## 1. Equations of motion in classical physics

### A. Newton's equations of motion

In classical (Newtonian) mechanics, the equations of motion define a *trajectory* of any particle. In Newton's formulation, equations of motion for a particle with mass *m* in Cartesian coordinate system are given by

$$F_x = m\frac{d^2x}{dt^2}, \quad F_y = m\frac{d^2y}{dt^2}, \quad F_z = m\frac{d^2z}{dt^2} \tag{1}$$

where $x$, $y$, $z$ are the Cartesian coordinates of the particle,

$$\vec{\mathbf{r}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{2}$$

$F_x$, $F_y$, $F_z$ are the three components of the force acting on the particle,

$$\vec{\mathbf{F}} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}, \tag{3}$$

and $t$ is time. Recall that the second derivative of a coordinate with respect to (w.r.t.) time defines particle's *acceleration*, i.e.

$$a_x = \frac{d^2x}{dt^2}, \quad a_y = \frac{d^2y}{dt^2}, \quad a_z = \frac{d^2z}{dt^2} \tag{4}$$

Thus, equation (1) can be re-written in the vector notation as

$$\vec{\mathbf{F}} = m\vec{\mathbf{a}} \tag{5}$$

which is a common way of writing *Newton's second law of motion*. Note that acceleration $\vec{\mathbf{a}}$ is also related to particle's *velocity* $\vec{\mathbf{v}}$ as

$$\vec{\mathbf{a}} = \frac{d\vec{\mathbf{v}}}{dt} \tag{6}$$

where the velocity of a particle is defined as:

$$\vec{\mathbf{v}} = \frac{d\vec{\mathbf{r}}}{dt} \tag{7}$$

$$v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}, \quad v_z = \frac{dz}{dt} \tag{8}$$

Finally, defining particle's *linear momentum* $\vec{\mathbf{p}}$ as

$$\vec{\mathbf{p}} = m\vec{\mathbf{v}} \tag{9}$$

$$p_x = m\frac{dx}{dt}, \quad p_y = m\frac{dy}{dt}, \quad p_z = m\frac{dz}{dt} \tag{10}$$

Newton's equations of motion can be written as:

$$\vec{\mathbf{F}} = m\vec{\mathbf{a}} = m\frac{d\vec{\mathbf{v}}}{dt} = \frac{d\vec{\mathbf{p}}}{dt} \tag{11}$$

under the assumption that the mass of the particle is constant. Newton's equations of motion along with the initial position and momentum of a particle, give us $\vec{\mathbf{r}}(t)$ which describes the

position of the particle as a function of time [1]. The three-dimensional path described by $\vec{\boldsymbol{r}}(t)$ is called the *trajectory* of the particle. The trajectory of a particle offers a complete description of the state of the particle. Classical mechanics provides a method for calculating the trajectory of a particle in terms of the forces acting upon the particle through Newton's equations (1), (5), and (11) [1].

**B. The Lagrangian function, and equations of motion in the Lagrangian form**

Newton's laws of motion properly explain the motion of macroscopic bodies, but there is always more than one way to model the behavior of an object [2]. For example, let's define the *kinetic energy* $T$ of a particle with the mass $m$ as [3]

$$T = \frac{1}{2} m(\vec{\mathbf{v}}^{\mathrm{T}}\vec{\mathbf{v}}) \tag{12}$$

$$T = \frac{1}{2} m\left(v_x^2 + v_y^2 + v_z^2\right) \tag{13}$$

Note that the kinetic energy of a particle is due solely to the velocity of that particle [2].

If we limit ourselves to a certain class of systems, called *conservative systems*, it is possible to define another quantity, the potential energy $V$, which is a function of the coordinates $x, y, z$ of the particle,

$$V \equiv V(x, y, z) \tag{14}$$

such that the force components acting on the particle is equal to partial derivative of the potential energy w.r.t. the coordinates of the particle (with a negative sign); that is

$$F_x = -\frac{\partial V}{\partial x}, \quad F_y = -\frac{\partial V}{\partial y}, \quad F_z = -\frac{\partial V}{\partial z} \tag{15}$$

It is possible to find a function $V$ which will express in this manner forces of the types usually designated as mechanical, electrostatic, and gravitational [3]. With these definitions, Newton's equations become [3]

$$\begin{aligned} \frac{d}{dt}\frac{\partial T}{\partial v_x} + \frac{\partial V}{\partial x} &= 0 \\ \frac{d}{dt}\frac{\partial T}{\partial v_y} + \frac{\partial V}{\partial y} &= 0 \\ \frac{d}{dt}\frac{\partial T}{\partial v_z} + \frac{\partial V}{\partial z} &= 0 \end{aligned} \tag{16}$$

These results are definitely restricted to Cartesian coordinates; but by introducing a new function, the *Lagrangian function* $L$, defined for Newtonian systems as the difference of the kinetic and potential energy [3],

$$L \equiv L(x, y, z) = T - V \tag{17}$$

the equations of motion can be written in a form that is valid in *any* system of coordinates. In Cartesian coordinates, $T$ is a function of the velocity $\vec{\mathbf{v}}$ only, and for the systems to which our treatment is restricted, $V$ is a function of the coordinates only. Hence the equations of motion given in equation (16) on introduction of the function $L$ assume the form [3]

$$\frac{d}{dt}\frac{\partial L}{\partial v_x} - \frac{\partial L}{\partial x} = 0$$
$$\frac{d}{dt}\frac{\partial L}{\partial v_y} + \frac{\partial L}{\partial y} = 0 \qquad (18)$$
$$\frac{d}{dt}\frac{\partial L}{\partial v_z} + \frac{\partial L}{\partial z} = 0$$

**C. Generalized coordinates and equations of motion in the Lagrangian form**

Instead of Cartesian coordinates $x$, $y$, $z$ it is frequently more convenient to use some other set of coordinates to specify the configuration of the system [3]. If we choose, any set of three coordinates, which we shall always assume to be independent and at the same time sufficient in number to specify completely the positions of the particles of the system, then there will in general exist three equations, called the *equations of transformation*, relating the new coordinates $q_k$ ($k = 1 \ldots 3$) to the set of Cartesian coordinates $x$, $y$, $z$ [3]:

$$x = f(q_1, q_2, q_3)$$
$$y = g(q_1, q_2, q_3) \qquad (19)$$
$$z = h(q_1, q_2, q_3)$$

The functions $f,\ g,\ h$ may be functions of any or all of the three new coordinates $q_k$ [3].

Using a new set of generalized coordinates, *equations of motion in the Lagrangian form* are written as [3]

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} = 0$$
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} = 0 \qquad (20)$$
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_3} - \frac{\partial L}{\partial q_3} = 0$$

or in a more compact form as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = 0, \quad k = 1 \ldots 3 \qquad (21)$$

where dots refer to differentiation w.r.t. time $t$, such as

$$\dot{q}_k = \frac{dq_k}{dt}, \quad k = 1 \ldots 3 \qquad (22)$$

and quantities $\dot{q}_k$ are called *generalized velocities*, even though they do not necessarily have the dimensions of length divided by time (for example, $\dot{q}_k$ may be an angle) [3].

Note that when Newton's equations are written in the form given by equations (20) and (21) they are valid for any choice of coordinate system, and by a proper choice of the function *L* nearly all dynamical problems can be treated with their use [3]. These equations are therefore

frequently chosen as the fundamental postulates of classical mechanics instead of Newton's equations.

Lagrange's equations, mathematically equivalent to Newton's equations, rely on being able to define the kinetic and potential energy of a system rather than the forces acting on the system [2]. Depending on the system, Lagrange's differential equations of motion can be easier to solve and understand than Newton's differential equations of motion [2].

**D. Generalized momenta** [3]

In Cartesian coordinates the momentum related to the direction is given by equations (9) and (10), or adopting the dot notation [3],

$$p_k = m_k \dot{x}_k, \quad k = 1 \dots 3 \tag{23}$$

which, since $V$ is restricted to be a function of the coordinates only, can be written as [3]

$$p_k = \frac{\partial T}{d\dot{x}_k} = \frac{\partial L}{d\dot{x}_k}, \quad k = 1 \dots 3 \tag{24}$$

The same way we defined the generalized velocities, we can define the *generalized momentum* $p_k$ conjugate to the coordinate $q_k$ as [3]

$$p_k = \frac{\partial L}{\partial \dot{q}_k}, \quad k = 1 \dots 3 \tag{25}$$

The form taken by Lagrange's equations (20) and (21) when the definition of $p_k$ is introduced is [3]

$$\dot{p}_k = \frac{\partial L}{\partial q_k}, \quad k = 1 \dots 3 \tag{26}$$

so that equations (25) and (26) form a set of six first-order differential equations equivalent to the three second-order equations (20) and (21) [3].

The quantity $\frac{\partial L}{\partial q_k}$ being in general a function of both the $q$'s and $\dot{q}$'s, the definition of $p_k$ given by equation (25) provides 3 relations between the variables $q_k$, $\dot{q}_k$ and $p_k$, permitting the elimination of the 3 velocities $\dot{q}_k$ so that the system can now be described in terms of the 3 coordinates $q_k$ and the 3 *conjugate momenta* $p_k$ [3].

**E. The Hamiltonian function and equations** [3]

The *Hamiltonian function* ("*the Hamiltonian*") is defined as [3]

$$H = \left( \sum_{k=1}^{3} p_k \dot{q}_k \right) - L(q_k, \dot{q}_k) \tag{27}$$

For *conservative systems*, i.e. systems in which $H$ does not depend explicitly on the time $t$, the function $H$ is the total energy (kinetic plus potential) of the system, expressed in terms of the $p_k$'s and $q_k$'s [3]. Although the definition of $H$ involves the velocities $\dot{q}_k$, $H$ may be made a function of the coordinates and momenta only, by eliminating the velocities through the use of equation (25) [3]. Differentiating $H$ w.r.t time

$$H = \left(\sum_{k=1}^{3} p_k d\dot{q}_k\right) + \left(\sum_{k=1}^{3} \dot{q}_k dp_k\right) - \left(\sum_{k=1}^{3} \frac{\partial L}{\partial q_k} dq_k\right) - \left(\sum_{k=1}^{3} \frac{\partial L}{\partial \dot{q}_k} d\dot{q}_k\right) \tag{28}$$

$$H = \sum_{k=1}^{3} (\dot{q}_k dp_k - \dot{p}_k dq_k) \tag{29}$$

one obtains the *equations of motion in the Hamiltonian* or *canonical form* [3]:

$$\left.\begin{aligned} \frac{\partial H}{\partial p_k} &= \dot{q}_k \\ \frac{\partial H}{\partial q_k} &= -\dot{p}_k \end{aligned}\right\} \quad k = 1 \ldots 3 \tag{30}$$

There are two equations for each of the three spatial dimensions [2]. For one particle in three dimensions, equations (30) give six first-order differential equations that need to be solved in order to understand the behavior of the particle. Both Newton's equations and Lagrange's equations require the solution of three second-order differential equations for each particle, so that the amount of calculus required to understand the system is the same [2]. The only difference lies in what information one knows to model the system or what information one wants to get about the system. This determines which set of equations to use. Otherwise, they are all mathematically equivalent [2].

