# Harmonic Analysis of Tides Using **TideHarmonics**

Alec Stephenson

May 4, 2017

## Summary

This document gives a brief introduction on the use of **TideHarmonics**. The package implements harmonic analysis of tidal and sea-level data. Up to 409 different harmonic tidal constituents can be estimated, and daily nodal corrections are implemented by default.Time-varying mean sea-levels can also be used.

Section 1 briefly outlines the theory of harmonic tidal analysis. Example code is contained in Section 2, which gives an application to the freely available hourly sea-level data recorded as part of the Australian Baseline Sea-Level Monitoring Project (ABSLMP). This involves several sites on the coast of Australia, which incorporates a wide range of tidal behaviours.

Section 3 gives details on the names used for the tidal harmonics. In particular, we use a different naming scheme to that which has been traditionally used, although both naming schemes can be used for user input.

To cite this package or this manual please use the following:

Stephenson, A. G. (2016). Harmonic Analysis of Tides Using TideHarmonics. URL: https://CRAN.R-project.org/package=TideHarmonics.

# 1 Basic Theory

## 1.1 Amplitude and Phase Estimates

Suppose we have a tidal data series $y(t)$, where $t$ is in hours. We can model the series with $N$ harmonic constituents (sinusoids) using the prediction

$$\hat{y}(t) = Z + \sum_{n=1}^{N} A_n \cos\left(\frac{\pi}{180}(\omega_n t - \psi_n)\right) \tag{1}$$

where $\omega_n$ is the known angular frequency of the $n$th component in degrees per hour, which in the context of tidal analysis is referred to as the speed of the tidal component. For example, the speed of the solar semi-diurnal (twice daily) component, typically called

$S_2$ or S2, is 30 degrees per hour, since the component repeats every $360/30 = 12$ hours. There are $2N+1$ parameters to be estimated from the data; the amplitudes $A_n$, the phase lags[1] $0 \leq \psi_n < 360$ (degrees), and the mean sea-level Z.

We can also introduce known nodal correction functions $f_n(t)$ and $u_n(t)$ for the amplitude and phase respectively, which are essentially designed to account for long period astronomical cycles. For solar components such as S2, we have $f_n(t) \equiv 1$ and $u_n(t) \equiv 0$, so that no adjustment is performed, but for other components $f_n(t)$ and $u_n(t)$ may differ (slightly) from one and zero at any given time $t$. We take the amplitudes as $A_n = H_n f_n(t)$ and take the phase lags as $\psi_n = g_n - u_n(t) - V_n$ modulo 360, where $V_n$ is a known value as discussed in Section 1.3.

This gives

$$\hat{y}(t) = Z + \sum_{n=1}^{N} H_n f_n(t) \cos\left(\frac{\pi}{180}(\omega_n t - g_n + u_n(t) + V_n)\right) \tag{2}$$

where the amplitudes $H_n$ (strictly speaking, the amplitude divided by $f_n(t)$), the phase lags $0 \leq g_n < 360$ (strictly speaking, the phase lags behind $u_n(t) + V_n$), and the mean sea-level $Z$ are now the $2N+1$ parameters to be estimated.

For numerical reasons we first estimate the mean sea-level $Z$ using the mean of the data, and then form the centred data $y(t) - \hat{Z}$. We then use a zero-intercept linear regression on the centred data. Using the trigonometric identity $\cos(A - B) = \cos(A)\cos(B) + \sin(A)\sin(B)$ we have

$$H_n f_n(t) \cos\left(\frac{\pi}{180}(\omega_n t - g_n + u_n(t) + V_n)\right) = \tag{3}$$

$$C_n f_n(t) \cos\left(\frac{\pi}{180}(\omega_n t + u_n(t) + V_n)\right) + S_n f_n(t) \sin\left(\frac{\pi}{180}(\omega_n t + u_n(t) + V_n)\right), \tag{4}$$

which is linear in the unknown (cosine and sine) parameters $C_n = H_n \cos(g_n)$ and $S_n = H_n \sin(g_n)$. The regression estimates for $C_n$ and $H_n$ can then be used to derive amplitude and phase lag estimates using $H_n = (C_n + S_n)^{1/2}$ and $g_n = arctan(S_n/C_n)$.

This estimation method is implemented by the function `ftide` in the package. As the estimation method is based on linear regression it is very robust, but the user should keep in mind the following points.

1. There may be identifiability problems if two harmonic components have very similar speeds. The package will produce an error if two specified harmonic components have the same or virtually identical speeds (specifically, if the first five Doodson numbers are the same), even if the nodal corrections are different.

2. You should have data over a period of time that is long enough to be able to estimate the component with the slowest speed (longest period). By default, the slowest estimated harmonic is the annual solar term Sa, corresponding to a period of one year. If you do not have at least one year of data you should not include Sa.

3. The frequency of observation should be high enough to be able to identify the component with the fastest speed (shortest period). By default, the fastest estimated harmonic has a period of just over 4 hours, but harmonics with periods of less than 2 hours can be selected.

[1] In tidal analysis it is common to use a minus sign before the phase $\psi_n$, calling this the phase lag, and taking it to be in the interval $[0, 360)$.

### 1.2 Time Varying Mean Sea-Levels

The function ftide allows the user to fit models with time varying mean sea-levels. The model is simply

$$\hat{y}(t) = Z(t) + \sum_{n=1}^{N} A_n \cos\left(\frac{\pi}{180}(\omega_n t - \psi_n)\right) \tag{5}$$

so that the parameter $Z$ is replaced by the function $Z(t)$. We use a non-parametric smoother (the loess function) to first estimate $Z(t)$, and then centre the data using $y(t) - \hat{Z}(t)$, proceeding as before.

The basic idea is to non-parametrically model the variability in the data that exists at periods longer than the slowest speed (longest period) harmonic component. The $Z(t)$ function can therefore be used to identify long period sea-level changes. Arguments can be passed to ftide to control the smoothness of the $Z(t)$ estimate; we recommend it should be smooth enough so that it does not compete with the slowest (parametric) harmonic.

