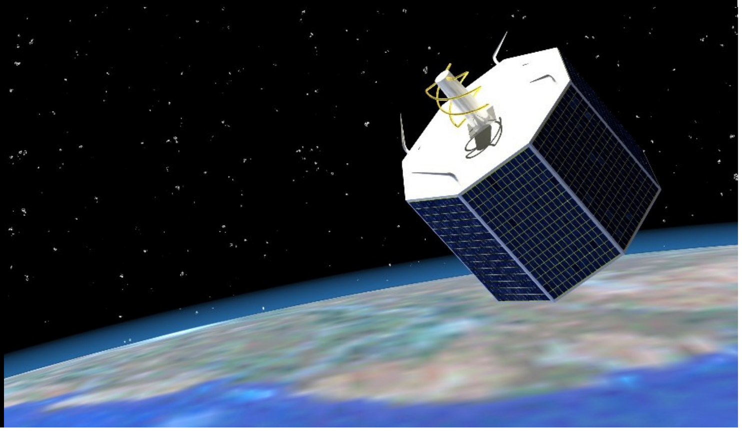
An Open Source Satellite Attitude and Orbit Simulator Toolbox for Matlab

Valdemir Carrara

valdemir.carrara@inpe.br

diname 2015



Overview

Introduction

PROPAT Toolbox

Attitude Control Simulation

CONASAT

Boomerang motion

Conclusions

Introduction

Heritage from CCS Fligth Dynamics software at INPE

Attitude simulator in C++ to support AOCS with HIL software

PROPAT was initially coded in 1998

It is still being improved with new functions

MATLAB pros and cons

PROPAT is aimed for academic use

Similar Matlab toolboxes:

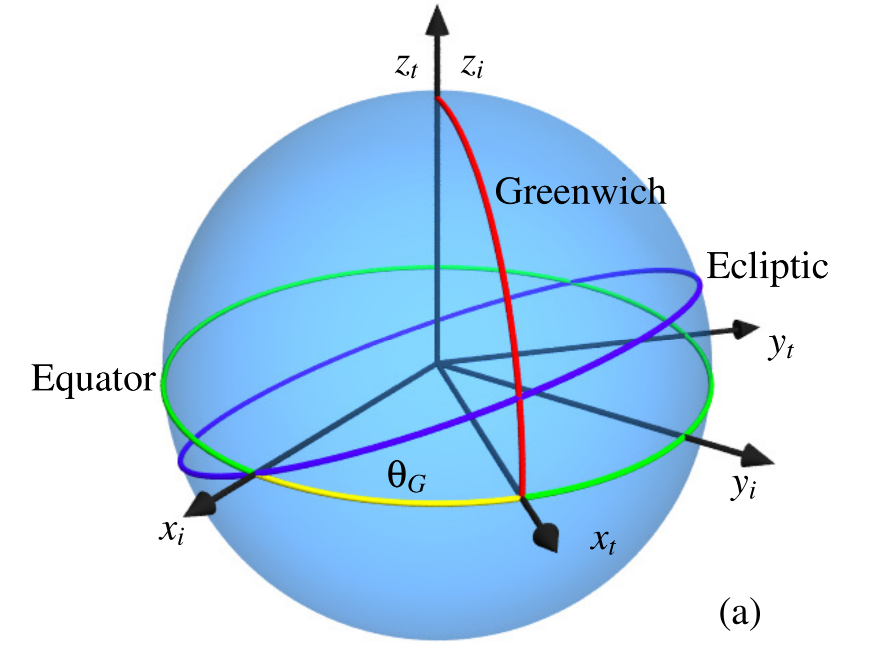
- Satellite Tool Kit (STK interface with Matlab) (commercial)
- Princeton Spacecraft Control Toolbox (commercial)
- Satellite Dynamics Toolbox (non commercial)

PROPAT Toolbox: Coordinate Systems

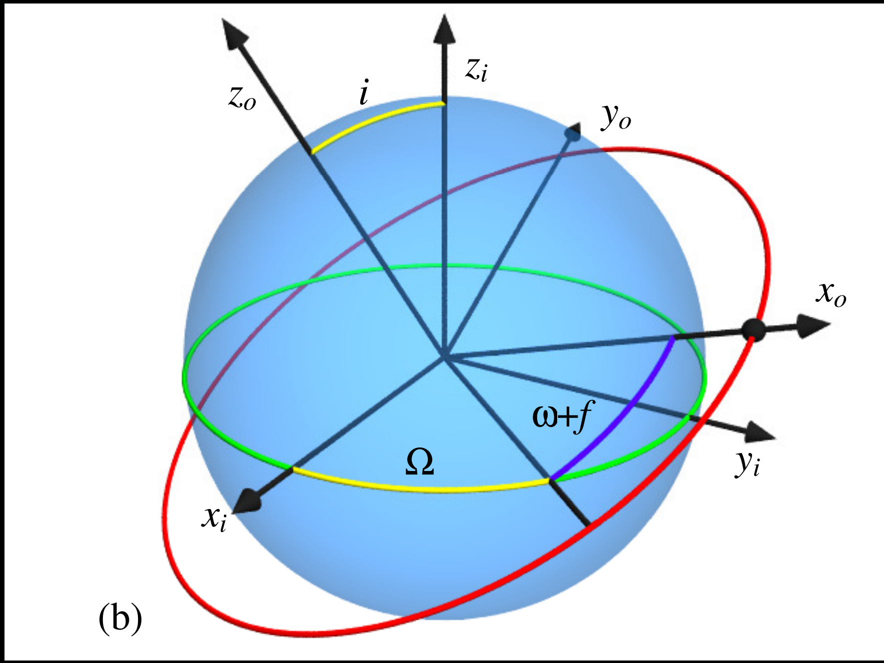
Inertial, Terrestrial, Orbital and NED Systems

$$M = u - e \operatorname{sen} u$$

$$\tan(f/2) = \sqrt{\frac{1+e}{1-e}} \tan(u/2)$$



(a)



(b)

