Modeling Meteors

Summer School of Science

Project report

Chloe Udressy
Marcell Dorian Kovacs
Nensi Komljenovic

Project leader: Dusan Pavlovic

2017. 07. 29 - 08. 07.

Contents

1 Abstract

By developing a Single Body Theory and using Euler's method, about 200 000 meteors with assumptions about their initial conditions (velocity, mass, zenith angle, density) were analysed. We simulated light curves and determined their intensity in terms of meteoroids' initial conditions. We also calculated the height of maximum intensity in order to apply these functions to real data from observations and, by getting information about their density, gaining insight into their origin.

2 Introduction

2.1 The concept of a meteor

Meteor showers are an atmospheric phenomena observable on the night sky, as a result of their light traces. Meteoroids are particles of matter which, while traveling through interplanetary space, enter Earth's atmosphere due to the intersection of Earth's and their orbit. That causes them to penetrate Earth's atmosphere and because of their collision with atmospheric particles, a process called ablation takes place and causes the radiation of light. Performance of the light curve can tell us a lot about meteor's properties. Furthermore, if the mass ablating through atmosphere doesn't burn down before touching the Earth's surface, we call the body a meteorite.

2.2 Structure of meteoroids

Meteoroids can originate from two different spacial bodies. Asteroids are one option for parental body of meteor and they give meteors denser, heterogeneous composition because then their structure contains silicates and metals such as iron or nickel. On the other hand, meteoroids can also originate from comets which cause them having homogeneous, low density structure made from dust and ice. Their structure greatly influences their emitted light, therefore from light curve's properties and knowledge about structure and density we can trace back parental bodies.

3 Model

We developed a phenomenological model that describes a single body's interaction with the atmosphere mainly using Newtonian concepts. Our theory connects the intensity of light emitted from the body with physical parameters of the meteoroid. Later, we used this model to simulate meteor light curves. We used several simplifying assumptions about both the body and the atmosphere.

3.1 Single Body Theory

For the theory, we assumed that our body is:

- rigid (it has constant density ρ)
- perfectly spherical (shape parameter is constant)
- interacting with the atmosphere which can be viewed as a collision

Furthermore, we neglected microscopical (except for the τ parameter's calculation) and fluid dynamical phenomena occurring in the interaction.
We had to describe three variables: the velocity ($\mathbf{v}$) and how it changes over time (as the meteoroid decelerates in the air), the mass ($\mathbf{m}$) loss of the body (as a result of the process called ablation, see [1]) and the height ($\mathbf{h}$) change, to complete the equation system (in this paper, we will use bold for these variables, not because they are vectors but to emphasize that these are the meteor's most important variables).

3.1.1 Determining velocity

In the interaction of the air and the body, the meteoroid transfers momentum to the air (because of the conservation of momentum), thus it decelerates. The amount of momentum transferred is given by:

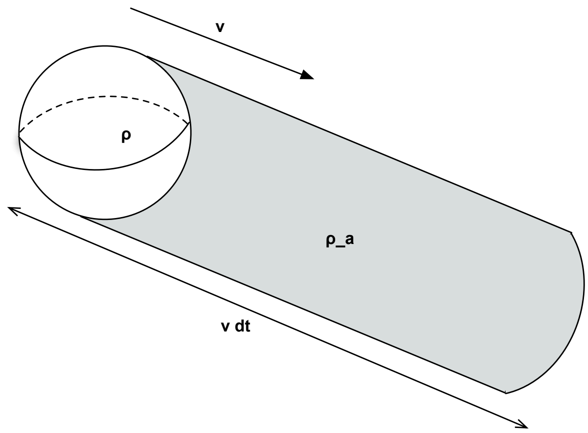

Figure 1: Our model of interaction

$$dp = -\Gamma \mathbf{v} dm_a,$$

where Γ is the drag coefficient of a sphere, dm_a is the mass of the cylinder in Figure 1 at a time instant dt. dm_a can be calculated from geometric properties of the sphere:

$$dm_a = \rho_a A \left(\frac{\mathbf{m}}{\rho}\right)^{\frac{2}{3}} \mathbf{v} dt,$$

where A is the shape parameter of the sphere and ρ_a is the atmospheric density. Substituting $dp = mdv + dmv$ while neglecting dm yields the final equation for deceleration (also known as Whipple's equation):

$$\dot{\mathbf{v}} = -\Gamma A \mathbf{m}^{-\frac{1}{3}} \rho^{-\frac{2}{3}} \rho_a \mathbf{v}^2$$