### 1.3 More Details

The ftide function requires as input the recorded sea-levels and a date/time[2] object giving the date and time of each observation. Nodal corrections $f_n(t)$ and $u_n(t)$ are each calculated on a daily basis, so that if the date/time object spans $n_d$ days, then for each harmonic component, $n_d$ different nodal corrections will be used. Some software will use a nodal correction at a fixed time point for the whole series; using nodal corrections that change on a daily basis ensures that they will remain accurate over long time periods.

The centre value of the date/time object is taken to be the origin $t = 0$, and $t$ is then taken as the number of hours from this point; for example, an earlier observation may be at e.g. $t = -45.65$ hours. Phase lags $g_n$ are calculated relative to a reference signal (equilibrium phase) $V_n$, which can be calculated for any given date; we use the date of the centre value of the date/time object (i.e. the date of the origin $t = 0$).

Note that the phase lags $g_n$ will always be given in UTC (Coordinated Universal Time). If a different time zone is required, the phase lags can be displayed in any other time zone using the utc argument of the coef function (see Section 2).

## 2 Application to Australian ABSLMP Data

The Australian Baseline Sea-Level Monitoring Project (ABSLMP) is designed to monitor sea-levels around the Australian coastline. The ABSLMP freely distributes hourly sea-level (and meteorological) data at sites within the ABSLMP monitoring network. In this section we model the hourly data using **TideHarmonics**.

We note that hourly sea-level data is also freely available at Australian GLOSS sites (Global Sea Level Observing System Sites) but at other Australian sites it is generally not freely available (i.e. not available at zero cost) at hourly frequencies.

[2] Specifically, a POSIXct vector or any object that can be converted to a POSIXct.

For Australian sites, the Australian National Tide Tables (ANTT) publishes estimates of 22 harmonic tidal constituents, but this is not free and must be purchased at cost. For USA sites, estimates of 37 harmonic tidal constituents (amplitudes and phases) are freely available on the NOAA website.

## 2.1 Data Construction

The ABSLMP hourly data is currently available at the website of the Australian Bureau of Meteorology. Obviously, there is no guarantee that the exact file locations will be correct in the future, or that the files will be named and stored in the same manner. The code of this section may therefore need to be altered.

Each csv file that can be downloaded gives hourly sea-level data at a particular site and in a particular calendar year. We use data from 2012-2014. However if you wish, you can go all the way back to 1992. There are currently 14 ABSLMP 'standard' sites and 2 ABSLMP 'supplementary' sites. We use eight of the 14 standard sites.

```
> years <- 2012:2014
> sites <- c("CapeFerguson","PortKembla","Portland","Thevenard",
+            "Esperance","Hillarys","Broome","Darwin")
> siteid <- paste0("IDO710",c("01","03","08","10","11","12","13","14"))
```

The csv files can be downloaded and read using the following double for loop over each of the eight sites and each of the three years. In each csv file, the first column is a date/time object in Universal Coordinated Time (UTC), the second column is the sea-level in metres above Tide Gauge Zero, and the remaining columns contain meteorological information that we do not use here. In the conversion of the Date/Time object, it is important to specify both the UTC time zone[3] and the correct format as used in the csv files. Missing or erroneous data points are set to -9999 in the csv files, which we convert to NA values.

```
> fn <- "http://www.bom.gov.au/ntc/"
> fn <- paste0(fn, siteid,"/",siteid,"_")
> abslmp <- list(); tp <- tempfile()
> for(i in 1:length(sites)) {
+   nms <- paste0(fn[i], years,".csv")
+   abslmp[[i]] <- data.frame()
+   for(j in 1:length(years)) {
+     download.file(nms[j], tp)
+     abslmp[[i]] <- rbind(abslmp[[i]], read.csv(tp,as.is=TRUE)[,1:2])
+   }
+   colnames(abslmp[[i]]) <- c("DateTime","SeaLevel")
+   abslmp[[i]]$DateTime <- as.POSIXct(abslmp[[i]]$DateTime, tz="UTC",
+     format = "%d-%b-%Y %H:%M")
```

[3] Timezone issues are somewhat system-specific, but consider this scenario: the timezone is not specified and is taken by default as a local time for a region where DST (daylight savings time) is used. On 2am on a day where the clocks go forward, the date/time in the csv file will be converted to NA because it will not exist.

```
+   abslmp[[i]]$SeaLevel[abslmp[[i]]$SeaLevel == -9999] <- NA
+ }
> names(abslmp) <- sites
```

The code creates a list called `abslmp`, where each list element corresponds to a single site. Each list element is a dataframe of two columns, where the first is the date/time object and the second is the sea-level. The observations are hourly over the three year period 2012-2014, and therefore each dataframe contains 26304 rows (although most sites contain some missing observations). Typing `lapply(abslmp, summary)` gives summary information at each site.

If you do not have an internet connection you can instead create the `abslmp` object using the following code. The 2012-2014 hourly data for the eight sites is included in the **TideHarmonics** package[4] using the site names as given in the `sites` vector. However, if you wish to use a different collection of sites or a different time period, you will need to download the data as above.

```
> library(TideHarmonics)
> abslmp <- list()
> for(i in 1:length(sites)) abslmp[[i]] <- get(sites[i])
> names(abslmp) <- sites
```

## 2.2 Tidal Models For Esperance

The main function in the **TideHarmonics** package is the `ftide` function, which creates a `tide`[5] object. This object contains all information regarding the fitted tidal model.

The code below loads the package and fits a model to the Esperance data. At Esperance there are 513 missing hourly observations in the 2012-2014 period. The first two arguments to `ftide` are the sea-levels (missing values are allowed) and the date/time object (missing values are not allowed). By default 60 harmonic constituents will be fitted, the names of which are given in the vector `hc60`. In the code below we specify only the four major harmonic constituents M2 S2 K1 O1 using the vector `hc4`.

```
> library(TideHarmonics)
> esp <- abslmp$Esperance
> m1 <- ftide(esp$SeaLevel, esp$DateTime, hc4)
> m1


Call:
lm(formula = x ~ -1 + ., data = xmat, na.action = na.exclude)

Form Factor: 1.299872

Features (Diurnal or Mixed):
```

[4] For each site the data is lazy loaded from a (compressed) rda file of approximately 80 KB.

