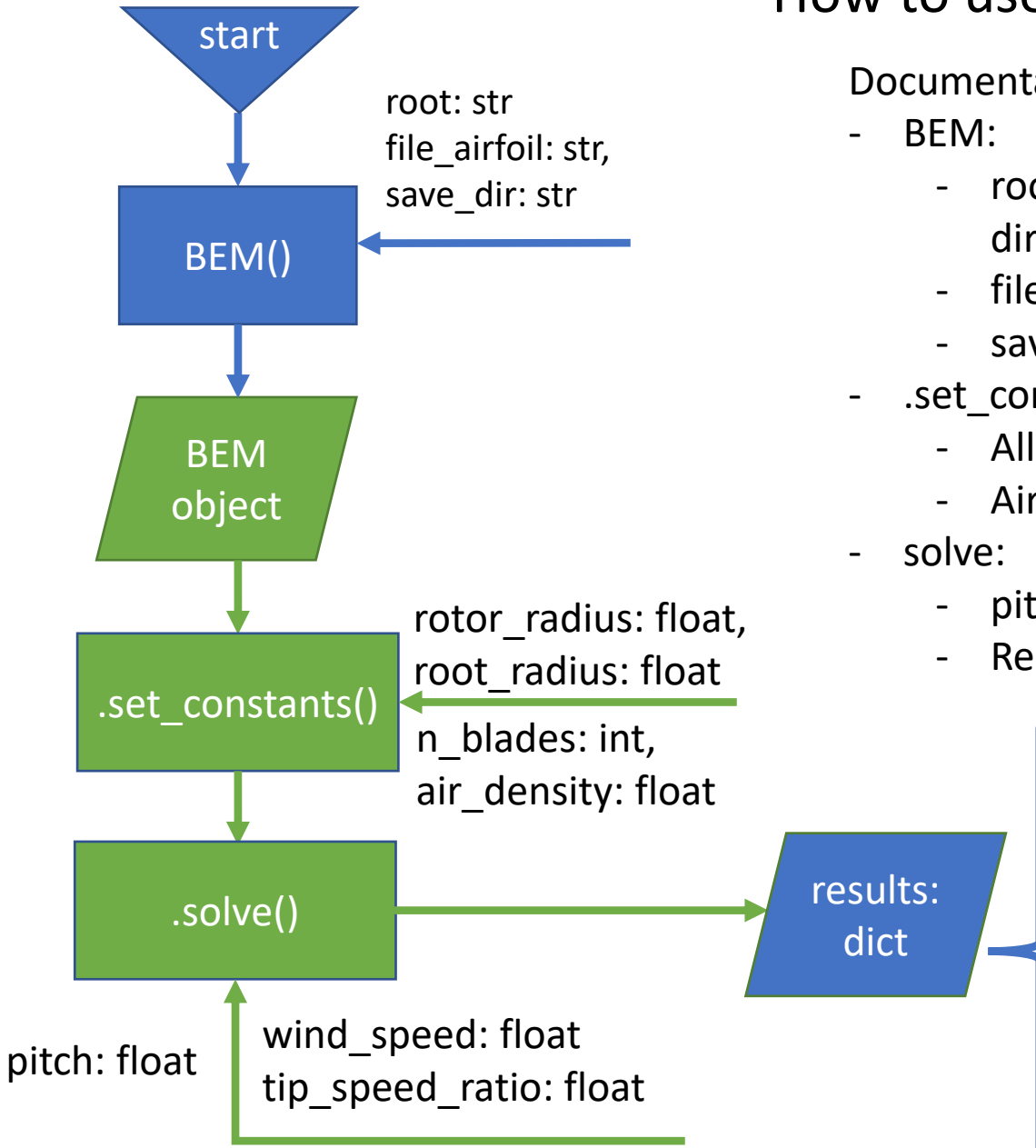
How to use the BEM class



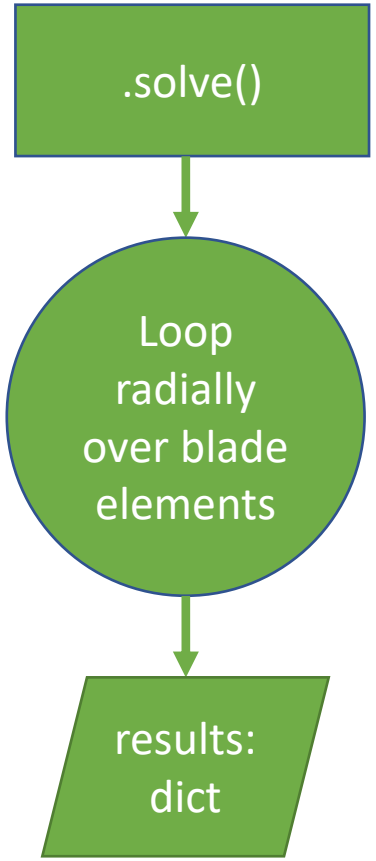
Documentation for non-trivial function inputs:

- BEM:
  - root: string to directory in which the airfoil file and the results directory is
  - file_airfoil: filename with filetype as string
  - save_dir: directory name of results as string
- .set_constants:
  - All radii in metre.
  - Air density in kg/m3
- solve:
  - pitch: in degree
  - Remaining values that are pre-set are explained later

Dictionary key	Explanation
positions	Radial positions in m
a	Axial induction factor
a_prime	Tangential induction factor
f_n	Normal load in N/m
f_t	Tangential load in N/m
bec	Blade end correction (see next slide)

What solve() does

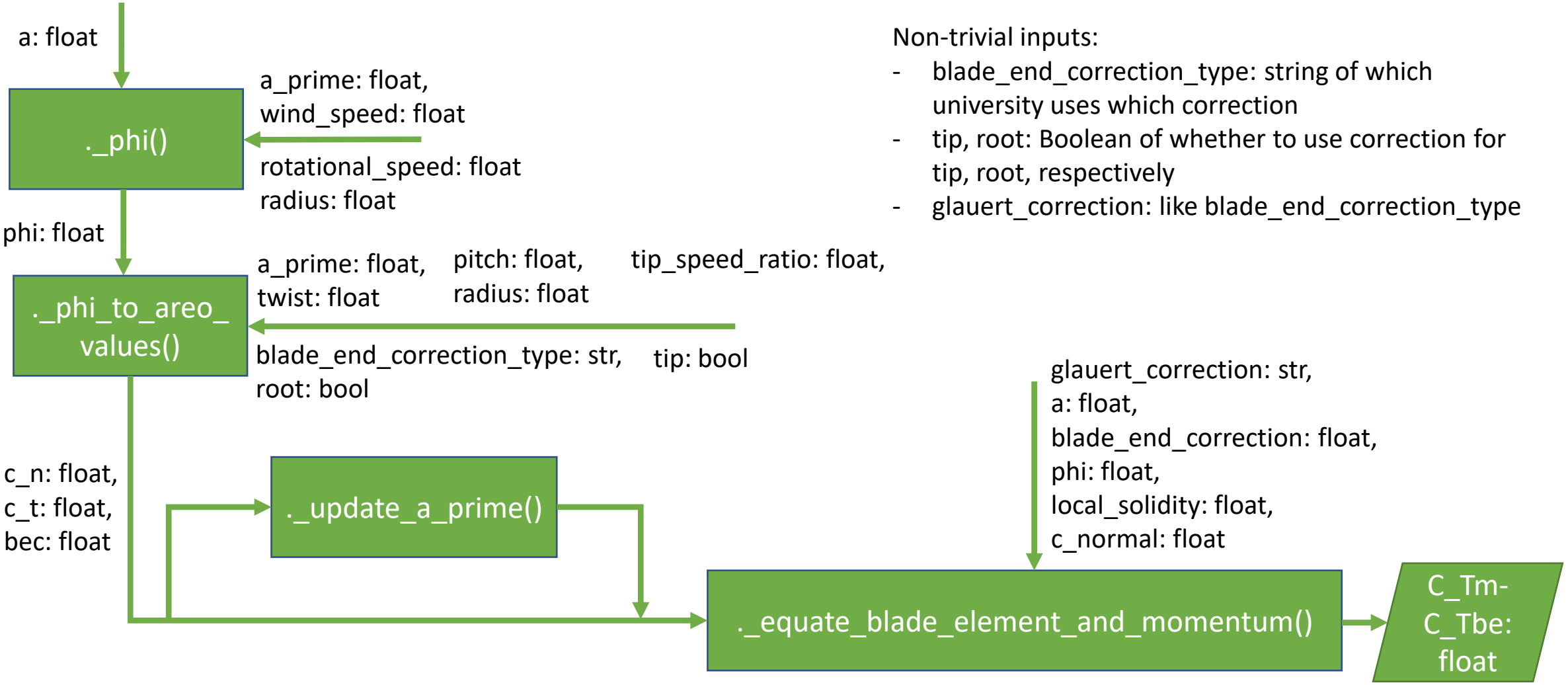
The numerical approach needed for equating the loads of the blade element and momentum theory is not done explicitly, but implicitly by solving a root finding problem for the difference of thrust coefficient coming from both theories. Here, it is implicit because scipy methods for root finding are used. These functions require, amongst other parameters, the function of which the root is to find as input.



Step	Action	Function name
1	Get chord	._get_chord
2	Get twist	._get_twist
3	Get local solidity	._local_solidity
4	Define function for root problem	residue (explained next slide)
4	Get induction factor (solve root problem)	brentq or newton
5	Translate induction factor to values of interest	._phi_to_aero_values and local calcuations
6	Return values of step 5	

What residue() does

The function residue() receives an axial induction factor *a* as input and returns the difference between the C_T from momentum theory (m) and blade element (be) theory that follows for the current *a*.



Non-trivial inputs:

- blade_end_correction_type: string of which university uses which correction
- tip, root: Boolean of whether to use correction for tip, root, respectively
- glauert_correction: like blade_end_correction_type