## 2. Planetary motion

### A. The Gravitational Force [4]

The *gravitational force* is the mutual force of attraction between any two objects in the Universe [4]. Although the gravitational force can be very strong between very large objects, it's the weakest of the fundamental forces. A good demonstration of how weak it is can be carried out with a small balloon. Rubbing the balloon in your hair gives the balloon a tiny electric charge. Through electric forces, the balloon then adheres to a wall, resisting the gravitational pull of the entire Earth! [4]

In addition to contributing to the understanding of motion, Newton studied gravity extensively [4]. Newton's law of universal gravitation states that [4]

> *Every particle in the Universe attracts every other particle with a force that is directly proportional to the product of the masses of the particles and inversely proportional to the square of the distance between them.*

If the particles have masses $m_1$ and $m_2$ and are separated by a distance $r$, as shown in Figure 1, the magnitude of the gravitational force $F_g$ is [4]

$$F_g = \frac{G m_1 m_2}{r^2} \tag{31}$$

where $G$ = 6.67384×10$^{-11}$ N m$^2$ kg$^{-2}$ is the *universal gravitation constant*. This force law is an example of an *inverse-square law*, in that it varies as one over the square of the separation of the particles.

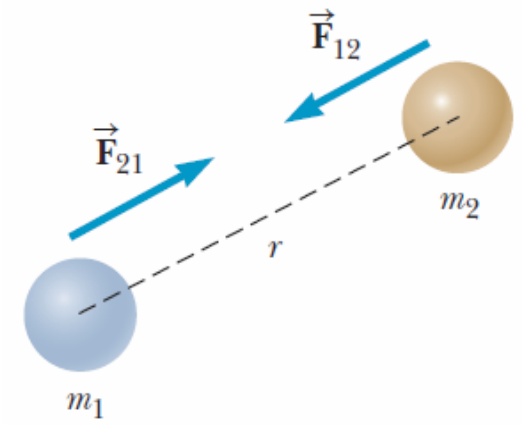



**Figure 1.** The gravitational force between two particles is attractive and acts along the line joining the particles [4].

According to *Newton's third law* [4],

*If object 1 and object 2 interact, the force* $\vec{\mathbf{F}}_{\mathbf{12}}$ *exerted by object 1 on object 2 is equal in magnitude but opposite in direction to the force* $\vec{\mathbf{F}}_{\mathbf{21}}$ *exerted by object 2 on object 1*,

which means that *a single isolated force can not exist*. The force $\vec{\mathbf{F}}_{\mathbf{12}}$ exerted by object 1 on object 2 is sometimes called the *action force*, and the force $\vec{\mathbf{F}}_{\mathbf{21}}$ exerted by object 2 on object 1 is called the *reaction force*. In reality, either force can be labeled the action or reaction force. The action force is equal in magnitude to the reaction force and opposite in direction. In all cases, the action and reaction forces act on different objects [4]. Thus, the force exerted by $m_1$ on $m_2$, designated $\vec{\mathbf{F}}_{\mathbf{12}}$ in Figure 1, is equal in magnitude but opposite in direction to the force $\vec{\mathbf{F}}_{\mathbf{21}}$ exerted by $m_2$ on $m_1$, forming an action–reaction pair.

Another important fact is that the gravitational force exerted by a uniform sphere on a particle outside the sphere is the same as the force exerted if the entire mass of the sphere were concentrated at its center. This is called *Gauss's law*, after the German mathematician and astronomer Karl Friedrich Gauss. Gauss's law is a mathematical result, true because the force falls off as an inverse square of the separation between the particles.

**B. Kepler's laws** [4]

The movements of the planets, stars, and other celestial bodies have been observed for thousands of years. In early history scientists regarded Earth as the center of the Universe. This *geocentric model* (Figure 2a) was developed extensively by the Greek astronomer Claudius Ptolemy in the second century A.D. and was accepted for the next 1,400 years [4].

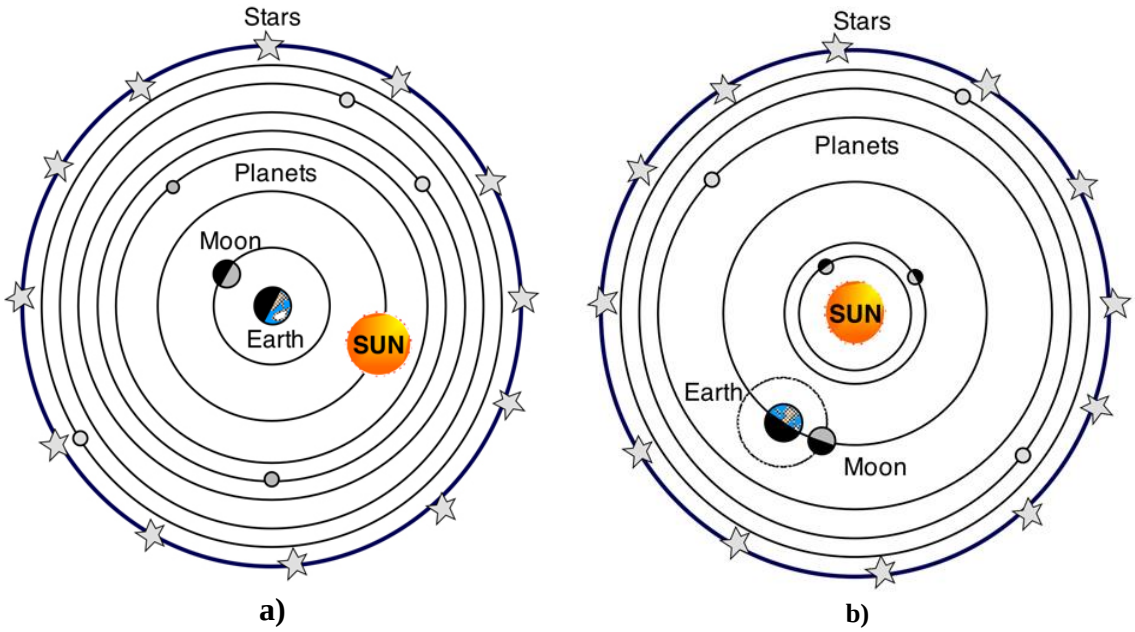



**Figure 2**. A hypothetical geocentric model of the solar system (**a**) in comparison to the heliocentric model (**b**). [5]

In 1543 Polish astronomer Nicolaus Copernicus (1473–1543) showed that Earth and the other planets revolve in circular orbits around the Sun (the *heliocentric model*), as shown in Figure 2b. Danish astronomer Tycho Brahe made accurate astronomical measurements over a period of 20 years, providing the data for the currently accepted model of the solar system [4]. Brahe's precise observations of the planets and 777 stars were carried out with nothing more elaborate than a large sextant and compass; the telescope had not yet been invented [4].

German astronomer Johannes Kepler, who was Brahe's assistant, acquired Brahe's astronomical data and spent about 16 years trying to deduce a mathematical model for the motions of the planets. After many laborious calculations, he found that Brahe's precise data on the motion of Mars about the Sun provided the answer. Kepler's analysis first showed that the concept of circular orbits about the Sun had to be abandoned. He eventually discovered that the orbit of Mars could be accurately described by an ellipse with the Sun at one on the *foci*. He then generalized this analysis to include the motions of all planets. The complete analysis is summarized in three statements known as Kepler's laws [4]:

1. *All planets move in elliptical orbits with the Sun at one of the focal points.*

2. *A line drawn from the Sun to any planet sweeps out equal areas in equal time intervals.*

3. *The square of the orbital period of any planet is proportional to the cube of the average distance from the planet to the Sun.*

Newton later demonstrated that these laws are consequences of the gravitational force that exists between any two objects. Newton's law of universal gravitation, together with his laws of motion, provides the basis for a full mathematical description of the motions of planets and satellites [4].

**B.1 Kepler's First Law** [4]

The first law arises as a natural consequence of the inverse-square nature of Newton's law of gravitation [4]. Any object bound to another by a force that varies as $1/r^2$ will move in an *elliptical orbit* [4]. As shown in Figure 3a, an *ellipse* is a curve drawn so that the sum of the distances from any point on the curve to two internal points called *focal points* or *foci* (singular, *focus*) is always the same [4].

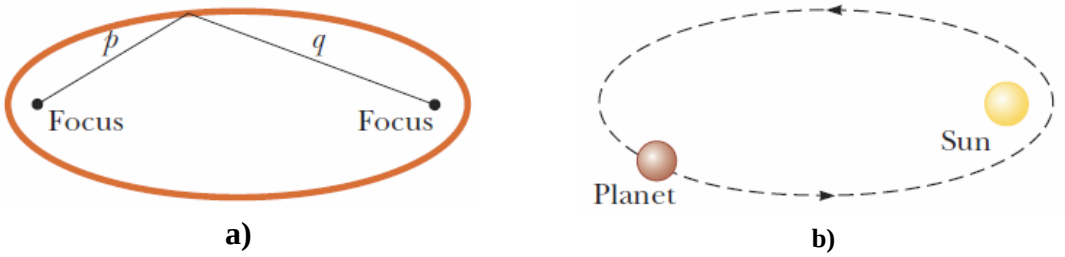



**Figure 3**. (**a**) The sum $p+q$ is the same for every point on the ellipse. (**b**) In the Solar System, the Sun is at one focus of the elliptical orbit of each planet and the other focus is empty [4].

For the Sun-planet configuration (Figure 3b), the Sun is at one focus and the other focus is empty. Because the orbit is an ellipse, the distance from the Sun to the planet continuously changes.

Ellipses have two mutually perpendicular axes about which the ellipse is symmetric [6]. These axes intersect at the center of the ellipse due to this symmetry. The larger of these two axes, which corresponds to the largest distance between antipodal points on the ellipse, is called the *major axis* or *transverse diameter*. The smaller of these two axes, and the smallest distance across the ellipse, is called the *minor axis* or *conjugate diameter*. [6]

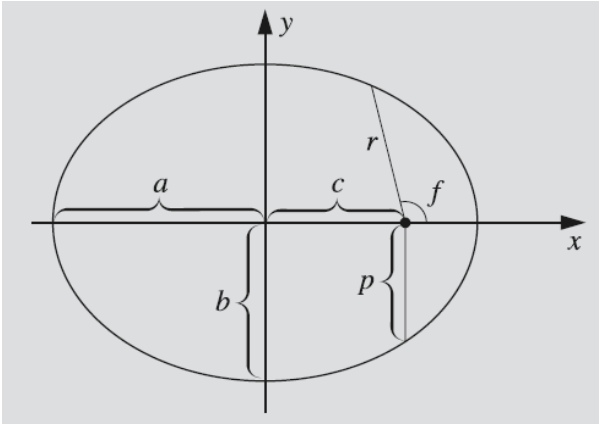



**Figure 4**. The ellipse and some of its mathematical properties.
The Sun is located at one the foci shown with "•". [7]

The *semimajor axis* (denoted by $a$ in the Figure 4) and the *semiminor axis* (denoted by $b$ in the Figure 4) are one half of the major and minor axes, respectively. These are sometimes called (especially in technical fields) the *major* and *minor semi-axes*, the *major* and *minor semiaxes*, or *major radius* and *minor radius* [6]. Note that the sum of the distances $p$ and $q$ as shown in Figure 3a is constant as equal to the length of the major axis:

$$p + q = 2a \tag{32}$$

The equation of the ellipse in Cartesian coordinates is [7]

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \tag{33}$$

where $x$ and $y$ are the coordinates of any point on the ellipse. An ellipse can be defined as the *locus* of all points that satisfy equation (33). The area of the ellipse $A$ is given by [7]

$$A=\pi ab \tag{34}$$

The *eccentricity* of an ellipse, usually denoted by $e$, is the ratio of the distance between the two foci, to the length of the major axis [6]

$$e=\frac{2c}{2a}=\frac{c}{a} \tag{35}$$

Thus, the distance from the center of the ellipse to each of the foci is [7]

$$c=ae \tag{36}$$

For an ellipse the eccentricity is between 0 and 1 ($0<e<1$) [6]. When the eccentricity is 0 the foci coincide with the center point and the figure is a circle. As the eccentricity tends toward 1, the ellipse gets a more elongated shape [6]. At $e=1$, the ellipse is transformed into a parabola, and at $e>1$ it becomes a hyperbola [7].

