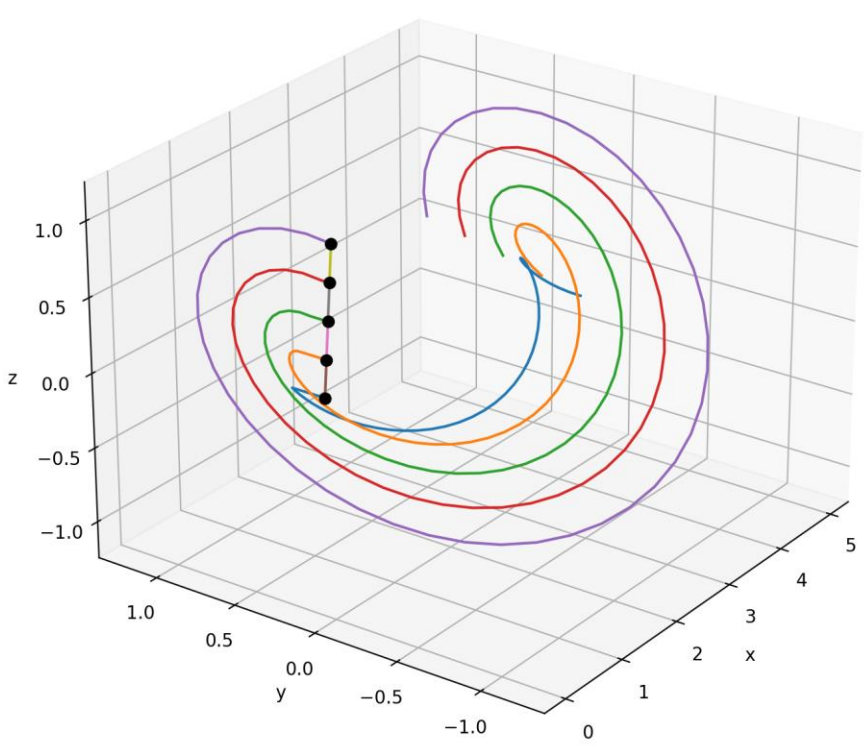
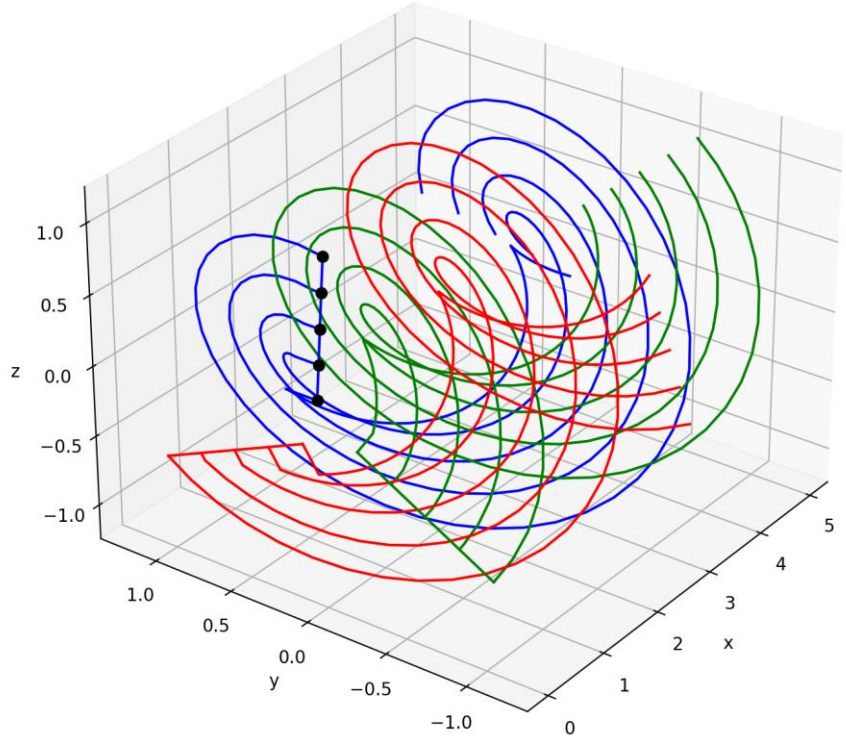
# Class VortexSystem

