Ellipse Parameters Conversion and Velocity Profiles for Tidal Currents in Matlab

Zhigang Xu

Maurice-Lamontagne Institute

Ocean Science Division, Fisheries and Oceans Canada

850 route de la Mer, Mont-Joli, Quebec, Canada, G5H 3Z4

xuz@dfo-mpo.gc.ca

First Edition Nov. 2000

Revised Oct. 2002

Contents

Revision Preface

Given an ellipse, one may take one of the two opposite directions as the semi-major axis direction. Foreman (1977) took the one that lies in the angle range of $[0, 180)$, which he called as the northern axis, as the semi major axis direction. Foreman's northern major axis convention is adopted by this revision.

The conversion between the elliptical parameters and the amplitude and phase lag parameters is a purely geometrical problem. One needs not to be concerned with the so-called astronomical nodal effects in such conversion. However, when the parameters are used to predict the tides in the ocean one has to consider the nodal effects. How to charge/discharge the parameters with the nodal effects are therefore discussed in a new subsection (section 1.2).

Also, the geometrical explanation for the phase angle has been revised for clarity, and the title of this document has been slightly modified!

Preface

Conversion between the tidal current **a**mplitude and **p**hase lag parameters (for short, referred to as ap-parameters hereafter) and tidal current **e**lli**p**se parameters (referred to as ep-parameters hereafter) is not as trivial as the conversion between Cartesian and polar coordinates. We spend time to figure it how to do so at one time and then forget it in a few months later (or shorter, my e-folding memory scale is short, how long is yours?). I have just completed a tidal data assimilation project, in which I did tons of such conversions with a sketchy MATLAB program. Recently some my colleagues inquired on how to do the conversion. Given that such inquiries are heard from time to time, I decided to pull all the relevant material together for convenience. The rest of this document consists of two parts: a theory on the conversion and several MATLAB programs (requiring Matlab version 5 or higher).

Having gone through the rotary decomposition for tidal ellipse parameters, it would be a waste if I did not go a step further and show the decoupling of the linear tidal momentum equations, for the setting for the two cases is the same, and any one who is interested in tidal ellipse parameters is likely also interested in the tidal momentum equations. Having presented the decoupled momentum equations, I might go another step to present a solution for the equations. Thus, the main body of the document consists 1) theories for the ellipse conversions and decoupling momentum equations, and 2) programs for conversion between ap- and ep- parameters (ap2ep.m and ep2ap.m) and for calculation of vertical tidal current profiles (cBEpm.m) and the associated auxiliary programs. An example program (example.m) is also included to demonstrate how to use ap2ep.m and ep2ap.m.

This document and associated programs are a revision of my first release made a few weeks ago (you may regard that release as a beta-version if you already have down-loaded it). After that release, I received good response from Dr. Rich Signell of U.S. Geological Survey, who not only debugged the program but also gave me many good suggestions, especially on the designs of the notation for the minus rotary components. I enjoyed discussions with him and would like to express my sincere thanks to him. I also thank my colleague in BIO, Dr. Charles Hannah, for his proofreading this document.

One more bit before I get into the real business: in putting up the mathematics, I chose a way that will make readers feel effortless in reading most of the derivations while still manage to keep the document from being fat. This contrasts to some traditional treatments for expressing mathematics, where readers are expected to read with pencils and paper. If there is any one out there being offended by seeing too much details, please forgive me!

1 Theory

1.1 Tidal ellipse and rotary components

Given tidal currents of u- (east or x-) and v- (north or y-) components, as

$$u = a_u \cos(\omega t - \phi_u) \tag{1}$$

$$v = a_v \cos(\omega t - \phi_v) \tag{2}$$

where a_u and ϕ_u are the amplitudes and phase lags for the u-components and likewise for a_v and ϕ_v, and ω is the tidal angular frequency, we can form a complex tidal current w as

$$w = u + iv \tag{3}$$

where $i = \sqrt{-1}$. If we trace w on a complex plane as time goes by a period (T=$2\pi/\omega$), we will see an ellipse. Our interest here is not only in seeing the ellipse, but also in calculating the following ellipse parameters (see Fig. 1):

- **Semi-major axis** (referred to as **SEMA** hereafter where appropriate) or maximum current velocity. A northern semi-major axis (whose angle with respect to the x-direction lies in [0, 180), anti-clockwisely positive) will be always chosen as the semi-major axis (following Foreman 1977).

- **Eccentricity (ECC)**, the ratio of semi-minor to semi major axis, negative values indicating that the ellipse is traversed in a clockwise rotation;

- **Inclination (INC)**, or angle between east (x-) and semi-major axis;

- **Phase (PHA)**, angle that the two oppositely rotating circular components have to traverse from their initial positions for them to meet. When the two circular radial vectors meet the maximum current occurs, thus PHA is also closely related with the **maximum current time** (= PHA/frequency).

Let us continue from (3):

$$\begin{aligned} w &= u + iv \\ &= a_u \cos(\omega t - \phi_u) + i a_v \cos(\omega t - \phi_v) \end{aligned}$$