From the definition of the eccentricity, the length of the semiminor axis $b$ is [7]

$$b=a\sqrt{1-e^2} \tag{37}$$

In astronomy, the distance $p$ (as defined in Figure 4) is called the *similatus rectum* and is given by [7]

$$p=a(1-e^2) \tag{38}$$

The angle $f$ which is measured from the point where the planet is the closest to the Sun (called the *perihelion*) is called the *true anomaly*. In terms of the true anomaly, eccentricity, and similatus rectum, the equation of the ellipse is [7]

$$r=\frac{p}{1+e\cos f} \tag{39}$$

where the $r$ is measured from one of the foci, not from the center [7].

**B.2 Kepler's Second Law** [4]

Kepler's second law states that a line drawn from the Sun to any planet sweeps out equal areas in equal time intervals. Consider a planet in an elliptical orbit about the Sun (Figure 5) [4].

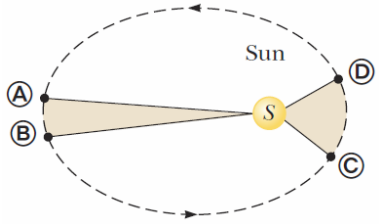



**Figure 5**. The two areas swept out by the planet in its elliptical orbit about the Sun are equal if the time interval between points A and B is equal to the time interval between points C and D [4].

In a given period $\Delta t$, the planet moves from point A to point B [4]. The planet moves more slowly on that side of the orbit because it's farther away from the Sun. On the opposite side of its orbit, the planet moves from point C to point D in the same amount of time, $\Delta t$, moving faster because it's closer to the Sun. Kepler's second law says that any two wedges formed as in Figure 5 will always have the same area. Kepler's second law is related to a physical principle known as *conservation of angular momentum* [4]. It follows that the planet will have the greatest speed at the *perihelion* of the ellipse, the closest point to the Sun, and the lowest speed at the *aphelion* of the ellipse, the farthest point from the Sun (Figure 6).

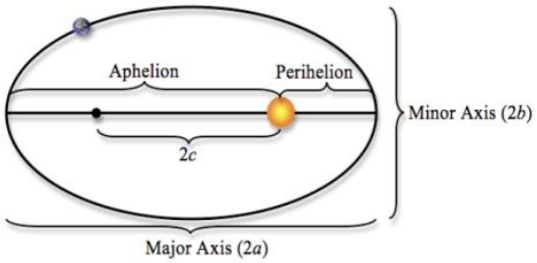



**Figure 6**. The perihelion and the aphelion of a planet. [8]

**B.3 Kepler's Third Law** [4]

The derivation of Kepler's third law is simple enough to carry out for the special case of a circular orbit. Consider a planet of mass $M_\mathrm{P}$ moving around the Sun, which has a mass of $M_\mathrm{S}$, in a circular orbit. Because the orbit is circular, the planet moves at a constant speed $v$. Newton's second law, his law of gravitation, and *centripetal acceleration* $a_c$ then give the following equation:

$$M_\mathrm{P} a_c = \frac{M_\mathrm{P} v^2}{r} = \frac{G M_\mathrm{S} M_\mathrm{P}}{r^2} \tag{40}$$

The speed $v$ of the planet in its orbit is equal to the circumference of the orbit divided by the time required for one revolution, $T$, called the [*sidereal*] *period* of the planet, so $v = 2\pi r/T$. Substituting, the preceding expression becomes

$$\frac{G M_\mathrm{S}}{r^2} = \frac{(2\pi r/T)^2}{r} \tag{41}$$

$$T^2 = \left(\frac{4\pi^2}{G M_\mathrm{S}}\right) r^3 = K_\mathrm{S} r^3 \tag{42}$$

where $K_\mathrm{S}$ is a constant given by

$$K_\mathrm{S} = \frac{4\pi^2}{G M_\mathrm{S}} = 2.97 \times 10^{-19}\ \mathrm{s^2/m^3} \tag{43}$$

Equation (43) is Kepler's third law for a circular orbit. The orbits of most of the planets are very nearly circular. Comets and asteroids, however, usually have elliptical orbits. For these orbits, the radius $r$ must be replaced with $a$, the semimajor axis - half the longest distance across the elliptical orbit. (This is also the average distance of the comet or asteroid from the Sun.) A more detailed calculation shows that $K_\mathrm{S}$ actually depends on the sum of both the mass of a given planet

and the Sun's mass. The masses of the planets, however, are negligible compared with the Sun's mass; hence can be neglected, meaning equation (43) is valid for any planet in the Sun's family. If we consider the orbit of a satellite such as the Moon around Earth, then the constant has a different value, with the mass of the Sun replaced by the mass of Earth. In that case, $K_{\mathrm{E}}$ equals $\frac{4\pi^2}{GM_{\mathrm{E}}}$.

The mass of the Sun can be determined from Kepler's third law because the constant $K_{\mathrm{S}}$ in equation (43) includes the mass of the Sun and the other variables and constants can be [relatively] easily measured. The value of this constant can be found by substituting the values of a planet's period and orbital radius and solving for $K_{\mathrm{S}}$. The mass of the Sun is then

$$M_{\mathrm{S}} = \frac{4\pi^2}{GK_S} \tag{44}$$

Currently, the mass of the Sun is estimated to be $1.98855(25)\times10^{30}$ kg, approximately 330,000 times the mass of Earth. This same process can be used to calculate the mass of Earth (by considering the period and orbital radius of the Moon) and the mass of other planets in the solar system that have satellites.

## 3. Computational project 3.I

### A. Objectives

The objective of Project 3.I is to write a program for simulation of motion of a planet around the Sun, and to perform the simulation for each of the eight planets (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, and Neptune) *under the following simplifications*:

1. Orbits of all planets lie in the same plane.
2. The major axes of all planets coincide, and the minor axes of all planets coincide.
3. Each planet moves under the gravitational influence of the Sun only, i.e. interplanetary gravitational interactions are to be ignored.

Using your simulation(s), determine the following parameters for each planet (Table 1):

a) distance (in km) and speed (in km/s) at the aphelion,
b) the length of the semimajor axis (in $\times10^6$ km),
c) sidereal orbit period (in days),
d) orbit eccentricity,
e) mean orbital velocity (km/s),

In the provided Microsoft Word file, 1) fill in blank/empty cells in Table 1 from your "best" simulation, 2) in the header of Table 1, specify the method (Euler, Cromer, midpoint/RK2) used in the simulation. Upload the completed file to D2L. Also, from each of your simulations, plot a trajectory (two-dimensional, 2D, plot would be sufficient) of a planet (orbit), and then combine all planetary orbits in a single [final] plot.

**Table 1.** Useful Planetary Data.
Complete the table (white cells only) using data from your "best" (most accurate) Project 3.I simulation.

| Planet | Mass ($\times 10^{24}$ kg) | Perihelion | | Aphelion | | Semimajor axis ($\times 10^{6}$ km) | Sidereal orbit period (days) | Orbit eccentricity | Mean orbital velocity (km/s) |
|---|---|---|---|---|---|---|---|---|---|
| | | Distance ($\times 10^{6}$ km) | Speed (km/s) | Distance ($\times 10^{6}$ km) | Speed (km/s) | | | | |
| **Mercury** | 0.3301 | 46.00 | 58.98 | 69.82 | 38.86 | 57.91 | 87.97 | 0.2056 | 47.87 |
| **Venus** | 4.8676 | 107.48 | 35.26 | 108.94 | 34.79 | 108.21 | 224.701 | 0.0067 | 35.02 |
| **Earth** | 5.9726 | 147.09 | 30.29 | 152.10 | 29.29 | 149.60 | 365.256 | 0.0167 | 29.78 |
| **Mars** | 0.64174 | 206.62 | 26.50 | 249.23 | 21.97 | 227.92 | 686.980 | 0.0935 | 24.13 |
| **Jupiter** | 1,898.3 | 740.52 | 13.72 | 816.62 | 12.44 | 778.57 | 4,332.59 | 0.0489 | 13.07 |
| **Saturn** | 568.36 | 1,352.55 | 10.18 | 1,514.50 | 9.09 | 1,433.5 | 10,759.22 | 0.0565 | 9.69 |
| **Uranus** | 86.816 | 2,741.30 | 7.11 | 3,003.62 | 6.49 | 2,872.46 | 30,685.4 | 0.0457 | 6.81 |
| **Neptune** | 102.42 | 4,444.45 | 5.50 | 4,545.67 | 5.37 | 4,495.06 | 60,189. | 0.0113 | 5.43 |

**B. Suggested implementation**

It is recommended to complete the project using equations of motion in either Newton's or the Lagrange form. In the following, I discuss the solution using Newton's equations only because the problem is simple enough to be solved in two-dimensional (2D) Cartesian coordinate system.

As discussed above, Newton's equations of motion along with the initial position and momentum of a particle, give us $\vec{\mathbf{r}}(t)$ which describes the position of the particle as a function of time $t$ [1]. The path described by $\vec{\mathbf{r}}(t)$ is called the *trajectory* of the particle. The trajectory of a particle offers a complete description of the state of the particle. Classical mechanics provides a method for calculating the trajectory of a particle in terms of the forces acting upon the particle through Newton's equations (1), (5), and (11) [1].

Consider the planet orbiting the Sun on an elliptical orbit as shown in Figure 7. Suppose at a given time $t_0$ the planet is located at perihelion ($\vec{\mathbf{r}}_0$), and its velocity ($\vec{\mathbf{v}}_0$) (the speed and the direction of motion), and mass ($M_\mathrm{P}$) are known. If we also know all forces that act on the planet (in this project, we only consider the gravitational attraction between the planet and the Sun), then we should be able to predict the location of the planet ($\vec{\mathbf{r}}_i$), and its velocity ($\vec{\mathbf{v}}_i$) as any point in time $t_i$ (Figure 7).

