The equation for the Flather boundary condition

$$\overline{u} = \overline{u}^{ext} - \sqrt{\frac{g}{h}}\left(\zeta - \zeta^{ext}\right), \qquad (1)$$

is similar to the one given in the ROMS wiki, but using h instead of D to link to the notation of {{ cite(bib['blayo05']) }}, who state in their section 3.2 that the Flather condition can be obtained by the Sommerfeld condition for the surface elevation ζ (with surface gravity waves phase speed) and a one-dimensional approximation to the continuity equation. Consider a small-amplitude shallow water wave propagating from the interior through an eastern boundary to the exterior, i.e. the outward unit normal of the boundary is the unit vector in direction of the x-axis, and the phase speed is positive. Note that $c/h = \sqrt{g/h}$, where c is the phase speed. Eq. (25) of {{ cite(bib['blayo05']) }} is then

$$\frac{\partial}{\partial x}\left(\overline{u} - \sqrt{\frac{g}{h}}\zeta\right) = 0 \qquad (2),$$

Multiplying a finite difference representation of the preceeding equation by dx gives

$$\overline{u}_i = \overline{u}_{i+1} - \sqrt{\frac{g}{h}}\left(\zeta_{i+1} - \zeta_i\right) \qquad (3)$$

Dropping the index i and labelling $i+1$ as exterior (ext) yields

$$\overline{u} = \overline{u}^{ext} - \sqrt{\frac{g}{h}}\left(\zeta^{ext} - \zeta\right). \qquad (4)$$

At a western boundary, one may use the identical equation but re-write it (and re-interpret it, see below) as

$$\overline{u}_{i+1} = \overline{u}_i - \sqrt{\frac{g}{h}}\left(\zeta_i - \zeta_{i+1}\right). \qquad (5)$$

Labelling the point i the exterior and dropping the index $i+1$, as appropriate for a western boundary condition, results in the same form as (4). If the wave speed is still positive x-direction, however, the condition has an active forcing character, rather than a passive absorbing one. In other words, whether the form (4) lets outward propagating waves pass through the boundary, or force the interior with waves coming from the outside, depends on whether the phase speed points into the domain or to the outside. In a finite difference expression for a given boundary, the phase speed can be flipped by swapping the ζ^i and ζ^{i+1} terms (or the ext label) on the right hand side of the expression. For example, Eq. (1) could serve as an absorbing condition at a western boundary.

Regarding the implementation in ROMS, consider a radiation condition for a western boundary, without coriolis force, wind stress and atmospheric pressure correction.

The corresponding FORTRAN code block ("Western edge, Flather boundary condition" in the file *u2dbc_im.F*) is

```
          ubar(Istr,j,kout)=bry_val-                              &
     &                      Cx*(0.5_r8*(zeta(Istr-1,j,know)+      &
     &                                  zeta(Istr  ,j,know))-     &
     &                          BOUNDARY(ng)%zeta_west(j))
```

Note that the code block may not represent a spatial gradient in any way (provided that our intepretation of the staggered location of *BOUNDARY(ng)%zeta_west(j)* is correct, see below). Both the average of *zeta*, and the term *BOUNDARY(ng)%zeta_west(j)* are centered at *ubar* points (for the centering of the latter term, see e.g. routine *set_tides.F*). A derivation of the radiation condition without the use of spatial gradients is given e.g. in {{ cite(bib['flather75']) }}. Denoting an internally generated disturbance as η' and an externally generated disturbance as $\hat{\eta}$

$$\eta' := \eta - \hat{\eta}. \tag{6}$$

In the context of their article, $\hat{\eta}$ is an elevation associated with (quote) *the externally generated* [storm] *surge entering the model* (end quote). If we understand correctly, η in their article is the sea level anomaly, which is why we chose the symbol η instead of ζ, which in ROMS terminology is the full sea level measured from a geoid. In other words, the anomaly η in Eq. (6) is partitioned into two contributions, an internally and an externally generated disturbance. They introduce a condition that ensures outward propagation of the internally generated disturbance

$$hq' = A'\eta', \tag{7}$$

where $q' = q - \hat{q}$ is the associated outward going current across the boundary and A' is an appropriate admittance coefficient with dimensions of velocity. Note again that they interpret q' always as the *outward* flux. To close the problem, $\hat{q}$ must be specified. The option $\hat{q} = 0$ yields

$$hq = A'\eta' \tag{8}$$

Another option is to associate $\hat{q}$ with incoming signals. Quoting {{ cite(bib['flather75']) }}, (quote) *a radiation condition should also be applied to this incoming part of the motion* (end quote), where the *incoming* part is storm surge energy in the context of their paper. The terminology used here is interesting, since it associates the term "radiation condition" with an actively forcing boundary condition (not a passively absorbing one). They write the condition as

