How to Set Up Boundary Conditions with Triangle and BreZo

Brett F. Sanders

May 23, 2007
August 26, 2008 (revised for BreZo 4.0)

There are two aspects to setting up boundary conditions for BreZo. The first is the location of either wall or inflow/outflow boundaries, and the second is the constraint imposed on the solution at the boundary. We'll address these one at a time.

When a mesh is generated by triangle, the required input is a Planar Straight Line Graph (PSLG), a series of x-y points that encircle the domain of interest as well as information about how the points are connected to form a loop. The PSLG is saved as a .poly file, which is read by Triangle to guide mesh generation. Below is a simple PSLG for a rectangular channel 1 km long and 100 m wide.

Filename: channel.poly

```
4 2 0 0
1 0.0 0.0
2 1000.0 0.0
3 1000.0 100.0
4  0.0 1000.0
4 0
1 1 2
2 2 3
3 3 4
4 4 1
0
```

The first line tells Triangle there will be 4 points. The next four lines list the point number and the x-y values of each point, the next line tells Triangle the number of segments (4) and the number of boundary markers (0), the next four lines tell Triangle the points connected by each segment, an the last line tells Triangle the number of holes in the mesh.

So hopefully it's clear that the above information describes a rectangle: 4 points and 4 sides.

Now, I were to execute Triangle and make a mesh, I would write the following at the command prompt:

```
c:\> triangle –p –q30 –a100 –n –e channel
```

The following mesh files would be produced including:

channel.1.node
channel.1.ele

channel.1.neigh
channel.1.edge

The edge file is important for boundary conditions. It lists every edge in the mesh, the two nodes that it connects, as well as a boundary marker. What we want to do control the value of the boundary marker, which in turn will control the boundary conditions.

By default, the value of the marker in the .edge file is set to 0 if the edge falls on the interior of the mesh, and 1 if the edge is on the boundary. In addition, BreZo is coded to treat all edges flagged with a "1" as free-slip walls.

So if we were to run a simulation with the "channel.1" grid above, we would essentially be working with a tank of water 1 km long and 100 m wide, with walls on all four sides. That's not very interesting or useful, except perhaps for some idealized dam-break test problems.

Below is a revised version of the channel.poly file that would allow for inflow and outflow boundary conditions:

Filename: channel.poly

```
4 2 0 0
1 0.0 0.0
2 1000.0 0.0
3 1000.0 100.0
4  0.0 1000.0
4 1
1 1 2 1
2 2 3 3
3 3 4 1
4 4 1 2
0
```

In this case, I've told Triangle that there is one boundary marker (line 6, "1" following "4"), I've assigned a unique boundary marker "2" and "3" for the inflow and outflow boundaries, respectively, and I've used "1" for all wall boundaries.

Now, I need to execute Triangle again with the same command as before,

```
c:\> triangle –p –q30 –a100 –n –e channel
```

And this will create necessary mesh files to run BreZo. End of Step 1.

Step 2 is to tell BreZo the boundary condition for each of the inflow/outflow boundaries. We essentially need to tell BreZo what to enforce on edges flagged with "2", and what to enforce on edges flagged with "3".

Below is a sample channel.1.bc file:

```
n_bound !number of boundary conditions enforced on domain
2
bc_id  !repeat starting here
2
type_bc !1:spec eta (nr) 2=spec eta (r), 3=spec Q (m^3/s +ve=inflow),4=dry, 5=soft
3
opt_bc !0=constant, 1=harmonic, 2=read from file
0
bc_file_name !use if opt_bc=2; otherwise, leave dummy name
LaJolla.dat                          !Note: record should be less than 50000 rows
datumoffset
0.0
ls1_bc,ls2_bc,ts1_bc,ts2_bc, Qs_bc (ls in meters, ts in hours, Qs in m3/s)
1.0d0  0.0d0   12.d0   3.0d0    1.0d2
c_bc  !j=1,nspec, list out horizontally. If scalar_options=0, leave a single float.
0.0d0
bc_id  !repeat starting here
3
type_bc !1:spec eta (nr) 2=spec eta (r), 3=spec Q (m^3/s +ve=inflow),4=dry, 5=soft
1
opt_bc !0=constant, 1=harmonic, 2=read from file
0
bc_file_name !use if opt_bc=2; else, use dummy name
LaJolla.dat                          !Note: record should be less than 50000 rows
datumoffset
0.0
ls1_bc,ls2_bc,ts1_bc,ts2_bc, Qs_bc (ls in meters, ts in hours, Qs in m3/s)
1.0d0  0.0d0   12.d0   3.0d0    1.0d2
c_bc  !j=1,nspec, list out horizontally. If scalar_options=0, leave a single float
0.0d0
```

The file tells BreZo to enforce a “specify discharge” boundary (type_bc = 3) for edges with a boundary marker of 2 (bc_id=2); and to enforce a “specify water level” boundary (type_bc=1) for edges with a boundary marker of 3 (bc_id=3). The numbers listed below “ls1_bc, ls2_bc, …” represent the boundary values used when the constant boundary condition option is selected (opt_bc=0). “ls” stands for length scale, “ts” stands for time scale and “Qs” stands for discharge scale. In this case, the value of the inflow discharge will be 100 m^3/s (Qs_bc for bc_id=2), and the downstream water level will be set to 1 m (ls1_bc for bc_id=3). If the problem involved supercritical flow, then this file would also tell BreZo to enforce an upstream water level of 1 m (ls1_bc for bc_id=2). In addition, brezo would not enforce an open downstream boundary condition if flow were supercritical. Note that these levels are heights above the datum, not depths.

Inflow boundaries can be very tricky to model because the desired discharge (Qs) has to be spread out over many different boundary edges, depending on how many are wetted. Users are not encouraged to start with a completely dry channel bed next to an inflow boundary. The model will struggle to figure out how to spread out the water at first, and will likely specify an unrealistic, highly supercritical flow along the thalweg of the inflow section. Instead, as a startup procedure for working with initially dry channels, I recommend that the user place a small pool of water in the river channel next to the

inflow boundary (wherever water is expected). This can be accomplished by setting ic_area=1 in the .input file and creating an .ic file to define a regional initial condition. The modeler should also specify an upstream water level that is consistent with the expected river height or elevation (using the parameter ls1_bc). According to open channel flow theory, discharge should be specified at an inflow boundary and depth should be specified at the outflow boundary under subcritical flow conditions. [Under supercritical flow conditions, both the discharge and depth should be specified at the inflow boundary.] Use of ls1_bc at an inflow boundary might therefore seem illogical under subcritical flow conditions. However, the model uses this parameter value to distribute discharge across the faces of the inflow boundary.

The option to use a “harmonic” boundary condition (opt_bc=2) is designed for tides. The water level is specified as eta(t)=ls1_bc + ls2_bc*cos[2*pi*(t-ts2_bc)/ts1_bc].

When using files to specify time series of either discharge or inflow (opt_bc=3), note that a discharge time series file should include two columns of data (in addition to several other required columns correspond to time): the columns should correspond to discharge (Qs_bc) and river elevation (ls1_bc). When a depth time series is specified, only a single column of data is required.