Contents

1 Introduction

TIDES is a FORTRAN subroutine library that contains some tide generating routines.

The TIDES library has been ported to various systems. Presently it is available at DEOS/LR on the IBM Risk 6000 and the SGI servers. At NOAA it is available for Linux.

The library can be linked to any of your FORTRAN programs by specifying the library in you FORTRAN link step, *e.g.*

```
f77 example.f -o example $ALTIM/lib/tides.a
```

The TIDES library contains drivers for several routines written by other institutes. The drivers are similar in definition of their calls in order to improve exchangability.

2 Subroutine Synopsis

2.1 AIRTIDE – Compute air tides according to Ray and Ponte model

```
    FUNCTION AIRTIDE (UTC, LAT, LON)
    REAL*8   AIRTIDE, UTC, LAT, LON

Compute air tide for given time and location based on grids of the
harmonic constituents S1 and S2 determined by Ray and Ponte (2003).

The resulting value TIDE is in millibar and is part of the ocean
tide model. However, this contribution should be removed from the
sea level pressure before computing the inverse barometric correction.

The current version uses only the S1 and S2 components with amplitudes
of the order of 1 mbar. Other constituents are much smaller.

The input grids can be found in $ALTIM/data/$TIDE. The grids are
in NetCDF format and were converted using the program "got2nc".

To initialize the computation, the function AIRINIT should be
called first. It allocates the appropriate amount of memory and
loads the grids into memory. To release the memory for further
use, call DALLOCF.

Longitude and latitude are to be specified in degrees; time in UTC
seconds since 1 Jan 1985. All predicted tides are output in meters.
If the tide is requested in a point where it is not defined, NAN
(Not-a-Number) is returned.

Input arguments:
 UTC      : UTC time in seconds since 1 Jan 1985
 LAT      : Latitude (degrees)
 LON      : Longitude (degrees)

Returned value:
 AIRTIDE  : Air tide (millibars)
```

2.1.1 AIRINIT – Initialize air tide model

```
    FUNCTION AIRINIT (NAME)
    CHARACTER*(*) NAME
    INTEGER*4 AIRINIT

Allocate memory for air tide modeling and read grids into memory.

Returned value:
 AIRINIT  : Pointer to memory buffer
```

2.2 CSRTIDE – Compute tides according to CSR model

```
    SUBROUTINE CSRTIDE (UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD)
    implicit none
    REAL*8    UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD

Compute ocean tidal height for given time and location from grids
of ortho weights for one of the CSR models.
This routine is heavily based on the routine CSRTPTIDE by Richard Eanes.

For partical purposes I have changed the input to netCDF grids.
These grids can be found in $ALTIM/data/csr_tides

To initialize the computation, the function CSRINIT should be
called first. It allocates the appropriate amount of memory and
loads the grids into memory. To release the memory for further
use, call DALLOCF.

Longitude and latitude are to be specified in degrees; time in UTC
seconds since 1 Jan 1985. All predicted tides are output in meters.
If the tide is requested in a point where it is not defined, NAN
(Not-a-Number) is returned.

Input arguments:
 UTC      : UTC time in seconds since 1 Jan 1985
 LAT      : Latitude (degrees)
 LON      : Longitude (degrees)

Output arguments:
 TIDE     : Predicted short-period tide (m)
 TIDE_LP  : Predicted long-period tide (m)
 TIDE_LOAD: Predicted loading effect (m)
```

2.2.1 CSRINIT – Initialize CSR tide model

```
    FUNCTION CSRINIT (NAME, WANTLOAD)
    implicit none
    CHARACTER*(*) NAME
    LOGICAL WANTLOAD
    INTEGER*4 CSRINIT

Allocate memory for CSR tide modeling and read grids into memory.
When WANTLOAD is .TRUE., loading tide grids are loaded and load tide
will be computed. When .FALSE., CSRTIDE will return a zero load tide.

Input arguments:
 NAME     : Name of the CSR tide model (csr_3.0 or csr_4.0)
 WANTLOAD : Specify that load tide has to be computed

Returned value:
 CSRINIT  : Pointer to memory buffer
```

2.3 ETIDE.CE – Solid Earth Tide, Cartwright and Edden

```
    FUNCTION ETIDE_CE (MJD, LAT, LON)
    implicit none
    REAL*8   MJD, LAT, LON, ETIDE_CE
```

```
This function returns the solid earth tide elevation at a specified
epoch (MJD) and location (LAT and LON).
The elevation is based on the tidal potential according to
Cartwright and Tayler (Geophys. J R. Astr Soc, 1971, 23, 45-74)
as corrected by Cartwright and Edden (Geophys. J. R. Astr Soc., 1973,
33, 253-264).

This implementation is conform the T/P algorithm G1062 for solid earth tides.
Although, some changes have been made to increase precision.

Input arguments:
 MJD : Modified Julian Date
 LAT : Geodetic latitude in degrees
 LON : Geodetic longitude in degrees

Function output:
 ETIDE_CE : Solid earth tide elevation in meters
```

