

Unified Tidal Analysis and Prediction

Using the UTide Matlab Functions

Daniel L. Codiga

Graduate School of Oceanography
University of Rhode Island

September 2011

GSO Technical Report 2011-01

Funded by National Science Foundation, Physical Oceanography Program,
Award 0826243, "Investigating Tidal Influences on Subtidal Estuary-Coast
Exchange Using Observations and Numerical Simulations"

Full citation:

Codiga, D.L., 2011. Unified Tidal Analysis and Prediction Using the UTide Matlab Functions. Technical Report 2011-01. Graduate School of Oceanography, University of Rhode Island, Narragansett, RI. 59pp. <ftp://www.po.gso.uri.edu/pub/downloads/codiga/pubs/2011Codiga-UTide-Report.pdf>

Abstract

A unified tidal analysis and prediction framework is developed. A self-consistent and complete set of equations is presented that incorporates several recent advances, with emphasis on facilitating applicability to the case of irregularly distributed times, and includes as special cases nearly all prior methods. The two-dimensional case treated is suitable for ocean currents, yields current ellipse parameters, and naturally reduces to the one-dimensional case suitable for sea level. The complex number formulation is used for matrix solution but relationships to the real formulation, needed for confidence interval estimation with irregular times, are included. The two-dimensional generalization of Foreman et al. (2009) leads to expressions (including in-matrix treatment instead of post-fit corrections) incorporating exact times in nodal/satellite corrections and in calculation of Greenwich phase from the astronomical argument, as well as exact constituent inference. Some of the resulting capabilities include accurate nodal/satellite corrections for records longer than 1-2 years, and inference of multiple constituents from a single reference. A comprehensive set of constituent selection diagnostics is summarized. Diagnostics to assess constituent independence are the conventional Rayleigh criterion and its noise-modified variant, the basis matrix condition number relative to the all-constituent signal-to-noise ratio (SNR), and a newly defined maximum correlation between model parameters; diagnostics to assess constituent significance are the SNR and percent energy. A confidence interval estimation method for current ellipse parameters, based on complex bivariate normal statistics, is presented that generalizes the colored Monte Carlo method of Pawlowicz et al (2002): the model parameter covariance matrix is not constrained to a presumed form and is scaled using both auto- and cross-spectra of the residual, as computed by fast Fourier transform or Lomb-Scargle periodogram in the case of regularly or irregularly distributed times respectively.

Descriptions are provided for the functionality and syntax of a pair of Matlab functions denoted “UTide”—*ut_solv()* and *ut_reconstr()*—that implement the unified analysis and prediction framework. Output of *ut_solv()* includes a table of all diagnostics, organized to make constituent selection efficient. The robust iteratively-reweighted least squares (IRLS) L^1/L^2 solution method, explored by Leffler and Jay (2009) for the one-dimensional case with uniformly distributed times, is used because it limits sensitivity to outliers and can substantially reduce confidence intervals. Prior methods (for example, capabilities of the *t_tide* Matlab package of Pawlowicz et al. (2002), including the automated decision tree of Foreman (1977) for constituent selection) are available using option flags: ordinary least squares can be used (instead of IRLS); nodal corrections and/or Greenwich phase lag calculations can be omitted, or carried out using linearized (instead of exact) times; inference can use the traditional approximate method (instead of the exact formulation); and confidence intervals can be estimated using the linearized method (instead of Monte Carlo simulation), and/or using the white noise floor assumption (instead of scaled by colored residual spectra). Reconstructed superposed harmonic fits (hind-casts or forecasts/predictions) can be generated by *ut_reconstr()* at arbitrarily chosen sets of times, using subsets of constituents (for example, based on meeting a SNR threshold, or as specified by the user). Finally, the same treatment can be applied to each record in a group of records—such as observations from multiple buoy sites and/or multiple depths, or numerical simulation output from multiple model grid nodes—by a single execution of *ut_solv()* and *ut_reconstr()*.

Table of contents

Abstract	1
I. Introduction	4
II. Unified equation development	8
II.A. Model equations	8
II.A.1. General case: complex, two-dimensional.....	8
II.A.1.a. Pre-filtering and nodal/satellite corrections; Greenwich phase lags	9
II.A.1.b. Current ellipse parameters.....	10
II.A.1.c. Constituent inference	11
II.A.1.d. Summary	13
II.A.2. One-dimensional case, complex.....	13
II.A.3. Relations to real formulation.....	14
II.A.4. Prior methods as special cases.....	16
II.A.4.a. Nodal/satellite corrections using linearized times	16
II.A.4.b. Greenwich phase lags: linearized times in astronomical argument	17
II.A.4.c. Approximate inferences.....	17
II.B. Matrix formulation and solution method.....	18
II.B.1. Iteratively reweighted least squares robust solution.....	19
II.C. Confidence intervals	20
II.C.1. Complex bi-variate normal statistics	22
II.C.2. White noise floor case with non-zero cross-correlations.....	23
II.C.3. Colored case using spectra of residuals.....	26
II.C.4. Implementation: Non-reference and reference constituents	28
II.C.5. Implementation: Inferred constituents.....	29
II.C.6. One-dimensional case.....	30
II.D. Constituent selection diagnostics	30
II.D.1. Diagnostics related to constituent independence.....	31
II.D.1.a. Conventional Rayleigh criterion (R^R).....	31
II.D.1.b. Noise-modified Rayleigh criterion (R^{NM})	32
II.D.1.c. Condition number (K) relative to SNR of entire model (SNR_{allc}).....	32
II.D.1.d. Maximum correlation ($Corr_{max}$) between model parameters.....	33
II.D.2. Diagnostics related to constituent significance	35
II.D.2.a. Signal to noise ratio (SNR).....	35
II.D.2.b. Percent energy (PE).....	36
II.D.3. Diagnostics characterizing reconstructed fits (PTV_{all} , PTV_{snrc}).....	36
II.D.4. Considerations for irregularly distributed times.....	37
III. The UTide Matlab functions.....	39
III.A. Obtaining and using UTide	39
III.B. Quick start suggestions.....	39
III.C. Functionality and syntax and for a single record	40
III.C.1. Solving for coefficients with <i>ut_solv()</i>	40
III.C.1.a. Input parameter descriptions	40
III.C.1.b. Output structure <i>coef</i>	41
III.C.1.c. Defaults and options	44
III.C.1.d. Summary diagnostics table and diagnostic plots.....	48
III.C.2 Reconstructing fits with <i>ut_reconstr()</i>	50

III.C.2.a. Input and output parameter descriptions	50
III.C.2.b. Defaults and options.....	50
III.D. Functionality and syntax for groups of records	51
III.E. Relationships to existing software.....	54
III.F. Computational demands	55
IV. Acknowledgements	57
V. References	58

I. Introduction

Development of this unified tidal analysis was motivated mostly by the need to carry out tidal analysis on a multi-year sequence of current observations collected at irregularly spaced times (Codiga 2007). Observational datasets with these characteristics, less common not long ago, are increasingly available; observing system developments mean longer field campaigns are being sustained, and it is also typical for them to have substantial gaps and/or irregularly distributed temporal sampling. However, it is widely recognized that many commonly available standard software packages for tidal analysis, while highly sophisticated and mature in many ways, require special treatment for such a dataset.

Throughout the following, reference is made to the topics of nodal/satellite corrections, computation of Greenwich phase using the astronomical argument, and constituent inference. These issues are explained comprehensively in numerous publications, including the textbook of Godin (1972) (G72), so they will not be reviewed here except superficially. Readers unfamiliar with them are referred to Foreman and Henry (1989) (FH89) and Parker (2007) as examples of accessible entry points to the literature.

One reason there are limitations to the applicability of traditional tidal methods to multi-year records is that results of the standard (linearized times) method for the nodal/satellite corrections becomes inaccurate for records longer than a year or two (e.g. FH89). This necessitates breaking the record into subsets with durations of about a year, subjecting each to separate analyses, and then combining the results in a final step, for which there seems not to be a standard practice. Another limitation of nearly all standard methods is the requirement of uniformly distributed temporal sampling. For irregular temporal sampling, while an effective approach has recently been developed for the one-dimensional case (e.g. sea level) (Foreman et al. 2009) (FCB09), constituent selection methods remain less well-defined, suggesting the need for new diagnostics. In addition, to the extent that the solution method or confidence interval calculation relies on auto- and/or cross-spectral quantities, in the case of irregular times the fast Fourier transform (FFT) relied on by some methods cannot be applied. A Lomb-Scargle least squares spectral estimation approach (e.g., Press et al. 1992) is suitable for this but has not been implemented. These issues are all addressed here.

The primary goal is to develop a tidal analysis approach and accompanying software tool (“UTide”) that (a) integrates several existing tidal analysis methods with each other (Table 1), and (b) includes enhancements specifically designed to enable treatment of multi-year records with irregular temporal sampling. The main foundation is the theory for harmonic analysis laid out by G72, then extended for practical applications by (Foreman 1977; 1978) (F77, F78) with accompanying Fortran programs, and further discussed by FH89. This foundation was coded in to Matlab as the “t_tide” package by Pawlowicz et al. (2002) (PBL02), which has become a widely accepted standard utility in the physical oceanographic community. PBL02 added a confidence interval estimation method that could use the spectral characteristics of the residual instead of presuming a

Table 1. Comparison of features of UTide and prior software products.

	F77, F78	PBL02	LJ09	FCB09	UTide
Nodal/satellite corrections	Post-fit, linearized times	(same)	(same)	In-matrix, exact times	In-matrix; Default exact times; Optional linearized times
Astronomical argument for Greenwich phase	Post-fit, linearized times	(same)	(same)	In-matrix, exact times	In-matrix; Default exact times; Optional linearized times
Consituent inference	Post-fit, approx.	(same)	(same)	In-matrix, exact	In-matrix; Default exact; Optional approx.
Missing-data gaps (in regular time grid)	Linearly interp'd	(same)	(same)	In-matrix (missing points omitted)	Default in-matrix; For regular temporal sampling, optional linearly interpolated
Irregular times	No	(same)	(same)	In-matrix	In-matrix
Confidence interval method	Cosine/sine coeff's; presumed white residual spectra	Current ellipse param's; lin'zd or Monte-Carlo; white or resid. spec. (FFT); simplified covar'nce Matrix	(same)	Cosine/sine coeff's; presumed white residual spectra	Current ellipse param's; lin'zd or Monte-Carlo; white or resid. spec. (FFT regular times, Lomb-Scargle irregular times); general covar'nce matrix
Solution method	OLS	(same)	IRLS	OLS	Default IRLS; Optional OLS
Complex two-dimensional case	Yes	(same)	(same)	No	Yes
Matlab	No	Yes	(same)	No	Yes
Enhanced diagnostics for constituent selection	No	(same)	(same)	(same)	Yes
Analyze multiple records with one execution	No	(same)	(same)	(same)	Yes

white noise floor. They also added the capability to generate confidence intervals for the four current ellipse parameters, from the uncertainties in the cosine and sine model coefficients, using either a linearized form of the nonlinear underlying relations or Monte Carlo uncertainty propagation. Leffler and Jay (2009) (LJ09) investigated robust solution methods and demonstrated that an iteratively-reweighted least squares (IRLS) approach can minimize the influence of outliers, thereby reducing confidence intervals relative to the standard ordinary least squares (OLS) method, leading to important consequences for the constituent selection process. They investigated the one-dimensional case with uniformly distributed times using a modified version of the PBL02 package that implements the IRLS approach with the *robustfit()* function provided in the Matlab Statistics Toolbox. Finally, Foreman et al (2009; FCB09) presented a method and accompanying Fortran code to handle irregularly distributed temporal sampling. In addition, their method includes exact “in-matrix” formulations for nodal/satellite corrections, the astronomical argument for Greenwich phase calculation, and inferences, such that the corrections are accurate over multi-year time periods. Their approach also permits inferring multiple constituents from a single reference constituent. The upper limit of record length for accurate nodal/satellite corrections by the FCB09 method is 18.6 years, beyond which methods without nodal/satellite corrections are applicable (Foreman and Neufeld 1991), although FCB09 demonstrated the utility of their formulation for longer records when nodal/satellite corrections are omitted.

UTide consists of a pair of Matlab functions designed to be easy to understand and implement: *ut_solv()* for analysis, and *ut_reconstr()* to use the analysis results for reconstruction of a time sequence for a hind-cast or forecast/prediction if needed. They are intended to be helpful in streamlining the various stages of most typical tidal analyses, including constituent selection and confidence interval estimation, for both two-dimensional (e.g., tidal currents) and one-dimensional (e.g., sea level) cases. The analysis function accepts records with times that are uniformly or irregularly distributed, and can provide accurate nodal correction results for records with durations of up to 18.6 years. The reconstruction function accepts arbitrary times and permits generation of reconstructed fits using a subset of constituents, for example based on a signal-to-noise (SNR) criterion or as specified by the user. While the functions incorporate a set of optimal default choices that should help make analysis straightforward for users that are less familiar with the details of tidal methods, they also accept options that enable convenient experimentation with different method choices.

In addition to combining most features of the prior approaches described above together in a single package, the following new contributions are made:

- development of single set of equations, for which (a) each of the prior methods can be obtained as a special case, and (b) nodal/satellite corrections, Greenwich phase computation by astronomical argument, and inferences are all included in-matrix instead of being carried out as post-fit corrections;
- backwards compatibility with each of the prior methods, including “mix and match” choices for individual methods of corrections and confidence intervals, which is a useful capability to ground-truth new results against results from prior analyses, as well as to investigate the sensitivity of results to method choices;

- enhanced diagnostics to aid constituent selection, particularly when the input times are irregularly distributed, by presenting signal to noise ratios with constituents ordered by their energy level, and by inclusion of indicators for straightforward application of the conventional Rayleigh criterion, as well as the noise-modified version proposed by Munk and Hasselmann (1964), and a new diagnostic based on the correlations among model parameters;
- an improved confidence interval calculation building on that of PBL02 to use both the auto- and cross-spectral character of the residual, estimate them by Lomb-Scargle periodograms in the case of irregularly distributed times, and apply Monte Carlo uncertainty propagation with fully general model parameters covariance matrix;
- extension of the FCB09 methodology to (a) solution for the two-dimensional case (e.g., tidal currents) in the complex formulation, (b) computation of reconstructed tidal series (superposed harmonics, or the “fit”) at a sequence of arbitrary times other than the input times, and (c) implementation in Matlab; and
- analysis of a group of records by a single execution of the UTide functions, which proves valuable when the tidal analysis is to be applied to multiple records (e.g., an array of fixed observation sites, or output from a simulation at multiple grid points).

II. Unified equation development

II.A. Model equations

This section considers the model equations for a single time sequence, referred to as the “raw input”, which can be observations, numerical simulation output, synthesized records, etc. An example for two dimensional raw input is a record of east and north velocity components from a single depth, or their depth-averages; an example for one dimensional raw input is a sea level record.

The raw inputs can have uniform or non-uniform temporal sampling, and the development is intended to incorporate nodal/satellite corrections for records of duration up to 18.6 years, but could be useful for longer records (as noted above) through omission of the nodal/satellite corrections.

The set of constituents chosen for inclusion in the model, along with the set of constituents (if any) chosen to be inferred and their associated reference constituents, are presumed known in this section, based on a previously completed constituent selection process. Section II.D below addresses diagnostics useful for constituent selection.

The matrix formulation and solution method is taken up in Section II.B and Section II.C covers the confidence interval calculations.

II.A.1. General case: complex, two-dimensional

The model equation in the most general case (two-dimensional, complex notation, including inferences) is presented here first. Section II.A.2 explains how the one-dimensional case follows as a subset of these general equations, and Section II.A.3 explains the relationships between the complex formulation and the corresponding real formulation, which are useful for the confidence interval calculation.

The most general case, described in this subsection, is the basis for the UTide Matlab code, since all prior methods (see Table 1) and all subsets (e.g., one-dimensional case, real notation case) can be obtained as special cases.

Symbols retain their meaning throughout the document. To the extent possible, they have been selected for consistency with the prior developments cited in Table 1.

The raw input consists of real-valued $u^{raw}(t_i)$ and real-valued $v^{raw}(t_i)$, where u and v are perpendicular Cartesian components of the velocity and the arbitrarily distributed times are t_i , where $i = 1 \dots n_t$. By convention the two components are directed eastward and northward respectively, but more generally they can be the pair of components along the first and second axes in any right-handed coordinate system. The complex form of the raw input

$$x^{raw}(t_i) = u^{raw}(t_i) + iv^{raw}(t_i) \quad (1)$$

is the quantity for which the model, $x^{\text{mod}}(t_i) = u^{\text{mod}}(t_i) + iv^{\text{mod}}(t_i)$, is constructed.

The model equation in its simplest form is

$$x^{\text{mod}}(t_i) = \sum_{q=1}^{n_{\text{allc}}} (E_{iq} a_q^+ + E_{iq}^* a_q^-) + \bar{x} + \dot{x} \cdot (t_i - t_{\text{ref}}). \quad (2)$$

The summation is over all $q = 1 \dots n_{\text{allc}}$ constituents (non-reference, reference, and inferred, as explained below) in the model. Each constituent has constant complex amplitudes a_q^+, a_q^- for components that rotate counter-clockwise and clockwise in time, which multiply the complex exponential functions E_{iq}, E_{iq}^* explained in detail in the next paragraph below. The mean $\bar{x} = \bar{u} + i\bar{v}$ combines the respective means of the two real components. The trend, if included in the model, has coefficient $\dot{x} = \dot{u} + i\dot{v}$ that similarly combines those of the two real components, and is computed relative to the reference time t_{ref} ; by convention t_{ref} is a time central among the raw input times, and here it is defined as the average of the first and last raw input times,

$$t_{\text{ref}} = (t_1 + t_{n_i}) / 2. \quad (3)$$

II.A.1.a. Pre-filtering and nodal/satellite corrections; Greenwich phase lags

In the complex plane an individual harmonic constituent of radian frequency ω_q consists of a superposed pair of components counter-rotating in time, with complex coefficients denoted by + and - superscripts for counterclockwise- and clockwise-rotation respectively. The counterclockwise- and clockwise-rotating elements that the complex coefficients multiply take exponential forms E_{iq} and E_{iq}^* , respectively, where

$$\begin{aligned} E_{iq} &= E(t_i, \omega_q) = P(\omega_q) \cdot F(t_i, \omega_q) \cdot \exp i[U(t_i, \omega_q) + V(t_i, \omega_q)] \\ &= P_q F_{iq} \exp i(U_{iq} + V_{iq}) \end{aligned} \quad (4)$$

and the shorthand expressions

$$P_q = P(\omega_q), F_{iq} = F(t_i, \omega_q), U_{iq} = U(t_i, \omega_q), \text{ and } V_{iq} = V(t_i, \omega_q) \quad (5)$$

represent the following real-valued functions:

- the correction factor for pre-filtering, $P(\omega_q)$, a dimensionless transfer function of the filter that was applied to the raw input $x_q^{\text{raw}}(t_i)$ prior to the analysis
 - $P(\omega_q)$ is set to unity in the case of no pre-filtering
 - $P(\omega_q)$ can be complex, in which case $\text{Re}(P_q) = \text{Im}(P_q)$;
- the nodal/satellite correction amplitude factor $F(t_i, \omega_q)$ (unitless) and phase offset $U(t_i, \omega_q)$ (radians), evaluated at time t_i for constituent q
 - $F(t_i, \omega_q)$ and $U(t_i, \omega_q)$ are set to unity and zero respectively, for the case of no nodal/satellite corrections
 - in addition, as explained more fully in Section II.A.4.a below, in the traditional linearized times development $F(t_{\text{ref}}, \omega_q)$ and $U(t_{\text{ref}}, \omega_q)$ appear in place of $F(t_i, \omega_q)$ and $U(t_i, \omega_q)$ here; and

- the astronomical argument $V(t_i, \omega_q)$ (radians), which ensures resulting phase lags ($g_q^+, g_q^-, g_q, g_q^u, g_q^v$, introduced below) are relative to the equilibrium tide at Greenwich
 - $V(t_i, \omega_q)$ is replaced by $\omega_q \cdot (t_i - t_{ref})$ in order for reported phase lags to instead be uncorrected “raw” phase lags relative to reference time t_{ref}
 - in addition, as explained more fully in Section II.A.4.b below, in the traditional linearized times development, $V(t_*, \omega_q) + \omega_q \cdot (t_i - t_{ref})$ appears in place of $V(t_i, \omega_q)$ here.

