Using the Sediment Transport and Vegetation Operators in ANUGA-Sed

Mariela Perignon

April 8, 2016

This document describes the implementation of operators for sediment transport and vegetation drag in ANUGA-Sed, a derivative of the hydrodynamic model ANUGA. The sediment transport operator can currently only be used with a single processor. Making it parallel safe would require modifying, at minimum, the algorithm for calculating sediment flux. The vegetation drag operator should be parallel safe but has not been tested.

These operators use the equations presented in *Simpson and Castelltort* [2006] and *Davy and Lague* [2009] for calculating sediment transport and momentum sinks, and *Nepf* [1999] and *Kean and Smith* [2006] for vegetation drag.

1 Files

The two operators are located in the directory `operators`, this documentation is in the directiory `doc`, and example files for using these two operators are in the directory `examples/operators/sed`.

2 Sediment Transport Operator

2.1 Creating the domain

In order to use the sediment transport operator, the quantity *concentration* must be defined as an evolved quantity when the domain is created. Exceptions are raised if this quantity does not exist when the operator is first called during the run.

The existing function for creating a rectangular domain accepts a list of evolved quantities as an argument:

```
evolved_quantities = ['stage', 'xmomentum', 'ymomentum', 'concentration']

points, vertices, boundary = rectangular_cross(int(length/dx),
                                               int(width/dy),
                                               len1=length,
                                               len2=width)

domain = Domain(points, vertices, boundary,
                evolved_quantities = evolved_quantities)
```

Creating a rectangular domain

Creating a domain that uses a mesh as input for *elevation* is usually done with the function `create_domain_from_regions`, which does not accept a list of evolved quantities. To accomplish the same functionality, use `create_mesh_from_regions` to generate the mesh and then create the domain:

```python
from anuga.pmesh.mesh_interface import create_mesh_from_regions

create_mesh_from_regions(bounding_polygon = bounding_polygon,
                         boundary_tags = boundary_tags,
                         maximum_triangle_area = 200,
                         filename = 'topo.msh')

evolved_quantities =  ['stage', 'xmomentum', 'ymomentum', 'concentration']

domain = Domain(filename_root + '.msh', evolved_quantities = evolved_quantities)
```

Creating a domain from a mesh file

Because it updates the elevation of the bed, the sediment transport operator can only be used with one of ANUGA's discontinuos elevation ('DE') flow algorithms. Changes in the elevation of the bed and the distribution of sediment in transport during the simulation can be recorded in the SWW file by including `elevation` and `concentration` in `domain.set_quantities_to_be_stored` with a value of 2 (to save them at every yieldstep):

```python
domain.set_flow_algorithm('DE0')

domain.set_quantities_to_be_stored({'elevation': 2,
                                    'stage': 2,
                                    'xmomentum': 2,
                                    'ymomentum': 2,
                                    'concentration': 2})
```

Setting the flow algorithm and recording the output

2.2 Initializing the operator

The sediment transport operator is initialized in the run file with:

```python
from anuga.operators.sed_transport_operator import Sed_transport_operator

sed_op = Sed_transport_operator(domain)
```

Initializing the sediment transport operator

Variable name	Default value	Units	Quantity
`rho_w`	1000	kg m^{-3}	Density of fluid
`rho_s`	2650	kg m^{-3}	Density of sediment
`nu`	$1\text{x}10^{-6}$	m^2 s^{-1}	Kinematic viscosity of water
`porosity`	0.3	-	Porosity of the bed
`criticalshear_star`	0.06	-	Dimensionless critical shear stress
`grain_size`	0.00013	m	Grain size

Table 1: Default values for equation parameters in Sediment transport operator

Table 1 summarizes the default values that various equation parameters take when `Sed_transport_operator` is initialized. These values can be modified from the run file after the operator is initialized.

The initial concentration of sediment in the flow within the domain can be set at any point after the domain is created with `domain.set_quantity()` (as a fraction of the volume of the water column). The concentration of flow entering the domain through any Dirichlet boundary can be set after initializing the operator as the operator variable `inflow_concentration`. If not set, `inflow_concentration` takes the maximum value of the quantity `concentration` at the first timestep. Currently, the value of `inflow_concentration` applies to all Dirichlet boundaries in the domain.

```
domain.set_quantity('concentration', 0.01) # 1 percent

sed_op.inflow_concentration = 0.02 # 2 percent

sed_op.grain_size = 0.0002 # 0.2 mm grains
sed_op.criticalshear_star = 0.03
```

Modifying parameter values

2.3 Mathematical Background and Assumptions

The sediment transport operator acts only on cells with a flow depth greater than a minimum depth (5 cm) and non-zero x-directed momentum. This avoids extremely high erosion rates due to abnormally high velocities at low flow depths and minimizes the likelihood of aggradation that exceed the water depth.

