



A note on interpreting tidal harmonic constants

Zhigang Xu¹

Received: 21 January 2017 / Accepted: 17 November 2017 / Published online: 19 December 2017
© The Author(s) 2017. This article is an open access publication

Abstract

The amplitudes and phase lags of tides or tidal currents, which are collectively known as tidal harmonic constants (THCs), cannot be interpolated linearly and separately. Mistreatment for the interpolations could occur, and indeed, it is seen even in the literature. This note clarifies this topic by providing correct formulas to interpret THCs. One has to perform a nonlinear and coupled interpolation of THCs to have equivalence to the linear interpolation in the time domain.

Similarly, tidal current ellipse parameters (ep-parameters) cannot be interpolated linearly and separately. To interpolate ep-parameters, this note recommends that one should first convert them to amplitude and phase lag parameters (ap-parameters) and then use the nonlinear and coupled interpolants proposed herein to interpolate the ap-parameters, followed by converting the results back to ep-parameters.

Examples are provided to illustrate the problems and MATLAB functions are provided in Appendix as interpolating tools.

Keywords Interpolation · Tidal harmonic constants · Amplitudes · Phase lags · Tidal current ellipse parameters

1 Introduction

Tidal heights or currents at a location with a single tidal constituent frequency ω can be described as follows:

$$u(t) = a \cos(\omega t - \phi) \quad (1)$$

where a and ϕ are the amplitude and the phase lag, respectively, and the value of ω can be assigned based on astronomic knowledge. The amplitude and phase lag at one location are constant in time; therefore, they are also collectively referred to as tidal harmonic constants (THCs). Often, interpolations of THCs must be performed. For example, given two pairs of THCs at two locations, say (a_1, ϕ_1) and (a_2, ϕ_2) at x_1 and x_2 , we may want to interpolate them to obtain a new pair of THCs at a point somewhere in between.

An incorrect approach is to interpolate the amplitudes and phase lags linearly and separately. For amplitude, one might calculate as follows:

$$a = \alpha a_1 + \beta a_2 \quad (2)$$

where α and β are interpolation weights, and their sum is equal to one. For phase lag, one might calculate as follows:

$$(\cos \phi, \sin \phi) = (\alpha \cos \phi_1 + \beta \cos \phi_2, \alpha \sin \phi_1 + \beta \sin \phi_2) \quad (3)$$

$$\phi = \text{atan2}(\sin \phi, \cos \phi) \quad (4)$$

where atan2 is an inverse function of the tangent and provides angles ranging from $-\pi$ to π (e.g., Park et al. (2012)). However, this is not correct as we will see in the next section. This incorrect approach could even be carried over to interpolating tidal current ellipse parameters (Byun and Hart 2017). To clarify the confusion, this note is dedicated to presenting correct THC interpolants.

The next section will derive correct THC interpolants. The “Examples” section will provide examples for 1-D and 2-D interpolations to demonstrate how the interpolants are applied and to illustrate the differences from the results obtained using the incorrect interpolants as of Eqs. (2), (3), and (4). Five MATLAB functions for the THC interpolations are presented in Appendix.

2 Nonlinear and coupled interpolants for THCs

This section derives two interpolants for amplitudes and phase lags, respectively. The interpolants will be nonlinear and coupled in terms of given amplitudes and phase lags (ap-

Responsible Editor: Lie-Yauw Oey

✉ Zhigang Xu
zhigang.xu@dfb-mpo.gc.ca

¹ Maurice Lamontagne Institute, Fisheries and Oceans Canada, Mont-Joli, Quebec, Canada

parameters). The starting point for the derivation is a linear interpolation in terms of the time functions of tidal heights or currents as expressed by Eq. (1). It is emphasized here from the outset that to have a linear interpolation in the time domain, one has to perform a nonlinear interpolation in the ap-parameters domain (which may also be referred as the spectrum domain). This is understandable because the ap-parameters work together nonlinearly to affect tidal heights and currents.

Suppose that at $x = x_1$ and $x = x_2$ (assuming $x_2 > x_1$), two pairs of THC's, (a_1, ϕ_1) and (a_2, ϕ_2) , are known, with which we can describe tidal height (or a current component) at these two locations as follows:

$$u_1(t) = a_1 \cos(\omega t - \phi_1), \tag{5}$$

$$u_2(t) = a_2 \cos(\omega t - \phi_2). \tag{6}$$

Also, suppose that we would like to know $u(t)$ at x somewhere in between x_1 and x_2 by a linear interpolation of $u_1(t)$ and $u_2(t)$, i.e.,

$$u(t) = \alpha u_1(t) + \beta u_2(t) \tag{7}$$

where α and β are interpolation weights, with $\alpha = (x_2 - x) / (x_2 - x_1)$ and $\beta = 1 - \alpha$ (interpolation in the time domain). Further, assume that we do not wish to perform interpolation for every point of time; rather, we wish to perform interpolation once and directly on the two pairs of THC's to obtain a new pair of THC's at x (interpolation in the spectrum domain). When the new pair of THC's interpolated from the spectrum domain is inserted into Eq. (1), the resulting time series $u(t)$ will be identical to the one interpolated in the time domain using Eq. (7). This is the mathematical setup for the problem. Now, let us solve it.

