Introduction to Flight Simulations

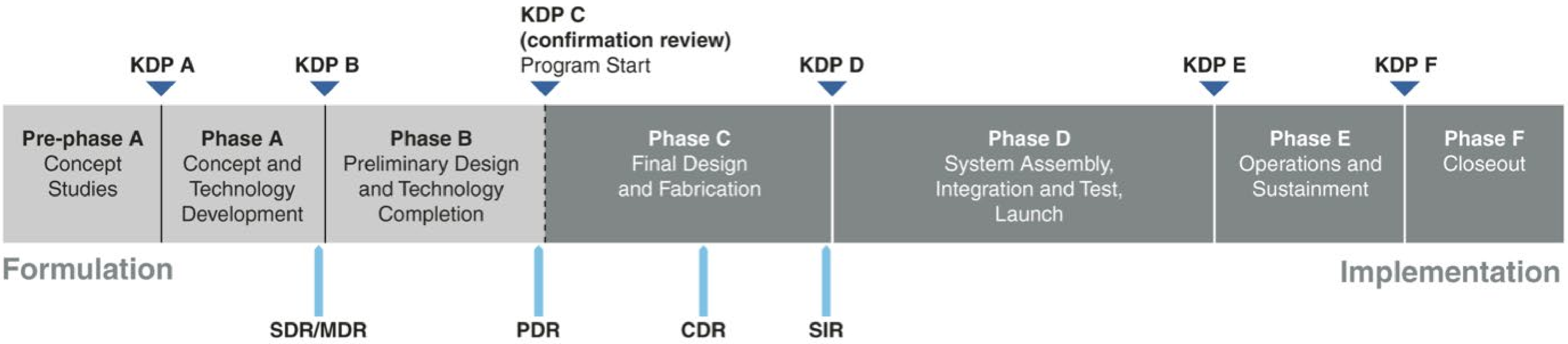
Episode 1: Dissecting the Black Box

Types of Flight Simulation Software

- Realtime Flight Simulations
- Trajectory Simulations

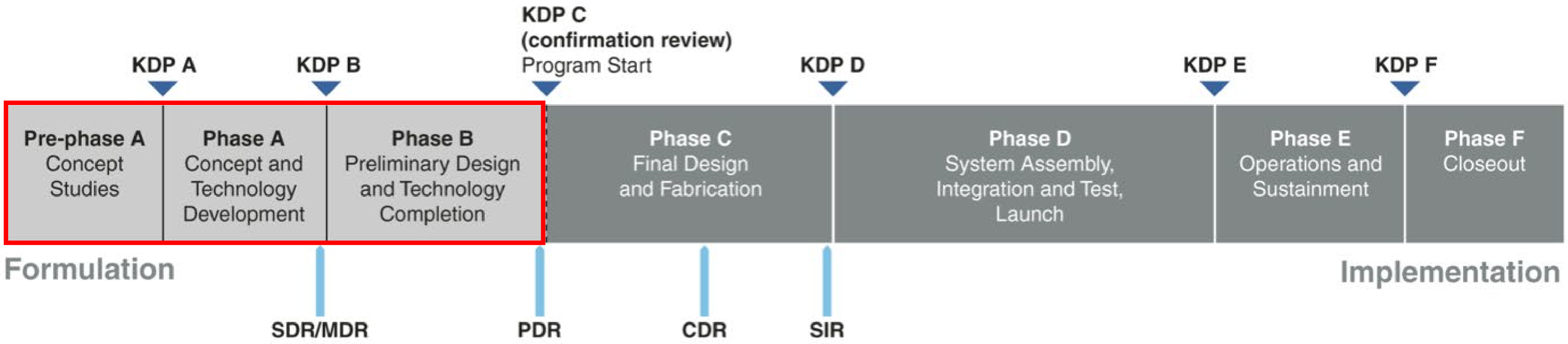
Types of Flight Simulation Software

- Realtime Flight Simulations
  - High fidelity flight vehicle model
  - Human-in-the-loop testing
  - Training
- Trajectory Simulations



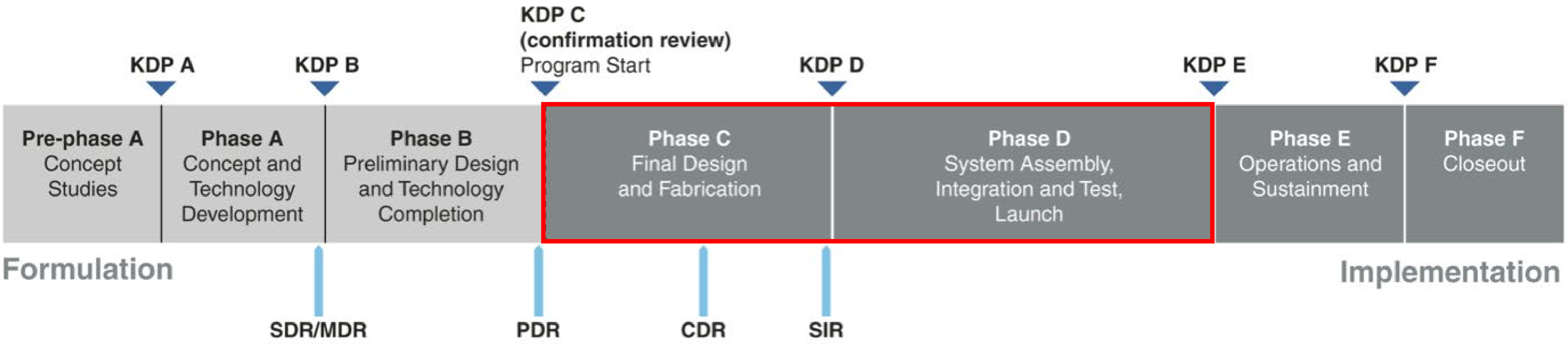
Types of Flight Simulation Software

- Realtime Flight Simulations
  - High fidelity flight vehicle model
  - Human-in-the-loop testing
  - Training
- Trajectory Simulations



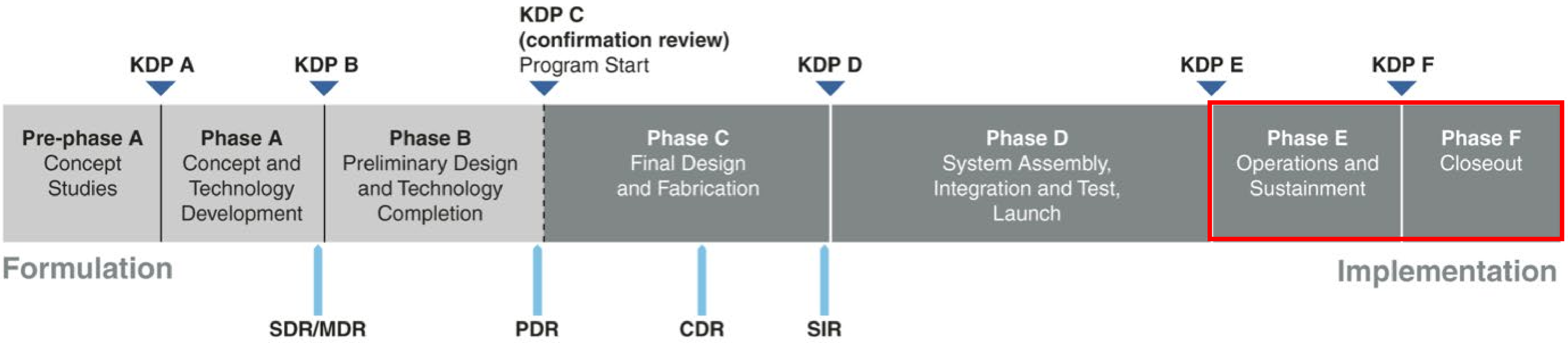
Types of Flight Simulation Software

- Realtime Flight Simulations
  - High fidelity flight vehicle model
  - Human-in-the-loop testing
  - Training
- Trajectory Simulations



Types of Flight Simulation Software

- Realtime Flight Simulations
  - High fidelity flight vehicle model
  - Human-in-the-loop testing
  - Training
- Trajectory Simulations



Inputs

Flight Simulation Tool

Flight Simulation Tool

Outputs

Flight Simulation Tool

Outputs

Inputs

Flight Simulation Tool

Outputs

Inputs

Flight Simulation Tool

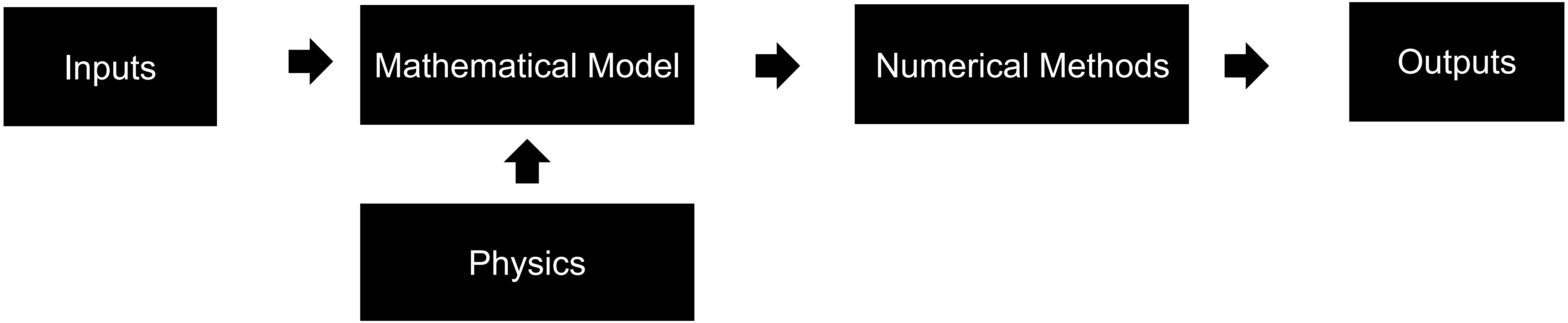
Outputs

Inputs

Flight Simulation Tool

Outputs

Flight Simulation Tool



Mathematical Model

$$\dot{R} = V \sin \gamma \tag{12}$$

$$\dot{\theta} = \frac{V \cos \gamma \sin \psi}{R \cos \phi} \tag{13}$$

$$\dot{\phi} = \frac{V \cos \gamma \cos \psi}{R} \tag{14}$$

$$\dot{V} = -D - g \sin \gamma + \Omega^2 R \cos \phi (\sin \gamma \cos \phi - \cos \gamma \sin \phi \cos \psi) \tag{15}$$

$$\dot{\gamma} = \frac{1}{V}\left[L \cos \sigma + \cos \gamma \left(\frac{V^2}{R} - g\right) + 2\Omega V \cos \phi \sin \psi + \Omega^2 R \cos \phi (\cos \gamma \cos \phi + \sin \gamma \cos \psi \sin \phi)\right] \tag{16}$$

$$\dot{\psi} = \frac{1}{V}\left[\frac{L \sin \sigma}{\cos \gamma} + \frac{V^2}{R} \cos \gamma \sin \psi \tan \phi - 2\Omega V (\tan \gamma \cos \psi \cos \phi - \sin \phi) + \frac{R\Omega^2}{\cos \gamma} \sin \psi \sin \phi \cos \phi\right] \tag{17}$$

Numerical Methods

5.1.1 Euler's Method

The simplest method for numerical integration utilizes a Taylor Series Expansion around the current iteration (y_i, t_i). This can be written as

$$y(t_i + \Delta t) \approx y(t_i) + \Delta t \cdot \frac{dy}{dt}(y_i, t_i) + O(\Delta t^2) \tag{5.1}$$

Neglecting higher order terms, the discrete form for the next iteration is

$$y_{i+1} \approx y_i + \Delta t \cdot \left.\frac{dy}{dt}\right|_i \tag{5.2}$$

Euler's method works by evaluating the derivative at the current state and propagating this change forward in time. Of course, it is always possible to propagate time backwards, but the reason for doing so is discussed later. The approximation sign is dropped for simplicity.

