

```

include(".././../code/sfd.jl")
using .SpaceFlightDynamics

# Case 1: short-way
r1_c1 = [8000.0, 0.0, 0.0]
r2_c1 = [7000.0, 7000.0, 0.0]
TOF_c1 = 3600.0

v1_c1, v2_c1, e_c1, rp_c1 = solve_lambert(r1_c1, r2_c1, TOF_c1; long_way=false)

println("Case 1 (short way):")
println("  v_1 = ", v1_c1)
println("  v_2 = ", v2_c1)
println("  e  = ", e_c1)
println("  r_p = ", rp_c1, " km\n")

# Case 2: long-way, using Earth radius
r1_c2 = [0.5, 0.6, 0.7] .* R_Earth
r2_c2 = [0.0, -1.0, 0.0] .* R_Earth
TOF_c2 = 16135.0

v1_c2, v2_c2, e_c2, rp_c2 = solve_lambert(r1_c2, r2_c2, TOF_c2; long_way=true)

println("Case 2 (long way):")
println("  v_1 = ", v1_c2)
println("  v_2 = ", v2_c2)
println("  e  = ", e_c2)
println("  r_p = ", rp_c2, " km")

Case 1 (short way):
  v_1 = [5.459317364023696, 3.466008449141647, 0.0]
  v_2 = [-4.705584118347704, -0.7444316050429653, -0.0]
  e   = 0.8486118938193392
  r_p = 1043.4116692745763 km

Case 2 (long way):
  v_1 = [-2.133759073847983, 7.024037548362913, -2.9872627033871755]
  v_2 = [4.792274218490246, 0.8071074997180263, 6.709183905886343]
  e   = 0.13798287545743462
  r_p = 6096.619935743475 km

```