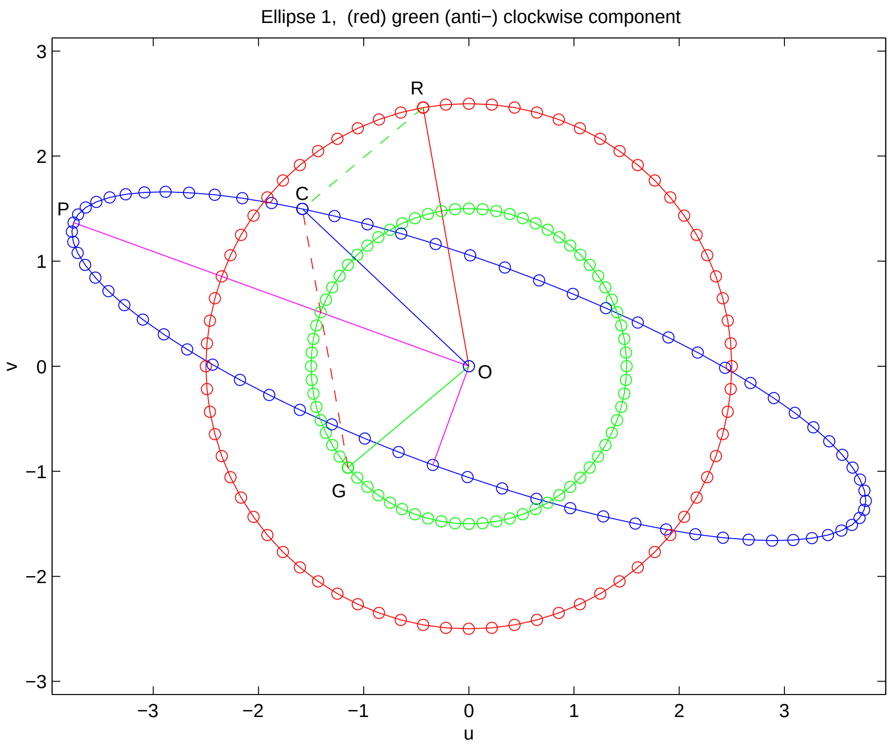

Figure 1: An ellipse can be constructed by two opposite rotating circular radial vectors (red: anti-clockwise, green: clockwise circle). The direction of the longer circular radial vector dictates the rotating direction of the elliptical radial vector (blue). As they are drawn, the two circular radial vectors are at their initial positions; half of the angle spanned by them is the so-called phase angle, (i.e., PHA=$\angle$ ROP=$\angle$ GOP). The initial current vector is indicated by the blue vector. The angle between the blue vector and the purple semi-major axis, (i.e., $\angle$ COP), should not be mistaken as the phase angle, although the time needed by the blue vector to reach the major axis is the same as the red (or green) vector does. This is because although the circular radial vectors rotate in the same angular speed, the elliptical radial vector does not, which can be clearly seen from the unevenly spaced small blue beads along its track.

$$
\begin{aligned}
&= a_u \frac{e^{i(\omega t-\phi_u)} + e^{-i(\omega t-\phi_u)}}{2} + i a_v \frac{e^{i(\omega t-\phi_v)} + e^{-i(\omega t-\phi_v)}}{2} \\
&= \frac{a_u e^{-i\phi_u} + i a_v e^{-i\phi_v}}{2} e^{i\omega t} + \frac{a_u e^{i\phi_u} + i a_v e^{i\phi_v}}{2} e^{-i\omega t} \qquad (4) \\
&= w_p e^{i\omega t} + w_m e^{-i\omega t} \qquad (5) \\
&\stackrel{or}{\equiv} W_p e^{i(\omega t+\theta_p)} + W_m e^{-i(\omega t-\theta_m)}. \qquad (6)
\end{aligned}
$$

In (4)-(3) we have introduced new notations:

$$
\begin{aligned}
w_p \equiv W_p e^{i\theta_p} &= \frac{\tilde{u} + i\tilde{v}}{2} \qquad (7) \\
w_m \equiv W_m e^{i\theta_m} &= \left(\frac{\tilde{u} - i\tilde{v}}{2}\right)^* \qquad (8) \\
\tilde{u} &= a_u e^{-i\phi_u} \qquad (9) \\
\tilde{v} &= a_v e^{-i\phi_v} \qquad (10)
\end{aligned}
$$

where (9) - (10) define $\tilde{u}$ and $\tilde{v}$ as u- and v- complex amplitudes respectively, the minus signs in front of their phase angles mean that positive algebraic values of ϕ_u and ϕ_v represent phase lags — a tidal convention, and notation '*' indicates a complex conjugate operator.

Thus, we have decomposed an ellipse into two circular components: the term with $e^{i\omega t}$ in (5) (or (6)) describes an anti-clockwise circle with a radius of W_p, and the term with $e^{-i\omega t}$ describes a clockwise circle with a radius of W_m (figure 1). Depending on whether W_p is greater than, equal to or less than W_m, the ellipse will traverse either anti-clockwise, rectilinear, or clockwise.

The following discussions assume that all the angles involved in the calculations are valued in the range of $[-\pi, \pi)$ (or equivalently $[-180, 180)$). Upon completion of the calculations, one may map the resultant angles, INC and PHA, to $[0, 2\pi)$, as is the case in **ap2ep.m** listed in section 2.

When the two circular radial vectors are aligned in the same direction, the tidal current will reach its maximum. From, (6), we can see that will happen when

$$
\omega t + \theta_p = -\omega t + \theta_m + 2k\pi \qquad (11)
$$

where integer $k = 0, \pm 1, \pm 2, \pm 3, \cdots$. Denote t_{max} as a t satisfying the above criterion, then the phase angle as introduced above is ωt_{max}, which is given by

$$
\text{PHA} = \omega t_{max} = \frac{\theta_m - \theta_p}{2} + k\pi. \qquad (12)
$$

It is sufficient to assign k with value of either 0 or 1. The two sensible k values correspond to the fact that the two oppositely rotating circular radial vectors will meet twice in one tidal period. Accordingly there are two directions along which the rotating circular radial vectors will be aligned successively and either of which can be equally well picked up as the direction for the semi major axis. Foreman (1977, p. 13, Manual For Tidal Currents Analysis Prediction, www.ios.bc.ca/ios/osap/people/foreman.htm) adopted a so-called northern axis convention, in which he always picks up the northern axis (whose angle lies in [0, 180)) as the semi-major axis. Note this convention may pick up the first or the second meeting time of the two rotating circular radial vectors as the maximum current time depending on the initial positions. For the sake of popularity of Foreman's tidal analysis package, his northern axis convention is adopted here.

Substitute (12) into (6), we can have a current vector whose length is maximum,

$$\begin{aligned} w_{max} &= W_p e^{i(\omega t_{max}+\theta_p)} + W_m e^{-i(\omega t_{max}-\theta_m)} \\ &= W_p e^{i\left(\frac{\theta_m+\theta_p}{2}+k\pi\right)} + W_m e^{i\left(\frac{\theta_m+\theta_p}{2}-k\pi\right)} \\ &= W_p e^{i\left(\frac{\theta_m+\theta_p}{2}+k\pi\right)} + W_m e^{i\left(\frac{\theta_m+\theta_p}{2}-k\pi+2k\pi\right)} \\ &= (W_p+W_m)e^{i\frac{\theta_m+\theta_p}{2}+k\pi} \end{aligned} \tag{13}$$

Thus, the maximum current, or semi-major axis (SEMA), is

$$\text{SEMA} = |w_{max}| = W_p + W_m \tag{14}$$

and its direction, or the inclination, is

$$\text{INC} = \arg(w_{max}) = \frac{\theta_m+\theta_p}{2} + k\pi. \tag{15}$$

According to Foremans' northern axis convention, k can fixed by using

$$k = \left[\frac{mod(\frac{\theta_m+\theta_p}{2}+2\pi, 2\pi)}{\pi}\right] \tag{16}$$

where $[\cdot]$ is an operator for taking integer part of the operand.

