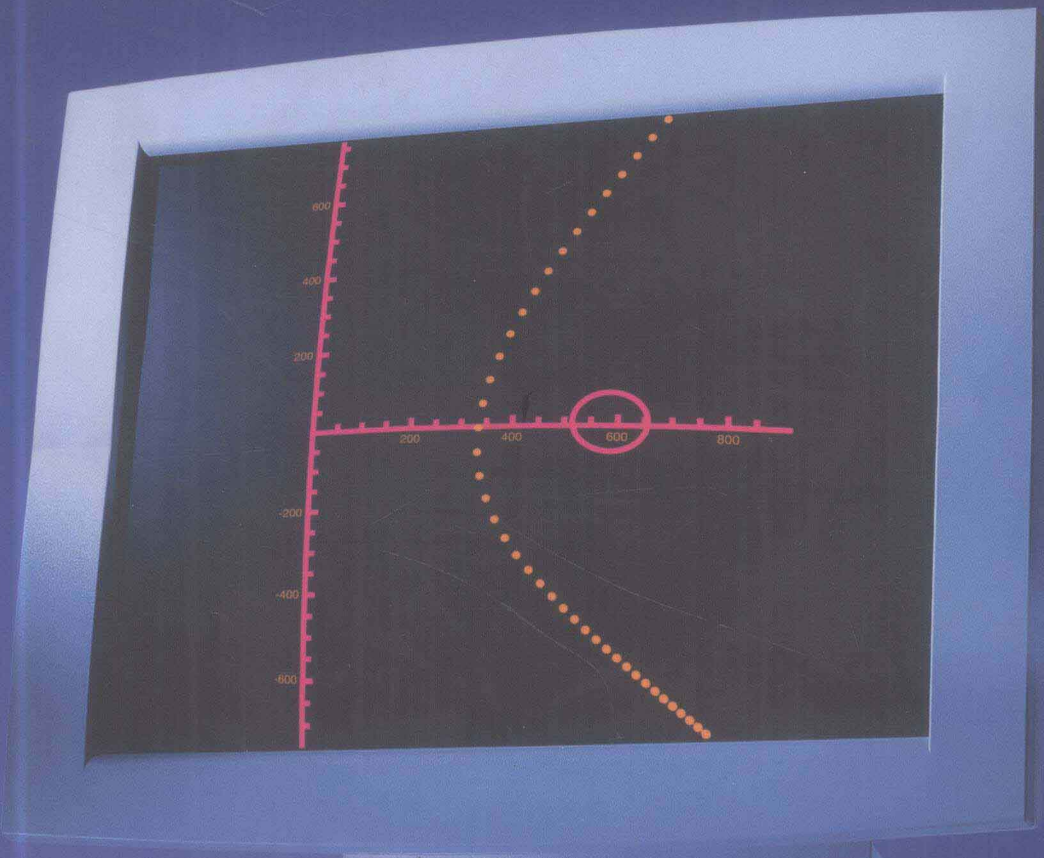
Computer Solutions in Physics

With Applications in Astrophysics, Biophysics, Differential Equations, and Engineering

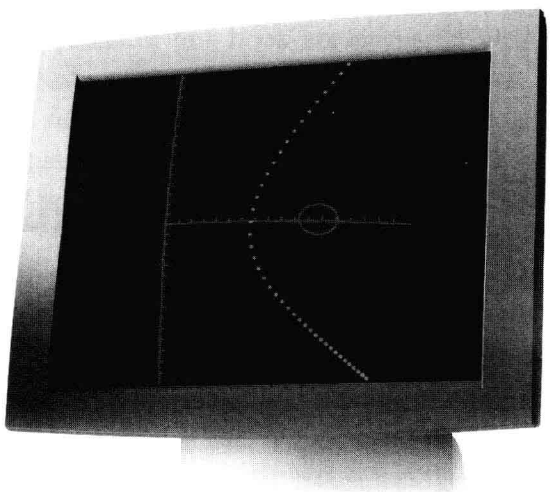


Steve VanWyk

World Scientific

Computer Solutions in Physics

With Applications in Astrophysics, Biophysics, Differential Equations, and Engineering



Steve VanWyk

Olympic College, USA

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

COMPUTER SOLUTIONS IN PHYSICS
With Applications in Astrophysics, Biophysics, Differential Equations, and Engineering
(With CD-ROM)

ISBN-13 978-981-270-936-3
ISBN-10 981-270-936-3
ISBN-13 978-981-277-499-6 (pbk)
ISBN-10 981-277-499-8 (pbk)

Printed in Singapore by World Scientific Printers

Computer Solutions in Physics

With Applications in Astrophysics, Biophysics, Differential Equations, and Engineering

Introduction

About thirty years ago, David Park wrote a book on Quantum Theory in which he advised the student to determine whether the difficulty in solving a problem was in the Physics or in the Mathematics. With the great amount of progress in numerical methods and with the speed of the modern personal computer, the problem is no longer in the mathematics. Now it is sufficient to set up the correct Physics equations and utilize any of the excellent mathematical softwares to graph and solve the problem.