2.4 FESTIDE – Compute tides according to FES model

```
    SUBROUTINE FESTIDE (UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD)
    REAL*8    UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD
```

```
This routine makes the tidal predictions of ocean and load tide
(optional) based on one of the FES models. This routine is heavily
based on the routines by J.M. Molines and F. Lefevre for the FES99,
FES2002 and FES2004 models.

The input grids can be found in $ALTIM/data/$TIDE. The grids are
in NetCDF format and were converted using the program "fes2nc".

To initialize the computation, the function FESINIT should be
called first. It allocates the appropriate amount of memory and
loads the grids into memory. To release the memory for further
use, call DALLOCF.

Longitude and latitude are to be specified in degrees; time in UTC
seconds since 1 Jan 1985. All predicted tides are output in meters.
If the tide is requested in a point where it is not defined, NAN
(Not-a-Number) is returned.

Input arguments:
 UTC      : UTC time in seconds since 1 Jan 1985
 LAT      : Latitude (degrees)
 LON      : Longitude (degrees)

Output arguments:
 TIDE     : Predicted short-period tide (m)
 TIDE_LP  : Predicted long-period tide (m)
 TIDE_LOAD: Predicted loading effect (m)
```

2.4.1 FESINIT – Initialize FES tide model

```
    FUNCTION FESINIT (NAME, WANTLOAD)
    CHARACTER*(*) NAME
    LOGICAL WANTLOAD
    INTEGER*4 FESINIT
```

```
Allocate memory for FES tide modeling and read grids into memory.
When WANTLOAD is .TRUE., loading tide grids are loaded and load tide
will be computed. When .FALSE., FESTIDE will return a zero load tide.

Input arguments:
 NAME     : Name of the FES tide model (FES95.2.1, FES99, or
            FES2002, or FES2004)
 WANTLOAD : Specify that load tide has to be computed

Returned value:
 FESINIT  : Pointer to memory buffer
```

2.5 GOTTIDE – Compute tides according to GOT model

```
    SUBROUTINE GOTTIDE (UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD)
    REAL*8    UTC, LAT, LON, TIDE, TIDE_LP, TIDE_LOAD
```

```
Compute ocean tidal height for given time and location from grids
of harmonic coefficients of one of the GOT models.
Current version uses the 8 largest constituents in the
semidiurnal & diurnal bands, with other tides inferred.
This routine is heavily based on the routine PERTH2 by Richard Ray.

The input grids can be found in $ALTIM/data/$TIDE. The grids are
in NetCDF format and were converted using the program "got2nc".

To initialize the computation, the function GOTINIT should be
called first. It allocates the appropriate amount of memory and
loads the grids into memory. To release the memory for further
use, call DALLOCF.

Longitude and latitude are to be specified in degrees; time in UTC
seconds since 1 Jan 1985. All predicted tides are output in meters.
If the tide is requested in a point where it is not defined, NAN
(Not-a-Number) is returned.

Input arguments:
 UTC      : UTC time in seconds since 1 Jan 1985
 LAT      : Latitude (degrees)
 LON      : Longitude (degrees)

Output arguments:
 TIDE     : Predicted short-period tide (m)
 TIDE_LP  : Predicted long-period tide (m)
 TIDE_LOAD: Predicted loading effect (m)
```

2.5.1 GOTINIT – Initialize GOT tide model

```
    FUNCTION GOTINIT (NAME, WANTLOAD)
    CHARACTER*(*) NAME
    LOGICAL WANTLOAD
    INTEGER*4 GOTINIT
```

```
Allocate memory for GOT tide modeling and read grids into memory.
When WANTLOAD is .TRUE., loading tide grids are loaded and load tide
will be computed. When .FALSE., GOTTIDE will return a zero load tide.

Input arguments:
 NAME     : Name of the GOT tide model (GOT99.2b or GOT00.2)
 WANTLOAD : Specify that load tide has to be computed

Returned value:
 GOTINIT  : Pointer to memory buffer
```