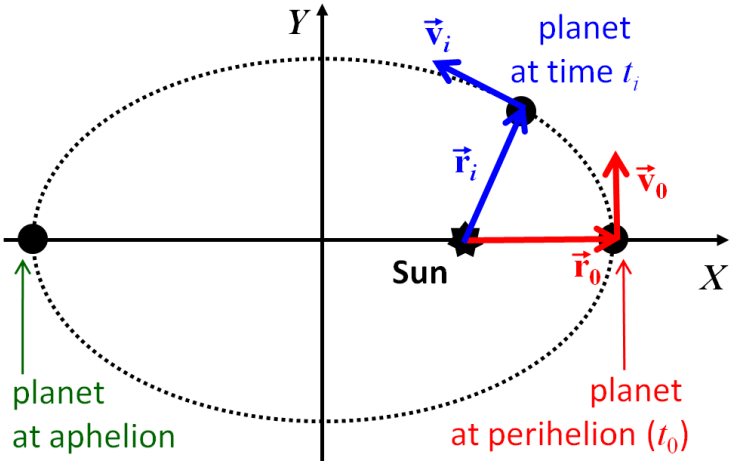



**Figure 7**.

Note that at time $t_i$ the gravitational forces (recall that forces always occur in pairs) are directed along the vector $\vec{\mathbf{r}}_i$ as shown in Figure 8. Because the mass of the Sun is so much larger than the mass of any planet (the Sun accounts for approximately 99.86% of the *total* mass of the Solar System), we can safely assume that the Sun is stationary, and it is the planet that moves in the gravitational field of the Sun. Indeed, according to Newton's third law (Figure 8):

$$\vec{\mathbf{F}}_{\text{Sun}-\text{planet}} = -\vec{\mathbf{F}}_{\text{planet-Sun}} \tag{45}$$

Using the second law we get

$$\vec{\mathbf{F}}_{\text{Sun}-\text{planet}} = M_\mathrm{P}\vec{\mathbf{a}}_P \tag{46}$$

$$\vec{\mathbf{F}}_{\text{planet-Sun}} = M_\mathrm{S}\vec{\mathbf{a}}_\mathrm{S} \tag{47}$$

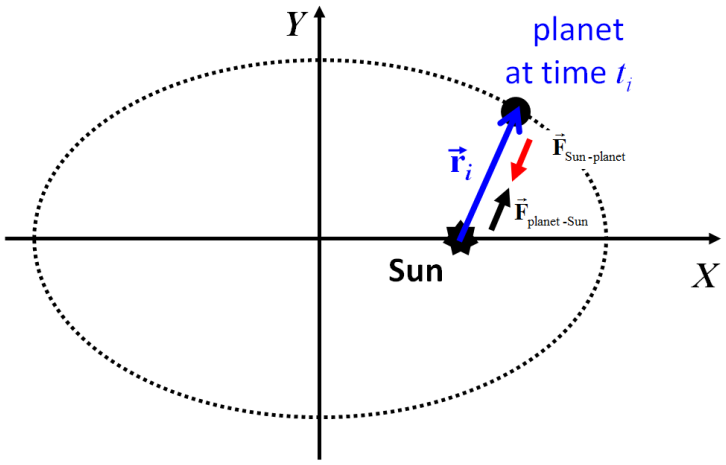



**Figure 8**.

Combining the two laws gives

$$M_{\mathrm{P}}\vec{\mathbf{a}}_{\mathrm{P}} = -M_{\mathrm{S}}\vec{\mathbf{a}}_{\mathrm{S}} \tag{48}$$

$$\frac{M_{\mathrm{P}}}{M_{\mathrm{S}}} = -\frac{\vec{\mathbf{a}}_{\mathrm{S}}}{\vec{\mathbf{a}}_{\mathrm{P}}} \tag{49}$$

Note that because the ratio $M_{\mathrm{P}}/M_{\mathrm{S}}$ is always positive, the two accelerations should point in the opposite directions. Also, because $M_{\mathrm{S}} \gg M_{\mathrm{P}}$, it is obvious that $\vec{\mathbf{a}}_{\mathrm{P}} \gg \vec{\mathbf{a}}_{\mathrm{S}}$, i.e. it is the planet that experiences significant acceleration, not the Sun.

All the necessary data for the planets (their masses, and distance to the Sun, $\vec{\mathbf{r}}_0$, and the velocity, $\vec{\mathbf{v}}_0$, at the perihelion) are given in Table 1. The mass of the Sun is given the discussion above. Thus, we have all the information we need to solve Newton's equations of motion for a given [sufficiently] small time step $\Delta t$, advance the planet from $\vec{\mathbf{r}}_0$ to $\vec{\mathbf{r}}_1$, and calculate the new velocity $\vec{\mathbf{v}}_1$. Because *in this project* we assume that all planets are orbiting around the Sun in the same plane, we do *not* need to consider the third dimension (*Z*), and limit Newton's equations to two dimensions (*X* and *Y*) only. First, let's calculate the planet's new location $\vec{\mathbf{r}}_1$. From the definition of velocity, we have

$$\vec{\mathbf{v}}_0 = \frac{d\vec{\mathbf{r}}}{dt} \approx \frac{\Delta\vec{\mathbf{r}}}{\Delta t} = \frac{\vec{\mathbf{r}}_1 - \vec{\mathbf{r}}_0}{\Delta t} \tag{50}$$

where $\vec{\mathbf{v}}_0$ is the velocity at $\vec{\mathbf{r}}_0$. **Note that the interval $\Delta t$ needs to be sufficiently small for the approximation to work**. Solving this equation for $\vec{\mathbf{r}}_1$ gives

$$\vec{\mathbf{r}}_1 = \vec{\mathbf{r}}_0 + \vec{\mathbf{v}}_0\Delta t \tag{51}$$

Now, all we need to do is calculate the [new] velocity $\vec{\mathbf{v}}_1$ at $\vec{\mathbf{r}}_1$. Combining the definition of the acceleration and Newton's second law we get

$$\vec{\mathbf{F}}_0 = m\vec{\mathbf{a}}_0 = m\frac{d\vec{\mathbf{v}}}{dt} \approx m\frac{\Delta\vec{\mathbf{v}}}{\Delta t} = m\frac{\vec{\mathbf{v}}_1 - \vec{\mathbf{v}}_0}{\Delta t} \tag{52}$$

where $\vec{\mathbf{F}}_0$ is the force exerted on an object with a mass $m$ at $\vec{\mathbf{r}}_0$. Solving this equation for $\vec{\mathbf{v}}_1$ gives

$$\vec{\mathbf{v}}_1 = \vec{\mathbf{v}}_0 + \vec{\mathbf{a}}_0\Delta t = \vec{\mathbf{v}}_0 + m^{-1}\vec{\mathbf{F}}_0\Delta t \tag{53}$$

where $\vec{\mathbf{a}}_0 = m^{-1}\vec{\mathbf{F}}_0$. In our case, $\vec{\mathbf{F}}_0$ are the components of the gravitational force (pull) of the Sun experienced by a planet located at $\vec{\mathbf{r}}_0$. In Figure 8, that force is labeled as $\vec{\mathbf{F}}_{\mathrm{Sun-planet}}$.

According to Newton's law of universal gravitation, the force exerted by the Sun (with a mass of $M_S$) on to the planet with mass $M_P$ located at $\vec{\mathbf{r}}_i$ is given by:

$$\vec{\mathbf{F}}_i = -\frac{GM_S M_P}{|\vec{\mathbf{r}}_i|^2}\frac{\vec{\mathbf{r}}_i}{|\vec{\mathbf{r}}_i|} \tag{54}$$

where $\vec{\mathbf{r}}_i/|\vec{\mathbf{r}}_i|$ is a unit vector pointing towards the planet, and the negative sign indicates that the force acting on the planet is in the direction opposite to $\vec{\mathbf{r}}_i$. Using a simplified notation for the norm of the position vector, $r = |\vec{\mathbf{r}}_i|$, equation (54) can be re-written as:

$$\vec{\mathbf{F}}_i = -\frac{GM_S M_P}{r^3}\vec{\mathbf{r}}_i \tag{55}$$

When working in two dimensions in the coordinate system $X'$,$Y'$ with the origin at the location of the Sun (Figure 9), the components of the gravitational force $\vec{\mathbf{F}}_i$ acting on a planet located at $\vec{\mathbf{r}}_i$ [here, $\vec{\mathbf{F}}_i \equiv (F_{x,i}, F_{y,i})$] can be readily identified.

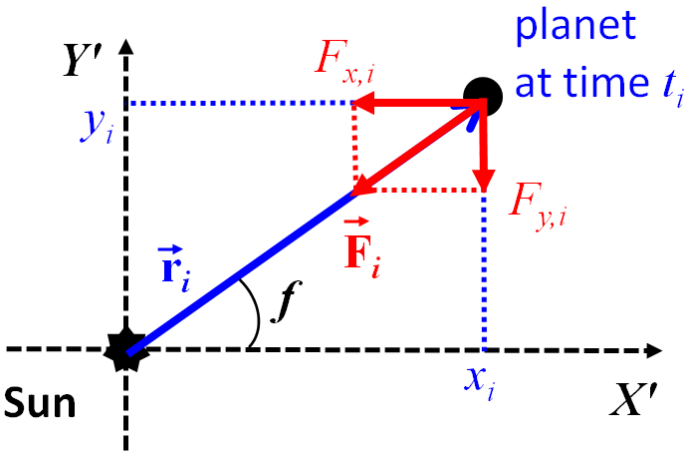



**Figure 9.**

The quantity $f$ in Figure 9 is the *true anomaly*. *Note that the direction of the gravitational force acting on the planet is opposite to that of the position vector*.

Equations (51) and (53) can be generalized for any pair of steps $i$ and $i+1$ as follows:

$$\vec{\mathbf{r}}_{i+1} = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_i \Delta t \tag{56}$$

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i + \vec{\mathbf{a}}_i \Delta t \tag{57}$$

Using the relationship between the acceleration and the force, $\vec{\mathbf{a}}_i = {M_P}^{-1}\vec{\mathbf{F}}_i$, where $M_P$ is the mass of the planet, we can re-write equation (57) in terms of the gravitational force acting on the planet:

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i + M_P^{-1}\left(-\frac{GM_S M_P}{r_i^3}\vec{\mathbf{r}}_i\right)\Delta t \tag{58}$$

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i - \frac{GM_S}{r_i^3}\vec{\mathbf{r}}_i \Delta t \tag{59}$$

### C. Alternative derivation of equations (56) and (57) and the better approximations

Equations (59) and (57) can be derived in a more rigorous ("mathematical") way. We start from Newton's equation of motion for a planet with mass $M_P$ at *i*-th point of its trajectory

$$\vec{\mathbf{F}}_i = M_P \vec{\mathbf{a}}_i \tag{60}$$

$$\vec{\mathbf{a}}_i = {M_P}^{-1} \vec{\mathbf{F}}_i \tag{61}$$

Using the relationship between the acceleration $\vec{\mathbf{a}}_i$, velocity $\vec{\mathbf{v}}_i$, and position $\vec{\mathbf{r}}_i$, we get:

$$\vec{\mathbf{a}}_i = \frac{d\vec{\mathbf{v}}_i}{dt} = \frac{d}{dt}\left(\frac{d\vec{\mathbf{r}}_i}{dt}\right) = {M_P}^{-1} \vec{\mathbf{F}}_i \tag{62}$$

If the force is given by equation (55):

$$\vec{\mathbf{F}}_i = -\frac{GM_S M_P}{r^3} \vec{\mathbf{r}}_i \tag{63}$$

the *second order* differential equation (62) can be re-written in terms of **two coupled** *first-order equations*:

$$\begin{aligned} \frac{d\vec{\mathbf{r}}_i}{dt} &= \vec{\mathbf{v}}_i \\ \frac{d\vec{\mathbf{v}}_i}{dt} &= \vec{\mathbf{a}}_i = -\frac{GM_S}{r^3} \vec{\mathbf{r}}_i \end{aligned} \tag{64}$$

Numerical integration of ordinary differential equations (ODEs) is well documented [9]. Consider the following ODE:

$$\frac{dy(x)}{dx} = f(x, y) \tag{65}$$

where the function on the right-hand side is known [9]. According to Euler's method, which advances a solution from $x_i$ to $x_{i+1} \equiv x_i + h$ [9],

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{66}$$

The formula is unsymmetrical: it advances the solution through an interval *h*, but uses derivative information only at the beginning of that interval (see Figure 10) [9]. That means (and you can verify by expansion in power series) that the step's error is only one power of *h* smaller than the correction, i.e $O(h^2)$ added to (66) [9].