In fact, it can be easily solved by expanding Eq. (7) as follows:

$$u = (\alpha a_1 \cos \phi_1 + \beta a_2 \cos \phi_2) \cos \omega t + (\alpha a_1 \sin \phi_1 + \beta a_2 \sin \phi_2) \sin \omega t \tag{8}$$

Equation (1) can be rewritten as follows:

$$u(t) = a \cos \phi \cos \omega t + a \sin \phi \sin \omega t. \tag{9}$$

By equating the coefficients of $\cos \omega t$ and $\sin \omega t$ from the right-hand sides of the above two equations, we obtain the following:

$$a \cos \phi = \alpha a_1 \cos \phi_1 + \beta a_2 \cos \phi_2 \tag{10}$$

$$a \sin \phi = \alpha a_1 \sin \phi_1 + \beta a_2 \sin \phi_2 \tag{11}$$

from which we can solve for a and ϕ as follows:

$$\phi = \text{atan2}(\alpha a_1 \sin \phi_1 + \beta a_2 \sin \phi_2, \alpha a_1 \cos \phi_1 + \beta a_2 \cos \phi_2) \tag{12}$$

$$a = \frac{\alpha a_1 \sin \phi_1 + \beta a_2 \sin \phi_2}{\sin \phi} \tag{13}$$

$$\underline{\underline{or}} \quad \frac{\alpha a_1 \cos \phi_1 + \beta a_2 \cos \phi_2}{\cos \phi} \tag{14}$$

Note that by using `atan2` (in MATLAB) instead of `atan` in Eq. (12), the amplitude a values in Eqs. (13) and (14) are guaranteed to be positive. Equations (12) and (13) (or Eq. (14)) are the correct interpolants to interpolate THC's. The time series reconstructed from so interpolated THC's will observe the linear interpolation in the time domain as specified in Eq. (7). This will not be the case if Eqs. (2) to (4), or any other formulas, are used as interpolants instead.

When 2-D interpolations are required, the 1-D interpolation can be applied in one direction first, and the results can be further interpolated in another direction. This process will be illustrated in examples 2 and 3 in the next section.

This paper is meant as a short note paper and could end here now. However, to help readers better intuit the problem and observe the consequences of correct and incorrect interpolations graphically, some examples are provided in the next section. The examples will also bring in MATLAB functions for THC interpolation, which should be useful for ocean modelling community.

3 Examples

This section shows three examples, one for a 1-D interpolation and two for 2-D interpolations. Five MATLAB functions listed in Appendix for the THC interpolations will also be briefly introduced.

3.1 Example 1: 1-D interpolation

Let $\omega = 2\pi/T$ with $T = 12$ h for the semi-diurnal frequency, let $a_1 = 1.50$ m and $\phi_1 = 45.00$ deg at $x_1 = 0$, let $a_2 = 2.00$ m, and $\phi_2 = -45.00$ deg at $x_2 = 1$ (the unit of the coordinates is understood because it is not relevant to the interpolation problem here). Suppose at $x_i = 0.6$, we want to know the THC's by interpolation. First, we need to calculate the interpolation weights of α and β according to the following:

$$\alpha = \frac{x_2 - x_i}{x_2 - x_1}, \tag{15}$$

$$\beta = 1 - \alpha = \frac{x_i - x_1}{x_2 - x_1}. \tag{16}$$

By substituting the given x_1 , x_2 , and x_i , we have $\alpha = 0.35$ and $\beta = 0.65$. With the given (a_1, ϕ_1) and (a_2, ϕ_2) , we then have the following:

$$\begin{aligned} \phi &= \text{atan2}(-0.5480, 1.2905) = -0.4016 \\ &= -23.0089\text{deg} \stackrel{or}{=} 336.9911\text{deg} \end{aligned} \tag{17}$$

$$a = -0.5480/(-0.3909) = 1.4020 \text{ m} \tag{18}$$

In comparison, using the interpolants of Eqs. (2)–(4), we obtain values of $a = 1.8250 \text{ m}$ and $\phi = -16.6992 \text{ deg} \stackrel{or}{=} 343.3008 \text{ deg}$, which differ greatly from the correct values.

Figure 1 shows the four time series of u_1 and u_2 and two versions of the interpolated u_i that represent the correct and incorrect versions. The green curve (the correct interpolation) always remains between the two data curves (blue and black), which is most evident when the green curve runs through the points where the black and blue curve intersect. This is what we expect from a linear interpolation in the time domain between the two data curves. If linear interpolation is performed for every point of time according to Eq. (7), the result will be the same as the green curve. However, the red curve (the incorrect interpolation) sometimes travels even beyond the envelope of the two data curves; in fact, it violates the linear interpolation required by Eq. (7) all the time.

List 1 and List 2 in Appendix are two MATLAB functions `interp1_w` and `THCinterp1`, which are used for this example. The function `interp1_w` calculates the interpolation weights of the given coordinates for a set of data points (x) and coordinates for the interpolation points (x_i). The function `THCinterp1` implements the 1-D interpolation using Eqs. (12) and (13).

