UNIVERSITAT POLITÈCNICA DE CATALUNYA

ENGINYERIA FÍSICA

ASTROPHYSICS and COSMOLOGY

Project Work 1

NUMERICAL INTEGRATION OF AN ELLIPTIC ORBIT

Alexandre Justo and José Javier Ruiz

05/10/2018

Abstract

Halley's Comet is perhaps the most famous comet. It returns to Earth's vicinity about every 75 years, thus allowing human to have the possibility to see it twice. It is the only known short-period comet that is regularly visible to the naked eye from Earth, and the only naked-eye comet that might appear twice in a human lifetime. Last time it approached the Earth was in 1986, and next time is projected to be in 2061. This article tries to make a numerical approach to this comet by implementing a set of equations in `MATLAB`, in order to calculate and draw its orbit, and to compute some important physical magnitudes such as its energy, its angular momentum, or its period.

Figure 1 Halley's Comet on 8 March 1986.

Contents

1 Introduction

The main goal of this project is to compute the trajectory of Halley's Comet around the Sun, as well as its physical and geometrical magnitudes. To do that, it is necessary to obtain its equations of motion. However, before starting a tedious mathematical development, there are some physical considerations that lead to very useful simplifications as for mathematics regards.

The first simplification is that the trajectory can be computed in two dimensions -instead of three dimensions, as one would initially expect- since the angular momentum is conserved. This is because the gravitational force $\vec{F}_g$, given by Newton's Law of universal gravitation, is a central force, that is, it is parallel to the vector radius.

$$\vec{F}_g = -G\frac{Mm}{r^2}\frac{\vec{r}}{r} \tag{1}$$

Hence, the torque $\vec{\tau}= \vec{r}$ x $\vec{F}_g= \dot{\vec{J}}$ is equal to zero and, consequently, the angular momentum is constant. The angular momentum $\vec{J}= \vec{r}$ x $m\dot{\vec{r}}$ is always perpendicular to the vector radius and to the velocity, which means that, since it is constant, the comet trajectory is confined in the plane perpendicular to it.

Thus, from now on, only two dimensions will be considered, belonging to the plane perpendicular to the angular momentum. In this report it is considered that the angular momentum has purely Z-component, then, the trajectory will be in the XY-plane.

Other important considerations are that the solar mass will remain fixed in space (at the origin of coordinates) and that the gravitational force depends on the distance between both masses, which implies that calculations will be easier in polar coordinates.

2 Mathematical foundations

The unitary vectors of the new base are given by:

$$\hat{\eta} = \cos(\theta)\hat{\imath} + \sin(\theta)\hat{\jmath} \tag{2}$$

$$\hat{\tau} = -\sin(\theta)\hat{\imath} + \cos(\theta)\hat{\jmath} \tag{3}$$

Where $\hat{\eta}$ and $\hat{\tau}$ are the unitary vectors parallel and perpendicular to the vector radius, respectively. In this coordinates, the position $\vec{r}$, velocity $\vec{v}$, acceleration $\vec{a}$ and angular momentum $\vec{J}$ have the following expressions:

$$\vec{r} = r\hat{\eta} \tag{4}$$

$$\vec{v} = \dot{r}\hat{\eta} + r\dot{\theta}\hat{\tau} \tag{5}$$

$$\vec{a} = (\ddot{r} - r(\dot{\theta})^2)\hat{\eta} + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\hat{\tau} \tag{6}$$

$$\vec{J} = mr^2\dot{\theta}\hat{k} \tag{7}$$

Finally, to obtain the equations of motion, Newton's Second Law $\vec{F}_g = m\vec{a}$ must be applied. By using (1), (6) and (7) the resulting equations of motion are:

$$\ddot{r} - \frac{J^2}{m^2r^3} + G\frac{M}{r^2} = 0 \tag{8}$$

$$r\ddot{\theta} + 2\dot{r}\dot{\theta} = 0 \tag{9}$$

Equation (9) is actually the conservation of the angular momentum, discussed before, and equation (8) is the second order differential equation that gives the modulus of the radius as a function of time $r(t)$. It can also be demonstrated the dependence of the radius as a function of the polar angle, given by the following expression:

$$r(\theta) = \frac{a(1-e^2)}{1+e\cos(\theta)} \tag{10}$$

Where a is the major semi-axis of the ellipse and e its eccentricity.