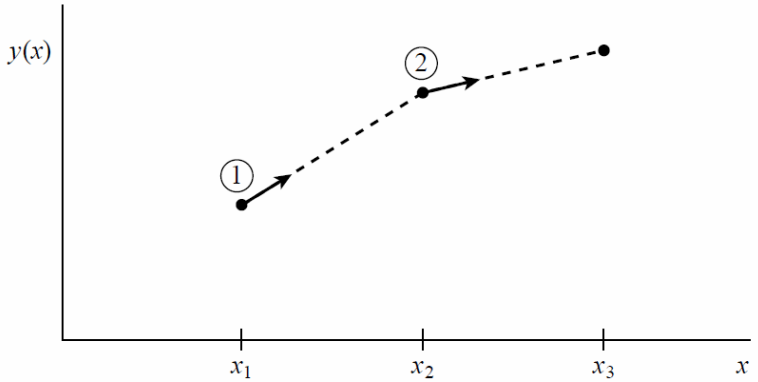



**Figure 10.** Euler's method [9]. In this simplest (and least accurate) method for integrating an ODE, the derivative at the starting point of each interval is extrapolated to find the next function value [9]. The method has first-order accuracy [9].

When applied to equations (64), the numerical solutions given by the Euler's method

$$\vec{\mathbf{r}}_{i+1} = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_i \Delta t \tag{67}$$

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i + \vec{\mathbf{a}}_i \Delta t \tag{68}$$

are absolutely identical to equations (56) and (57) [I hope you did see it coming!]. The problem is that, in general, Euler's method is not recommended for practical use because the method is (a) not very accurate when compared to other, fancier, methods run at the equivalent stepsize, and (b) not very stable [9].

In 1981, Cromer [10,11] suggested a minor modification of the standard Euler approximation for the solution of oscillatory problems in mechanics which yields solutions that are stable for arbitrarily large number of iterations, regardless of the size of the iteration interval [10]. The modification is to include an updated value of the velocity of the particle to compute the position [11]:

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i + \vec{\mathbf{a}}_i \Delta t \tag{69}$$

$$\vec{\mathbf{r}}_{i+1} = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_{i+1} \Delta t \tag{70}$$

Note that velocity is updated first, and then its *new* value $\vec{\mathbf{v}}_{i+1}$ is used to compute the position. As shown by Cromer [10], the period of a nonlinear oscillator converges rapidly to its exact value as the size of the iteration interval is decreased [10]. In two dimensions, closed orbits are given for the two-body Kepler problem and the restricted three-body problem can be iterated indefinitely to produce space-filling orbits [10].

However, we can do a lot better if we consider the use of a step like (66) to take a "trial" step to the midpoint of the interval [9]. Then use the value of both *x* and *y* at that midpoint to compute the "real" step across the whole interval [9]. Figure 11 illustrates the idea [9]. In equations [9],

$$\begin{aligned} k_1 &= hf(x_i, y_i) \\ k_2 &= hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1\right) \\ y_{i+1} &= y_i + k_2 + O(h^3) \end{aligned} \tag{71}$$

As indicated in the error term, this symmetrization cancels out the first-order error term, making the method *second order* [9]. A method is conventionally called *n*th order if its error term is $O(h^{n+1})$ [9]. In fact, the approach presented in equation (71) is called the *second-order Runge-Kutta* (RK; pronunciation: Roong-eh-Kutta) or *midpoint* method [9].

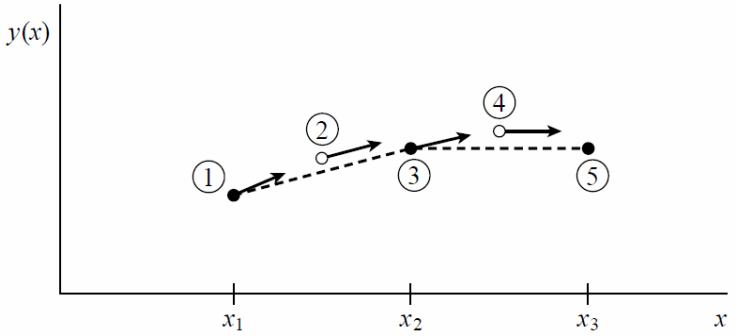



**Figure 11.** Midpoint method [9]. Second-order accuracy is obtained by using the initial derivative at each step to find a point halfway across the interval, then using the midpoint derivative across the full width of the interval[9]. In the figure, filled dots represent final function values, while open dots represent function values that are discarded once their derivatives have been calculated and used [9].

When formulas (71) are applied to our equations of motion, we get the following sequence of steps:

1. Take a "trial" step to the mid-point of the interval

$$\vec{\mathbf{r}}_{i+\frac{1}{2}} = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_i \frac{\Delta t}{2} \tag{72}$$

$$\vec{\mathbf{v}}_{i+\frac{1}{2}} = \vec{\mathbf{v}}_i + \vec{\mathbf{a}}_i \frac{\Delta t}{2} \tag{73}$$

2. Evaluate $\vec{\mathbf{a}}$ at mid-point

$$\vec{\mathbf{a}}_{i+\frac{1}{2}} = -\frac{GM_S}{r^3} \vec{\mathbf{r}}_{i+\frac{1}{2}} \tag{74}$$

3. Take the full step using the properties at mid-point

$$\vec{\mathbf{r}}_{i+1} = \vec{\mathbf{r}}_i + \vec{\mathbf{v}}_{i+\frac{1}{2}} \Delta t \tag{75}$$

$$\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i + \vec{\mathbf{a}}_{i+\frac{1}{2}} \Delta t \tag{76}$$

In one of the projects down the road we will explore higher-order Runge-Kutta methods, but **at this time your code should include the three methods discussed above: 1) Euler's, 2) Cromer's, and 2) the midpoint (second-order RK, RK2).**

**D. Miscellaneous notes**

You are free to choose any *reasonable* step $\Delta t$. For example, you can experiment with $\Delta t$ = 1 sec, 1 min, 10 min, 30 min, 1 hour, 10 hours, 1 day etc. However, due to significant variation of the major axes considered, it may be advantageous to have *the same number of simulation steps for all planets*. Suppose the orbital [sidereal] period of planet $j$ is $T_j$ (Table 1), and the number of simulation steps you wish to use is $N$. Then, the simulation step $\Delta t_j$ for planet $j$ can be [approximately] calculated as

$$\Delta t_j = \frac{T_j}{N} \tag{77}$$

Keep in mind that in this equation the units of $\Delta t_{jj}$ and $T_j$ must be the same. This means that if the orbital period $T_j$ is given in days, then the simulation time step $\Delta t_j$ will also be in days. In my own simulations, I used $N = 10^6$.

When plotting the orbits of the planets, it is recommended (that is, if your software is flexible enough) to use following color scheme:

1. Mercury - green,
2. Venus - pink (the planet is named after the Roman goddess of beauty),
3. Earth - blue (of course!),
4. Mars - red (the planet is named after the Roman god of war),
5. Jupiter - black,
6. Saturn - brown (the planet is named after the Roman god of agriculture),
7. Uranus - purple,
8. Neptune - [dark] cyan (the planet is named after the Roman god of the sea).

It is permissible to define the origin of the plot at the location of the Sun (after all, the name "heliocentric" means that the Sun is at the center!).

It may also be advisable to create *two* final plots of the orbits: one for *terrestrial* planets (also called *inner* planets): Mercury, Venus, Earth, and Mars, and one for *Jovian* planets (also called outer planets): Jupiter, Saturn, Uranus, and Neptune. This is because the orbits of terrestrial planets are much smaller than those of Jovian planets.

## 4. Computational project 3.II

### A. Objectives

The objective of Project 3.II is to write a program for numerical simulation and visualization of the motion the eight major planets around the Sun **without significant simplifications**.

*That is, given locations and velocities of the planets at some point in time (see discussion below), use any of the discussed method (midpoint/RK2 is highly recommended) to simulate the motion of all major planets in the solar system, and visualize their trajectories and orbits for the next 165 years (i.e. the length of the sidereal period of Neptune) including at each simulation step not only the gravitational force of the Sun, but also all the interplanetary gravitational interactions.*

Files with the planetary data for 12:00:00.0000 January 1, 2000 of the so-called "Barycentric Dynamical Time" scale are available for download from the "Project 3 data" section in D2L. Barycentric Dynamical Time (TDB) is a relativistic coordinate time scale, intended for astronomical use as a time standard to take account of time dilation when calculating orbits and astronomical ephemerides of planets, asteroids, comets and interplanetary spacecraft in the Solar System [12]. TDB is now (since 2006) defined as a linear scaling of Barycentric Coordinate Time (TCB) [12]. Barycentric Coordinate is a coordinate time standard intended to be used as the independent variable of time for all calculations pertaining to orbits of planets, asteroids, comets, and interplanetary spacecraft in the Solar system [13]. It is equivalent to the proper time experienced by a clock at rest in a coordinate frame co-moving with the barycenter of the Solar system: that is, a clock that performs exactly the same movements as the Solar system but is outside the system's gravity well [13]. It is therefore not influenced by the gravitational time dilation caused by the Sun and the rest of the system [13].

