1) Updating zs based on bed tilting is deactivated

Edited part

flow_timestep.F90

```
 Metronome simulation
      if (par%metro_period > 0.d0) then

         par%metro_dzb=(-
par%metro_omega*par%metro_amp1*sin(par%metro_omega*par%t-
par%metro_phase1) &
            -
par%metro_omega2*par%metro_amp2*sin(par%metro_omega2*par%t-
par%metro_phase2)) *par%dt

s%zb=s%zb+par%metro_dzb*(0.5d0*par%metro_basinlength-(s%xz-
s%xz(1,1)))/(0.5d0*par%metro_basinlength)
         !
!s%zs(2:s%nx,:)=s%zs(2:s%nx,:)+par%metro_dzb*(0.5d0*par%metro_basinleng
th-(s%xz(2:s%nx,:)-s%xz(1,1)))/(0.5d0*par%metro_basinlength)

      endif
```

2) 'metro_omega2' is considered in defining the function of bed tilting

Tiding function is

$$z_b = a_1 \cos\left(\frac{2\pi t}{T} - \varphi_1\right) + a_2 \cos\left(\frac{2\pi f t}{T} - \varphi_2\right)$$

metro_period = T

metro_amp1 = a_1

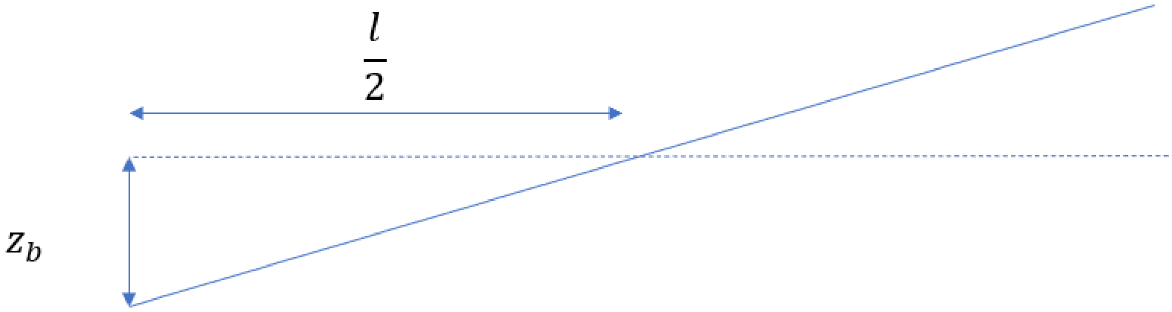
metro_amp2 = a_2

metro_phase1 = φ_1

metro_phase2 = φ_2

metro_basinlength = l

metro_period2 = f



Edited part

flow_timestep.F90

```
 par%metro_dzb=(-
par%metro_omega*par%metro_amp1*sin(par%metro_omega*par%t-
par%metro_phase1) &
          -
par%metro_omega2*par%metro_amp2*sin(par%metro_omega2*par%t-
par%metro_phase2)) *par%dt
```

3) ‘num_boundary_layer_water_level’ is used to imposed the boundary conditions (water level) directly on defined layers of cells near sea and river boundaries.

Edited part

flow_timestep.F90

```
 s%zs(1:par%num_boundary_layer_water_level,1:s%ny+1) = s%zs01
                !*****----> Saeb
     s%zs(s%nx-par%num_boundary_layer_water_level+1:s%nx+1,1:s%ny+1) =
s%zs02 !s%zs0(1:2,:)            !*****----> Saeb
```

4) ‘relaxtaion_parameter’ is added to consider relaxation for velocity

Edited part

flow_timestep.F90

```
do j=1,s%ny+1
         do i=1,s%nx+1
            !s%uu(i,j) = ((relaxtaion_parameter)*s%uu(i,j)) + ((1-
relaxtaion_parameter)*uu_old(i,j)) !***** Saeb
            s%uu(i,j) = ((par%relaxtaion_parameter)*s%uu(i,j)) + ((1-
par%relaxtaion_parameter)*uu_previous_time_step(i,j)) !***** Saeb
         end do
      end do
```