Regarding notation, the traditional symbols for nodal/satellite corrections f and u are capitalized here in order to reduce ambiguity with their commonplace use in oceanographic literature as symbols for the Coriolis parameter and eastward velocity component respectively. In addition, for convenience the Greek symbol ν commonly used for the astronomical argument is replaced by V —also capitalized, to help reduce ambiguity with its commonplace use for the northward velocity component.

II.A.1.b. Current ellipse parameters

In the two-dimensional case the amplitude and phase information for each constituent is conventionally reported as four current ellipse parameters. The complex coefficients have associated positive, real magnitudes A_q^+, A_q^- and associated phases

$\varepsilon_q^+, \varepsilon_q^-$,

$$\begin{aligned} a_q^+ &= A_q^+ \exp i \varepsilon_q^+ \\ a_q^- &= A_q^- \exp i \varepsilon_q^-, \end{aligned} \quad (6)$$

where

$$\begin{aligned} A_q^+ &= |a_q^+| & \varepsilon_q^+ &= \arctan[\text{Im}(a_q^+), \text{Re}(a_q^+)] \\ A_q^- &= |a_q^-| & \varepsilon_q^- &= \arctan[\text{Im}(a_q^-), \text{Re}(a_q^-)], \end{aligned} \quad (7)$$

and the Greenwich phase lags for the rotating components (see, e.g., G72) are

$$\begin{aligned} g_q^+ &= -\varepsilon_q^+ \\ g_q^- &= \varepsilon_q^-. \end{aligned} \quad (8)$$

For an individual constituent the tip of the velocity vector in the complex plane traces out an ellipse during each full period. Current ellipse parameters are expressed in terms of the magnitudes and phases of the complex amplitudes as

$$\begin{aligned} L_q^{smaj} &= (A_q^+ + A_q^-) \\ L_q^{smi\eta} &= (A_q^+ - A_q^-) \\ \theta_q &= \text{mod} \left[\frac{\varepsilon_q^+ + \varepsilon_q^-}{2}, \pi \right] \\ g_q &= -\varepsilon_q^+ + \theta_q. \end{aligned} \quad (9)$$

and are defined as

- the semi-major axis length L_q^{smaj} (positive; same units as u, v);
- the semi-minor axis length $L_q^{smi\eta}$ (positive for counterclockwise rotation in time, negative for clockwise rotation in time; same units as u, v);
- the orientation angle θ_q (positive counterclockwise from the positive u axis, which in the conventional case is eastward; radians, range 0 and π) of the semi-major axis that is (following PBL02) directed toward the positive v axis, which in the conventional case is northward; and
- the Greenwich phase lag g_q (radians, range 0 to 2π) of the vector velocity relative to the time of its alignment with the semi-major axis that has a component directed toward positive v , which in the conventional case is northward.

Note that g_q with no superscript denotes the Greenwich phase lag of the vector velocity, and care should be taken to avoid confusion between it and the Greenwich phase lags g_q^+, g_q^- (8) of the counterclockwise- and clockwise-rotating components, as well as the Greenwich phase lags g_q^u, g_q^v of the u and v components, defined in (24) below. The inverse relations for the complex amplitudes,

$$\begin{aligned} a_q^+ &= [(L_q^{smaj} + L_q^{smi\eta})/2] \exp i(\theta_q - g_q) \\ a_q^- &= [(L_q^{smaj} - L_q^{smi\eta})/2] \exp i(\theta_q + g_q), \end{aligned} \quad (10)$$

prove useful in the confidence intervals development below. By convention, the four current ellipse parameters (9) are reported, but the equivalent information could be reported as complex magnitudes A_q^+, A_q^- and Greenwich phase lags g_q^+, g_q^- from (7), or as the real amplitudes A_q^u, A_q^v and Greenwich phase lags g_q^u, g_q^v of the u and v components, defined in (24) below.

II.A.1.c. Constituent inference

To make constituent inference clear, model equation (2) is rewritten in the form

$$\begin{aligned} x^{\text{mod}}(t_i) &= \sum_{j=1}^{n_{NR}} (E_{ij} a_j^+ + E_{ij}^* a_j^-) + \sum_{k=1}^{n_R} \left[E_{ik} \hat{a}_k^+ + E_{ik}^* \hat{a}_k^- + \sum_{l_k=1}^{n_I^k} (E_{il_k} \hat{a}_{l_k}^+ + E_{il_k}^* \hat{a}_{l_k}^-) \right] \\ &+ \bar{x} + \dot{x} \cdot (t_i - t_{ref}). \end{aligned} \quad (11)$$

Here, the first summation represents contributions of a sequence of harmonic constituents, with complex coefficients a_j^+, a_j^- for $j = 1 \dots n_{NR}$, that are denoted “non-reference” (subscript NR), because they are not used as reference constituents to infer other constituents. The second summation is non-zero only if inferred constituents are included in the model; it represents the combined contributions from both (a) the sequence of $k = 1 \dots n_R$ reference constituents (subscript R), in its first two terms, and (b) a

total of $n_I = \sum_{k=1}^{n_R} n_I^k$ inferred constituents (subscript I), including a sequence of $l_k = 1 \dots n_I^k$

inferred constituents for the k th reference constituent, in the interior summation term.

The hat notation for the complex coefficients \hat{a}_k^+, \hat{a}_k^- indicates reference constituents, and

the double-hat notation for the complex coefficients $\hat{a}_{l_k}^+, \hat{a}_{l_k}^-$ indicates inferred constituents. Thus all constituents in the model collectively number $n_{allc} = n_{NR} + n_R + n_I$.

It can thus be seen that, in the expressions prior to (11) above, (a) $q = 1 \dots n_{allc}$ denotes any of the indices j, k , or l_k and (b) for a, A, ε, g variables, and current ellipse parameters, the hat and double-hat notation is implied; for example, a_q^+ represents a_j^+ , \hat{a}_k^+ , or $\hat{a}_{l_k}^+$.

By definition, an inferred constituent is characterized by an amplitude ratio and phase offset that are known, relative to its designated reference constituent, from auxiliary information prior to the tidal analysis at hand. The real-valued inference amplitude ratios $r_{l_k}^+, r_{l_k}^-$ and phase offsets $\varsigma_{l_k}^+, \varsigma_{l_k}^-$,

$$\begin{aligned} r_{l_k}^+ &= \hat{A}_{l_k}^+ / \hat{A}_k^+ & r_{l_k}^- &= \hat{A}_{l_k}^- / \hat{A}_k^- \\ \varsigma_{l_k}^+ &= \hat{g}_k^+ - \hat{g}_{l_k}^+ & \varsigma_{l_k}^- &= \hat{g}_k^- - \hat{g}_{l_k}^- \end{aligned} \quad (12)$$

are specified for the l_k th inferred constituent relative to the k th reference constituent.

The complex coefficients of the inferred constituents are

$$\begin{aligned} \hat{a}_{l_k}^+ &= R_{l_k}^+ \hat{a}_k^+ \\ \hat{a}_{l_k}^- &= R_{l_k}^- \hat{a}_k^- \end{aligned} \quad (13)$$

where the complex inference constants are defined

$$\begin{aligned} R_{l_k}^+ &= r_{l_k}^+ \exp(i\varsigma_{l_k}^+) \\ R_{l_k}^- &= r_{l_k}^- \exp(-i\varsigma_{l_k}^-) \end{aligned} \quad (14)$$

The model equation that is recast in matrix form below, and solved in practice, results on substitution of (13) into (11) and is

$$x^{\text{mod}}(t_i) = \sum_{j=1}^{n_{NR}} (E_{ij} a_j^+ + E_{ij}^* a_j^-) + \sum_{k=1}^{n_R} (\tilde{E}_{ik}^+ \hat{a}_k^+ + \tilde{E}_{ik}^{-*} \hat{a}_k^-) + \bar{x} + \dot{x} \cdot (t - t_{ref}), \quad (15)$$

in which the modified exponential functions are, following FCB09 but here in the complex formulation,

$$\begin{aligned} \tilde{E}_{ik}^+ &= E_{ik} \left(1 + \sum_{l_k=1}^{n_I^k} Q_{il_k} R_{l_k}^+ \right) \\ \tilde{E}_{ik}^- &= E_{ik} \left(1 + \sum_{l_k=1}^{n_I^k} Q_{il_k} R_{l_k}^{-*} \right) \end{aligned} \quad (16)$$

The latter expressions include summations over the $l_k = 1 \dots n_I^k$ constituents to be inferred from the k th reference constituent, and the unitless complex weighting parameter for inferences

$$Q_{il_k} = Q(t_i, \omega_{l_k}) = \frac{E(t_i, \omega_{l_k})}{E(t_i, \omega_k)} = \frac{E_{il_k}}{E_{ik}} = \frac{P_{l_k} F_{il_k}}{P_k F_{ik}} \exp i(U_{il_k} - U_{ik} + V_{il_k} - V_{ik}). \quad (17)$$

Note that (15) is cast in terms of (a) the complex coefficients for the non-reference and reference constituents, and (b) the complex inference constants (14), which appear in the modified exponential functions of (16). Because the complex coefficients of the inferred constituents do not appear in (15), the inferred constituents are solved for indirectly.

It should be emphasized that this formulation for inferences is exact. In contrast to the traditional approximate method (details provided in Section II.A.4.c below), in which the amplitude and phase of the reference frequency are corrected post-solution using an approximation, here inferences are accomplished inherently as part of the model equation (FCB09) and can affect all the other constituents (not just their reference constituents). Furthermore, the present formulation permits inference of multiple constituents from a single reference frequency, which is not the case in the standard traditional method. Finally, unlike the traditional method, this treatment of inferences does not break down where the amplitude of the reference constituent tends toward zero, such as near amphidromic points, as Godin (1972) pointed out is the case for the traditional method.

II.A.1.d. Summary

The model characteristics can be summarized in terms of the input and output information. The input information consists of

- complex-valued raw input $x^{raw}(t_i)$, formed as the combination of the real-valued raw inputs $u^{raw}(t_i)$ and $v^{raw}(t_i)$, as in (1);
- the names and frequencies of n_{NR} non-reference constituents to be included;
- if inferred constituents will be included,
 - the names and frequencies of n_R reference constituents to be included
 - the names and frequencies of a total of n_I inferred constituents (n_I^k inferred constituents for the k th reference constituent), together with real-valued inference amplitude ratios r_k^+ , r_k^- and phase offsets $\zeta_{l_k}^+$, $\zeta_{l_k}^-$ for each inferred constituent relative to its reference constituent; and
- whether or not a trend is to be included in the model.

The output information conventionally consists of

- four current ellipse parameters (L^{smaj} , $L^{smi\eta}$, θ , g) for each of the $n_{allc} = n_{NR} + n_R + n_I$ (non-reference, reference, and inferred) constituents;
- mean values \bar{u} and \bar{v} ; and
- trend coefficients \dot{u} and \dot{v} , if the trend was included.

II.A.2. One-dimensional case, complex

Exposition of how the above general equations for two-dimensional raw input are simplified in the special case of one-dimensional raw input is useful. In this case the raw input is the real-valued $\eta^{raw}(t_i)$, representing any one-dimensional quantity, for example sea level. In the above equations, substitute η everywhere for u , and take all v

parameters to be identically zero. The result is that for any constituent (subscripts and/or hats dropped),

$$a^+ = a^{-*} \quad (18)$$

or equivalently

$$\begin{aligned} A^+ &= A^- \\ \varepsilon^+ &= -\varepsilon^-, \end{aligned} \quad (19)$$

and therefore the current ellipse is degenerate ($L^{smi\eta} = 0$, $\theta = 0$) and lies along the real axis with real amplitude and Greenwich phase lag

$$\begin{aligned} A^\eta &= L^{smaj} = A^+ = A^- \\ g^\eta &= -\varepsilon^+ = \varepsilon^-. \end{aligned} \quad (20)$$

With respect to inference, for known real-valued inference amplitude ratios and phase offsets

$$\begin{aligned} r_{l_k}^\eta &= \hat{A}_{l_k}^\eta / \hat{A}_k^\eta \\ \zeta_{l_k}^\eta &= \hat{g}_k^\eta - \hat{g}_{l_k}^\eta \end{aligned} \quad (21)$$

it follows that the modified exponential functions are $\tilde{E}_{ik}, \tilde{E}_{ik}^*$, where

$$\begin{aligned} \tilde{E}_{ik} &= E_{ik} \left(1 + \sum_{l_k}^{n_j^k} Q_{il_k} R_{l_k}^\eta \right) \\ & (= \tilde{E}_{ik}^+ = \tilde{E}_{ik}^-), \end{aligned} \quad (22)$$

and

$$\begin{aligned} R_{l_k}^\eta &= r_{l_k}^\eta \exp i \zeta_{l_k}^\eta \\ & (= R_{l_k}^+ = R_{l_k}^{-*}). \end{aligned} \quad (23)$$

II.A.3. Relations to real formulation

It is valuable to understand the relation of the above complex formulation to its real counterparts—both to gain a more intuitive understanding of the one-dimensional analysis, and also for later use in the derivation of confidence intervals (Section II.C). The two-dimensional case is treated first, followed by the one-dimensional case.

Underlying the complex form of the model equation above are expressions for the real-valued components of the q th constituent (non-reference, reference, or inferred),

$$\begin{aligned} u_q^{\text{mod}}(t_i) &= A_q^u P_q F_{iq} \cos(U_{iq} + V_{iq} - g_q^u) + \bar{u} + \dot{u} \cdot (t_i - t_{ref}) \\ v_q^{\text{mod}}(t_i) &= A_q^v P_q F_{iq} \cos(U_{iq} + V_{iq} - g_q^v) + \bar{v} + \dot{v} \cdot (t_i - t_{ref}), \end{aligned} \quad (24)$$

where q represents any of the j , k , or l_k indices, and throughout this subsection hat and double-hat variables are not explicitly shown but all relations apply to them. Here, $A_q^u, g_q^u, A_q^v, g_q^v$ are the real-valued amplitudes and Greenwich phase lags of the respective Cartesian velocity components. Real-valued cosine and sine coefficients are defined as

$$X_q^u = A_q^u \cos g_q^u \quad X_q^v = A_q^v \cos g_q^v \quad (25)$$

$$Y_q^u = A_q^u \sin g_q^u \quad Y_q^v = A_q^v \sin g_q^v,$$

such that corresponding relations to the amplitudes and Greenwich phase lags are

$$\begin{aligned} A_q^u &= \sqrt{X_q^{u^2} + Y_q^{u^2}} & A_q^v &= \sqrt{X_q^{v^2} + Y_q^{v^2}} \\ g_q^u &= \arctan(Y_q^u, X_q^u) & g_q^v &= \arctan(Y_q^v, X_q^v). \end{aligned} \quad (26)$$

By use of shorthand definitions

$$\begin{aligned} C_{iq} &= P_q F_{iq} \cos(U_{iq} + V_{iq}) \\ S_{iq} &= P_q F_{iq} \sin(U_{iq} + V_{iq}) \end{aligned} \quad (27)$$

and sine and cosine addition formulae, (24) is written

$$\begin{aligned} u_q^{\text{mod}}(t_i) &= C_{iq} X_q^u + S_{iq} Y_q^u \\ v_q^{\text{mod}}(t_i) &= C_{iq} X_q^v + S_{iq} Y_q^v, \end{aligned} \quad (28)$$

leading to the relation between the complex and real formulations,

$$\begin{aligned} C_{iq} X_q^u + S_{iq} Y_q^v + i(C_{iq} X_q^v + S_{iq} Y_q^u) \\ = E_{iq} a_q^+ + E_{iq}^* a_q^-. \end{aligned} \quad (29)$$

The complex coefficients can be expressed in terms of complex combinations of the cosine and sine coefficients,

$$X_q = X_q^u + iX_q^v \quad \text{and} \quad Y_q = Y_q^u + iY_q^v, \quad (30)$$

as

$$a_q^+ = (X_q - iY_q)/2 \quad \text{and} \quad a_q^- = (X_q + iY_q)/2, \quad (31)$$

and therefore take the form

$$\begin{aligned} a_q^+ &= [(X_q^u + Y_q^v) + i(X_q^v - Y_q^u)]/2 \\ a_q^- &= [(X_q^u - Y_q^v) + i(X_q^v + Y_q^u)]/2. \end{aligned} \quad (32)$$

such that

$$\begin{aligned} A_q^+ &= \frac{1}{2} \sqrt{(X_q^u + Y_q^v)^2 + (X_q^v - Y_q^u)^2} \\ \varepsilon_q^+ &= \arctan[(X_q^v - Y_q^u), (X_q^u + Y_q^v)] \\ A_q^- &= \frac{1}{2} \sqrt{(X_q^u - Y_q^v)^2 + (X_q^v + Y_q^u)^2} \\ \varepsilon_q^- &= \arctan[(X_q^v + Y_q^u), (X_q^u - Y_q^v)]. \end{aligned} \quad (33)$$

The inverse relations of (31) are, using (30),

$$\begin{aligned} X_q &= X_q^u + iX_q^v = a_q^+ + a_q^- \\ -iY_q &= Y_q^v - iY_q^u = a_q^+ - a_q^-, \end{aligned} \quad (34)$$

or equivalently

$$\begin{aligned} X_q^u &= \text{Re}(a_q^+ + a_q^-) & Y_q^u &= -\text{Im}(a_q^+ - a_q^-) \\ X_q^v &= \text{Im}(a_q^+ + a_q^-) & Y_q^v &= \text{Re}(a_q^+ - a_q^-), \end{aligned} \quad (35)$$

which are useful, for example, to compute the real amplitudes and Greenwich phase lags of the u and v components via (26), when the complex coefficients are known.

In the one-dimensional case, for all v variables identically zero and $a^+ = a^{-*}$ (18), replacing u by η , it follows that

$$X_q = X_q^\eta = a_q^+ + a_q^- = a_q^+ + a_q^{+*} = 2 \operatorname{Re}(a_q^+) = 2 \operatorname{Re}(a_q^-) \quad (36)$$

$$Y_q = Y_q^\eta = i(a_q^+ - a_q^-) = i(a_q^+ - a_q^{+*}) = 2i \operatorname{Im}(a_q^+) = -2i \operatorname{Im}(a_q^-),$$

which results in some simplification of the above relations. With respect to inference, the complex form $\tilde{E}_{iq}^+ a_q^+ + \tilde{E}_{iq}^{-*} a_q^-$ is recast as

$$\tilde{C}_{iq}^\eta X_q^\eta + \tilde{S}_{iq}^\eta Y_q^\eta \quad (37)$$

where (as shown by FCB09)

$$\begin{aligned} \tilde{C}_{ik}^\eta &= C_{ik} + \sum_{l_k=1}^{n_l^k} (C_{il_k} R_{l_k}^c - S_{il_k} R_{l_k}^s) \\ \tilde{S}_{ik}^\eta &= S_{ik} + \sum_{l_k=1}^{n_l^k} (C_{il_k} R_{l_k}^s + S_{il_k} R_{l_k}^c), \end{aligned} \quad (38)$$

using inference constants $R_{l_k}^c, R_{l_k}^s$ defined as

$$\begin{aligned} R_{l_k}^c &= r_{l_k}^\eta \cos \zeta_{l_k}^\eta \\ R_{l_k}^s &= r_{l_k}^\eta \sin \zeta_{l_k}^\eta. \end{aligned} \quad (39)$$

II.A.4. Prior methods as special cases

The equations in the preceding sections generalize those of FCB09 and differ from various earlier methods, including PBL02, in three main ways. The first difference is that nodal/satellite corrections use exact times, instead of estimated linearized times. The second difference is that the astronomical argument uses exact times. The third difference is that inferences are handled in an exact way instead of using an approximation. This subsection explains the modifications to the above development that are required in order to recover results from the earlier methods.

The decision to use an earlier method in the UTide functions can be made independently for some or all (in ‘mix and match’ fashion) of these three differences. This enables complete sensitivity analyses to be carried out when investigating the relative importance of the differences.

II.A.4.a. Nodal/satellite corrections using linearized times

In order to implement the nodal/satellite corrections using linearized times instead of exact times, throughout the above development replace $F(t_i, \omega_q)$ and $U(t_i, \omega_q)$ with $F(t_{ref}, \omega_q)$ and $U(t_{ref}, \omega_q)$, respectively. As above, q represents any of the j , k , or l_k indices. The fixed reference time t_{ref} is arbitrary but usually taken to be a time that is central among the raw input times, in order to increase the accuracy of the corrections; here t_{ref} is as defined in (3).