3.1.2 Determining mass loss

The meteoroid is also gaining energy because of the collision and the conservation of energy. This causes it to raise its temperature which results in the emission of light and the decrease of mass. The following equation describes this energy transfer:

$$d\mathbf{m} \cdot Z = -\Lambda \cdot dE_{kin},$$

where Z represents the specific heat (the amount of latent heat the body gains) and Λ represents the heat transfer coefficient, the amount of kinetic energy radiated away. The value of these parameters are determined from experiment. E_{kin} (kinetic energy) can be calculated from:

$$dE_{kin} = \frac{1}{2} d\mathbf{m} v^2,$$

where $d\mathbf{m} = dm_a$. Substituting yields the equation of ablation:

$$\dot{\mathbf{m}} = -\frac{\Lambda A}{2Z} \mathbf{m}^{\frac{2}{3}} \rho^{-\frac{2}{3}} \rho_a \mathbf{v}^3$$

3.1.3 Determining height change

The change in height ($\dot{h}$) can be calculated as the vertical projection of velocity:

$$\dot{\mathbf{h}} = -\mathbf{v} \cos(z)$$

In the equation above, z represents the angle of penetration (also called zenith angle) which is the angle between the vertical axis and the initial velocity vector.

3.2 Simple Atmospheric Model

To complete Single Body Theory, we needed to find how atmospheric density changes over height. For this, we derived the simple atmospheric model (also known as the isothermal atmospheric model).

For this, we postulated that the atmosphere :

- is made out of an ideal gas with constant molar mass
- is isothermal
- has uniform gravitational attraction
- is behaving like a plan parallel area of air.

To arrive at the density function, we had to determine the pressure function, as these properties are related by the ideal gas law. To find the pressure, we used that an infinitely thin layer of air is in equilibrium, so the pressure coming from the bottom cancels out the pressure of the top layer. As a result, we obtained the following equation of density (with the usual constants, ρ_0 representing the surface density, and M representing air's molar mass):

$$\rho_a = \rho_0 e^{-\frac{Mg}{RT}\mathbf{h}}$$

The proof is left to the reader as an exercise.

3.3 Light curve modeling

Since the light intensity is the emitted energy through visible light, we calculated it from the change in kinetic energy ($\frac{1}{2}d\mathbf{m} \cdot \mathbf{v}^2$):

$$I = \tau \cdot \frac{dE_{kin}}{dt} = \tau \frac{1}{2}\mathbf{v}^2 \frac{d\mathbf{m}}{dt}$$

The τ parameter's value is determined from microscopical analysis as well as numerical methods, but discussion of it goes beyond this paper's scope. More information can be found in the attached C++ code. The final equation for light intensity:

$$\mathbf{I} = -\frac{1}{2}\tau \dot{\mathbf{m}}\mathbf{v}^2$$

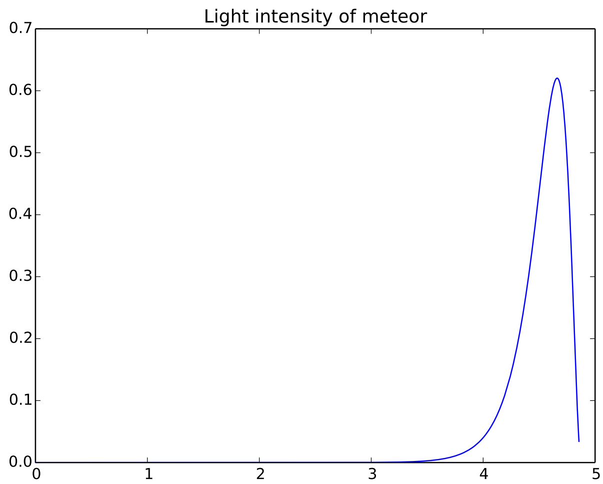

Figure 2: An example of a light curve

4 Simulation

Analyzing single light curves are useful to study the phenomenon of a meteor, since they describe the intensity of the light resulting from the ablation of the mass. However, simulating numerous light curves from a wide range of different types of meteoroids could give us precious information beyond the phenomenon itself. It indeed allowed us to have an insight into the structure of observed meteoroids for which we have some data, by calculating their density as a function of measurable parameters (height, velocity, zenith angle and apparent magnitude of the resulting meteor).

4.1 Distinct meteors

For making the density calculation easier, we focused on the point of highest intensity (I_{max}) of the meteor. We named the height where I_{max} occurs H_{max}. To get functions for I_{max} and H_{max} which were only dependent on the measurable variables and on the density of the meteoroid, we first had to simulate the intensity and height functions of meteoroids differing in their initial velocity, initial mass, zenith angle and density.

We generated about 200,000 meteoroids which could tangibly intercept the Earth's atmosphere, by setting a realistic range for each individual variable. Thus, the mass has been constrained between 10^{-8} and $10^{-1.5}$ kg, the zenith angle from $0°$ to $80°$, the density of the meteoroid from 300 to $3,100\frac{kg}{m^3}$ and the velocity from $11,200$ to $72,800\frac{m}{s}$. These last two values are derived from cosmic speeds. The minimum velocity is the second cosmic[1] speed with respect to Earth, while the second one is the second cosmic speed with respect to the Sun added to Earth's velocity around the Sun.

4.2 Our simulation program

To calculate I_{max} and H_{max} of the resulting meteors, we first had to solve the equations of the Single Body Theory for each of their meteoroid. Since they're ordinary differential equations, we used Euler numerical approximation method. The principle of this iterative process is based on the concept of the derivative. Knowing initial values, any point of a function can be calculated as a function of the previous term (f represents the functional relation of y and $\dot{y}$):

$$y_{n+1} = y_n + dtf(y_n)$$

The obtained recurrence relations have then been written in a C++ program. They have been simultaneously solved inside four nested for-loops, which generated the distinct meteors. I_{max} and H_{max} has thereafter been calculated for each meteoroid. For more information, see the attached code.

4.3 Intensity and height functions

With the resulting data, we then were able to calculate the coefficients and exponents of the I_{max} and H_{max} functions by fitting them to the obtained values (multiple regression analysis).

$$I_{max} = Km^{k_1}\rho^{k_2}v^{k_3}\cos^{k_4}(z)$$

$$H_{max} = Lm^{l_1}\rho^{l_2}v^{l_3}\cos^{l_4}(z)$$

We first estimated I_{max} since the H_{max} function depends on the apparent magnitude M_{app} which is proportional to the intensity of the light in the following way:

$$M_{app} = -14.18 - 2.5\log\frac{683}{4\pi 10^{10}}\mathbf{I}$$

After fitting H_{max} function, we just had to rearrange it, to isolate the meteoroid's density as a function of measurable parameters.

$$\rho = 10^{\frac{1}{l_3}[H_{max}-\log(L)-l_1\log M_{app}-l_2\log(v)-l_4\log(\cos(z))]}$$

[1]Second cosmic speed is the speed which a body needs to have in order to escape the solar system from a given object.