The computer solutions in this book are primarily written in *Mathematica* because of its reasonably straightforward approach to mathematical problem-solving. It should be noted that the computational powers of *Maple* and *MatLab* are equally good. A comparison of *Mathematica*, *Maple*, and *MatLab* is given in Appendix A.

The programming with *Mathematica* 5 should present few difficulties. There are 50-plus computer solutions from physics and engineering and 10-plus animations contained on the compact disc. Any of these can be downloaded to your computer and run with *Mathematica* 5 or *Mathematica* 6. Once a program from the CD is on your computer, you may change the conditions or modify the equations to address a problem of your choosing. A listing of all programs is given in Appendix B. If you have never used *Mathematica* before, a capsule summary of *Mathematica* commands is given in Appendix C.

It is important to note that this is not a first-year textbook on Physics. Anyone who wants to program Physics equations should have at least one year of College Physics and a good introduction to the Calculus.

Here is the premise of this text: If you can write down the correct Physics equations, then it is only necessary to program a few lines of code to get the answer. And if the Physics equations are not correct, then the program output will tell you that as well. Either way, you win. Let's get started by setting up some Physics equations ... and let the computer knock them down.

Contents

Chapter 1 Equations of Motion

1.0 Newton's Laws

Some 320 years ago, Isaac Newton wrote down the two basic equations of classical Physics

the basic equation of force $$F = m\frac{dv}{dt} = m\dot{v}$$

and the force of gravity $$m\ddot{r} = -\frac{mGM}{r^2}$$

Let us begin with motion of a planet in a gravity field. We'll use the metric system to describe the motion of the Earth about the Sun.

Cancelling m, $\ddot{r} = -\frac{GM}{r^2}$ where $M = M_{sun} = 1.989 \times 10^{30}$ kg

and $G = 6.673 \times 10^{-11} \frac{m^3}{kg \cdot s^2}$

Notice that if we wish to measure distance in km and time in hours, then we need only change the units of G

$$G = 6.673 \times 10^{-11} \frac{m^3}{kg \cdot s^2} = 6.673 \times 10^{-20} \frac{km^3}{kg \cdot s^2}$$

$$G = 3600^2 \times 6.673 \times 10^{-20} \frac{km^3}{kg \cdot (hr)^2}$$

G is exactly the same. Only the units are changed. Choose the G you need, and express your distance units in m or km and your time in s or hr, but <u>be consistent</u> throughout your equation.

Now, back to the Earth. If we place the Earth at its *average distance* $r = 149.6 \times 10^6$ km away from the Sun, in a circular orbit with velocity v,

$$\frac{mv^2}{r} = \frac{mGM}{r^2}$$

$$v = \sqrt{\frac{GM}{r}} = 29.786 \text{ km/s}$$

Let's now write the physics equations for *circular motion* of the Earth about the Sun. To begin, we'll use x-y coordinates.

$$\ddot{x} = -\frac{GM}{x^2+y^2}\cos\theta = -\frac{GM\,x}{(x^2+y^2)^{3/2}} \qquad x_o = 149.6\times10^6\text{ km} \qquad \dot{x}_o = 0$$

$$\ddot{y} = -\frac{GM}{x^2+y^2}\sin\theta = -\frac{GM\,y}{(x^2+y^2)^{3/2}} \qquad y_o = 0 \qquad \dot{y}_o = 29.786\text{ km/s}$$

These two equations describe the motion of the Earth for all time, based on the initial conditions. Let's look at the requirements for finding the speed and location of the Earth at all times after $t = 0$. First of all these two equations are complicated and they are not linear. We shall find a solution by using numerical methods (**NDSolve** in *Mathematica*). We will solve from $t = 0$ to $t = 9000$ hours.

Here is how we do it. To get the time in hours, and the distance from the sun in kilometers, take

$$G = 3600^2 \times 6.673 \times 10^{-20}\ \frac{\text{km}^3}{\text{kg}\cdot(\text{hr})^2} \quad \text{and} \quad M_{sun} = 1.989\times10^{30}\text{ kg}$$

Then the x- and y- distance of the Earth from the center of the Sun will be measured in kilometers. Notice that to be consistent we will also have to convert the initial velocity into kilometers per hour.