As noted above, in order to omit nodal/satellite corrections entirely, $F(t_i, \omega_q)$ and $U(t_i, \omega_q)$ are replaced everywhere by one and zero, respectively.

II.A.4.b. Greenwich phase lags: linearized times in astronomical argument

In order for linearized times to be used instead of exact times, in the conversion to Greenwich phase lag by correction with the astronomical argument, replace $V(t_i, \omega_q)$ everywhere by $V(t_{ref}, \omega_q) + \omega_q \cdot (t_i - t_{ref})$.

As noted above, in order to entirely omit the conversion to Greenwich phase lags, $V(t_i, \omega_q)$ should be replaced everywhere by $\omega_q \cdot (t_i - t_{ref})$, with the result that g values are raw phase lags relative to the reference time.

II.A.4.c. Approximate inferences

In cases where the nodal/satellite corrections are either omitted or carried out using linearized times, and the Greenwich phase calculations are also either omitted or carried out using linearized times—that is, in cases for which neither of these two corrections are exact—inferences by the approximate method (see G72, F77, and F78 for the derivation and explanation) can be recovered as follows. Note that the customary application of the approximate method precludes use of a single reference constituent for multiple inference constituents. As a result the index l_k never differs from 1, or equivalently $n_l^k = 1$ for any reference constituent k .

- First, solve the model equation using $R_{l_k}^+ = R_{l_k}^- = 0$ for all k . This has the effect that the non-reference and reference constituents are treated identically in the model, such that the first and second summations in (15) differ only in the ranges of their indices.
- Then, use the results \hat{a}_k^+, \hat{a}_k^- from that solution, for the complex coefficients of the reference constituents, to compute corrected complex coefficients of the reference constituents

$$\begin{aligned}\hat{a}_{k,corr}^+ &= \hat{a}_k^+ \frac{1}{1 + \beta_{l_k} R_{l_k}^+ Q(t_{ref}, \omega_{l_k})} \\ \hat{a}_{k,corr}^- &= \hat{a}_k^- \frac{1}{1 + \beta_{l_k} R_{l_k}^- Q^*(t_{ref}, \omega_{l_k})},\end{aligned}\tag{40}$$

where Q is as defined in (17) and

$$\beta_{l_k} = \frac{\sin((t_{n_l} - t_1)(\omega_{l_k} - \omega_k))}{(t_{n_l} - t_1)(\omega_{l_k} - \omega_k)}.\tag{41}$$

- Finally, use $\hat{a}_{k,corr}^+, \hat{a}_{k,corr}^-$ instead of \hat{a}_k^+, \hat{a}_k^- (i) in (7) and (9) to compute the current ellipse parameters for the reference constituents, and (ii) in (13) to calculate the complex coefficients of the inferred constituents, from which the current ellipse parameters for the inferred constituents are computed.

II.B. Matrix formulation and solution method

The model equation (15) as developed above is readily cast in matrix form

$$x^{\text{mod}} = Bm, \quad (42)$$

where symbols without subscripts are understood to be matrices, column vectors, or row vectors throughout this section. The modeled values x^{mod} are a $n_t \times 1$ column vector, complex in the two-dimensional case and real in the one-dimensional case. The basis functions comprise the complex-valued $n_t \times n_m$ matrix B , where $n_m = 2(n_{NR} + n_R) + 2$ is the number of model parameters directly solved for when the trend is included in the model, that has form

$$B = [E \quad E^* \quad \tilde{E}^+ \quad \tilde{E}^{-*} \quad I(n_t, 1) \quad t] \quad (43)$$

with sub-matrices

- E , a complex-valued $n_t \times n_{NR}$ matrix with elements E_{ij} defined as in (4);
- \tilde{E}^+, \tilde{E}^- , each a complex-valued $n_t \times n_R$ matrix with elements $\tilde{E}_{ik}^+, \tilde{E}_{ik}^-$ defined as in (16);
- $I(n_t, 1)$, an $n_t \times 1$ column vector of unit values; and
- t , an $n_t \times 1$ column vector with real elements $(t_i - t_{ref}) / (t_n - t_1)$, normalized such that they are order one and unitless as are the other elements of B , in order to keep it well-conditioned.

The model parameters vector m is a $n_m \times 1$ complex-valued column vector of form, when the trend is included in the model,

$$m = [a_1^+ \quad \cdots \quad a_{n_{NR}}^+ \quad a_1^- \quad \cdots \quad a_{n_{NR}}^- \quad \hat{a}_1^+ \quad \cdots \quad \hat{a}_{n_R}^+ \quad \hat{a}_1^- \quad \cdots \quad \hat{a}_{n_R}^- \quad \bar{x} \quad \dot{x}']^T, \quad (44)$$

where $\dot{x}' = (t_n - t_1) \cdot \dot{x}$ in order to accommodate the normalization of t . When the trend is not included in the model, the final column of (43) and the final element of (44) are omitted, and the number of model parameters is $n_m = 2(n_{NR} + n_R) + 1$.

The matrix formulation is cast in terms of complex-valued matrices. For most cases an equivalent formulation using real-valued matrices exists; an example, though its equations were not explicitly presented by PBL02, is the `t_tide` Matlab code. The complex formulation is used here because it facilitates solution of the case of exact inferences with two-dimensional raw input, for which there is no equivalent in real-valued matrices, and because the confidence interval development is in terms of complex bi-variate normal statistics of the coefficients. The complex formulation is also the most general, and proves to be convenient, although it should be noted that in some cases it may not be the most efficient computationally.

The problem reduces to determining the set of model parameters that minimizes a suitable measure of the residual, or misfit between the raw input and the model,

$$e = x^{\text{raw}} - x^{\text{mod}} = x^{\text{raw}} - Bm. \quad (45)$$

The residual is an $n_t \times 1$ complex-valued column vector with real-valued corresponding Cartesian components

$$e^u = u^{\text{raw}} - u^{\text{mod}} = \text{Re}(x^{\text{raw}} - Bm) \quad (46)$$

$$e^v = v^{raw} - v^{mod} = \text{Im}(x^{raw} - Bm)$$

that are of use in the confidence interval calculation below. As long as n_t exceeds n_m , the system is over-determined with $n_t - n_m$ degrees of freedom and the standard solution method (e.g., F77, F78, PBL02) is ordinary least squares (OLS). The OLS solution minimizes the L^2 norm of e and takes the form

$$m = (B^H B)^{-1} B^H x^{raw}, \quad (47)$$

where superscript H indicates the transpose-conjugate or Hermitian adjoint. To determine the OLS solution, the UTide code uses the built-in Matlab ‘backslash’ operator.

Once the solution is complete, the resulting model parameters vector (44) comprises a set of complex coefficients for the non-reference and reference constituents. The complex coefficients of the inferred constituents are then determined from those of the reference constituents using (13) with the known complex inference constants (14). Finally, for all constituents, the complex magnitudes and phases are then determined using (7), from which the current ellipse parameters are computed from (9).

From the solution a hind-cast or a forecast/prediction time sequence can be reconstructed at any arbitrary set of times. The simplest way to carry this out is using (2). The arbitrarily distributed times t_i^* at which the reconstructed values are to be computed are used, together with the pre-filtering coefficient, the nodal/satellite amplitude/phase corrections and the astronomical argument, to compute E_{ij}, E_{ik}, E_{il_k} (4). The reconstruction uses E_{ij}, E_{ik}, E_{il_k} in (2) with the known model outputs, namely (i) the complex amplitudes $a_j^+, a_j^-, \hat{a}_k^+, \hat{a}_k^-$ of non-reference and reference constituents that were solved for directly, (ii) the complex amplitudes of inferred constituents $\hat{a}_{l_k}^+, \hat{a}_{l_k}^-$, computed using (13) with the known amplitude ratios and phase offsets, (iii) the mean \bar{x} , and (iv) if included, the trend \dot{x} and its reference time t_{ref} .

II.B.1. Iteratively reweighted least squares robust solution

Robust methods using L^1/L^2 hybrid norms offer a number of advantages, as explored by LJ09. Confidence intervals can be reduced substantially, relative to those for the OLS method. In practice, the reduction in confidence intervals relative to the OLS solution method is commonly larger than the differences among confidence interval results when calculated by the white or colored methods, and/or the linearized or Monte Carlo methods (described in detail in section II.C below). In addition, because reduced confidence intervals increase SNR, this can mean that a substantially larger number of constituents will be considered significant (for example, when using a fixed minimum SNR threshold, Section II.D), and therefore selected for inclusion in the model, as compared to the OLS case.

In the iteratively-reweighted least squares (IRLS) approach, a weighting of the observations is determined as part of the solution, such that the influence of outliers is

minimized. In this case, the minimized quantity is a measure of the weighted residual, or weighted misfit between the raw input and the model,

$$e_w = We = W(x^{raw} - Bm), \quad (48)$$

and similarly

$$e_w^u = We^u \quad e_w^v = We^v, \quad (49)$$

where W is an $n_t \times n_t$ diagonal matrix with the scalar weight values on the diagonal. In this case the general solution is

$$m = (B^H W B)^{-1} B^H W x^{raw}. \quad (50)$$

In the case of equally-weighted observations, W is the identity matrix and (50) reduces to the OLS solution as expected.

For IRLS cases, following LJ09, the UTide code uses the Matlab Statistics Toolbox function *robustfit()*, an iterative solver which implements a user-specified weight function selected from among a range of common shapes, with a corresponding constant value of the tuning parameter (see Matlab documentation for *robustfit()*). The output, for given choices of weight function and tuning parameter, includes the set of optimal model parameters with the corresponding weight matrix W for the raw inputs.

There are no established guidelines regarding the choice of weight function and tuning parameter that is appropriate for any given analysis, and it should be expected that the optimal choices will vary depending on the nature of the raw input (one-dimensional or two-dimensional, noise conditions), as well as the nature of the sampling duration and resolution (e.g., uniformly or irregularly distributed times), in addition to the model configuration (number of constituents, etc). A lower tuning parameter generally causes a greater penalty against outliers and requires a higher number of iterations. Based on analysis of a sea level record from a single location in a tidal estuary, with uniformly distributed times of hourly resolution and duration 6 months, LJ09 concluded that the best weight function was Cauchy and the best tuning parameter was 0.795 (the Matlab default value 2.385, reduced by a factor of 3).

In UTide the default weight function is Cauchy and alternative weight functions can be specified by an optional input (e.g., ‘Huber’,...). The default tuning parameter used by UTide is that provided by Matlab for the given weight function (see Matlab documentation for *robustfit()*), and an optional UTide input (‘TunRdn’) can reduce the tuning parameter by a specified factor (e.g. 3, as found optimal by LJ09 for their example record). Empirical experimentation will be needed for analysis of any given record, to determine the appropriate weighting function and tuning parameter.

II.C. Confidence intervals

A method to compute uncertainties of the cosine/sine model parameters using the basis function matrix was explained by G72, F77, and FH89. In contrast to that method, which did not distinguish the spectral nature of the residual and can thus be referred to as a presumed “white noise floor” approach, the method put forth by PBL02 and denoted “colored” was based on using spectral properties computed from the actual residual. In addition, PBL02 presented two methods to determine confidence intervals for the current

ellipse parameters, which are not model parameters themselves but are nonlinear functions of the model parameters: a linearization approach, and Monte Carlo uncertainty propagation (which they denoted the “bootstrap” method).

This subsection presents a new method, the default case in UTide, that builds on earlier approaches and (a) generates confidence intervals of the current ellipse parameters, using Monte Carlo uncertainty propagation, but generalizes the PBL02 method by relaxing the simplifying assumptions it made about the form of the covariance matrix of functions of the model parameters, (b) is colored and generalized PBL02 to incorporate both the auto-spectra and cross-spectra of the u and v component residuals, using their weighted forms (48) in the case of the IRLS solution method, and (c) is amenable to use of spectral estimates for the latter that have been computed either by FFT for evenly distributed times, or by Lomb-Scargle periodogram for irregularly distributed times.

The default “colored, Monte Carlo” UTide method is explained in the remainder of this section. However, UTide allows the alternative to specify (a) use of the white noise floor assumption instead of the colored residual spectra, and/or (b) use of the linearized approach instead of Monte Carlo. The development below makes clear how the white case can be implemented. The linearized method used is that of PBL02 (and therefore not explicitly presented here), in which it is presumed that there are no non-zero correlations among the coefficients.

For any of these methods, confidence interval estimates are based on the assumption that all energy in the residual is noise. In most typical analyses, the record contains a sub-tidal (i.e. non-tidal, low frequency, and/or weather-band) signal at frequencies lower than tidal, in addition to the random noise and the tidal component. It should therefore be recognized that the sub-tidal signal, if not somehow removed prior to the analysis, contributes to the confidence intervals and will in general make them artificially higher. If not removed, the sub-tidal signal will also tend to make the statistics of the residual deviate from the normal statistics presumed by the confidence interval calculations. For these reasons, to obtain the most accurate confidence intervals, a simple and effective strategy is first to compute the low-pass of the raw inputs and subtract it from them, then to perform the tidal analysis on the resulting difference. This approach, in effect, accomplishes high-pass filtering. It is simpler than formal application of a high-pass or band-pass pre-filter, and should in general not require a pre-filtering correction (P_q in (5) remains unity).

As described above, relevant context is provided by the fact that using the IRLS solution method instead of OLS can have a substantial impact on confidence intervals. In many cases they can be reduced to a greater degree than the differences between when they are computed using the linearized or Monte Carlo method, and/or the white or colored method.

Finally, it is important to note that certain assumptions underlying the development in this section are strictly valid, as discussed in FH89, only when the raw

input times are uniformly spaced. Therefore the results in the case of irregular times should be considered potentially reasonable and approximate first estimates, but should be compared against the results for uniform times whenever possible, and used with a measure of caution.

The subsections that follow develop expressions for a key covariance matrix needed by the Monte Carlo uncertainty propagation method. Implementation of the confidence interval calculation, using that result to generate random realizations for Monte Carlo simulation, is then explained.

II.C.1. Complex bi-variate normal statistics

The model parameters vector m (44), which includes the complex coefficients for the non-reference and reference constituents, is assumed to be a *complex normal vector*; its individual elements are assumed to be complex random variables related to each other by complex bivariate normal statistics. Because the complex coefficients of inferred constituents do not appear in the model parameters vector, due to the fact that inferred constituents are included in the model indirectly (see above), computation of confidence intervals for inferred constituents follows a different approach based on the statistics of the corresponding reference constituent, as explained in Section II.C.5.

A complex normal vector m of size $n_m \times 1$ is characterized by its complex-valued

- (i) mean $\mu = E[m]$, an $n_m \times 1$ vector,
- (ii) covariance $\Gamma_C = E[(m - \mu)(m - \mu)^H]$, a Hermitian $n_m \times n_m$ matrix, and
- (iii) pseudo-covariance $\Gamma_p = E[(m - \mu)(m - \mu)^T]$, a symmetric $n_m \times n_m$ matrix,

where $E[\]$ is the expectation operator and superscripts H and T indicate the complex conjugate transpose or Hermitian adjoint, and the ordinary transpose, respectively. Full generality is retained, in the sense that no presumption is made that the complex variables are “circular” or “proper”, thus allowing for non-zero pseudocovariance Γ_p .

A property of the complex bivariate normal statistics (e.g., Goodman 1963) is that the variance-covariance matrices between the complex random vector formed from the real part of the model parameters vector and the complex random vector formed from its imaginary part are

$$\begin{aligned}
 \text{cov}[\text{Re}(m), \text{Re}(m)] &= \text{Re}(\Gamma_C + \Gamma_p) / 2 \\
 \text{cov}[\text{Re}(m), \text{Im}(m)] &= -\text{Im}(\Gamma_C - \Gamma_p) / 2 \\
 \text{cov}[\text{Im}(m), \text{Re}(m)] &= \text{Im}(\Gamma_C + \Gamma_p) / 2 \\
 \text{cov}[\text{Im}(m), \text{Im}(m)] &= \text{Re}(\Gamma_C - \Gamma_p) / 2 .
 \end{aligned} \tag{51}$$

As will be seen below, these expressions allow the remainder of the development to proceed in terms of real-valued quantities. This is needed to facilitate the use of the Lomb-Scargle periodogram for spectral estimates when the distribution of times is irregular, because they are computed using the real-valued u^{raw}, v^{raw} .

II.C.2. White noise floor case with non-zero cross-correlations

The following generalizes the development of G72, F77, and FH89 to include the two dimensional case, non-zero cross-correlation terms, and the use of complex model parameters instead of real cosine/sine coefficients. The weighted complex error e_w (48) is assumed be a zero-mean complex normal variable. The development is carried out in terms of the IRLS case, including the weight matrix, throughout this section; the OLS case is recovered through use of the identity matrix for the weight matrix. The total error variance σ_C^2 is a real scalar representing the variance of the complex error, equivalent to the mean square residual or mean square misfit (MSM) between the raw input and the model output,

$$\sigma_C^2 = \sigma_{MSM}^2 = \frac{x^{rawH} W x^{raw} - m^H B^H W x^{raw}}{n_t - n_m}. \quad (52)$$

By direct analogy with the expression for the total error variance σ_C^2 , the total error pseudo-variance σ_P^2 is the complex constant

$$\sigma_P^2 = \frac{x^{rawT} W x^{raw} - m^T B^T W x^{raw}}{n_t - n_m}. \quad (53)$$

It follows that estimates of the $n_m \times n_m$ covariance matrix and pseudo-covariance matrix for the model parameters are

$$\begin{aligned} \Gamma_C &= (B^H W B)^{-1} \sigma_C^2 \\ \Gamma_P &= (B^T W B)^{-1} \sigma_P^2. \end{aligned} \quad (54)$$

Implementation of formulas based on (51) is simplified below by defining the sum and difference of the covariance and pseudo-covariance matrices, which here incorporate all model parameters, as

$$\begin{aligned} G^{all} &= \Gamma_C + \Gamma_P \\ H^{all} &= \Gamma_C - \Gamma_P. \end{aligned} \quad (55)$$

Each individual constituent, indexed by $c = 1 \dots n_c$ where $n_c = n_{NR} + n_R$, has two complex model parameters

$$a_c^+ = m_c \quad \text{and} \quad a_c^- = m_{c+n_c}. \quad (56)$$

Throughout Sections II.C.2-4, the symbols a_c^+, a_c^- are used to denote the complex coefficients of either a non-reference constituent or a reference constituent; that is, for reference constituents, the hats on these variables in the previous sections are dropped, such that a_c^+ represents either a_j^+ or \hat{a}_k^+ , for example. The 2 x 2 sub-matrices of G^{all} and

H^{all} pertinent to constituent c are

$$\begin{aligned} G^{cc} &= \begin{bmatrix} G_{c,c}^{all} & G_{c,c+n_c}^{all} \\ G_{c+n_c,c}^{all} & G_{c+n_c,c+n_c}^{all} \end{bmatrix} \\ H^{cc} &= \begin{bmatrix} H_{c,c}^{all} & H_{c,c+n_c}^{all} \\ H_{c+n_c,c}^{all} & H_{c+n_c,c+n_c}^{all} \end{bmatrix}. \end{aligned} \quad (57)$$

It is presumed that the constituent selection process (Section II.D) has been completed such that the covariances and pseudo-covariances between model parameters of different constituents can be neglected.

Because the statistics of the real and imaginary parts of the model parameters (51) are known, it proves convenient to compute the current ellipse parameters for a given constituent from the vector of 4 real-valued (superscript R) parameters

$$m_c^R = [\text{Re}(a_c^+) \quad \text{Im}(a_c^+) \quad \text{Re}(a_c^-) \quad \text{Im}(a_c^-)] \quad (58)$$

rather than from the complex coefficients a_c^+, a_c^- themselves (in m). By definition, the variance-covariance matrix of m_c^R is

$$\text{var cov}(m_c^R) = \text{cov}(m_c^R, m_c^R) = \begin{bmatrix} \sigma_{\text{Re}(a^+)}^2 & \text{cov}[\text{Re}(a^+), \text{Im}(a^+)] & \text{cov}[\text{Re}(a^+), \text{Re}(a^-)] & \text{cov}[\text{Re}(a^+), \text{Im}(a^-)] \\ & \sigma_{\text{Im}(a^+)}^2 & \text{cov}[\text{Im}(a^+), \text{Re}(a^-)] & \text{cov}[\text{Im}(a^+), \text{Im}(a^-)] \\ & & \sigma_{\text{Re}(a^-)}^2 & \text{cov}[\text{Re}(a^-), \text{Im}(a^-)] \\ & & & \sigma_{\text{Im}(a^-)}^2 \end{bmatrix} \quad (59)$$

where the lower left triangle is left blank because the matrix is real-valued and symmetric with 10 unique values. In (59), as in similar expressions in the remainder of this subsection, for clarity the c subscripts are dropped from elements within the matrix. Whereas for simplicity the method of PBL02 (see `t_tide` code) presumed specific relationships among the elements of $\text{var cov}(m_c^R)$, for example that $\sigma_{\text{Re}(a^+)}^2 = \sigma_{\text{Im}(a^+)}^2$ and that $\text{cov}[\text{Re}(a^+), \text{Im}(a^+)] = 0$, these assumptions are relaxed in the present development.