5 Application

5.1 Obtaining observational data

In order to check our model's validity, we obtained observational data about atmospheric and cometary meteor showers from [2]. This paper reports on 4 kinds of shower meteors: Leonids, Orionids, Perseids and Geminids. These types got their name after the position with respect to the night sky's stars. The measured properties we used were the height of reaching maximum light intensity (H_{max}), apparent brightness (M_{app}, which can be expressed in terms of light intensity), initial velocity (v_{init}) and zenith angle of penetration (z). The following table shows the measurements:

Shower Meteors	$H_{max}[km]$	$M_{app}[mags]$	$z(°)$	$v_{init}[kms^{-1}]$
Leonids	106.9 ± 3.8	−0.3 ± 3.4	44.2 ± 1.7	70.7
Orionids	106.7 ± 2.1	−1.0± 3.8	42.7 ± 1.0	66.4
Perseids	104.4 ± 2.9	−2.1± 4.2	42.7 ± 0.9	59.6
Geminids	91.7 ± 3.4	−0.3± 4.6	32.2 ± 2.0	34.4

5.2 Density calculations

The formula for meteor density in terms of observable parameters is determined from the simulation data using a multiple regression fit (the fit yielded the L, l_i parameters from the previous section):

$$H_{max} = \log(L) + l_1 \log(M_{app}) + l_2 \log(v) + l_3 \log(\rho_m) + l_4 \log(\cos(z))$$
$$\rho = 10^{\frac{1}{l_3}[H_{max} - \log(L) - l_1 M_{app} - l_2 \log(v) - l_4 \log(\cos(z))]}$$

5.2.1 Origin of meteors

We can infer that the asteroidal meteoroids have a density larger than 1000 kgm^{-3} while the density of cometary meteoroids are less than that due to the structural differences of the two kinds. Our calculated densities for the meteor showers:

Shower Meteors	Density $\frac{kg}{m^3}$	Likely origin
Leonids	0.61672	Comet
Orionids	0.36578	Comet
Perseids	0.26926	Comet
Geminids	1.48109	Asteroid

It is clear that Geminid meteors have a higher density than that of the other types. Thus we can deduce that Geminids originate from asteroids while Leonids, Orionids, Perseids have a cometary origin.

5.3 Error factors

In our project, errors arise from various factors. First of all, systematic errors come from our model of the atmospheric interaction of a single body, as we used several simplifying assumptions noted in Section 3.1. Another error factor is that the parameters mentioned in Section 3.1 (such as Λ, Γ...) are not actual constant, but rather dependent on other parameters. These errors' magnitude could be determined from a more elaborate simulation.

The measurements themselves have errors as indicated in the previous section. Furthermore, we had two sources of numerical errors: one coming from Euler's method of approximating the equations of interaction and the other one coming from the error of multiple regression fit.

6 Conclusion

By using physical laws and simplifying assumptions, we have consequently been able to presume the origin of meteor showers meteoroids for which data have been measured. The final result of our project was actually a program which calculated the density of any meteoroid as a function of its measured parameters.

Even if the ratio of the cometary/asteroidal meteoroids densities are as expected, the actual values we obtained are off by a factor of 1000. This error can be due to a unit conversion mistake or some other miscalculation (in the regression analysis for example). Through this multi-subject project, we therefore learned and deepened our knowledge in the meteor phenomenon, as well as in general physics, mathematics and programming.

References

[1] Zdenek Ceplecha et al. Meteor phenomena and bodies. *Space Science Reviews*, 84:327–471, 1998.