Which follows now is to obtain $r(t)$ and $\theta(t)$ in order to compute the trajectory. Besides that, geometrical parameters such as the major semi-axis or the eccentricity are related with physical magnitudes such as the energy, the angular momentum or the period. This relations will be numerically checked along with Kepler Laws.

The main differential equation to be numerically solved is the one that follows.

$$\ddot{r} - \frac{J^2}{m^2 r^3} + G\frac{M}{r^2} = 0 \tag{11}$$

Where r is the modulus of the distance from Halley's Comet to The Sun, J is the modulus of the angular momentum, m is the mass of Halley's Comet, G is the universal gravitational constant, and M is the mass of the Sun.

The way of solving (11) is by transforming one second order differential equation into a system of two first order differential equations. This can be performed by defining an auxiliary variable x_1.

$$\dot{r} = x_1 \tag{12}$$

$$\dot{x_1} = \ddot{r} = \frac{J^2}{m^2 r^3} - G\frac{M}{r^2} \tag{13}$$

Since at the aphelion, which is the point where the radial distance is maximum, the radial speed is zero, an easy set of initial conditions could be:

$$r(t_0) = aphelion \tag{14}$$

$$\dot{r}(t_0) = 0 \tag{15}$$

Then, as it will be indicated in the codes, the parameters needed to perform this numerical integration are: G, M, $aphelion$, a and e. The value of $\frac{J^2}{m^2}$, which will be considered as a parameter in the numerical solving, is computed as follows:

$$\frac{J^2}{m^2} = GMa(1 - e^2) \tag{16}$$

From the solution of (11), which yields $r(t)$ and $\dot{r}(t)$, the angle $\theta(t)$ can be found by using (10). However, the function (x) in `MATLAB` returns angles between 0^{o} and 180^{o}, which only belong to the upper half of the elliptic orbit.

Alternatively, $\theta(t)$ will be obtained by integrating (7) (where it is also needed the value of m) from the values of r previously obtained:

$$\theta(t) = \int_0^t \frac{J}{mr^2} dt \tag{17}$$

Furthermore, it is desired to check the conservation of energy and angular momentum. Therefore, it must be known how to compute them. The energy in polar coordinates can be computed as follows:

$$E = \frac{1}{2} m\dot{r}^2 + \frac{1}{2} \frac{J^2}{mr^2} - G\frac{Mm}{r} \tag{18}$$

It can be also demonstrated that the energy fulfills the following relation:

$$E = -G\frac{Mm}{2a} \tag{19}$$

To compute the angular momentum, one could use whether (16) or (7). As the first option has been used to compute $\frac{J^2}{m^2}$ in the numerical integration, it has been chosen the second option. Nevertheless, performing a numerical derivation of $\theta(t)$ amplifies the numerical error it has already been committed when computing it by integrating (17). Instead, $\dot{\theta}$(t) is obtained by considering (10) with the following formula:

$$\dot{\theta} = \frac{a(1-e^2)}{er^2\sqrt{1-\left(\frac{1}{e}\frac{a(1-e^2)}{r} - \frac{1}{e}\right)^2}}\dot{r} \tag{20}$$

Where $r(t)$ and $\dot{r}(t)$ have been already obtained. It has been checked that the result by means of this method is clearer than the one obtained through numerical derivation.

In the next section, all the equations explained are implemented and solved numerically.

3 Computation and results

There follow three `MATLAB` codes. Main Code, where all the parameters are defined and everything is calculated. Auxiliary Orbit Function, which is a script that includes the system of differential equations formed by (12) and (13). Runge-Kutta 4th Order Integrator, which is responsible for actually solving the aforementioned system.

Since the codes are extensively commented, the only thing to state here is the structure of the Main Code. First, equations of motion are solved along ten complete orbits and solutions for the radius and the angle are plotted in Figure 2. From them, important quantities such as the energy (and the corresponding numerical losses), the angular momentum, or all kind of geometrical orbital parameters are checked to be in agreement with theoretical data, and some of them are plotted in Figure 3. Finally, the three Kepler Laws are extensively checked, with detailed results in Figure 4.

The main results of this section are contained in those three Figures, located at the end of this section.