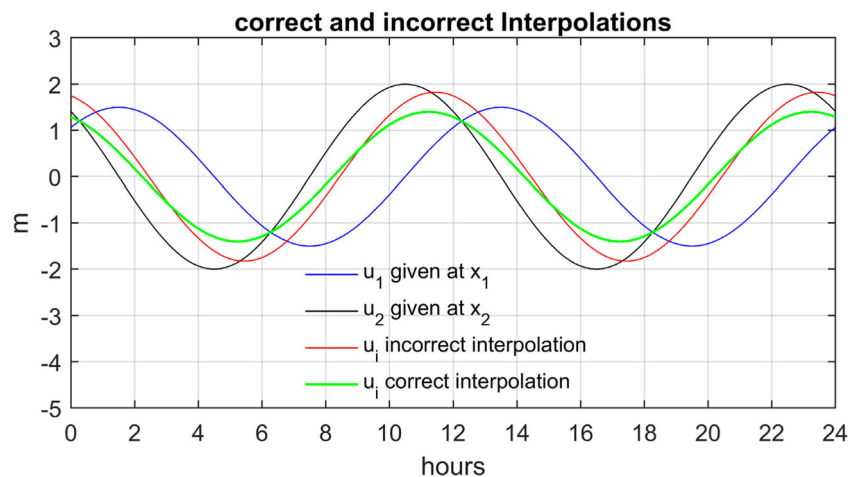
3.2 Example 2: 2-D interpolation of amplitudes and phase lags

This example demonstrates how the 1-D interpolants can be extended for a 2-D interpolation. The top two panels of Fig. 2 show amplitudes (left panel) and phase lags (right

panel), with the original THC data defined on the grid points printed in blue. The interpolation problem here is to interpolate the original THC data on to the centers of the cells as indicated by the circles. The MATLAB function `THCinterp2mat` in List 3 of Appendix is used for interpolation. It applies the 1-D interpolants of Eqs. (12) and (13) twice. The first application is in the x -direction, where the amplitudes and phase lags on each of the horizontal lines are interpolated to the middle points of each segment marked by the “x” on the horizontal lines. With the interpolated amplitudes assigned on the “x” points, the 1-D interpolation is applied again but in the y -direction. In terms of spatial coverage, this bi-directional approach is the same as a standard bi-linear interpolation in the sense that they both apply a one-dimensional interpolant twice. However, in terms of interpolants used, the bi-directional approach proposed here is different from the standard bi-linear interpolation. The former uses a nonlinear interpolant, and the latter uses a linear interpolant.

The interpolated results are printed in green beneath the circle points. For comparison, the results obtained using the incorrect 1-D interpolants of Eqs. (2)–(4) in the above bi-directional approach are also printed in red above the circles. To see how the time series will differ using these correct and incorrect THCs, let us focus on the shaded cell on the second row and fourth column of the top two panels. From the correctly and incorrectly interpolated amplitudes and phase lags shown in the middle of the cell, we can reconstruct two versions of the time series. We can also use the amplitudes and phase lags shown on the four corners of the cell to reconstruct four data time series. All the time series are shown in the bottom panel in green, red and gray, respectively. Since the interpolation point is in the center of the cell, the correct interpolated value is an arithmetical average of the four gray curves at each time step, as is the case with the green curve. The red curve is clearly incorrect.

Fig. 1 Four time series of u_1 and u_2 and two versions of u_i that represent the correct and incorrect interpolations. The correct interpolation (green) is reconstructed from the amplitude and phase lags interpolated using Eqs. (12) and (13). The incorrect interpolation (red) is reconstructed using Eqs. (2)–(4)