2.6 WEBTIDE – Compute tide from BIO/OSD tide model

```
    INTEGER FUNCTION WEBTIDEINIT (DIRNAME, EXT)
    CHARACTER*(*) DIRNAME, EXT

    INTEGER FUNCTION WEBTIDE (UTC, LAT, LON, TIDE, TIDE2)
    REAL*8 UTC, LAT, LON, TIDE, TIDE2

    SUBROUTINE WEBTIDEFREE ()

The Bedford Insitute of Ocenography (BIO) Ocean Science Division (OSD) of the
Department of Fisheries and Oceans Canada provides various local tide models
for the coasts of the North American Continent. These models can be found at:
http://www.mar.dfo-mpo.gc.ca/science/ocean/coastal_hydrodynamics/WebTide/webti

The WEBTIDE routine has to be initialised using WEBTIDEINIT. Use the name of
the directory and the preferred file extension as arguments. The directory
DIRNAME should contain the following files:
- tidecor.cfg, which contains the names of the node, elements and IOS file
- The files mentioned in tidecor.cfg
- constituents.txt, which contains the names of all constituents ending with "
- Files CONST.barotropic.EXT, where CONST is the name of any consituent and EX
  the preferred extension.

The extension EXT can be:
- s2c : tidal elevation
- v2c : tidal velocities
- v2r : relative tide (add Z0 component)

WEBTIDE can be called to determine the tidal elevation or velocity at any time
(UTC) and location (LAT, LON). If the tidal model called for is a velocity mod
(v2c files instead of s2c files), both TIDE and TIDE2 will be filled.
When the location is outside the model (that is outside any of the finite
elements), TIDE (and TIDE2) will be filled with NaN (Not-A-Number).

After the last call to WEBTIDE use WEBTIDEFREE to free the memory occupied by
the tide model data.

Arguments:
  DIRNAME: Path name of the directory containing the requested WEBTIDE tide
           model
  EXT    : Extension of the constituent file names (s2c, v2c, v2r)
  UTC    : Tide in UTC seconds since 1.0 January 1985
  LAT    : Latitude in degrees
  LON    : Longitude in degrees
  TIDE   : Tide in meters
In case the tide model is complex (.v2c files are used):
  TIDE   : X-component of tidal velocity (m/s)
  TIDE2  : Y-component of tidal velocity (m/s)

Return codes:
  WEBTIDEINIT: 0 = Scalar tide, 1 = Complex tide
  WEBTIDE    : Returns index number of the element in which the location
               resides. Returns -1 if outside model region.

This code can also be called from C programs using the functions WebTideInit,
WebTide and WebTideFree.
```

2.7 LPETIDE – Compute long-period equilibrium ocean tides

```
    FUNCTION LPETIDE (TIME, LAT, MODE)
    REAL*8   LPETIDE, TIME, LAT
    INTEGER  MODE

Computes the long-period equilibrium ocean tides.
Fifteen tidal spectral lines from the Cartwright-Tayler-Edden
tables are summed over to compute the long-period tide.

The terms with monthly and fortnightly frequencies can be
excluded and smaller terms can be included depending on
the value of MODE:
MODE = 0 : Use the 15 Cartwright-Taylor-Edden waves
     = 1 : Idem, but exclude the Mm, Mf and Mtm waves
     = 2 : Use all 123 second and third order waves
     = 4 : Idem, but exclude the Mm, Mf, Mtm and Msqm waves
When using LPETIDE in combination with the FES2004
tide model MODE should be 2 since FES2004 already includes
the four monthly and fortnightly components.

Technical references:
  Cartwright & Tayler, Geophys. J. R.A.S., 23, 45, 1971.
  Cartwright & Edden, Geophys. J. R.A.S., 33, 253, 1973.
  Tamura Y., Bull. d'information des mares terrestres, Vol. 99, 1987.

Based on LPEQMT by Richard Ray.

Input arguments:
 TIME    : UTC time (MJD)
 LAT     : Latitude (degrees)
 MODE    : Determines how many waves are include (see above)

Returned value:
 LPETIDE : Computed long-period tide (meters)
```

2.8 POLETIDE – Compute pole tide elevation

```
    FUNCTION POLETIDE (MJD, LAT, LON)
    REAL*8   POLETIDE, MJD, LAT, LON
```

```
This function computes the pole tide elevation (POLETIDE) for a given
location (LAT and LON) and epoch (MJD). The position of the pole
is interpolated from the daily IERS 90C04 values in
$ALTIM/data/tables/eopc04.62-now  and
$ALTIM/data/tables/eopc04.pred

These files originate from
http://hpiers.obspm.fr/eoppc/eop/eopc04/eopc04.62-now  and
http://hpiers.obspm.fr/eoppc/eop/prediction/eopc04.pred

Although the tables provide values from 1962 till present, only
values from 1985 onward are stored. The daily values are interpolated
linearly.

The pole tide elevation algorithm is based on Munk and MacDonald,
The Rotation of the Earth, 1960, and uses the mean location of the
pole as reference (though XPOLE and YPOLE are absolute values).

The pole tide provided is that of the solid earth plus ocean. This
means that the value provided is (1 + k2) times the equilibrium pole
tide (He) with k2 = 0.302. The respective contributions from the solid
earth and ocean are h2 He (with h2 = 0.609) and (1 - h2 + k2) He,
respectively. In other words, of the total 46.8% is due to the solid
earth and 53.2% to the ocean.

Ref: J. W. Wahr, "Deformation of the Earth induced by polar motion",
J. Geophys. Res., 90, 9363-9368, 1985.

Input arguments:
 MJD      : Modified Julian date (floating point double)
 LAT, LON : Geodetic latitude and longitude of the observer in degrees.

Returned value:
 POLETIDE    : Pole tide elevation (solid + ocean) in meters.
```