PROPAT Toolbox: Time and Date Conversions

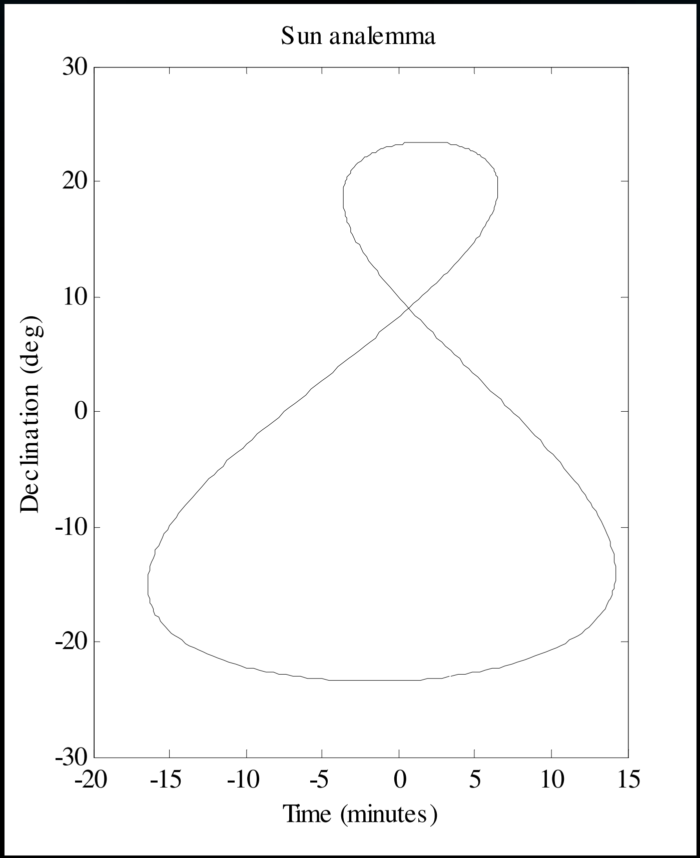
Modified Julian Date, Day Time

```
t = 0;
i = 0;
mjdi = djm(13, 4, 2014);  % day, month and year
while i < 10
   teta = gst(mjdi, t);   % Greenwich sidereal time
   t = t + 100;
   if (t >= 86400)
   t = 0; % new day
   mjdi = mjdi + 1;
   i = 1 + 1;
end
data = djm_inv(mjdi); % corresponding day, month and year of MJD
hms = dayf_to_time(t);% hour, minute and second of the day
end
```

PROPAT Toolbox: Ephemeris and Environment

- Sun (Earth) position
- Earth's magnetic field (IGRF11)
- Greenwich sidereal time
- Sun-Earth eclipse test

```
mjdi = djm(13, 4, 2014);      % day, month and year
dayf = time_to_dayf(8, 35, 44);                % hour
sun_pos = sun(mjdi, dayf)              % Sun position
tsid = gst(mjdi, dayf)    % Greenwich sidereal time
sat_t = [6000000; 3000000; 1000000];
magf = mag_field(mjdi, sat_t)  % geomagnetic field
sat_i = [4000000; 5000000; 2000000];
esha = earth_shadow(sat_i, sun_pos) % Earth shadow
```

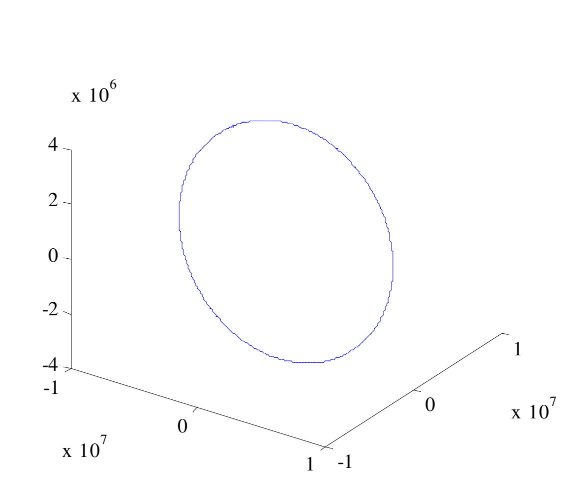


PROPAT Toolbox: Orbit propagation

Analytical Brower's model

Perturbations due to J2 and J4

Keplerian elements: k = [a, e, i, Ω, ω, M]



$$\dot{\mathbf{k}} = \Delta\mathbf{K}(\mathbf{k}_o) \qquad \mathbf{k} = \mathbf{k}_o + \dot{\mathbf{k}}\,(t - t_o)$$

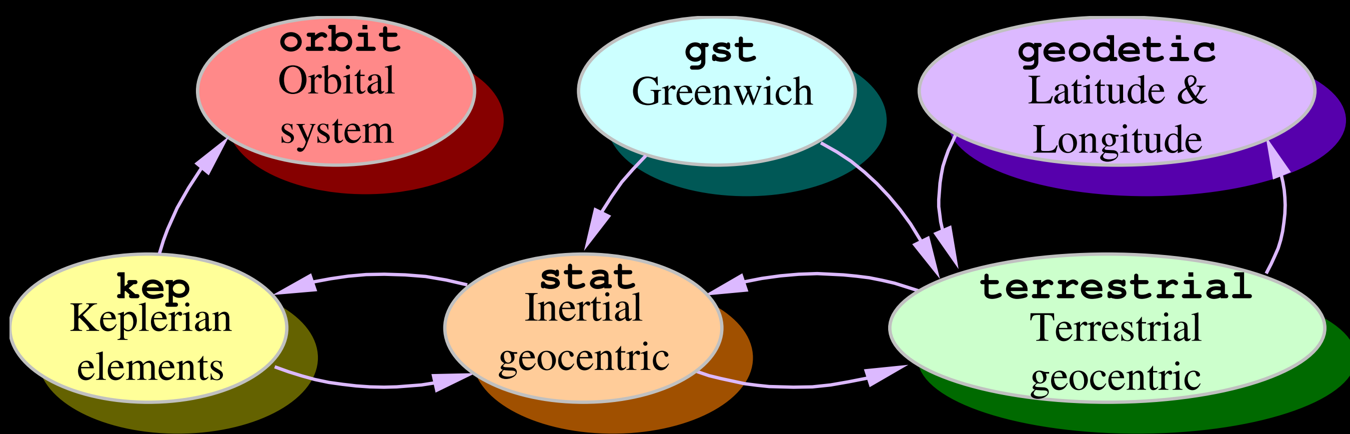
```
kep = [7000000, 0, 0.43, 1, 0, 1.6]; % Kepler el
dkep = delkep(kep); % Kepler elements derivative
stvc = zeros([6,600]);
for t = 0:10:6000
    kept       = kep + dkep*t;    % Orbit propagation
    stvc(:,t/10+1) = kepel_statvec(kept); % State vector
end
plot3(stvc(1,:), stvc(2,:), stvc(3,:))
```

