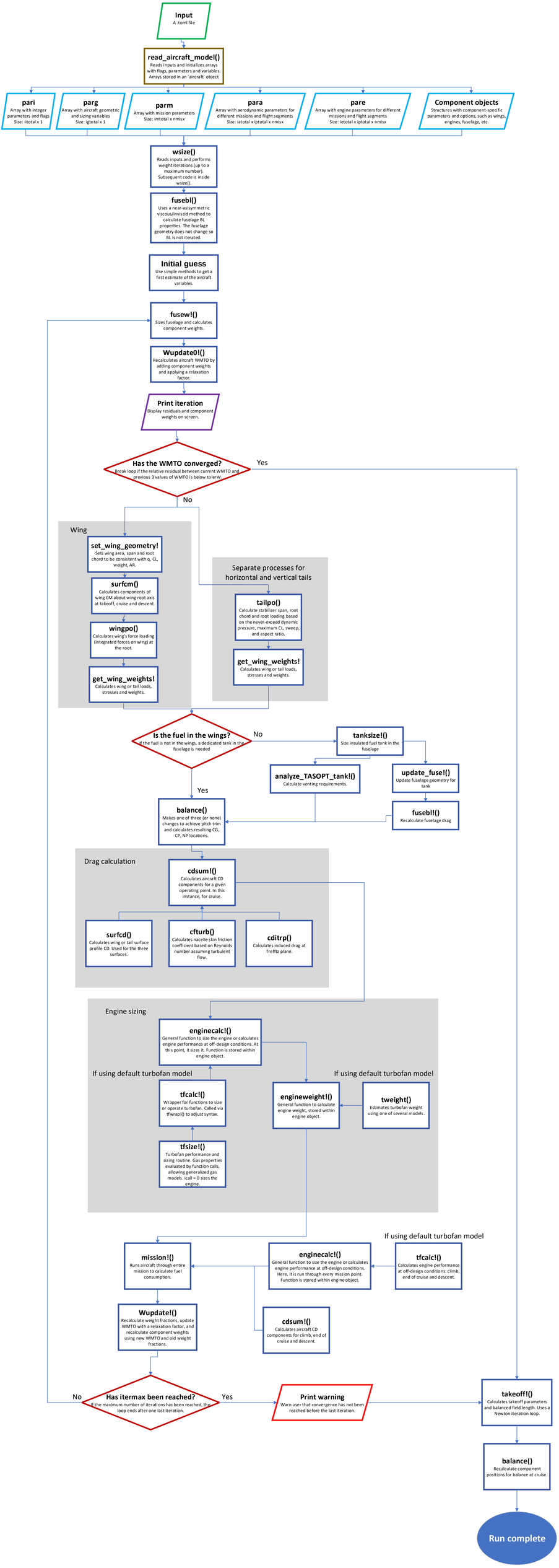
Input
A .toml file
read_aircraft_model()
Reads inputs and initializes arrays with flags, parameters and variables. Arrays stored in an 'aircraft' object
pari
Array with integer parameters and flags
Size: iitotal x 1
parg
Array with aircraft geometric and sizing variables
Size: igtotal x 1
parm
Array with mission parameters
Size: imtotal x nmisx
para
Array with aerodynamic parameters for different missions and flight segments
Size: iatotal x iptotal x nmisx
pare
Array with engine parameters for different missions and flight segments
Size: ietotal x iptotal x nmisx
Component objects
Structures with component-specific parameters and options, such as wings, engines, fuselage, etc.
wsize()
Reads inputs and performs weight iterations (up to a maximum number). Subsequent code is inside wsize().
fusebl()
Uses a near-axisymmetric viscous/inviscid method to calculate fuselage BL properties. The fuselage geometry does not change so BL is not iterated.
Initial guess
Use simple methods to get a first estimate of the aircraft variables.
fusew!()
Sizes fuselage and calculates component weights.
Wupdate0!()
Recalculates aircraft WMTO by adding component weights and applying a relaxation factor.
Print iteration
Display residuals and component weights on screen.
Has the WMTO converged?
Break loop if the relative residual between current WMTO and previous 3 values of WMTO is below tolerW.
Yes
No
Wing
set_wing_geometry!
Sets wing area, span and root chord to be consistent with q, CL, weight, AR.
surfcm()
Calculates components of wing CM about wing root axis at takeoff, cruise and descent.
wingpo()
Calculates wing's force loading (integrated forces on wing) at the root.
get_wing_weights!
Calculates wing or tail loads, stresses and weights.
Separate processes for horizontal and vertical tails
tailpo()
Calculate stabilizer span, root chord and root loading based on the never-exceed dynamic pressure, maximum CL, sweep, and aspect ratio.
get_wing_weights!
Calculates wing or tail loads, stresses and weights.
Is the fuel in the wings?
If the fuel is not in the wings, a dedicated tank in the fuselage is needed
No
Yes
tanksize!()
Size insulated fuel tank in the fuselage
analyze_TASOPT_tank!()
Calculate venting requirements.
update_fuse!()
Update fuselage geometry for tank
fusebl!()
Recalculate fuselage drag
balance()
Makes one of three (or none) changes to achieve pitch trim and calculates resulting CG, CP, NP locations.
Drag calculation
cdsum!()
Calculates aircraft CD components for a given operating point. In this instance, for cruise.
surfcd()
Calculates wing or tail surface profile CD. Used for the three surfaces.
cfturb()
Calculates nacelle skin friction coefficient based on Reynolds number assuming turbulent flow.
cditrp()
Calculates induced drag at Trefftz plane.
Engine sizing
enginecalc!()
General function to size the engine or calculates engine performance at off-design conditions. At this point, it sizes it. Function is stored within engine object.
If using default turbofan model
tfcalc!()
Wrapper for functions to size or operate turbofan. Called via tfwrap!() to adjust syntax.
tfsize!()
Turbofan performance and sizing routine. Gas properties evaluated by function calls, allowing generalized gas models. icall = 0 sizes the engine.
If using default turbofan model
engineweight!()
General function to calculate engine weight, stored within engine object.
tweight()
Estimates turbofan weight using one of several models.
mission!()
Runs aircraft through entire mission to calculate fuel consumption.
enginecalc!()
General function to size the engine or calculates engine performance at off-design conditions. Here, it is run through every mission point. Function is stored within engine object.
If using default turbofan model
tfcalc!()
Calculates engine performance at off-design conditions: climb, end of cruise and descent.
Wupdate!()
Recalculate weight fractions, update WMTO with a relaxation factor, and recalculate component weights using new WMTO and old weight fractions.
cdsum!()
Calculates aircraft CD components for climb, end of cruise and descent.
Has itermax been reached?
If the maximum number of iterations has been reached, the loop ends after one last iteration.
No
Yes
Print warning
Warn user that convergence has not been reached before the last iteration.
takeoff!()
Calculates takeoff parameters and balanced field length. Uses a Newton iteration loop.
balance()
Recalculate component positions for balance at cruise.
Run complete