A sample file for Mercury (`Mercury.txt`) downloaded from the NASA Jet Propulsion Laboratory HORIZONS on-line solar system data and ephemeris computation service [14] is shown below.

```
*******************************************************************************
 Revised: Jul 31, 2013              Mercury                              199 / 1

 GEOPHYSICAL DATA (updated 2008-Feb-07):
  Mean radius (km)       =  2440(+-1)     Density (g cm^-3)     =  5.427
  Mass (10^23 kg )       =     3.302      Flattening, f         =
  Volume (x10^10 km^3)   =     6.085      Semi-major axis       =
  Sidereal rot. period   =    58.6462 d   Rot. Rate (x10^5 s)   =  0.124001
  Mean solar day         =   175.9421 d   Polar gravity ms^-2   =
  Mom. of Inertia        =     0.33       Equ. gravity  ms^-2   =  3.701
  Core radius (km)       = ~1600          Potential Love # k2   =

  GM (km^3 s^-2)         = 22032.09       Equatorial Radius, Re =    2440 km
  GM 1-sigma (km^3 s^-2)=   +-0.91        Mass ratio (sun/plnt) = 6023600

  Atmos. pressure (bar) =                 Max. angular diam.    = 11.0"
  Mean Temperature (K)  =                 Visual mag. V(1,0)    = -0.42
  Geometric albedo      =   0.106         Obliquity to orbit[1] = 2.11' +/- 0.1'
  Sidereal orb. per.    =   0.2408467 y   Mean Orbit vel.  km/s = 47.362
  Sidereal orb. per.    =  87.969257  d   Escape vel. km/s      =  4.435
  Hill's sphere rad. Rp =  94.4           Planetary Solar Const = 9936.9 (Wm^2)

[1] Margot et al., Science 316, 2007
*******************************************************************************


*******************************************************************************
Ephemeris / WWW_USER Wed Apr 16 18:35:00 2014 Pasadena, USA      / Horizons
*******************************************************************************
Target body name: Mercury (199)                   {source: DE-0431LE-0431}
Center body name: Sun (10)                        {source: DE-0431LE-0431}
Center-site name: BODY CENTER
*******************************************************************************
Start time      : A.D. 2000-Jan-01 12:00:00.0000 CT
Stop  time      : A.D. 2000-Jan-01 12:00:01.0000 CT
Step-size       : 1 minutes
*******************************************************************************
Center geodetic : 0.00000000,0.00000000,0.0000000 {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.00000000,0.00000000,0.0000000 {E-lon(deg),Dxy(km),Dz(km)}
Center radii    : 696000.0 x 696000.0 x 696000.0 k{Equator, meridian, pole}
Output units    : KM-S
Output format   : 03
Reference frame : ICRF/J2000.0
Output type     : GEOMETRIC cartesian states
Coordinate systm: Ecliptic and Mean Equinox of Reference Epoch
*******************************************************************************
JDCT
   X     Y     Z
   VX    VY    VZ
   LT    RG    RR
*******************************************************************************
$$SOE
2451545.000000000 = A.D. 2000-Jan-01 12:00:00.0000 (CT)
 X =-1.946172639275932E+07 Y =-6.691327522588462E+07 Z =-3.679854414596553E+06
 VX= 3.699499188030234E+01 VY=-1.116441595690670E+01 VZ=-4.307627980092748E+00
 LT= 2.327714970537363E+02 RG= 6.978313925407939E+07 RR= 6.149432668355459E-01
$$EOE
*******************************************************************************
Coordinate system description:

  Ecliptic and Mean Equinox of Reference Epoch

    Reference epoch: J2000.0
    xy-plane: plane of the Earth's orbit at the reference epoch
    x-axis  : out along ascending node of instantaneous plane of the Earth's
              orbit and the Earth's mean equator at the reference epoch
    z-axis  : perpendicular to the xy-plane in the directional (+ or -) sense
              of Earth's north pole at the reference epoch.

Symbol meaning

    JDCT     Epoch Julian Date, Coordinate Time
      X      x-component of position vector (km)
      Y      y-component of position vector (km)
      Z      z-component of position vector (km)
      VX     x-component of velocity vector (km/sec)
      VY     y-component of velocity vector (km/sec)
      VZ     z-component of velocity vector (km/sec)
      LT     One-way down-leg Newtonian light-time (sec)
      RG     Range; distance from coordinate center (km)
      RR     Range-rate; radial velocity wrt coord. center (km/sec)

Geometric states/elements have no aberration corrections applied.

 Computations by ...
     Solar System Dynamics Group, Horizons On-Line Ephemeris System
     4800 Oak Grove Drive, Jet Propulsion Laboratory
```

```
      Pasadena, CA  91109   USA
      Information: http://ssd.jpl.nasa.gov/
      Connect    : telnet://ssd.jpl.nasa.gov:6775  (via browser)
                   telnet ssd.jpl.nasa.gov 6775    (via command-line)
      Author     : Jon.Giorgini@jpl.nasa.gov
*******************************************************************************
```

From each file you will need to read **1)** planet's mass (highlighted in the example above in **red**) , **2)** the three components of the position vector $\vec{\boldsymbol{r}}_0$, X, Y, and Z (highlighted in **blue)**, and **3)** the three components of the velocity vector $\vec{\boldsymbol{v}}_0$, VX, VY, and VZ (highlighted in **green**). The units of X, Y, Z are kilometers (km), and the units of VX, VY, and VZ are km/sec (see text highlighted in **black**).

## B. Visualization

"*The purpose of visualization is insight, not pictures*" [15]

The computer visualization is defined as "the use of computer-supported, interactive, visual representations of data to amplify cognition" [15].

When using computer algebra software (Mathematica, Maple, or MATLAB), visualization of a simulation in three dimensions (3D) is easy, and does not need any additional description. In contrast, when using a programming language, such as Fortran, C, C++ etc., visualization may be challenging, and most of the time requires interface to a third-party visualization software.

In this section, we will describe in detail how to use the so-called Tripos Mol2 file format [16], **which is originally intended to represent molecules**, to produce high-quality visualization of the motion of planets and their orbits in the molecular visualization system PyMOL created by Dr. Warren Lyford DeLano (1972-2009) and now developed by Schrödinger, Inc. [17,18], or any other software that can read "molecular" trajectories from a Tripos Mol2 file. Note that PyMOL and some other similar visualization tools are capable of real-time rendering in stereoscopic 3D using the NVIDIA Quadro quad buffered professional stereo [19].

### B.1. Plotting locations of planets and their trajectories

In the example below, we use the Tripos Mol2 file format to plot locations of the planets using Cartesian coordinates (X, Y, Z) read directly from the planetary data files described above. Note that Mol2 files are written out in a free format to avoid the restrictions created by fixed format files [16]. The extension of the Tripos Mol2 file should always be ".mol2" (".mol" *may* also work).

```
@<TRIPOS>MOLECULE
nPRT=1,step=0,time=0.0sec=0.00min=0.0000hrs=0.0000days=0.000000years
       9       0       1
SMALL
NO_CHARGES

@<TRIPOS>ATOM
```

```
0     Sun         0.0000E+00  0.0000E+00  0.0000E+00   S      1  RES1     0.000
1     Mercury    -1.9462E+01 -6.6913E+01 -3.6799E+00   H      1  RES1     0.000
2     Venus      -1.0746E+02 -4.8850E+00  6.1356E+00   Mn     1  RES1     0.000
3     Earth      -2.6499E+01  1.4470E+02 -6.1122E-04   Fe     1  RES1     0.000
4     Mars        2.0805E+02 -2.0071E+00 -5.1563E+00   O      1  RES1     0.000
5     Jupiter     5.9857E+02  4.3960E+02 -1.5227E+01   K      1  RES1     0.000
6     Saturn      9.5839E+02  9.8286E+02 -5.5213E+01   Zn     1  RES1     0.000
7     Uranus      2.1590E+03 -2.0546E+03 -3.5625E+01   N      1  RES1     0.000
8     Neptune     2.5150E+03 -3.7387E+03  1.9032E+01   N      1  RES1     0.000
```

Let's go over each of the entries.

**@<TRIPOS>MOLECULE**

Each entry in the Tripos Mol2 file starts with the "@<TRIPOS>MOLECULE" keyword. Of course, our "molecule" is the Solar System!

**nPRT=1,step=0,time=0.0sec=0.00min=0.0000hrs=0.0000days=0.000000years**

This is just a comment. You can put anything you want here.

```
      9       0       1
```

This numbers say that there are 9 objects that need to be plotted (in our case, the Sun and eight planets), there will be 0 connections between objects, and that there is 1 substructure (in our case, it will always be "1").

**SMALL**

This is the "molecule" type. In general, there are several options available: SMALL, BIOPOLYMER, PROTEIN, NUCLEIC_ACID, SACCHARIDE. Since we are not working with any of those, we can choose any option, but perhaps the safest choice is SMALL.

**NO_CHARGES**

When working with molecules, this option allows to assign electric charges for all atoms. Again, we do not care about it right now, and should always set it to NO_CHARGES.

**@<TRIPOS>ATOM**

This is where the section with our "atoms" (planets) starts.

```
0     Sun         0.0000E+00  0.0000E+00  0.0000E+00   S      1  RES1     0.000
1     Mercury    -1.9462E+01 -6.6913E+01 -3.6799E+00   H      1  RES1     0.000
2     Venus      -1.0746E+02 -4.8850E+00  6.1356E+00   Mn     1  RES1     0.000
3     Earth      -2.6499E+01  1.4470E+02 -6.1122E-04   Fe     1  RES1     0.000
4     Mars        2.0805E+02 -2.0071E+00 -5.1563E+00   O      1  RES1     0.000
5     Jupiter     5.9857E+02  4.3960E+02 -1.5227E+01   K      1  RES1     0.000
6     Saturn      9.5839E+02  9.8286E+02 -5.5213E+01   Zn     1  RES1     0.000
7     Uranus      2.1590E+03 -2.0546E+03 -3.5625E+01   N      1  RES1     0.000
8     Neptune     2.5150E+03 -3.7387E+03  1.9032E+01   N      1  RES1     0.000
```

Each data record associated with a single "atom" (planet) consists of a single data line. The data line contains all the information necessary to reconstruct one "atom" (planet) contained within the "molecule" (Solar System). The format of each line is as follows:

**atom_id atom_name x y z atom_type subst_id subst_name charge**

- **atom_id (integer)** = the sequence number of the "atom" (planet) at the time the file was created. In my example, Sun is number 0, and the planets are numbered starting from Mercury.
- **atom_name (string)** = the name of the "atom" (planet). How convenient!
- **x (real)** = the x coordinate of the "atom" (planet). In my example, the unit is $10^9$ m.
- **y (real)** = the y coordinate of the "atom" (planet). In my example, the unit is $10^9$ m.
- **z (real)** = the z coordinate of the "atom" (planet). In my example, the unit is $10^9$ m.
- **atom_type (string)** = the atom type for the "atom" (planet). I have tried to match the properties of "atom types" in PyMOL to the Sun and planets. For example, Sun will be represented by a yellow sphere (sulfur atom, S). Earth is blue (iron atom, Fe). Mars is red (oxygen atom, O) etc.
- **subst_id (integer)** = the ID number of the substructure containing the "atom" (planet). I always use "1" here.
- **subst_name (string)** = the name of the substructure containing the "atom" (planet). I use a generic "residue" name (RES1).
- **charge (real)** = the charge associated with the "atom" (planet). As discussed above, all charges should really be zero.

As the simulation is running, the code writes out the locations of the planets (and also the Sun) to a Tripos Mol2 file at a certain time step which is usually longer than the simulation step. This is because we try not to print out more than, say, 100,000 geometries since the visualization software may have problems handling too many of those (what if the simulation includes $10^{12}$ steps?).

When a Mol2 file with multiple geometries (i.e. "`@<TRIPOS>MOLECULE`" entries) is loaded into PyMOL, the program interprets the sequence of the "`@<TRIPOS>MOLECULE`" entries as a *Trajectory* if all entries have the same number of "atoms" and connections between them ("bonds"). As such, when loading multiple locations of planets in the Solar System (Sun, of course, is always considered to be at the center of it), PyMOL will be able to visualize the trajectories of planets.

### B.2. Plotting outlines of the planetary orbits

In addition to the motion of the planets, we need to outline the orbit of planets. A planetary orbit can be considered simply as a line through a sequence of points along the trajectory. These can be also plotted using the Tripos Mol2 file Format. In the following example, we show how to do it for one of the planets (Mercury). Unlike planetary trajectories which were written to a single file, each planetary orbit should probably be written to its own file (for example, "Mercury_orbit.mol2"). As you can probably guess, there will be a single "`@<TRIPOS>MOLECULE`" entry that includes all selected points along a trajectory (a good idea is to make orbit from a single complete revolution of a planet around the Sun).