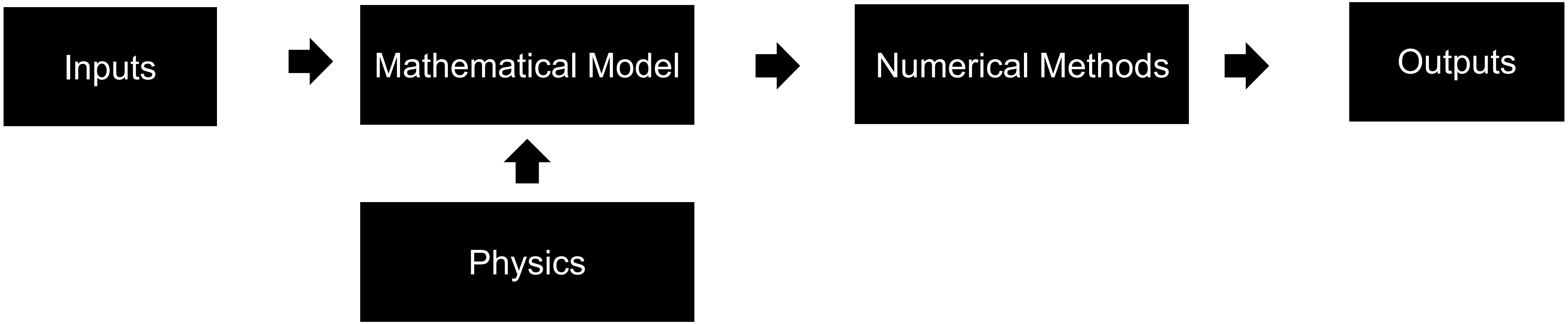
$$y_{i+1} = y_i + \Delta t \cdot \left.\frac{dy}{dt}\right|_i \tag{5.3}$$

Inputs

Flight Simulation Tool

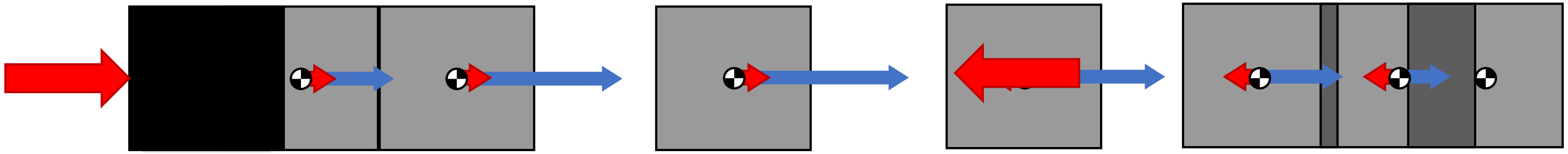
Outputs

Flight Simulation Tool



Physics

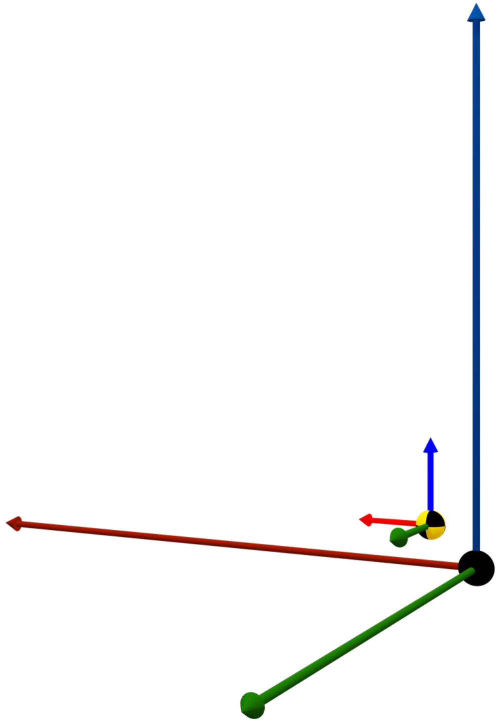
- The motion of a rigid body is driven by the forces that act on it
- Motion is defined by the states of the rigid body throughout time



Physics

Dynamics

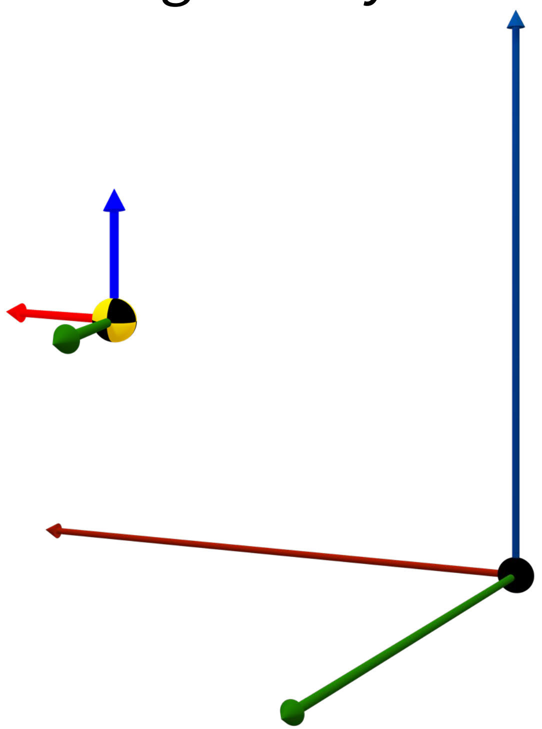
- Kinetics
  - The motion of a rigid body is driven by the forces and moments that act on it
- Kinematics
  - Motion is defined by the translational and rotational states of the rigid body throughout time



Physics

Dynamics

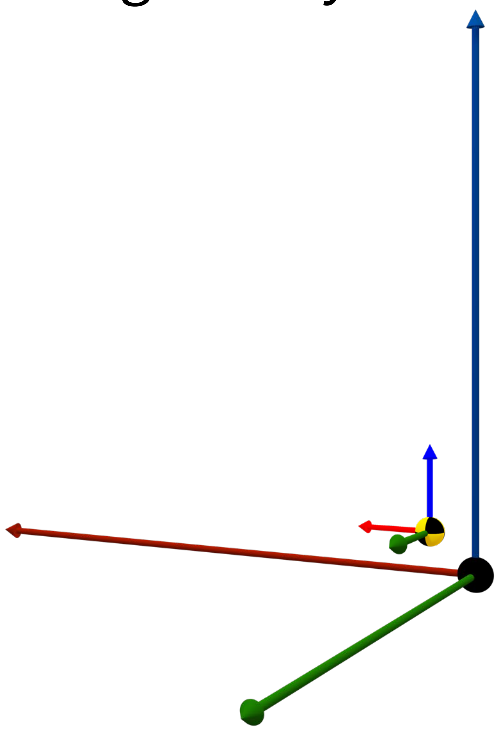
- Kinetics
  - The motion of a rigid body is driven by the forces and moments that act on it
- Kinematics
  - Motion is defined by the translational and rotational states of the rigid body throughout time



Physics

Dynamics

- Kinetics
  - The motion of a rigid body is driven by the forces and moments that act on it
- Kinematics
  - Motion is defined by the translational and rotational states of the rigid body throughout time



Physics

Dynamics

- Kinetics
  - The motion of a rigid body is driven by the forces and moments that act on it
- Kinematics
  - Motion is defined by the translational and rotational states of the rigid body throughout time

Forces and moments

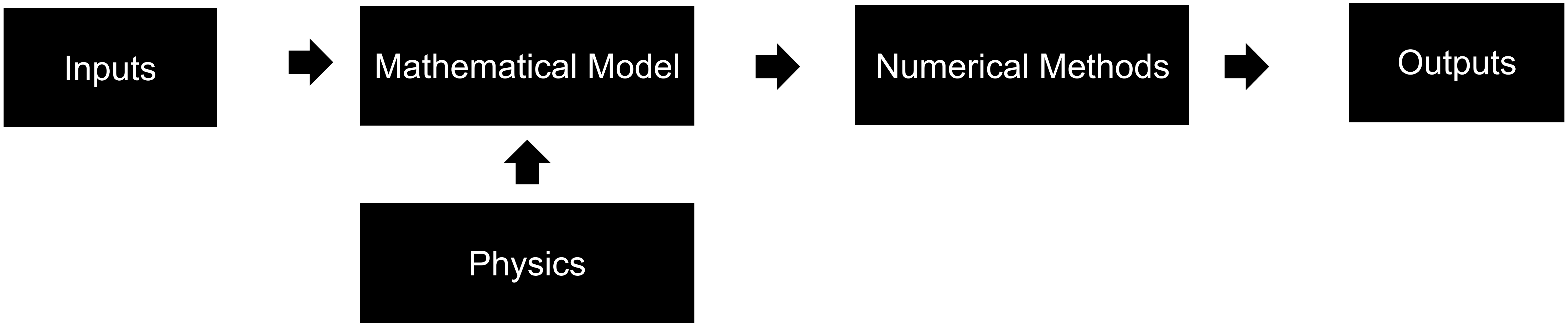
- Gravitational
- Aerodynamic
- Propulsive

⬅

Dependencies

- Atmosphere
- Geodesy

Flight Simulation Tool



[redacted]

Mathematical Model

Equations of Motion

- Translational Equations of Motion
- Rotational Equations of Motion

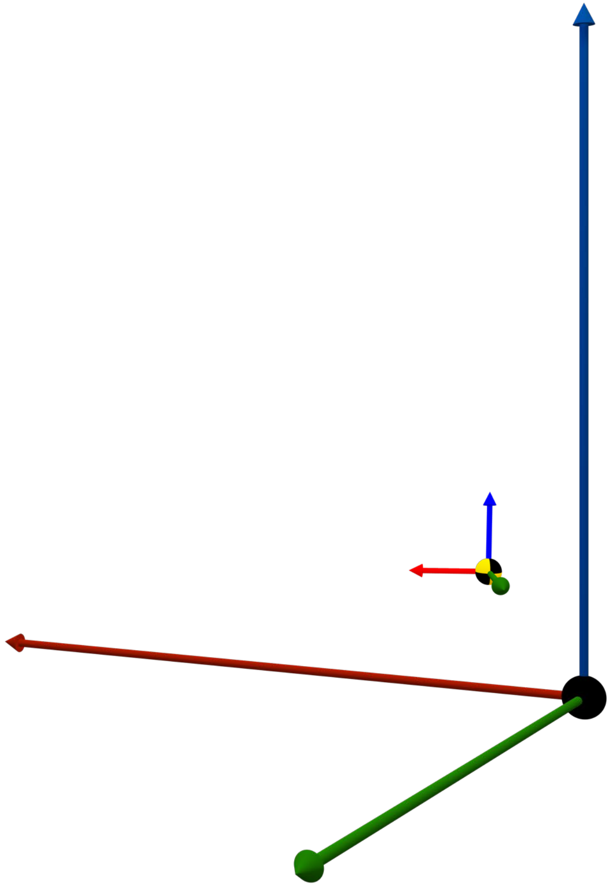
Mathematical Model

Equations of Motion

- Translational Equations of Motion
  - Kinetics:

$$\vec{F} = \frac{d}{dt}(m\vec{v}) = \mathrm{m}\frac{d\vec{v}}{dt} + \vec{v}\frac{dm}{dt} \qquad \frac{dm}{dt} = 0$$

  - Kinematics:
- Rotational Equations of Motion



Mathematical Model

Equations of Motion

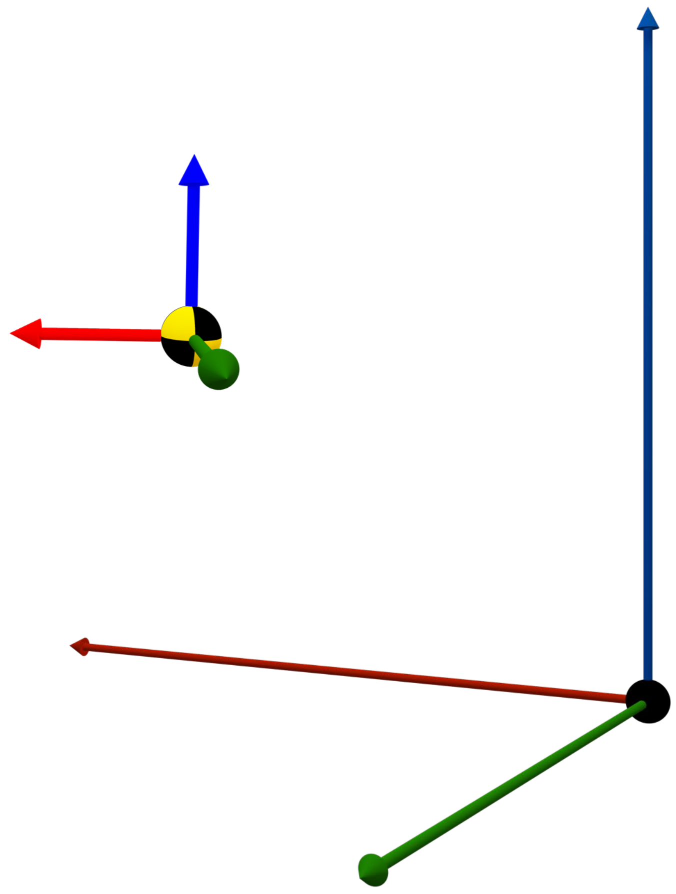
- Translational Equations of Motion
  - Kinetics:

$$\vec{F} = \frac{d}{dt}(m\vec{v}) = \mathrm{m}\frac{d\vec{v}}{dt} \qquad \frac{dm}{dt} = 0$$

  - Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

- Rotational Equations of Motion



Mathematical Model

Equations of Motion

Translational Equations of Motion

- Kinetics:

$$\vec{F} = \frac{d}{dt}(m\vec{v}) = \mathrm{m}\frac{d\vec{v}}{dt} \qquad \frac{dm}{dt} = 0$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

