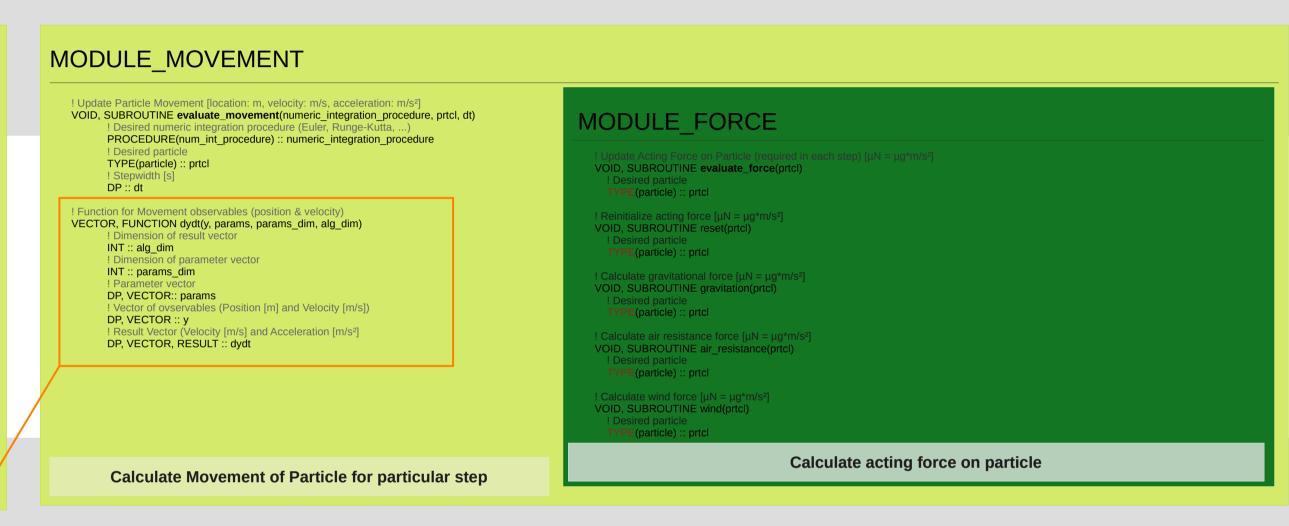
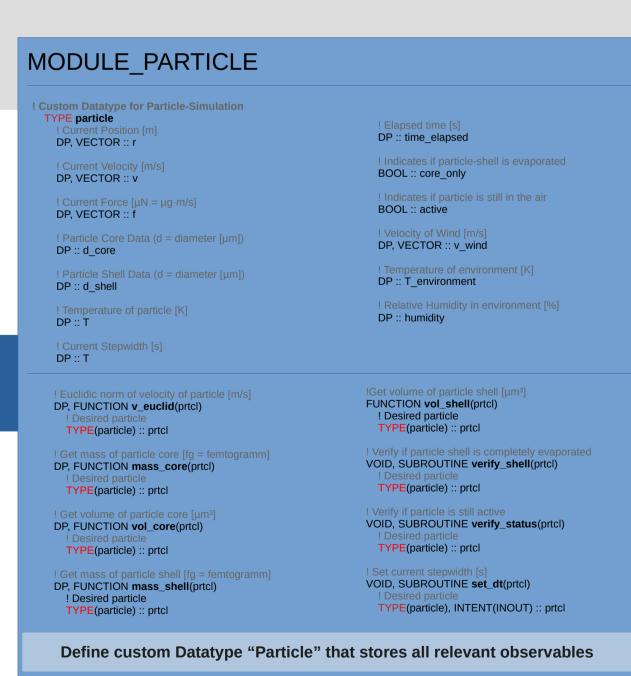
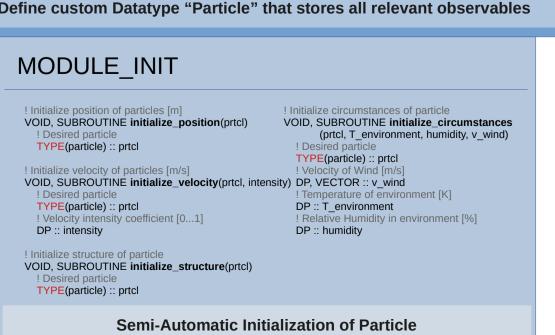