PROPAT Toolbox: Orbit transforms

Analytical Brower's model

Perturbations due to J2 and J4

Keplerian elements: $k = [a, e, i, \Omega, \omega, M]$



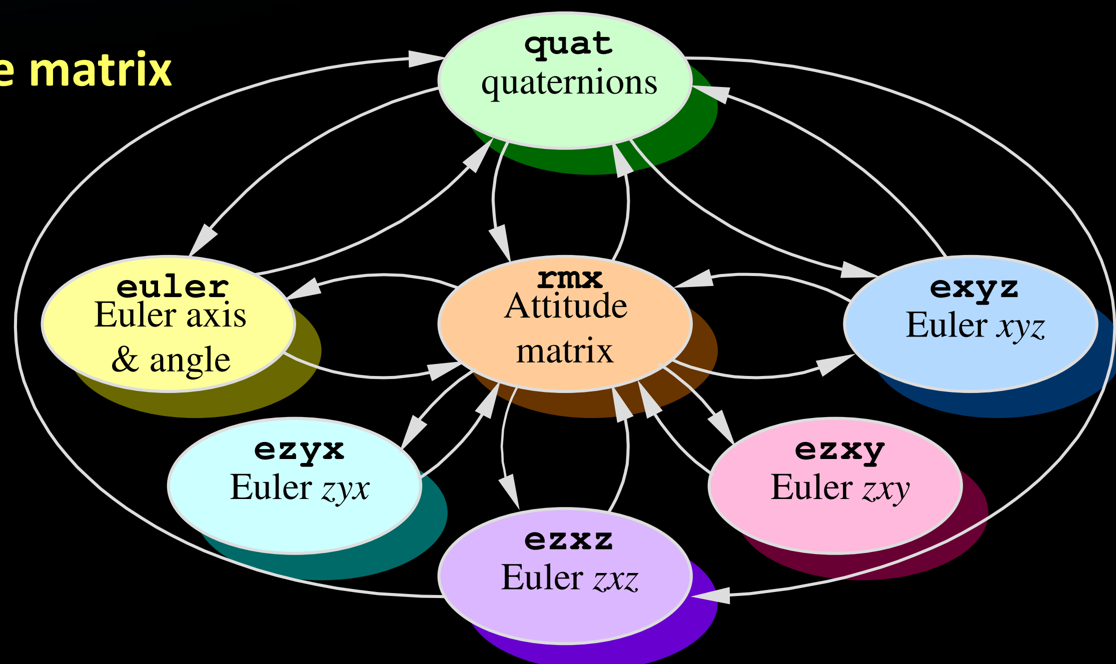
PROPAT Toolbox: Attitude transforms

Euler angles (1-2-3, 3-2-1, 3-1-3 e 3-1-2)

Quaternions

Euler axis and angle

Direction cosine matrix



PROPAT Toolbox: Attitude motion

Kinematics: quaternions

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\ \mathbf{q} = \frac{1}{2}\begin{pmatrix} -\boldsymbol{\omega}^{\times} & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{pmatrix}\mathbf{q}$$

Dynamics: Euler equations of rigid body

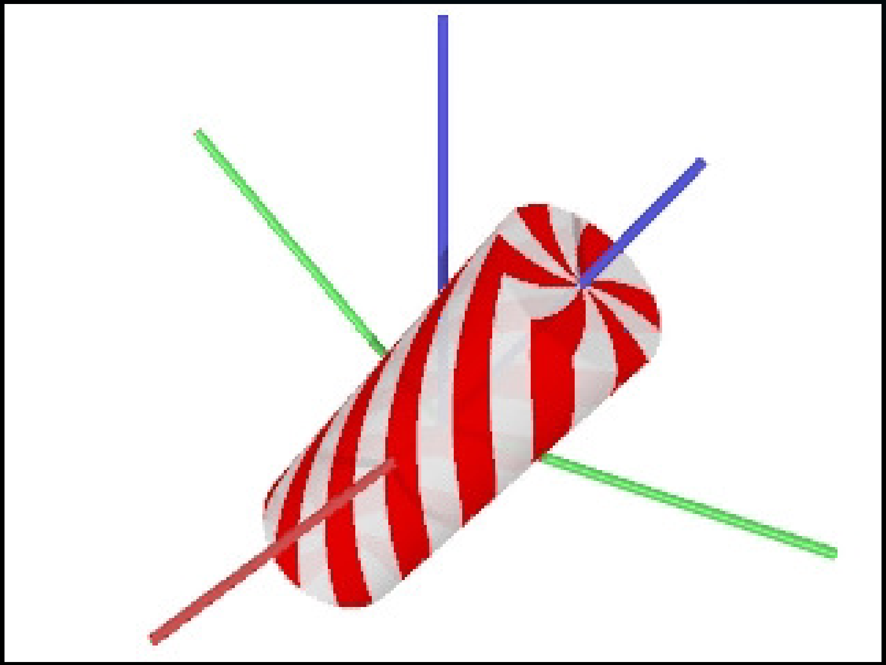
$$\dot{\boldsymbol{\omega}}^{b} = \mathbf{I}^{-1}\left(\mathbf{g}_{cm} - \boldsymbol{\omega}^{\times}\,\mathbf{I}\,\boldsymbol{\omega}\right)$$