The strong spatial variability in shear stress caused by complex flow patterns requires the calculation of an explicit sediment mass balance within the water column in order to handle the local disequilibria between grain entrainment and settling rates [*Davy and Lague*, 2009]. The mass balance for sediment in the water column and moving bed layer is written as

$$\frac{\partial Ch}{\partial t} = \dot{E} - \dot{D} - \left(\frac{\partial q_{s_x}}{\partial x} + \frac{\partial q_{s_y}}{\partial y} \right) \tag{1}$$

where $\dot{E}$ is the entrainment flux, $\dot{D}$ is the deposition flux, C is sediment concentration (as a fraction of the volume of the water column), and q_{s_x} and q_{s_y} are specific volumetric sediment fluxes in the x and y directions [e.g., *Davy and Lague*, 2009]. Sediment flux is given by $q_{s_{x,y}} = \beta C q_{x,y}$, where $q_{x,y}$ is specific discharge in the x or y direction and β (≤ 1) describes the effective speed of sediment relative to that of the water.

The rate of change of elevation of the bed is a mass balance between material entrained and deposited:

$$\frac{\partial z}{\partial t} = \frac{\dot{D} - \dot{E}}{1 - \phi} \tag{2}$$

where z is the elevation of the bed above some datum, and ϕ is the porosity of the bed material.

2.3.1 Entrainment

:

The entrainment flux $\dot{E}$ is given by [e.g., **?**]:

$$\dot{E} = K_e \left(\tau - \tau_c \right) \tag{3}$$

where K_e is an erodibility parameter and τ_c is the critical shear stress for particle entrainment. The erodibility parameter K_e can be written as [**?**]:

$$K_e = \frac{0.2x10^{-6}}{\tau_c^{0.5}} \tag{4}$$

The critical shear stress τ_c can be found from the dimensionless critical shear stress τ_c^* through:

$$\tau_c^* = \frac{\tau_c}{(\rho_s - \rho_w) g D_{50}} \tag{5}$$

where ρ_w and ρ_s are the density of the fluid and sediment, g is acceleration due to gravity and D_{50} is the median grain size of the bed material.

τ_b is the shear stress that the flow applies on the bed:

$$\tau_b = \rho_w {u_*}^2 \tag{6}$$

where u_* is the shear velocity. Shear stress τ can also be written as $\tau_b = \rho g h S$, where S is the local slope of the bed. An equation for shear velocity u_* can then be obtained:

$$u_* = \sqrt{gSh} \tag{7}$$

2.3.2 Aggradation

:

Following *Davy and Lague* [2009], the rate of aggradation $\dot{D}$ can be written as:

$$\dot{D} = d^* \, C \, v_s \tag{8}$$

where $d^* = C^*/C$ is a dimensionless number that describes the vertical distribution of sediment in the water column, C^* is the sediment concentration at the bed bed interface and C is the average sediment concentration in the water column. The value of d^* in natural rivers will be between 1 and 3 for Rouse numbers smaller than 0.1, and close to 1 for large rivers or small particles. In small river with large particles, where much of the entrainment mechanism is bed load, d_* will be much larger than 1. Our method for finding d^* is described below.

The term v_s is the settling velocity of grains in the fluid [*Ferguson and Church*, 2004]:

$$v_s = \frac{R \, g \, {D_{50}}^2}{C_1 \, \nu + (0.75 \, C_2 \, R \, g \, {D_{50}}^3)^2} \tag{9}$$

where C_1 and C_2 are constants related to the shape and roughness of the particles and ν is the kinematic viscosity of water.

