captures the M_2 resonance in this gulf; any of the global models should be usable for the diurnal tides. Stock actually computed only the M_2, S_2, and K_1 tides; the model for N_2 has been scaled from the M_2 model using the amplitude and phase differences found by Filloux (1973) from tide-gauge data in the northern Gulf.

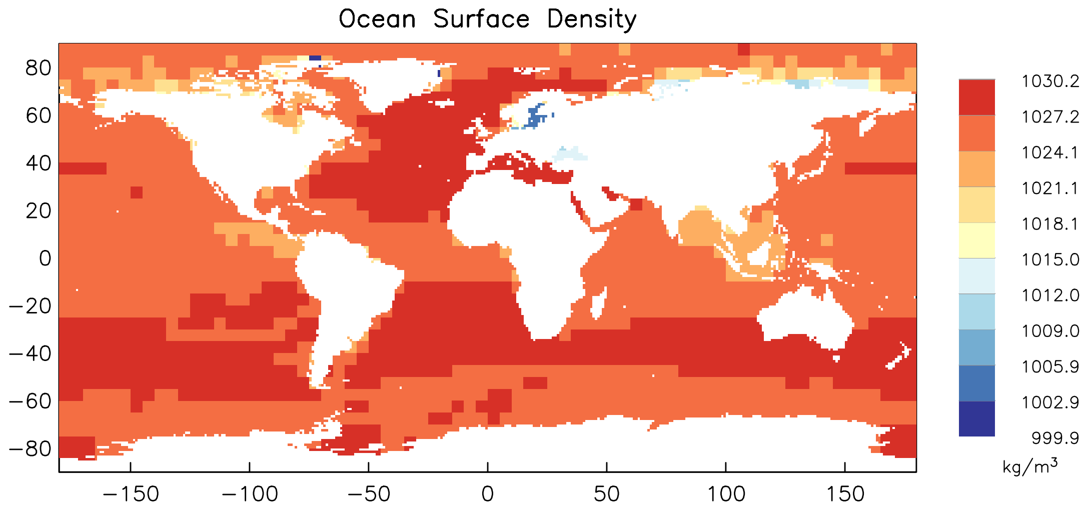

Figure 6: Map of surface densities used for the loading computation, derived from the 2009 World Ocean Atlas. The result is shown on a 1° land-sea grid; the spatial resolution of the density model is 5°. The low density of the Baltic Sea and Black Sea are the largest deviations from the average.

5.3 Uniform-load Models

One non-tidal model is included in the distribution, largely for demonstrating how SPOTL can be used to model non-tidal loads (see Section 6 and the example in Section 2.6). This model, `z0.salton`, specifies a uniform 1-m load for a 0.01° grid covering the Salton Sea, in Southern California. The phase of the load is 0°, so only the real part is nonzero.

5.4 Water Density

All tidal models give the tides in terms of water height; to convert this to mass loads requires that we assume a density. Previous versions of SPOTL used a constant density of 1025 $\mathrm{kg\,m^{-3}}$. Almost everywhere the difference between this and the actual surface density in the ocean is less than 1%, but it seemed worthwhile to eliminate this as a possible source of systematic error (Bos and Baker, 2005). Monthly averages of the surface temperature and salinity for 5° squares were taken from the 2009 World Ocean Atlas ((Antonov *et al.*, 2010; Locarnini *et al.*, 2010); see Section 7 for link) and converted to density using the International Equation of State for seawater; these densities were then combined to produce an annual average. For a few squares at high latitudes only annual averages of temperature and salinity were available; and for a few others the density was interpolated from neighboring squares. Figure 6 shows the resulting map of surface density; outside a few seas (all with small tides), the densities are close to the value previously used.

6 Using SPOTL for Non-tidal Loads

Very little in the loading computation is actually specific to the tides; anything that loads the Earth will have the same effects. Obviously, the main thing that will be different is where the load is; to compute the load for (say) loading by a reservoir, the load needs to turned into an imitation ocean model. It must be gridded into latitude-longitude form, and a file formatted in accordance with Table 6. The can then be converted to binary with `modcon`, and the loading effects computed with `nloadf`. In most cases there will not be any need, as there is for the tides, to keep track of the phase of the load; all of the imaginary parts of the loading can be set to zero. Section 2.6 shows a simple example.