[2] P. Koten and al. Atmospheric trajectories and light curves of shower meteors. *Astronomy and Astrophysics*, 428:683–690, 2004.

```cpp
//  main.cpp
//  NewEuler
//
//  Software tool implementing Euler's method for simulating meteoric properties
//
//  Created by Marcell Dorian Kovacs on 2017. 08. 04.
//
//  Project: Modeling Meteors
//
//  Summer School of Science
//  Pozega, Croatia
//
//  Young scientists:
//      Chloé Udressy
//      Marcell Dorian Kovacs
//      Nensi Komljenovic
//
//  Project leader:
//      Dušan Pavlovic
//
//  Copyright © 2017. Interplanetary Dušanoids. All rights reserved.
//

#include <iostream>     // For input and output on the console
#include <fstream>      // For file I/O
#include <cmath>        // For mathematical functions

using namespace std;

ofstream MeteorOutput("Meteor2.txt");// Output file for simulation


const double ZeroDensity = 0.122;     // Density of surface air [kg/m^3]
const double MolarMass = 0.029;       // Molar mass of ideal gas athmosphere [kg]
const double GasConstant = 8.314;     // Universal Gas Constant [kg*m^2/s^2/K/mol]
const double Gravity = 9.81;          // Gravitational acceleration on Earth [m/s^2]
const double Temperature = 200.0;     // Temperature of the gas [Kelvin]
const double Lambda = 1.0;            // Heat transfer coefficient (Fraction of energy)
const double Zet = 6e6;               // Latent heat coefficient (Energy lost during heating)
const double Gamma = 1.0;             // Drag coefficient (Fraction of momentum)
const double Shape = 1.21;            // Shape coefficient (Geometrical property of a sphere)

const double h = 0.0001;              // dt [s], timestep, time "instant"

double AthmosphericDensity(double height) {

    // Value of athmospheric density at a certain height
    // Simplifications: we model the athmosphere as an isothermic, ideal gas
    // This formula is also called the simple athmospheric model or bariometric formula

    return ZeroDensity * exp( - (MolarMass * Gravity * height) / (GasConstant * Temperature));

}
double tau(double velocity) {

    // Tau parameter of a meteor
    // Describes the amount of kinetic energy is radiated away in form of (visible) light
    // Connects the kinetic energy to light intensity coming from the body

    const double v = velocity * 0.001; // Velocity must be expressed in [km/s]

    // Coefficients

    const double  c0 =  -0.001365811756924721;
    const double  c1 =  4.03316302906481e-4;
    const double  c2 = -3.837923319944585e-5;
    const double  c3 = 1.605242334632151e-6;
    const double  c4 = -2.995326211721562e-8;
    const double  c5 = 2.405216587475041e-10;
    const double  c6 = -4.425555198678479e-13;
    const double  d1 = -0.1939106756278702;
    const double  d2 = 0.02036594095461284;
    const double  d3 = -0.00130047417574090;
    const double  d4 = 4.580467150003708e-5;
    const double  d5 = -7.505453593442128e-7;
    const double  d6 = 4.80068343434027e-9;

    const double emi = 7.668;

    const double zetav2 = ( c0 + c1 * pow(v, 1.0)+c2 * pow(v, 2.0)+ c3 * pow(v, 3.0) + c4 *
pow(v, 4.0) + c5 * pow(v, 5.0) + c6 * pow(v,6.0) ) / ( 1 + d1 * pow(v,1.0) + d2 * pow(v,
2.0) + d3 * pow(v, 3.0) + d4 * pow(v,4.0 ) + d5 * pow(v, 5.0) + d6 * pow(v, 6.0) );

    // parameter "emi" is the mean energy of excitation divided by a "mu" parameter
```

```
determined from spectral analysis
    // parameter zetav2 is the coefficient of excitation divided by the velocity squared
    // our program and project approximated the value as a rational polynomial, which is only applicable in the range of our velocitites


    return 2 * emi * zetav2;
}


// Functions of equations of motion (originally a differential equation system known as Single Body Theory)

double velo(double velocity, double mass, double height, double density, double z){

    // A single body's velocity at a certain instant with the relevant parameters
    // Coming from transfer of momentum

    return - Gamma * Shape * pow( density , -2.0 / 3.0 ) * AthmosphericDensity(  height ) * pow( mass , -1.0 / 3.0 ) * pow( velocity , 2.0);

}
double mass(double velocity, double mass, double height, double density, double z){

    // A single body's mass at a certain instant with the relevant parameters
    // Coming from transfer of energy

    return -( (Lambda * Shape) / (2.0 * Zet) ) * pow(density, -2.0 / 3.0) * AthmosphericDensity(height) * pow(mass, 2.0 / 3.0) * pow(velocity , 3.0);
}
double height(double velocity, double mass, double height, double density, double z){

    // A single body's height at a certain instant with the relevant parameters
    // Coming from geometry and velocity (the time integral of the "y" axis projection of velocity)
    return - velocity * cos( z );
}
double intensity(double velocity, double mass, double height, double density, double z) {

    // Intensity of light coming from a meteoroid at a certain instant
    // Masschange is the mass() function for the arguments, but I could not include a function call of mass() here

    const double masschange = - ( (Lambda * Shape) / (2.0 * Zet) ) * pow(density, -2.0 / 3.0) * AthmosphericDensity(height) * pow(mass, 2.0 / 3.0) * pow(velocity , 3.0);

    return - ( 1.0 / 2.0 ) * tau(velocity) * pow( velocity , 2.0 ) * masschange ;
}


void Euler(double InitVelo, double InitMass, double InitHeight, double InitDensity, double Initz) {

    // Implementation of Euler's method of approximation to solve the differential equation system

    double Velo = InitVelo;
    double Mass = InitMass;
    double Height = InitHeight;
    double Density = InitDensity;
    double z = Initz;
    double Intensity = intensity(Velo, Mass, Height, Density, z);

    double MaxIntensity = 0;
    double MaxHeight = 0;

    int i = 0;

    while (true) {

        //  In our simplified model, the lifetime of the meteor ends, when it mass decreases under a certain point or we reach a certain point in calculating intensity

        if ( Mass < 0.001*InitMass || Intensity < MaxIntensity / 2)
           break;

        Velo += h * velo(Velo, Mass, Height, Density, z);

        Mass += h * mass(Velo, Mass, Height, Density, z);

        Height += h * height(Velo, Mass, Height, Density, z);

        Intensity = intensity(Velo, Mass, Height, Density, z);