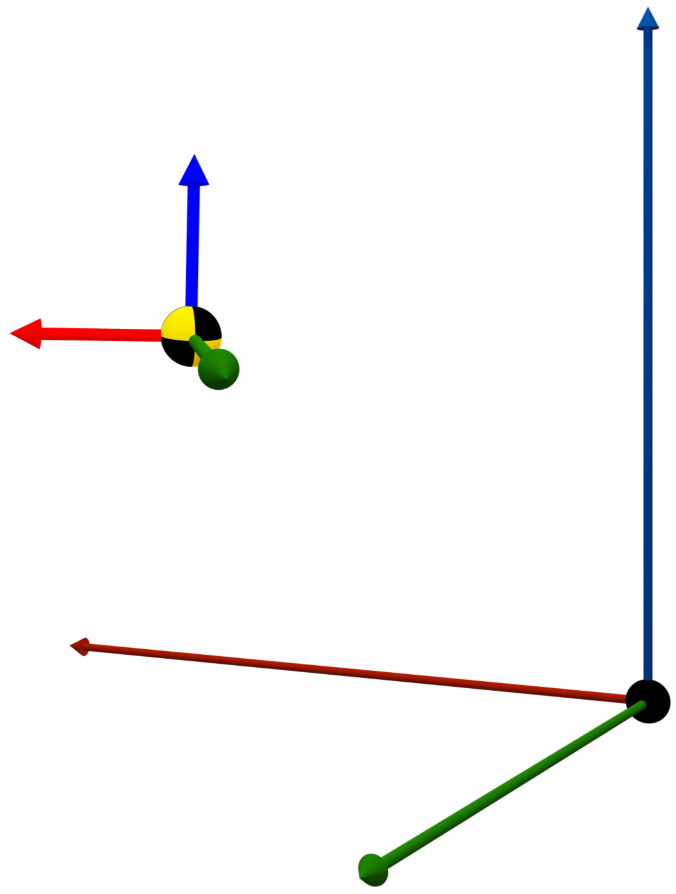
Rotational Equations of Motion

- Kinetics:

$$\vec{M} = \frac{d}{dt}(I\vec{\omega}) = \mathrm{I}\frac{d\vec{\omega}}{dt} + \vec{\omega}\frac{dI}{dt} \qquad \frac{dI}{dt} = 0$$

- Kinematics:

$$\frac{d\vec{\theta}}{dt} = \vec{\omega}$$



Translational Equations of Motion

- Kinetics:

$$\vec{F} = \mathrm{m}\frac{d\vec{v}}{dt}$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

Translational Equations of Motion

- Kinetics:

$$\vec{F}_{grav} + \vec{F}_{aero} + \vec{F}_{prop} = \mathrm{m}\frac{d\vec{v}}{dt}$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

Translational Equations of Motion

- Kinetics:

$$\vec{F} = \mathrm{m}\frac{d\vec{v}}{dt}$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

Mathematical Model

Equations of Motion

Translational Equations of Motion

- Kinetics:

$$\vec{F} = \mathrm{m}\frac{d\vec{v}}{dt}$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v}$$

Rotational Equations of Motion

- Kinetics:

$$\vec{M} = \frac{d}{dt}(I\vec{\omega}) = \mathrm{I}\frac{d\vec{\omega}}{dt} + \vec{\omega}\frac{dI}{dt} \qquad \frac{dI}{dt} = 0$$

- Kinematics:

$$\frac{d\vec{\theta}}{dt} = \vec{\omega}$$

Rotational Equations of Motion

- Kinetics:

$$\vec{M} = \mathrm{I}\frac{d\vec{\omega}}{dt}$$

- Kinematics:

$$\frac{d\vec{\theta}}{dt} = \vec{\omega}$$

Mathematical Model

Equations of Motion

Translational Equations of Motion

- Kinetics:

$$\vec{F} = m\frac{d\vec{v}}{dt} \qquad \vec{v}(t_0) = \vec{v}_0$$

- Kinematics:

$$\frac{d\vec{x}}{dt} = \vec{v} \qquad \vec{x}(t_0) = \vec{x}_0$$

Rotational Equations of Motion

- Kinetics:

$$\vec{M} = I\frac{d\vec{\omega}}{dt} \qquad \vec{\omega}(t_0) = \vec{\omega}_0$$

- Kinematics:

$$\frac{d\vec{\theta}}{dt} = \vec{\omega} \qquad \vec{\theta}(t_0) = \vec{\theta}_0$$

Mathematical Model

Equations of Motion

Translational Equations of Motion

$$\frac{d\vec{x}}{dt} = \vec{v}$$

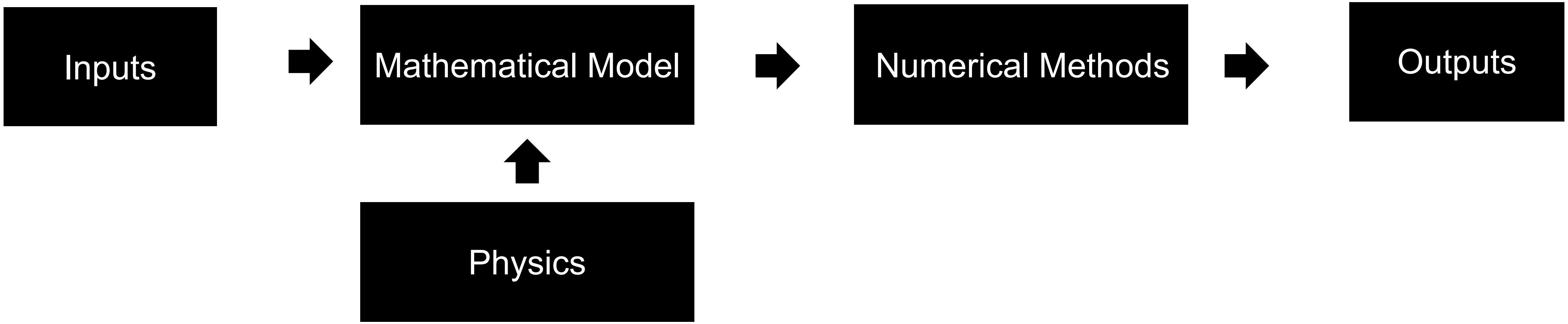
(12)

(13)

(14)

$$\frac{d\vec{v}}{dt} = \frac{\vec{F}}{m} + (non - intertial\ terms)$$

Flight Simulation Tool



Numerical Methods

NUMERICAL METHODS 101

Numerical integrators

- Solves the differential equation at each timestep

INITIAL CONDITIONS

$\vec{x}(t_0) = \vec{x}_0$

$\vec{v}(t_0) = \vec{v}_0$

$\vec{\omega}(t_0) = \vec{\omega}_0$

$\vec{\theta}(t_0) = \vec{\theta}_0$

➡

CHANGE IN STATE

CURRENT TIMESTEP

$\vec{x}(t_i) = \vec{x}_i$

$\vec{v}(t_i) = \vec{v}_i$

$\vec{\omega}(t_i) = \vec{\omega}_i$

$\vec{\theta}(t_i) = \vec{\theta}_i$

+

$\Delta t \dfrac{d\vec{x}}{dt}$

$\Delta t \dfrac{d\vec{v}}{dt}$

$\Delta t \dfrac{d\vec{\theta}}{dt}$

$\Delta t \dfrac{d\vec{\omega}}{dt}$

=

NEXT TIMESTEP

$\vec{x}(t_{i+1}) = \vec{x}_0$

$\vec{v}(t_{i+1}) = \vec{v}_0$

$\vec{\omega}(t_{i+1}) = \vec{\omega}_0$

$\vec{\theta}(t_{i+1}) = \vec{\theta}_0$

➡

FINAL STATE

$\vec{x}(t_f) = \vec{x}_f$

$\vec{v}(t_f) = \vec{v}_f$

$\vec{\omega}(t_f) = \vec{\omega}_f$

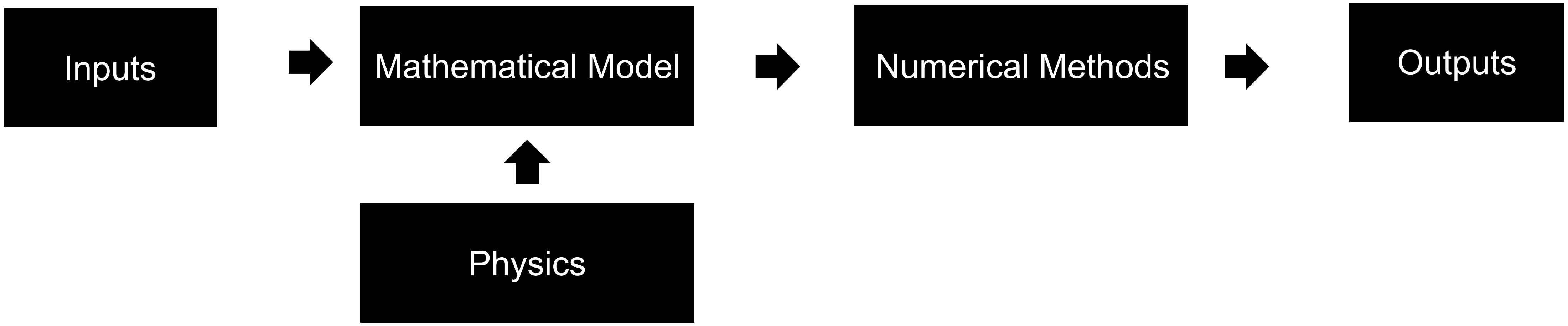
$\vec{\theta}(t_f) = \vec{\theta}_f$

NUMERICAL METHODS 101

Numerical integrators

- Selection criteria
  - Accuracy
  - Behavior of differential equation(s)
  - Computational Cost
- Examples
  - Runge-Kutta 4
    - Fixed time step
    - Explicit
    - Fourth Order Accurate
  - Dormand-Prince 5
    - Adaptive time step
    - Explicit
    - Fifth Order Accurate

Flight Simulation Tool



Inputs

Flight Simulation Tool

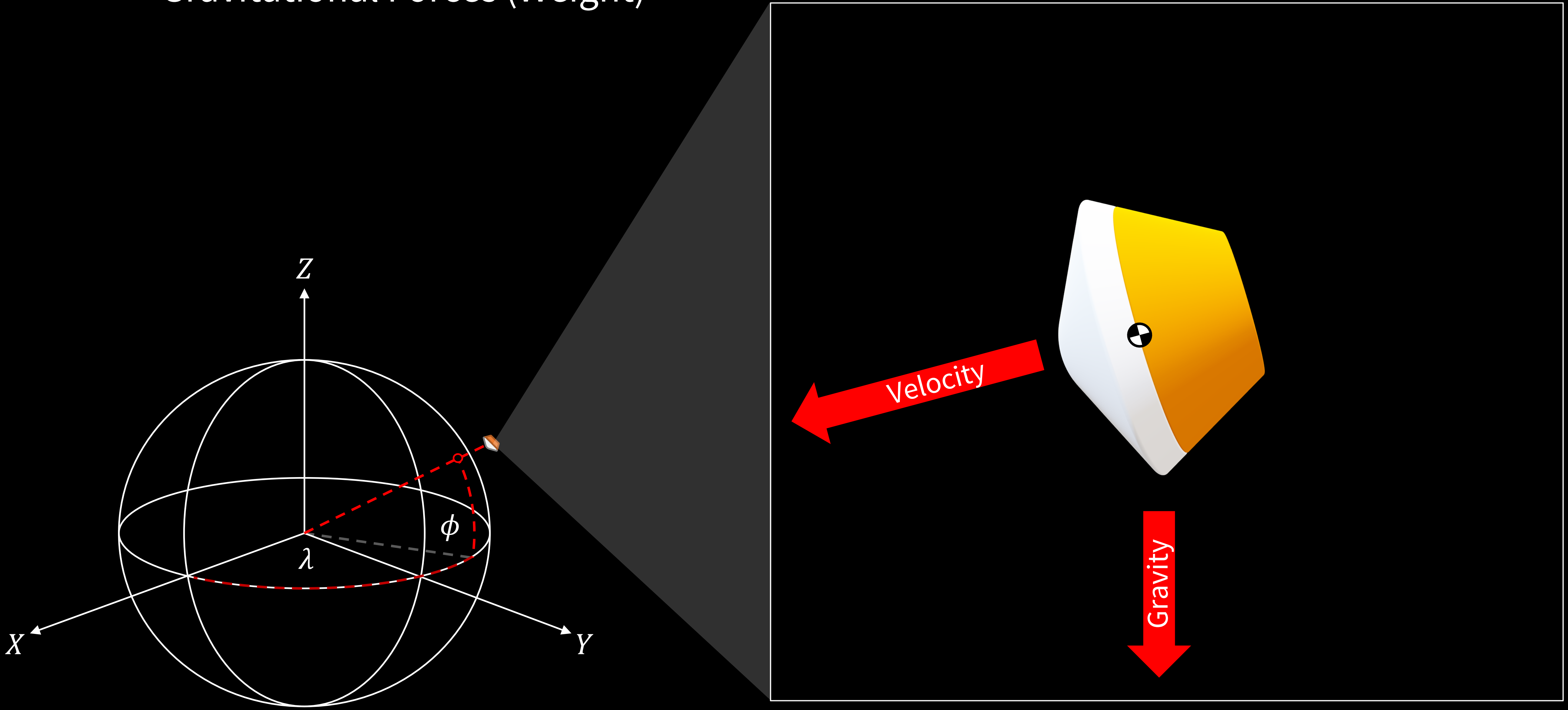
Outputs

Stardust Capsule Return (2006)

- Ballistic Reentry
  - Assume rotational dynamics are taken care of
  - 3 DOF Simulation
  - Required Inputs:
    - Mass
    - Forces
    - Initial position and velocity

Modelling the Forces

- In vacuum
  - Gravitational Forces (Weight)