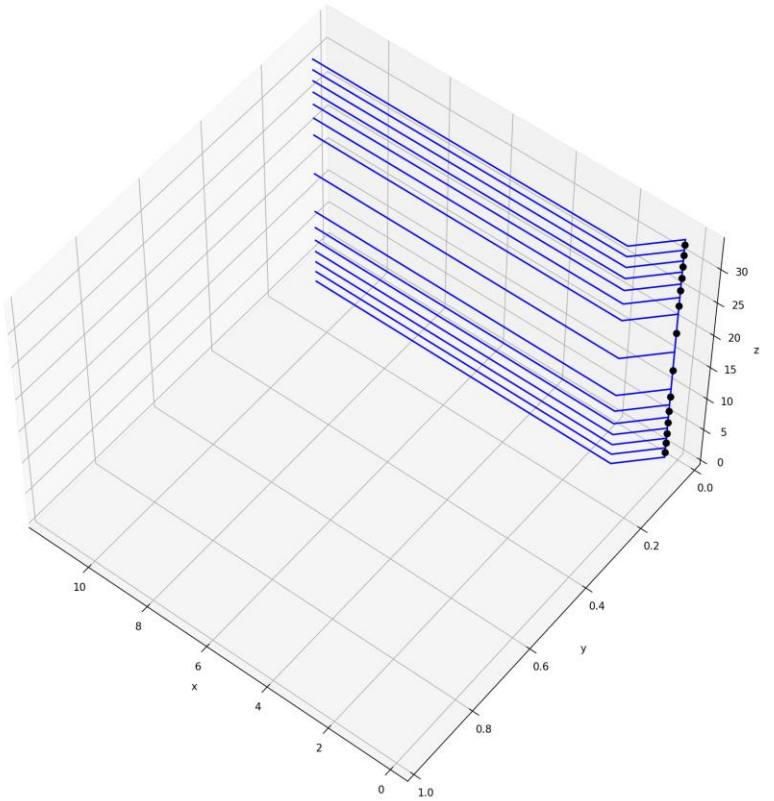


**What is meant with “vortex system”?**
All vortices related to a blade or rotor, namely trailing and bound vortices.

**What VortexSystem is based on:**
- A vortex system of separated trailing and bound vortices (not one horseshoe vortex per blade element)
- Control points

**What VortexSystem can do:**
- Calculate the coordinates of all vortices
- Visualise the vortex system and control points
- Calculate factors which, when multiplied with circulation values, yield the induced velocities at the control points




Using VortexSystem for a flat plate.
Blue: vortices, black dots: control points

# For the user – general idea

**Blade representation:**
VortexSystem sees the blade as a collection of blade elements. Every element has a bound vortex with a constant circulation. The difference in bound circulation from element to element is shed as a trailing vortex. Each trailing vortex has a constant circulation along ist length.

**Needed from the user:**
Blade geometry, rotor properties, and control points for which the induced velocities will be calculated.

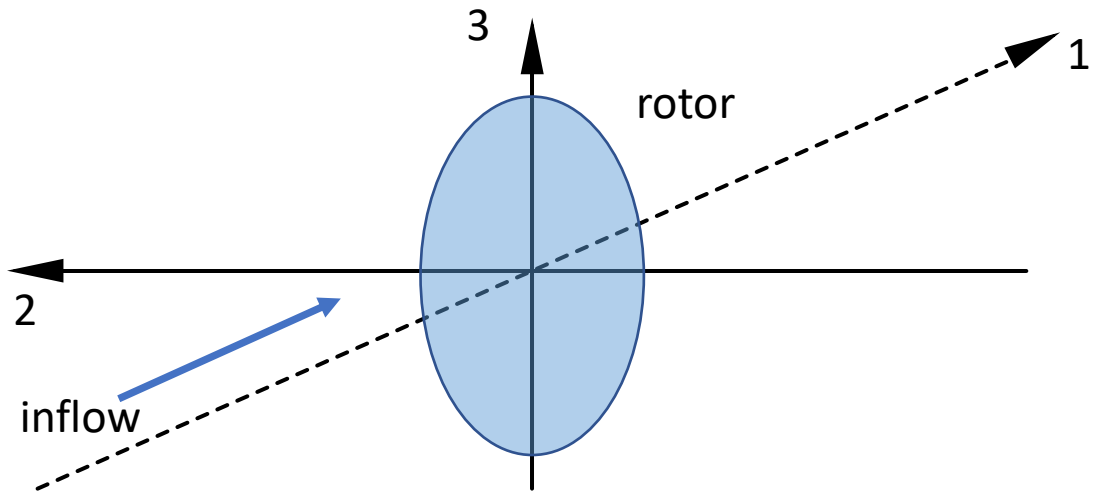
# For the user – important to know

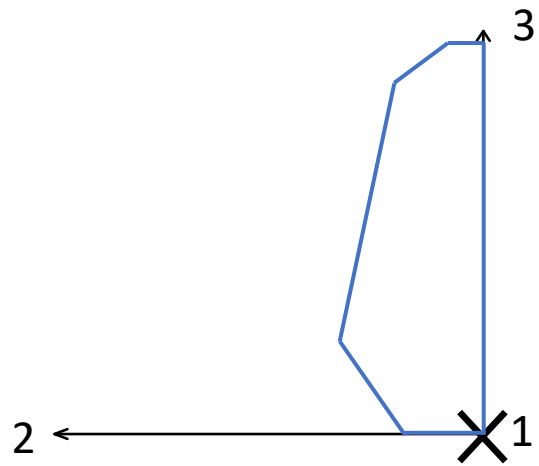
**Coordinate system (see slide 6 for a visualisation):**
The right hand rule is followed. The origin lies in the axis of rotation of the rotor. The first axis points downstream along the axis of rotation. The second axis points to the left looking downstream at the rotor. The third axis points upwards (parallel to the tower). These descriptions are not valid for tilted shaft turbines. Anlges follow the right hand rule.

