Project 2: Armageddon - Atmospheric entry and disruption of asteroids

Synopsis:

Asteroids entering Earth's atmosphere are subject to extreme drag forces that decelerate, heat and disrupt the space rocks. The fate of an asteroid is a complex function of its initial mass, speed, trajectory angle and internal strength.

Asteroids 10-100 m in diameter can penetrate deep into Earth's atmosphere and disrupt catastrophically, generating an atmospheric disturbance (airburst) that can cause damage on the ground. Such an event occurred over the city of Chelyabinsk in Russia, in 2013, releasing energy equivalent to about 520 kilotons of TNT (1 kt TNT is equivalent to 4.184×10^{12} J), and injuring thousands of people (Popova et al., 2013; Brown et al., 2013). An even larger event occurred over Tunguska, an unpopulated area in Siberia, in 1908.

The purpose of this exercise is to develop a fast numerical simulator to predict the fate of asteroids entering Earth's atmosphere for the purposes of hazard assessment.

Problem definition

Equations

The dynamics of an asteroid in Earth's atmosphere is governed by a coupled set of ordinary differential equations:

$$
\begin{aligned}
\frac{dv}{dt} &= \frac{-C_D\rho_a A v^2}{2m} + g\sin\theta \\
\frac{dm}{dt} &= \frac{-C_H\rho_a A v^3}{2Q} \\
\frac{d\theta}{dt} &= \frac{g\cos\theta}{v} - \frac{C_L\rho_a A v}{2m} - \frac{v\cos\theta}{R_P + z} \\
\frac{dz}{dt} &= -v\sin\theta \\
\frac{dx}{dt} &= \frac{v\cos\theta}{1 + z/R_P}
\end{aligned}
$$

In these equations, v, m, and A are the asteroid speed (along trajectory), mass and cross-sectional area, respectively. θ is the meteoroid trajectory angle to the horizontal (in radians), x is the downrange distance of the meteoroid from its entry position, z is the altitude and t is time; C_D is the drag coefficient, ρ_a is the atmospheric density (a function of altitude), C_H is an ablation efficiency coefficient, Q is the specific heat of ablation; C_L is a lift coefficient; and R_P is the planetary radius. All terms use MKS units.

It is common to assume (for simplicity) that, prior to break-up, the radius (and cross-sectional area) of the asteroid remains constant; that is, any mass-loss by **ablation** does not change the cross-sectional area of the asteroid. We will further assume a spherical asteroid.

A commonly used criterion for the break-up of an asteroid in the atmosphere is when the ram pressure of the air interacting with the asteroid $\rho_a v^2$ first exceeds the strength of the asteroid Y.

$$\rho_a v^2 = Y$$

Should break-up occur, the asteroid deforms and spreads laterally as it continues its passage through the atmosphere. As a result its radius and cross-sectional area increase, but the asteroid density is assumed to remain constant (often referred to as 'pancaking'). It is conventional to define the cross-sectional area of the expanding cloud of fragments as $A = \pi r^2$ (i.e., assuming a circular cross-section), for use in the above equations.