In the "@<TRIPOS>MOLECULE" section, we write out 2112 equally-spaced [in time] points along a single revolution of Mercury around the Sun. Note that unlike the previous example, it says that in addition to 2112 points (i.e., points that outline Mercury's orbit), there will also be 2112 connections between those points. To save space, the example below lists the first and the last ten locations of Mercury along its complete revolution around the Sun.

```
@<TRIPOS>MOLECULE
ORBIT OF Mercury
               2112               2112                  1
SMALL
NO_CHARGES

@<TRIPOS>ATOM
                 1     Mercury    -1.9462E+01 -6.6913E+01 -3.6799E+00   H      1  RES1     0.000
                 2     Mercury    -1.9328E+01 -6.6953E+01 -3.6954E+00   H      1  RES1     0.000
                 3     Mercury    -1.9195E+01 -6.6993E+01 -3.7108E+00   H      1  RES1     0.000
                 4     Mercury    -1.9062E+01 -6.7032E+01 -3.7263E+00   H      1  RES1     0.000
                 5     Mercury    -1.8928E+01 -6.7071E+01 -3.7417E+00   H      1  RES1     0.000
                 6     Mercury    -1.8795E+01 -6.7110E+01 -3.7572E+00   H      1  RES1     0.000
                 7     Mercury    -1.8661E+01 -6.7148E+01 -3.7726E+00   H      1  RES1     0.000
                 8     Mercury    -1.8527E+01 -6.7186E+01 -3.7879E+00   H      1  RES1     0.000
                 9     Mercury    -1.8393E+01 -6.7224E+01 -3.8033E+00   H      1  RES1     0.000
                10     Mercury    -1.8259E+01 -6.7261E+01 -3.8187E+00   H      1  RES1     0.000
. . . . . .
              2103     Mercury    -2.0701E+01 -6.6522E+01 -3.5341E+00   H      1  RES1     0.000
              2104     Mercury    -2.0569E+01 -6.6565E+01 -3.5498E+00   H      1  RES1     0.000
              2105     Mercury    -2.0437E+01 -6.6608E+01 -3.5654E+00   H      1  RES1     0.000
              2106     Mercury    -2.0304E+01 -6.6650E+01 -3.5811E+00   H      1  RES1     0.000
              2107     Mercury    -2.0172E+01 -6.6693E+01 -3.5967E+00   H      1  RES1     0.000
              2108     Mercury    -2.0039E+01 -6.6734E+01 -3.6123E+00   H      1  RES1     0.000
              2109     Mercury    -1.9906E+01 -6.6776E+01 -3.6279E+00   H      1  RES1     0.000
              2110     Mercury    -1.9773E+01 -6.6817E+01 -3.6434E+00   H      1  RES1     0.000
              2111     Mercury    -1.9640E+01 -6.6858E+01 -3.6590E+00   H      1  RES1     0.000
              2112     Mercury    -1.9507E+01 -6.6898E+01 -3.6745E+00   H      1  RES1     0.000
@<TRIPOS>BOND
                 1                  1                  2                 1
                 2                  2                  3                 1
                 3                  3                  4                 1
                 4                  4                  5                 1
                 5                  5                  6                 1
                 6                  6                  7                 1
                 7                  7                  8                 1
                 8                  8                  9                 1
                 9                  9                 10                 1
                10                 10                 11                 1
. . . . . .
              2103               2103               2104                 1
              2104               2104               2105                 1
              2105               2105               2106                 1
              2106               2106               2107                 1
              2107               2107               2108                 1
              2108               2108               2109                 1
              2109               2109               2110                 1
              2110               2110               2111                 1
              2111               2111               2112                 1
              2112               2112                  1                 1
```

The "@<TRIPOS>MOLECULE" section is followed by the "@<TRIPOS>BOND" section with 2112 entries, i.e. connections between points that outline Mercury's orbit. Each record in the "@<TRIPOS>BOND" section consists of a single line in the following format:

**bond_id origin_atom_id target_atom_id bond_type**

- **bond_id (integer)** = the ID number of the bond at the time the file was created.
- **origin_atom_id (integer)** = the ID number of the atom at one end of the bond.
- **target_atom_id (integer)** = the ID number of the atom at the other end of the bond.
- **bond_type (string)** = the bond type as shown below:

  1 = single (single line connecting the two objects will be drawn)
  2 = double (double line connecting the two objects will be drawn)
  3 = triple (triple line connecting the two objects will be drawn)
  am = amide
  ar = aromatic
  du = dummy
  un = unknown (cannot be determined from the parameter tables)
  nc = not connected

  For our purposes, a "single" bond type works the best.

Thus, the following entry in the "@<TRIPOS>MOLECULE" section

```
2105              2105              2106              1
```

says that "bond" (connection) number 2105 will be between "atoms" (points) 2105 and 2106, and that a single line should connect the two "atoms" (points). The last entry in the section

```
2112              2112               1                1
```

, which connects the last point and the first points along the trajectory and "completes" the orbit outline.

### B.3. Creating and customizing a visualization in PyMOL

Finally, we combine Mol2 files with the planetary trajectories and outlines of their orbits. The easiest way to do so is via a PyMOL script file that we shall call “solarsystem.pml”. **Warning!** The following instructions have been tested with PyMOL versions 1.8 and 1.9, and may not work with the latest PyMOL 2.x distribution from Schrödinger, Inc.

First, we set the background color to black:

```
# set the color of the background
bg_color black
```

Then, we load file with the planetary trajectories (see section B.1) and explicitly instruct PyMOL to draw planets using “spheres”:

```
# load planetary trajectories
load solarsystem_3.mol2, main

hide lines
hide sticks
show spheres
```

Note how we assigned the name "main" to the PyMOL object that contains the planetary trajectories. When using PyMOL under Windows, it may be necessary to specify the full path to the Mol2 file; for example:

```
load D:\MTSU\COMS_7100\Project_3\solarsystem_3.mol2, main
```

We now adjust the size of each sphere (planet) to make the visualization more aesthetically pleasing:

```
alter elem S,  vdw=20
alter elem H,  vdw=10
alter elem Mn, vdw=10
alter elem Fe, vdw=10
alter elem O,  vdw=10
alter elem K,  vdw=50
alter elem Zn, vdw=30
alter elem N,  vdw=30
```

Then, we load files that contain outlines of the planetary orbits (see section B.2):

```
# load outlines of planetary orbits
load  Mercury_orb.mol2, mercury
load  Venus_orb.mol2, venus
load  Earth_orb.mol2, earth
load  Mars_orb.mol2, mars
load  Jupiter_orb.mol2, jupiter
load  Saturn_orb.mol2, saturn
load  Uranus_orb.mol2, uranus
load  Neptune_orb.mol2, neptune
```

Note how we assigned the name "mercury" to the PyMOL object that contains the outline of the Mercury orbit, etc. When using PyMOL under Windows, it may be necessary to specify the full path to each Mol2 file; for example:

```
load  D:\MTSU\COMS_7100\Project_3\Mercury_orb.mol2, mercury
load  D:\MTSU\COMS_7100\Project_3\Venus_orb.mol2, venus
. . . . . .
load  D:\MTSU\COMS_7100\Project_3\Neptune_orb.mol2, neptune
```

We now instruct PyMOL to use the "line" representation for the outlines of the planetary orbits:

```
hide spheres, mercury
hide spheres, venus
hide spheres, earth
hide spheres, mars
hide spheres, jupiter
hide spheres, saturn
hide spheres, uranus
hide spheres, neptune

show lines, mercury
show lines, venus
show lines, earth
show lines, mars
show lines, jupiter
show lines, saturn
show lines, uranus
show lines, neptune
```

The following instructions modify the color of both the sphere and orbit outline for each planet:

```
color lightpink, elem Mn
```

```
color skyblue, elem Fe
color tv_red, elem O
color paleyellow, elem K
color violet, elem Zn
color cyan, elem N
```

Finally, we instruct PyMOL to refresh/rebuild the visualization and set the desired speed of the visualized trajectories in terms of the number of frames per second:

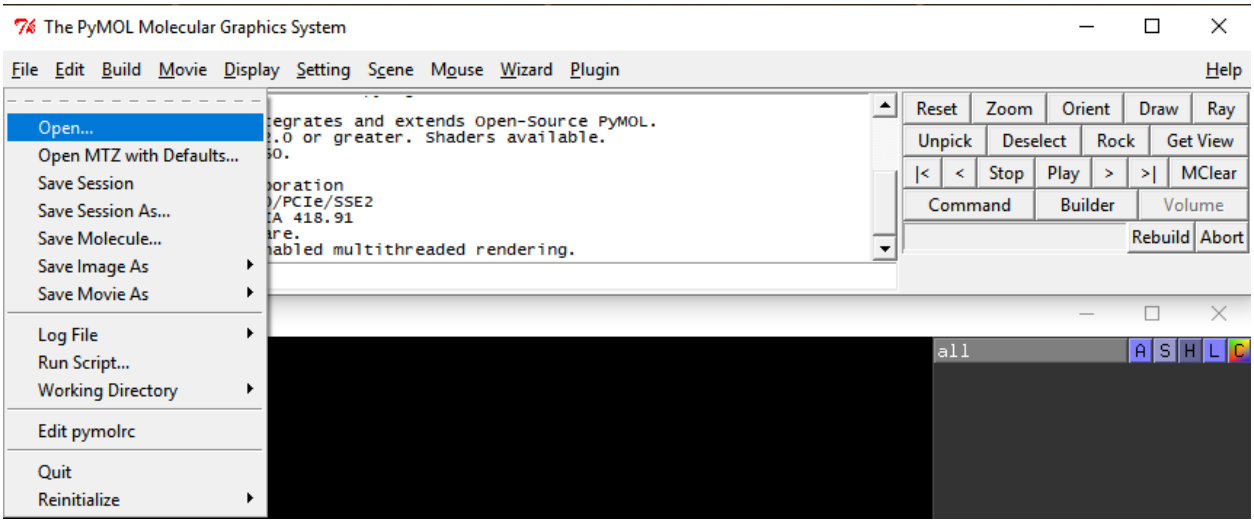
```
rebuild
set movie_fps, 120
```

Once we have veirfied that all the input files (*.mol2 and solarsystem.pml) are in the same directory, we load the "solarsystem.pml" script into PyMOL via the following command line instructions [20,21]:

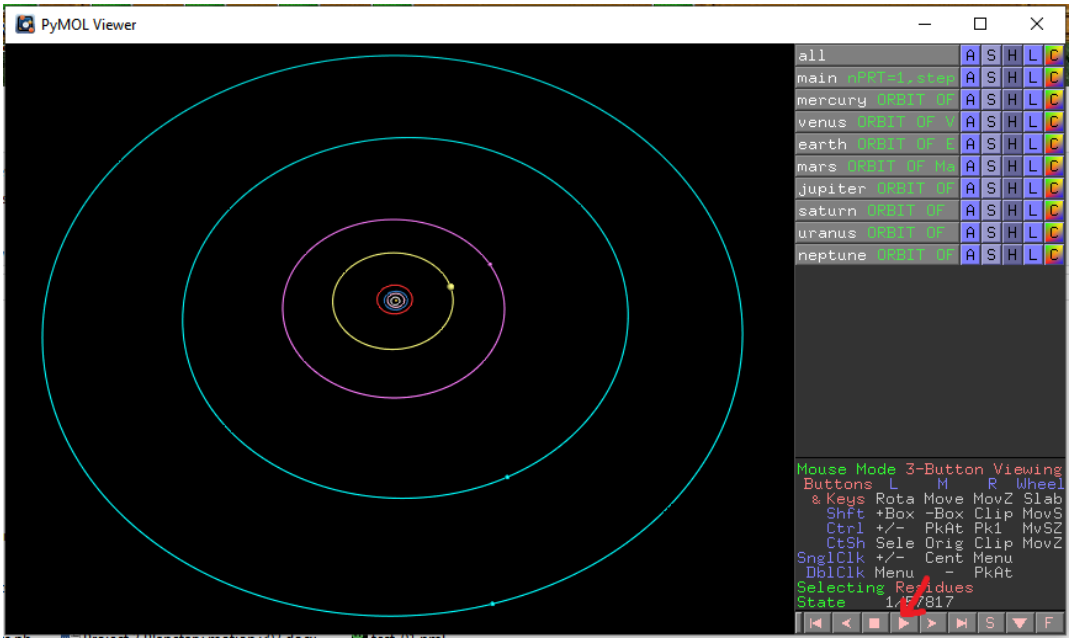
>> pymol solarsystem.pml

Note that depending on the Linux distribution, you may have to explicitly specify whether the PyMOL executable is based on Python2 or Python3 (for example, pymol-2.7 or pymol-3.6, respectively).