[5] Specifically, this is a list object of class `c("tide","lm")`.

```
     MLLW      MHLW       MSL      MLHW      MHHW
0.4288305 0.6398880 0.8472419 1.0545959 1.2656534

Harmonics:
    amplitude  phase      sine       cosine
K1    0.18424  228.89038   -0.13881   -0.12114
S2    0.13517  123.62640    0.11255   -0.07486
O1    0.12864  239.56604   -0.11092   -0.06516
M2    0.10553  141.09417    0.06628   -0.08212
```

The printed object displays the form factor, which in terms of fitted amplitudes is always calculated as (K1+O1)/(M2+S2). This is an important value for determining whether the site is diurnal (one high tide per day) or semi-diurnal (two high tides per day) or mixed semi-diurnal (two high tides per day with one being higher than the other).

Lower values of the form factor indicate semi-diurnal sites, with higher values indicating diurnal sites, and mid-range values indicating mixed semi-diurnal sites. Different cut-off values are used[6] in different publications: we take values less than 0.5 to be semi-diurnal, and values above 0.5 to be diurnal or mixed semi-diurnal. The form factor here shows that Esperance is a diurnal or mixed semi-diurnal site (sites on the south-west coastline of Australia are diurnal or mixed semi-diurnal).

The features vector gives a standard set of features that summarise the tidal behaviour. The features are calculated from the estimated amplitudes and not the raw data. Different features are displayed depending on the site classification (semi-diurnal, diurnal or mixed). In terms of amplitudes, the full definitions for the features are as given below.

For semi-diurnal sites:

- MLWS = Mean Low Water Springs = Z - (M2 + S2)
- MLWN = Mean Low Water Neaps = Z - |M2 - S2|
- MSL = Mean Sea-Level = Z
- MHWN = Mean High Water Neaps = Z + |M2 - S2|
- MHWS = Mean High Water Springs = Z + (M2 + S2)

For diurnal or mixed semi-diurnal sites:

- MLLW = Mean Lower Low Water = Z - (M2 + K1 + O1)
- MHLW = Mean Higher Low Water = Z - |M2 - (K1 + O1)|
- MSL = Mean Sea-Level = Z
- MLHW = Mean Lower High Water = Z + |M2 - (K1 + O1)|
- MHHW = Mean Higher High Water = Z + (M2 + K1 + O1)

[6] One common scheme is that: 0 to 0.25 is semi-diurnal; 0.25 to 1.5 is mixed, mainly semi-diurnal; 1.5 to 3.0 is mixed, mainly diurnal; above 3.0 is diurnal.

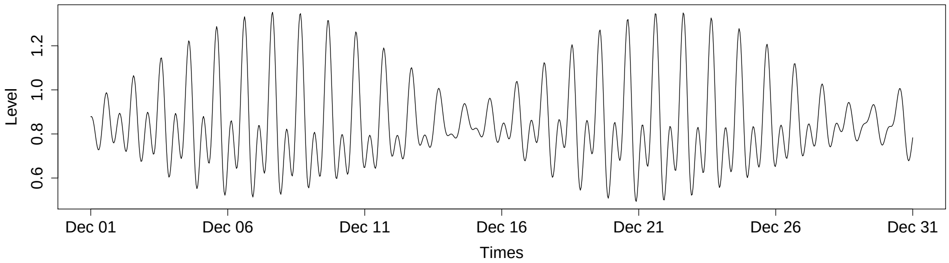

Figure 1: Estimated tide at Esperance for Dec 2014, using a harmonic analysis with only the four constituents M2 S2 K1 O1, applied to hourly data recorded in the period 2012-2014.

The columns of the harmonics matrix gives the estimates for $A_n$, $g_n$, $S_n$ and $C_n$ respectively, with the rows ordered by decreasing amplitude. The estimate for the mean sea-level (MSL) is given in the centre of the features vector. For semi-diurnal sites the components M2 and/or S2 tend to have the largest amplitudes, whereas for diurnal sites the components K1 and/or O1 tend to dominate.

There are four method functions specific to `tide` objects: `coef`, `plot`, `predict` and `print`, the last of which is typically not called directly but controls the printing of the object as above. Method functions for linear regression objects such as `residuals` can also be applied to `tide` objects, in order to e.g. calculate sea-level residuals.

The `coef` function allows for the easy extraction of any of the model estimates (sine and cosine coefficients, amplitudes and phase lags). It can also be used to convert the phase lags to different time zones. The `predict` function returns estimates of the tide at any given dates/times. The code below shows examples of the use of `plot`, which displays line traces of the estimated tide levels.

```
> t1 <- as.POSIXct("2014-12-01 00:00", tz = "UTC")
> t2 <- as.POSIXct("2014-12-31 00:00", tz = "UTC")
> plot(m1, t1, t2)
> plot(m1, t1, t2, split = TRUE, ylim = c(-0.18,0.18))
```

The first two lines specify the time/date limits of the plot region. Here we plot over the month of December 2014. The first plot is the estimated tide (including the MSL), as displayed in Figure 1. The second plot uses the `split` argument to plot every harmonic constituent individually, as displayed in Figure 2. Here we only have M2 S2 K1 O1, and therefore only these are plotted (excluding the MSL). We fix the y-limits in order to emphasise the difference in the amplitudes. The `which` argument can be used to plot only particular constituents, and the `msl` argument can be specified to control the inclusion or exclusion of the mean sea-level.

From the estimated tides plot in Figure 1 we can see that there is fairly regular behaviour as a result of only four harmonics being fitted. The largely diurnal nature of the site is also clear, with one (higher) high tide per day, and with the lower high tide being at a similar level to the higher low tide.