It follows from the statistics of real and imaginary parts of the model parameters (51) that under the white noise floor assumption the unique terms in $\text{var cov}(m_c^R)$ are, in the upper left quadrant,

$$\begin{aligned} \sigma_{\text{Re}(a_c^+)}^2 &= \text{cov}[\text{Re}(a_c^+), \text{Re}(a_c^+)] = \text{Re}(G_{11}^{cc}) / 2 \\ \text{cov}[\text{Re}(a_c^+), \text{Im}(a_c^+)] &= -\text{Im}(H_{11}^{cc}) / 2 \\ \sigma_{\text{Im}(a_c^+)}^2 &= \text{cov}[\text{Im}(a_c^+), \text{Im}(a_c^+)] = \text{Re}(H_{11}^{cc}) / 2 ; \end{aligned} \quad (60)$$

in the lower right quadrant,

$$\begin{aligned} \sigma_{\text{Re}(a_c^-)}^2 &= \text{cov}[\text{Re}(a_c^-), \text{Re}(a_c^-)] = \text{Re}(G_{22}^{cc}) / 2 \\ \text{cov}[\text{Re}(a_c^-), \text{Im}(a_c^-)] &= -\text{Im}(H_{22}^{cc}) / 2 \\ \sigma_{\text{Im}(a_c^-)}^2 &= \text{cov}[\text{Im}(a_c^-), \text{Im}(a_c^-)] = \text{Re}(H_{22}^{cc}) / 2 ; \end{aligned} \quad (61)$$

and in the upper right quadrant,

$$\begin{aligned} \text{cov}[\text{Re}(a_c^+), \text{Re}(a_c^-)] &= \text{Re}(G_{12}^{cc}) / 2 \\ \text{cov}[\text{Re}(a_c^+), \text{Im}(a_c^-)] &= -\text{Im}(H_{12}^{cc}) / 2 \\ \text{cov}[\text{Im}(a_c^+), \text{Re}(a_c^-)] &= \text{Im}(G_{12}^{cc}) / 2 \\ \text{cov}[\text{Im}(a_c^+), \text{Im}(a_c^-)] &= \text{Re}(H_{12}^{cc}) / 2. \end{aligned} \quad (62)$$

The variance-covariance matrix of m_c^R can thus be expressed in terms of G^{cc} and H^{cc} as

$$\text{var cov}^{white}(m_c^R) = \begin{bmatrix} \text{Re}(G_{11}^{cc}) & -\text{Im}(H_{11}^{cc}) & \text{Re}(G_{12}^{cc}) & -\text{Im}(H_{12}^{cc}) \\ & \text{Re}(H_{11}^{cc}) & \text{Im}(G_{12}^{cc}) & \text{Re}(H_{12}^{cc}) \\ & & \text{Re}(G_{22}^{cc}) & -\text{Im}(H_{22}^{cc}) \\ & & & \text{Re}(H_{22}^{cc}) \end{bmatrix} / 2. \quad (63)$$

In order to facilitate more convenient computation of spectral estimates in the next subsection using the Lomb-Scargle periodogram for the case of irregular times, the spectra of Cartesian (u and v) components of the residual are computed, as opposed to, for example, rotary spectra. However, the Cartesian spectral quantities are not directly suitable for scaling $\text{var cov}^{white}(m_c^R)$ (63), which is cast in terms of the complex coefficients; the Cartesian spectral quantities are instead appropriate to scale the variance-covariance matrix of the vector of the four Cartesian (superscript C) cosine/sine coefficients of the constituent (35),

$$m_c^C = [X_c^u \quad Y_c^u \quad X_c^v \quad Y_c^v]. \quad (64)$$

The corresponding symmetric 4x4 variance-covariance matrix needed is, by definition,

$$\text{var cov}(m_c^C) = \text{cov}(m_c^C, m_c^C) = \begin{bmatrix} D_{uu} & D_{uv} \\ D_{uv}^T & D_{vv} \end{bmatrix}, \quad (65)$$

where

$$D_{uu} = \begin{bmatrix} \sigma_{X^u}^2 & \text{cov}(X^u, Y^u) \\ & \sigma_{Y^u}^2 \end{bmatrix}, \quad D_{vv} = \begin{bmatrix} \sigma_{X^v}^2 & \text{cov}(X^v, Y^v) \\ & \sigma_{Y^v}^2 \end{bmatrix}, \text{ and} \quad (66)$$

$$D_{uv} = \begin{bmatrix} \text{cov}(X^u, X^v) & \text{cov}(X^u, Y^v) \\ \text{cov}(Y^u, X^v) & \text{cov}(Y^u, Y^v) \end{bmatrix}.$$

By the defining relations (34) and (35), the 10 unique elements of $\text{var cov}(m_c^C)$ consist of those in D_{uu} ,

$$\begin{aligned} \sigma_{X^u}^2 &= \sigma_{\text{Re}(a^+)}^2 + \sigma_{\text{Re}(a^-)}^2 + 2\text{cov}[\text{Re}(a^+), \text{Re}(a^-)] \\ \text{cov}(X^u, Y^u) &= -\text{cov}[\text{Re}(a^+), \text{Im}(a^+)] + \text{cov}[\text{Re}(a^+), \text{Im}(a^-)] \\ &\quad - \text{cov}[\text{Re}(a^-), \text{Im}(a^+)] + \text{cov}[\text{Re}(a^-), \text{Im}(a^-)] \\ \sigma_{Y^u}^2 &= \sigma_{\text{Im}(a^+)}^2 + \sigma_{\text{Im}(a^-)}^2 - 2\text{cov}[\text{Im}(a^+), \text{Im}(a^-)]; \end{aligned} \quad (67)$$

those in D_{vv}

$$\begin{aligned} \sigma_{X^v}^2 &= \sigma_{\text{Im}(a^+)}^2 + \sigma_{\text{Im}(a^-)}^2 + 2\text{cov}[\text{Im}(a^+), \text{Im}(a^-)] \\ \text{cov}(X^v, Y^v) &= \text{cov}[\text{Im}(a^+), \text{Re}(a^+)] - \text{cov}[\text{Im}(a^+), \text{Re}(a^-)] \\ &\quad + \text{cov}[\text{Im}(a^-), \text{Re}(a^+)] - \text{cov}[\text{Im}(a^-), \text{Re}(a^-)] \\ \sigma_{Y^v}^2 &= \sigma_{\text{Re}(a^+)}^2 + \sigma_{\text{Re}(a^-)}^2 - 2\text{cov}[\text{Re}(a^+), \text{Re}(a^-)]; \end{aligned} \quad (68)$$

and those in D_{uv} ,

$$\text{cov}(X^u, X^v) = \text{cov}[\text{Re}(a^+), \text{Im}(a^+)] + \text{cov}[\text{Re}(a^+), \text{Im}(a^-)] \quad (69)$$

$$\begin{aligned}
& + \text{cov}[\text{Re}(a^-), \text{Im}(a^+)] + \text{cov}[\text{Re}(a^-), \text{Im}(a^-)] \\
\text{cov}(X^u, Y^v) &= \text{cov}[\text{Re}(a^+), \text{Re}(a^+)] - \text{cov}[\text{Re}(a^-), \text{Re}(a^-)] \\
\text{cov}(Y^u, X^v) &= -\text{cov}[\text{Im}(a^+), \text{Im}(a^+)] + \text{cov}[\text{Im}(a^-), \text{Im}(a^-)] \\
\text{cov}(Y^u, Y^v) &= -\text{cov}[\text{Im}(a^+), \text{Re}(a^+)] + \text{cov}[\text{Im}(a^+), \text{Re}(a^-)] \\
& + \text{cov}[\text{Im}(a^-), \text{Re}(a^+)] - \text{cov}[\text{Im}(a^-), \text{Re}(a^-)].
\end{aligned}$$

Finally, based on combining (60)-(62) and (67)-(69), and dropping the cc superscripts on G and H for clarity, the unique elements of $\text{var cov}^{\text{white}}(m_c^C)$ in D_{uu} are

$$\begin{aligned}
\sigma_{X^u}^2 &= [\text{Re}(G_{11}) + \text{Re}(G_{22}) + 2\text{Re}(G_{12})]/2 \\
\text{cov}(X^u, Y^u) &= [\text{Im}(H_{11}) - \text{Im}(H_{12}) + \text{Im}(H_{21}) - \text{Im}(H_{22})]/2 \\
\sigma_{Y^u}^2 &= [\text{Re}(H_{11}) + \text{Re}(H_{22}) - 2\text{Re}(H_{12})]/2;
\end{aligned} \tag{70}$$

in D_{vv} are

$$\begin{aligned}
\sigma_{X^v}^2 &= [\text{Re}(H_{11}) + \text{Re}(H_{22}) + 2\text{Re}(H_{12})]/2 \\
\text{cov}(X^v, Y^v) &= [\text{Im}(G_{11}) - \text{Im}(G_{12}) + \text{Im}(G_{21}) - \text{Im}(G_{22})]/2 \\
\sigma_{Y^v}^2 &= [\text{Re}(G_{11}) + \text{Re}(G_{22}) - 2\text{Re}(G_{12})]/2;
\end{aligned} \tag{71}$$

and in D_{uv} are

$$\begin{aligned}
\text{cov}(X^u, X^v) &= [-\text{Im}(H_{11}) - \text{Im}(H_{12}) - \text{Im}(H_{21}) - \text{Im}(H_{22})]/2 \\
\text{cov}(X^u, Y^v) &= [\text{Re}(G_{11}) - \text{Re}(G_{22})]/2 \\
\text{cov}(Y^u, X^v) &= [-\text{Re}(H_{11}) + \text{Re}(H_{22})]/2 \\
\text{cov}(Y^u, Y^v) &= [-\text{Im}(G_{11}) + \text{Im}(G_{12}) + \text{Im}(G_{21}) - \text{Im}(G_{22})]/2.
\end{aligned} \tag{72}$$

II.C.3. Colored case using spectra of residuals

The residual in general has a non-white, or colored, spectral nature typified by redness. A means by which to incorporate this (PBL02) in the confidence intervals is to scale elements of the covariance matrix derived above for the white noise approach using estimates of the actual residual spectrum at a frequency appropriate to the given constituent. In the IRLS solution case the appropriate residual is the weighted residual (48), as pointed out by LJ09.

Following the approach of PBL02, the spectra are considered “locally white”, and averaged over a fixed set of frequency bands encompassing the main groups of constituents, after lines of constituents included in the model that fall within that band are omitted. The nine averaging bands, in cycles per day (cpd), are $M_0 \pm 0.1$ cpd, $M_1 \pm 0.2$ cpd, $M_2 \pm 0.2$ cpd; $M_3 \pm 0.2$ cpd, $M_4 \pm 0.2$ cpd, $M_5 \pm 0.2$ cpd, $M_6 \pm 0.21$ cpd; 0.26-0.29 cpd (includes M_7), and 0.30-0.50 cpd (includes M_8). For each averaging band, this computation yields three real values, each a line-decimated, band-averaged, one-sided (2 times the two-sided density, except at the zero and Nyquist frequencies, for which the one-sided and two-sided values are the same) spectral density: (i) $\overline{P}_{e_w^u e_w^u}^{1s}$, the auto-spectral density of e_w^u , the weighted u component of the residual; (ii) $\overline{P}_{e_w^v e_w^v}^{1s}$, the auto-spectral

density of e_w^v , the weighted v component of the residual; and (iii) $\overline{P}_{e_w^u e_w^v}^{1s}$, the rel part of the cross-spectral density between e_w^u and e_w^v or cospectrum. The overbars denote the result of the line-decimation and frequency-band averaging.

The auto- and cross-spectral power terms (not densities) that contribute to the collective uncertainties in the model parameters of a constituent, index $c = 1 \dots n_c$, with frequency ω_c that lies within the averaged band, are

$$\begin{aligned} P_c^{uu} &= \overline{P}_{e_w^u e_w^u}^{1s} \Delta\omega \\ P_c^{vv} &= \overline{P}_{e_w^v e_w^v}^{1s} \Delta\omega \\ P_c^{uv} &= \overline{P}_{e_w^u e_w^v}^{1s} \Delta\omega, \end{aligned} \quad (73)$$

where $\Delta\omega$ is the frequency resolution of the spectral calculation. For raw input with length of record (LOR) $t_{n_t} - t_1$, the frequency resolution is $\Delta\omega = 1/LOR_e$ where the effective LOR is

$$LOR_e = (n_t / (n_t - 1)) LOR, \quad (74)$$

so formulated because $LOR_e = n_t \cdot \Delta t$ in the case of evenly spaced times with time separation Δt .

In the case of uniformly distributed times, the spectral quantities are computed by the FFT method after application of a record-length Hanning weighting, using the Matlab Signal Processing Toolbox function *pwelch()*. In the case of irregular times, the spectral estimates are made using the un-normalized, mean-removed, Lomb-Scargle periodogram (Lomb 1976; Scargle 1982; Press et al. 1992). Calculation of the Lomb-Scargle periodogram requires specification of a frequency oversampling factor, the amount by which the grid of frequencies at which the periodogram estimates are computed is more dense than an equivalent FFT. In common applications of the Lomb-Scargle periodogram such as astronomy, the goal is to resolve a spectral peak with a certain degree of confidence, and high oversampling factors (for example 4 or more) are used in order to increase confidence in peak detection. In the present application peak detection is not the goal and an oversampling factor of one is used. This approach is taken for practical reasons as well: for long records the computational burden of the Lomb-Scargle periodogram increases dramatically, meaning that higher oversampling factors will require significantly more computing resources, particularly in terms of memory but also with respect to processing time. Prior to the Lomb-Scargle periodogram calculation, in a manner similar to the approach of Schulz and Stattegger (1997) for irregularly distributed times, a record-length Hanning weighting is applied. As a result, in the case of equally spaced times the UTide Lomb-Scargle periodogram function with oversampling factor of one returns exactly the same result as does *pwelch()*.

The colored variance-covariance matrix is computed by scaling $\text{var cov}^{white}(m_c^C)$ (defined in (70)-(72)) using the residual spectral power. The spectral power P_c^{uu} , for the weighted residual e_w^u of the u component at the frequency of constituent c , contributes

to the variances of X_c^u and Y_c^u . Because the elements of D_{uu} consist of these two variances on the diagonal, and the associated covariance off the diagonal, all elements of D_{uu} are normalized by its trace and then scaled by P_c^{uu} . Similarly, D_{vv} is scaled by P_c^{vv} after normalization by its trace. The cross-spectral power P_c^{uv} contributes to covariances between one u coefficient (X_c^u or Y_c^u) and one v coefficient (X_c^v or Y_c^v). Because all elements of D_{uv} are covariances (including the elements on the diagonal, unlike for D_{uu} and D_{vv}), the elements of D_{uv} are normalized by the sum of the absolute values of all elements (instead of its trace), and then scaled by $|P_c^{uv}|$. These relations are expressed

$$\text{var cov}^{\text{colored}}(m_c^C) = \begin{bmatrix} P_c^{uu} D_{uu} / \text{tr}(D_{uu}) & |P_c^{uv}| D_{uv} / \sum_4 |D_{uv}| \\ |P_c^{uv}| D_{uv}^T / \sum_4 |D_{uv}| & P_c^{vv} D_{vv} / \text{tr}(D_{vv}) \end{bmatrix}, \quad (75)$$

where $\text{tr}()$ is the trace operator and \sum_4 indicates the element-wise summation. In the special case that all elements of D_{uv} are zero, in order to avoid division by zero the upper right four elements and the lower left four elements (all the D_{uv} terms) in (75) are set to zero.

In certain cases, depending on the model configuration and the spectral characteristics of the residual, the estimated $\text{var cov}(m_c^C)$ is not positive semi-definite. Although generally the difference from positive semi-definiteness is minor, when this occurs it is not a valid variance-covariance matrix. In this situation, the nearest positive semi-definite covariance matrix is used instead, as determined using the method of Higham (2002) with identity weight matrix. The method is iterative and if it does not converge, a warning message is provided and the off-diagonal elements are simply set to zero, as an ad-hoc solution.

II.C.4. Implementation: Non-reference and reference constituents

Once the model solution is determined, the optimal complex-valued model parameter vector m (44) is known. For each non-reference or reference constituent of index c (where $c = 1 \dots n_c$), two elements of m are the complex coefficients $m_c = a_c^+$ and $m_{c+n_c} = a_c^-$. These are used, by (35), to compute the four Cartesian cosine/sine coefficients m_c^C (64) for the constituent. Next, the associated variance-covariance matrix $\text{var cov}^{\text{colored}}(m_c^C)$ is computed from (75); this makes use of the spectral quantities (73) from the weighted residuals to scale the results of the white case (70)-(72), which is in turn based on expressions for the complex bivariate normal statistics using the model equation basis function matrix and associated weight matrix (57). Random realizations, denoted $\{m_c^C\}$, are then generated from the known m_c^C and $\text{var cov}^{\text{colored}}(m_c^C)$. In the UTide code, the random realizations of m_c^C are generated using the Matlab Statistics Toolbox function `mvnrnd()`. By Monte Carlo uncertainty propagation through (9) and (33), random realizations of the current ellipse parameters, $\{L^{smaj}\}, \{L^{smi\eta}\}, \{\theta\}, \{g\}$, are

generated. Standard errors of current ellipse parameters are computed using the median-average-deviation formulation,

$$\begin{aligned}
\sigma_{L^{smaj}} &= \text{Median}[|\{L^{smaj}\} - \text{Median}[\{L^{smaj}\}]|] / 0.6745 \\
\sigma_{L^{smi\eta}} &= \text{Median}[|\{L^{smi\eta}\} - \text{Median}[\{L^{smi\eta}\}]|] / 0.6745 \\
\sigma_{\theta} &= \text{Median}[|\{\theta\} - \text{Median}[\{\theta\}]|] / 0.6745 \\
\sigma_g &= \text{Median}[|\{g\} - \text{Median}[\{g\}]|] / 0.6745.
\end{aligned} \tag{76}$$

Finally, the 95% confidence intervals CI are 1.96 times these standard errors, such that, for example, it is 95% probable that L^{smaj} lies between $L^{smaj} - CI_{L^{smaj}}$ and $L^{smaj} + CI_{L^{smaj}}$.

For the white noise case the only change is to use $\text{varcov}^{white}(m_c^C)$ (70)-(72) instead of $\text{varcov}^{colored}(m_c^C)$ (75).

II.C.5. Implementation: Inferred constituents

For each inferred constituent, in the Monte Carlo method, the following approach is used. First, the realizations $\{\hat{L}^{smaj}\}, \{\hat{L}^{smi\eta}\}, \{\hat{\theta}\}, \{\hat{g}\}$ of the current ellipse parameters for its reference constituent, computed as just described, are used to compute realizations $\{\hat{a}^+\}, \{\hat{a}^-\}$ of the complex coefficients for the reference constituent, by (10). Next, realizations $\{\hat{a}^+\}, \{\hat{a}^-\}$ of the complex coefficients for the inferred constituent are computed from those of the reference constituent using (13), then converted to realizations $\{\hat{X}^u\}, \{\hat{Y}^u\}, \{\hat{X}^v\}, \{\hat{Y}^v\}$ of the cosine/sine coefficients of the inferred constituent by (35), and finally to realizations $\{\hat{L}^{smaj}\}, \{\hat{L}^{smi\eta}\}, \{\hat{\theta}\}, \{\hat{g}\}$ of the current ellipse parameters of the inferred constituent, by (9) and (33). From the realizations of the current ellipse parameters of the inferred constituent, the confidence intervals are computed as in (76).