A geometrical interpretation of the phase formula, (12), and the inclination formula, (15), can be given as follows: the two oppositely rotating radial vectors are initially separated apart by $\theta_m - \theta_p$, as is shown by the green and red vectors in Fig. 1; half of this angle difference is the angle that each of the radial vectors has to rotate before they can meet at their middle point $(\theta_m + \theta_p)/2$ when the maximum current occurs. If $\theta_m >= \theta_p$, $k = 0$ corresponds to the first maximum current time and $k = 1$ the second maximum current time; if $\theta_m < \theta_p$, $k = 1$ and $k = 0$ correspond to the first and the second maximum current times respectively.

When the two circular radial vectors are aligned in opposite directions, i.e.,

$$\omega t + \theta_p \quad = \quad -\omega t + \theta_m + (2k+1)\pi \tag{17}$$

then the tidal current reaches minimum in its speed. At this time, $t = t_{min}$

$$\omega t_{min} \quad = \quad \frac{\theta_m - \theta_p}{2} + (k + \frac{1}{2})\pi \tag{18}$$

and

$$\begin{aligned} w_{min} &= W_p e^{i\left(\frac{\theta_m+\theta_p}{2}+(k+\frac{1}{2})\pi\right)} + W_m e^{i\left(\frac{\theta_m+\theta_p}{2}-(k+\frac{1}{2})\pi\right)} \\ &= W_p e^{i\left(\frac{\theta_m+\theta_p}{2}\right)} e^{i\frac{\pi}{2}} + W_m e^{i\left(\frac{\theta_m+\theta_p}{2}\right)} e^{-i\frac{\pi}{2}} \\ &= \left(W_p + W_m e^{-\pi}\right) e^{i\left(\frac{\theta_m+\theta_p}{2}+\frac{\pi}{2}\right)} \\ &= \left(W_p - W_m\right) e^{i\left(\frac{\theta_m+\theta_p}{2}+\frac{\pi}{2}\right)} \end{aligned} \tag{19}$$

therefore, the minimum speed of the tidal current, or semi-minor axis (SEMI) is

$$\text{SEMI} = |w_{min}| \quad = \quad W_p - W_m \tag{20}$$

and its angle is

$$\arg(w_{min}) \quad = \quad \frac{\theta_m + \theta_p}{2} + \frac{\pi}{2}. \tag{21}$$

Thus, the eccentricity, ECC, is

$$\text{ECC} \quad = \quad \frac{\text{SEMI}}{\text{SEMA}} = \frac{W_p - W_m}{W_p + W_m}. \tag{22}$$

When $W_m > W_p$, ECC is negative and the ellipse rotates clock-wisely.

The above is the conversion from ap-parameter to ep-parameter, and is used by the program **ap2ep.m** listed in section 2. Now consider the inverse: given the four ep-parameters of SEMA, ECC, INC, and PHA, how can we recover ap-parameters of a_u, ϕ_u, a_v and ϕ_v?

As a middle step, we need to recover W_p, θ_p, w_p, W_m, θ_m and w_m. From (14), (20) and (22), we can have

$$
\begin{aligned}
W_p &= \frac{1+\mathrm{ECC}}{2}\mathrm{SEMA} && (23)\\
W_m &= \frac{1-\mathrm{ECC}}{2}\mathrm{SEMA} && (24)
\end{aligned}
$$

and from (12) (when $k = 0$) and (15), we can have

$$
\begin{aligned}
\theta_p &= \mathrm{INC} - \mathrm{PHA} && (25)\\
\theta_m &= \mathrm{INC} + \mathrm{PHA}. && (26)
\end{aligned}
$$

Hence we can know

$$
\begin{aligned}
w_p &= W_p e^{-i\theta_p} && (27)\\
w_m &= W_m e^{i\theta_m}. && (28)
\end{aligned}
$$

We then can know further from (7) - (10) that

$$
\begin{aligned}
a_u e^{-i\phi_u} &= w_p + w_m^* && (29)\\
a_v e^{-i\phi_v} &= \frac{1}{i}\left(w_p - w_m^*\right) && (30)\\
&\stackrel{or}{=} -i\left(w_p - w_m^*\right) && (31)
\end{aligned}
$$

So,

$$
\begin{aligned}
a_u &= \left|w_p + w_m\right| \\
\phi_u &= -\arg\left(w_p + w_m\right) && (32)
\end{aligned}
$$

and,

$$
\begin{aligned}
a_v &= \left|\left(w_p - w_m^*\right)\right| && (33)\\
\phi_v &= -\arg\left(\frac{w_p - w_m^*}{i}\right) && (34)
\end{aligned}
$$

The program **ep2ap.m** listed in section 2 assumes this inverse conversion.

1.2 On the astronomical nodal correction

The above conversions are independent of astronomical nodal effects; the conversion between the ap- and ep- parameters has been treated as a pure geometric problem. This implies that the nodal corrections have been dealt with before you use this conversion package. For example, if you have a set of ep-parameters obtained by applying tidal analysis to original current vector time series, then the nodal effects will have been corrected in that tidal analysis package, and so the ep-parameters are then free of nodal effects. So long as your inputs to **ap2ep** or **ep2ap** are nodal effects free, the outputs will also be nodal effects free. And this is what we want! As the so-called tidal harmonic *constants* imply, we wish to record tidal parameters that are time invariant, whereas the astronomical nodal effects are functions of time.

In using the harmonic constants to predict the tidal currents, however, one has to consider the nodal effects. In this case, (1) and (2) need to be modified as (e.g., Foreman, 1977)

$$u = f a_u \cos(\omega t + U + V - \phi_u) \tag{35}$$

$$v = f a_v \cos(\omega t + U + V - \phi_v) \tag{36}$$

where f, U and V are functions of time to represent the so-called nodal effects and their notations are conventional (so be not confused by the meaning of U and u, and V and v). One may then want to ask what kind of modifications would be required by the conversion formulae if we had started with the (35) and (36). To answer this, let us redo the derivation

$$\begin{aligned} w &= u + iv \\ &= f a_u \cos(\omega t + U + V - \phi_u) + i f a_v \cos(\omega t + U + V - \phi_v) \\ &= e^{i(U+V)} f \frac{a_u e^{-i\phi_u} + i a_v e^{-i\phi_v}}{2} e^{i\omega t} + e^{-i(U+V)} f \frac{a_u e^{i\phi_u} + i a_v e^{i\phi_v}}{2} e^{-i\omega t} \\ &= f W_p e^{i(\omega t + \theta_p + U + V)} + f W_m e^{-i(\omega t - \theta_m + U + V)} \\ &= W_p' e^{i(\omega t + \theta_p')} + W_m' e^{-i(\omega t - \theta_m')} \end{aligned} \tag{37}$$

where

$$W_p' = f W_p \tag{38}$$

$$W_m' = f W_m \tag{39}$$

$$\theta_p' = \theta_p + U + V \tag{40}$$

$$\theta'_m = \theta_m - (U+V) \tag{41}$$

and W_p, W_m, θ_p and θ_m are defined the same as in (7) and (8). Compare (37) with (6), we see that they are of the same format. Thus the further derivations following (6) can be borrowed here to produce the final results:

$$\mathrm{SEMA'} = \mathrm{W'_p} + \mathrm{W'_m} = f(\mathrm{W_p} + \mathrm{W_m}) = f\mathrm{SEMA} \tag{42}$$

$$\mathrm{SEMI'} = \mathrm{W'_p} - \mathrm{W'_m} = f(\mathrm{W_p} - \mathrm{W_m}) = f\mathrm{SEMI} \tag{43}$$

$$\mathrm{ECC'} = \frac{\mathrm{SEMA'}}{\mathrm{SEMI'}} = \frac{\mathrm{SEMA}}{\mathrm{SEMI}} = \mathrm{ECC} \tag{44}$$

$$\mathrm{INC'} = \frac{\theta'_m + \theta'_p}{2} + k\pi = \frac{\theta_m + \theta_p}{2} + k\pi = \mathrm{INC} \tag{45}$$

$$\mathrm{PHA'} = \frac{\theta'_m - \theta'_p}{2} + k\pi = \frac{\theta_m - \theta_p}{2} + k\pi - (U+V) = \mathrm{PHA} - (U+V) \tag{46}$$

The quantities with primes contain the nodal effects, whereas those without are free of the nodal effects. The former will describe the actual tidal ellipse at a given time in oceans, whereas the latter cannot. But the latter is time invariantly valuable! What will actually happen inside of a tidal analysis program is that the primed quantities will be calculated first from a given time series and then the unprimed quantities are further calculated in the same line as the above formulae to produce the results free of the nodal effects. For prediction, the procedure needs to be reversed. The nodal effects need to be reassigned back to the unprimed quantities so that the actual ellipse in the ocean at the predicted time can be correctly described. The above formulae provides an easy tool for charging and discharging of the nodal effects. Note that the eccentricity, ECC, and the inclination, INC, are always independent of nodal effects.

Comparing (35) and (36) with (1) and (2), one can get a similar set of formulae for discharging and charging the nodal effects to the ap-parameters.

$$(a'_u,\ a'_v) = f(a_u,\ a_v) \tag{47}$$

$$(\phi'_u,\ \phi'_v) = (\phi_u,\ \phi_v) - (U+V). \tag{48}$$

By now you may have been wandering how you can calculate these nodal effect functions, f, U and V themselves? These are complicated astronomical functions but you are encouraged to look at t_vuf.m function in Pawlowicz tidal analysis package (available in www.sea-mat.whoi.edu). If you are a Fortran man, you will find a similar subroutine in Foreman's package. Figure 2 shows what these three time series, f(t), U(t), and V(t), look like for M2 tide at $45N$ latitude, which I calculated using Pawlowicz's *t_vuf.m*.

1.3 Decoupling of the linear tidal momentum equations

Consider

$$\frac{\partial u}{\partial t} - fv = -g\frac{\partial \eta}{\partial x} + \frac{\partial}{\partial z}\left(\nu\frac{\partial u}{\partial z}\right) \tag{49}$$

$$\frac{\partial v}{\partial t} + fu = -g\frac{\partial \eta}{\partial y} + \frac{\partial}{\partial z}\left(\nu\frac{\partial v}{\partial z}\right) \tag{50}$$

where all the variables are real and other notations are hopefully standard to you. By adding (49) and $i\times$ (50), and using w defined by (3), we can merge the above two equations into the following complex one,

$$\frac{\partial w}{\partial t} + ifw = -g \bigtriangledown \eta + \frac{\partial}{\partial z}\left(\nu\frac{\partial w}{\partial z}\right) \tag{51}$$

where

$$\bigtriangledown \equiv \frac{\partial}{\partial x} + i\frac{\partial}{\partial y}. \tag{52}$$

Assume u and v of the forms given in (1) and (2), and similarly for η, i.e.,

$$\eta = a_\eta \cos(\omega t - \phi_\eta) \tag{53}$$

which can be split into two circular parts as we did for u and v,

$$\begin{aligned} \eta &= a_\eta \cos(\omega t - \phi_\eta) \\ &= \frac{a_\eta e^{-i\phi_\eta}}{2} e^{i\omega t} + \frac{a_\eta e^{i\phi_\eta}}{2} e^{-i\omega t} \\ &= \eta_p e^{i\omega t} + \eta_m e^{-i\omega t}, \end{aligned} \tag{54}$$