The tidal range is large around 8th Dec and 22nd Dec, and small around 15th Dec and 29th Dec. This behaviour is essentially a spring/neap tidal cycle, although the spring/neap

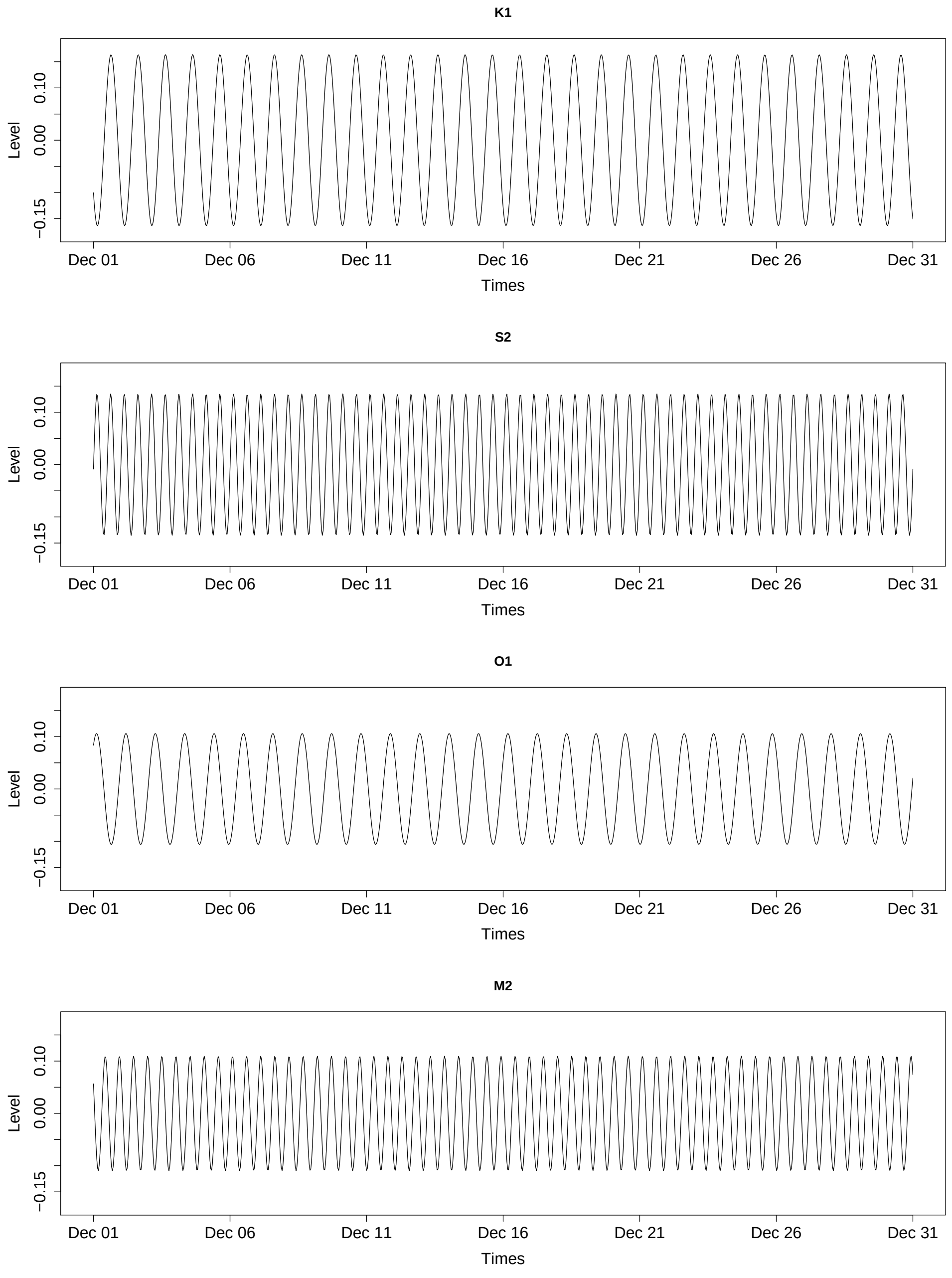

Figure 2: Estimated individual harmonic constituents of tide at Esperance for Dec 2014, using a harmonic analysis with only the four constituents (from top to bottom) K1 S2 O1 M2, applied to hourly data recorded in the period 2012-2014. The harmonics are ordered by decreasing amplitude. K1 and O1 are diurnal (roughly one peak per day), whereas S2 and M2 are semi-diurnal (roughly two peaks per day).

tide cycle is technically produced from the amplitude modulation that arises from adding M2 and S2. In this case we have a largely diurnal site, and therefore the behaviour here actually arises primarily from adding K1 and O1. The difference in speed between M2 and S2 is about 1.016 degrees per hour, and therefore the spring/neap cycle has a period of 360/1.016 hours or 14.8 days. But the difference in speed between K1 and O1 leads to a slightly shorter cycle period of 13.7 days, which is the case here.

The number of fitted harmonics can be increased by using a vector other than `hc4` in the call to `ftide`. For example, the vector `hc114` can be used to fit 114 harmonics, or a character vector can be constructed to give the user complete control over the harmonics to be fitted. Single harmonic components can also be fitted by passing only the name of that component e.g. `"M2"`. If the fitted harmonics do not include all of the four major harmonic constituents M2 S2 K1 O1 then the form factor and features vector cannot be calculated and so will not be displayed.

The code below gives an example of fitting 60 harmonic constituents, and plots the tidal predictions for each calendar month within the year 2014. The actual sea-levels are also plotted as a dashed red line. The (sea-level) residuals are therefore the differences between the two lines. The plots for the first four months of 2014 are displayed in Figure 3. The monthly plots are produced within a for loop over the months, so if you are using a graphical interface that overwrites previous plots you may first need to specify `par(ask=TRUE)` in order to pause between displays.

```
> m2 <- ftide(esp$SeaLevel, esp$DateTime, hc60)
> tt <- c(rep(2014,12),2015)
> tt <- paste(tt,sprintf("%02d",c(1:12,1)),"01",sep="-")
> tt <- as.POSIXct(tt, tz = "UTC")
> for(i in 1:12) {
+   plot(m2, tt[i], tt[i+1], main=paste(month.abb[i],2014))
+   ind <- esp$DateTime >= tt[i] & esp$DateTime <= tt[i+1]
+   lines(esp[ind,], lty=2, col="red")
+ }
```

On a basic level, the observed sea-level can be thought to consist of a tide component and a storm surge (also called tidal surge) component, where storm surges are caused by changes in wind and barometric pressure. The sea-level residuals therefore represent[7] the storm surge component, with the harmonic constituents representing the tide component. Large storm surges can typically be identified using plots such as Figure 3; they occur in periods where the observed sea-levels (dashed red lines) are much larger than the tidal estimates (black lines).