Confidence intervals for inference constituents can also be computed when using the linearized method instead of Monte Carlo. For any constituent (inferred or not), the linearized method computes the variances of the current ellipse parameters,

$\sigma_{L^{smaj}}^2, \sigma_{L^{smi\eta}}^2, \sigma_{\theta}^2, \sigma_g^2$, under the presumption of zero cross-correlations and by error propagation rules, from the variances of the cosine/sine coefficients, $\sigma_{X^u}^2, \sigma_{Y^u}^2, \sigma_{X^v}^2, \sigma_{Y^v}^2$ (PBL02). The latter are known for the reference constituents, as described above for the white or colored case, but need to be computed for the inference constituents. From (35) and (13), in the case of the l_k th constituent inferred from the k th reference constituent,

$$\begin{aligned}
\hat{X}_{l_k}^u &= \text{Re}(\hat{a}_{l_k}^+ + \hat{a}_{l_k}^-) = \text{Re}(R_{l_k}^+ \hat{a}_k^+) + \text{Re}(R_{l_k}^- \hat{a}_k^-) \\
\hat{Y}_{l_k}^u &= -\text{Im}(\hat{a}_{l_k}^+ - \hat{a}_{l_k}^-) = -\text{Im}(R_{l_k}^+ \hat{a}_k^+) + \text{Im}(R_{l_k}^- \hat{a}_k^-) \\
\hat{X}_{l_k}^v &= \text{Im}(\hat{a}_{l_k}^+ + \hat{a}_{l_k}^-) = \text{Im}(R_{l_k}^+ \hat{a}_k^+) + \text{Im}(R_{l_k}^- \hat{a}_k^-) \\
\hat{Y}_{l_k}^v &= \text{Re}(\hat{a}_{l_k}^+ - \hat{a}_{l_k}^-) = \text{Re}(R_{l_k}^+ \hat{a}_k^+) - \text{Re}(R_{l_k}^- \hat{a}_k^-)
\end{aligned} \tag{77}$$

where the double-hat variables represent inferred constituents and the hat variables represent reference constituents, as in Section II.A. The properties of complex numbers z_1 and z_2 , that $\text{Re}(z_1 z_2) = \text{Re}(z_1)\text{Re}(z_2) - \text{Im}(z_1)\text{Im}(z_2)$ and

$\text{Im}(z_1 z_2) = \text{Im}(z_1)\text{Re}(z_2) + \text{Re}(z_1)\text{Im}(z_2)$, mean these expressions can be written

$$\begin{aligned}\hat{X}_{l_k}^u &= \text{Re}(R_{l_k}^+) \text{Re}(\hat{a}_k^+) - \text{Im}(R_{l_k}^+) \text{Im}(\hat{a}_k^+) + \text{Re}(R_{l_k}^-) \text{Re}(\hat{a}_k^-) - \text{Im}(R_{l_k}^-) \text{Im}(\hat{a}_k^-) \\ \hat{Y}_{l_k}^u &= -\text{Im}(R_{l_k}^+) \text{Re}(\hat{a}_k^+) - \text{Re}(R_{l_k}^+) \text{Im}(\hat{a}_k^+) + \text{Im}(R_{l_k}^-) \text{Re}(\hat{a}_k^-) + \text{Re}(R_{l_k}^-) \text{Im}(\hat{a}_k^-) \\ \hat{X}_{l_k}^v &= \text{Im}(R_{l_k}^+) \text{Re}(\hat{a}_k^+) + \text{Re}(R_{l_k}^+) \text{Im}(\hat{a}_k^+) + \text{Im}(R_{l_k}^-) \text{Re}(\hat{a}_k^-) + \text{Re}(R_{l_k}^-) \text{Im}(\hat{a}_k^-) \\ \hat{Y}_{l_k}^v &= \text{Re}(R_{l_k}^+) \text{Re}(\hat{a}_k^+) - \text{Im}(R_{l_k}^+) \text{Im}(\hat{a}_k^+) - \text{Re}(R_{l_k}^-) \text{Re}(\hat{a}_k^-) + \text{Im}(R_{l_k}^-) \text{Im}(\hat{a}_k^-).\end{aligned}\tag{78}$$

By uncertainty propagation rules, and ignoring cross-correlations just as in the linearized case for non-inferred constituents, it follows that

$$\begin{aligned}\sigma_{\hat{X}_{l_k}^u}^2 &= \sigma_{\hat{Y}_{l_k}^v}^2 = [\text{Re}(R_{l_k}^+)]^2 \sigma_{\text{Re}(\hat{a}_k^+)}^2 + [\text{Im}(R_{l_k}^+)]^2 \sigma_{\text{Im}(\hat{a}_k^+)}^2 \\ &\quad + [\text{Re}(R_{l_k}^-)]^2 \sigma_{\text{Re}(\hat{a}_k^-)}^2 + [\text{Im}(R_{l_k}^-)]^2 \sigma_{\text{Im}(\hat{a}_k^-)}^2 \\ \text{and } \sigma_{\hat{Y}_{l_k}^u}^2 &= \sigma_{\hat{X}_{l_k}^v}^2 = [\text{Im}(R_{l_k}^+)]^2 \sigma_{\text{Re}(\hat{a}_k^+)}^2 + [\text{Re}(R_{l_k}^+)]^2 \sigma_{\text{Im}(\hat{a}_k^+)}^2 \\ &\quad + [\text{Im}(R_{l_k}^-)]^2 \sigma_{\text{Re}(\hat{a}_k^-)}^2 + [\text{Re}(R_{l_k}^-)]^2 \sigma_{\text{Im}(\hat{a}_k^-)}^2.\end{aligned}\tag{79}$$

By similar application of uncertainty propagation to the real and imaginary parts of (32),

$$\begin{aligned}\sigma_{\text{Re}(\hat{a}_k^+)}^2 &= \sigma_{\text{Re}(\hat{a}_k^-)}^2 = (\sigma_{\hat{X}_k^u}^2 + \sigma_{\hat{Y}_k^v}^2) / 4 \\ \sigma_{\text{Im}(\hat{a}_k^+)}^2 &= \sigma_{\text{Im}(\hat{a}_k^-)}^2 = (\sigma_{\hat{X}_k^v}^2 + \sigma_{\hat{Y}_k^u}^2) / 4.\end{aligned}\tag{80}$$

It is clear then that the known values of $\sigma_{\hat{X}_k^u}^2, \sigma_{\hat{Y}_k^v}^2, \sigma_{\hat{X}_k^v}^2, \sigma_{\hat{Y}_k^u}^2$, together with the known values of $R_{l_k}^+, R_{l_k}^-$, are sufficient with (79) and (80) to compute the needed variances of the inferred cosine/sine coefficients, and hence by the linearization formulae the variances and confidence intervals of the current ellipse parameters of the inferred constituents.

II.C.6. One-dimensional case

In the case of one-dimensional raw input (real x^{raw} ; $a^+ = a^{*-}$) the above development applies unmodified except that only the first two elements of the Cartesian cosine/sine coefficients (64) are non-zero, only D_{uu} is non-zero in (65) and (66), only (67) and (70) need be considered, and the scaling in (75) is based solely on P_c^{uu} from (73). With the additional assumption that $\text{cov}(X^u, Y^u) = 0$, the white noise floor result in the one-dimensional case corresponds to that of F77 and FH89.

II.D. Constituent selection diagnostics

This section reviews various diagnostics that are useful to determine which constituents are to be included in a model solution. All diagnostics are defined and explained first for the case of uniformly distributed times, then comments regarding the case of irregular times are made. An explanation of the structure of the diagnostic table included by default in the UTide output, which is designed to present the diagnostics

collectively in a concise fashion for convenient inspection and usage, is given in Section III.C.1.d.

Introductory guidance on the constituent selection process can be found in a number of references, including FH89 and Section 5.5 of the textbook by Emery and Thomson (1998). In general constituent selection is an iterative process, in which initial iterations of the analysis include more constituents than are expected to capture significant energy and/or to be resolved from the other included constituents. This initial set of constituents tends to be created using auxiliary information such as results from other, prior analyses based on similar raw input from the same region. Then by inspecting diagnostics of the output, the decision to keep certain constituents in the model or remove them from it, or to infer them in a subsequent solution, can be made and the next iteration carried out.

The decision to keep, omit, or infer (presuming sufficient needed auxiliary information is available) a given constituent rests on two main, related criteria: (1) the extent to which it is *independent* from the other included constituents, and (2) the extent to which it is *significant*, relative to noise and to the other included constituents. By iterative solutions that include different combinations of constituents, one can determine the extent to which omitting a constituent affects the diagnostics for the remaining included constituents. In general, the goal is to remove constituents that are not sufficiently independent and significant, which can be confirmed by verifying that omitting them causes sufficiently small changes to diagnostics of the remaining constituents.

On this basis, diagnostics quantities are put in to three groups here: one group related to confirming independence of constituents, one group related to ascertaining significance of constituents, and one group to characterize reconstructed harmonic fits that superpose some or all of the constituents.

II.D.1. Diagnostics related to constituent independence

II.D.1.a. Conventional Rayleigh criterion (R^R)

In the case of raw input with uniformly distributed times the widely accepted approach to constituent selection is an automated decision tree method, based on the equilibrium tide and the conventional Rayleigh criterion, as developed by F77 and summarized in FH89. The conventional Rayleigh criterion (superscript R) states that two constituents, of frequencies ω_{q_1} and ω_{q_2} , are resolvable by a record with uniformly distributed times if

$$R^R(q_1, q_2) = \left(\frac{LOR_e}{1/|\omega_{q_2} - \omega_{q_1}|} \right) / R_{\min} \geq 1, \quad (81)$$

where R_{\min} is a minimum threshold taken to be 1 in most cases. For $R_{\min} = 1$, the criterion is equivalent to requiring that the record length (numerator) is sufficiently long that the two frequencies are resolved from each other with respect to traditional spectral estimation. The F77 decision tree essentially compares all constituent pairs and omits any

constituent that does not meet the conventional Rayleigh criterion, relative to a constituent that has larger equilibrium tide amplitude, for a user-specified R_{\min} value. The decision tree can be applied for any choice of R_{\min} , for example a value greater than one in order to conservatively omit a larger number of constituents (as one example, this could be appropriate in the case that the condition number of the basis function matrix is high, as explained below). The decision tree is a standard option in UTide, as it is for `t_tide` (PBL02).

II.D.1.b. Noise-modified Rayleigh criterion (R^{NM})

It is recognized that the conventional Rayleigh criterion (81) is incomplete, because it does not take in to account the fact that noise in a record affects its ability to resolve constituents from each other. As a consequence, in the case of a strongly tidal record, the conventional Rayleigh criterion is overly conservative, rejecting constituents that may be well resolved from each other. For these reasons, Munk and Hasselmann (1964) suggested a modified Rayleigh criterion such that the record duration required to resolve a pair of frequencies was scaled by the square root of the SNR. Munk and Hasselmann did not provide a particular expression for SNR, nor has one been commonly adopted in the literature, so it is suggested here to use the constituent-specific noise-modified (superscript NM) Rayleigh criterion for constituent q_1 relative to constituent q_2 , defined as

$$R^{NM}(q_1, q_2) = \left(R^R(q_1, q_2) \sqrt{(SNR_{q_1} + SNR_{q_2})/2} \right) \geq 1. \quad (82)$$

The denominator is the average of the SNRs of the two constituents, where SNR for an individual constituent is

$$SNR_q = \frac{L^{smaj^2} + L^{smi\eta^2}}{\sigma_{L^{smaj}}^2 + \sigma_{L^{smi\eta}}^2}, \quad (83)$$

in which, for convenience, the q subscripts are dropped on the right hand side. (In the one-dimensional case this expression for SNR is the same as used by PBL02, but in the two-dimensional case it is slightly more general due to inclusion of the terms related to the minor axis.)

In effect, criterion (82) states that the constituents are resolved even for record lengths that are not long enough for R^R to be greater than 1, as long as the SNR factor in (82) is greater than 1. Conversely, if the SNR factor is less than 1, the record length must be longer than for the conventional Rayleigh criterion, since R^R is then required to be greater than 1. It should be borne in mind that even the noise-modified criterion is an incomplete metric, since in the limit of no noise it incorrectly suggests all constituents will be resolvable. Nonetheless it can provide useful information, in combination with the conventional Rayleigh criterion, particularly for analysis of records in which the non-tidal signal is comparable to or larger than the tidal signal.

II.D.1.c. Condition number (K) relative to SNR of entire model (SNR_{allc})

FH89 pointed out that by matrix theory there is an upper bound on the fractional error of the norm of the model parameters,

$$\frac{\|m' - m\|}{\|m\|} \leq K \frac{\|x^{raw'} - x^{raw}\|}{\|x^{raw}\|}, \quad (84)$$

where $\|\cdot\|$ is a suitable norm, the primed variables indicate a case including random noise, the unprimed variables indicate a case with no random noise, and K is the condition number (ratio of largest and smallest singular values) of the matrix B . A reasonable interpretation of the fraction on the RHS of (84), for L^2 norm, is the inverse of the SNR for the entire model including contributions from all directly modeled constituents,

$$SNR_{allc} = \frac{m^H B^H W B m}{\sigma_{MSM}^2}. \quad (85)$$

It follows that

$$SNR_{allc} / K > 1 \quad (86)$$

is a criterion that, when met, means the upper bound of the fractional uncertainty in the norm of the model parameters is of order one or less. Comparison of the condition number and SNR_{allc} can therefore in some cases be a useful diagnostic of whether or not the constituents are, collectively, well resolved by the model; in a model that includes constituents with small differences in frequency, the condition number will be higher, so in order to resolve the constituents well by this criterion, SNR_{allc} must be high enough to exceed the condition number.

By this line of reasoning an additional criterion for whether the constituents are independent from each other, collectively, could be the requirement that SNR_{allc} is at least as high as K . However, in practice there is a limit to the usefulness of this approach because (a) (84) gives an upper bound only and hence the model parameter uncertainties may be sufficiently small even when SNR_{allc} is less than K ; and, of somewhat less importance, (b) (86) is based on the L^2 norm but the hybrid L^1 / L^2 norm applies in the case of the IRLS solution. Furthermore, it is clear that because both SNR_{allc} and K are defined using the entire model, as opposed to a specific constituent pair, comparing them can provide guidance about the model as a whole (including its mean and, if included, trend), but does not provide information relevant to any specific constituent or pair of constituents.

II.D.1.d. Maximum correlation (Corr_{\max}) between model parameters

A related means by which to gauge the extent to which a pair of constituents is independent is to use the cross-covariances among their model parameters. High cross-covariance between model parameters of a pair of constituents indicates they that are less independent, so the less energetic of the pair should be considered for removal from the model, or for inclusion by inference if possible. Here, a new diagnostic for cross-covariances is presented and expressed using elements of the confidence interval development of Section II.C.

The “maximum correlation” diagnostic for constituents with indices c_1 and c_2 is the maximum magnitude of the 16 correlation coefficients between the elements of their Cartesian model parameter vectors $m_{c_1}^C = [X_{c_1}^u \ Y_{c_1}^u \ X_{c_1}^v \ Y_{c_1}^v]$ and

$m_{c_2}^C = [X_{c_2}^u \ Y_{c_2}^u \ X_{c_2}^v \ Y_{c_2}^v]$ (64). It can be written

$$\text{corr}_{\max}(c_1, c_2) = \max\left[|\text{corr}(X_{c_1}^u, X_{c_2}^u)|, |\text{corr}(X_{c_1}^u, Y_{c_2}^u)|, \dots, |\text{corr}(Y_{c_1}^v, Y_{c_2}^v)|\right], \quad (87)$$

where the correlations are defined in the standard way, for example,

$$\text{corr}(X_{c_1}^u, X_{c_2}^u) = \frac{\text{cov}(X_{c_1}^u, X_{c_2}^u)}{\sigma_{X_{c_1}^u} \sigma_{X_{c_2}^u}}. \quad (88)$$

The standard deviations in the denominator of (88) are calculated using the diagonal elements of $\text{var cov}^{\text{white}}(m_c^C)$, expressions for which are given in (70)-(72). The covariances in the numerator of (88) are the elements of (65) when generalized to a constituent pair,

$$\text{cov}^{\text{white}}(m_{c_1}^C, m_{c_2}^C) = \begin{bmatrix} D_{uu}^{c_1 c_2} & D_{uv}^{c_1 c_2} \\ D_{vu}^{c_1 c_2} & D_{vv}^{c_1 c_2} \end{bmatrix}, \quad (89)$$

where the 2×2 D submatrices are

$$\begin{aligned} D_{uu}^{c_1 c_2} &= \begin{bmatrix} \text{cov}(X_{c_1}^u, X_{c_2}^u) & \text{cov}(X_{c_1}^u, Y_{c_2}^u) \\ \text{cov}(Y_{c_1}^u, X_{c_2}^u) & \text{cov}(Y_{c_1}^u, Y_{c_2}^u) \end{bmatrix} & D_{uv}^{c_1 c_2} &= \begin{bmatrix} \text{cov}(X_{c_1}^u, X_{c_2}^v) & \text{cov}(X_{c_1}^u, Y_{c_2}^v) \\ \text{cov}(Y_{c_1}^u, X_{c_2}^v) & \text{cov}(Y_{c_1}^u, Y_{c_2}^v) \end{bmatrix} \\ D_{vu}^{c_1 c_2} &= \begin{bmatrix} \text{cov}(X_{c_1}^v, X_{c_2}^u) & \text{cov}(X_{c_1}^v, Y_{c_2}^u) \\ \text{cov}(Y_{c_1}^v, X_{c_2}^u) & \text{cov}(Y_{c_1}^v, Y_{c_2}^u) \end{bmatrix} & D_{vv}^{c_1 c_2} &= \begin{bmatrix} \text{cov}(X_{c_1}^v, X_{c_2}^v) & \text{cov}(X_{c_1}^v, Y_{c_2}^v) \\ \text{cov}(Y_{c_1}^v, X_{c_2}^v) & \text{cov}(Y_{c_1}^v, Y_{c_2}^v) \end{bmatrix}. \end{aligned} \quad (90)$$

The expressions in (57) are generalized to their two-constituent forms,

$$\begin{aligned} G^{c_1 c_2} &= \begin{bmatrix} G_{c_1, c_2}^{\text{all}} & G_{c_1, c_2 + n_c}^{\text{all}} \\ G_{c_1 + n_c, c_2}^{\text{all}} & G_{c_1 + n_c, c_2 + n_c}^{\text{all}} \end{bmatrix} \\ H^{c_1 c_2} &= \begin{bmatrix} H_{c_1, c_2}^{\text{all}} & H_{c_1, c_2 + n_c}^{\text{all}} \\ H_{c_1 + n_c, c_2}^{\text{all}} & H_{c_1 + n_c, c_2 + n_c}^{\text{all}} \end{bmatrix}. \end{aligned} \quad (91)$$

It follows from the definitions (34) and (35) with relations of the same form as (60)-(62), and omitting the $c_1 c_2$ superscripts from G and H for clarity, that the terms of $D_{uu}^{c_1 c_2}$ are

$$\begin{aligned} \text{cov}(X_{c_1}^u, X_{c_2}^u) &= [\text{Re}(G_{11}) + \text{Re}(G_{12}) + \text{Re}(G_{21}) + \text{Re}(G_{22})]/2 \\ \text{cov}(X_{c_1}^u, Y_{c_2}^u) &= [\text{Im}(H_{11}) - \text{Im}(H_{12}) + \text{Im}(H_{21}) - \text{Im}(H_{22})]/2 \\ \text{cov}(Y_{c_1}^u, X_{c_2}^u) &= [-\text{Im}(G_{11}) - \text{Im}(G_{12}) + \text{Im}(G_{21}) + \text{Im}(G_{22})]/2 \\ \text{cov}(Y_{c_1}^u, Y_{c_2}^u) &= [\text{Re}(H_{11}) - \text{Re}(H_{12}) - \text{Re}(H_{21}) + \text{Re}(H_{22})]/2; \end{aligned} \quad (92)$$

the terms of $D_{uv}^{c_1 c_2}$ are

$$\begin{aligned} \text{cov}(X_{c_1}^u, X_{c_2}^v) &= [-\text{Im}(H_{11}) - \text{Im}(H_{12}) - \text{Im}(H_{21}) - \text{Im}(H_{22})]/2 \\ \text{cov}(X_{c_1}^u, Y_{c_2}^v) &= [\text{Re}(G_{11}) - \text{Re}(G_{12}) + \text{Re}(G_{21}) - \text{Re}(G_{22})]/2 \\ \text{cov}(Y_{c_1}^u, X_{c_2}^v) &= [-\text{Re}(H_{11}) - \text{Re}(H_{12}) + \text{Re}(H_{21}) + \text{Re}(H_{22})]/2 \end{aligned} \quad (93)$$

$$\text{cov}(Y_{c_1}^u, Y_{c_2}^v) = [-\text{Im}(G_{11}) + \text{Im}(G_{12}) + \text{Im}(G_{21}) - \text{Im}(G_{22})]/2;$$

the terms of $D_{vu}^{c_1 c_2}$ are

$$\begin{aligned} \text{cov}(X_{c_1}^v, X_{c_2}^u) &= [\text{Im}(G_{11}) + \text{Im}(G_{12}) + \text{Im}(G_{21}) + \text{Im}(G_{22})]/2 \\ \text{cov}(X_{c_1}^v, Y_{c_2}^u) &= [-\text{Re}(H_{11}) + \text{Re}(H_{12}) - \text{Re}(H_{21}) + \text{Re}(H_{22})]/2 \\ \text{cov}(Y_{c_1}^u, X_{c_2}^v) &= [\text{Re}(G_{11}) + \text{Re}(G_{12}) - \text{Re}(G_{21}) - \text{Re}(G_{22})]/2 \\ \text{cov}(Y_{c_1}^u, Y_{c_2}^v) &= [\text{Im}(H_{11}) - \text{Im}(H_{12}) - \text{Im}(H_{21}) + \text{Im}(H_{22})]/2; \end{aligned} \tag{94}$$

and the terms of $D_{vv}^{c_1 c_2}$ are

$$\begin{aligned} \text{cov}(X_{c_1}^v, X_{c_2}^v) &= [\text{Re}(H_{11}) + \text{Re}(H_{12}) + \text{Re}(H_{21}) + \text{Re}(H_{22})]/2 \\ \text{cov}(X_{c_1}^v, Y_{c_2}^v) &= [\text{Im}(G_{11}) - \text{Im}(G_{12}) + \text{Im}(G_{21}) - \text{Im}(G_{22})]/2 \\ \text{cov}(Y_{c_1}^v, X_{c_2}^v) &= [-\text{Im}(H_{11}) - \text{Im}(H_{12}) + \text{Im}(H_{21}) + \text{Im}(H_{22})]/2 \\ \text{cov}(Y_{c_1}^v, Y_{c_2}^v) &= [\text{Re}(G_{11}) - \text{Re}(G_{12}) - \text{Re}(G_{21}) + \text{Re}(G_{22})]/2. \end{aligned} \tag{95}$$

(The case $c_1 = c_2$, not considered in this section, would yield the three groups of expressions (70)-(72).)