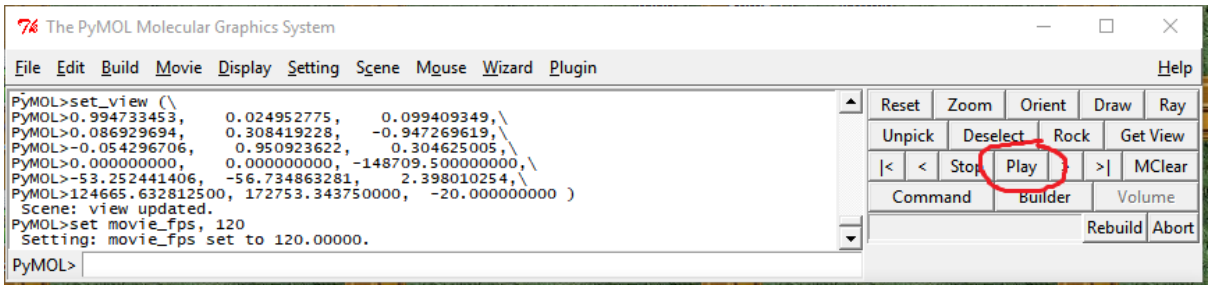
When using PyMOL under Windows, use the "File" → "Open" menu in the so-called External (Tcl/Tk) GUI Window to locate and load the "solarsystem.pml" script (what a nuisance.. sigh..):



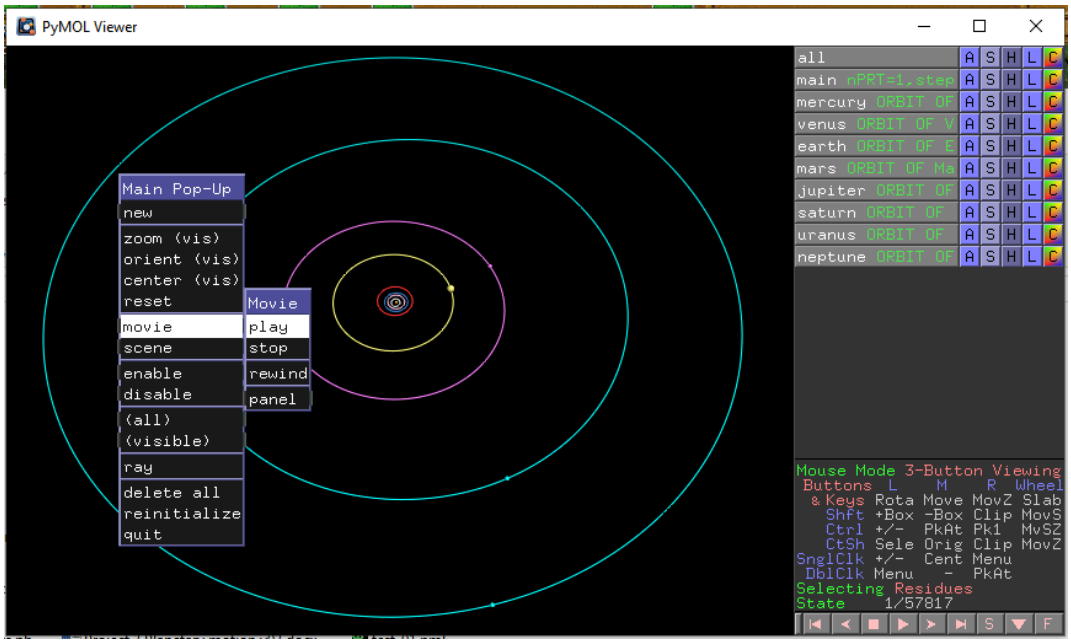

Depending on the speed of your computer and the operating system it may take some time for PyMOL to read in all the files, after which the solar system visualization appears in the PyMOL Viewer window. Bring the mouse cursor within the PyMOL Viewer window. Click and hold the **right** mouse button while dragging the mouse down/up to zoom in/out, and release when you are satisfied with the size of the solar system object. Now, click and hold the **left** mouse button while moving the mouse around to rotate the solar system object. Finally, start animation by clicking on the ► button in the so-called internal GUI window



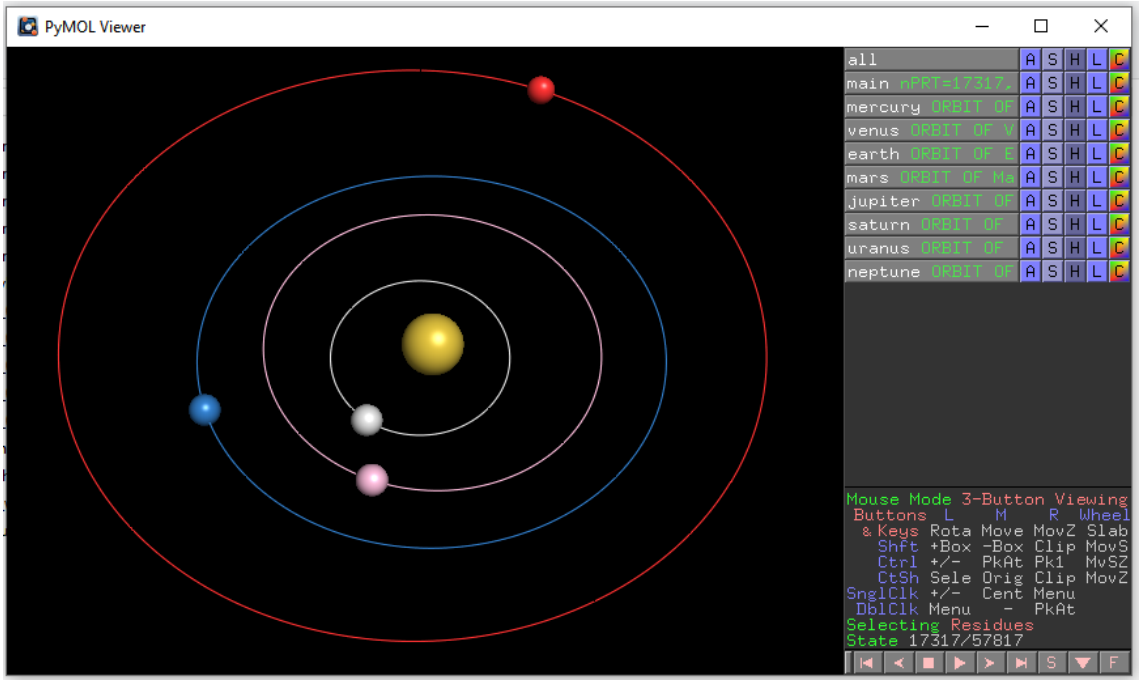

, or using the “Play” button in the External (Tcl/Tk) GUI Window,




, or right-click anywhere in the object windows, select “movie” and then “play”



Once the planets are moving along their trajectories (and the "State" counter in the internal GUI starts counting the frames/time steps), you can zoom in/out and rotate the animation to adjust the view. For example, zooming on the terrestrial planets will shows something like that:




As expected, planets follow their orbits on successive revolutions around the Sun.. Nice!

You can stop the animation at any frame, and check the time that has passed since the start of the simulation. For example, to create the image shown above, we stopped the animation at State=17317 (nPRT=17317) which according to the trajectory Mol2 file,

```
@<TRIPOS>MOLECULE
nPRT=17317,step=432900,time=1558440000.0sec=25974000.00min=432900.0000hrs=18037.5000days=49.417808years
        9       0       1
SMALL
NO_CHARGES

@<TRIPOS>ATOM
   0      Sun          0.0000E+00  0.0000E+00  0.0000E+00   S       1  RES1      0.000
   1      Mercury     -8.9310E+00 -6.8976E+01 -4.8204E+00   H       1  RES1      0.000
   2      Venus        1.7047E+01 -1.0744E+02 -2.4662E+00   Mn      1  RES1      0.000
   3      Earth       -9.9566E+01 -1.1381E+02  1.4553E-02   Fe      1  RES1      0.000
   4      Mars        -4.1046E+01  2.3468E+02  5.9235E+00   O       1  RES1      0.000
   5      Jupiter     -8.4705E+01  7.6548E+02 -1.3051E+00   K       1  RES1      0.000
   6      Saturn       4.3362E+02 -1.4460E+03  7.8283E+00   Zn      1  RES1      0.000
   7      Uranus      -2.6389E+03  7.4008E+02  3.6933E+01   N       1  RES1      0.000
   8      Neptune      2.6530E+03  3.5653E+03 -1.3456E+02   N       1  RES1      0.000
```

corresponds to 18037.5 days ≈ 49.42 years since the start of the simulation.

**That's it! Do not forget to submit your program to D2L by the due date/time.**

**References**

[1] McQuarrie, D. A., Simon, J. D. *Physical Chemistry: A Molecular Approach*, University Science Books, 1997.
[2] Ball, D. W. *Physical Chemistry*, Cengage Learning, 2002.
[3] Pauling, L., Wilson, B. E. Jr. *Introduction to Quantum Mechanics with Applications to Chemistry*, Dover Publications, 1985.
[4] Serway, R. A., Vuille, C. *College Physics*, 9th edition. Cengage Learning, 2011.
[5] http://hendrianusthe.wordpress.com/2012/06/21/heliocentric-vs-geocentric
[6] http://en.wikipedia.org/wiki/Ellipse
[7] Karttunen, H., Kröger, P., Oja, H., Poutanen, M., Donner, K. J. *Fundamental Astronomy*, 5th edition. Springer, 2007.
[8] http://jwilson.coe.uga.edu/EMAT6680Fa08/Broderick/essay1/essay1.html
[9] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd edition. Cambridge University Press, 2007.
[10] Cromer, A. "*Stable solutions using the Euler approximation*". Am. J. Phys, 49, 455 (1981).
[11] http://www.physics.buffalo.edu/phy410-505/2011/topic2/app1/index.html
[12] http://en.wikipedia.org/wiki/Barycentric_Dynamical_Time
[13] http://en.wikipedia.org/wiki/Barycentric_Coordinate_Time
[14] http://ssd.jpl.nasa.gov/?horizons
[15] Card, S. K., Mackinlay, J., Shneiderman, B. *Readings in Information Visualization: Using Vision to Think (Interactive Technologies)*, Morgan Kaufmann, 1999.
[16] http://www.tripos.com/tripos_resources/fileroot/pdfs/mol2_format.pdf
[17] http://en.wikipedia.org/wiki/PyMOL
[18] http://pymol.org/2/
[19] http://www.nvidia.com/object/quadro_stereo_technology.html
[20] http://pymolwiki.org/index.php/Command_Line_Options
[21] http://www.pymolwiki.org/index.php/Talk:Practical_Pymol_for_Beginners