3.1 Main Code

```
%% ASTROPHYSICS and COSMOLOGY - Project Work 1

% Numerical integration of an elliptic orbit, by A. Justo and J. ...
    J. Ruiz.


%% General constants of the Solar System

G  = 6.67408*10^-11;        % Gravitational constant in [m^3 ...
    kg^-1 s^-2].
M  = 1.98847*10^30;         % Solar mass in [kg].
AU = 149597870700;          % Astronomical Unit in [m].
yr = 365.25*24*3600;        % Year in [s].


%% Data of the orbiting body: Halley's Comet

aph = 35.082*AU;            % Aphelion in [m].
a   = 17.834*AU;            % Major semiaxis in [m].
e   = 0.96714;              % Eccentricity (adimensional).
m   = 2.2*10^14;            % Mass in [kg].
Jm  = G*M*a*(1-e^2);        % Parameter equal to J^2/m^2 in SI units.
Jtheoretical = m*sqrt(Jm); % Angular momentum in SI units.


%% Estimation of the period by means of 3rd Kepler Law

% Period expected in [yr].
Ttheoretical=sqrt(4*pi^2*a^3/(G*M))/yr;


%% Parameters for the ODE solver

```

```
% First instant of integration.
t0 = 0;

% Final instant of integration, corresponding to 10 complete orbits.
tN = 10*Ttheoretical*yr;

% Time step.
k  = tN/10^5;

% Initial conditions. Impose that, at t=t0, the comet is located ...
    at the
% aphelium, where dr(t)/dt = 0.
initcond = [aph; 0];


%% Integration and results of the integration

% Obtain r(t) and dr(t)/dt.
rk4out = rk4(@orbit,initcond,Jm,k,tN,t0);
t       = rk4out(1,:);
r       = rk4out(2,:);
rdot    = rk4out(3,:);

% Numerically calculated aphelion.
aphelion=max(r);

% Numerically calculated perihelion.
perihelion=min(r);


%% Computing theta as a function of time

% Integrate dtheta(t)/dt = J/(m*r^2). Do the integration as the ...
    sum of
% trapezoid areas.

fvec=r.^-2;

% Preallocating for speed.
theta=zeros(1,length(t));

for i=2:length(t)
    if i==2
    theta(i) = sqrt(Jm)*k*(0.5*fvec(1) + 0.5*fvec(i));
    else
    theta(i) = sqrt(Jm)*k*(0.5*fvec(1) + sum(fvec(2:i-1)) + ...
        0.5*fvec(i));
    end
end


%% Plot r(t) and theta(t) as a function of time

figure(1)
set(gcf,'Position',[175 350 1000 500])

subplot(2,1,1)
plot(t/yr,r/AU,'b',t/yr,aphelion*ones(1,length(t))/AU,'--g',t/yr,...
    perihelion*ones(1,length(t))/AU,'--r')
title('1P/Halley'); xlabel('t [years]'); ylabel('r [AU]'); grid
legend('r(t)','Aphelion','Perihelion')