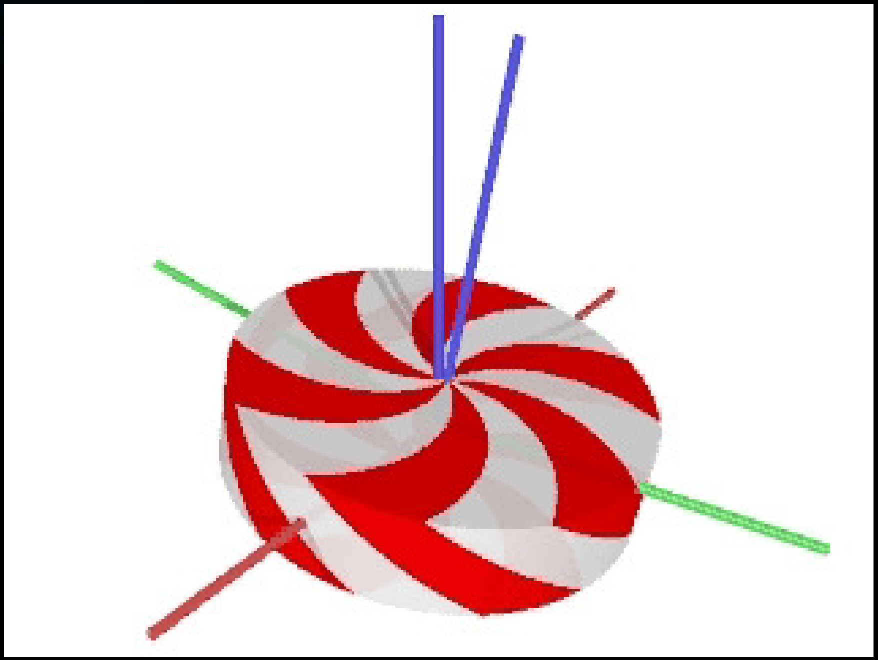
```
% Attitude elements in Euler angles of a 3-1-3 (z-x-z) rotation
eulzxz = [30, 50, 20]'*pi/180;   % converted from degrees to radians
% Attitude in quaternions
quat = ezxzquat(eulzxz);         % converted from Euler angles
% Angular velocity vector in body frame:
w_ang = [0.1, 0, 0.5]';          % in radians/sec
% Inertia matrix of a axis-symmetric rigid body:
iner = [8 0 0; 0 8 0; 0 0 12];   % in kg*m*m
% Inverse inertia matrix:
invin = inv(iner);
% ODE solver precision:
options = odeset('abstol', 1e-4, 'reltol', 1e-4);

for t = 0: 0.5: 600
    % External torques (perturbation + control)
    ext_torq = [0 0 0]';
    % Initial attitude vector:
    att_vec = [quat; w_ang]'; % quaternions and angular velocity
    % ODE Solver parameters
    tspan = [t, t+0.25, t+0.5];
    % Numeric integration (ODE45)
    [T, Y] = ode45('rigbody', tspan, att_vec, options, ext_torq, iner, invin);
    att_vec = Y(3, :)';          % propagated attitude vector
    quat = att_vec(1:4);         % propagated quaternion
    w_ang = att_vec(5:7);        % propagated angular velocity
    eulzxz = quatezxz(quat);     % euler angles
end
```

PROPAT Toolbox: Attitude motion

Videos: POV





PROPAT Toolbox: Attitude motion

Dynamics: Euler equations for non-rigid body:

- **Reaction wheels (3 orthogonal axis)**

$$\mathbf{I}_b \dot{\boldsymbol{\omega}}^b = \mathbf{g}_{cm} - \mathbf{g}_w - \boldsymbol{\omega}^\times \left(\mathbf{I}_b \boldsymbol{\omega} + \mathbf{h}_w \right)$$

$$\dot{\mathbf{h}}_w = \mathbf{g}_w \qquad \mathbf{I}_b \triangleq \left(\mathbf{I} - \sum_{n=1}^{N} I_{n,s} \mathbf{a}_n \mathbf{a}_n^T \right)$$

- **Reaction wheels (*N* wheels)**

$$\mathbf{I}_b \dot{\boldsymbol{\omega}}^b = \mathbf{g}_{cm} - \sum_{n=1}^{N} g_n \mathbf{a}_n - \boldsymbol{\omega}^\times \left(\mathbf{I}_b \boldsymbol{\omega} + \sum_{n=1}^{N} h_{w,n} \mathbf{a}_n \right)$$

$$\dot{h}_{w,n} = g_n$$

PROPAT Toolbox: Attitude motion

Dynamics: Euler equations for non-rigid body:

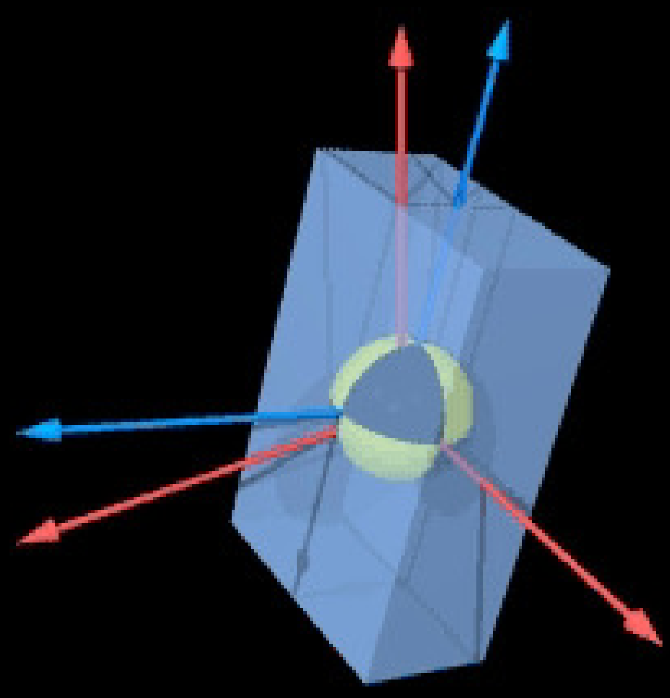
- **Nutation damper**

$$\mathbf{I}_b\,\dot{\boldsymbol{\omega}}^b = \mathbf{g}_{cm} - \sum_{n=1}^{N} g_n\,\mathbf{a}_n - \boldsymbol{\omega}^{\times}\left(\mathbf{I}_b\,\boldsymbol{\omega} + \sum_{n=1}^{N} h_{w,n}\,\mathbf{a}_n\right)$$

