Symmetry Boundary Conditions in SU2

Tobias Kattmann

February 2019

1 Symmetry Boundary condition

The low Mach approximation of the Navier Stokes equations write:

$$\frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) - Q = 0, \tag{1}$$

with the convective and viscous fluxes beeing:

$$\bar{F}^c(U) = \begin{pmatrix} \rho\bar{v} \\ \rho\bar{v}\bar{v}^\mathsf{T} + \bar{\bar{I}}p \\ \rho c_p T\bar{v} \end{pmatrix}, \quad \bar{F}^v(U, \nabla U) = \begin{pmatrix} \cdot \\ \bar{\bar{\tau}} \\ \kappa\nabla T \end{pmatrix}, \tag{2}$$

with

$$\bar{\bar{\tau}} = \mu \left[\left(\nabla\bar{v} + (\nabla\bar{v})^\mathsf{T} \right) - \frac{2}{3} (\nabla \cdot \bar{v}) \bar{\bar{I}} \right] \tag{3}$$

$$\mu = \mu_{dyn} + \mu_{turb} \tag{4}$$

$$\kappa = \frac{c_p \mu_{dyn}}{Pr_{dyn}} + \frac{c_p \mu_{turb}}{Pr_{turb}}. \tag{5}$$

In this section the implementation of the symmetry BC in the incompressible solver is explained. On a symmetry plane the following conditions have to be met:

$$\bar{v} \cdot \bar{n} = 0, \tag{6}$$

$$\nabla V \cdot \bar{n} = 0,\ V \text{ is scalar quantity } (p, T, \text{vars of turb model}) \tag{7}$$

$$\nabla (\bar{v} \cdot \bar{n}) \cdot \bar{t} = 0, \tag{8}$$

$$\nabla (\bar{v} \cdot \bar{t}) \cdot \bar{n} = 0, \tag{9}$$

with the outward facing unit normal vector $\bar{n}_i$ which is associated with the boundary vertex:

$$\bar{n}_i = \frac{1}{2} (\bar{n}_{i-1,i} + \bar{n}_{i1,i+1}) \tag{10}$$

Note that in SU2 only area normal vectors are stored such that $\bar{n}^A = A\bar{n}$ with A being the face Area.

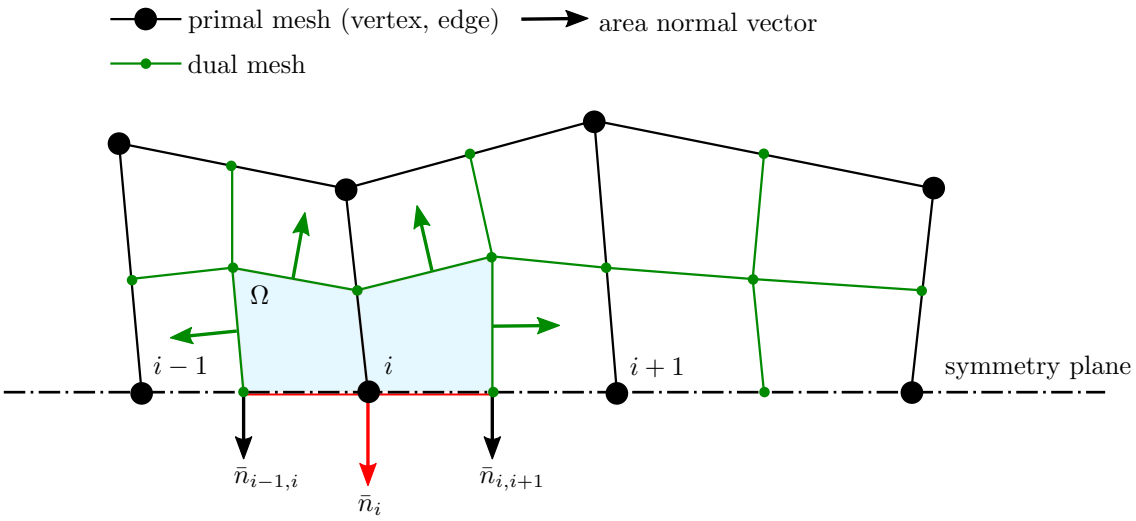

Figure 1: Sketch of the primal and dual meshes at a symmetry BC.

1.1 Reflected cell approach

In the reflected cell approach (not an official name) a reflected/mirrored state is constructed such that the symmetry boundary conditions are fullfilled on the face (by averaging the states) between the real and reflected state. This approach is kind of a halo node approach, but without explicitly keeping the halo node. The reflected values are constructed out of the boundary state just for the flux contribution.

For the convective fluxes the scalar quantities in the reflected cell are identical to that of the boundary cell. The velocity is mirrored along the symmetry axis which is, removing the normal component of the velocity twice:

$$V_r = V_i, (p, T, c_p \text{ etc.}) \tag{11}$$

$$\bar{v}_r = \bar{v}_i - 2\left(\bar{v}_i \cdot \bar{n}\right)\bar{n} \tag{12}$$

where the subscript r depicts the reflected state. The viscous fluxes do require the gradients of the state variables. Their computation, especially for the velocity components is a bit more involved in order to fullfill the above formulated conditions. For gradients of scalar variables the reflected gradient is computed as the velocity above:

$$\bar{V}_r = \bar{v}_i - 2\left(\bar{V}_i \cdot \bar{n}\right)\bar{n},\ V \in p, T \tag{13}$$

To assure that the gradient of the normal velocity in tangential direction is zero and the gradient of the tangential velocity in normal direction is zero the velocity gradient has to be decomposed in its normal and tangential parts. These parts are reflected in transformed (Cartesian to normal/tangential) system and the reflected gradients in Cartesian coordinate system have to be recovered.