If the load is on land, one other modification will be needed, which is to produce a Green-function file in which the variable `fine` is set either to `G` or to `L` for all distance ranges. This will stop `nloadf` from checking the land-sea database and excluding any regions that are land – which for a land-based load would set all the load to zero. For a purely land-based load either `G` or `L` will produce the same result. For a load (such as air pressure) that might cover both land and ocean areas, the `L` flag will exclude any cell which the land-sea database shows to be ocean, while the `G` flag will allow all cells to be included. The `L` flag would be used to exclude ocean areas because some loads do not contribute in the same way to loading in ocean and land areas: notably, air pressure, because of the "inverted barometer" effect. Both the `L` and `G` flags also assume that the load has a constant density of 1000 $\mathrm{kg\,m^{-3}}$, rather than the variable density of the ocean; this density assumption is in fact the only difference between the `G` and `C` flags.

The final adjustment that might be needed for computing local loads, at least for tilt and gravity, occurs because of SPOTL's assumption that the load has an elevation of zero. If you wanted to compute the loading from (say) a reservoir, you would need to set the station elevation to be relative to (above or below) the water level. This matters for gravity and tilt because of the Newtonian contribution.

One kind of loading that cannot not easily modeled with SPOTL is atmospheric loading for gravity and tilt, again because of the Newtonian part (density changes in the atmosphere). There is no problem with these changes being above the station – that is the same as ocean tide loads measured on the seafloor. But because the density changes are at a range of of elevations, a separate computation would be needed for each layer of the atmosphere, including its effect on surface pressure. with the station elevation set to appropriately negative amounts. Since displacements and strains do not involve direct attraction, they can easily be computed from surface pressure with (as noted above) the `fine` flag in the Green-function files set to `L` for all distance ranges.

6.1 A Note on Water Boundaries

In computing loads from lakes, reservoirs, or just loads very close to the coast, it may be necessary to have detailed representation of the land-water boundary built into your loading model, so that the `C` or `G` flags in the Green function will have the right effect. (The `C` flag would be appropriate for local tides, since it will use the local seawater density). There are many digital files of boundaries for specific water bodies. Two global ones are the GSHHS[9] database of Wessel and Smith (1996) (see Section 7 for link). The ocean shoreline is based on the World Vector Shoreline (Soluri and Woodson, 1990), which itself was based on a raster model compiled from nautical charts. This coastline can suffer from errors in position, and has some errors. The inland shorelines in GSHHS are from a variety of sources and should be used with caution. An alternative at most latitudes

---

[9] Global Self-consistent, Hierarchical, High-resolution Shoreline

is the SRTM Water Body Data, a byproduct of the SRTM mapping mission (see Section 7 for link). This gives water boundaries (coastal and inland) at 30 m resolution (higher than the available SRTM resolution for most of the work, and significantly higher than GSHHS) and is on a consistent coordinate system. Like the SRTM model itself it does suffer from gaps and voids, and so should not be used without visually checking against some other source. Google Earth usually provides high-resolution photos than can be used for this.

7 Web Links

NAO99 model `http://www.miz.nao.ac.jp/staffs/nao99/index_En.html`

FES2004 model `http://www-apache.legos.obs-mip.fr/en/soa/`

Oregon State (TPXO and local) model `http://volkov.oce.orst.edu/tides/`

DTU10 model `http://www.space.dtu.dk/English/Research/Scientific_data_and_models/Global_Ocean_Tide_Model.aspx`

EOT11A model `ftp://ftp.dgfi.badw.de/pub/EOT11a/data/`

HAMTIDE model `http://icdc.zmaw.de/hamtide.html?&L=1`

Polar tide models `http://www.esr.org/ptm_index.html`

World Ocean Atlas `http://www.nodc.noaa.gov/OC5/WOA09/pr_woa09.html`

SRTM Water Body Data `http://dds.cr.usgs.gov/srtm/version2_1/SWBD/`

GSHHS Shorelines `http://www.soest.hawaii.edu/wessel/gshhs/`

8 Acknowledgements

I thank Tonie van Dam for helping provide the land-sea mask, and also for providing a version of the CM Green functions for checking. I also thank the people mentioned in Section 1.1 for alerting me to errors, and their patience in waiting for them to be fixed.

References

Agnew, D. C. (1996), SPOTL: Some Programs for Ocean-Tide Loading, *SIO Reference Series 96-8*, Scripps Institution of Oceanography.