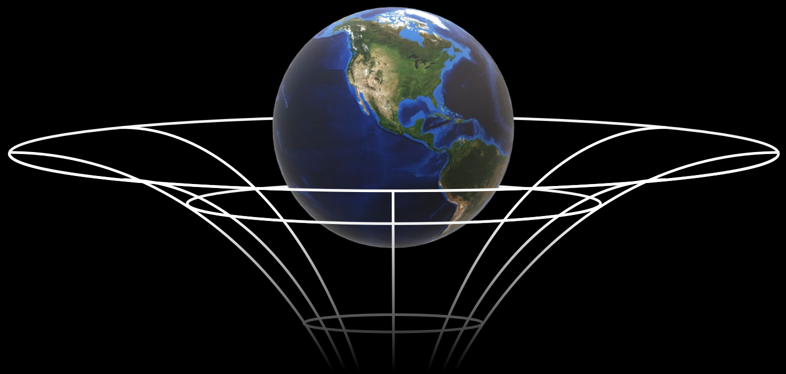


Gravity Models

- Constant Gravity

$$|\vec{g}| = 9.81\ m/s$$

- Inverse Square



$$\vec{g} = -\mu \frac{\vec{r}}{|\vec{r}|^3}$$

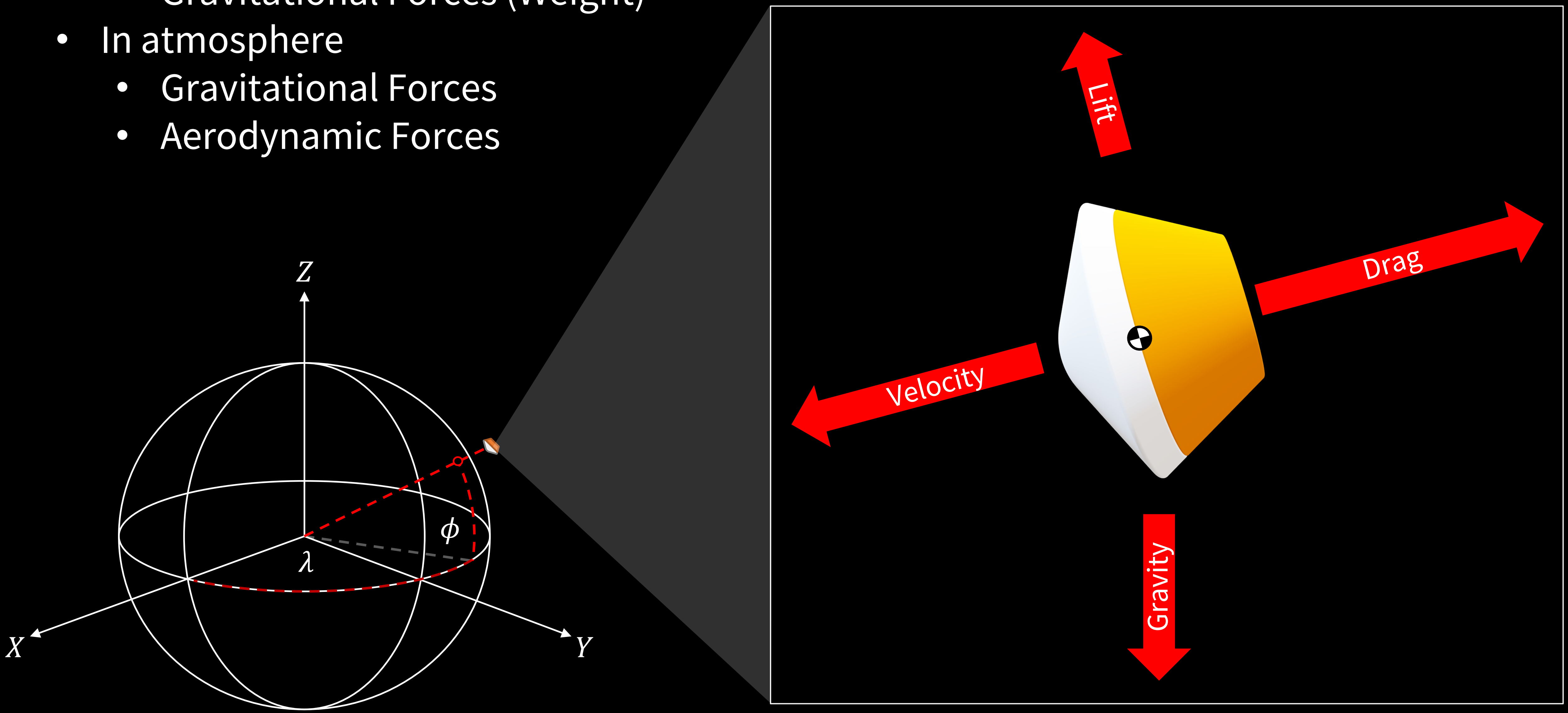
- Spherical Harmonic Gravitation

$$\vec{g} = \nabla \vec{U}$$

$$U(r, \lambda, \phi) = \frac{\mu}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{R}{r}\right)^n \bar{P}_{n,m}(sin\phi)(\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda)$$

Modelling the Forces

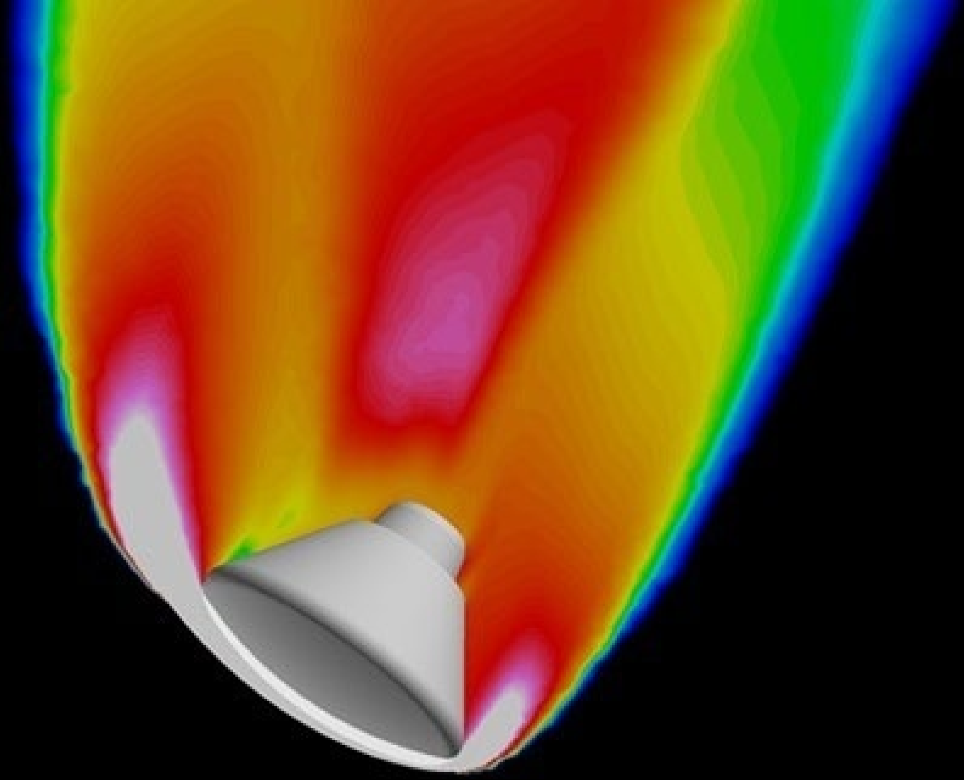
- In vacuum
  - Gravitational Forces (Weight)
- In atmosphere
  - Gravitational Forces
  - Aerodynamic Forces



Aerodynamic Models

- Expressed in coefficients
  - $L = C_L \cdot \frac{1}{2} \rho V^2 S$
- Aerodynamic Coefficients
  - Lift and Drag (C_L, C_D)
  - Aerodynamic moments (C_l, C_m, C_n)
  - Dynamic coefficients ($C_{m_q}, \ldots$)
  - …
  - And many more!

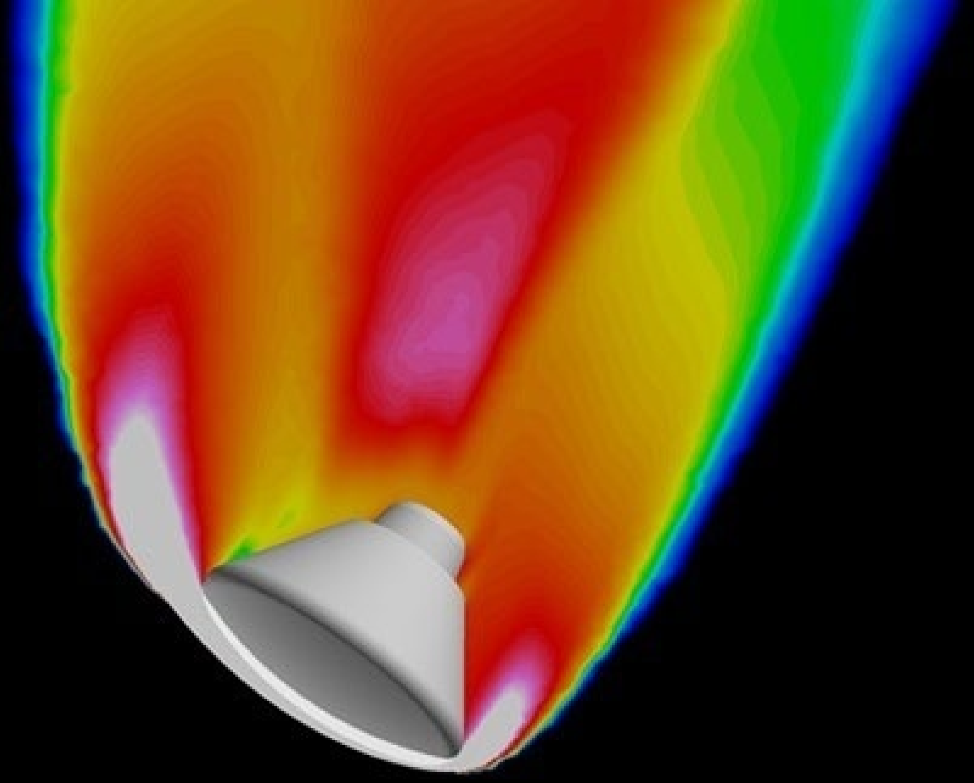
- Fidelity depends on
  - How the data was obtained
  - What conditions are being considered



Aerodynamic Models

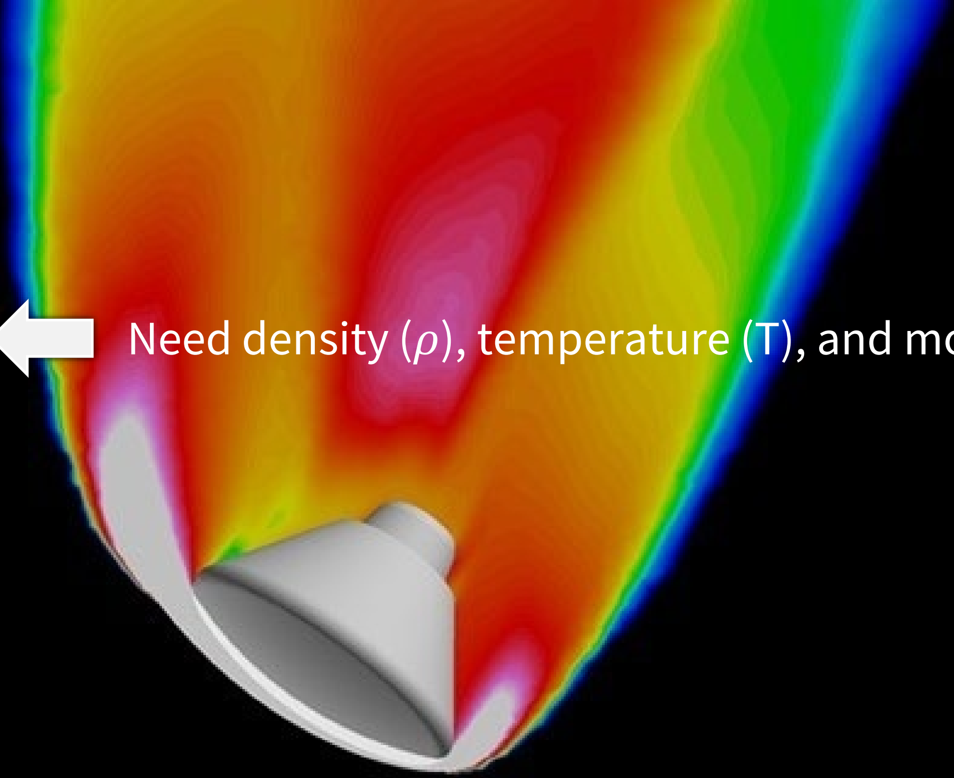
- Expressed in coefficients
  - $L = C_L(\alpha, Re, Ma) \cdot \frac{1}{2}\rho V^2 S$
- Aerodynamic Coefficients
  - Lift and Drag (C_L, C_D)
  - Aerodynamic moments (C_l, C_m, C_n)
  - Dynamic coefficients ($C_{m_q}, \dots$)
  - $\dots$
  - And many more!