Selecting a value for d^* Calculating d^* for a natural river requires knowing both the velocity profile and sediment concentration profile in the flow. *Davy and Lague* [2009] derive an expression for d^* from the Rouse-Vanoni profile and the assumption that the sediment discharge of the flow q_s is the integral of the concentration and flow velocity with depth:

$$\begin{aligned} d^* &= \frac{C_s(a)}{C_s} = C_s(a)\frac{q}{q_s} \\ &= \frac{\int_a^h u(z)dz}{\int_a^h (\frac{z-a}{h-a}\frac{a}{z})^Z u(z)dz} \\ &= \frac{\int_a^h \ln(\frac{z}{z_o})dz}{\int_a^h (\frac{z-a}{h-a}\frac{a}{z})^Z \ln(\frac{z}{z_o})dz} \end{aligned} \tag{10}$$

where Z is the Rouse number, given by

$$Z = \frac{v_s}{\kappa u_*} \tag{11}$$

We solve this expression using numerical integration to find the value of d^* at every cell, assuming $z_o = D_{50}/30$ and a flow depth of 1 meter. For computational speed, values of d^* for all cells in the domain are only calculated once every 10 timesteps.

3 Vegetation

3.1 Initializing the operator

The vegetation operator is initialized in the run file with:

```python
from anuga.operators.vegetation_operator import Vegetation_operator

veg_op = Vegetation_operator(domain)
```

Initializing the vegetation operator

The vegetation operator uses quantities `veg_diameter` and `veg_spacing` for calculating the drag that vegetation applies on the flow. These quantities can be created at any point before the run starts by writing:

```python
Quantity(domain, name='veg_diameter', register=True)
domain.set_quantity('veg_diameter', 0.00064) # meters

Quantity(domain, name='veg_spacing', register=True)
domain.set_quantity('veg_spacing', 0.15) # meters
```

Creating vegetation quantities

The argument `register=True` is required for the quantities to be available to the operators.

3.2 Mathematical Background and Assumptions

The drag that vegetation imparts on the flow can dramatically lower the flow velocity and reduce the boundary shear stress, limiting erosion of the surface and potentially causing particles in transport to settle from suspension (Li and Shen, 1973; Pasche and Rouve, 1985; L?opez and Garc??a, 1998; Jordanova and James, 2003). A frequently used method for modeling vegetation uses a modified Manning?s equation to account for the increased roughness seen by the flow (e.g. Guardo and Tomasello, 1995). This approach does not consider the drag imparted by vegetation on the body of the flow and the quantifiable differences between various vegetation types (e.g. Kadlec, 1990). To counter these problems, other approaches treat vegetation as objects, most often cylinders, that flow must go around (e.g. Burke and Stolzenbach, 1983; Nepf , 1999).

We follow a similar approach by Kean and Smith (2004) to calculate the drag force FD that vegetation imparts on the flow:

$$F_D = \frac{1}{2}\rho\overline{C_D}\alpha {U_{ref}}^2 \tag{12}$$

where ρ is the density of the fluid, U_{ref} is the flow velocity in the absence of vegetation, $\alpha = d_s/\lambda^2$ is the projected plant area per unit volume, d_s is the stem diameter, λ is the mean stem spacing and $\overline{C_D}$ is the bulk drag coefficient for the vegetation array. If vegetation is approximated as a regular array of emergent cylindrical stems, $\overline{C_D} = 1.2$. A method for calculation the value of $\overline{C_D}$ as a function of ad is described below.

The drag force is allowed to reduce the flow velocity to zero but not to change its direction. The velocity of the flow is reduced by the drag force using the following relationship:

$$U = U_{ref} - F_D\Delta t \tag{13}$$

where Δt is the model timestep.

3.3 Drag coefficient

Following Nepf (1999), we expand the cylinder-drag vegetation model by defining the effects of stem population density on drag coefficient. Nepf (1999) presents a force balance that is used to solve for the bulk drag coefficient $\overline{C_D}$:

$$(1 - ad)C_B U^2 + \frac{1}{2}\overline{C_D}ad(\frac{h}{d})U^2 = gh\frac{\partial h}{\partial x} \tag{14}$$

where C_B is the bed drag coefficient [e.g. Munson et al., 1990, p. 673]. The term $\frac{\partial h}{\partial x}$ depends on ad. We selected the value of this term at multiple points in the range of valid values of ad such that they matched the values of $\overline{C_D}$ in Figure 6 of Nepf (1999) and fit a cubic equation to this term. Solving the force balance above with this fit and quantities listed for Nepf (1999) experiments, we found a relationship between $\overline{C_D}$ and ad that matches the curve in Figure 6 of Nepf (1999):

$$\overline{C_D} = \begin{cases} 1.2 & \text{if } ad \leq 0.006 \\ 56.11\ (ad)^2 - 15.28\ ad + 1.3 - 5.465e - 4\ (ad)^{-1} & \text{if } ad > 0.006 \end{cases}$$

The value of the drag coefficient $\overline{C_D}$ is calculated for every cell in the domain when the quantities for stem spacing and stem diameter are set.