```

```
subplot(2,1,2)
plot(t/yr,theta*180/pi); grid
title('1P/Halley'); xlabel('t [years]'); ylabel('\theta [ ]')


%% Obtain the corresponding cartesian coordinates

x      = r.*cos(theta);
y      = r.*sin(theta);


%% Calculate the energy, which must be conserved

% Compute E(t).
E = 1/2*m*rdot.^2+1/2*Jm*m*r.^(-2)-G*M*m./r;

% Compute the % of lost E over all the integration.
lostE = 100*abs(1-E(end)/E(1));

% Since tN corresponds to 10 orbits...
lostEperOrbit = lostE/10;

% Compute the theoretical E, which must be conserved.
Etheoretical = -G*M*m/(2*a);


%% Calculate the angular momentum, which must be conserved

% Compute dtheta(t)/dt from the expression r(theta).
thetadot = ...
    a*(1-e^2)./(e*r.^2.*sqrt(1-(1/e*a*(1-e^2)./r-1/e).^2)).*rdot;

% Compute J(t).
J = m*r.^2.*thetadot;


%% Calculate the geometrical orbital parameters

% Numerically calculated major semiaxis in [AU].
a_num  = 0.5*(aphelion+perihelion)/AU

% Numerically calculated minor semiaxis in [AU].
b_num  = max(abs(y))/AU

% Numerically calculated eccentricity (three different ways).
e_num1 = sqrt(1-(b_num/a_num)^2)
e_num2 = sqrt(1-4*aphelion*perihelion/(aphelion+perihelion)^2)
e_num3 = (G^2*M^2*m^4/Jtheoretical^4+...
    2*m*Etheoretical/Jtheoretical^2)^(1/2)*Jtheoretical^2/(G*M*m^2)


%% Plot the energy and the angular momentum as a function of time

figure(2)
set(gcf,'Position',[175 350 1000 500])

subplot(2,1,1)
plot(t/yr,E,t/yr,Etheoretical*ones(1,length(t)),'--r')
title('1P/Halley')
xlabel('t [years]'); ylabel('E [J]'); grid
legend('Numerical E','Theoretical E')
```

```
subplot(2,1,2)
plot(t/yr,abs(J),t/yr,Jtheoretical*ones(1,length(t)),'--r')
title('1P/Halley')
xlabel('t [years]'); ylabel('|J| [kg m^2 s^-^1]'); grid
legend('Numerical |J|', 'Theoretical |J|')


%% Checking 1st Kepler Law: Elliptic trajectory with the Sun at ...
    a focus

% Check that the Sun is located at one of the focus.

% The first focus is located at the origin.
F1=[0,0];

% The second focus is located, from the aphelion, at a distance ...
    equal
% than the perihelion.
F2=[aphelion-perihelion,0];

% Now compute the sum of the distances of each point of the ...
    orbit from
% F1 and F2.
dF1=sqrt(x.^2+y.^2);
dF2=sqrt((x-F2(1)*ones(1,length(x))).^2+y.^2);

% Numerically calculated sum of the distance to both focuses, in ...
    [AU].
sum_numerical   =mean(dF1+dF2)/AU
sum_theoretical=2*a/AU


%% Checking 2nd Kepler Law: Equal areas are swept in equal times

% Compute the area swept in a given time.
x2=[0, x(1:2000)];
y2=[0, y(1:2000)];
area1=polyarea(x2/AU,y2/AU)

% Compute the area swept in the same time.
x3=[0, x(2001:2000*2)];
y3=[0, y(2001:2000*2)];
area2=polyarea(x3/AU,y3/AU)


%% Checking 3rd Kepler Law. Period

% Find the time it takes to come back to the aphelion. To do ...
    that, find
% the peak of r(t) when it arrives to the aphelion, as plotted ...
    in Fig (1).
[pks,locs] = findpeaks(r/AU, 'MinPeakHeight', aphelion/AU-1);

% The period is:
T=t(locs(1))/yr


%% Plot the resulting orbit and swept areas

figure(3)
set(gcf,'Position',[175 350 1000 500])
```

```
% Check that the points of the orbit fulfill expression of r(theta)
thetavec=linspace(0,2*pi,200);
r_theta=a*(1-e^2)./(1+e*cos(thetavec));
x_theta=-r_theta.*cos(thetavec);
y_theta=r_theta.*sin(thetavec);

plot(x/AU,y/AU,'k',x_theta/AU,y_theta/AU,'.b','markersize',10); ...
    hold on
plot(0,0,'.r','markersize',20);
title('1P/Halley');
xlabel('x [AU]'); ylabel('y [AU]'); axis equal; hold on; grid


fill(x2/AU,y2/AU,'g'); hold on; grid
fill(x3/AU,y3/AU,'r')

legend('Orbit (1st way)','Orbit (2nd way)','Sun (focus)'); grid; ...
    xlim([-5 40])
text(20,1,['Area 1 = ' num2str(area1) ' AU^2'])
text(15,3,['Area 2 = ' num2str(area2) ' AU^2'])

text(15,-1.5,['a = ' num2str(a/AU) ' AU'])
text(15,-2.5,['T = ' num2str(T) ' years'])
```

3.2 Auxiliary Orbit Function

See equations (12) and (13).

```
function ODEsystem=orbit(v0,Jm,¬)

G=6.67408*10^-11; % Gravitational constant in [m^3 kg^-1 s^-2].
M=1.98847*10^30;  % Solar mass in [kg].


r=v0(1);
aux=v0(2);

eq1=aux;              % r'
eq2=Jm*r^-3-G*M*r^-2; % aux'
ODEsystem=[eq1;eq2];
end
```