MODULE_EVAPORATION 1 1st Order Derivative of particle diameter [um/s] Calculate Sherwood number [dimensionless] DP, FUNCTION Sh(velocity, diameter, T_environment, T_particle) VOID, SUBROUTINE evaluate_evaporation(integration_procedure, prtcl, dt) VECTOR, FUNCTION dddt(y, params, params_dim, alg_dim) DP :: T_environment, T_particle INT :: alg_dim PROCEDURE(num_int_procedure) :: integration_procedure INT :: params_dim TYPE(particle) :: prtcl ! Diameter of particle [µm] DP, VECTOR:: params DP, VECTOR :: y ! Diffusion Coefficient [$m^2/s = \mu m^2/s * 10^{12}$] ! Calculate Reynolds number [dimensionless] DP, FUNCTION D_Coeff(T_particle, T_environment) DP, FUNCTION Re(velocity, diameter, T_environment) DP, VECTOR :: dddt DP :: T_particle, T_environment DP :: velocity ! Calculate partial pressure of H2O at certain temperature [Pa = $kg \cdot m^{-1} \cdot s^{-2}$] DP :: diameter DP, FUNCTION p0_H2O(T) Temperature of environment and particle [K] DP :: T_environment ! Calculate Schmidt number [dimensionless] ! Calculate partial pressure of H2O in particle (fluid water) [Pa = $kg \cdot m^{-1} \cdot s^{-2}$] DP, FUNCTION Sc(T_environment, T_particle) DP, FUNCTION pw_H2O(T_particle) DP :: T_environment, T_particle DP :: T_particle ! Calculate mass transfer coefficient [µm/s] ! Calculate partial pressure of H2O in environment (steam) [Pa = $kg \cdot m^{-1} \cdot s^{-2}$] DP, FUNCTION h_m(diameter, T_environment, T_particle, velocity) DP, FUNCTION pinf_H2O(T_environment, humidity) DP :: T_environment, T_particle DP :: velocity DP :: diameter Calculate Evaporation of Particle-Shell for particular step



MODULE_PARAMETERS INT, PARAMETER :: dim = 3 DP, PARAMETER :: rho_H20 DP, DIMENSION(dim) :: g DP, PARAMETER :: rho_cov2 DP, PARAMETER :: R DP, PARAMETER :: D 0 DP, PARAMETER :: T_0 DP, PARAMETER :: p_atm DP, PARAMETER :: M_H2O DP, PARAMETER :: A DP, PARAMETER :: B DP, PARAMETER :: C DP, PARAMETER :: M_air DP, PARAMETER :: mmHg_Pa_conversion DP, PARAMETER :: PI DP, FUNCTION rho_air(T_environment) DP :: T_environment DP, FUNCTION nu_air(T_environment) DP :: T_environment DP, FUNCTION etha_air(T_environment) DP :: T_environment **Define global calculation parameters**





NUMERIC_INTEGRATION VECTOR, FUNCTION **func**(y, params, params_dim, alg_dim) VOID, SUBROUTINE num_int_procedure(f, y, dt, params, params_dim, alg_dim) PROCEDURE(func) :: f INT :: alg_dim INT :: alg_dim INT :: params_dim INT :: params_dim DP, VECTOR:: params DP, VECTOR:: params DP, VECTOR :: y DP, VECTOR :: y DP, VECTOR, RESULT :: dydx VOID, SUBROUTINE runge_kutta_2k(f, y, dt, params, params_dim, alg_dim) VOID, SUBROUTINE runge_kutta_4k(f, y, dt, params, params_dim, alg_dim) VOID, SUBROUTINE **euler**(f, y, dt, params, params_dim, alg_dim) PROCEDURE(func) :: f PROCEDURE(func) :: f PROCEDURE(func) :: f INT :: alg_dim INT :: alg dim INT :: alg_dim INT :: params_dim INT :: params_dim INT :: params_dim DP, VECTOR:: params DP, VECTOR:: params DP, VECTOR:: params DP, VECTOR :: y DP, VECTOR :: y DP, VECTOR :: y **Generic Implementation of several numeric integration procedures**