where the tidal convention to define a complex variable is used again, i.e.,

$$\eta_p = \frac{a_\eta e^{-i\theta_\eta}}{2} \tag{55}$$

$$\eta_m = \frac{a_\eta e^{i\theta_\eta}}{2} \tag{56}$$

$$\overset{or}{=} \eta_p^*. \tag{57}$$

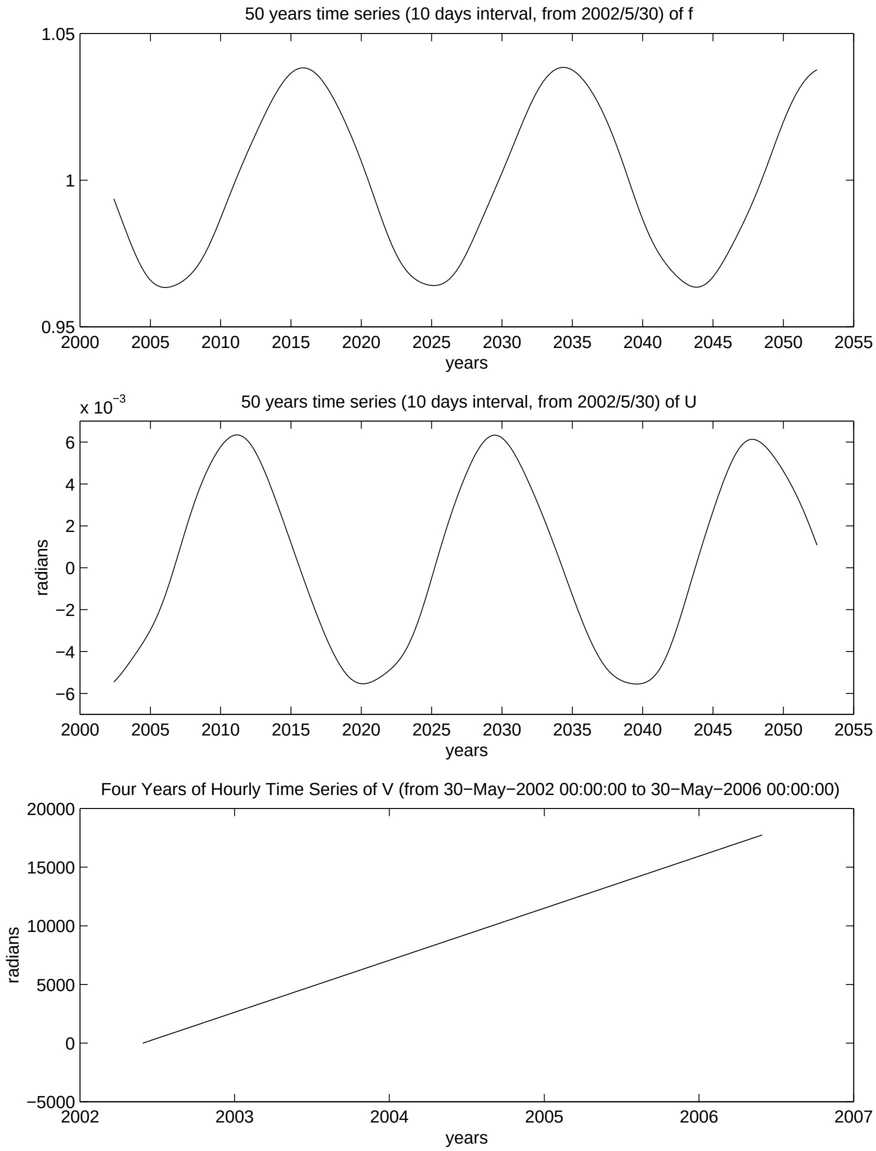

Figure 2: Three nodal effect functions, f(t), U(t) , and V(t), for M2 tide at 45° N latitude. Long term, somewhat between 18 to 19 years, periodicity is clearly seen in f and U curves.

Using the above equation and (5) we can rewrite (51) as

$$\left[i(f+\omega)w_p + g \bigtriangledown \tilde{\eta_p} - \frac{\partial}{\partial z}\left(\nu\frac{\partial w_p}{\partial z}\right)\right] e^{i\omega t} + \left[i(f-\omega)w_m + g \bigtriangledown \tilde{\eta_m} - \frac{\partial}{\partial z}\left(\nu\frac{\partial w_m}{\partial z}\right)\right] e^{-i\omega t} \quad = \quad 0 \quad (58)$$

Since $e^{i\omega t}$ and $e^{-i\omega t}$ are linearly independent of each other, for the above equation to hold, their coefficients must be zero, i.e.,

$$i(f+\omega)w_p \quad = \quad -g \bigtriangledown \eta_p + \frac{\partial}{\partial z}\left(\nu\frac{\partial w_p}{\partial z}\right), \quad (59)$$

$$i(f-\omega)w_m \quad = \quad -g \bigtriangledown \eta_m + \frac{\partial}{\partial z}\left(\nu\frac{\partial w_m}{\partial z}\right). \quad (60)$$

In the literature, you may find the following form of equation for w_m,

$$-i(f-\omega)w_m^* \quad = \quad -g \bigtriangledown^* \eta_p + \frac{\partial}{\partial z}\left(\nu\frac{\partial w_m^*}{\partial z}\right). \quad (61)$$

(where the conjugate signs on w_m may then be dropped) (60) and (61) are equivalent, being complex conjugate of each other.

1.4 Solutions to w_p and w_m when ν is depth invariant

Let ν be constant and subject the decoupled (59) and (60) to the following boundary conditions

$$\nu\frac{\partial(w_p, w_m)}{\partial z} \quad = \quad (0,0) \quad \text{at } z=0 \quad (62)$$

$$\nu\frac{\partial(w_p, w_m)}{\partial z} \quad = \quad \kappa(w_p, w_m) \quad \text{at } z=-h(x,y) \quad (63)$$

where κ is a bottom frictional parameter. The adoption of this parameter allows us to accommodate two types of bottom conditions: 1) slip conditions: $(w_p, w_m)_{z=-h} \neq (0,0)$, this is achieved by setting κ as a finite but non-zero number, in this case, if there is, and shall be, a certain size of bottom stress, there will be non-zero bottom velocities, which means flows are allowed to have motion relative to the "bottom" (in this case, the bottom is understood as the top of the bottom boundary log-layer); 2) non-slip bottom conditions, i.e., $(w_p, w_m)_{z=-h} = (0,0)$, in this case flow is not allowed to have motion relative to the bottom (the real sea bottom), this condition can be achieved by setting κ to be infinitely large.

The solutions to w_p and w_m can be found as

$$w_p \quad = \quad \mathrm{BE}_p \bigtriangledown \eta_p \quad (64)$$

$$w_m \quad = \quad \mathrm{BE}_m \bigtriangledown \eta_m \quad (65)$$

where the two-letter variables $\mathrm{BE_p}$ and $\mathrm{BE_m}$ stand for two **B**ottom **E**kman spirals in rotating components, they spiral near the bottom (just like wind driven surface Ekman spirals near the sea surface) and approach to geostrophic flow in the interior. Their details are expressed in the following:

$$\mathrm{BE_p}(x,y,z) = -\frac{g}{\nu\alpha^2}\left[1-\frac{\cosh\alpha z}{\cosh\alpha h+\frac{\alpha\nu}{\kappa}\sinh\alpha h}\right] \tag{66}$$

$$\mathrm{BE_m}(x,y,z) = -\frac{g}{\nu\beta^2}\left[1-\frac{\cosh\beta z}{\cosh\beta h+\frac{\beta\nu}{\kappa}\sinh\beta h}\right] \tag{67}$$

where

$$\alpha = \frac{1+i}{\delta_e}\sqrt{1+\frac{\sigma}{f}} \tag{68}$$

$$\beta = \frac{1+i}{\delta_e}\sqrt{1-\frac{\sigma}{f}} \tag{69}$$

$$\delta_e = \sqrt{\frac{2\nu}{f}} \qquad \text{(Ekman depth).} \tag{70}$$

The solution to (61) is

$$w_m^* = \mathrm{BE}_m^* \, \nabla^* \, \eta_p \tag{71}$$

where BE_m^* is the conjugate of BE_m given by (67). (Note the function cosh has a property of $(\cosh\alpha z)^* = \cosh(\alpha^* z)$).