$$h\hat{q} = -\hat{A}\hat{\eta}, \tag{9}$$

where $\hat{A}$ is another admittance coefficient and the negative sign indicates that the energy propagates inward. Using (9) in (7) yields

$$hq = A'\eta' - \hat{A}\hat{\eta} \tag{10}$$

and setting $A' = \hat{A} = \sqrt{gh}$ yields

$$q = \sqrt{\frac{g}{h}}(\eta' - \hat{\eta}) \tag{11}$$

To associate Eq. (7) with the ROMS code block shown above, note again that q is the outward flow, and the code block is for a western boundary. Setting

$$q' = -(\overline{u} - \overline{u}^{ext}) \tag{12}$$

in Eq. (7) yields

$$\overline{u} = \overline{u}^{ext} - \sqrt{\frac{g}{h}}\eta', \tag{13}$$

leaving the difference between *zeta* at time *know* and the boundary value to be interpreted as η'. The term $\overline{u}^{ext}$ is computed in ROMS as follows (we assume zero coriolis force, wind stress and atmospheric pressure correction)

```
          bry_pgr=-g*(zeta(Istr,j,know)-                          &
     &                 BOUNDARY(ng)%zeta_west(j))*                &
     &              0.5_r8*GRID(ng)%pm(Istr,j)
          Cx=1.0_r8/SQRT(g*0.5_r8*(GRID(ng)%h(Istr-1,j)+          &
     &                         zeta(Istr-1,j,know)+               &
     &                         GRID(ng)%h(Istr  ,j)+              &
     &                         zeta(Istr  ,j,know)))
          cff2=GRID(ng)%om_u(Istr,j)*Cx
          bry_val=ubar(Istr+1,j,know)+                            &
     &            cff2*bry_pgr
```

Substituting the value for *bry_pgr*, the last line can be rewritten as

```
          bry_val=ubar(Istr+1,j,know)-                            &
     &            cff2*g*(zeta(Istr,j,know)-                      &
     &                 BOUNDARY(ng)%zeta_west(j))*                &
     &              0.5_r8*GRID(ng)%pm(Istr,j)
```

To get rid of the grid increment term *GRID(ng)%om_u(Istr,j)* and the inverse grid increment *GRID(ng)%pm(Istr,j)*, let's assume the ξ-axis of the grid points eastward and the spacing is equidistant. Then

$$GRID(ng)\%om\_u(Istr,j) * GRID(ng)\%pm(Istr,j) = 1. \quad (14)$$

The term *Cx* represents

$$Cx = \frac{1}{\sqrt{gD}}, \quad (15)$$

where D is the bottom depth, which we loosely interpret as h such that *g*Cx* represents $\sqrt{g/h}$. Rewriting the code block informally with these continous expressions,

```
          bry_val=ubar(Istr+1,j,know)-                            &
     &            sqrt(g/h)*(zeta(Istr,j,know)-                   &
     &                 BOUNDARY(ng)%zeta_west(j))*                &
     &              0.5_r8
```

The term *bry_val* seems to be centered at the "physical" western boundary of the grid, i.e. at u-points. This is supported by the first code block shown above, where *bry_val* is added to *ubar(Istr,j,kout)*. Hence the preceding code block looks similar to a finite difference representation of Eq. (3), i.e.

$$\overline{u}_i = \overline{u}_{i+1} - \sqrt{\frac{g}{h}}(\zeta_{i+1} - \zeta_i), \quad (16)$$

which, as noted above, has an active forcing character at western boundaries. In the ROMS user forum, the significance of the factor 0.5 in the preceding code block was discussed. Some users argued that the term *zeta(Istr,j,know)-BOUNDARY(ng)%zeta_west(j)* represents a differenc between two grid locations that are separated by *half* a grid increment. This interpretation is supported by a comment in the file *set_tides.F*:

```
!  If appropriate, load tidal forcing into boundary arrays.  The "zeta"
!  boundary arrays are important for the Flather or reduced physics
!  boundary conditions for 2D momentum. To avoid having two boundary
!  points for these arrays, the values of "zeta_west" and "zeta_east"
!  are averaged at u-points.  Similarly, the values of "zeta_south"
!  and "zeta_north" is averaged at v-points. Noticed that these
!  arrays are also used for the clamped conditions for free-surface.
!  This averaging is less important for that type ob boundary
!  conditions.
```

If you know which factor should best be applied, please let me know. Hernan pointed out that it's a matter of testing. Maybe I can test it in spring.