# Notes on the boundary condition treatment in Rosa

Conservative vs primitive variables

*The conservation equations are always solved in conservative form*, but the resulting nonlinear equations can be solved for primitive or conservative variables when dealing with implicit methods. The switch (solve\_conservative) determines whether the equation system is solved for unknowns U or W:

1. **solve\_cons = 0**. solving for W requires two Jacobians, dU/dW and dF/dW and is therefore more expensive. We solve the conservative equations dU/dt + F(W) = 0 by Newton iterations on W. U and dU/dW follow from UW.m
2. **solve\_cons = 1.** solving for U requires that F(U) can be evaluated, which we do by solving dU/dt + F(W(U)) for U. The mapping W(U) in WU.m is not always available, e.g. it requires that the pressure can be expressed in terms of the density, but works well when the liquid is incompressible and the gas follows a linear equation of state.

For explicit time integration methods we stick to the second approach, and solve dU/dt + F(W(U))=0.

**Types**: inflow (0), outflow (1), or periodic (9)

This is self-explanatory.

**Form**: primitive (0) or conservative (1)

The boundary condition treatment is in principle separate from the conservative or primitive solution approach to the conservation equations described above.

If the boundary conditions are set in conservative form, then in Ubc the unknowns on the boundary are set to be the conservative variables.

If the boundary conditions are set in primitive form, then in Ubc the unknowns on the boundary are set to be the primitive variables.

**Algebraic:** differential (0) or algebraic (1)

1. Boundary conditions in differential form follow the normal time integration method of the interior points, i.e. dU/dt + F = 0 is solved, where F is given by Fbc at the boundary points. Fbc is determined in Calc\_BC\_TWO\_FLUID, which looks at the eigenvalues at the boundary and uses the time derivatives specified in get\_BC\_left and get\_BC\_right.
2. Boundary conditions in algebraic form are of the form p=p\_exit, or mass flow = value. They are specified in get\_BC\_left and get\_BC\_right and are directly called in FX.m.

**Waves:** number of incoming waves at a boundary

Incompressible exceptions

* Inflow: prescribe conservative variables in algebraic way for the momentum equations (rho\*u\*A). For the mass equations it is unclear what to do yet. The mass equations in the first interior point take as flux the inflow conditions. The mass equations on the boundary (e.g. for the holdup) need some sort of boundary treatment, because the holdup on the boundary influences the Laplace operator.
* Outflow: the last velocity point becomes an unknown, and the boundary condition treatment can be skipped for the momentum equations. The pressure at the exit is set to the exit pressure. The mass equations at the exit boundary need some sort of boundary treatment.