5) ‘dtset’ is added to limit maximum of time step

$\Delta t = \min(dtset, \Delta t_{CFL})$

Edited part

timesteps.F90

```
par%dt = min(par%dt, par%dtset)       !--->**** Saeb
```

6) ‘reduced_velocity_activation’ and ‘reduced_velocity_activation’ are considered to reduce the velocity near sea and river boundaries virtually.

```
if (par%reduced_velocity_activation  == 1) then
```

```
      s%uu(1:par%num_boundary_layer_water_level,1:s%ny+1)
=par%reduced_velocity_parameter_boundary_sea_river*
s%uu(1:par%num_boundary_layer_water_level,1:s%ny+1)                !*****---->
Saeb
      s%uu(s%nx-par%num_boundary_layer_water_level+1:s%nx+1,1:s%ny+1) =
par%reduced_velocity_parameter_boundary_sea_river * s%uu(s%nx-
par%num_boundary_layer_water_level+1:s%nx+1,1:s%ny+1)    !*****----> Saeb

      endif
```

7) required variables are defined
   params.def

8) updating continuity equation by considering bed movement

regarding the XBeach document

$$w(x,y,-d,t)=\frac{\partial d}{\partial t}-u\frac{\partial d}{\partial x}-v\frac{\partial d}{\partial y} \tag{1.8}$$

Because we assume that the timescales at which the bed changes are much larger than the timescales of the fluid motion the time derivative in (1.8) is neglected.

But for our simulation bed movement time scale is relatively as same order as fluid surface motion consequently it should be considered.

When the equations (1.7) and (1.8) are substituted into equation (1.6) and use is made of the Leibniz rule of integration the integrated continuity equation becomes

$$\frac{\partial \zeta}{\partial t}-\underline{\frac{\partial d}{\partial t}}+\frac{\partial UH}{\partial x}+\frac{\partial VH}{\partial y}=0 \tag{1.9}$$

```
! Update water level using continuity eq.

!
metro_dzb_continuity=(par%metro_omega*par%metro_amp1*sin(par%metro_omega*par%t-
par%metro_phase1) &
-par%metro_omega2*par%metro_amp2*sin(par%metro_omega2*par%t-par%metro_phase2))

 if (s%ny>0) then
   do j=jmin_zs,jmax_zs
     do i=imin_zs,imax_zs
      s%dzsdt(i,j) = (-1.d0)*( s%qx(i,j)*s%dnu(i,j)-s%qx(i1,j)*s%dnu(i-1,j)   &
                        + s%qy(i,j)*s%dsv(i,j)-s%qy(i,j-1)*s%dsv(i,j-1)
)*s%dsdnzi(i,j) &
                        - s%infil(i,j) + s%rainfallrate(i,j)

d_zb_dt(i,j)=metro_dzb_continuity*(0.5d0*par%metro_basinlength-(s%xz(i,j)-
s%xz(1,1)))/(0.5d0*par%metro_basinlength)     !*****----> Saeb

            end do
```

```
         end do
         s%zs(imin_zs:imax_zs,jmin_zs:jmax_zs) =
s%zs(imin_zs:imax_zs,jmin_zs:jmax_zs) &
            + (s%dzsdt(imin_zs:imax_zs,jmin_zs:jmax_zs)*par%dt) -
(d_zb_dt(imin_zs:imax_zs,jmin_zs:jmax_zs)*par%dt)

      else
         j=1
         do i=imin_zs,imax_zs
            s%dzsdt(i,j) = (-1.d0)*( s%qx(i,j)*s%dnu(i,j)-s%qx(i-1,j)*s%dnu(i-
1,j) )*s%dsdnzi(i,j) &
               - s%infil(i,j) + s%rainfallrate(i,j)

            d_zb_dt(i,j)=metro_dzb_continuty*(0.5d0*par%metro_basinlength-
(s%xz(i,j)-s%xz(1,1)))/(0.5d0*par%metro_basinlength)     !*****----> Saeb

         end do
         s%zs(imin_zs:imax_zs,1) =
s%zs(imin_zs:imax_zs,1)+(s%dzsdt(imin_zs:imax_zs,1)*par%dt) -
(d_zb_dt(imin_zs:imax_zs,1)*par%dt)
      endif !s%ny>0
```

User adjustments

By using the following keywords in *params.txt* file, we could activate or deactivate the corresponding modification

- *num_boundary_layer_water_level*, range 0 to 99

It defines the number of cell layers in sea and river side boundaries to impose water level directly.
If it sets 0, this option is deactivated.