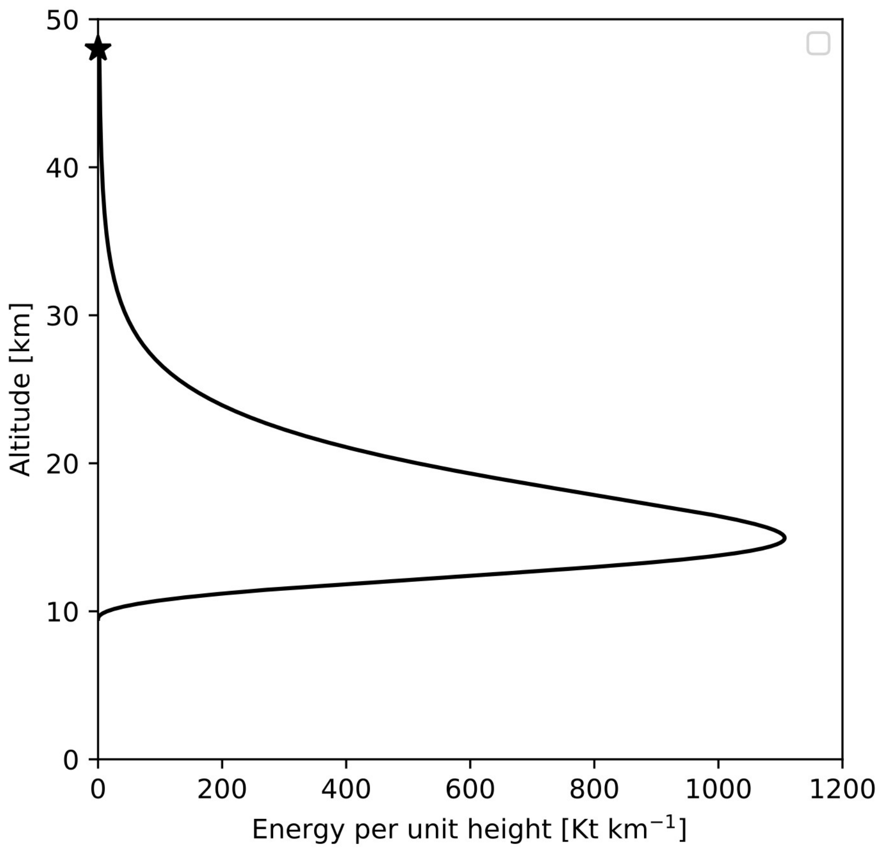
Several models for the spreading rate $\frac{dr}{dt}$ have been proposed. In the simplest model, the fragmented asteroid's spreading rate is related to its along trajectory speed (Hills and Goda, 1993):

$$\frac{dr}{dt} = \left[\frac{7}{2}\alpha\frac{\rho_a}{\rho_m}\right]^{1/2} v$$

Where r is the asteroid radius, ρ_m is the asteroid density and α is a spreading coefficient, often taken to be 0.3.

Note that fragmentation and spreading **ceases** ($\frac{dr}{dt} = 0$) when the ram pressure drops back below the strength of the asteroid $\rho_a v^2 < Y$.

The figure below shows a typical solution to the above set of equations for an impact airburst scenario. The kinetic energy loss per unit height, which is a good approximation for the energy transferred to the atmosphere as heat (and is often referred to as the energy deposited per km altitude), is shown as a function of altitude.



In this scenario the asteroid experiences breakup at approximately 48-km altitude (denoted by the star), before spreading and decelerating rapidly until an altitude of approximately 15 km at which point the **energy loss per unit height** is **maximum**. This point is often considered to be the **burst altitude**. The total kinetic energy lost by the asteroid at this point is a good estimate of the airburst energy (i.e., the total energy deposited into the atmosphere) for hazard analysis.

Impact scenarios with a burst altitude well (>5 km) above the surface can be considered to be **airburst events** and are unlikely to form a sizable crater on the ground. In such cases, even if a substantial fraction of the original asteroid survives to the ground it will be decelerated to a very small fraction of its initial speed.

Impact scenarios with a burst altitude below the surface (i.e., peak energy deposition per unit height is not reached before the asteroid strikes the ground) will form a sizable impact crater on the ground and can be considered as a **cratering event** (although a sizable proportion of the original kinetic energy of the asteroid may be transferred to the air).

Between these scenarios is a complex regime where a **low altitude airburst combined with a substantial crater-forming event** is likely. This regime is not well understood.

Function

Python asteroid airburst calculator

class `armageddon.Planet`(*atmos_func='exponential'*, *atmos_filename=None*, *Cd=1.0*, *Ch=0.1*, *Q=10000000.0*, *Cl=0.001*, *alpha=0.3*, *Rp=6371000.0*, *g=9.81*, *H=8000.0*, *rho0=1.2*)

The class called Planet is initialised with constants appropriate for the given target planet, including the atmospheric density profile and other constants

Set up the initial parameters and constants for the target planet :param atmos_func: Function which computes atmospheric density, rho, at altitude, z.