$$\dot{h}_{w,n} = -k_n\,\theta_n - b_n\,\omega_n$$

$$\dot{\theta}_n = \omega_n$$

PROPAT Toolbox: Attitude motion



PROPAT Toolbox: Environment disturbances

Gravity gradient torque

$$\mathbf{g}_g = \frac{3\mu}{R^3}\,\mathbf{v}\times(\mathbf{I}\;\mathbf{v})$$

Magnetic torque

$$\mathbf{g}_m = \mathbf{m}\times\mathbf{B}$$

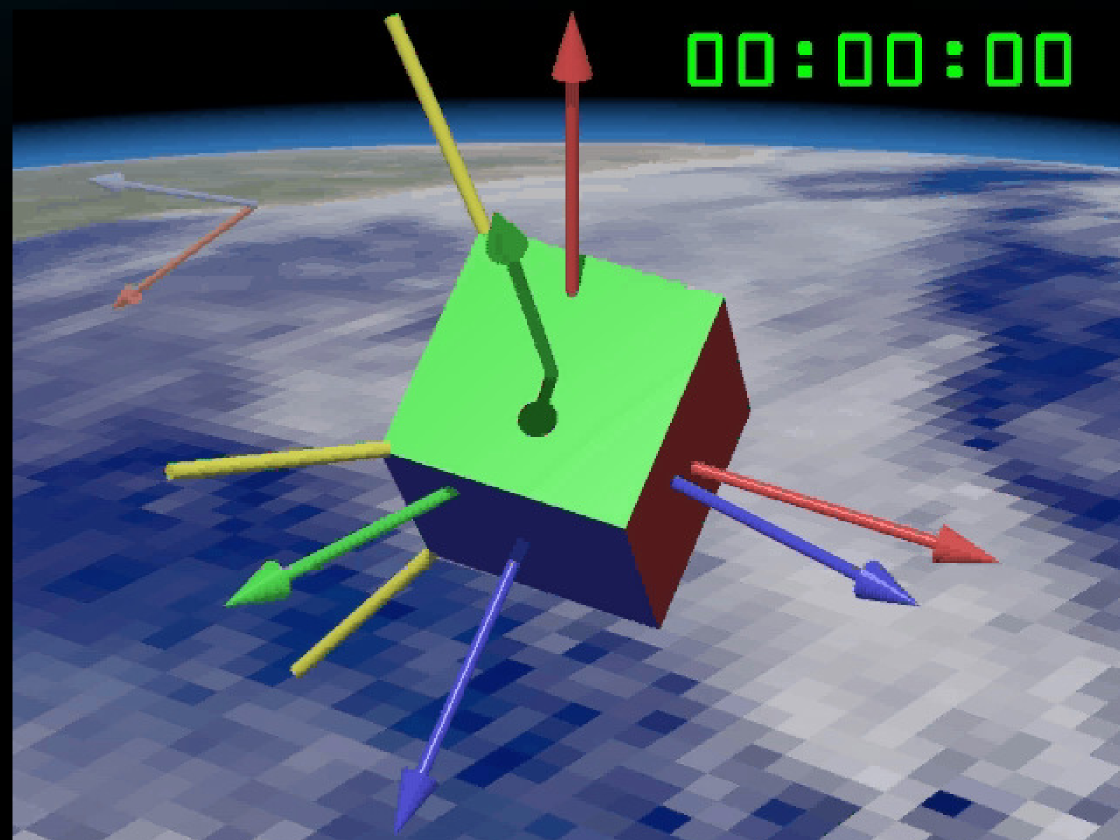
PROPAT Toolbox: Example 1

Magnetic attitude control of a Cubesat (NBR-1)

$$\mathbf{m} = \frac{\mathbf{B} \times \mathbf{u}}{B}$$

$$\mathbf{u} = k_p \, \theta \, \mathbf{a} + k_d \, \boldsymbol{\omega}_{orb}$$

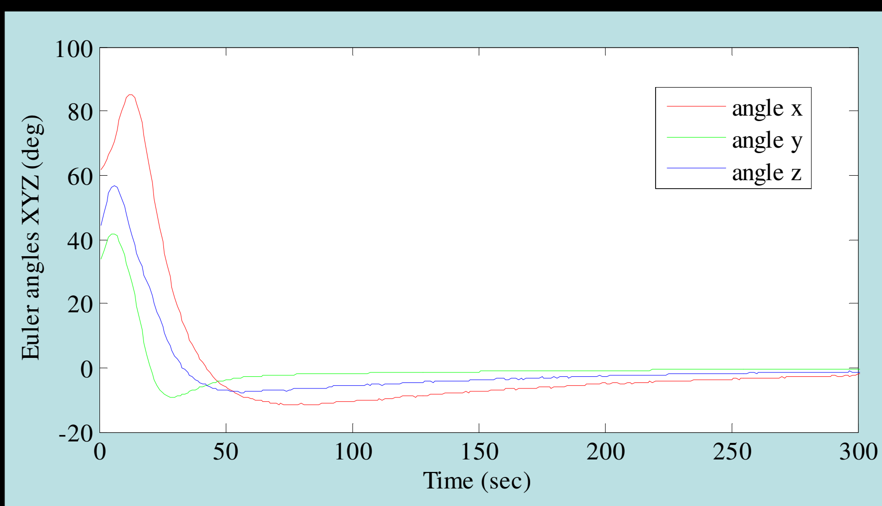
$$\boldsymbol{\omega}_{orb}^{\times} = -\frac{\Delta \mathbf{C}_{b,orb}}{\Delta t} \mathbf{C}_{b,orb}^{T}$$

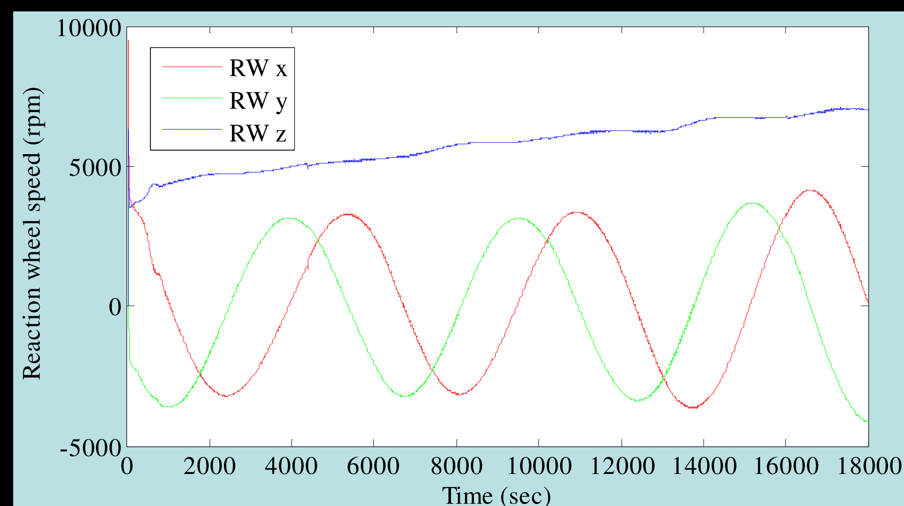


PROPAT Toolbox: Example 2

Attitude control of CONASAT

- Attitude determination (TRIAD)
- Attitude estimation (Kalman)
- Attitude control (magnetic + reaction wheels)