In the case of one-dimensional raw input, the Cartesian model parameter vectors $m_{c_1}^C, m_{c_2}^C$ have only two elements each, so corr_{\max} is the maximum among 4 (not 16) correlations. The above expressions are unchanged except for the fact that only the D_{uu} portions (upper left quadrants) of the matrices are nonzero and need be considered.

The corr_{\max} diagnostic is analogous to the correlation diagnostic for one-dimensional raw input based on the singular value decomposition, developed by Cherniawsky et al (2001) and used in FCB09. In the analysis of FCB09, correlations up to about 0.2 were considered acceptable. However, experimentation is required in order to determine acceptable levels of corr_{\max} for each individual analysis.

In the UTide constituent selection diagnostic table (described in Section III.C.1.d below), corr_{\max} values are computed only for adjacent pairs of constituents in the model, that is, those pairs that have frequencies nearer to each other than to any other constituents (excluding inferred constituents). While this is not as complete as computing and examining corr_{\max} values for every pair of constituents, it is a practical approach based on the expectation that constituent pairs with adjacent frequencies are typically more likely to lack independence.

II.D.2. Diagnostics related to constituent significance

II.D.2.a. Signal to noise ratio (SNR)

Constituents $q = 1 \dots n_{\text{allc}}$ for a given solution are considered to be significant with respect to the noise in the raw input if their SNR_q (83) obeys

$$SNR_q \geq SNR_{\min}, \quad (96)$$

where SNR_{\min} is a minimum threshold value. Common practice takes SNR_{\min} to be 1 or 2, but in certain situations other values may be appropriate. For example, a higher value might be used in order to conservatively neglect marginally significant constituents if the estimates of the standard deviations of the model parameters, on which the SNR values are founded (83), are thought to be biased low.

II.D.2.b. Percent energy (PE)

The model solution x^{mod} (42) is a reconstructed harmonic fit that superposes all the constituents. Independently of their significance with respect to the SNR threshold, the relative importance of a constituent can be gauged by the percent energy (Codiga and Rear 2004) it contributes to the model solution. For constituent q , the percent energy is

$$PE_q = 100 \frac{E_q}{\sum_{q=1}^{n_{\text{allc}}} E_q}, \quad (97)$$

where

$$E_q = (L_{\text{smaj}}^2 + L_{\text{smi}\eta}^2), \quad (98)$$

the q subscripts have been dropped on the right hand side of (98), and the summed E_q values equal 100. In the two-dimensional case of horizontal velocity components, E_q is proportional to the kinetic energy; in the one-dimensional case of sea level it is a gauge of potential energy. It is useful to rank the constituents by their percent energy so that the importance of the constituents in an amplitude-weighted sense is clear. This ranking usually parallels the SNR ranking but it can be less sensitive to the confidence interval calculation method, and in certain cases provides important complementary information to SNR.

II.D.3. Diagnostics characterizing reconstructed fits (PTV_{all} , PTV_{snrc})

A diagnostic of the model solution is its percent tidal variance,

$$PTV_{\text{allc}} = 100 \frac{TV_{\text{allc}}}{TV_{\text{raw}}} = 100 \frac{|x^{\text{mod}} - \bar{x} \cdot I(n_t, 1) - \dot{x} \cdot t|^2}{|x^{\text{raw}} - \bar{x} \cdot I(n_t, 1) - \dot{x} \cdot t|^2}, \quad (99)$$

where TV_{allc} and TV_{raw} (each in units of squared raw input units) are the tidal variance, after removal of the mean and trend, of the (all-constituent) model solution and the raw input, respectively.

Reconstructed fits other than the model solution can be calculated based on inclusion of a subset of the constituents. Denote by q_{incl} the subset of $n_{\text{incl}} \leq n_{\text{allc}}$ constituents (among non-reference, reference, and inferred constituents) that are chosen, based on some criteria, to be included in a reconstructed fit. Following model equation (2), the reconstructed fit computed using that subset of constituents is

$$\sum_{q_{\text{incl}}=1}^{n_{\text{incl}}} (E_{i_{q_{\text{incl}}}} a_{q_{\text{incl}}}^+ + E_{i_{q_{\text{incl}}}^*} a_{q_{\text{incl}}}^-) + \bar{x} + \dot{x} \cdot (t_i - t_{\text{ref}}). \quad (100)$$

Substituting this in (99) for x^{mod} yields the percent tidal variance of the reconstructed fit,

$$PTV_{incl} = 100 \frac{TV_{incl}}{TV_{raw}} = 100 \frac{|\sum_{q_{incl}=1}^{n_{incl}} (E_{iq_{incl}} a_{q_{incl}}^+ + E_{iq_{incl}}^* a_{q_{incl}}^-)|^2}{TV_{raw}}, \quad (101)$$

based on the ratio of the tidal variance of the reconstructed fit to that of the raw input. A straightforward example is the reconstructed fit using only the constituents that meet the SNR criterion (“snrc”) for significance (96), denoted by indices q_{snrc} . The percent tidal variance of the corresponding reconstructed fit is

$$PTV_{snrc} = 100 \frac{TV_{snrc}}{TV_{raw}} = 100 \frac{|\sum_{q_{snrc}=1}^{n_{snrc}} (E_{iq_{snrc}} a_{q_{snrc}}^+ + E_{iq_{snrc}}^* a_{q_{snrc}}^-)|^2}{TV_{raw}}. \quad (102)$$

II.D.4. Considerations for irregularly distributed times

When the times of the raw input are distributed irregularly, some of the underlying assumptions behind the above diagnostics are violated, making proper constituent selection a major challenge. Irregular times can be viewed loosely as if certain parts of the record have more highly concentrated temporal sampling and could resolve a higher number of constituents, whereas the opposite is true for other parts of the record. It follows that to be conservative one should select constituents based on the limitations of the portions of the record where temporal sampling is most sparse. However, there are no guidelines or accepted practices for how to carry out this goal on the basis of knowledge about the distribution of irregular times.

Even though strictly speaking the underlying assumptions are violated, for the case of irregularly distributed times it is nonetheless straightforward to follow the above approach, without modification, for both the implementation of the automated decision tree and the computation of all of the above diagnostics. As a result, although it is certainly not rigorously justified by the underlying statistics, it is straightforward to calculate and inspect all the same diagnostics in the case of irregularly distributed times as are used for regularly distributed times. This approach is at least a starting point, in the absence of suitable diagnostics that are well-defined in terms of the characteristics of the distribution of the irregular times.

It will of course be most justified to make use of diagnostics so calculated in cases when the irregularity in the distribution of the times is modest. An example of modest irregularity is a distribution of times that deviates from an equispaced time series only by small random deviations, as opposed to including numerous gaps that are long, in the sense that their duration spans at least several samples in a comparable regular time series.

When the times are irregular a crude but practical approach to being conservative, in the sense of omitting constituents that might not be resolved from each other, is to use the same diagnostics but judge them in relation to different threshold values. For example, if the Rayleigh criteria ((81),(82)) are used with $R_{\min}=1$ for a uniformly

sampled record, then for an irregularly sampled record an R_{\min} value higher than 1 can be used, in order to be more stringent in omitting constituents.

There is no accepted practice for determining the multiple by which to increase R_{\min} based on the arbitrary, but known, irregular distribution of the times. As a starting point, consider the comparison between a record with uniform sampling of time difference Δt between samples, and a modestly irregular record with time differences between samples that are variable but Gaussian with mean $\overline{\Delta t}$ and standard deviation $\sigma_{\Delta t}$ less than the mean. A reasonable choice for the appropriate ratio by which to increase R_{\min} is

$$R_{\min}^{irregular} / R_{\min} = \overline{\Delta t} / (\overline{\Delta t} - \alpha \sigma_{\Delta t}) \quad (103)$$

where α is a constant, nominally 1, that can be increased (as long as the denominator remains positive) in order to implement more conservative constituent rejection. This approach is equivalent to decreasing the numerator in the Rayleigh criterion (81), from the length of record or $(n_t - 1)\Delta t$ in the uniformly distributed times case, to

$(n_t - 1)(\overline{\Delta t} - \alpha \sigma_{\Delta t})$. Most real-world raw inputs with irregular temporal sampling are likely to have non-Gaussian distributions of the time differences, in which case some improved robustness should follow from using the median (instead of the mean) for $\overline{\Delta t}$ and the median-absolute-deviation (instead of the standard deviation) for $\sigma_{\Delta t}$.

However, the deviation of the irregular sampling distribution from Gaussian is commonly very severe, with numerous long gaps, such that in general (103) may not be applicable and it is expected that some empirical experimentation will be necessary to arrive at an acceptable value of $R_{\min}^{irregular}$.

III. The UTide Matlab functions

UTide consists of a pair of Matlab functions designed to be easy to understand and implement: *ut_solv()* to carry out the analysis, the results of which are passed to *ut_reconstr()* for reconstructing a hind-cast or forecast/prediction, or “fit”, as needed.

III.A. Obtaining and using UTide

There are three UTide files: *ut_solv.m*, *ut_reconstr.m*, and *ut_constants.mat*. The current version can be downloaded in a compressed bundle, together with this report, at <ftp://www.po.gso.uri.edu/pub/downloads/codiga/utide/UTideCurrentVersion.zip>; the version history, and the bundle file for older versions, will be available in that same folder. Uncompress the zipfile contents to a single folder/directory and make sure it is on the Matlab path. No other formal installation is needed.

UTide makes use of functions from both the Signal Processing Toolbox (*pwelch()*, *cpsd()*, *hanning()*) and the Statistics Toolbox (*robustfit()*, *mvnrnd()*). Thus if either of these toolboxes are not available, executing UTide in its default configuration will result in errors. If the Signal Processing Toolbox is not available, the colored method for confidence intervals will not be possible, so to avoid such errors UTide must be run using the ‘White’ option flag (explained below). Similarly, if the Statistics Toolbox is not available, the IRLS solution method and the Monte Carlo confidence interval approach will not be possible, so to avoid such errors UTide must be run using both the ‘OLS’ and ‘LinCI’ option flags (explained below).

As noted by LJ09, calls to *robustfit()* to carry out the IRLS solution commonly result in a warning or error related to reaching the iteration limit. This can be remedied by editing the line *iterlim = 50*; in the file *statrobustfit.m*, found in the *MATLABROOT\toolbox\stats\private* folder, to replace 50 by a sufficiently larger maximum number of iterations, for example 500. Other than permitting a higher number of iterations, this has no effect on the functionality of *robustfit()* or any other aspects of Matlab. In general, increasing the tuning parameter can be a remedy, in cases for which the iteration limit is reached. When the iteration limit is reached, the results for current ellipse parameters (amplitude/phase parameters in the 1D case), means, slopes (if trend included), and confidence intervals are set to NaN.

III.B. Quick start suggestions

The following steps outline the most efficient way to get started quickly with an initial computation (for analysis of a single record) using the default settings of UTide. First, read the opening portion of Section III.C, which briefly summarizes the syntax. Next, read section III.C.1.a, where the formats of the input parameters to *ut_solv()* are explained, and manipulate the raw values you wish to analyze in order to create *t_raw*, *u_raw*, *v_raw*, *lat*, and *cnsit* in the needed formats (for the latter, a typical initial choice is ‘auto’, which invokes the F77 decision tree to carry out constituent selection). Then, pass them in to *ut_solv()* to generate the structure *coef*. If the computational burden is too much for the available resources, try again but also pass in the ‘OLS’, ‘white’, and ‘LinCI’ flags, after the *cnsit* input, as explained in Section III.F. By default the

diagnostics table *coef.diagn.table* (explained in Section III.C.1.d) will be displayed at runtime, and can be inspected in order to iterate towards a more refined constituent selection for a subsequent call to *ut_solv()*. Read Section III.C.1.b to understand the contents and formats of the field in *coef*, which include all analysis results and are available for manipulation in further custom analysis or plotting (of the constituent statistics, coefficients and confidence intervals, current ellipses, etc). Finally, if reconstructing a hind-cast or forecast/prediction fit (superposed harmonics) using the resulting coefficients is desired, construct a vector *t_fit* (in the same format as *t_raw*) of times and pass them in to *ut_reconstr()*, with specification of the subset of constituents to include (default includes constituents with $SNR \geq 2$), as explained in Section III.C.2. To treat a group of records, see Section III.D.

III.C. Functionality and syntax and for a single record

Syntax for two-dimensional raw input, such as velocities, is

```
coef = ut_solv ( t_raw, u_raw, v_raw, lat, cnsit , {options} );
[ u_fit, v_fit ] = ut_reconstr ( t_fit, coef , {options} );
```

and syntax for one-dimensional raw input, such as sea level, is

```
coef = ut_solv ( t_raw, sl_raw, [], lat, cnsit , {options} );
[ sl_fit, ~ ] = ut_reconstr ( t_fit, coef , {options} );
```

for which a brief overview of the various variables/parameters is as follows:

- *coef* is the output structure generated by *ut_solv()* and accepted by *ut_reconstr()*;
- the raw input records have times *t_raw*, current components *u_raw/v_raw* or sea level values *sl_raw*, and latitude *lat*;
- *cnsit* specifies the constituents to be included in the model;
- *t_fit* contains the arbitrary times at which the reconstructed output (*u_fit/v_fit* or *sl_fit*) is computed; and
- {*options*} represents optional inputs, as explained in more detail below.

More detailed explanations of variables that must be passed in to *ut_solv()* are given in Section III.C.1.a; a detailed explanation of the contents of the output from *ut_solv()*, the structure *coef*, is given in Section III.C.1.b; explanations of the default configuration and option flags for *ut_solv()* are given in Section III.C.1.d. Information regarding the inputs and outputs of *ut_reconstr()*, and its default configuration and option flags, is given in Section III.C.2.

III.C.1. Solving for coefficients with *ut_solv()*

III.C.1.a. Input parameter descriptions

The times are in a real-valued column vector *t_raw* that

- contains Matlab “datenum” values for the sampled times in coordinated universal time UTC (Greenwich mean time, GMT), with units of days. For the unfamiliar, the definition and characteristics of “datenum” values are explained in the Matlab documentation; see the help descriptions for functions *datenum()*, *datevec()*, and

- datestr()*. For example, to generate an N-day time vector with hourly resolution, $1:(1/24):N$ will not suffice; instead use, e.g., for a start time of 8:15AM on Nov. 1, 2001, $datenum(2001,11,1,8,15,0)+(1:(1/24):N)$;
- can have NaN values, but they will be removed during analysis, along with the corresponding (NaN or non-NaN) associated *u_raw/v_raw* or *sl_raw* values;
 - contains values that can be either regularly/uniformly distributed (“equispaced”) or irregularly distributed;
 - is considered equispaced if (after NaNs are removed) the Matlab expression $\text{var}(\text{unique}(\text{diff}(t_raw))) < \text{eps}$ is true [where *var()*, *unique()*, *diff()*, and *eps()* are built-in functions], in which case FFT methods (*pwelch()*, *cpsd()*) are used for the periodogram of the residual for colored confidence intervals; and
 - is considered irregularly distributed if (after NaNs are removed) $\text{var}(\text{unique}(\text{diff}(t_raw))) \geq \text{eps}$, in which case, the Lomb-Scargle periodogram of the residual is used for colored confidence intervals.

Raw input vectors *u_raw* and *v_raw*, or *sl_raw*

- are real-valued column vectors of the same size as *t_raw*; and
- are permitted to include NaNs, in which case if the times are equispaced (see above) the NaNs will be filled by linear interpolation prior to FFT spectral analysis.

The scalar input *lat* is

- the latitude (in decimal degrees, positive North and negative South);
- a required input, because nodal/satellite corrections are implemented by default; and
- not used if nodal/satellite corrections are omitted (‘NodsatNone’ option).

The input *constit* determines the constituents included in the model and is one of the following:

- the string ‘auto’ (not case-sensitive)
 - the F77 automated decision tree (with default value R_{\min} (81) of 1, unless a different value is specified with the ‘Rmin’ option) is implemented;
 - if inference/reference constituents are specified together with this option, the inference/reference constituents are included in the model whether or not they are selected by the decision tree; a constituent selected by the decision tree will be removed from the non-reference group of constituents if it is designated as a reference constituent or an inferred constituent.
- a cell array of 4-character strings (not case-sensitive)
 - each string contains the name of a non-reference constituent to be included, including trailing blanks if needed to fill out 4 characters;
 - the constituents available are those (including shallow-water constituents) in the *const.name* variable in the “ut_constants.mat” file, for example ‘M2’, ‘MSF’, etc.

III.C.1.b. Output structure *coef*

The main output of the call to *ut_solv()* is a single structure, *coef*. It consists of various scalar, vector, and string array fields, which are described here.

The sizes of the cell array and numeric fields and subfields of *coef* are either 1x1 or n_{allc} x1. For fields with size n_{allc} x1, there is one value for each of the constituents, and they are ordered such that the results for a given constituent are in the same element of each such field. By default the elements are ordered by decreasing percent energy PE_q (97). Using the ‘OrderCnstit’ option flag they can instead be ordered by decreasing SNR, increasing frequency, or in a user-specified sequence (the last option is possible as long as the automatic decision tree is not used for constituent selection, that is, *cnsit* is not ‘auto’).