Default is the exponential function `rho = rho0 exp(-z/H)`. Options are `exponential`, `tabular`, `constant` and `mars`

Parameters:

- **atmos_filename** (*string, optional*) – If `atmos_func = 'tabular'`, then set the filename of the table to be read in here.
- **Cd** (*float, optional*) – The drag coefficient
- **Ch** (*float, optional*) – The heat transfer coefficient
- **Q** (*float, optional*) – The heat of ablation (J/kg)
- **Cl** (*float, optional*) – Lift coefficient
- **alpha** (*float, optional*) – Dispersion coefficient
- **Rp** (*float, optional*) – Planet radius (m)
- **rho0** (*float, optional*) – Air density at zero altitude (kg/m^3)
- **g** (*float, optional*) – Surface gravity (m/s^2)
- **H** (*float, optional*) – Atmospheric scale height (m)

Returns:

Return type: None

`analyse_outcome`(*result*)

Inspect a prefound solution to calculate the impact and airburst stats :param result: pandas Data-Frame with velocity, mass, angle, altitude, horizontal

distance, radius and dedz as a function of time

Returns: **outcome** – dictionary with details of airburst and/or cratering event. For an airburst, this will contain the following keys: `burst_peak_dedz`, `burst_altitude`, `burst_total_ke_lost`. For a cratering event, this will contain the following keys: `impact_time`, `impact_mass`, `impact_speed`. All events should also contain an entry with the key `outcome`, which should contain one of the following strings: `Airburst`, `Cratering` or `Airburst and cratering`

Return type: Dict

`calculate_energy`(*result*)

Function to calculate the kinetic energy lost per unit altitude in kilotons TNT per km, for a given solution. :param result: A pandas DataFrame with columns for the velocity, mass, angle,

altitude, horizontal distance and radius as a function of time

Returns: **Result** – Returns the DataFrame with additional column `dedz` which is the kinetic energy lost per unit altitude

Return type: DataFrame

`dedz`(*radius*, *velocity*, *density*, *strength*, *angle*, *z*, *degree=True*)

Calculating velocity from altitude.

———-input———– v0, initial speed, m/s A, cross-sectional area, m**2 m, mass in kg angle, angle of injection, inradians z, altitude in metres, usually an array Cd, drag coefficient, preset as 1 rho0, density of air, preset as 1.2kg/m**3 H, height of atmosphere, preset as 8000m radian, boolean datum that tells whether the angle is given in radians; preset as False ———-output———– change in energy per unit length in kT/km, usually an array

`dvdz`(*radius*, *velocity*, *density*, *strength*, *angle*, *z*, *degree=True*)

Calculating velocity from altitude.

———-input———– v0, initial speed, m/s A, cross-sectional area, m**2 m, mass in kg angle, angle of injection, inradians z, altitude in metres, usually an array Cd, drag coefficient, preset as 1 rho0, density of air, preset as 1.2kg/m**3 H, height of atmosphere, preset as 8000m radian, boolean datum that tells whether the angle is given in radians; preset as False ———-output———– change in speed per unit altitude in /s, usually an array

`fun`(*t*, *state*, *strength*)

RHS function for impact system

`imp_eu`(*fun*, *radius*, *velocity*, *density*, *angle*, *strength*, *init_altitude*, *dt*, *t_max*, *t0*, *x*)

the improved Euler function from the lecture we have modified u to include all 5 of our variables

`impact`(*radius*, *velocity*, *density*, *strength*, *angle*, *init_altitude=100000.0*, *dt=0.05*, *radians=False*)

Solve the system of differential equations for a given impact event. Also calculates the kinetic energy lost per unit altitude and analyses the result to determine the outcome of the impact. :param radius: The radius of the asteroid in meters :type radius: float :param velocity: The entery speed of the asteroid in meters/second :type velocity: float :param density: The density of the asteroid in kg/m^3 :type density: float :param strength: The strength of the asteroid (i.e., the ram pressure above which

fragmentation and spreading occurs) in N/m^2 (Pa)