It is assumed that the blade‘s leading edge lies on the first axis, which means that the leading edge points directly into the axis of rotation.

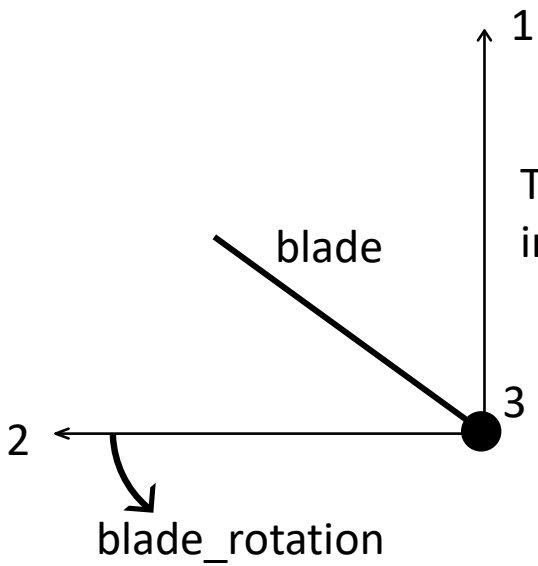
# For the user - inputs

| Function | Parameter | Description |
|---|---|---|
| set_blade() | r_elements | Radial positions separating the blade elements. |
| | c_elements | Chord at r_elements |
| | blade_rotation | Rotational position of the blade with respect to the freestream |
| | rotor_rotation_speed | Rotational speed of the rotor |
| | n_blades | Number of blades of the rotor |
| set_wake() | wake_speed | Speed at which the wake is convected |
| | wake_length | Length of the wake |
| | resolution | Number of elements per trailing vortex |
| set_control_points() | x_control_points (and y and z) | Set coordinates of coordinates along axis 1 (x), 2 (y) and 3 (z) |