### Other Features

The `ftide` function also contains arguments to specify the method used for longitude formulas (which control the exact calculations for $f_n(t)$, $u_n(t)$ and $V_n$), and to implement time-varying mean sea-levels $Z(t)$, and to turn off nodal corrections (i.e. setting $f_n(t) \equiv 1$ and $u_n(t) \equiv 0$ for all harmonics).

[7] This is a simplification. There are a large number of different factors, in addition to storm surge, which may result in non-cyclic components e.g. seiches, coastally trapped waves, bathymetry, climate change.

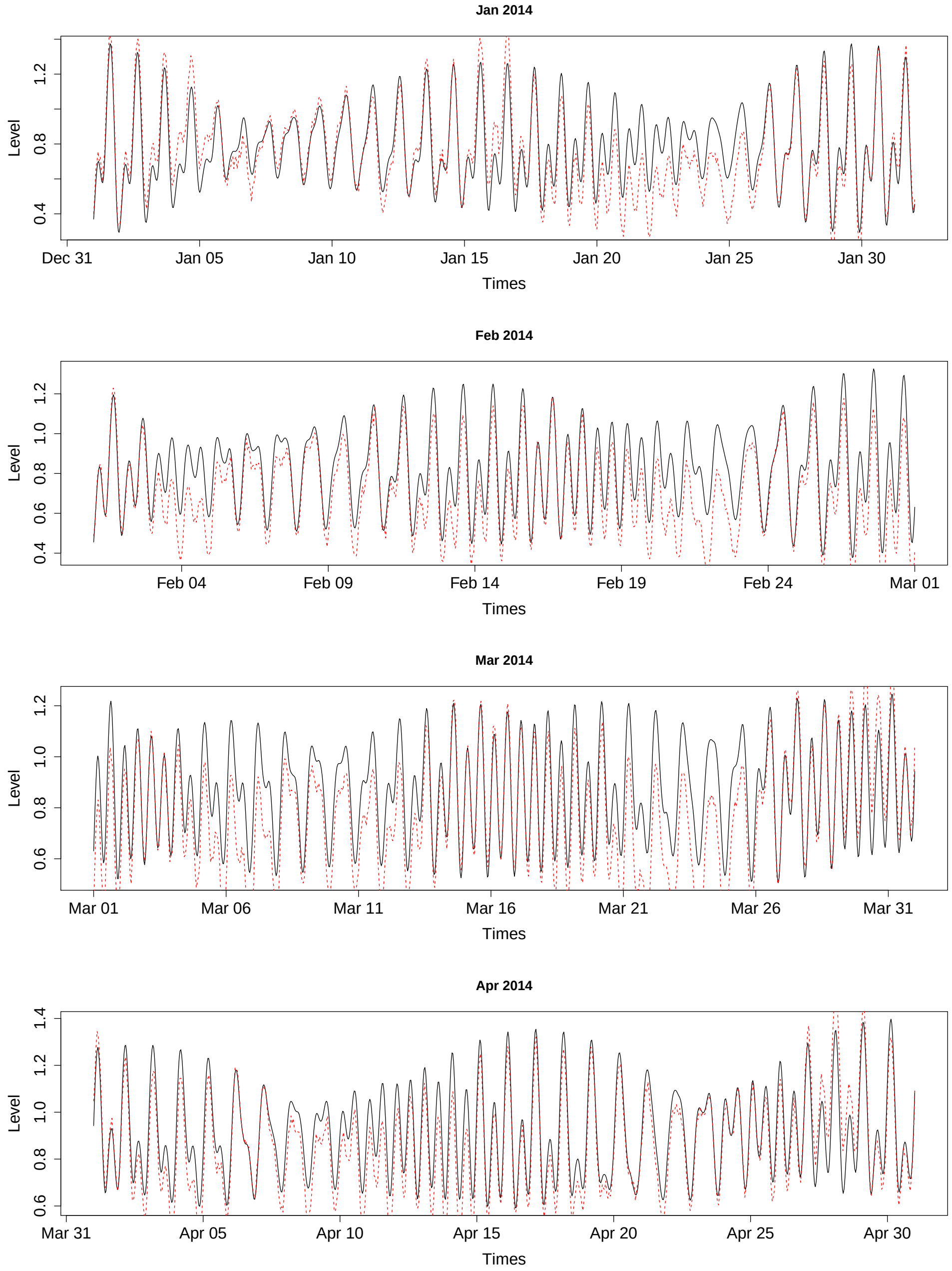

Figure 3: Estimated tide (black line) at Esperance for the first four months of 2014, using a harmonic analysis with 60 harmonic constituents, for hourly data recorded in the period 2012-2014. The dashed red line denotes actual sea-level observations.

The **TideHarmonics** package does not explicitly contain any functions for model selection, to e.g. determine the number of harmonics required. However, a fitted `tide` object is based on (i.e. inherits from) a linear regression object, and therefore many of the model selection tools for regression that currently exist in R can also be applied to `tide` objects.

## 2.3 Tidal Models For All Australian Sites

The analysis above was restricted to a single site, but we can easily fit models to every site using the following code. Here we construct a list, where the elements are the `tide` objects for each site. We use 60 harmonics for the fit using the `hc60` vector. This is the default, and so does not need to be specified.

```
> mlst <- list()
> for(i in 1:length(sites)) {
+   df <- abslmp[[i]]
+   mlst[[i]] <- ftide(df$SeaLevel, df$DateTime)
+ }
> names(mlst) <- sites
> sapply(mlst, function(x) x[["fval"]])
> lapply(mlst, function(x) round(head(coef(x, hc = TRUE),10),3))
```

```
CapeFerguson   PortKembla     Portland    Thevenard
   0.4505400    0.4453566    1.1568622    0.4977660
   Esperance     Hillarys       Broome       Darwin
   1.2973369    2.9910486    0.1063020    0.3237611
```

The second-to-last line of code above displays the estimated form factor at each site. It shows that the Australian coastline permits a wide range of tidal behaviours. The Broome site (and other locations on the north-west coastline) has a very low form factor which indicates a purely semi-diurnal site, whereas the Hillarys site (and other locations on the south-west coastline) has a very high form factor which indicates a purely diurnal site. Locations on the eastern and south-eastern coastlines tend to be more mixed.