```

```
        // Selecting the maximum of intensity and the height where such maximum occurs (MaxHeight)

        if (Intensity > MaxIntensity) {

            MaxIntensity = Intensity;
            MaxHeight = Height;

        }

        ++i;

        // Commands for outputting the actual light curves
        // MeteorOutput <<  i * h  << "        " << Velo << "       " << Mass << "       " << Height << "       "<< Intensity << endl;
        // cout <<  i * h  << " " << Velo << " " << Mass << " " << Height << " "<< Intensity << " " << MaxIntensity <<  endl;
    }

    // Commands for outputting only the maximums of the light curves

    MeteorOutput/* << InitVelo << " " <<  InitMass << " " << InitDensity << " " << Initz << " " */<< MaxIntensity << " " << MaxHeight << endl;
    cout/* << InitVelo << " " <<  InitMass << " " << InitDensity << " " << Initz << " " */<< MaxIntensity << " " << MaxHeight << endl;

    return;


}

int main(int argc, const char * argv[]) {

    // A little tribute to Hello World! but in physics style

    cout << "Hello Universe!" << endl;

    // Simulation of light curve maximums and height maximums with the following initial parameters for a meteor

    double InitCosZ, InitMass, InitVelo, InitDensity;

    const double InitHeight = 2e5; // Initial height is a paramater for the Euler, but is constant over the simulation

    int counter = 0; // Tracking the number of meteors calculated

    // Looping over several meteors with various initial conditions, saving the set of initials and the output of Euler's method


    for(InitCosZ = 1.0; InitCosZ >= 0.173648; InitCosZ /= pow(10.0 , 0.08))

        for(InitMass = pow(10.0, -8.0); InitMass <= pow(10.0, -1.5); InitMass *= pow(10.0, 0.08))

            for(InitVelo = 11200.0; InitVelo <= 72800.0; InitVelo *= pow(10.0, 0.032))

                for(InitDensity = 300.0; InitDensity <= 3100.0; InitDensity *= pow(10.0, 0.1)) {

                    ++counter;

                        // Printing the initial conditions to the file

                    cout << counter << " " << InitVelo << " " << InitMass << " " << InitDensity << " " << InitCosZ << " ";

                    MeteorOutput << counter << ". " << InitVelo << " " << InitMass << " " << InitDensity << " " << InitCosZ << " ";

                        // Printing the results of Euler's method

                    Euler(InitVelo, InitMass, InitHeight, InitDensity, acos(InitCosZ));

                }

    cout << " A total of " << counter << " meteoroids were simulated." << endl;

    // Return gracefully

    return 0;
}
```