- *reduced_velocity_parameter_boundary_sea_river*, range 0 to 1

$$u_{x\ sea/river} = \text{reduced\_velocity\_parameter\_boundary\_sea\_river} * u_{x\ sea/river}$$

virtually reduces the velocity at sea and river sides. It is an option to consider reservoir effect.
The keyword for activating this option is *reduced_velocity_activation*.
reduced_velocity_activation =0 deactivate and *reduced_velocity_activation* =1 activate this option.

- *relaxation_parameter*, range 0 to 1

It relaxes the velocity all over the geometry

$$u_x^{n+1} = (1 - \text{relaxation\_parameter}) * u_x^n + (\text{relaxation\_parameter}) * u_x^{n+1}$$

u_x^{n+1} is current time step velocity and u_x^n is the previous time step velocity

- *reduced_velocity_activation*, range (0 or 1)

activate or deactivate *reduced_velocity_parameter_boundary_sea_river*
reduced_velocity_activation =0 deactivate and *reduced_velocity_activation* =1 activate this option.

- *zs_updating_with_zb_activation*, range (0 or 1)

it activates or deactivates water level updating with bed level moving.

zs_updating_with_zb_activation = 1 consider updating water surface by bed movement (Dano approach), *zs_updating_with_zb_activation* = 0 ignore the mentioned updating.

- *vy_zero_boundary_sea_river_activation*, range (0 or 1)

vy_zero_boundary_sea_river_activation =1, y velocity direction zero at *num_boundary_layer_water_level*-th cells of sea and river side boundaries.

- *countinuty_with_Dzb_activation*, range (0 or 1)

countinuty_with_Dzb_activation =1 consider bed motion derivatives in time in continuity equation as

$$\frac{\partial \zeta}{\partial t} - \underline{\frac{\partial d}{\partial t}} + \frac{\partial UH}{\partial x} + \frac{\partial VH}{\partial y} = 0$$

countinuty_with_Dzb_activation =0 ignore bed motion derivatives in time in continuity equation

$$\frac{\partial \zeta}{\partial t} + \frac{\partial UH}{\partial x} + \frac{\partial VH}{\partial y} = 0$$

- *zero_gradient_u_activation,*

zero_gradient_u_activation =1 set zero gradient for velocity in x direction at sea and river boundaries. *zero_gradient_u_activation* = 0 deactivates this option.

$u_x\,(last\_row\_of\_cells,:) = \; u_x\,(last\_row\_of\_cells - 1,:)$

$u_x\,(first\_row\_of\_cells,:) = \; u_x\,(lfirst\_row\_of\_cells + 1,:)$

- *impicit_iteration_threshold* range (0,1,2,3,…)

number of iterations for correction-solution continuity and momentum equations inside each time step. It implements implicit simulation based on Gauss-Seidel method.

if *impicit_iteration_threshold* = 0 it converts to explicit approach.

- *Dirichlet_water_level_activation* range (0 or 1)

Dirichlet_water_level_activation =1, activate imposing the water level time series boundary conditions

- *water_level_loc_file* defines the location of boundary to impose water level boundary condition. It works as same as *disch_loc_file*
- *water_level_timeseries_file* defines time series of water level at boundaries. It works as same as *disch_timeseries_file*.
- *num_boundary_layer_water_level = 1.* Number of layers of mesh to impose water level boundary conditions. It should be set 1 at least to involve imposed boundary conditions in solving continuity and momentum equation.