The last line of code above displays the top ten harmonics (in the sense that they have the largest amplitude) for each of the sites. The output is lengthy and so is not displayed here. The (diurnal) K1 harmonic is the largest harmonic for the Hillarys, Esperance and Portland sites. The (semi-diurnal) M2 harmonic is largest for all other sites except for Thevenard (which is located on the Great Australian Bight), where S2 is largest and M2 is second-largest.

The code below plots the estimated tides in December 2014 for each of the eight sites, as displayed in Figures 4 and 5. The actual observed sea-levels are also plotted as dashed red lines. We have removed the mean sea-level (MSL) from both the tide estimates and the observed data. Gaps in the dashed red line are due to missing data.

```
> t1 <- as.POSIXct("2014-12-01 00:00", tz = "UTC")
> t2 <- as.POSIXct("2014-12-31 00:00", tz = "UTC")
> msl <- sapply(mlst, function(x) x[["msl"]])
> for(i in 1:length(sites)) {
+   plot(mlst[[i]], t1, t2, msl = FALSE, main=sites[i])
+   df <- abslmp[[i]]
+   ind <- df$DateTime >= t1 & df$DateTime <= t2
+   lines(df[ind,1], df[ind,2]-msl[i], lty=2, col="red")
+ }
```

Figures 4 and 5 illustrate the different types of tidal behaviour around the coastline of Australia, from pure semi-diurnal (Broome) to pure diurnal (Hillarys) and anything in-between. Also notice that the tidal ranges are very different; the y-axes limits for Broome and Darwin are around -4 and 4 metres, whereas the y-axes limits for Hillarys and Esperance are around -0.4 and 0.4 metres. It is left as an exercise for the reader to edit the call to `plot` in the code above by adding `ylim=c(-4,4)`, which fixes the y-axes limits, and has a dramatic effect on the visual impact of the plots.

The comparison between the observed data and the tide estimates in Figures 4 and 5 shows that some sites have more regular behaviour and therefore smaller residuals (relatively) than others; the sea-levels at e.g. Cape Ferguson and Port Kembla are easier to estimate (at least for this month) than those at e.g. Portland and Esperance.

Storm surges can occur at any time of the year, but at most sites there are typically more surges in the Australian winter, and so we might expect the December residuals to be smaller than for other months.

# 3 Names of Harmonic Tidal Constituents

The basic source for the harmonic tidal constituents that we use in the package is the standard list of tidal constituents of the IHO (International Hydrographic Organisation) tidal committee, which is proved in the file doc/Constituent_List.pdf.

From this file we have constructed the dataframe `harmonics` which contains information on the 409 harmonic constituents that can be used. The dataframe contains the Doodson number, the phase constant, the type of nodal correction, and the speed of the constituent, which is calculated directly from the Doodson number.

Some corrections and adjustments (not documented here) have been made to the standard list in the pdf file. We have retained lower case characters and have used the first three characters for greek letters e.g. `the1` for `theta1`. The code has been written for robustness and will still work if you use upper case characters and/or full names for greek letters.

There are many variations of constituent names and I have attempted to write the software so that it will work whatever names are used. For example, NOAA (National Oceanic and Atmospheric Administration) uses RHO rather than the more common rho1 or RHO1. The names RHO and RHO1 are therefore automatically converted to rho1. The TASK-2000 (Tidal Analysis Software Kit) software uses LAMDA1 (with the B missing), which is automatically converted to lam1.

The first two columns of the `harmonics` dataframe are called `name` and `sname`. Both columns are names that uniquely identify the constituent. The `sname` column gives standard names that are traditionally used. For constituents based on one or more underlying