3.3 Runge-Kutta 4th Order Integrator

This algorithm integrates equation (11) to obtain $r(t)$ and $\dot{r}(t)$.

```
function rk4out = rk4(F,initcond,Jm,k,tN,t0)

t=t0:k:tN;

auxmat(:,1)=initcond;

for i = 1 : (tN-t0)/k

    a = k*feval(F, auxmat(:,i),     Jm, k*(i-1)     );
    b = k*feval(F, auxmat(:,i)+a/2, Jm, k*(i-1)+k/2 );
    c = k*feval(F, auxmat(:,i)+b/2, Jm, k*(i-1)+k/2 );
```

```
    d = k*feval(F, auxmat(:,i)+c,    Jm, k*(i-1)+k    );

    auxmat(:,i+1) = auxmat(:,i) + 1/6*( a + 2*b + 2*c + d );
end

rk4out=[t;auxmat(:,:)];
% First row is t. Second row is r. Third row is rdot.

end
```

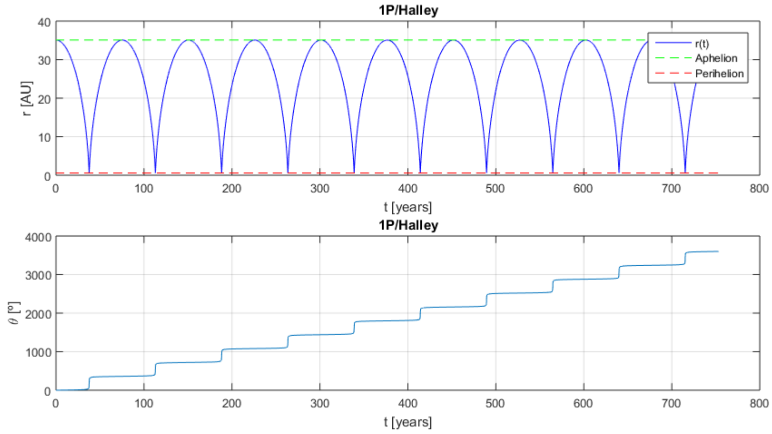

Figure 2 $r(t)$ and $\theta(t)$.

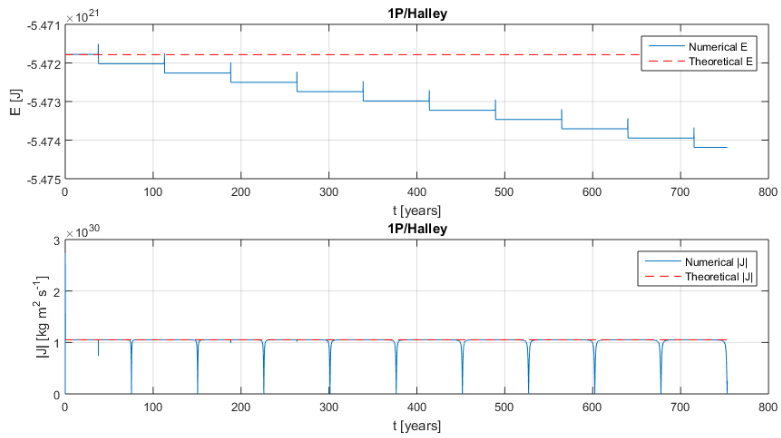

Figure 3 $E(t)$ and $\|J\|(t)$. Numerical values of $E(t)$ and $\|J\|(t)$ are computed with (19) and (7), respectively.

In Figure 2 it is shown how $r(t)$ oscillates periodically between the aphelion and the perihelion, whose values have been computed numerically: $35.0820AU$ and $0.5860AU$, respectively. Taking into account $\theta(t)$, when the comet is at the aphelion the angle is 0^{o}, which means that the aphelion is at the right of the origin of coordinates and the perihelion is at the left, as can be verified in Figure

(4). In addition, from both representations of $r(t)$ and $\theta(t)$ the period can be computed, which gives a numerical value of 75.3151 years.

From Figure 3 one can realize that, although the energy should be conserved, it is not. This is mainly due to numerical errors that cause energy not to be conserved. These numerical error is as small as the 0.0044005% of the initial energy per one orbit. Note that the initial energy numerically calculated is equal to the total theoretical energy (see equation (19)).

Except for punctual numerical singularities, the modulus of the angular momentum is fully conserved.