- Fidelity depends on
  - How the data was obtained
  - What conditions are being considered



Aerodynamic Models

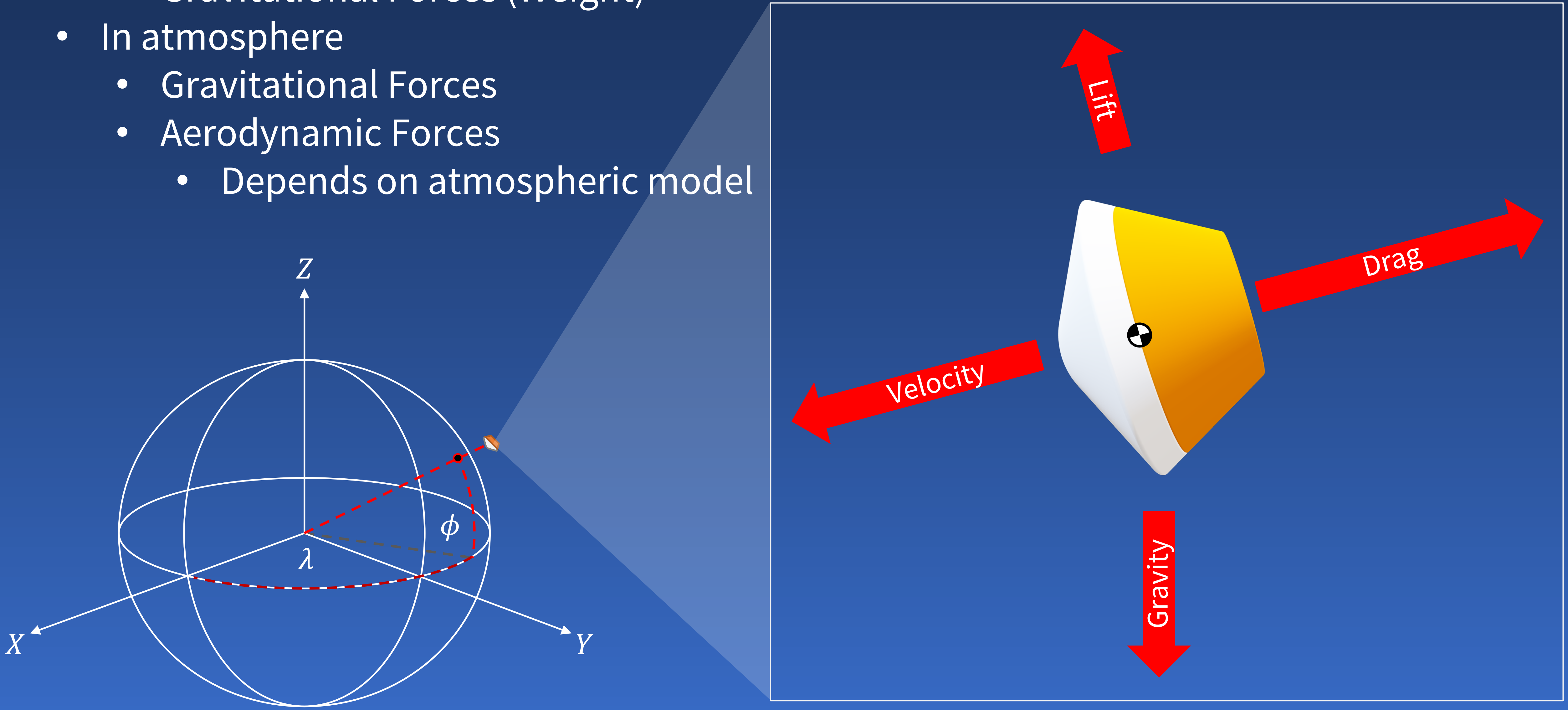
- Expressed in coefficients
  - $L = C_L(\alpha, Re(\rho, \mu), Ma(\gamma, R, T)) \cdot \frac{1}{2}\rho V^2 S$ ← Need density (ρ), temperature (T), and more…
- Aerodynamic Coefficients
  - Lift and Drag (C_L, C_D)
  - Aerodynamic moments (C_l, C_m, C_n)
  - Dynamic coefficients (C_{m_q}, …)
  - …
  - And many more!



- Fidelity depends on
  - How the data was obtained
  - What conditions are being considered

Modelling the Forces

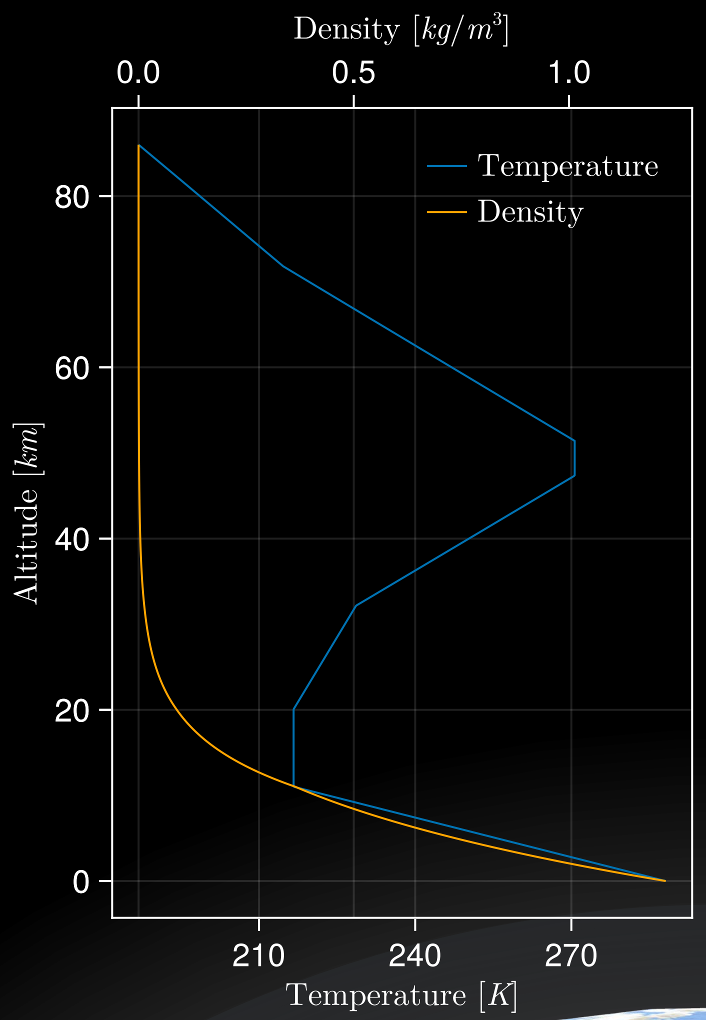
- In vacuum
  - Gravitational Forces (Weight)
- In atmosphere
  - Gravitational Forces
  - Aerodynamic Forces
    - Depends on atmospheric model



Atmospheric Models

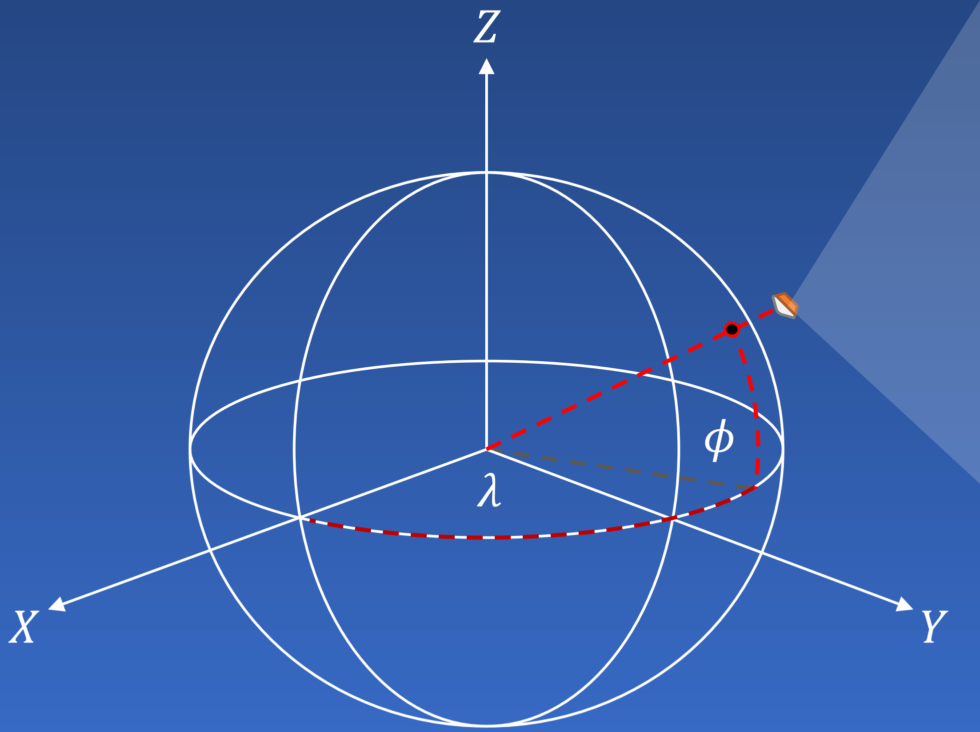
- 1976 Standard Atmospheric Model
  - Linear, piecewise variation in temperature
  - Solves for pressure and density using hydrostatic equations

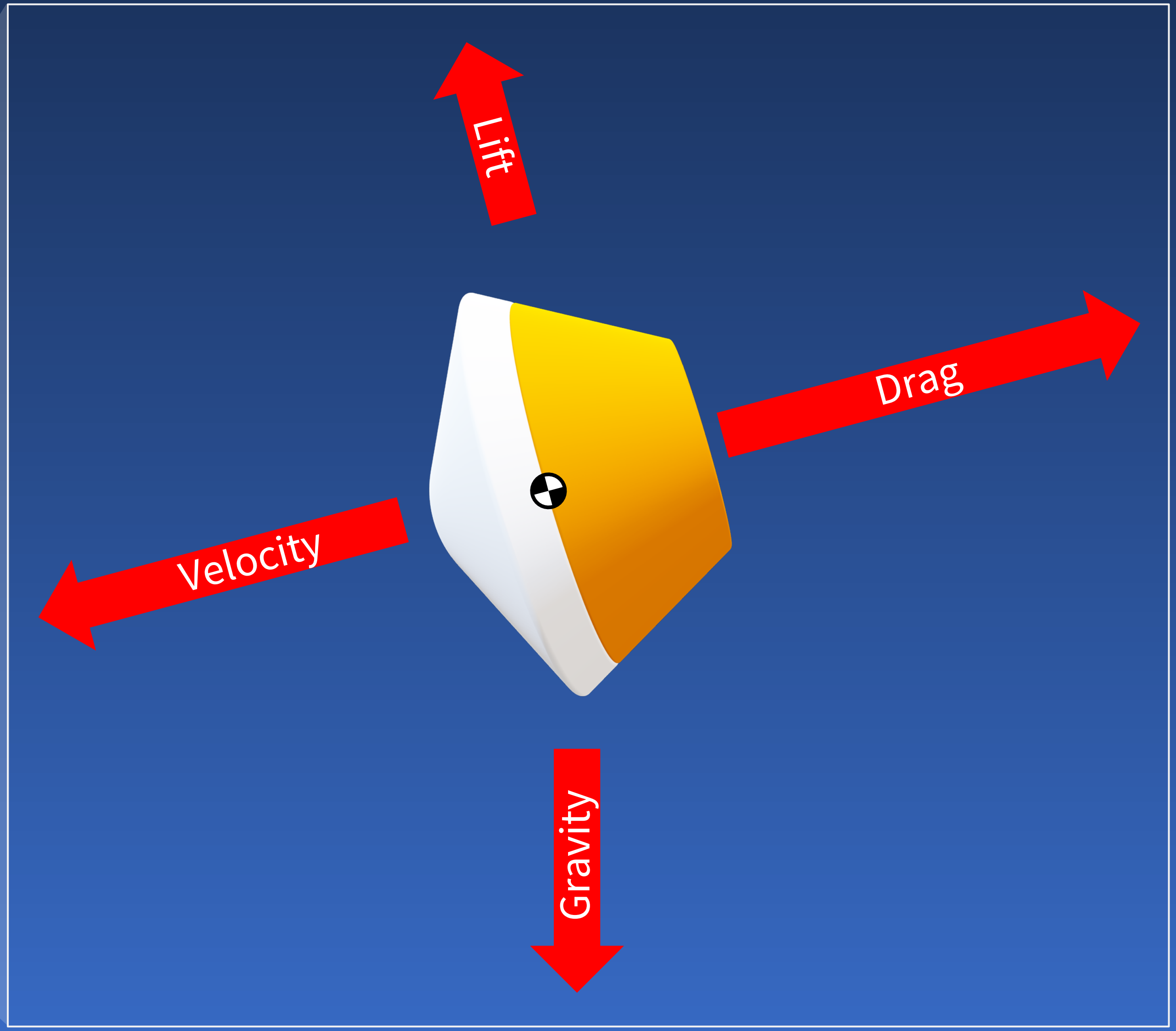
- Global Reference Atmospheric Model
  - Variations in seasons
  - Statistical variations can be modelled