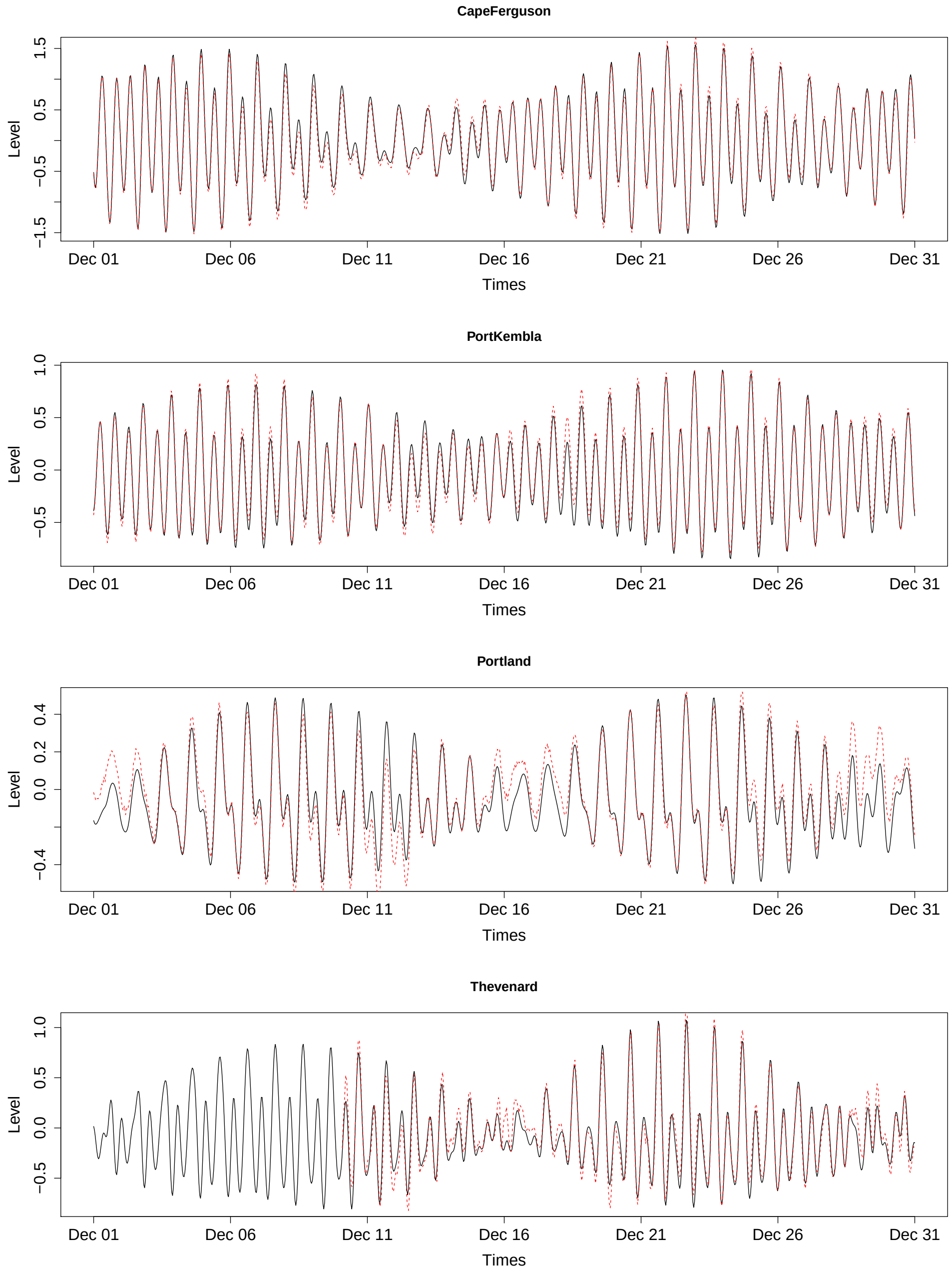

Figure 4: Estimated tide (black line) at different sites for December 2014, using a harmonic analysis with 60 harmonic constituents, for hourly data recorded in the period 2012-2014. The dashed red line denotes actual sea-level observations. The MSL has been removed from both the tide estimates and the sea-level observations. The y-axis limits are variable.

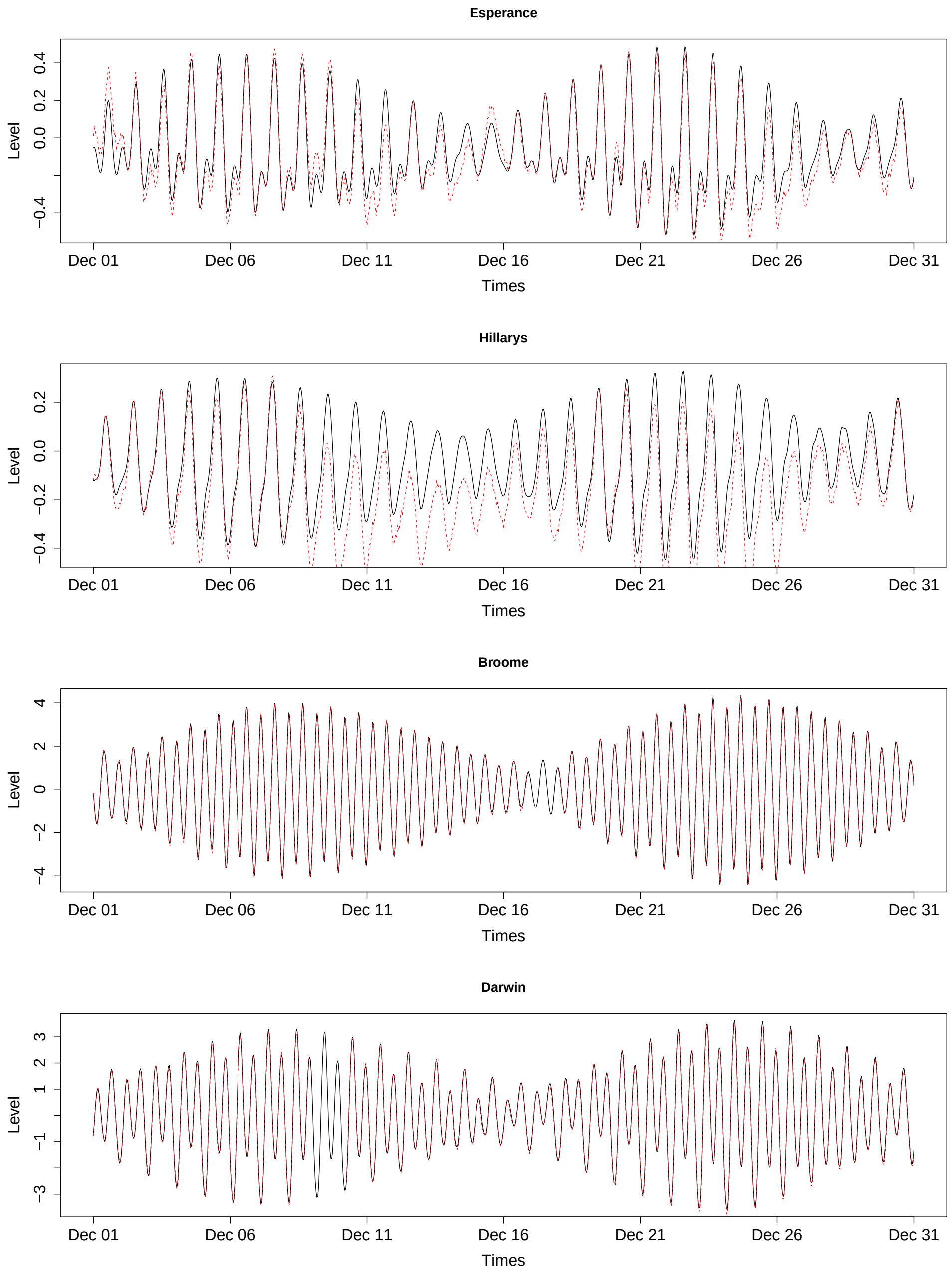

Figure 5: Estimated tide (black line) at different sites for December 2014, using a harmonic analysis with 60 harmonic constituents, for hourly data recorded in the period 2012-2014. The dashed red line denotes actual sea-level observations. The MSL has been removed from both the tide estimates and the sea-level observations. The y-axis limits are variable.

components, this package use a different naming scheme that is much more logical and consistent (and allows for easy automated checking), and these are given in the `name` column. For readers interested in the details, the traditional naming scheme and our naming scheme are described in Section 3.1 below. Either naming scheme can be used in the code.