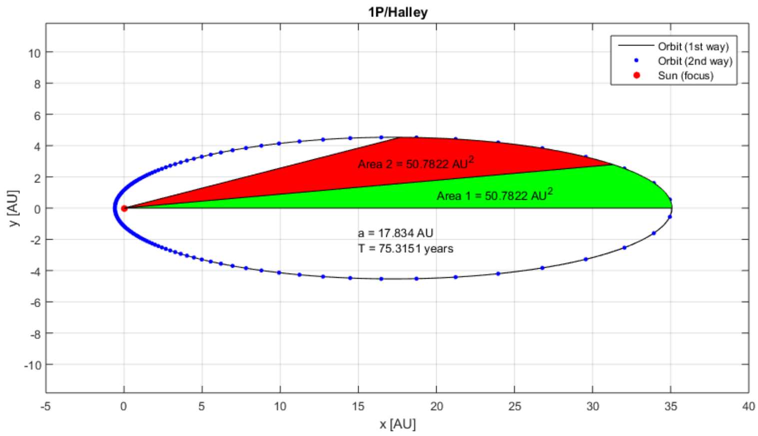

Figure 4 Halley's Comet orbit and Kepler laws.

Figure 4 shows the results of drawing the orbit. It has been calculated in two ways. The first one, in black, is obtained by solving (11) and, from it, integrating equation (17). The second one, in blue dots, is by taking a set of angles and plugging them into equation (10). By this, it has been checked that both expressions are consistent.

Regarding Kepler laws...

- First Kepler Law is fulfilled, since the comet describes an elliptic orbit around the Sun, located at a focus. The values of the major and minor semi-axis numerically computed are $a = 17.8340AU$ and $b = 4.5356AU$ To show that the Sun is located at a focus, there have been computed the sum of distances from each point of the orbit to both focuses $F1 = (0, 0)$ and $F2 = (aphelion - perihelion, 0)$, which gives an average of $35.6646AU$. This differs very little from the theoretical value of $2a = 35.6680AU$. Except for numerical errors, it can be considered that the Sun is located at focus F1.

- Second Kepler Law is fulfilled, since areas swept in equal times are the same. To check that, the area swept in a given interval of time has been computed, and then the area swept in the same contiguous interval of time. Both areas have been painted in the orbit and have the same value, as it was expected.

- Third Kepler Law is also fulfilled. According to this law, the expected period is given by:

$$T^2 = \frac{4\pi^2 a^3}{GM} \tag{21}$$

Which gives a value of $75.3151 years$. The period computed numerically from Figure 2 is the same, as it should be.

Along this report, there have been used several equations which relate geometrical parameters of the orbit, such as a or e, with physical magnitudes, such as energy or angular momentum (equations (19) and (7), for instance).

The eccentricity is also related with these physical magnitudes. Actually, it has been computed in three different ways:

$$e_1 = \sqrt{1 - \frac{b^2}{a^2}} = 0.9671 \tag{22}$$

$$e_2 = \sqrt{1 - 4\frac{aphelion * perihelion}{(aphelion + perihelion)^2}} = 0.9671 \tag{23}$$

$$e_3 = \frac{J^2}{GMm^2}\sqrt{\frac{G^2M^2m^4}{J^4} + \frac{2mE}{J^2}} = 0.9671 \tag{24}$$

All three ways yield the same result. Parameters a, b, *aphelion* and *perihelion* are the numerically computed ones. In equation (24), E and J are the theoretical ones.

4 Conclusions

The orbit of Halley's Comet around the Sun can be properly reproduced starting from its basic parameters and solving the corresponding equations of motion.

In spite of not having considered any other of the bodies belonging to the Solar System, the trajectory seems to be accurate and the results are satisfactory. Perhaps the huge mass of the Sun, which accounts for the 99.85% of the total mass of the Solar System, is the factor responsible for that.

All the results obtained are in agreement with theoretical data, except for the case of the energy, which is not conserved, perhaps due to numerical imprecision.

References

[1] Data from the comet. `https://minorplanetcenter.net/db_search/show_object?utf8=%E2%9C%93&object_id=1P`.

[2] Gravitational constant. `https://physics.nist.gov/cgi-bin/cuu/Value?bg`.

[3] Overview of the comet. `https://solarsystem.nasa.gov/small-bodies/comets/1p-halley/in-depth/`.

[4] Space. `https://www.space.com/19878-halleys-comet.html`.

[5] Wikipedia. `https://en.wikipedia.org/wiki/Halley%27s_Comet`.