As $f \to \sigma$, $\beta \to 0$ and $\mathrm{BE_m}$ becomes indefinite (0/0 type). This can be the situation for the northern hemisphere, as I have been encountered with recently in my Canadian Arctic archipelago M_2 tidal data assimilation project. Similar concern exist for $\mathrm{BE_p}$ for the southern hemisphere where $f+\sigma$ (hence α) can be zero. This type singularity may be referred as inertial frequency singularity. The singularity is not essential but we have to remove it before we can feed the formula into the computer, for otherwise the computer will generate garbage since there are only limited significant numbers used in a computer. This can be done by expanding (68) and (69) in power series:

$$\mathrm{BE_p}(x,y,z) = C\sum_{t=1}^{\infty}\left[\frac{1-(z/h)^{2t}}{2t}+\frac{\nu}{\kappa h}\right]\frac{(\alpha h)^{2t-2}}{(2t-1)!} \tag{72}$$

$$\mathrm{BE_m}(x,y,z) = D\sum_{t=1}^{\infty}\left[\frac{1-(z/h)^{2t}}{2t}+\frac{\nu}{\kappa h}\right]\frac{(\beta h)^{2t-2}}{(2t-1)!} \tag{73}$$

where

$$C = -\frac{gh^2}{\nu\,(1+\frac{\alpha\nu}{k}\tanh\alpha h)}\frac{2e^{-\alpha h}}{1+e^{-2\alpha h}} \tag{74}$$

$$D = -\frac{gh^2}{\nu\left(1+\frac{\beta\nu}{k}\tanh\beta h\right)}\frac{2e^{-\beta h}}{1+e^{-2\beta h}} \tag{75}$$

Assume the ocean depth, h, is always finite and let us now consider a case where $\alpha h \to 0$, which will arises when either or both i) $f \to -\sigma$ or/and ii) $\nu \to \infty$ happens. In this limiting case,

$$\lim_{\alpha\to 0} \mathrm{BE_p}(x,y,z) = \begin{cases} -\left[\frac{g}{\nu}\frac{h^2-z^2}{2}+\frac{gh}{\kappa}\right] & \text{when } \nu \text{ is finite} \\ -\frac{gh}{\kappa} & \text{when } \nu \text{ is infinite} \end{cases} \tag{76}$$

Likewise,

$$\lim_{\beta\to 0} \mathrm{BE_m}(x,y,z) = \begin{cases} -\left[\frac{g}{\nu}\frac{h^2-z^2}{2}+\frac{gh}{\kappa}\right] & \text{when } \nu \text{ is finite} \\ -\frac{gh}{\kappa} & \text{when } \nu \text{ is infinite} \end{cases} \tag{77}$$

Thus we have resolved the apparent inertial frequency singularity.

If use the ratio test, we will see that the radii of convergence of the two series are infinite. Hence for any significantly large values of αh and βh, the series of (72) and (73) will converge to the same values of its finite forms of (66) and (67). However we might use the finite forms themselves for the sake of convergence rate. A MATLAB program, called cBEpm, is listed in the following section for calculating $\mathrm{BE_p}$ and $\mathrm{BE_m}$. You will see in that program that when $|\alpha h|$ is less than 1, the series form is used, otherwise, the finite form is used. (Even when αh goes up to 10, experiments show the series converges fast enough.) Why is it named as cBEpm? This is because I have lived with the combination of (59) and (60) and have made a program called BEpm and used it everywhere for calculating BE_p and BE_m^*. So to me "c" in cBEpm means conjugate, but to you it may be interpreted as a reminder that BEp and BEm are complex valued. (You do not want to punish me by modifying many of my other programs, do you?).

2 Programs

Three programs are included here: ap2ep.m, which converts ap-parameters to ep-parameters, ep2ap.m which is the inverse of ap2ep.m, and cBEpm.m for calculating the bottom Ekman spirals (BE_p and BE_m). See the comments in the programs for more details.

2.1 ap2ep.m

```
function [SEMA,  ECC, INC, PHA, w]=ap2ep(Au, PHIu, Av, PHIv, plot_demo)
```

```
%
% Convert tidal amplitude and phase lag (ap-) parameters into tidal ellipse
% (ep-) parameters. Please refer to ep2app for its inverse function.
%
% Usage:
%
% [SEMA,  ECC, INC, PHA, w]=ap2ep(Au, PHIu, Av, PHIv, plot_demo)
%
% where:
%
%     Au, PHIu, Av, PHIv are the amplitudes and phase lags (in degrees) of
%     u- and v- tidal current components. They can be vectors or
%     matrices or multidimensional arrays.
%
%     plot_demo is an optional argument, when it is supplied as an array
%     of indices, say [i j k l], the program will plot an  ellipse
%     corresponding to Au(i,j, k, l), PHIu(i,j,k,l), Av(i,j,k,l), and
%     PHIv(i,j,k,l);
%
%     Any number of dimensions are allowed as long as your computer
%     resource can handle.
%
%     SEMA: Semi-major axes, or the maximum speed;
%     ECC:  Eccentricity, the ratio of semi-minor axis over
%           the semi-major axis; its negative value indicates that the ellipse
%           is traversed in clockwise direction.
%     INC:  Inclination, the angles (in degrees) between the semi-major
%           axes and u-axis.
%     PHA:  Phase angles, the time (in angles and in degrees) when the
%           tidal currents reach their maximum speeds,  (i.e.
%           PHA=omega*tmax).
%
%           These four ep-parameters will have the same dimensionality
%           (i.e., vectors, or matrices) as the input ap-parameters.
%
%     w:    Optional. If it is requested, it will be output as matrices
%           whose rows allow for plotting ellipses and whose columns are
%           for different ellipses corresponding columnwise to SEMA. For
%           example, plot(real(w(1,:)), imag(w(1,:))) will let you see
%           the first ellipse. You may need to use squeeze function when
%           w is a more than two dimensional array. See example.m.
%
% Document:   tidal_ellipse.ps
```

```
%
% Revisions: Mar. 2, 2002, by Zhigang Xu,  --- adopting Foreman's northern
% semi major axis convention.
%
%  For a given ellipse, its semi-major axis is undetermined by 180. If we borrow
% Foreman's terminology to call a semi major axis whose direction lies in a range of
% [0, 180) as the northern semi-major axis and otherwise as a southern semi major
% axis, one has freedom to pick up either northern or southern one as the semi major
% axis without affecting anything else. Foreman (1977) resolves the ambiguity by
% always taking the northern one as the semi-major axis. This revision is made to
% adopt Foreman's convention. Note the definition of the phase, PHA, is still
% defined as the angle between the initial current vector, but when converted into
% the maximum current time, it may not give the time when the maximum current first
% happens; it may give the second time that the current reaches the maximum
% (obviously, the 1st and 2nd maximum current times are half tidal period apart)
% depending on where the initial current vector happen to be and its rotating sense.


if nargin < 5
     plot_demo=0;  % by default, no plot for the ellipse
end



% Assume the input phase lags are in degrees and convert them in radians.
   PHIu = PHIu/180*pi;
   PHIv = PHIv/180*pi;

% Make complex amplitudes for u and v
   i = sqrt(-1);
   u = Au.*exp(-i*PHIu);
   v = Av.*exp(-i*PHIv);

% Calculate complex radius of anticlockwise and clockwise circles:
   wp = (u+i*v)/2;      % for anticlockwise circles
   wm = conj(u-i*v)/2;  % for clockwise circles
% and their amplitudes and angles
   Wp = abs(wp);
   Wm = abs(wm);
   THETAp = angle(wp);
   THETAm = angle(wm);

% calculate ep-parameters (ellipse parameters)
    SEMA = Wp+Wm;              % Semi  Major Axis, or maximum speed
```

```
   SEMI = Wp-Wm;              % Semin Minor Axis, or minimum speed
    ECC = SEMI./SEMA;          % Eccentricity

   PHA = (THETAm-THETAp)/2;   % Phase angle, the time (in angle) when
                              % the velocity reaches the maximum
   INC = (THETAm+THETAp)/2;   % Inclination, the angle between the
                              % semi major axis and x-axis (or u-axis).

   % convert to degrees for output
   PHA = PHA/pi*180;
   INC = INC/pi*180;
   THETAp = THETAp/pi*180;
   THETAm = THETAm/pi*180;

 %map the resultant angles to the range of [0, 360].
  PHA=mod(PHA+360, 360);
  INC=mod(INC+360, 360);

 % Mar. 2, 2002 Revision by Zhigang Xu    (REVISION_1)
 % Change the southern major axes to northern major axes to conform the tidal
 % analysis convention  (cf. Foreman, 1977, p. 13, Manual For Tidal Currents
 % Analysis Prediction, available in www.ios.bc.ca/ios/osap/people/foreman.htm)
   k = fix(INC/180);
   INC = INC-k*180;
   PHA = PHA+k*180;
   PHA = mod(PHA, 360);

% plot at the request
if nargout == 5
    ndot=36;
    dot=2*pi/ndot;
    ot=[0:dot:2*pi-dot];
    w=wp(:)*exp(i*ot)+wm(:)*exp(-i*ot);
    w=reshape(w, [size(Au) ndot]);
 end

if any(plot_demo)
   plot_ell(SEMA, ECC, INC, PHA, plot_demo)
end
```