For both naming schemes, in order to ensure that names are unique, we prepend a lower case 'x' or 'a' to duplicates. A lower case 'x' indicates that components are the same except for the nodal correction. For example `xM1` and `M1` have the same Doodson number, and therefore the same speed, but have different nodal corrections. A lower case 'a' denotes that the speeds (and possibly the nodal corrections) are different. For example, `aS1` and `S1` differ in the sixth Doodson number.

## 3.1 Naming Scheme

### 3.1.1 Traditional Naming Scheme

The traditional naming scheme for naming constituents based on more than one underlying component is largely historical and (to me) seems fairly illogical. It is best demonstrated using two examples, as follows.

4M2N12: This is a twelfth-diurnal constituent, as demonstrated by the 12 at the end. The M and N represent the components M2 and N2. The 4M means '4 of M2' and the 2N means '2 of N2'. The cycles need to match: since $4 \times 2 + 2 \times 2 = 12$, we can determine that 4M2N12 represents the addition of the two components i.e. $4 \times M2 + 2 \times N2$.

6MNS12: Similar to before with M,N and S representing M2, N2 and S2, but since there is no number before the N or the S, these represent '1 of N2' and '1 of S2' respectively. Also, $6 \times 2 + 1 \times 2 + 1 \times 2$ does not equal twelve, so we need to consider swapping positive signs for negative signs (from right to left), giving $6 \times 2 + 1 \times 2 - 1 \times 2 = 12$ as the correct equation. We can therefore determine that 6MNS12 represents $6 \times M2 + 1 \times N2 - 1 \times S2$.

The above involve some thought but they are fairly straightforward cases. Sometimes there are terms involving brackets such as 3(MN)12 and 2(MS)K10 which can be expanded to 3M3N12 and 2M2SK10. Problems can occur with the ordering of components where the multiplier is the same: this is not consistent and often different orderings are used[8]. The letter K can represent both K1 and K2 (similarly, on rare occasions S can represent S1 rather than S2), and this can also lead to problems and ambiguities.[9]

### 3.1.2 Our Naming Scheme

Our naming scheme is based on the following single characters:

M S N K T L R V D
k o j q p s

[8] In rare cases even orderings on multiplicity are not satisfied.

[9] There are many inconsistencies and exceptions in order to avoid the inherent ambiguities of the system. For example, in both M(SK)2 and M(KS)2, where S and K represent S1 and K1, the brackets presumably exist to distinguish it from MSK2 and MKS2, where S and K represent S2 and K2. And the term 3(SM)N2 actually expands to 3SN3M2 because the 3M component is negative.

Upper case characters represent semi-diurnal components, so M is M2, and K is K2, while lower case character represent diurnal components, so k is K1 and o is O1. As is the case in the traditional system, V represents the greek letter nu, and we let D represent lambda, so that V is nu2 and D is lam2.

Each single character always has a single number preceding it that represents a multiplier. In particular, if the multiplier is one, then the numeral 1 must still always be present. As in the traditional system, a number at the end of the name represents the number of diurnal cycles.

A period symbol is placed after the positive terms and before the negative terms (i.e. between positive and negative). If there are no negative terms, then it is placed after the positive terms and before the diurnal cycle number.

The ordering of terms with identical multiplicities is given by the list specified above, so e.g. S is before N. As is usually the case in the traditional method, positive terms are before negative, semi-diurnal terms are before diurnal terms, and then higher multiplicities are listed first.

That is it. Some of the advantages are

1. Every tidal constituent that represents more than one underlying component will have exactly one period symbol, with the location of the symbol making positive and negative terms immediately identifiable.

2. The name can easily be read into a computer so that the proper speed and Doodson number can be checked, and the nodal correction can be calculated (in fact, the calculation of the nodal corrections was the original motivation for constructing a consistent naming scheme).

3. The cycle equation can now be used as a validation check, to ensure that the name and number of cycles is correct.

4. There are no brackets and no ambiguous cases.

For an illustration of how names under this new system correspond to names under the traditional system, see the first two columns of the `harmonics` dataframe.

### 3.1.3 Names of Long Period Constituents

There are 21 long period (low speed) constituents with speeds ranging from 0.04 to 2.58 degrees per hour. Unlike virtually all other[10] constituents, long period constituent names do not have a diurnal cycle number and therefore do not end in a numeral. Many long period constituent names seem to correspond to the period of interest e.g. Sa (solar annual), Ssa (solar semi-annual), Sta (solar tri-annual), Mm (lunar monthly; anomalistic month), MSf (lunisolar synodic fortnight), Mf (lunisolar fortnightly; sidereal fortnight).

Other long period constituent names are based on underlying components; however, as it was not clear to me what all of these underlying components were, I have not used a component-style naming scheme here and have left these names unchanged. Hence, the variables `name` and `sname` are the same for long period constituents.

[10] The only exceptions are the diurnal M1A M1B xM1B M1C and the semi-diurnal NA2* MA2* L2A L2B, all of which have the diurnal cycle number in the second-to-last digit.

### 3.1.4 Names of Shorter Period Constituents

At third-diurnal and beyond (speeds above 42 degrees per hour) most constituents are based on two or more different underlying components and therefore their names are of the component-style type described in Section 3.1.2. The exceptions are listed as follows: M3 S3 K3 xK3 N4 MA4 M4 S4 K4 xM5 M5 aM5 MB5 N6 MA6 M6 S6 M7 aM7 MA8 M8 S8 MA9 M10 MA12 M12.