Parameters:

- **angle** (*float*) – The initial trajectory angle of the asteroid to the horizontal By default, input is in degrees. If 'radians' is set to True, the input should be in radians
- **init_altitude** (*float, optional*) – Initial altitude in m
- **dt** (*float, optional*) – The output timestep, in s
- **radians** (*logical, optional*) – Whether angles should be given in degrees or radians. Default=False Angles returned in the DataFrame will have the same units as the input

Returns:

- **Result** (*DataFrame*) – A pandas DataFrame containing the solution to the system. Includes the following columns: `velocity`, `mass`, `angle`, `altitude`, `distance`, `radius`, `time`, `dedz`
- **outcome** (*Dict*) – dictionary with details of airburst and/or cratering event. For an airburst, this will contain the following keys: `burst_peak_dedz`, `burst_altitude`, `burst_total_ke_lost`. For a cratering event, this will contain the following keys: `impact_time`, `impact_mass`, `impact_speed`. All events should also contain an entry with the key `outcome`, which should contain one of the following strings: `Airburst`, `Cratering` or `Airburst and cratering`

`r2A`(*radius*)

calculate area from given radius

`rrho2m`(*radius*, *density*)

calculate mass from given radius & density

`solve_atmospheric_entry`(*radius*, *velocity*, *density*, *strength*, *angle*, *init_altitude=100000.0*, *dt=0.05*, *radians=False*)

Solve the system of differential equations for a given impact scenario :param radius: The radius of the asteroid in meters :type radius: float :param velocity: The entery speed of the asteroid in meters/second :type velocity: float :param density: The density of the asteroid in kg/m^3 :type density: float :param strength: The strength of the asteroid (i.e., the ram pressure above which

fragmentation and spreading occurs) in N/m^2 (Pa)

Parameters:

- **angle** (*float*) – The initial trajectory angle of the asteroid to the horizontal By default, input is in degrees. If 'radians' is set to True, the input should be in radians
- **init_altitude** (*float, optional*) – Initial altitude in m
- **dt** (*float, optional*) – The output timestep, in s
- **radians** (*logical, optional*) – Whether angles should be given in degrees or radians. Default=False Angles returned in the DataFrame will have the same units as the input

Returns: **Result** – A pandas DataFrame containing the solution to the system. Includes the following columns: `velocity`, `mass`, `angle`, `altitude`, `distance`, `radius`, `time`

Return type: DataFrame

`vz`(*radius*, *velocity*, *density*, *strength*, *angle*, *z*, *degree=True*)

Calculating velocity from altitude.

———-input——— v0, initial speed, m/s A, cross-sectional area, m**2 m, mass in kg angle, angle of injection, inradians z, altitude in metres, usually an array Cd, drag coefficient, preset as 1 rho0, density of air, preset as 1.2kg/m**3 H, height of atmosphere, preset as 8000m radian, boolean datum that tells whether the angle is given in radians; preset as False ———-output——— speed in m/s, usually an array

`armageddon.`**`solve_ensemble`**(*planet*, *fiducial_impact*, *variables*, *radians=False*, *rmin=8*, *rmax=12*)

Run asteroid simulation for a distribution of initial conditions and find the burst distribution

Parameters:

- **planet** (*object*) – The Planet class instance on which to perform the ensemble calculation
- **fiducial_impact** (*dict*) – Dictionary of the fiducial values of radius, angle, strength, velocity and density
- **variables** (*list*) – List of strings of all impact parameters to be varied in the ensemble calculation

- **rmin** (*float, optional*) – Minimum radius, in m, to use in the ensemble calculation, if radius is one of the parameters to be varied.
- **rmax** (*float, optional*) – Maximum radius, in m, to use in the ensemble calculation, if radius is one of the parameters to be varied.