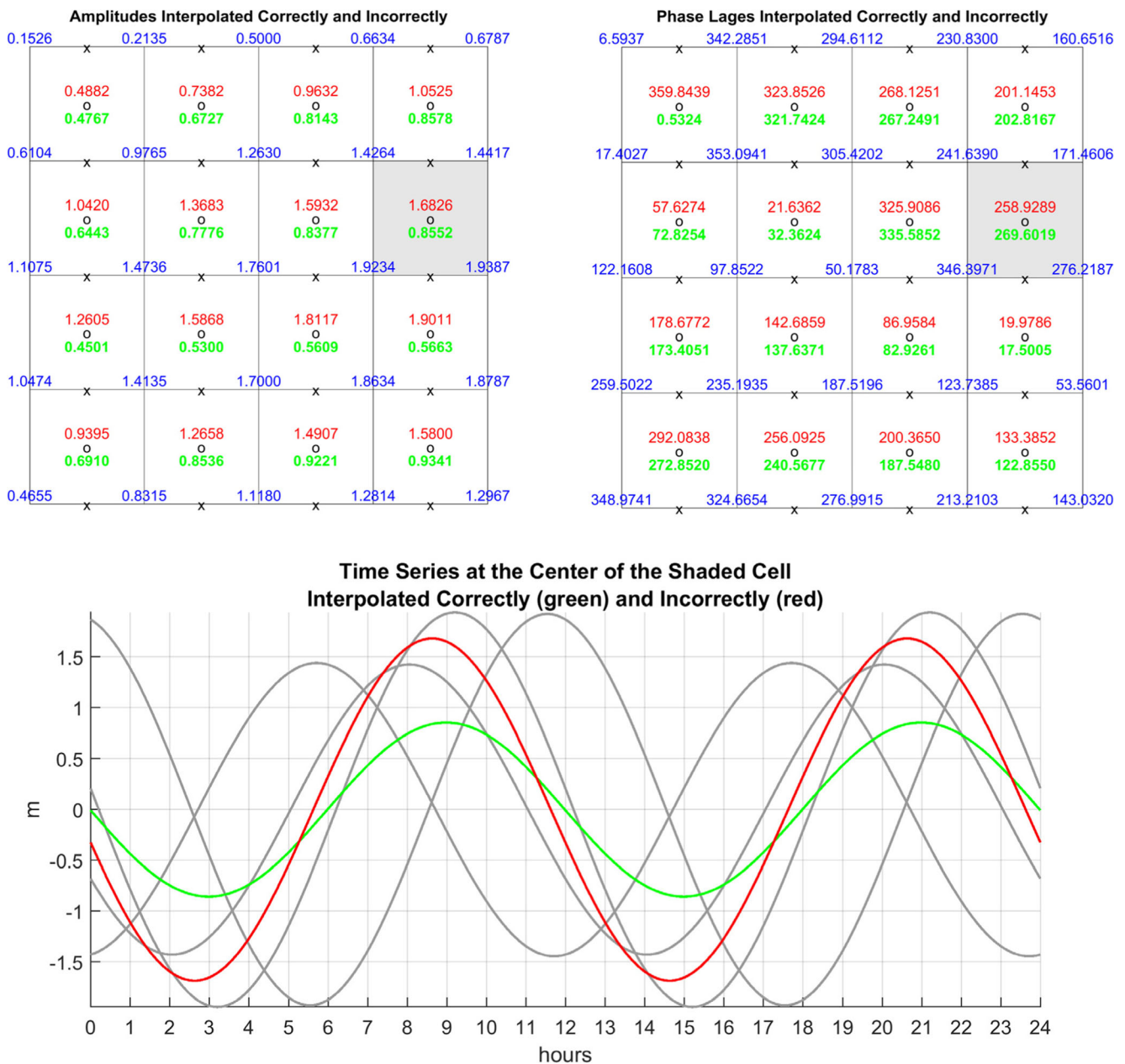


Fig. 2 2-D interpolation of tidal harmonic constants. The top left and right panels show amplitudes and phase lages, respectively. The original data are shown in blue on the corners of the cells, and interpolated data are shown in green and red in the centers of the cells. The green values are correctly interpolated using Eqs. (12) and (13), and the red values are incorrectly interpolated using Eqs. (2) and (4). Using the correct and incorrect amplitudes and phase lages shown in centers of the two shaded

cells, we can reconstruct two versions of time series. The correct one in green and the incorrect one in red are shown together with the four data time series reconstructed using the original data shown on the four corners of the same cells. The incorrectness of the red curve becomes evident when it moves out of scope of the four data curves. The green curve consistently remains in between the four data curves.

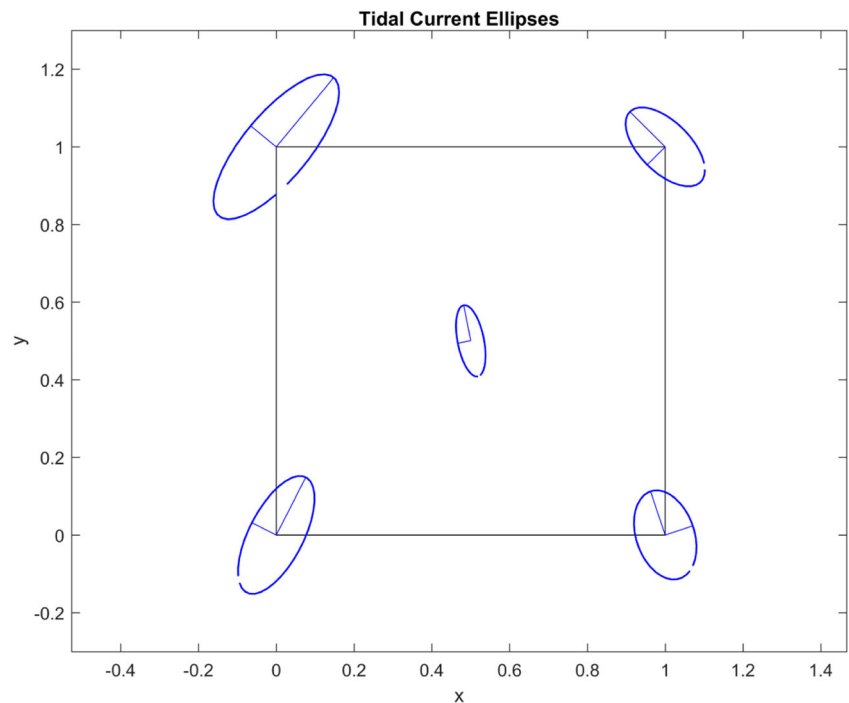
The function `THCinterp2mat` assumes that `xx` and `yy` are gridded data coordinates and the interpolation points of `xxi` and `yyi` are also gridded. There are also cases where we need to interpolate from gridded data points to irregular points. The MATLAB function `THCinterp2` provided in List 4 in Appendix satisfies this need. This function also works for cases where the interpolation points are gridded points as well; however, the computational speed will be slightly slower than using

`THCinterp2mat`. Interpolations from irregular points to irregular points can be also considered but are not addressed further here.