Agnew, D. C. (1997), NLOADF: a program for computing ocean-tide loading, *J. Geophys. Res.*, **102**, 5109–5110.

Agnew, D. C. (2012), SPOTL: Some Programs for Ocean-Tide Loading, *SIO Technical Report*, Scripps Institution of Oceanography, available at `http://escholarship.org/uc/sio_techreport`.

Antonov, J. I., D. Seidov, T. P. Boyer, R. A. Locarnini, A. V. Mishonov, H. E. Garcia, O. K. Baranova, and M. M. Zweng (2010), World Ocean Atlas 2009: Salinity, *NOAA Atlas NESDIS 69*, p. 184.

Blewitt, G. (2003), Self-consistency in reference frames, geocenter definition, and surface loading of the solid Earth, *J. Geophys. Res.*, **108**, 2103, doi:10.1029/2002JB002082.

Bos, M. S., and T. F. Baker (2005), An estimate of the errors in gravity ocean tide loading computations, *J. Geod.*, **79**, 50–63.

Cheng, R. T., and J. W. Gartner (1984), Tides, tidal and residual currents in San Francisco Bay, California: results of measurements 1979-1980,, *USGS Water-Resources Investigations Report 84-4339*, U. S. Geological Survey.

Cheng, R. T., V. Casulli, and J. W. Gartner (1993), Tidal, residual, intertidal mudflat (TRIM) model and its applications to San Francisco Bay, California, *Estuarine Coastal Shel Sci.*, **36 235-280**.

Cheng, Y., and O. B. Andersen (2010), *Improvement in global ocean tide model in shallow water regions*, Poster, SV.1-68 45 pp., OST-ST Meeting on Altimetry for Oceans and Hydrology, Lisbon.

Eanes, R. J. (1994), Diurnal and semidiurnal tides from TOPEX/POSEIDON altimetry, *Eos Trans. AGU, 1994 Spring Meeting Suppl.*, **108**.

Egbert, G. D., and S. Y. Erofeeva (2002), Efficient inverse modeling of barotropic ocean tides, *J. Atmos. Ocean. Tech.*, **19**, 183–204.

Egbert, G. D., and R. D. Ray (2003), Deviation of long-period tides from equilibrium: Kinematics and geostrophy, *J. Phys. Oceanogr.*, **33**, 822–839.

Egbert, G. D., A. F. Bennett, and M. G. G. Foreman (1994), TOPEX/POSEIDON tides estimated using a global inverse model, *J. Geophys. Res.*, **99**, 24,821–24,852.

Farrell, W. E. (1972), Deformation of the earth by surface loads, *Rev. Geophys.*, **10**, 761–797.

Farrell, W. E. (1973), Earth tides, ocean tides, and tidal loading, *Phil. Trans. Roy. Soc. Ser. A*, **272**, 253–259.

Filloux, J. H. (1973), Tidal patterns and energy balance in the Gulf of California, *Nature*, **243**, 217–221.

Francis, O., and P. Mazzega (1990), Global charts of ocean tide loading effects, *J. Geophys. Res.*, **95**, 11,411–11,424.

Fu, Y., J. Freymueller, and T. van Dam (2012), The effect of using inconsistent ocean tidal loading models on GPS coordinate solutions, *J. Geodesy*, **86**, ??–??, doi:10.1007/s00190-011-0528-1.

Goad, C. C. (1980), Gravimetric tidal loading computed from integrated Green's functions, *J. Geophys. Res.*, **85**, 2679–2683.

Harkrider, D. (1970), Surface waves in multilayered elastic media (2): higher mode spectra and spectral ratios from point sources in plane-layered earth models, *Bull. Seismo. Soc. Amer.*, **60**, 1937–????

Le Provost, C., M. L. Genco, F. Lyard, P. Vincent, and P. Canceil (1994), Spectroscopy of the world ocean tides from a finite element hydrodynamic model,, *J. Geophys. Res.*, **99**, 24,777–24,797.

Le Provost, C., F. Lyard, J. M. Molines, and M. L. Genco (1998), A hydrodynamic ocean tide model improved by assimilating a satellite altimeter-derived data set, *J. Geophys. Res.*, **103**, 5513–5529.

Lefevre, F., F. H. Lyard, and C. Le Provost (2000), FES98: a new global tide finite element solution independent of altimetry, *Geophys. Res. Lett.*, **27**, 2717–2720.