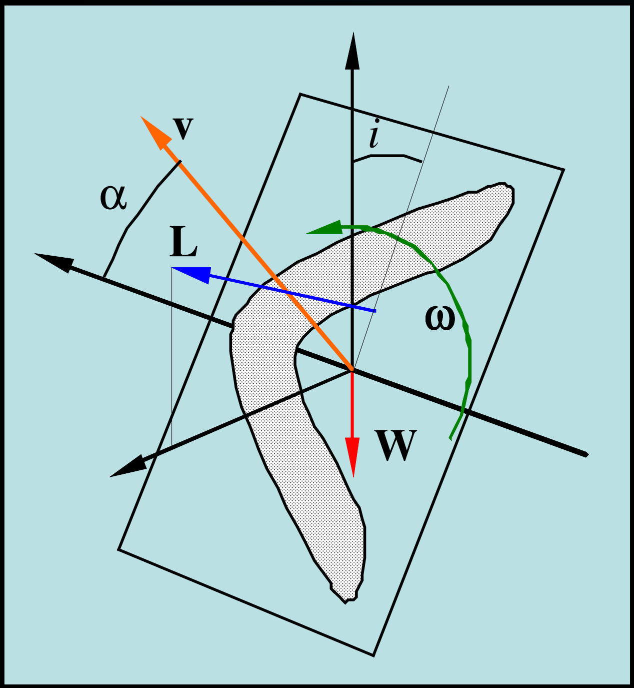


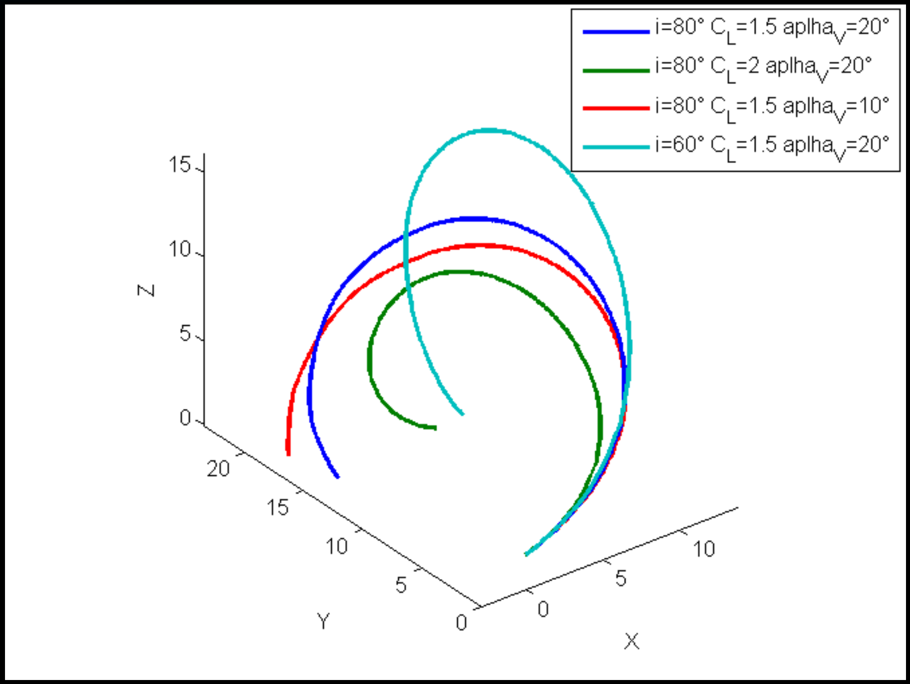


PROPAT Toolbox: Example 3

Trajectory of a boomerang

Drag, lift, weight, throwing attitude and direction





Conclusions

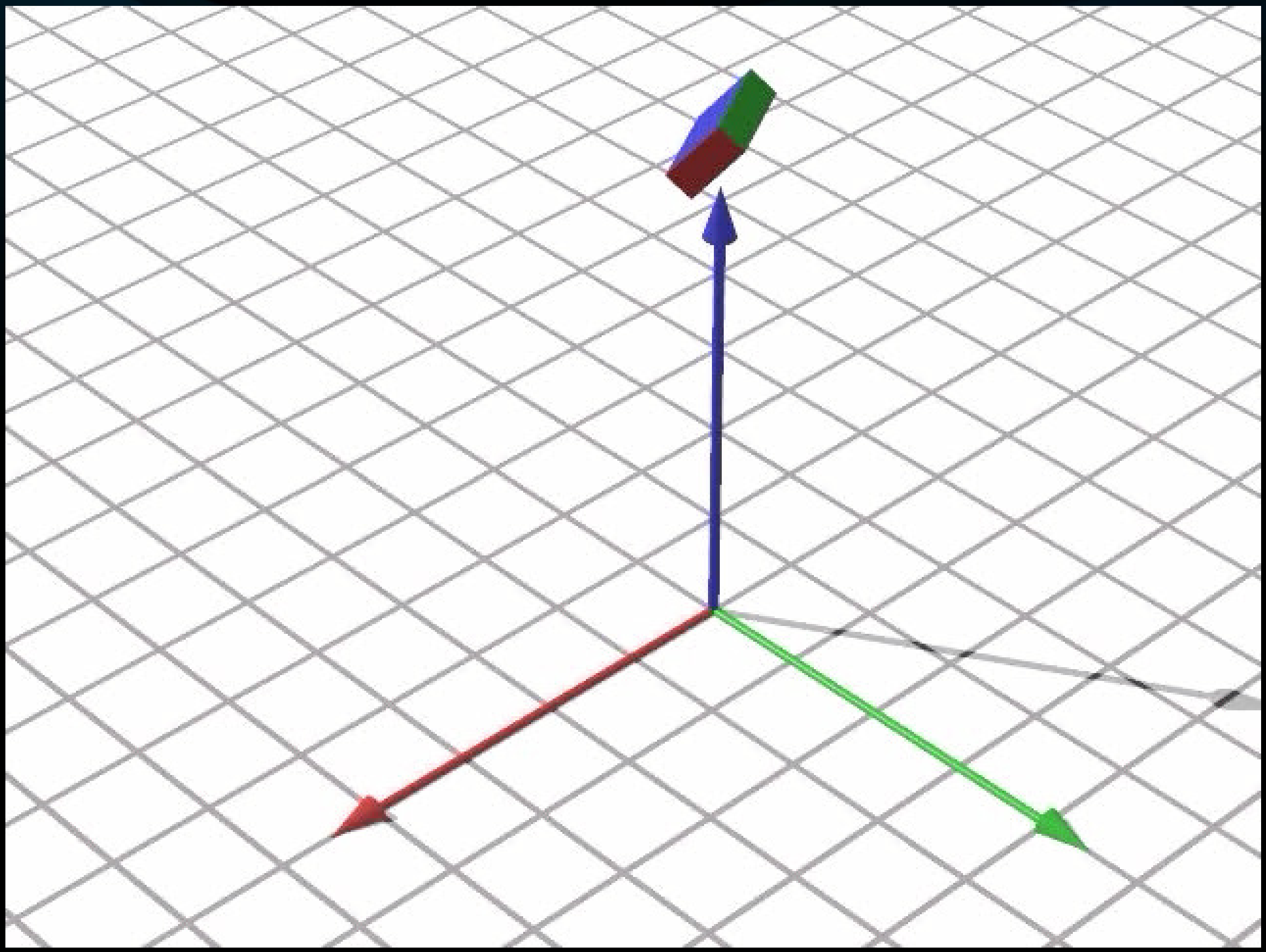
PROPAT is freely available (www.dem.inpe.br/~val)