3.3 Example 3: 2-D interpolation of tidal current ellipse parameters

For tidal current, we can generally neglect its weak vertical component and focus on its two strong horizontal

Fig. 3 Tidal current ellipse parameter interpolation. The data ellipses distributed at the four corners of the grid cells and the interpolated ellipse at the center of the cell are shown. The interpolation is performed by the function *THCepinterp2* as provided in List 5 of Appendix



components, u and v . Referring to Eq. (1), we see that to describe u and v , we need two pairs of parameters, (a_u, ϕ_u) and (a_v, ϕ_v) . Interpolations on these ap-parameters can be performed with Eqs. (12) and (13), following the procedures illustrated by the above two examples.

On the other hand, by plotting v against u , we will see an ellipse. Therefore, tidal currents can also be described by ellipse parameters (ep-parameters). The ap-parameters and ep-parameters can be converted into each other by using the *ap2ep* and *ep2ap* MATLAB functions developed by Xu (2000, 2002). One might work out a set of interpolants that can be directly applied to ep-parameters, but such a set might be very complicated. It is better to convert ep-parameters to ap-parameters first, then using Eqs. (12) and (13) as the interpolants, and then convert the results back to the ep-parameters. The MATLAB function *THCepinterp2* shown in List 5 in Appendix assumes these steps.

Figure 3 illustrates an interpolation of tidal ellipses. Four data ellipses are provided at four corners of a grid cell. The ellipse drawn on the center of the cell is interpolated from the four data ellipses. The interpolation can be performed by calling *THCepinterp2* in this way:

$$EP_i = \text{THCepinterp2}(xx, yy, EP, xi, yi) \tag{19}$$

where

$$xx = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad yy = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \tag{20}$$

are two matrices containing the coordinates of the four grid points, and the EP is a MATLAB structure with the ep-parameters of the data ellipses as its fields:

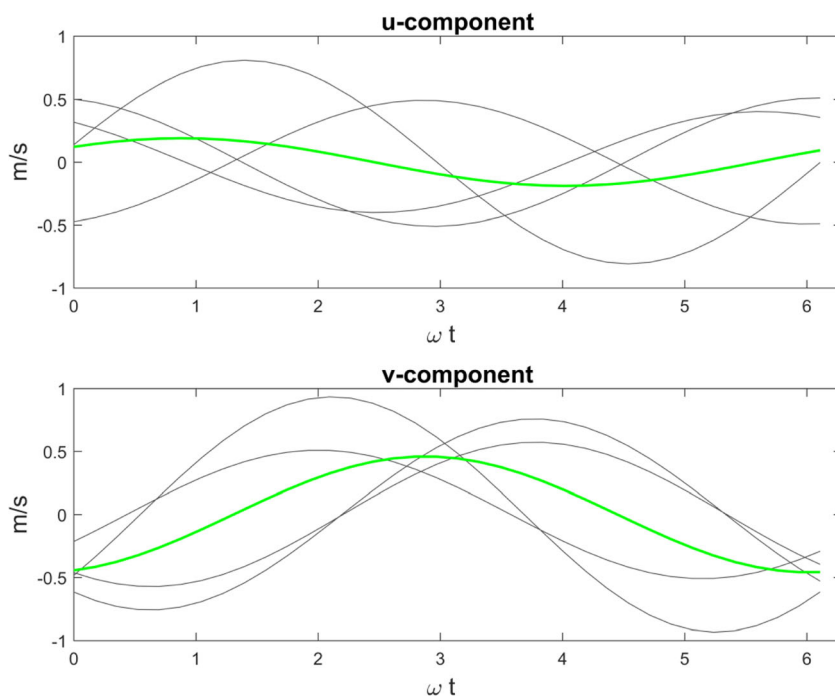
$$\begin{aligned} EP.SEMA &= \begin{bmatrix} 0.8333 & 0.5912 \\ 1.1587 & 0.6428 \end{bmatrix}, & EP.INC &= \begin{bmatrix} 62.9095 & 108.5705 \\ 50.4532 & 135.0000 \end{bmatrix}, \\ EP.ECC &= \begin{bmatrix} 0.4220 & -0.6338 \\ 0.3688 & 0.5090 \end{bmatrix}, & EP.PHA &= \begin{bmatrix} 203.5677 & 204.2242 \\ 104.2663 & 141.8655 \end{bmatrix}, \end{aligned} \tag{21}$$

where the four fields EP.SEMA, EP.INC, EP.ECC, and EP.PHA contain the lengths of the semi-major axes, the orientations (inclinations) of the semi-major axes, the eccentricities,¹ and the phase angles of the four data ellipses, respectively. The phase angle of a tidal ellipse indicates a time that is expressed as an angle (= time × frequency) when the tidal current reaches its maximum speed. The interpolation point is the center of the cell, which means that $x_i = 0.5$ and $y_i = 0.5$.