2.2 ep2ap.m

```
function [Au, PHIu, Av, PHIv, w]=ep2ap(SEMA, ECC, INC, PHA, plot_demo)
%
% Convert tidal ellipse parameters into amplitude and phase lag parameters.
% Its inverse is ap2ep.m. Please refer to ap2ep for the meaning of the
% inputs and outputs.
%
% Zhigang Xu
% Oct. 20, 2000
%
% Document:  tidal_ellipse.ps
%
if nargin < 5
     plot_demo=0;  % by default, no plot for the ellipse
end

   Wp = (1+ECC)/2 .*SEMA;
   Wm = (1-ECC)/2 .*SEMA;
   THETAp = INC-PHA;
   THETAm = INC+PHA;

   %convert degrees into radians
   THETAp = THETAp/180*pi;
   THETAm = THETAm/180*pi;

   %Calculate wp and wm.
   wp = Wp.*exp(i*THETAp);
   wm = Wm.*exp(i*THETAm);

   if nargout == 5
      ndot=36;
      dot = 2*pi/ndot;
      ot = [0:dot:2*pi-dot];
      w = wp(:)*exp(i*ot)+wm(:)*exp(-i*ot);
      w=reshape(w, [size(wp) ndot]);
   end

   % Calculate cAu, cAv --- complex amplitude of u and v
   cAu = wp+conj(wm);
   cAv = -i*(wp-conj(wm));
   Au  = abs(cAu);
```

```
   Av  = abs(cAv);
   PHIu = -angle(cAu)*180/pi;
   PHIv = -angle(cAv)*180/pi;

   % flip angles in the range of [-180 0) to the range of [180 360).
   id = PHIu < 0; PHIu(id) = PHIu(id) + 360;
   id = PHIv < 0; PHIv(id) = PHIv(id) + 360;

  if any(plot_demo)
    plot_ell(SEMA,ECC,INC,PHA,plot_demo)
  end
```

2.3 plot_ell.m

```
function w=plot_ell(SEMA, ECC, INC, PHA, IND)
%
% An auxiliary function used in ap2ep and ep2ap for plotting tidal ellipse.
% The inputs, MA, ECC, INC and PHA are the output of ap2ep and IND is a
% vector for indices for plotting a particular ellipse, e.g., if IND=[2 3 1];
% the ellipse corresponding to the indices of (2,3,1) will be plotted.
%_____________________________

 Size_SEMA=size(SEMA);
  len_IND=length(IND);
  if IND
       cmd=['sub2ind(size_SEMA' ];
       if len_IND==1
     titletxt=['Ellipse '];
       else
     titletxt=['Ellipse ('];
       end

       for k=1:len_IND;
          cmd=[cmd ','num2str(IND(k))];
          if k<len_IND
          titletxt=[titletxt num2str(IND(k)) ','];
  elseif len_IND==1
                titletxt=[titletxt num2str(IND(k))];
          else
                titletxt=[titletxt num2str(IND(k)) ')'];
          end
       end
```

```
        cmd=['n=' cmd ');'];
        eval(cmd);

        figure(gcf)
        clf
        do_the_plot(SEMA(n), ECC(n), INC(n), PHA(n));
        title(titletxt);
    elseif len_IND
        msg1=['IND input contains zero element(s)!'];
        msg2=['No ellipse will be plotted.'];
        disp(msg1);
        disp(msg2);
    end

%----------------------------
%begin of plot subfunction
function w=do_the_plot(SEMA, ECC, INC, PHA)

    SEMI = SEMA.*ECC;
    Wp = (1+ECC)/2 .*SEMA;
    Wm = (1-ECC)/2 .*SEMA;
    THETAp = INC-PHA;
    THETAm = INC+PHA;

    %convert degrees into radians
    THETAp = THETAp/180*pi;
    THETAm = THETAm/180*pi;
    INC = INC/180*pi;
    PHA = PHA/180*pi;

    %Calculate wp and wm.
    wp = Wp.*exp(i*THETAp);
    wm = Wm.*exp(i*THETAm);

    dot = pi/18;
    ot = [0:dot:2*pi-dot];
    a = wp*exp(i*ot);
    b = wm*exp(-i*ot);
    w = a+b;

    wmax = SEMA*exp(i*INC);
    wmin = SEMI*exp(i*(INC+pi/2));
```

```
    plot(real(w), imag(w))
    axis('equal');
    hold on
    plot([0 real(wmax)], [0 imag(wmax)], 'm');
    plot([0 real(wmin)], [0 imag(wmin)], 'm');
    xlabel('u');
    ylabel('v');
    plot(real(a), imag(a), 'r');
    plot(real(b), imag(b), 'g');
    hnd_a=line([0 real(a(1))], [0 imag(a(1))], 'color', 'r', 'marker','o');
    hnd_b=line([0 real(b(1))], [0 imag(b(1))], 'color', 'g', 'marker','o');
    hnd_w=line([0 real(w(1))], [0 imag(w(1))], 'color', 'b', 'marker','o');
    plot(real(a(1)), imag(a(1)), 'ro');
    plot(real(b(1)), imag(b(1)), 'go');
    plot(real(w(1)), imag(w(1)), 'bo');
    hnd_ab=line(real([a(1) a(1)+b(1)]), imag([a(1) a(1)+b(1)]), ...
               'linestyle', '--', 'color', 'g');
    hnd_ba=line(real([b(1) a(1)+b(1)]), imag([b(1) a(1)+b(1)]), ...
               'linestyle', '--', 'color', 'r');

    for n=1:length(ot);
          set(hnd_a, 'xdata', [0 real(a(n))], 'ydata', [0 imag(a(n))]);
          set(hnd_b, 'xdata', [0 real(b(n))], 'ydata', [0 imag(b(n))]);
          set(hnd_w, 'xdata', [0 real(w(n))], 'ydata', [0 imag(w(n))]);
               hold on
          plot(real(a(n)), imag(a(n)), 'ro');
          plot(real(b(n)), imag(b(n)), 'go');
          plot(real(w(n)), imag(w(n)), 'bo');
          set(hnd_ab, 'xdata',real([a(n) a(n)+b(n)]), 'ydata', ...
          imag([a(n) a(n)+b(n)]))
          set(hnd_ba, 'xdata',real([b(n) a(n)+b(n)]), 'ydata', ...
          imag([b(n) a(n)+b(n)]))
    end

    hold off

%end of plot subfunction
%---------------------------
%
%
```