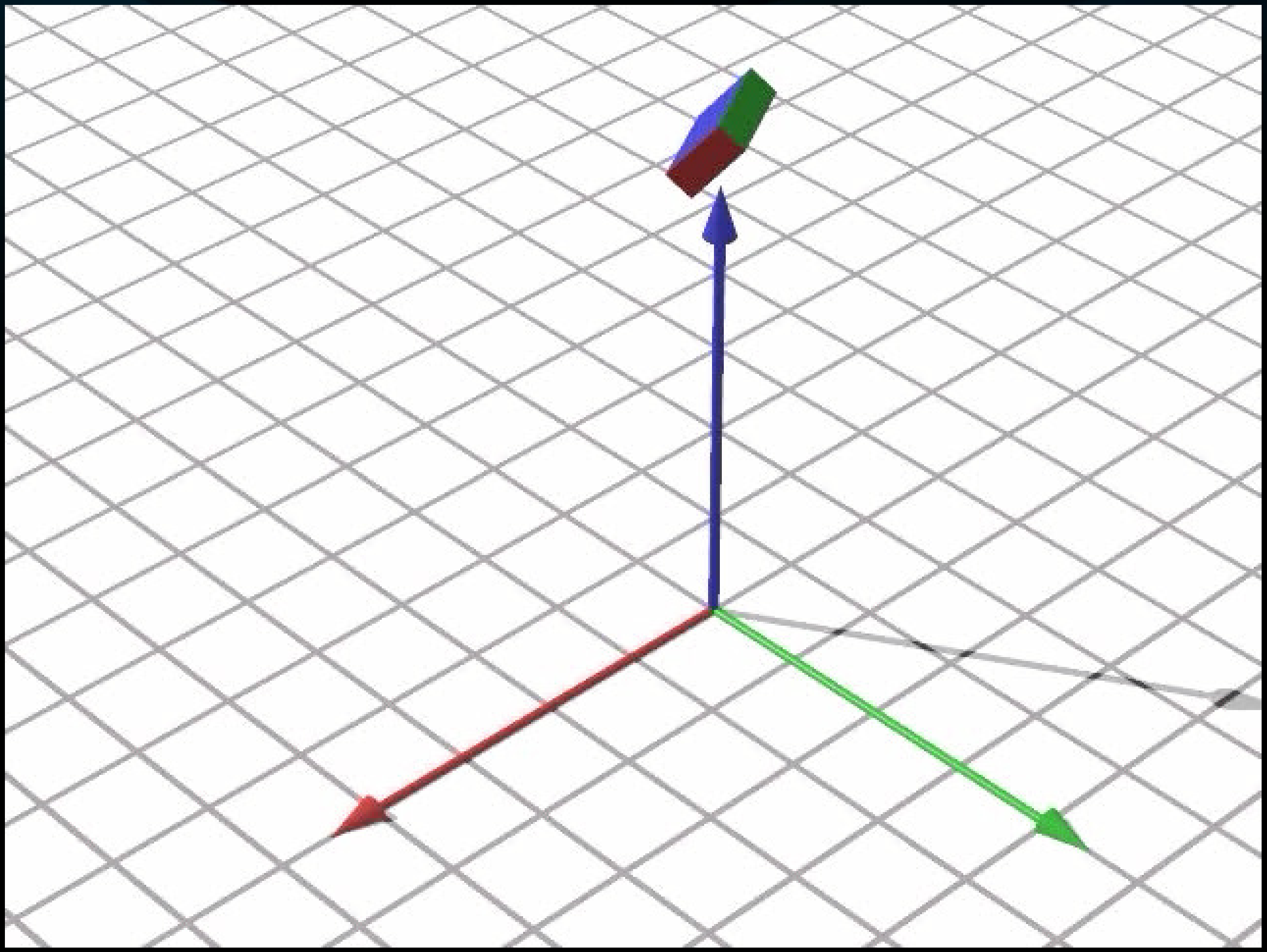
Main applications are:

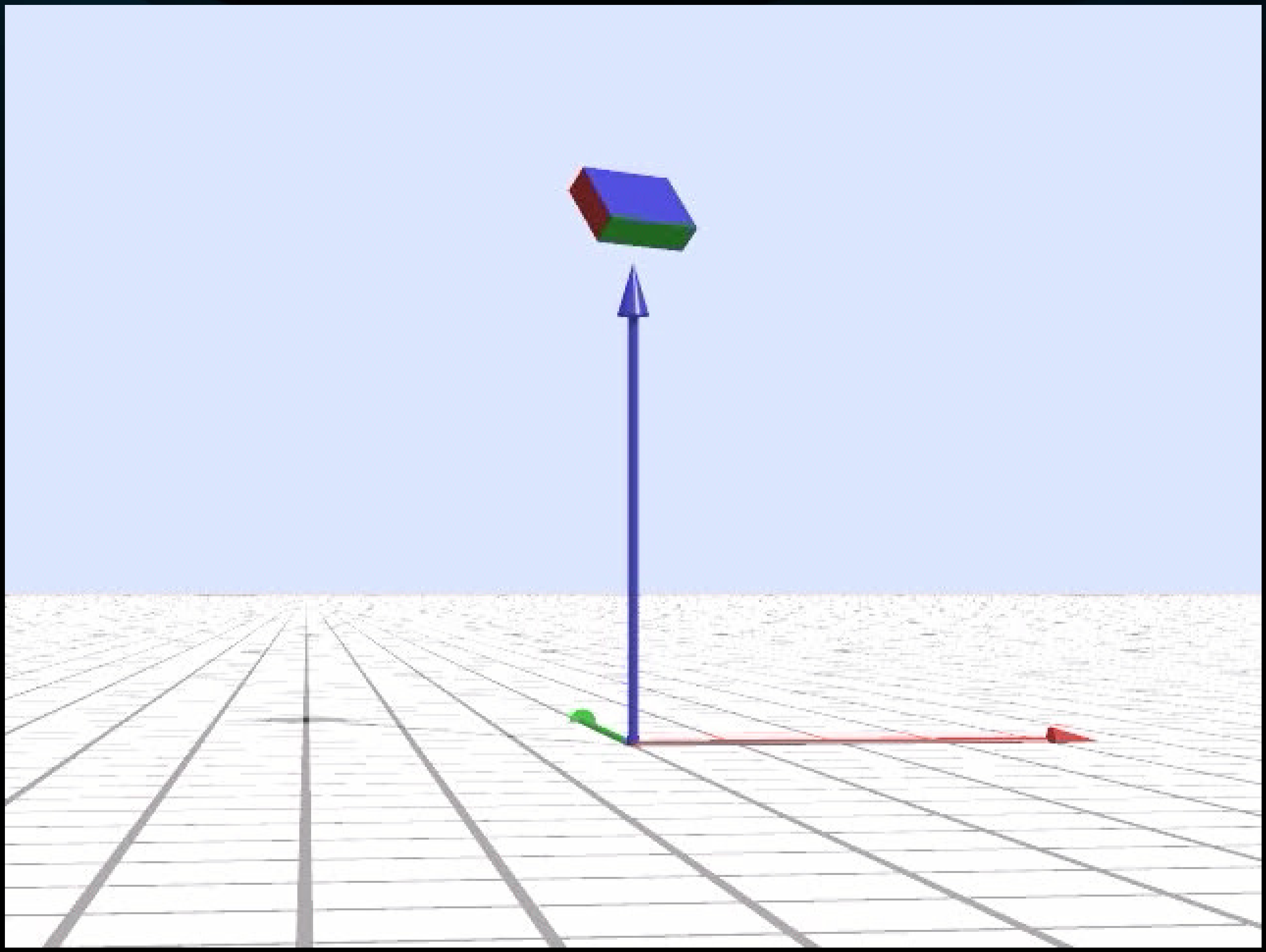
- Attitude simulation
- Attitude control law design
- Attitude determination & control algorithm testing
- Algorithm developing for attitude estimation

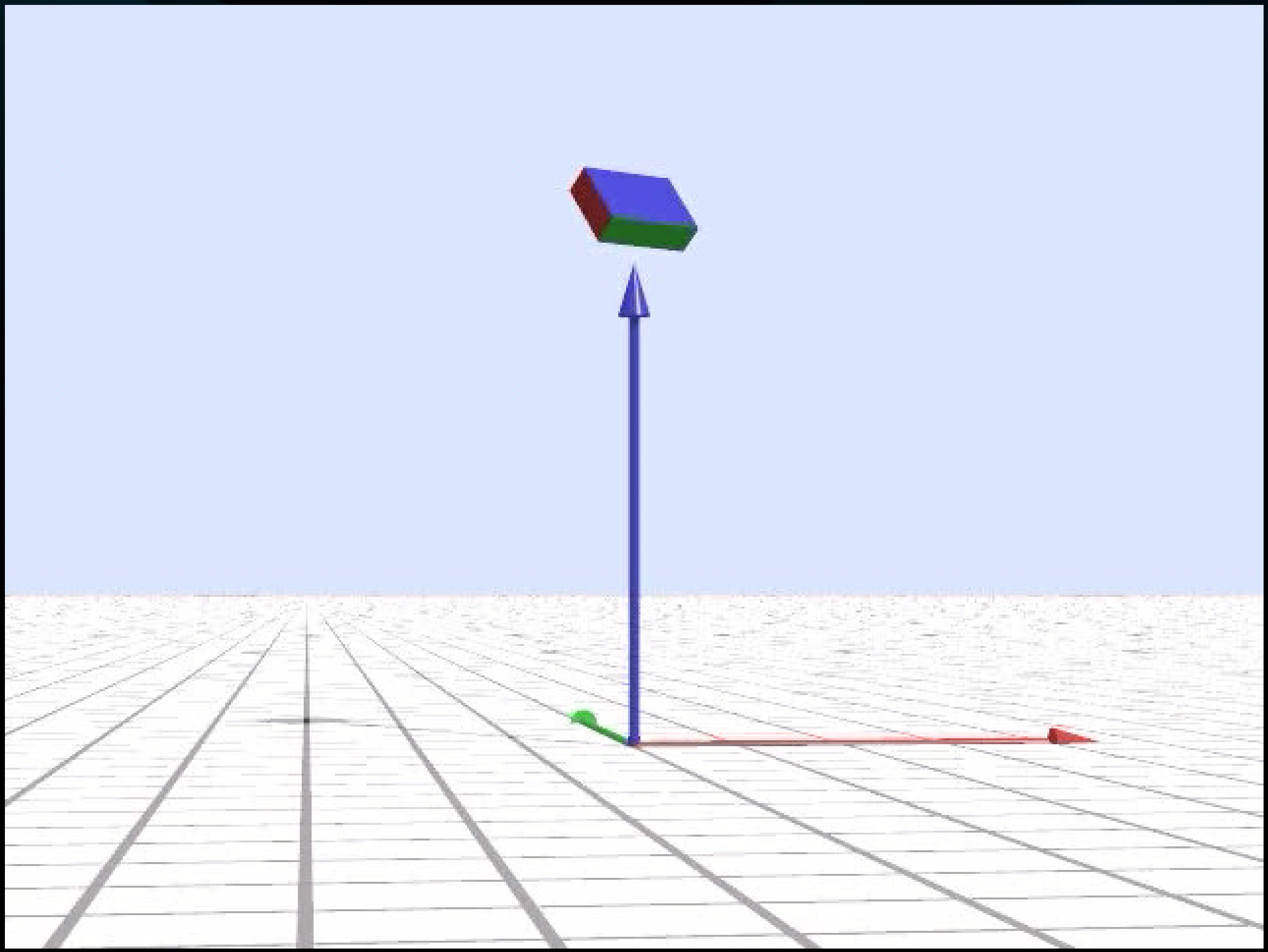
Academic use

Still being developed (new functions added)









Thank you all