There are three main groups of results in *coef*: primary results, auxiliary results, and diagnostic results. The primary results are the n_{allc} x1 cell array

- *coef.name*, an array of 4-character constituent names, the following real-valued n_{allc} x1 vectors,
 - in the two-dimensional case ((9),(76)),
 - *coef.Lsmaj*, the current ellipse major axis length (units of *u_raw/v_raw*)
 - *coef.Lsmaj_ci*, the 95% confidence interval for *coef.Lsmaj*,
 - *coef.Lsmin*, the current ellipse minor axis length,
 - *coef.Lsmin_ci*, the 95% confidence interval for *coef.Lsmin*,
 - *coef.theta*, the current ellipse orientation angle (degrees)
 - *coef.theta_ci*, the 95% confidence interval for *coef.theta*,
 - *coef.g*, the Greenwich phase lag (degrees) of the vector velocity,
 - *coef.g_ci*, the 95% confidence interval for *coef.g*,
 - in the one-dimensional case ((20),(76))
 - *coef.A*, the amplitude (units of *sl_raw*)
 - *coef.A_ci*, the 95% confidence interval for *coef.A*
 - *coef.g*, the Greenwich phase lag (degrees),
 - *coef.g_ci*, the 95% confidence interval for *coef.g*,

and the following real-valued scalars,

- in the two-dimensional case,
 - *coef.umean* and *coef.vmean*, the mean values \bar{u}, \bar{v} (2) (*u_raw/v_raw* units) for the u/v components,
 - *coef.uslope* and *coef.vslope*, the trend slope \dot{u}, \dot{v} (2) (*u_raw/v_raw* units per day) for the u/v components (omitted if the trend is not included in the model),
- in the one-dimensional case,
 - *coef.mean*, the mean value (*sl_raw* units),
 - *coef.slope*, the trend slope (*sl_raw* units per day), omitted if the trend is not included in the model.

The field *coef.results* is a character array that presents all the above fields in an easy to read format and is displayed during runtime by default. If the IRLS solution method is used and it does not converge, a warning is given in *coef.results*, and the values of the above fields are set to NaN.

The auxiliary results are included as fields of *coef.aux* and consist of

- *coef.aux.rundescr*, a cell string array with a descriptive explanation of the run,
- *coef.aux.opt*, a series of fields containing the option settings and input parameters,

the n_{allc} x1 vectors

- *coef.aux.frq*, the frequencies of the constituents (cycles per hour),
- *coef.aux.lind*, the list indices (used by *ut_reconstr()*) of the constituents as referenced in the file *ut_constants.mat*,

and the scalars

- *coef.aux.lat*, the latitude,
- *coef.aux.reftime*, the reference time (*datenum* UTC/GMT as for *t_raw*).

The diagnostic results are included in *coef.diagn*, which is created unless the ‘NoDiagn’ option is selected, and consists of the following n_{allc} x1 fields, each with the same (default decreasing PE_q (97); otherwise can be specified, using the ‘OrderCnstit’ option, to be decreasing SNR, increasing frequency, or user-specified) element order:

- *coef.diagn.name*, the four-character constituent names, in order of decreasing PE_q (identical to *coef.name*; except when the ‘OrderCnstit’ option is used, in which case it is the same list of constituents but ordered differently),
- *coef.diagn.PE*, the percent energy (97),
- *coef.diagn.SNR*, the signal to noise ratio (83),
- *coef.diagn.lo.name*, the name of the constituent with the nearest lower frequency (NaN if no other constituent has a lower frequency),
- *coef.diagn.lo.RR*, the conventional Rayleigh criterion ratio R^R (81) relative to the constituent with the nearest lower frequency (NaN if no other constituent has a lower frequency),
- *coef.diagn.lo.RNM*, the noise-modified Rayleigh criterion ratio R^{NM} (82) relative to the constituent with the nearest lower frequency (NaN if no other constituent has a lower frequency),
- *coef.diagn.lo.CorMx*, the maximum model parameter correlation $corr_{max}$ (87) relative to the constituent with the nearest lower frequency (NaN if no other constituent has a lower frequency),
- *coef.diagn.hi.name*, *coef.diagn.hi.RR*, *coef.diagn.hi.RNM*, and *coef.diagn.hi.CorMx*, which are the same as the corresponding above four fields, but for the nearest higher frequency,

and the following scalars:

- *coef.diagn.K*, the condition number of the basis function matrix (84),
- *coef.diagn.SNRallc*, the all-constituent signal to noise ratio (85),
- *coef.diagn.TVraw*, the tidal variance (99) of the raw inputs, with units *u_raw/v_raw* units squared (two-dimensional case) or *sl_raw* units squared (one-dimensional case),
- *coef.diagn.TVallc*, the tidal variance of the model solution (all constituents superposed) (99), with units *u_raw/v_raw* units squared (two-dimensional case) or *sl_raw* units squared (one-dimensional case),

- ***coef.diagn.TVsnrc***, the tidal variance of the reconstructed fit using only constituents that meet the SNR criterion (102),
- ***coef.diagn.PTVallc***, the percent tidal variance captured by the (all-constituent) model solution (99), and
- ***coef.diagn.PTVsnrc***, the percent tidal variance of the reconstructed fit using only constituents that meet the SNR criterion (102).

The main diagnostic results, in addition to appearing in the above fields, are summarized in the constituent selection diagnostics table ***coef.diagn.table***. This is a table (described in detail in Section III.1.C.d) that is formatted for easy viewing within Matlab to aid in the constituent selection process. Unlike the above $n_{allc} \times 1$ fields of ***coef.diagn***, for which the ordering can be changed using the ‘OrderCnstit’ option flag, the rows in ***coef.diagn.table*** are always ordered by decreasing PE_q .

Runtime display. All results of a call to ***ut_solv()*** are stored in ***coef***. For convenience, by default when its execution is complete ***ut_solv()*** outputs a three-part runtime display (a key subset of the contents of ***coef***). The runtime display consists of (a) the coefficients and confidence intervals, ***coef.results***; (b) the run description meta-information ***coef.aux.rundescr***; and (c) the constituent selection diagnostics table ***coef.diagn.table***, if it has been generated (unless the ‘NoDiagn’ option has been selected). The runtime display can be omitted entirely, or can include subsets of the three components, as controlled by the ***opt.RunTimeDisp*** option to ***ut_solv()*** as described below.

III.C.1.c. Defaults and options

The default configuration for a call to the function ***ut_solv()***, which is implemented when no ***{options}*** parameters are passed in, is as follows.

- The linear (secular, non-tidal) trend (2) is included in the model .
- No pre-filtering correction is made ($P_q = 1$, in (5)).
- Nodal/satellite corrections with exact times (4) are implemented.
- Greenwich phase lags are computed by use of the astronomical argument with exact times (4).
- The model includes no inferred constituents.
- If ***cnstit*** is ‘auto’ then the automated decision tree constituent selection method is applied with $R_{min} = 1$ (81).
- The solution method is robust IRLS with the Cauchy weight function and tuning parameter 2.385, which is the Matlab default value (TunRdn = 1).
- The Monte Carlo uncertainty propagation method (Section II.C) is used, with 200 realizations (Nrlzn=200), to determine confidence intervals of the current ellipse parameters from those of the model parameters.
- Confidence intervals are computed based on the (colored) spectra computed from the actual residuals (Section II.C.3). If the input times are uniformly distributed, the spectra are computed using FFT methods, otherwise they are computed using the Lomb-Scargle periodogram with frequency oversampling factor 1 (LSFrqOSmp=1).

- Computation of the constituent selection diagnostics table (Section III.C.1.d) is carried out, using $SNR_{\min} = 2$.
- The order of constituents in the output variables and diagnostics table is based on decreasing percent energy PE_q (97).
- All three components of the runtime display are presented.

To change these defaults, the following option flags can be passed in. Option flags must be passed in after the *cnstit* argument. The option flags are not case-sensitive, but they cannot be abbreviated. For some flags, accompanying variables are passed in, as noted in boldface italics. For those options appearing in a list headed by “One of the following”, an error will result if more than one on the list is specified.

- ‘NoTrend’
 - This will omit the linear/secular trend term from the model.
- ‘PreFilt’, *PreFilt*
 - This will implement the correction to account for pre-filtering that was applied to the raw inputs before the analysis; *PreFilt* is a structure that specifies the pre-filter transfer function (not the inverse transfer function, as is input to *t_tide*, PBL02) as
 - *PreFilt.P*, an $n_{frq} \times 1$ vector with real-valued P (4) for the one dimensional case, and complex $P + iP$ for the two-dimensional case, where n_{frq} is an arbitrary number of frequencies by which the filter shape is to be specified
 - *PreFilt.frq*, an $n_{frq} \times 1$ vector of the frequencies (in cycles per hour) of the *PreFilt.P* values, and
 - *PreFilt.rng*, a two-value vector with the range (minimum and maximum) of acceptable P magnitudes, for example [0.01 100]; values outside this range will be set to 1.
- One of the following:
 - ‘NodsatLinT’
 - This will cause nodal/satellite corrections to use linearized times, instead of the default exact formulation.
 - ‘NodsatNone’
 - This will cause nodal/satellite corrections to be omitted, instead of the default exact formulation.
- One of the following:
 - ‘GwchLinT’
 - This will cause the astronomical argument in the Greenwich phase lag calculation to use linearized times, instead of the default exact formulation.
 - ‘GwchNone’
 - This will omit the astronomical argument, such that the reported phase lags are “raw” (not Greenwich-referenced) relative to the reference time t_{ref} .
- ‘Infer’, *Infer*

- This causes a total of n_I inference constituents, and n_R reference constituents ($1 \leq n_R \leq n_I$), to be included in the model (11). If any of the specified reference or inference constituents are in the group of non-reference constituents, as determined by the automatic decision tree or as specified manually by the *cnstit* input, they are removed from the group of non-reference constituents. *Infer* is a structure with elements
 - *Infer.infnam*, a cell-array of n_I 4-character names of the constituents to be inferred,
 - *Infer.refnam*, a cell-array of n_I 4-character names of the corresponding reference constituents, not all of which need be unique from each other (unless ‘InferAprx’ is chosen, see below) because multiple constituents can be inferred from a single reference constituent,
 - *Infer.amprat*, which is
 - for the two-dimensional case, a $2n_I \times 1$ vector of real-valued unitless amplitude ratios, r^+, r^- (12), with r^+ values in the first n_I elements and r^- values in the second n_I elements
 - for the one-dimensional case, an $n_I \times 1$ vector of real-valued unitless amplitude ratios r^n (21),
 - *Infer.phsoff*, which is
 - for the two-dimensional case, a $2n_I \times 1$ array of real-valued phase offsets (in degrees), ζ^+, ζ^- (12), with ζ^+ values in the first n_I elements and ζ^- values in the second n_I elements
 - for the one-dimensional case, an $n_I \times 1$ vector of real-valued phase offsets ζ^n (21).
- ‘InferAprx’
 - This causes the inference calculation to follow the approximate method (Section II.A.4.c). Ignored unless ‘Infer’ also selected. With this option selected, an error will result if the constituents in *infer.refnam* are not unique, i.e. when $n_R < n_I$ (as when inferring multiple constituents from a single reference constituent, not possible for the approximate method).
- ‘Rmin’, *Rmin*
 - This will specify the R_{\min} (81) value (positive) to be used in automated constituent selection. The default is $R_{\min} = 1$. Ignored if *cnstit* is not ‘auto’.
- One of the following:
 - ‘OLS’
 - This will change the solution method to the Matlab “backslash” operator, to implement ordinary least squares analysis instead of the default IRLS.
 - ‘Andrews’, ‘Bisquare’, ‘Fair’, ‘Huber’, ‘Logistic’, ‘Talwar’, OR ‘Welsch’

- This will cause the robust IRLS method to be implemented by the *robustfit()* function using the named weight function (instead of the default Cauchy weight function), and using a tuning parameter that is the Matlab default value for that weight function divided by the tuning factor reduction parameter *TunRdn*.
- ‘TunRdn’, *TunRdn*
 - To reduce the IRLS tuning parameter relative to the Matlab default value, for the given weight function (default Cauchy; otherwise specified by an option input), by the tuning parameter reduction factor *TunRdn* (the tuning parameter used is the default tuning parameter divided by *TunRdn*.) The default is *TunRdn* = 1. Ignored if using ‘OLS’.
- ‘LinCI’
 - This causes the confidence intervals on the current ellipse parameters to be computed from the uncertainties in the model parameters by the linearized method, instead of the Monte Carlo method.
- ‘White’
 - This causes the white noise floor assumption to be implemented in the confidence interval calculation such that spectra of the residual are presumed white instead of calculated from the actual colored residual.
- ‘Nrlzn’, *Nrlzn*
 - This will cause the Monte Carlo calculations to use *Nrlzn* realizations instead of the default, which is 200. Ignored if ‘LinCI’ flag is passed in.
- ‘LSFrqOSmp’, *LSFrqOSmp*
 - This will cause the Lomb-Scargle periodogram calculation to use frequency oversampling factor of *LSFrqOSmp* instead of the default, which is 1. If *LSFrqOSmp* is not an integer it is rounded. Ignored if raw input times are uniformly distributed or if the ‘White’ flag is passed in.
- ‘DiagnMinSNR’, *MinSNR*
 - This will specify the SNR_{\min} value used in (i) calculation of TV_{snrc} and PTV_{snrc} , and (ii) in the constituents included in the reconstructed fits of the diagnostic figures. Default value is 2.
- One of the following:
 - ‘NoDiagn’
 - Skip both the summary diagnostics table and the diagnostic figures.
 - ‘DiagnPlots’
 - Generate the diagnostic figures (described in Section III.C.1.d) in addition to the diagnostics table.
- ‘OrderCnstt’, *CnsttSeq*
 - This will override the default PE_q -ranked ordering by which the constituent-based parameters in the *coef* output structure (explained below, e.g., *coef.name*, *coef.g*, etc; *not* the diagnostics in *coef.diagn*) are listed. The parameter *CnsttSeq* is one of:
 - ‘snr’, to order by decreasing SNR_q , OR
 - ‘frq’, to order by increasing frequency, OR

- (allowed only in the case that *cnstit* is not ‘auto’) a cell array of 4-character strings that, if it differs from *cnstit*, only differs in the order of its rows.
 - The row order of the fields in *coef.diagn*, and the rows in the summary diagnostics table, are always by decreasing PE_q and are not affected by the ‘OrderCnstit’ option.
- ‘RunTimeDisp’, *RunTimeDisp*
 - To suppress all three components of the default runtime display information (described at the end of section III.C.1.b.) use *RunTimeDisp* = ‘nnn’ (not case sensitive). The default is ‘yyy’. To suppress one or two of the components, replace *y* by *n* in the three-letter string; for example to show only the coefficients and confidence intervals (first component) use ‘ynn’, to show only the constituent selection diagnostic table (third component) use ‘nny’, etc. If the ‘NoDiagn’ option is selected then no constituent selection diagnostic table will be computed nor shown at runtime, regardless of the third character in *RunTimeDisp*.

III.C.1.d. Summary diagnostics table and diagnostic plots

Based on explanation of diagnostics in Section II.D above, by default UTide generates a summary table of diagnostics (*coef.diagn.table*) that provides information useful in the constituent selection process and is a character array that can be viewed easily within Matlab after execution of *ut_solv()*. The table is computed for uniformly or irregularly distributed times, as explained above. Its computation can be skipped using the ‘NoDiagn’ option.

The heading lines of the table show quantities not specific to individual constituents or constituent pairs. The first heading line shows the user-specified R_{\min} (81) and SNR_{\min} (96) values. The second heading line shows the basis matrix condition number K (84) and the all-constituent SNR_{allc} (85). The third heading line shows the tidal variances (a) TV_{allc} (99) of the model solution, (b) TV_{snrc} (102) of the reconstructed fit using constituents meeting the SNR criterion (96), and (c) TV_{raw} (99) of the raw input. The fourth and final heading line shows the percent tidal variances (a) PTV_{allc} (99) of the all-constituent model solution, and (b) PTV_{snrc} (102) of the reconstructed fit using constituents meeting the SNR criterion.

Within the table there is one row for each constituent, of frequency ω_q , and the rows are ordered by decreasing percent energy PE_q (97) values. The first three columns of the table are the name of the constituent, the PE_q value, and the constituent-specific SNR_q (83); at the far left an asterisk appears adjacent to the constituent name if it meets the SNR criterion (96). Next there is a group of columns presenting diagnostics related to the constituent with the next-lower frequency compared to ω_q ; finally, there is a group of columns with the same diagnostics related to the constituent with the next-higher

frequency compared to ω_q . In these latter two groups, the columns include the name of the neighboring constituent followed by R^R (81), R^{NM} (82), and $corr_{\max}$ (87), each computed for the respective constituent pair.

All types of constituents—non-reference, reference, and inferred—are listed together in the table. Diagnostics based on constituent pairs (R^R , R^{NM} , and $corr_{\max}$) are computed only between non-reference and reference constituents, with pairs chosen based on frequencies that are nearest to each other, regardless of whether an inferred constituent has frequency between them. If the solution included inference of one or more constituents, a list of them is shown at the bottom of the table with their respective reference constituents.

The layout of the table columns is such that the importance of the diagnostics within the columns is generally highest in the columns toward the left. That is, the PE_q and SNR_q diagnostics are likely to be of the most use, with the R^R , R^{NM} , and $corr_{\max}$ values also providing relevant information, but each of increasingly lower importance for most typical situations.

The PE_q rank-order of the constituents in the table makes it visually apparent which among them have captured the most energy. The SNR values of the higher-ranked constituents typically decrease in a similar manner to PE_q , but for the lower- PE_q constituents it is useful to inspect the SNR values carefully, and if they are too low, consider omitting or inferring the associated constituents.

The table is designed so that it is also easy to scan for and identify any R^R and R^{NM} values that are lower than 1. Such values indicate violations of the Rayleigh criteria and suggest that consideration should be given to removing these constituents from the model or inferring them. When the decision tree method of F77 has been used, the R^R values will all be greater than 1, but the R^{NM} values will provide useful additional information. Similarly, the $corr_{\max}$ values can be scanned easily for the relatively higher values, which will help identify potential pairs of constituents that may not be sufficiently independent from each other to both be included in the model unless one is inferred.

Through use of the ‘DiagnPlots’ option, two diagnostic figures (in addition to the diagnostics table) can be generated by `ut_solv()`. Each figure is generated with a call to the built-in function `figure()` without application of any rescaling, so the user will need to manually maximize them (e.g. by mouse) to make the plots more legible on the screen. The two figure windows will typically overlies each other when initially created.

The first figure has four frames in the two-dimensional case and three frames in the one-dimensional case. The top frame shows the text field `coef.diagn.rundescrip`, to provide a descriptive summary of the run characteristics. The second and third frames, in the two-dimensional case, show time series for the u and v components respectively: the

raw input, the reconstructed fit using constituents that meet the SNR criterion, and the residual; in the one-dimensional case there is one such frame. The bottom frame is a semi-logarithmic plot that shows, in the two-dimensional case, a vertical bar extending to $L^{smaj^2} + L^{smi\eta^2}$ for each constituent in order of increasing frequency, colored red if the SNR criterion is met and blue if not, along with a green dotted line showing $(\sigma_{L_{smaj}}^2 + \sigma_{L_{smi\eta}}^2) \cdot SNR_{\min}$ to indicate the height required for each bar to meet the SNR criterion. In the one-dimensional case the vertical bar heights are A^2 and the green dotted line is $\sigma_A^2 \cdot SNR_{\min}$.

The second figure has four frames in the two-dimensional case and two frames in the one-dimensional case. It shows information about only the constituents that meet the SNR criterion, and they are ordered by decreasing PE_q from left to right in each of the frames. In the two-dimensional case the four frames show the current ellipse parameter values, together with their 95% confidence intervals. In the one-dimensional case, the two frames show the amplitude and phase, together with their 95% confidence intervals.

III.C.2 Reconstructing fits with *ut_reconstr()*

The *ut_reconstr()* function has two main purposes. The first purpose is to enable calculation of reconstructed fits at an arbitrary set of times. The second purpose is to enable calculation of reconstructed fits that include a user-specified subset of constituents (see (100)), for example as identified based on other criteria in addition to, or in place of, the SNR threshold.

III.C.2.a. Input and output parameter descriptions

The *ut_reconstr()* input *t_fit* is a column vector of arbitrary times that

- contains Matlab *datenum* values, with units of days (as for *t_raw*, see above description);
- can be either equispaced or irregular; and
- can include NaNs and if so the outputs (*u_fit/v_fit* or *sl_fit*) will have corresponding NaNs.

The reconstructed fit (*u_fit/v_fit* or *sl_fit*) are column vectors that have

- the same size as *t_fit*; and
- the same units as *u_raw/v_raw* or *sl_raw*.