2.4 example.m

```
%demonstrate how to use ap2ep and ep2ap

Au=rand(4,3,2);         % so 4x3x2 multi-dimensional matrices are used for the
Av=rand(4,3,2);         % demonstration.
Phi_v=rand(4,3,2)*360;  % phase lags inputs are expected to be in degrees.
Phi_u=rand(4,3,2)*360;

figure(1)
clf
[SEMA ECC INC PHA w]=ap2ep(Au, Phi_u, Av, Phi_v, [2 3 1]);

figure(2)
clf
[rAu rPhi_u rAv rPhi_v rw]=ep2ap(SEMA, ECC, INC, PHA, [2 3 1]);

%check if ep2ap has recovered Au, Phi_u, Av, Phi_v
max(max(max(abs(rAu-Au))))              %  = 9.9920e-16
max(max(max(abs(rAv-Av))))              %  = 6.6613e-16
max(max(max(abs(rPhi_u-Phi_u))))        %  = 4.4764e-13
max(max(max(abs(rPhi_v-Phi_v))))        %  = 1.1369e-13
max(max(max(max(abs(w-rw)))))           %  = 1.3710e-15
% for the random realization I had, the differences are listed on the right
% hand of the above column. What are yours?

% The above example function calls have already plotted an ellipse for you.
% To plot an ellipse separately, you may do
%
figure(3)
clf
plot(real(squeeze(w(2,3,1,:))), imag(squeeze(w(2,3,1,:))));

%here squeeze is needed because w is a multiple dimensional array.
```

2.5 cBEpm.m

```
function [BEp, BEm]=cBEpm(g, f, sigma, nu, kappa, z, h)
%
%Evaluate the theoretical vertical profiles (or Bottom Ekman spiral profiles)
%of tidal currents in the two rotary directions driven by half-unit of sea
```

```
%surface gradients in the two directions respectively. Eddy viscosity is
%assumed as vertically invariant. See tidal_ellipse.ps for more details.
%
%
%inputs:
%
%        g,  the gravity acceleration,
%        f,  the Coriolis parameter,
%       nu,  the eddy viscosity
%     kappa, the bottom frictional coefficient
%       z,   the vertical coordinates, can be a vector but must be
%            within [0 -h];
%       h,   the water depth, must be positive.
%
%       Note: except for z, all other inputs must be scalars.
%
%outputs:
%
%   BEp and BEm, the same dimensions of z,  the outputs for the vertical
%                velocity profiles driven respectively by a unit of sea
%                surface slope in the positive rotation direction and negative
%                rotation direction for when the eddy viscosity is vertically
%                invariant. See the associated  document for more details.

if length(g)>1  | length(f)>1     | length(sigma)>1 | ...
   length(nu)>1 | length(kappa)>1 | length(h)>1
   error('inputs of g,f,sigma, nu, kappa, and h should be all scalars!')
end

if any(z/h>0) | any(z/h<-1)
   disp('z must be negative and must be within [0 -h]')
end

delta_e = sqrt(2*nu/f);  %Ekman depth
  alpha = (1+i)/delta_e*sqrt(1+sigma/f);
  beta  = (1+i)/delta_e*sqrt(1-sigma/f);

BEp = get_BE(g, alpha, h, z, nu, kappa);
BEm = get_BE(g, beta,  h, z,  nu, kappa);


%subfunction
%-----------------------------------------------
```

```
function   BE=get_BE(g, alpha, h, z, nu, kappa)

       z = z(:);
     z_h = z/h;
      ah = alpha*h;
      az = alpha*z;
     ah2 = ah*2;
   anu_k = alpha*nu/kappa;
   nu_kh = nu/(kappa*h);

    if abs(ah) < 1 %series solution
 T = 10;
 C = -g*h*h/(nu*(1+anu_k*tanh_v5_2(ah)))*2;
 A1 = (1-z_h.*z_h)/2+nu_kh;
 B1 = exp(-ah)/(1+exp(-ah2));
 B  = B1;
    series_sum=A1*B1;

 for t = 2:T;
             t2=2*t;
             A = (1-z_h.^t2)./t2+nu_kh;
             B = B*ah*ah/(t2-1)/(t2-2);
             series_sum = series_sum+A*B;
 end

  BE = C*series_sum;

     else      %finite solution

          c = -g*h*h/nu;
          denom=(exp(az-ah)+exp(-(az+ah)))./(1+exp(-2*ah));
                               % =cosh(az)/cosh(ah);
                               %but this a better way to evaluate it.
  numer=1+anu_k*tanh_v5_2(ah);
          BE = c*((1-denom/numer)/(ah*ah));
     end

%end of subfunction
%
%Note tanh_v5_2 is a copy of tanh from Matlab v5.2, which has worked well!
%It seems that Matlab v5.3 has some bug(s) in tanh function! It cannot deal
%with large argument. try z=7.7249e02*(1+i), tanh(z) and tanh_v5_2(z) to
%see the difference.
```