Modelling the Forces

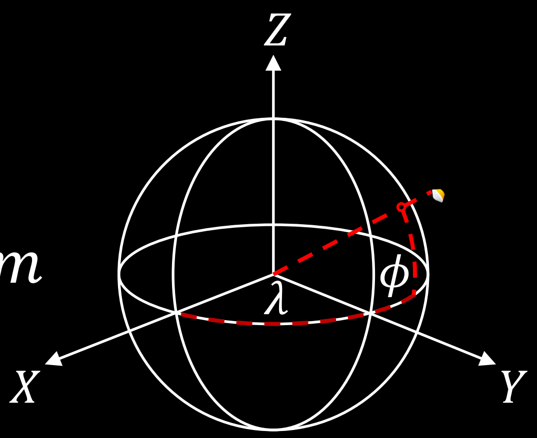
- In vacuum
  - Gravitational Forces (Weight)
- In atmosphere
  - Gravitational Forces
  - Aerodynamic Forces
    - Depends on atmospheric model
      - Depends on geodesy model

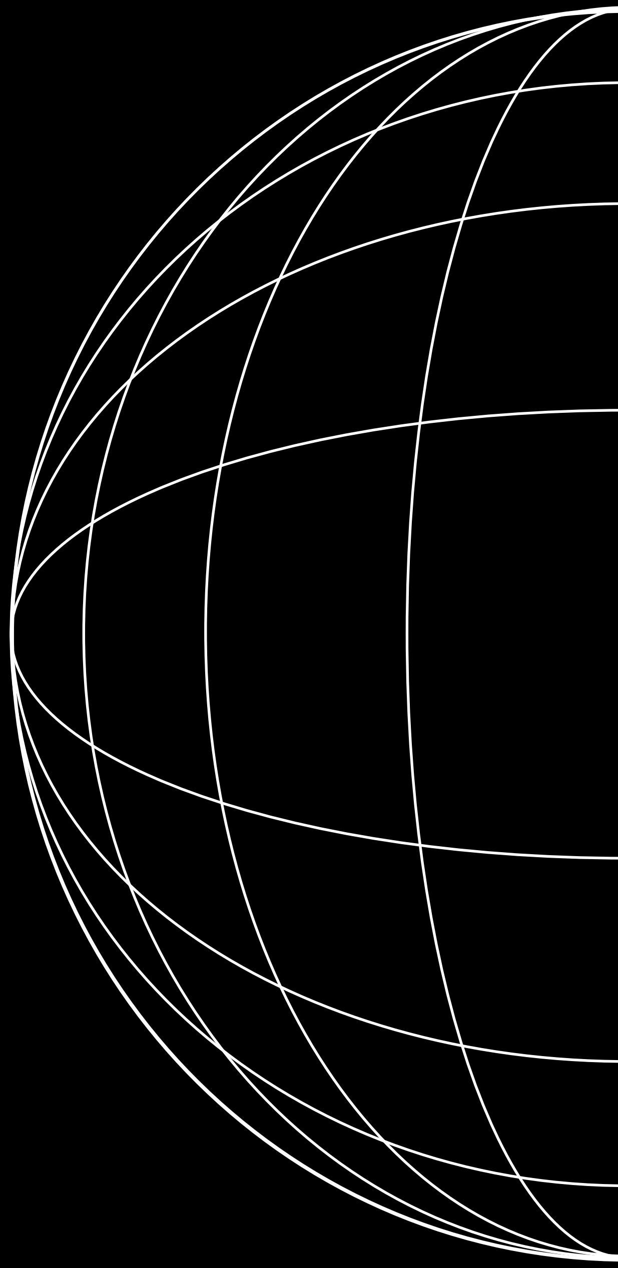




Geodesy Models

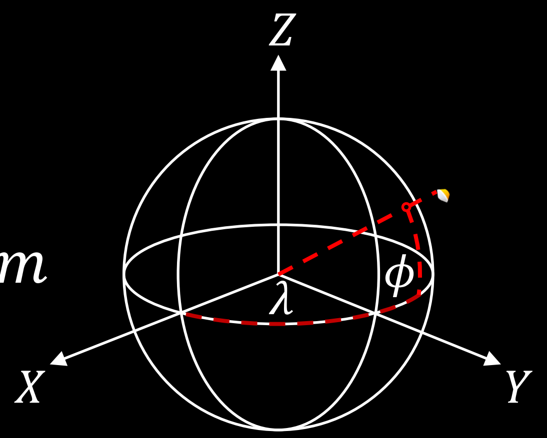
- Sphere
  - $r_s = 6{,}371{,}007.1809\ m$



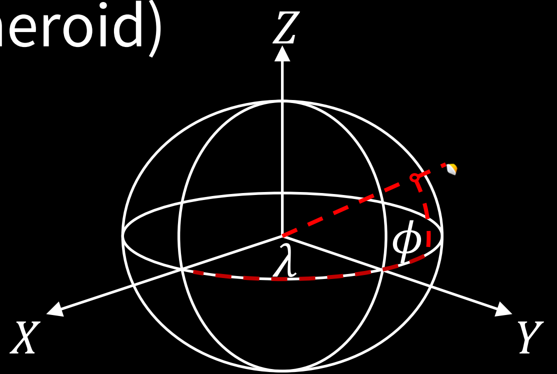


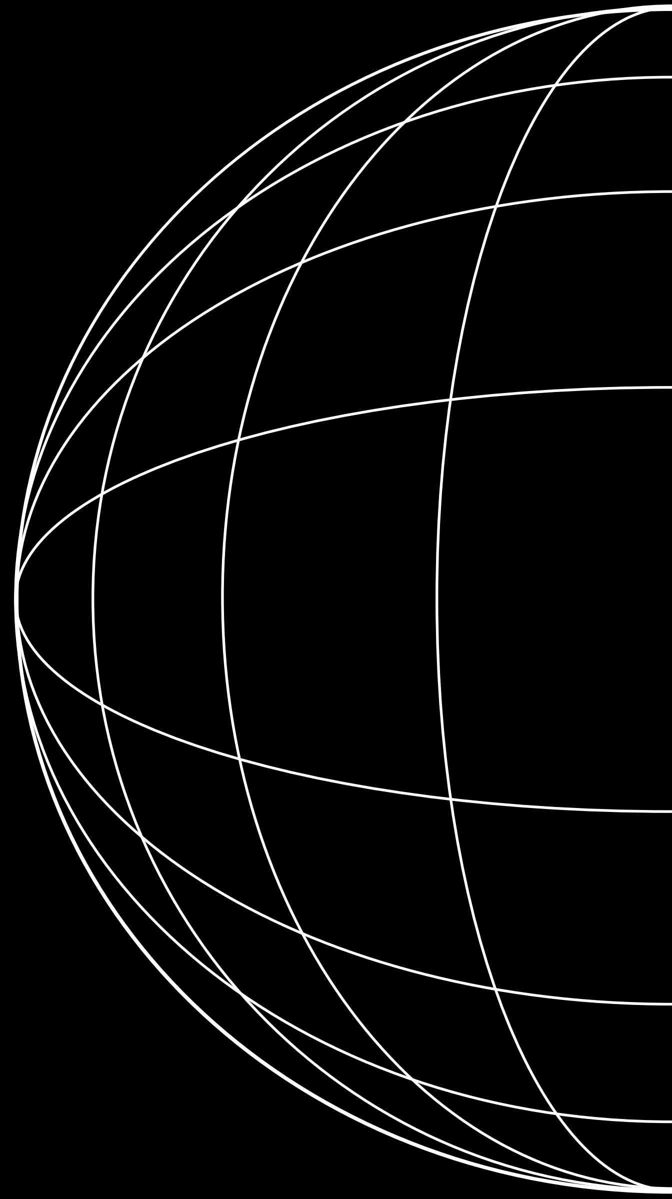
Geodesy Models

- Sphere
  - $r_s = 6{,}371{,}007.1809\ m$



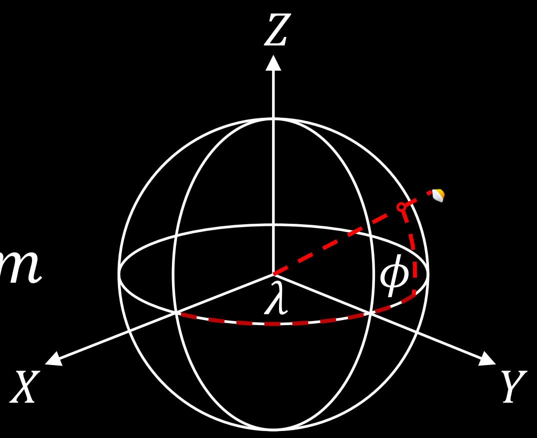
- Ellipsoid (WGS84/Oblate Spheroid)
  - $r_e = 6{,}378{,}137.0\ m$
  - $r_p = 6{,}356{,}752.3\ m$



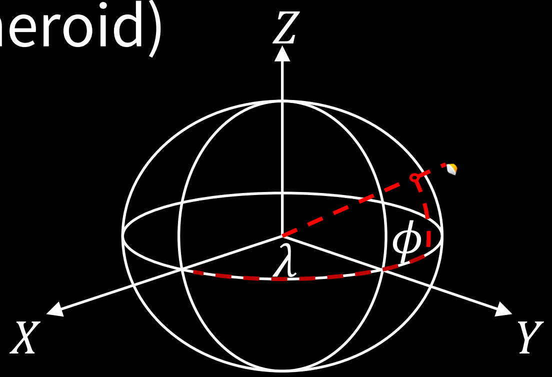


Geodesy Models

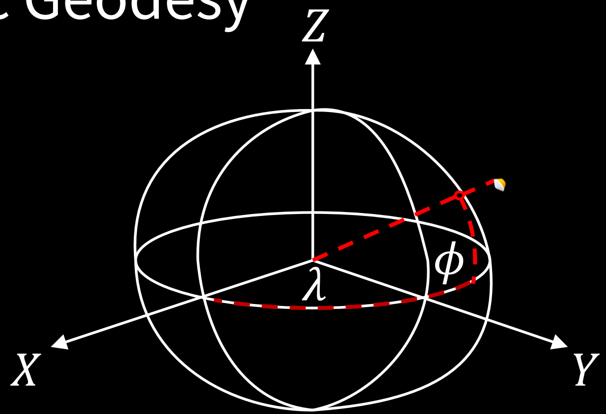
- Sphere
  - $r_s = 6{,}371{,}007.1809\ m$

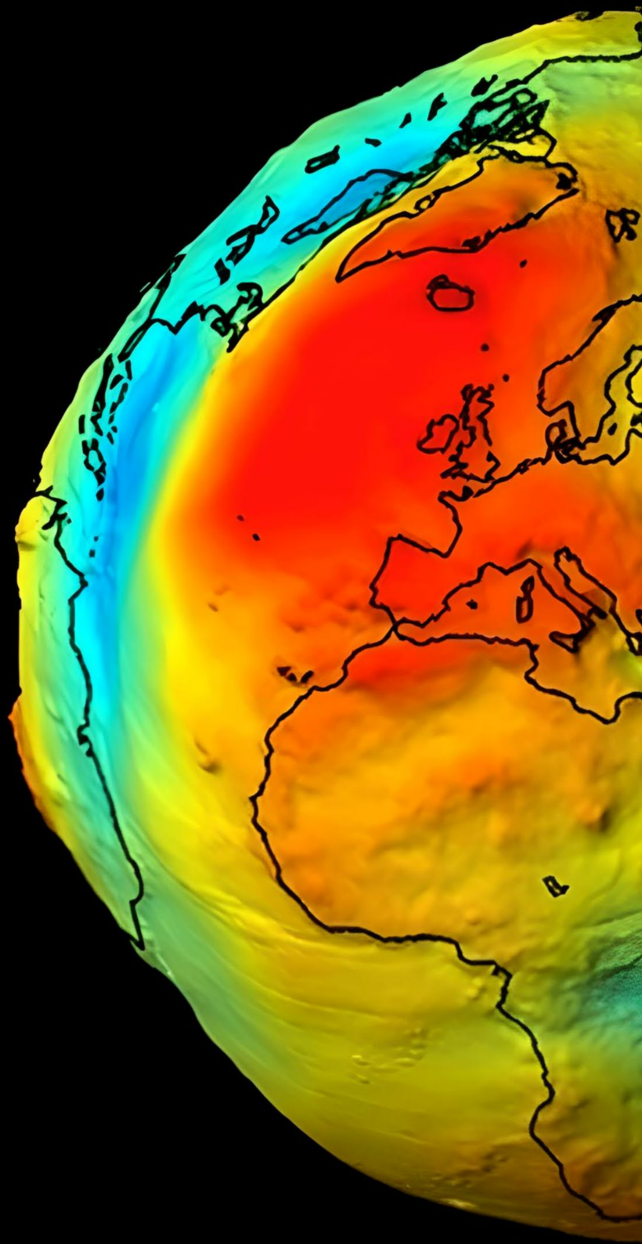


- Ellipsoid (WGS84/Oblate Spheroid)
  - $r_e = 6{,}378{,}137.0\ m$
  - $r_p = 6{,}356{,}752.3\ m$



- Spherical Harmonic Geodesy





Stardust Capsule Return Summary

Assumptions: ballistic trajectory, 3DOF, constant ballistic coefficient

Mass: 45.76 kg

Entry Altitude: 134.441696 km
Entry Geodetic Latitude: 41.7529875 deg
Entry Longitude: 236.079820 deg