As we can see, the interpolated ellipse is much smaller than any of the four data ellipses. This seems surprising but it is actually correct, because the u - and v -time series represented by the four ellipses are out of phase to a large degree. Figure 4 shows time series of u -components (top panel) and v -components (bottom panel) reconstructed from the data ellipses (in gray) and from the interpolated ellipse (in green). Since the

¹ Due to a historical reason in oceanography, the eccentricity is defined as the ratio of the semi-minor axis to the semi-major axis of the ellipse, which is different to the usual mathematical definition. Its negative value indicates that the ellipse traverses in a clockwise rotation.

Fig. 4 Time series of u and v reconstructed from the four original tidal ellipses in gray and the interpolated ellipse in green. Because the interpolation point is at the center of the cell, the green curve is an average of the four gray curves



interpolation point is in the center of the cell, the green curve for the u -component should simply be an arithmetic average of the four gray curves of the u -components; so is the green curve for the v -component. As we can see, the four time series at the four corners of the cell have very different phases, which causes their average in the middle of the cell to have a much smaller amplitude. This example demonstrates the importance of treating tidal ellipse interpolation appropriately.

The semi-major axes of the ellipses are shown in the range of $[0, 180^\circ]$ in Fig. 3. The semi-minor axes are shown 90° from the semi-major axes, and they are either counter-clockwise or clockwise depending on whether the eccentricity is positive or negative. The inclinations of the semi-major axes do not have to be in the direction as shown. Their opposite directions could have well been chosen as the inclinations. However, to choose the inclinations to be in the range of $[0, 180^\circ]$ is conventional, the so-called the northern axis convention (Foreman 1978, p. 13). The `ap2ep` and `ep2ap` functions have two versions. The one provided by Xu (2000) allows inclinations in the range of $[0, 360^\circ]$, whereas the revised version by Xu (2002) adopts the northern axis convention. Both versions will give the same ap-parameters however. Therefore, no matter which version one may prefer, the interpolation will not be affected.

4 Conclusions

The amplitude and phase parameters (ap-parameters) for tidal heights and tidal currents cannot be interpolated in space as though they are independent quantities—errors arise when they are interpolated separately and linearly. This note paper provides correct non-linear and coupled interpolation interpolants: Eqs. (12) and (13) (or (14)), together with examples and MATLAB functions. When these interpolants are used, the time series reconstructed from the interpolated ap-parameters are identical to those obtained by interpolating the tidal height or current time series.

Similarly, the tidal current ep-parameters cannot be interpolated linearly and separately. First, they should be converted to ap-parameters. Then, the nonlinear and coupled interpolants can be used to interpolate the ap-parameters, followed by converting the results back to ep-parameters.

The MATLAB functions listed in Appendix provide tools to interpolate THCs. These functions relieve uses from the complexity of the interpolations and provide an easy user interface. The function `THCepinterp2` is a tool for interpolating tidal ep-parameters, a procedure that can otherwise be quite complicated and prone to error.

Appendix: MATLAB functions for interpolating THC

List 1 Function `interp1_w` for calculating the interpolation weights.

```

1  function W = interp1_w(x, xi)
2  % interp1_w: Outputs 1-D interpolation weights and associated info.
3  %
4  % Inputs: x: the data points for the interpolation
5  %         xi: the points for the interpolation
6  % Output: W: a structure with fields for interpolation weights for
7  %           alpha and beta and the indices i1 and i2 to locate the
8  %           interpolation data.
9  %
10 % Zhigang Xu, 2017/Jan/03

11 [x, n]=sort(x); % sort x in ascending order and
12 % re-arrange amp and phi.
13 if min(xi) < x(1) || max(xi) > x(end)
14     error('There are interpolation points outside the range of x!')
15 end
16 dx=diff(x);

17 if any(dx)==0
18     error('x must be distinct!')
19 end

20 i=1:numel(x);
21 jmax=numel(xi);
22 W.alpha=nan(size(xi));
23 W.beta=nan(size(xi));
24 W.i1=zeros(size(xi));
25 W.i2=zeros(size(xi));

26 for j=1:jmax
27     x0=xi(j);
28     i1=max(i(x0>=x)); i2=min(i(x0<=x));
29     x1=x(i1); x2=x(i2); dx=x2-x1;
30     if dx==0
31         W.alpha(j)=1;
32     else
33         W.alpha(j)=(x2-x0)/dx;
34     end
35 W.beta(j)=1-W.alpha(j);
36 W.i1(j)=n(i1);
37 W.i2(j)=n(i2);
38 end % % of the function