1. Following quantities are given: $\nabla\bar{v}, \bar{n}$. Note that the outward facing unit normal vector $\bar{n}$ is constant on a symmetry plane. Note also that in the following everything is based on a given velocity gradient (or on velocity component gradients), at no point the velocity itself is involved.

2. Compute a valid (unit) tangential vector $\bar{t}$. Following conditions have to be fullfilled: $\bar{n} \cdot \bar{t} = 0$ and $\|\bar{t}\|_2 = 1$. Otherwise the direction of the tangential does not have any influence. In 2D one can straight forward set $\bar{t} = (-nx\, ny)$ or $\bar{t} = (nx\, -ny)$ as 2 equations and unknowns are given. In 3D the system is under determined such that a suitable 3rd condition has to be found. We choose to set one component of $\bar{t}$ to zero. This choice is not arbitrary though. If $t_z = 0$ is chosen one gets 2 choices where one is $t_x = -n_y \frac{1}{\sqrt{n_x^2+n_y^2}}$ and $t_y = n_x \frac{1}{\sqrt{n_x^2+n_y^2}}$. Now consider a normal vector which only has a z component, that would lead to a zero denominator. The implemented approach takes the largest entry of the normal, sets one of the two remaining to zero and switches values with the last remaining just as seen above but with the respective indices.

3. Get the gradient of the normal and the tangential velocity

$$\nabla (\bar{v} \cdot \bar{n}) = \nabla(v_x) n_x + \nabla(v_y) n_y \tag{14}$$
$$\nabla (\bar{v} \cdot \bar{t}) = \nabla(v_x) t_x + \nabla(v_y) t_y \tag{15}$$

   The extension to 3D is straightforward, adding $\nabla(v_z) n_z$ to the first and $\nabla(v_z) t_z$ to the second equation.

4. Compute the velocity gradients of the reflected cell:

$$\nabla (\bar{v} \cdot \bar{n})_r = \nabla (\bar{v} \cdot \bar{n}) - 2 \left[\nabla (\bar{v} \cdot \bar{n}) \cdot \bar{t}\right] \bar{t} \tag{16}$$
$$\nabla (\bar{v} \cdot \bar{t})_r = \nabla (\bar{v} \cdot \bar{t}) - 2 \left[\nabla (\bar{v} \cdot \bar{t}) \cdot \bar{n}\right] \bar{n} \tag{17}$$

5. Now that the reflected gradients are computed in the transformed coordinate system, the only thing left to do is to retransform into the Cartesian coordinate system:

$$\nabla \bar{v} = \begin{pmatrix} \nabla(v_x)^\mathsf{T} \\ \nabla(v_y)^\mathsf{T} \end{pmatrix} \tag{18}$$

$$\nabla \bar{v} = \nabla \left[(\bar{v} \cdot \bar{n})\, \bar{n} + (\bar{v} \cdot \bar{t})\, \bar{t}\right] = \nabla \left[(\bar{v} \cdot \bar{n})\, \bar{n}\right] + \nabla \left[(\bar{v} \cdot \bar{t})\, \bar{t}\right] \tag{19}$$

$$\nabla v_x = \nabla (\bar{v} \cdot \bar{n})_r\, n_x + \nabla (\bar{v} \cdot \bar{t})_r\, t_x \tag{20}$$
$$\nabla v_y = \nabla (\bar{v} \cdot \bar{n})_r\, n_y + \nabla (\bar{v} \cdot \bar{t})_r\, t_y \tag{21}$$

   In 3D, one simply adds the equation:

$$\nabla v_z = \nabla (\bar{v} \cdot \bar{n})_r\, n_z + \nabla (\bar{v} \cdot \bar{t})_r\, t_z \tag{22}$$

1.2 Edwin's approach

To sum it up here the final formula (reverse engineered from the DG FEM code) for the reflected gradient in 3D (for 2D just omit the z direction):

$$\nabla\ (v_x)_r = \nabla v_x - 2 \left(\nabla v_x \cdot \bar{n}\right) \bar{n} - 2 n_x \nabla (\bar{v} \cdot \bar{n}) + 2 * 2 n_x \bar{n} \left[\nabla (\bar{v} \cdot \bar{n}) \cdot \bar{n}\right] \tag{23}$$
$$\nabla (v_y)_r = \nabla v_y - 2 \left(\nabla v_y \cdot \bar{n}\right) \bar{n} - 2 n_y \nabla (\bar{v} \cdot \bar{n}) + 2 * 2 n_y \bar{n} \left[\nabla (\bar{v} \cdot \bar{n}) \cdot \bar{n}\right] \tag{24}$$
$$\nabla (v_z)_r = \nabla v_z - 2 \left(\nabla v_y \cdot \bar{n}\right) \bar{n} - 2 n_z \nabla (\bar{v} \cdot \bar{n}) + 2 * 2 n_z \bar{n} \left[\nabla (\bar{v} \cdot \bar{n}) \cdot \bar{n}\right] \tag{25}$$

Both presented approaches (Edwin's and mine) are identical! I wrote a little python script which tests that with random gradient and normal vector input. The derivation of Edwin though is not clear to me. I suspect he did a derivation just like me and then either put everything together by hand into one formula or extracted this easier form from a tool like Wolfram Mathematica.