Entry Velocity: 12.799359 km/s (inertial), 12.4506423 km/s (relative)
Entry Angle: -8.234229 (inertial), -8.46652246 (relative)
Entry Heading: 96.536837 deg (inertial), 96.7247718 deg (relative)

Numerical Integrator: RK4, $\Delta t = 0.05$ s

Recommended Models:

- **Gravity Model:** J2
- **Aerodynamic Model:** Constant Cd from Ballistic Coefficient
  - **Ballistic Coefficient:** 60 kg/m^2
- **Atmospheric Model:** 1976 US Standard Atmosphere
- **Geodesy Model:** WGS84

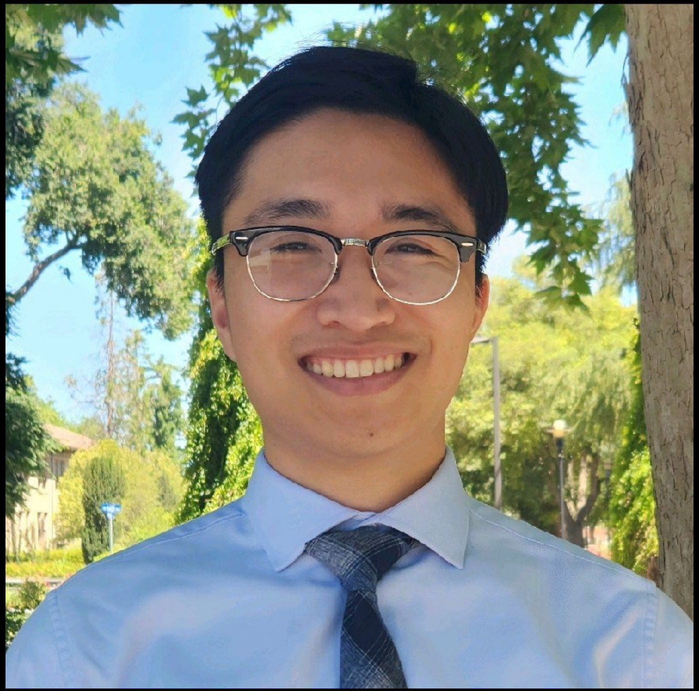
THIS MODULE WAS SPONSORED BY

NESC Flight Mechanics Technical Discipline Team

HEATHER KOEHLER
NASA Technical Fellow, NASA Marshall Space Flight Center

DEVELOPED AND PRESENTED BY

CARLOS ANTHONY NATIVIDAD
Entry Systems and Vehicle Development Branch Intern



IN COLLABORATION WITH

NASA Ames Research Center

SARAH D'SOUZA
NESC Flight Mechanics TDT Deputy, Mentor

GARY ALLEN
TRAJ Developer Lead, Analytical Mechanics Associates, Inc.

JAYME BERSTELL
Entry Systems and Technology Branch Technical Lead
Analytical Mechanics Associates, Inc.

NASA Glenn Research Center

ROBERT FALCK
OpenMDAO Development Team Lead

NASA Johnson Space Center

DANIEL MATZ
Flight Dynamics Modeling Technical Discipline Lead

BREANNA JOHNSON
Guidance, Navigation & Control Lead, Planetary Entry Vehicle Design

DAVID HOFFMAN
Flight Mechanics Modelling and Analysis Engineer, Jacobs Engineering

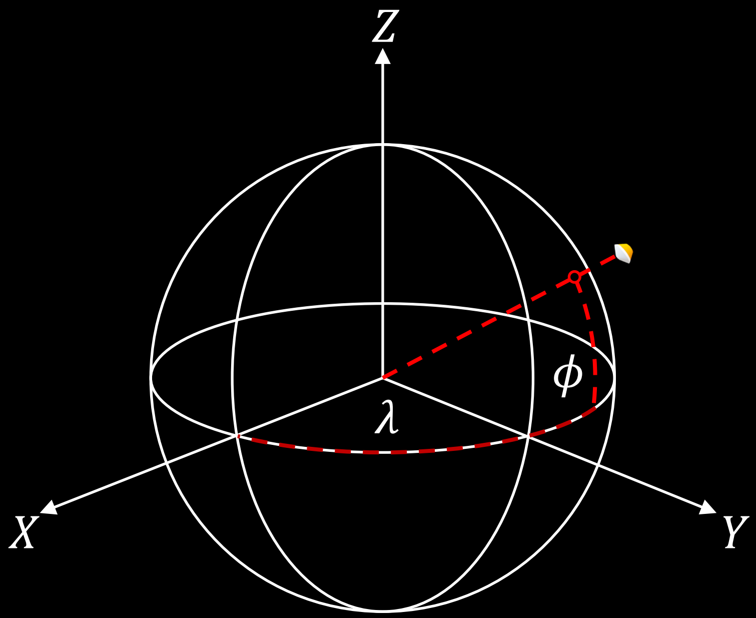
NASA Langley Research Center

RAFAEL LUGO
AST for Human-Scale EDL Simulation Architectures
Lead, POST2 Development Team

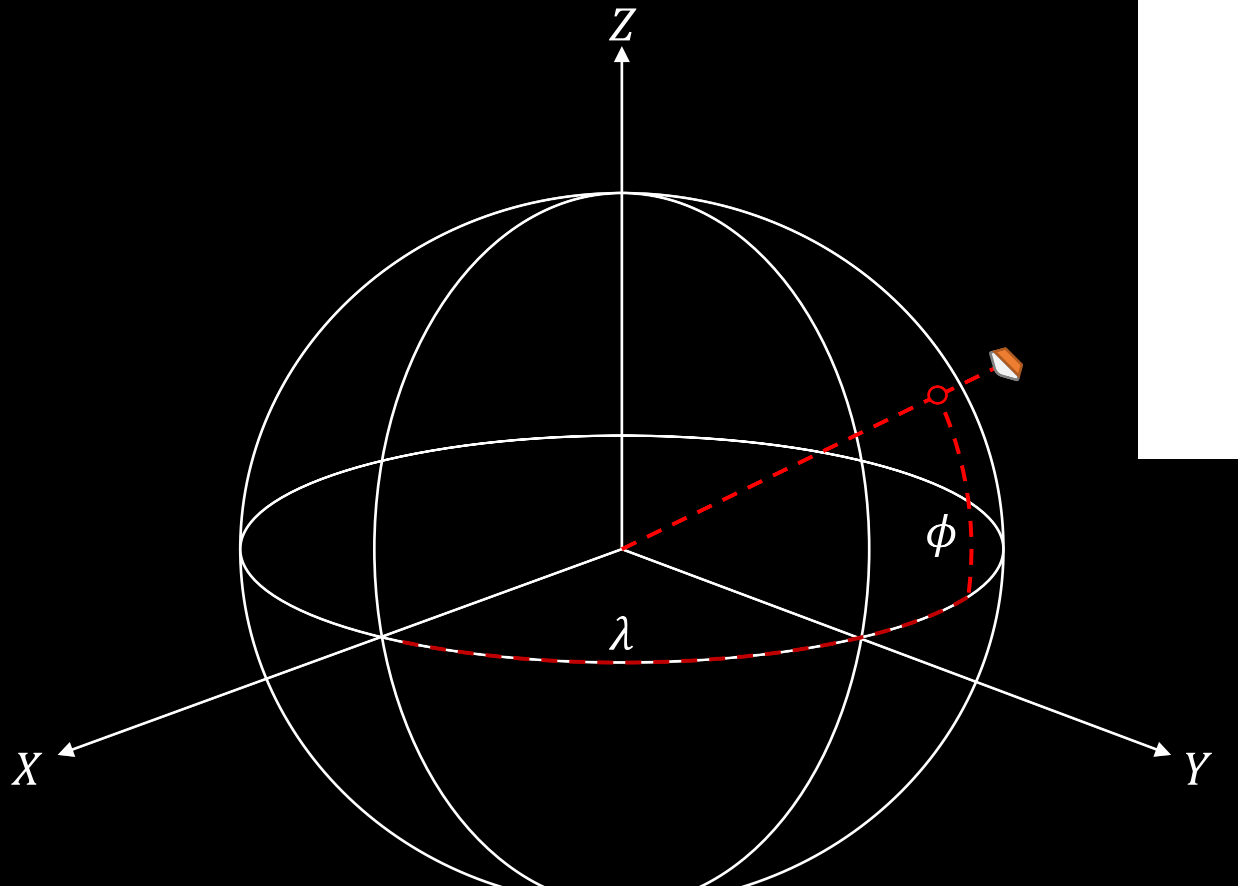
SOUMYO DUTTA
Flight Mechanics Simulation and Modelling Lead, Planetary Entry Systems

REFERENCES

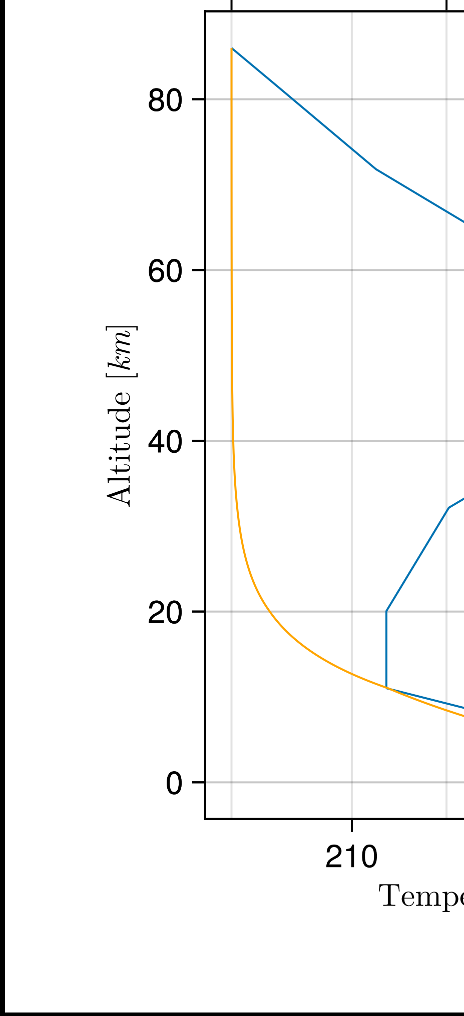
1. Murri, Daniel G.; Jackson, E. Bruce; and Shelton, Robert O.: *Check-Cases for Verification of 6-Degree-of-Freedom Flight Vehicle Simulations*. NASA TM-2015-218675, National Aeronautics and Space Administration, Hampton, VA, 2015.
2. Beaty, J. R.: *Fundamentals of Geodetic Kinematics*. May 2014. URL https://nescacademy.nasa.gov/video/4c830bb2f3844670a0cf4ec9390407af1d.
3. Parcero, Kayla; Allen, Gary; Witkowski, Al; McKee, Sharon; and Torres, Leah: *PLANETARY MISSION ENTRY VEHICLES Quick Reference Guide Version 4.0*. National Aeronautics and Space Administration, Moffett Field, CA, 2015.
4. Allen, Gary; Wright, Michael J.; and Gage, Peter: *The Trajectory Program (TRAJ): Reference Manual and User's Guide*. National Aeronautics and Space Administration, Moffett Field, CA, 2005.
5. Matz, Daniel A.; *Genesis.jl*. National Aeronautics and Space Administration, Houston, TX, 2018.
6. Lugo, Rafael; *Program to Optimize Simulated Trajectories (POST) Overview*. September 2016. URL https://nescacademy.nasa.gov/video/512a0772cb9a449d806b9e751eeb52141d.
7. Takahashi, Yusuke; Nakasato, Reo; and Oshima, Nobuyuki: *Analysis of Radio Frequency Blackout for a Blunt-Body Capsule in Atmospheric Reentry Missions*, *Aerospace*, vol. 3, a. 2.
8. Zipfel, Peter H.: *Modelling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics, Reston, VA, second ed., 2007.
9. Vinh, Nguyen. X.; Busemann, Adolf; Culp, Robert D: *Hypersonic and Planetary Entry Flight Mechanics*. University of Michigan Press, Ann Harbor, MI, 1980.