III.C.2.b. Defaults and options

In a call to the function *ut_reconstr()* (see opening portion of Section III.C for syntax) the default implementation, which is executed when no *{options}* parameters are passed in, includes in the reconstruction of the fit only the constituents for which SNR is greater than 2. This default behavior can be changed by use of the following option flag choices (which, as for the options to *ut_solv()*, are case-insensitive but cannot be abbreviated or truncated):

- ‘MinSNR’, *MinSNR*
 - This causes only those constituents with $SNR_q \geq \mathit{MinSNR}$ (a real scalar) to be used in the reconstruction. The default value is 2.

- ‘MinPE’, *MinPE*
 - This causes only those constituents with percent energy $PE_q \geq \mathit{MinPE}$ (a real scalar) to be used in the reconstruction. The default value is zero. If both of ‘MinSNR’ and ‘MinPE’ are selected then no constituent with *either* SNR or PE values lower than their respective specified thresholds will be included in the reconstruction. Constituents will be included in the reconstruction, or removed from it, by the *MinSNR* and/or *MinPE* criteria regardless of whether they were non-reference, reference, or inferred constituents in the solution.
- ‘Cnstit’, *Cnstit*
 - This causes only those constituents named in *Cnstit*, which must be selected from those which were included in the model during the *ut_solv* calculation that generated *coef*, to be used in the reconstruction. *Cnstit* is a cell array of 4-character strings, of the same format as the *cnsit* input to *ut_solv()* described above. Constituents that are listed in, or omitted from, *Cnstit* are included in the reconstruction, or not, regardless of whether they were non-reference, reference, or inferred constituents in the solution. They are also included regardless of their SNR and PE values; if ‘Cnstit’ is used then MinSNR and MinPE are ignored.

All other attributes of the reconstructed fit computed by *ut_reconstr()* are determined based on their configuration during the call to *ut_solv()* that created the *coef* input to *ut_reconstr()*. This includes, for example, whether the trend is included, whether nodal/satellite corrections use exact or linearized times, etc; this information is stored in *coef.aux.opt* and summarized in *coef.aux.rundescrip*. To compute a reconstruction with any of these attributes changed, an additional run of *ut_solv()* must be made, and the resulting *coef* passed to *ut_reconstr()*.

III.D. Functionality and syntax for groups of records

A group of multiple time sequences can be analyzed with a single execution of *ut_solv()* and, if needed, a corresponding group of hindcast/forecasts can be calculated with a single execution of *ut_reconstr()*. Each record in the group can have a different number and distribution of time values, a different latitude, and different inference constants. The associated modifications to the functionality and syntax of the inputs and outputs are described in this section.

- In an analysis of a group of records, the following should be borne in mind.
 - The automated constituent selection option is not available, so the group of constituents to be included must be manually specified. This should be straightforward to overcome, by determining a suitable set of constituents based on some preliminary runs with a few representative members of the group.
 - Each record in the group must have the same number of times. This can be facilitated, if necessary, by padding shorter records with NaNs. At the start and end of the shorter records the padded values of *t_raw* must be NaNs (rather than non-NaN time values); this is to ensure that the corresponding *u_raw/v_raw* or *sl_raw* values are not filled, in the equispaced times case.

- Diagnostics can be computed for each member in the group but the diagnostic figures cannot be generated; the ‘DiagnPlots’ option is not allowed.
- No runtime display is generated; ‘RunTimeDisp’ other than ‘nnn’ is not allowed; however, all potential runtime display information (*coef.results*, *coef.aux.rundescr*, and *coef.diagn.table*) for each record in the group is included in the output.
- The order of the elements in the output fields is by increasing frequency and the ‘OrderCnstit’ option is not allowed. This due to the fact that frequencies are the only ordering (unlike ordering by SNR or PE value) that is certain to be uniform across all records in the group, which is required by the form of the output fields as explained below.
- Any other options, if passed in, are applied to all members of the group: ‘NoTrend’, ‘PreFilt’, ‘NodSatLint’, ‘NodSatNone’, ‘GwchLint’, ‘GwchNone’, ‘Method’, ‘TunRdn’, ‘LinCI’, ‘White’, ‘Nrlzn’, ‘LSFrqOSmp’, ‘NoDiagn’, and ‘DiagnMinSNR’.

The group of n_s time sequences to be analyzed is indexed as an n_d -dimensional array of size $n_1 \times n_2 \times n_3 \dots \times n_{n_d}$, where each n value gives the size of that dimension of the array, and $n_s = n_1 n_2 n_3 \dots n_{n_d}$. For example, if the group consists of numerical simulation time series of sea level from a 20x10 array of lat-lon gridpoints, there are $n_s = 200$ time sequences, and a valid choice would be $n_1 = 20$ and $n_2 = 10$, for $n_{n_d} = 2$. Alternatively, they could be treated using $n_1 = 200$ and $n_{n_d} = 1$. As another example, if the group consists of a current observations from bottom-mounted acoustic Doppler current profilers (ADCPs) deployed along two across-shelf lines, each line having 5 ADCPs and each ADCP collecting current measurements from 50 depth bins, one configuration to treat the $n_s = 500$ time sequences would be $n_1 = 2$, $n_2 = 5$ and $n_3 = 50$, for $n_{n_d} = 3$.

The inputs to *ut_solv()* are just as in the single-record case described in the previous section, except for the following changes.

- *t_raw* can be either
 - a single $n_t \times 1$ vector of times that applies to all time sequences in the group, in which is case it is specified exactly as in the single-record case described above, or
 - an $n_t \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ array of times, as necessary when more than one record in the group has a different set of times; in this case the number of times must be the same (n_t) for each record, necessitating that if there are records with fewer times their time vectors must be padded with NaNs and their corresponding *u_raw/v_raw* or *sl_raw* must be padded with NaNs. Note also that none of n_1, n_2, \dots, n_{n_d} can be 1.
- *u_raw* and *v_raw*, or *sl_raw*, are each $n_t \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ arrays (again, none of n_1, n_2, \dots, n_{n_d} can be 1).

- *lat* is either a scalar, in which case the analysis of all the records will use the same value, or a $n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ array.
- *cnstit* cannot be ‘auto’, but rather must be a specific list of constituents to include; this is required in order that the same group of constituents is included for each individual analysis, which enables convenient grouping of the results fields in *coef* (explained below).
- If constituents are to be inferred, then the same inference and reference constituents will be used (same *Infer.infnam* and *Infer.refnam*) in each individual analysis, and either
 - *Infer.amprat* and *Infer.phsoff* are the same size as each other and are $2n_1 \times 1$ (two-dimensional case) or $n_1 \times 1$ (one-dimensional case) as explained above for treatment of a single record, in which case the same inference constants will be applied to every record, OR
 - *Infer.amprat* and *Infer.phsoff* are the same size as each other and are $2n_1 \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ (two-dimensional case) or $n_1 \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ (one-dimensional case), such that different inference constants can be applied to each record.
- Option flags (other than ‘infer’ as just noted) will be applied identically to each individual time sequence analysis, with the exception of ‘OrderCnstit’ and ‘DiagnPlots’, which will be ignored; ordering of constituent-indexed outputs is always by increasing frequency.

The output *coef* from *ut_solv()* is as in the single-record case described above except for the following changes.

- Fields of size $n_{allc} \times 1$ in the single-record case have size $n_{allc} \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ in the case of a group analysis.
- Fields of size 1×1 in the single-record case have size $n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ in the case of a group analysis.
- The ordering of the elements in each of these fields is by increasing constituent frequency, as explained above.

The exceptions are that, to avoid redundancy in fields of the output *coef*,

- *coef.name*, *coef.aux.frq*, and *coef.aux.lind* are each $n_{allc} \times 1$ (as determined by the fixed set of constituents in *cnstit*, which are included identically in the analysis of each record in the group),
- the only fields of *coef.aux.opt* that potentially differ from the single-record case are *equi*, *infer.amprat* and *infer.phsoff*, and the latter two are the same size as their corresponding inputs,
- if a single *lat* value was passed in then *coef.aux.lat* is a scalar.

The inputs to *ut_reconstr()* in the group case are as for the single-record case described above, with the following exception.

- *t_fit* can be either

- a single $n_t \times 1$ vector of times to be used for all time sequences in the group, in which case it is specified exactly as in the single-record case described above, or
- an $n_t \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ array of times, in which case each record is computed at its own set of times, though the number of times must be the same (n_t) for each record.

The output from *ut_reconstr()* in the group case is the same as for the single-record case except that for *u_fit* and *v_fit*, or *sl_fit*, each is an $n_t \times n_1 \times n_2 \times n_3 \dots \times n_{n_d}$ array instead of an $n_t \times 1$ column vector.

III.E. Relationships to existing software

The UTide code incorporates features of (a) the *t_tide* Matlab functions (PBL02), including use of certain important *t_tide* components unmodified (the database of harmonic constants, the constituent selection decision tree code, the linearized confidence interval calculations, the band-averaging of residual spectra); (b) the *r_t_tide* Matlab functions (LJ09); and (c) the “versatile tidal analysis” Fortran program (FCB09).

UTide includes all functionality of *t_tide* except for its XTide capabilities. However, exact agreement with *t_tide* in the case of a record with an even number of points cannot be achieved by UTide because the way *t_tide* drops the last point is not compatible in general with the capability of UTide to accept irregular times. In addition, the confidence interval calculations of UTide cannot exactly recover those of *t_tide*. In *t_tide* the scaling of spectral quantities for the colored case included an extra factor of two, and simplifying assumptions were made (as explained in detail above) about the covariance matrix (59), that affect both the white and colored case, but are not made in UTide. An additional, though minor, contribution to the differences in Monte Carlo cases is due to the stochastic nature of the calculation, which causes each run (of either UTide or *t_tide*) to yield slightly different results. As a result of these relationships, in order to achieve the closest agreement of UTide results with those of *t_tide*, for testing purposes, there are a number of requirements. First, UTide must be executed with an odd number of points. Second, if *t_tide* is called with both the *start_time* and *lat* inputs, the equivalent call to UTide requires the ‘NodsatLinT’ and ‘GwchLinT’ options; if *t_tide* is called with only the *start_time* input, the equivalent call to UTide requires the ‘NodsatNone’ and ‘GwchLinT’ options; if *t_tide* is called without the *start_time* input, the equivalent call to UTide requires the ‘NodsatNone’ and ‘GwchNone’ options; and if inference calculations are done by *t_tide* the equivalent call to UTide requires the ‘InferAprx’ option. Third, the ‘OLS’ option to UTide must be used because *t_tide* does not implement the IRLS method that is the default for UTide. Finally, it should be noted that calling UTide with the ‘OrderCnstit’, ‘*freq*’ option will make comparisons to *t_tide* output more convenient.

The IRLS features of *r_t_tide* are included in UTide, including their application to two-dimensional raw input and irregularly distributed times. Because *r_t_tide* is a modification of *t_tide* to include the IRLS solution method, the above noted relationships

between UTide and `t_tide` generally apply when comparisons between UTide and `r_t_tide` outputs are made.

All features of the FCB09 Fortran code for the one-dimensional case are available in UTide (in addition, UTide includes their generalization to the two-dimensional case), with the exceptions that (a) UTide does not include the same covariance-based constituent selection diagnostics developed from the FCB09 singular value decomposition, and (b) UTide confidence intervals are based on the new formulation and reported for the current ellipse parameters, while those of the FCB09 code are for the cosine/sine coefficients and based on the white noise presumption. The only change to the default configuration of UTide, in order to create results with the closest agreement to those of the “versatile tidal analysis” program, is to use the option ‘OLS’.

III.F. Computational demands

The computational demands of UTide are significantly higher than some previous tidal analysis software, for the following reasons: (i) the IRLS solution method is used, which involves multiple iterations of solutions each with demand similar to an OLS solution; (ii) treating nodal/satellite corrections and Greenwich phase calculations using the exact times substantially increases the memory requirements and the number of computations, (iii) the complex-valued formulation of the matrix system is solved, for reasons explained above, which can be less efficient than solving comparable real-valued formulations; (iv) in the case of irregular times, the Lomb-Scargle periodogram calculations are slower than their FFT counterparts for uniformly distributed times; (v) the generality of the new confidence interval calculation is slightly more costly than earlier versions; and (vi) the constituent selection diagnostics require additional computation. The relative importance of each of these factors in contributing to the increased computation demand will of course differ depending on the particular analysis at hand (number of raw input times, whether they are irregular, number of constituents, one-dimensional or two-dimensional raw input, whether Monte Carlo is used for confidence intervals, etc). However, the above list of reasons for increased burden is roughly in order of decreasing importance, very generally.

Crude guidelines for the computational burden result from summarizing the results of numerous analyses (each an execution of `ut_solv()` then `ut_reconstr()` in sequence; the large majority of the time is spent on the former) of various test datasets of hourly sea level (Newport, RI) and currents (Martha’s Vineyard Coastal Observatory). The datasets were sampled uniformly or irregularly for durations between a month and 5 years. Using a modest-capability 2007-era laptop PC with Matlab 2010a, when configured to mimic `t_tide` computations (i.e. ‘OLS’, ‘NodsatLinT’, ‘GwchLinT’) UTide used a comparable amount of memory, and was approximately twice as slow, compared to `t_tide`. This meant run times from between about 1-2 seconds and 20-30 seconds, for the one month and 5 year records respectively. When the exact formulations for nodal/satellite and Greenwich phase lag calculations were implemented, the run times increased by about 2-5 times, but remained comparable in speed or faster than the Fortran code of FCB09, except for records of a year or more. Such records become significantly slower and, notably when the Lomb-Scargle periodogram is calculated (colored case with

irregular times), very memory-intensive. When the IRLS solution method was also implemented, with the Cauchy weight function and default tuning parameter, the run times increased by an additional factor of 2 for the shorter records and by a higher amount, more than an order of magnitude in some cases, for the longer records.

These results just described are of course only applicable for the particular test records and computational system used; in general, results will vary depending on signal to noise characteristics of the raw inputs, the nature of the sampling and duration of the records, and the configuration of the runtime options, as well as the computing resources. Nonetheless, it is clear that relative to other tidal analysis software, the additional burden of UTide is at most a few orders of magnitude higher, for records up to 5 years long, and in many cases a smaller increase. Considering that availability of computing power for a typical researcher goes well beyond a typical 2007 laptop, in most applications these costs seem modest enough not to be a major constraint.

In this context, UTide has been developed based on the view that the additional features it incorporates are sufficiently valuable that they offset the undesirable increase in computational demands. However, in order to lessen the computational burden, to the extent it is possible, certain aspects can be omitted from a calculation. For example, the ‘OLS’ option will forego the cost of the IRLS computations, the ‘White’ option will eliminate the spectral calculations (notably the Lomb-Scargle algorithm for irregular times), the ‘LinCI’ option obviates the Monte Carlo random realizations, the ‘NoDiagn’ option omits computation of constituent selection diagnostics, and a smaller ‘Nrlzn’ value will require less cycles when using Monte Carlo. If computational demands are a constraint, then a good strategy will be to do initial calculations using these options (e.g., ‘OLS’, ‘White’, ‘LinCI’, possibly ‘NoDiagn’) and then later carry out a select few runs that involve the slower features as needed.

There are aspects of UTide for which future modifications could potentially increase its computational efficiency substantially. Examples would be to (a) create a precompiled executable in a different language, (b) implement more efficient code, including the matrix solution formulations, and/or (c) use a ‘fast’ Lomb-Scargle algorithm, such as that of Press and Rybicki (1989), which will reduce processing time although, as explained above, the major demand of Lomb-Scargle calculations is on memory. There are also potential modifications that would lessen its speed in return for reduced memory demands, which is useful when memory limitations are more of a constraint than processing speed; for example, an alternative loop arrangement for the Lomb-Scargle code could reduce the memory demand with the trade-off of slower runtimes. Pursuit of such improvements can follow on an as-needed basis.

IV. Acknowledgements

The effort would not have been possible without code made available by Mike Foreman and colleagues, Rich Pawlowicz and colleagues, Keith Leffler, and David Jay, all of whom were generous with their advice. Comments from Wendy Callendar on an early version resulted in improvements and motivated the group analysis capability. Even Haug (Hydrographic Service, Norwegian Mapping Authority) made several helpful suggestions, including an improved handling of cross-spectral quantities in the colored scaling calculation. This material is based upon work supported by the National Science Foundation, Physical Oceanography, under Grant No. 0826243, “Investigating Tidal Influences on Subtidal Estuary-Coast Exchange Using Observations and Numerical Simulations”. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

V. References

- Cherniawsky, J.Y., M.G.G. Foreman, W.R. Crawford, R.F. Henry. 2001. Ocean Tides from TOPEX/Poseidon Sea Level Data. *J. Phys. Ocean.* 18, 649-664.
- Codiga, D.L., L.V. Rear. 2004. Observed tidal currents outside Block Island Sound: Offshore decay and effects of estuarine outflow. *J. Geophys. Res.* 109, doi:10.1029/2003JC001804.
- Codiga, D.L. 2007. FOSTER-LIS Gridded Data Products: Observed Current Profiles and Near-Surface Water Properties from Ferry-based Oceanographic Sampling in Eastern Long Island Sound. In, Graduate School of Oceanography, University of Rhode Island, Narragansett, RI., pp. 14.
- Emery, W.J., R.E. Thomson. 1998. Data Analysis Methods in Physical Oceanography. Pergamon, New York, 634pp.
- Foreman, M.G.G. 1977. Manual for tidal heights analysis and prediction. . *Pacific Marine Science Rep.* 77-10, Institute of Ocean Sciences, Patricia Bay, 101 pp. [Revised 2004; Available online at <http://www.pac.dfo-mpo.gc.ca/SCI/osap/publ/online/heights.pdf>].
- Foreman, M.G.G. 1978. Manual for tidal currents analysis and prediction. . *Pacific Marine Science Rep.* 78-6, Institute of Ocean Sciences, Patricia Bay, 70 pp. [Revised 2004; Available online at <http://www.pac.dfo-mpo.gc.ca/SCI/osap/publ/online/heights.pdf>].
- Foreman, M.G.G., R.F. Henry. 1989. The harmonic analysis of tidal model time series. *Adv. Wat. Res.* 12, 109-120.
- Foreman, M.G.G., E.M. Neufeld. 1991. Harmonic tidal analyses of long time series. *Int. Hydrogr. Rev.* LXVIII, 85–108.
- Foreman, M.G.G., J.Y. Cherniawsky, V.A. Ballantyne. 2009. Versatile Harmonic Tidal Analysis: Improvements and Applications. *J. Atmos. Oceanic Tech.* 26, 806-817. DOI: 810.1175/2008JTECHO1615.1171.
- Godin, G. 1972. The analysis of tides. University of Toronto Press, Toronto.
- Goodman, N.R. 1963. Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). *Ann. Math. Statist.* 34, 152–177.
- Higham, N.J. 2002. Computing the nearest correlation matrix—A problem from finance. *IMA J. Numer. Anal.* 22, 329–343.
- Leffler, K.E., D.A. Jay. 2009. Enhancing tidal harmonic analysis: Robust (hybrid L-1/L-2) solutions *Cont. Shelf Res.* 29, 78-88. DOI: 10.1016/j.csr.2008.1004.1011
- Lomb, N.R. 1976. Least-squares frequency analysis of unequally spaced data. *Astrophys. Space Sci.* 39, 447-462.
- Munk, W., K. Hasselmann. 1964. Super-resolution of tides. *Studies on Oceanography - A Collection of Papers Dedicated to Koji Hidaka* K. Yoshida, Ed., University of Tokyo, 339-344.
- Parker, B.B. 2007. Tidal analysis and prediction. *NOAA Special Publication NOS CO-OPS 3, U.S. Department of Commerce*, 378 pp.
- Pawlowicz, R., B. Beardsley, S. Lentz. 2002. Classical tidal harmonic analysis including error estimates in MATLAB using T-TIDE. *Computers & Geosciences* 28, 929-937.
- Press, W.H., G.B. Rybicki. 1989. Fast algorithm for spectral analysis of unevenly sampled data. *Astrophysical Journal* 338, 277-280.

- Press, W.H., S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. 1992. Numerical Recipes in FORTRAN: The Art of Scientific Computing. Cambridge University Press, Cambridge, U. K.
- Scargle, J.D. 1982. Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data. *Astrophys. J.* 263, 835-853.
- Schulz, M., K. Statterger. 1997. SPECTRUM: Spectral analysis of unevenly spaced paleoclimatic time series. *Comp. Geosci.* 23(9), 929-945.