3.4 Turbulence

By default, ANUGA assumed that fluid is inviscid and does not include kinematic viscosity. We incorporate kinematic viscosity only for vegetated systems using the formulations described here. We did not include this term for unvegetated channels with the assumption that the contribution was minimal.

"In addition to affecting the mean velocity, vegetation also affects the turbulent intensity and the diffusion. The conversion of mean kinetic energy to turbulent kinetic energy wihin stem wakes augments the turbulence intensity, and because wake turbulence is generated at the stem scale, the dominant turbulent lengthscale is shifted downward relative to unvegetated, open-channel conditions [Nepf 1997]". The formulation of Nepf (1999) for turbulent diffusivity within vegetated flows also incorporates a term for mechanical diffusivity, caused by the routing of parcels of flow along different paths between stems. We start by calculating a turbulence intensity that is given by the balance of the work input for wake production and the viscous dissipation rate. The work input for wake production is given by:

$$Pw = \frac{1}{2}\overline{C_D}\alpha U_{ref}^2 \tag{15}$$

The viscous dissipation rate is:

$$\varepsilon\ k^{3/2} d_s^{-1} \tag{16}$$

where k is the turbulence intensity. The balance of these two equations gives:

$$\frac{\sqrt{k}}{U_{ref}} = a\left[\overline{C_D}ad\right]^{1/3} \tag{17}$$

where a is a coefficient ($a \sim 1$). The turbulence intensity increases with bulk drag coefficient and with population density. Because the bulk drag coefficient $\overline{C_D}$ is a function of ad, the turbulence intensity can be considered to be only a function of population density.

The turbulent kinetic energy within the stems can be written as:

$$k = ((1 - ad)C_B + (\overline{C_D}ad)^{2/3}U^2 \tag{18}$$

The total diffusivity D for a vegetated channel, including both turbulent and mechanical diffusion, can then be written as:

$$D \sim k^{1/2}\ell + [ad]Ud \tag{19}$$

where ℓ is the mixing length scale. The mixing length scale is assumed to be equal to the flow depth in unvegetated channels ($ad = 0$). "If only sparse vegetation is present in the flow, that is, vegetation with spacing $\Delta S > h$, where ΔS is the stem spacing, the channel-scale eddies persist and continue to dminate the diffusive transport. Thus for sparse vegetation the dominant mixing length is still equal to the flow depth. For dense populations $\Delta S < h$, the stems break apart channel scale eddies, reducing the mixing-length scale, until at $ad > 0.01$, $\ell \sim d$ [Nepf et al., 1997]. This model assumes that ℓ varies linearly from h to d between these limits."

The model for mixing length looks like:

$$\ell = \begin{cases} h & \text{if } \Delta S \geq h \\ K_\ell ad - K_\ell\big(d/h\big)^2 + h & \text{if } \Delta S > h \\ d & \text{if } ad \geq 0.01 \end{cases}$$

where K_ℓ is:

$$K_\ell = \frac{d - h}{0.01 - (d/h)^2} \tag{20}$$

References

Davy, P., and D. Lague (2009), Fluvial erosion/transport equation of landscape evolution models revisited, *Journal of Geophysical Research-Earth Surface*, *114* (F3), F03,007, doi:10.1029/2008JF001146.

Ferguson, R., and M. Church (2004), A simple universal equation for grain settling velocity, *Journal of Sedimentary Research*, *74* (6), 933–937, doi:10.1306/051204740933.

Kean, J., and J. Smith (2006), Form drag in rivers due to small-scale natural topographic features: 2. Irregular sequences, *Journal of Geophysical Research*, *111* (F4), F04,010, doi:10.1029/2006JF000490.

Nepf, H. (1999), Drag, turbulence, and diffusion in flow through emergent vegetation, *Water Resources Research*, *35* (2), 479–489, doi:10.1029/1998WR900069.

Simpson, G., and S. Castelltort (2006), Coupled model of surface water flow, sediment transport and morphological evolution, *Computers & Geosciences*, *32*, 1600–1614.