Lefevre, F., F. H. Lyard, C. Le Provost, and E. J. O. Schrama (2002), FES99: a global tide finite element solution assimilating tide gauge and altimetric information, *J. Atmos. Oceanic Technol.*, **19**, 1345–1356.

Locarnini, R. A., A. V. Mishonov, J. I. Antonov, T. P. Boyer, H. E. Garcia, O. K. Baranova, M. M. Zweng, and D. R. Johnson (2010), World Ocean Atlas 2009: Temperature, *NOAA Atlas NESDIS 68*, p. 184.

Lyard, F., F. Lefevre, T. Letellier, and O. Francis (2006), Modelling the global ocean tides: modern insights from FES2004, *Ocean Dynam.*, **56**, 394–415.

Matsumoto, K., T. Takanezawa, and M. Ooe (2000), Ocean tide models developed by assimilating Topex/Poseidon altimeter data into hydrodynamical model: a global model and a regional model around Japan, *J. Oceanogr.*, **56**, 567–581.

Matsumoto, K., T. Sato, T. Takanezawa, and M. Ooe (2001), GOTIC2: A program for computation of oceanic tidal loading effect, *J. Geod. Soc. Japan*, **47**, 243–248.

Merriam, J. B. (1992), An ephemeris for gravity tide predictions at the nanogal level, *Geophys. J. Internat.*, **108**, 415–422.

Munk, W. H., and D. E. Cartwright (1966), Tidal spectroscopy and prediction, *Phil. Trans. Roy. Soc. Ser. A*, **259**, 533–581.

Olsson, P. A., H. G. Scherneck, and J. Agren (2009), Effects on gravity from non-tidal sea level variations in the Baltic Sea, *J. Geodyn.*, **48**, 151–156, doi:10.1016/j.jog.2009.09.002.

Padman, L., and S. Erofeeva (2004), A barotropic inverse tidal model for the Arctic Ocean, *Geophys. Res. Lett.*, **31**, L02,303, doi:10.1029/2003GL01900.

Padman, L., and H. A. Fricker (2005), Tides on the Ross Ice Shelf observed with ICESat, *Geophys. Res. Lett.*, **32**, L14,503, doi:10.1029/2005GL023214.

Padman, L., H. A. Fricker, R. Coleman, S. Howard, and L. Erofeeva (2002), A new tide model for the Antarctic ice shelves and seas, *Annal. Glac.*, **34**, 247–254.

Ponte, R. M., and R. D. Ray (2002), Atmospheric pressure corrections in geodesy and oceanography: A strategy for handling air tides, *Geophys. Res. Lett.*, **29**(24), 2153, doi:10.1029/2002GL016340.

Ray, R. D. (1999), A global ocean tide model from TOPEX/POSEIDON altimetry: GOT99, *NASA Tech. Mem. 209478*, Goddard Space Flight Center, Greenbelt, MD, USA.

Ray, R. D., and G. D. Egbert (2004), The global S_1 tide, *J. Phys. Oceanogr.*, **34**, 1922–1935.

Ray, R. D., G. D. Egbert, and S. Y. Erofeeva (2011), Tide predictions in shelf and coastal waters: status and prospects, in *Coastal Altimetry*, edited by S. Vignudelli, A. G. Kostianoy, P. Cipollini, and J. Benveniste, pp. 191–216, Springer-Verlag, New York.

Savcenko, R., and W. Bosch (2008), EOT08a - empirical ocean tide model from multi-mission satellite altimetry, *DFGI Report 81*, Deutsches Geodätisches Forschungsinstitut (DGFI), Munich.

Schwiderski, E. W. (1980), On charting global ocean tides, *Rev. Geophys.*, **18**, 243–268, doi:10.1029/RG018i001p00243.

Soluri, E. A., and V. A. Woodson (1990), World Vector Shoreline, *Internat. Hydrogr. Rev.*, **68**, 28–35.

Stock, G. (1976), Modeling of Tides and Tidal Dissipation in the Gulf of California, Ph.D. thesis, University of California, San Diego, La Jolla.

Taguchi, E., D. Stammer, and W. Zahel (2011), Estimation of deep ocean tidal energy dissipation based on the high-resolution data-assimilative HAMTIDE model, *J. Geophys. Res.*, **???**, in press.

Wessel, P., and W. H. F. Smith (1996), A global, self-consistent, hierarchical, high-resolution shoreline database, *J. Geophys. Res.*, **101**, 8741–8743.

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