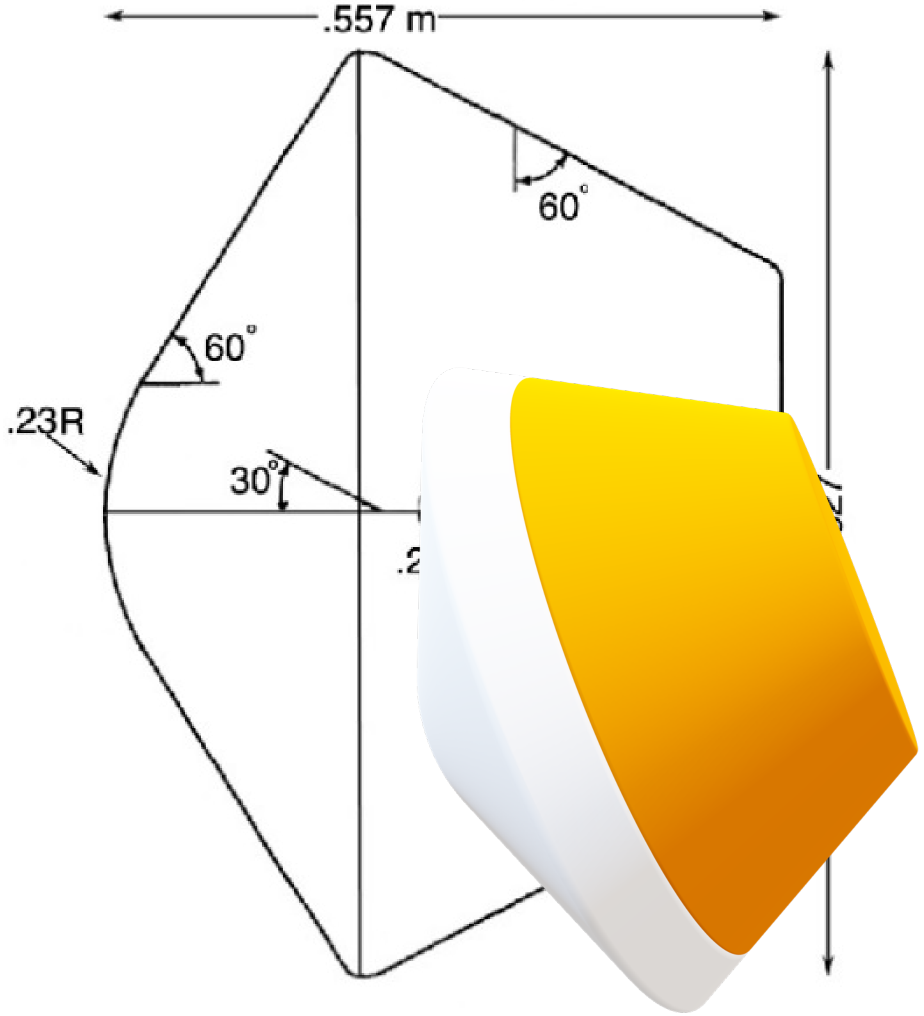
Z
X
Y
λ
ϕ



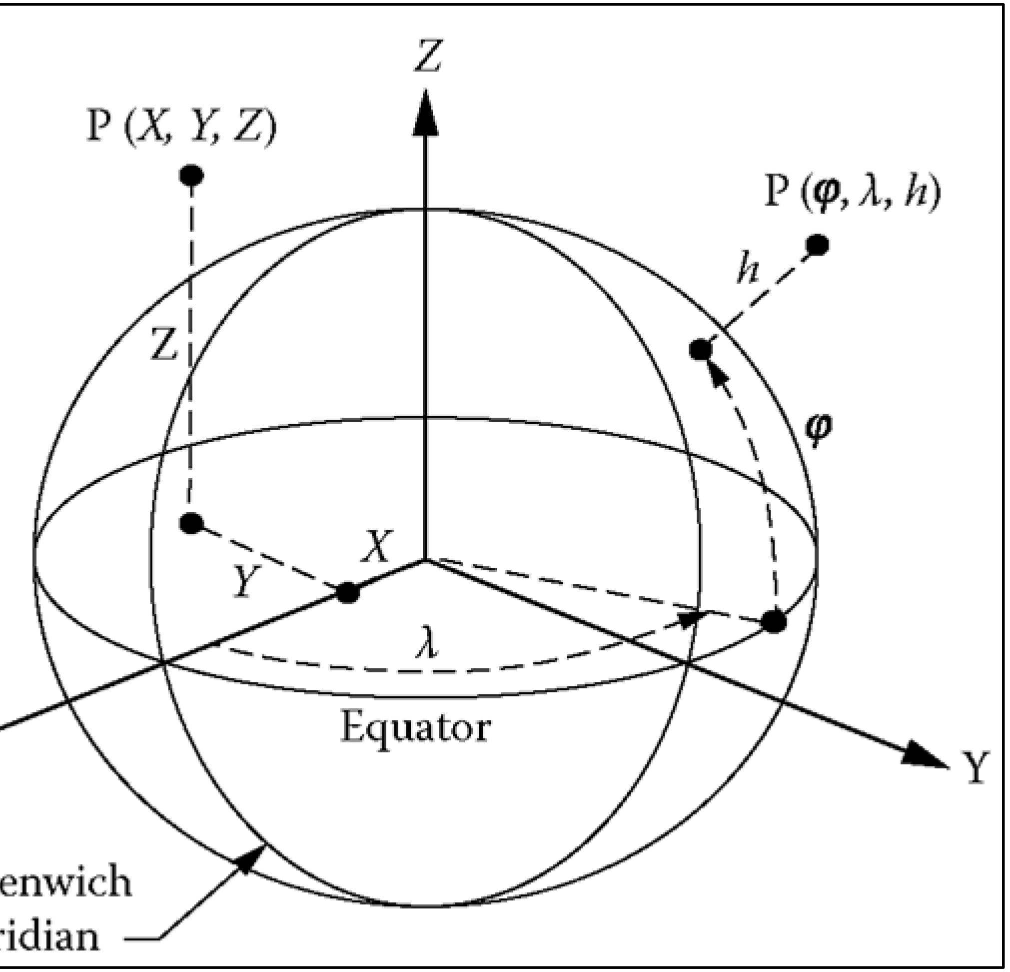
Z
X
Y
λ
ϕ

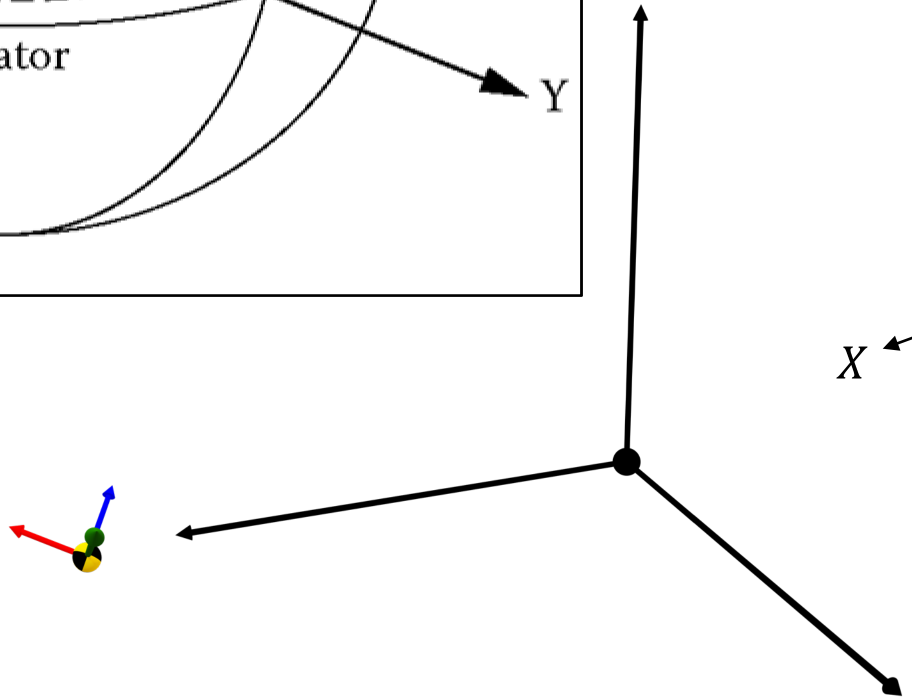


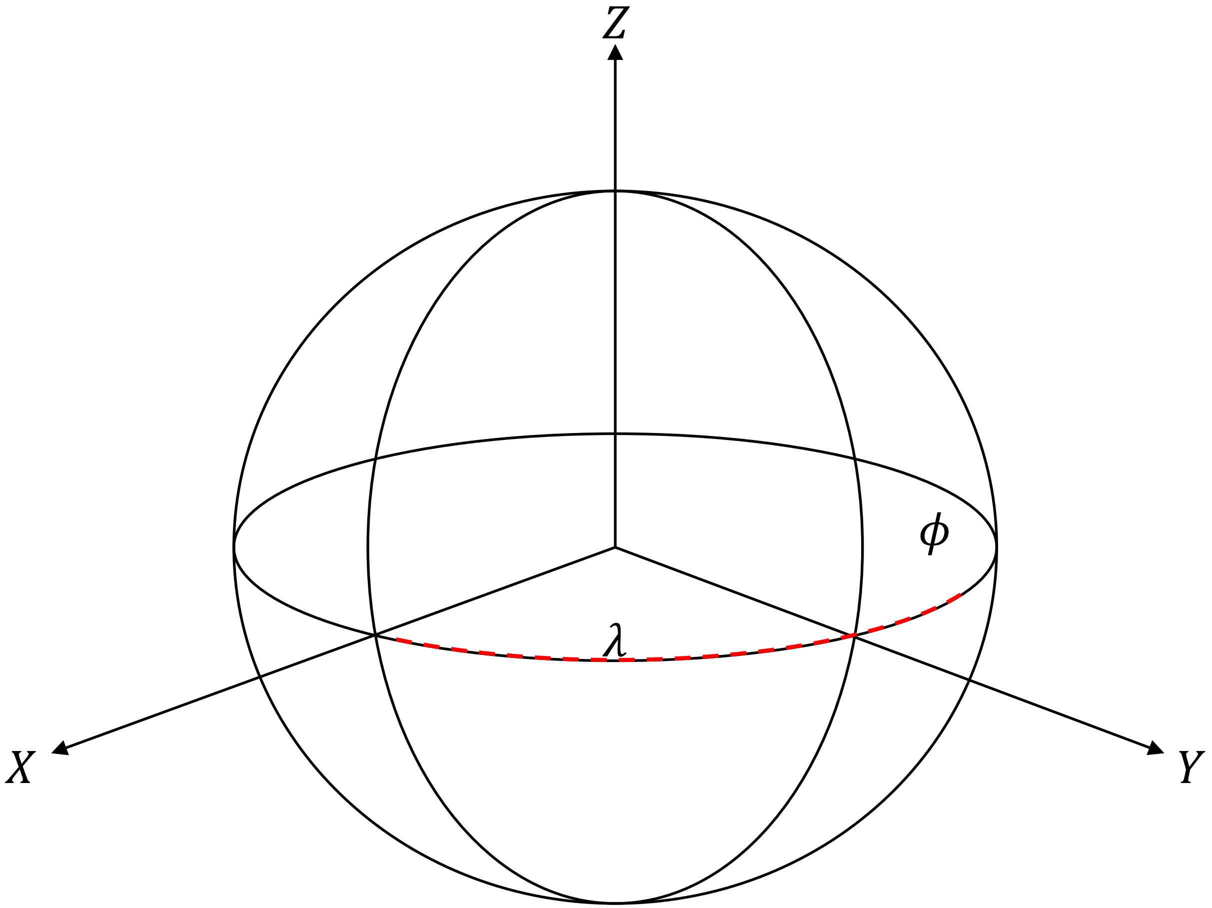
80
60
40
20
0
Altitude [km]
210

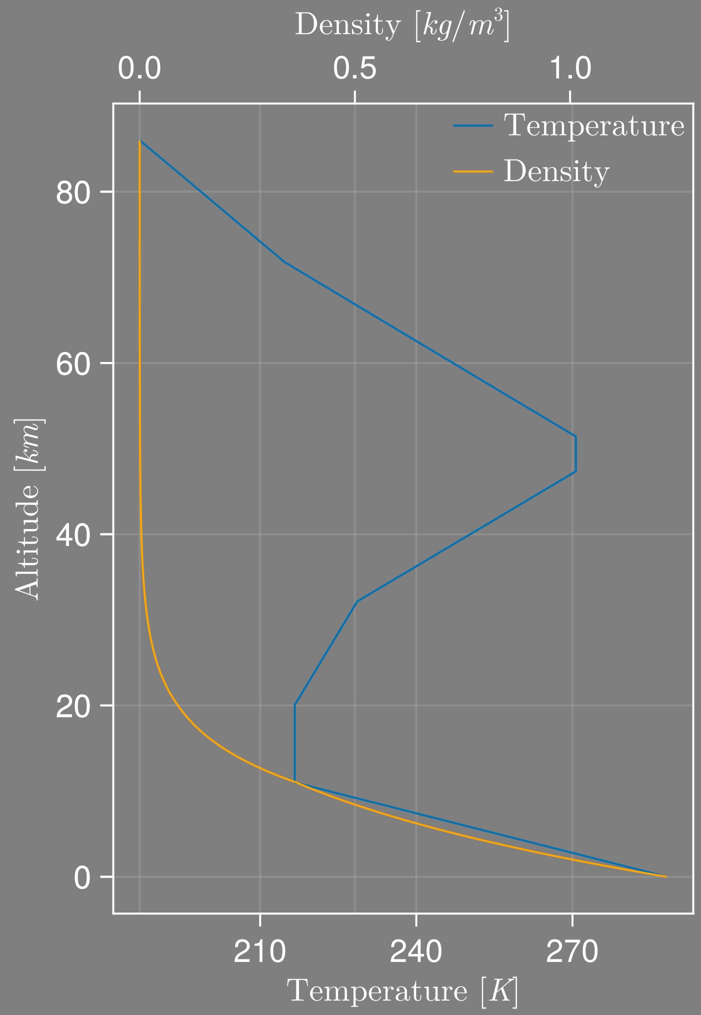


.557 m
60°
60°
.23R
30°


Z
P (X, Y, Z)
P (φ, λ, h)
h
Z
φ
X
Y
λ
Equator
Y
enwich
ridian




Z
ϕ
λ
X
Y


Density [kg/m³]
0.0
0.5
1.0
Temperature
Density
80
60
40
20
0
Altitude [km]
210
240
270
Temperature [K]