```

List 2 Function `THCinterp1` for performing 1-D interpolations of tidal harmonic constants

```

1  function [amp,phi]=THCinterp1(x,amp,phi,xi,isdeg)
2  % THCinterp1: 1-D interpolation of tidal harmonic constants (THC).
3  % Inputs
4  %     x: the 1-D spatial coordinates where the amplitudes and phase
5  %     lags are defined.
6  %     amp and phi: amplitudes and phase lags defined on x
7  %     xi: the coordinates for the interpolations.
8  %     isdeg: a logical variable to indicate if phi is input in
9  %     degrees or not (then in radians). Its default value is
10 %     true.
11 % Outputs:
12 %     amp and phi, the interpolated results.
13 %
14 % Zhigang Xu, 2017/Jan/03

15 if nargin<5, isdeg=true; end
16 if isdeg, phi=phi*(pi/180); end

17 W=interp1_w(x,xi);
18 i1=W.i1; i2=W.i2;
19 alpha=W.alpha;
20 beta=W.beta;
21 acpd=amp.*cos(phi);
22 aspd=amp.*sin(phi);

23 acp=alpha.*acpd(i1)+beta.*acpd(i2);
24 asp=alpha.*aspd(i1)+beta.*aspd(i2);

25 phi=atan2(asp, acp);
26 amp=asp./sin(phi);

27 if isdeg
28     phi=phi*(180/pi); id=phi<0;
29     phi(id)=phi(id)+360; % so that ampi is within the range[0 360);
30 end
31 end % end of the function

```


List 3 Function `THCinterp2mat` for performing 2-D tidal harmonic constant interpolations. Both the data coordinate points and interpolation points must be gridded

```

1  function [amp, phi] = THCinterp2mat (xx, yy, amp, phi, xxi, yyi, isdeg)
2  % THCinterp2mat: Tidal Harmonic Constant (THC) 2-D interpolant.
3  % Inputs:
4  %     xx, yy:      2-D grid coordinate matrices
5  %     amp, phi:    amplitudes and phase lags defined on xx and yy
6  %     xxi, yyi:   submatrices within xx and yy in which
7  %                 interpolated amplitudes and phases are needed.
8  %     isdeg,      an optional logical variable to indicate if phi
9  %                 is input in degrees or not. Its default value
10 %                 is true.
11 % Outputs: amp, phi: the interpolated amplitudes and phase lags.
12 %
13 % Zhigang Xu, 2017/Jan/06

14 if nargin<7, isdeg=true; end
15 if isdeg, phi=phi*(pi/180); end
16 x=xx(1, :); xi=xxi(1, :);
17 y=yy(:, 1); yi=yyi(:, 1);

18 %% Interpolation in the x-direction
19 acp=amp.*cos(phi); asp=amp.*sin(phi);
20 W=interp1_w(x, xi); j1=W.i1; j2=W.i2;
21 [m, ~]=size(asp);
22 alpha=ones(m, 1)*W.alpha; beta=ones(m, 1)*W.beta;
23 acp=alpha.*acp(:, j1)+beta.*acp(:, j2);
24 asp=alpha.*asp(:, j1)+beta.*asp(:, j2);

25 %% Interpolation in the y-direction
26 W=interp1_w(y, yi); i1=W.i1; i2=W.i2;
27 [~, n]=size(asp);
28 alpha=W.alpha*ones(1, n); beta=W.beta*ones(1, n);
29 acp=alpha.*acp(i1, :)+beta.*acp(i2, :);
30 asp=alpha.*asp(i1, :)+beta.*asp(i2, :);
31 phi=atan2(asp, acp); amp=asp./sin(phi);

32 %% output
33 if isdeg
34     phi=phi*(180/pi); id=phi<0;
35     phi(id)=phi(id)+360; % so that phi is within the range[0 360);
36 end

37 end %

