

## Method cheat sheet for class VortexSystem

What it does	Function	Jump to
Calculates the coordinates of the bound vortices of one blade and saves them to "coordinates_blade_bound_elementwise". It does that by purely rotating the given blade structure to the current "blade_rotation".	<code>_blade_bound_elementwise</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L348">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L348</a>
Calculates the coordinates of the trailing vortices of one blade and saves them to "coordinates_blade_trailing_elementwise". It does that by first calculating the first trailing vortex which is fixed to the blade. This vortex is parallel to the chord and goes from 1/4 chord to 5/4 chord. After that it's helix geometry.	<code>_blade_trailing_elementwise</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L361">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L361</a>
Combines elementwise coordinates into one numpy array. If the coordinates do not exist run an input function "if_not_do" (which should then calculate these coordinates). After that, continue with the combining.	<code>_combine_elementwise</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L434">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L434</a>
Rotates combined coordinates. If the coordinates do not exist run an input function "if_not_do" (which should then calculate these coordinates). After that, continue with the rotating.	<code>_rotate_combined</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L455">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L455</a>
Calculate the induced velocities at "control_point" of a vortex with circulation=1 that spans from "vortex_star" to "vortex_end".	<code>_vortex_induction_factor</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L485">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L485</a>
Build up a matrix that relates the induced velocities at a control point with all existing bound vortices. The nested loops are (1. outermost, 3. innermost): <ol style="list-style-type: none"> <li>1. Bound vortex systems of all blades</li> <li>2. Control points</li> <li>3. Bound vortices of the current bound vortex system</li> </ol>	<code>bound_induction_matrices</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L185">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L185</a>
Build up a matrix that relates the induced velocities at a control point with all existing trailing vortices. The nested loops are (1. outermost, 4. innermost): <ol style="list-style-type: none"> <li>1. Trailing vortex system of all blades</li> <li>2. Control points</li> <li>3. Blade elements</li> <li>4. Trailing vortices of the current blade element of the current trailing vortex system</li> </ol>	<code>trailing_induction_matrices</code>	<a href="https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L217">https://github.com/JonasMPN/LiftingLine/blob/master/src/vortex_system.py#L217</a>
All other functions are either for error reporting, setting values of the class, plotting the vortex system, or changing the data format of method inputs.		

