```julia
function two_body!(du, u, μ, t)
        # u = [ x, y, z, vx, vy, vz ]
        # du[1:3] = v
        # du[4:6] = acceleration
        @views du[1:3] .= u[4:6]
        r = @view u[1:3]
        r_norm = norm(r)
        @views du[4:6] .= -μ .* r ./ (r_norm^3)
end

function two_body_thrust!(du, u, params, t)
        μ, T_spec_kN_per_kg = params

        # unpack position & velocity views
        @views du[1:3] .= u[4:6]
        r = @view u[1:3]
        v = @view u[4:6]

        # gravity
        r_norm = norm(r)
        grav_acc = -μ .* r ./ (r_norm^3)

        a_thrust_mag = T_spec_kN_per_kg
        v_norm = norm(v)
        thrust_acc = v_norm > 0 ? a_thrust_mag .* (v ./ v_norm) : zero(v)

        @views du[4:6] .= grav_acc .+ thrust_acc
end

function solve_2BP(initial::StateVectors,
                    tspan::Tuple{Float64, Float64};
                    μ::Float64 = μ_Earth,
                    reltol::Float64 = 1e-9,
                    abstol::Float64 = 1e-9,
                    int_pts::Int64 = 2)

        # pack initial state
        u0 = vcat(initial.r, initial.v)

        # setup and solve ODE problem
        prob = ODEProblem(two_body!, u0, tspan, μ)
        sol  = solve(prob, Tsit5(), reltol=reltol, abstol=abstol,
saveat=range(start=tspan[1], stop=tspan[2], length=int_pts))

        # unpack back into StateVectors
        return [StateVectors(u[1:3], u[4:6]) for u in sol.u]
end

function solve_2BP_thrust(initial::StateVectors,
                            tspan::Tuple{Float64, Float64};
                            μ::Float64 = μ_Earth,
                            T_spec::Float64 = 1e-4,
                            reltol::Float64 = 1e-9,
                            abstol::Float64 = 1e-9,
                            int_pts::Int64 = 2)

        # pack initial state
        u0 = vcat(initial.r, initial.v)
```

```julia
        # setup and solve ODE problem
        prob = ODEProblem(two_body_thrust!, u0, tspan, (μ, T_spec))
        sol  = solve(prob, Tsit5();
                reltol = reltol,
                abstol = abstol,
                saveat = range(tspan[1], tspan[2], length=int_pts))

        # unpack back into StateVectors
        return [ StateVectors(u[1:3], u[4:6]) for u in sol.u ]
end

export solve_2BP, solve_2BP_thrust

Error: UndefVarError: `StateVectors` not defined in `Main.var"##WeaveSandBo
x#233"`
Suggestion: check for spelling errors or missing imports.
```

```julia
        # setup and solve ODE problem
        prob = ODEProblem(two_body_thrust!, u0, tspan, (μ, T_spec))
        sol  = solve(prob, Tsit5();

        # unpack back into StateVectors
```