```

List 4 Function `THCinterp2` for performing 2-D tidal harmonic constant interpolations. The data coordinate points must be gridded, whereas the interpolation points do not have to be gridded

```

1 function [amp,phi]=THCinterp2(xx,yy,amp,phi,xxi,yyi,isdeg)
2 % THCinterp2: Tidal Harmonic Constant (THC) 2-D interpolant.
3 % Inputs:
4 %     xx, yy:      2-D grid coordinate matrices
5 %     amp, phi:    amplitudes and phase lags defined on xx and yy
6 %     xxi, yyi:    coordinates of a set of points within xx
7 %                 and yy in which interpolated amplitudes and
8 %                 phases are needed. Note that xxi and yyi can be
9 %                 matrices or both column or row vectors.
10 %     isdeg,      an optional logical variable to indicate if phi
11 %                is input in degrees or not (then in radians).
12 %                Its default value is true.
13 % Outputs: amp, phi: the interpolated amplitudes and phase lags.
14 %
15 % Zhigang Xu, 2017/Jan/06

16 if nargin<7, isdeg=true; end
17 if isdeg,phi=phi*(pi/180);end
18 x=xx(1,:); y=yy(:,1);

19 %% Interpolation in the x-direction
20 acp=amp.*cos(phi); asp=amp.*sin(phi);
21 [xi, g]=numunique(xxi);
22 if ~isrow(xi)
23     xi=xi.';
24 end
25 W=interp1_w(x,xi); j1=W.i1; j2=W.i2;

26 [m,~]=size(asp);
27 alpha=ones(m,1)*W.alpha; beta=ones(m,1)*W.beta;
28 acp=alpha.*acp(:,j1)+beta.*acp(:,j2);
29 asp=alpha.*asp(:,j1)+beta.*asp(:,j2);

30 %% Interpolation in the y-direction
31 yi=yyi(g{1});
32 if ~iscolumn(yi), yi=yi(:);end
33 W=interp1_w(y,yi);
34 yiold=yi;
35 amp=nan(size(xxi));
36 phi=nan(size(xxi));

37 for c=1:numel(xi)
38 yi=yyi(g{c}); if ~iscolumn(yi), yi=yi(:);end
39 renewW=true;
40 if numel(yi)==numel(yiold);
41     if sum(yi-yiold)==0,renewW=false;end
42 end
43 if renewW,
44     W=interp1_w(y,yi); yiold=yi;
45 end
46 i1=W.i1; i2=W.i2;
47 alpha=W.alpha; beta=W.beta;
48 ac=alpha.*acp(i1,c)+beta.*acp(i2,c);
49 as=alpha.*asp(i1,c)+beta.*asp(i2,c);
50 p=atan2(as, ac);
51 phi(g{c})=p;
52 amp(g{c})=as./sin(p);
53 end

54 %% output
55 if isdeg
56     phi=phi*(180/pi); id=phi<0;
57     phi(id)=phi(id)+360; % so that phi is within the range[0 360);
58 end

```

List 5 Function THCEpinterp2 for performing 2-D interpolations of the tidal ellipse parameters

```

1  function EP=THCEpinterp2(XX, YY, EP, xi, yi, isdeg)
2  % Inputs
3  %     XX, YY: two matrices for rectangular grid points coordinates.
4  %     They should be created by meshgrid function.
5  %
6  %     EP:     a structure with fields for the tidal current tidal
7  %     ellipse parameters SEMA, ECC, INC, and PHA.
8  %     See ap2ep and ep2ap (Xu,2000; Xu,2002).
9  %
10 %     Xi, Yi, coordinates of the interpolation points. They can be
11 %     matrices for the rectangular grid points or vectors
12 %     for the irregular points.
13 %
14 %     isdeg,  an optional argument to indicate whether the phase and
15 %     inclination angles (PHA and INC) are input in degrees
16 %     or not (then in radians). Its default value is true.
17 %
18 % Outputs:
19 %     EP: a structure with fields for the tidal current tidal
20 %     ellipse parameters SEMA, ECC, INC, and PHA. See ap2ep
21 %     and ep2ap (Xu,2000; Xu,2002).
22 %
23 % Zhigang Xu, 2017/Jan/08
24 if nargin<6, isdeg=true; end

25 SEMA=EP.SEMA; ECC=EP.ECC; INC=EP.INC; PHA=EP.PHA;
26 [Au, PHIu, Av, PHIV]=ep2ap(SEMA, ECC, INC, PHA);

27 [Au, PHIu]=THCinterp2(XX, YY, Au, PHIu, xi, yi, isdeg);
28 [Av, PHIV]=THCinterp2(XX, YY, Av, PHIV, xi, yi, isdeg);

29 [SEMA, ECC, INC, PHA]=ap2ep(Au, PHIu, Av, PHIV);
30 EP.SEMA=SEMA; EP.ECC=ECC; EP.INC=INC; EP.PHA=PHA;
31
32 end % of the function
33

```

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Byun DS, Hart DE (2017) A robust interpolation for producing tidal current ellipse inputs to regional and coastal ocean numerical models. *Ocean Dyn* 67(3–4):451–463. <https://doi.org/10.1007/s10236-017-1037-4>
- Foreman MGG (1978) Manual for Tidal Currents Analysis and Prediction, Pacific Marine Science Report 78-6, Institute of Ocean Sciences, Patricia Bay, Sidney. British Columbia, 70
- Park SY, Byun DS, Hart DE (2012) A solution for the correct interpolation of data across regions with abrupt changes in tidal phase-lag. *Ocean Sci J* 47(3):215–219. <https://doi.org/10.1007/s12601-012-0022-6>
- Xu Z (2000) Ellipse parameters conversion and vertical velocity profiles for tidal currents. Bedford Institute of Oceanography, Fisheries and Oceans Canada. Canada, 20 p. The first edition. Available at https://www.researchgate.net/publication/312234150_Ellipse_Parameters_Conversion_and_Velocity_Profile_For_Tidal_Currents
- Xu 2002 Ellipse Parameters Conversion and Velocity Profiles for Tidal Currents in Matlab. Maurice Lamontagne Institute, Ocean Science Division, Fisheries and Oceans Canada. 2nd edn. Available at https://www.researchgate.net/publication/284367541_Ellipse_Parameters_Conversion_and_Velocity_Profiles_for_Tidal_Currents_in_Matlab