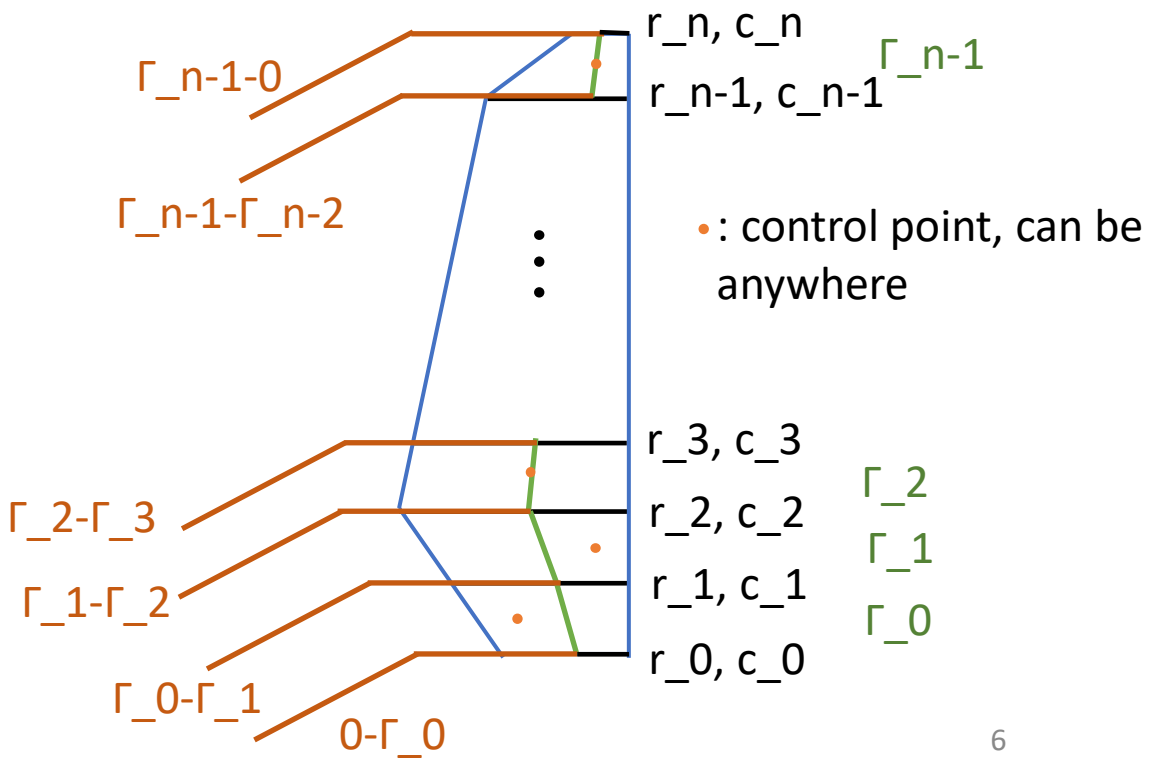



Blade outline for
blade_rotation>-pi/2
(and smaller than +pi/2)




The blade is parallel to the
inflow for blade_rotation=-pi/2



# For the user – calculating induction matrices

1. Create instance of VortexSystem()
2. set_rotor(*params1*): set the rotor properties
3. set_wake(*params2*): set the wake properties
4. rotor(): calculates the coordinates of the vortex system
5. set_control_points(*params3*): set the control points
6. tu, tv, tw = trailing_induction_matrices(): get induction matrices from the trailing vortices for the three velocity components
7. bu, bv, bw = bound_induction_matrices() : get induction matrices from the bound vortices for the three velocity components
8. Do calculations with the bound and trailing circulations.

# For the coder – idea of calculating the vortex system coordinates

Same procedure for bound and trailing vortices.

First, the coordinates of the vortex filaments of the vortex system have to be calculated. There are 3 levels of the vortex systems:

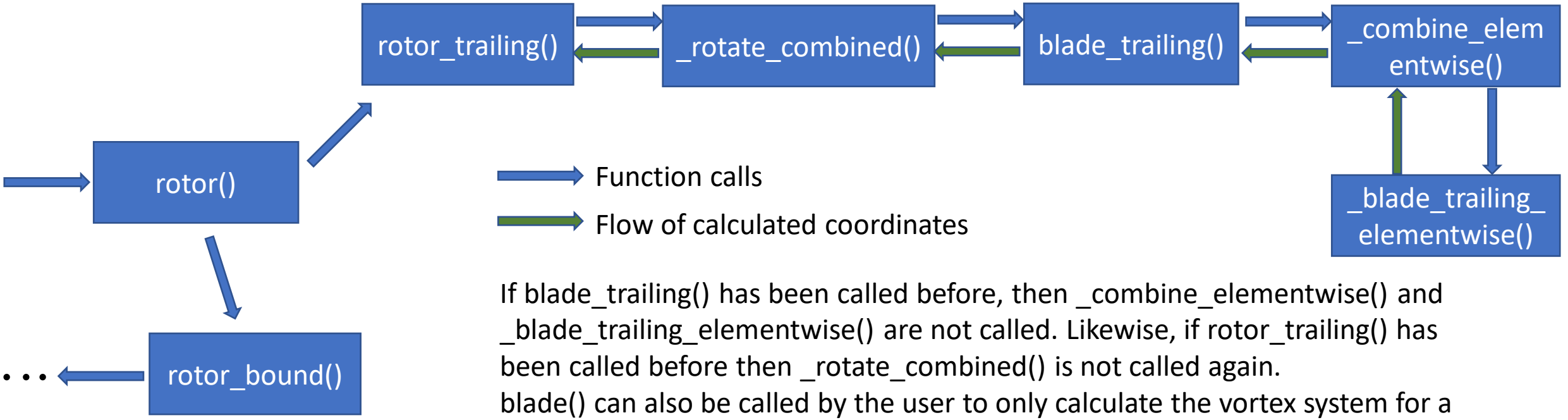
- “element_wise”: dictionary for which the keys are the radial positions and their values are their respective vortex’ coordinates
- “blade”: same points as “element_wise” but all coordinates are in one large numpy array
- “rotor”: a list of the array “blade” and rotations of “blade”

The calculations are performed in the order “element_wise” -> “blade”

-> “rotor”.

# For the coder – flowchart of calculating the vortex system coordinates

If no prior calculations have been done, the following happens (remember that trailing and bound have the same workflow):




If blade_trailing() has been called before, then _combine_elementwise() and _blade_trailing_elementwise() are not called. Likewise, if rotor_trailing() has been called before then _rotate_combined() is not called again.
blade() can also be called by the user to only calculate the vortex system for a blade, not for the whole rotor. The flowchart then omits rotor_trailing() and _rotate_combined().

# For the coder – idea of calculating the induction matrices

This one is simpler. The methods trailing_induction_matrices() and bound_induction_matrices() work like this:

1. Loop over the vortex system of each blade
    2. Loop over all control points
        3. Loop over the vortex filaments of the current vortex system
            4. Calculate the vortex induction factor