Here is the *Mathematica* program

```
In[1]:= G = 3600^2 * 6.673 * 10^-20; M = 1.989 * 10^30;
        sol = NDSolve[{
            x''[t] == G * M * (-x[t]) / (x[t]^2 + y[t]^2)^1.5,
            y''[t] == G * M * (-y[t]) / (x[t]^2 + y[t]^2)^1.5,
            x[0] == 149.6 * 10^6, y[0] == 0, x'[0] == 0, y'[0] == 29.786 * 3600},
          {x, y}, {t, 0, 9000}]
```

When we press **Shift-Enter**, the *Mathematica* program returns

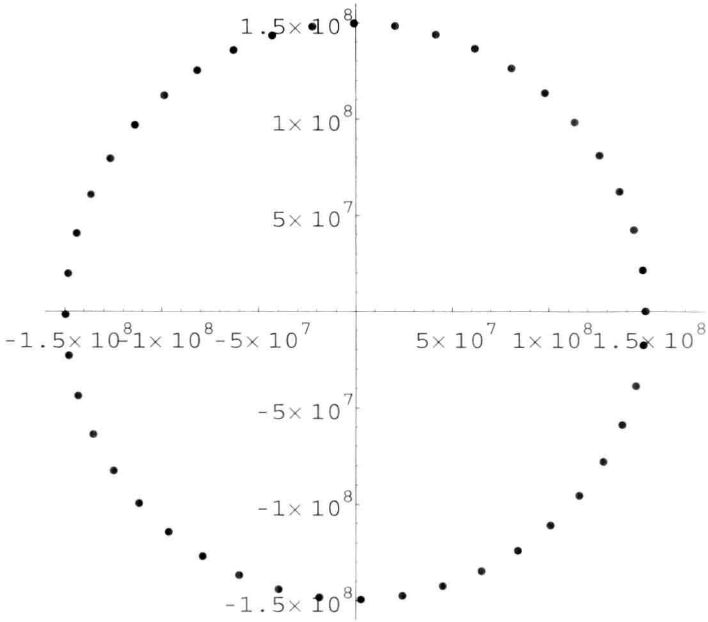
```
Out[1]= {{x → InterpolatingFunction[{{0., 9000.}}, <>],
          y → InterpolatingFunction[{{0., 9000.}}, <>]}}
```

x and y are given by an Interpolating Function. We can plot x vs y if we first identify an interpolation function in x, and an interpolation function in y and table these values versus time.

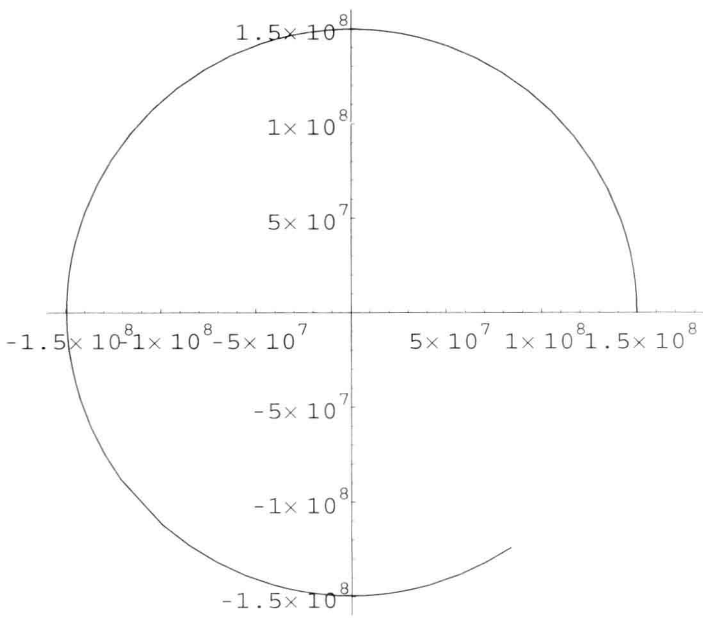
```
InterpFunc1 = x /. sol[[1]]; InterpFunc2 = y /. sol[[1]];
InterpFunc3 = x′ /. sol[[1]]; InterpFunc4 = y′ /. sol[[1]];
tbl = Table[{InterpFunc1[t], InterpFunc2[t]}, {t, 0, 8600, 200}];
```

Now we are free to plot a series of (x, y) points versus time. Either as a series of dots (**ListPlot**) or as a continuous line (**ParametricPlot**).

```
ListPlot[tbl, AspectRatio → Automatic, Prolog → AbsolutePointSize[3]]
```




```
ParametricPlot[{x[t], y[t]} /. sol, {t, 0, 7400}, AspectRatio → Automatic]
```



A Parametric Plot of the Earth in a circular orbit.

Let us now **`Table`** the data from the Interpolation Functions.

```
In[191]:= r[t_] = √(InterpFunc1[t]^2 + InterpFunc2[t]^2);
          v[t_] = √(InterpFunc3[t]^2 + InterpFunc4[t]^2) / 3600;
          Table[{t, InterpFunc1[t], InterpFunc2[t], r[t], v[t]},
            {t, 0, 8800, 400}] // TableForm
```

t (hr)	x (km)	y (km)	r (km)	v (km / s)
0	1.496×10^{8}	-1.43599×10^{-21}	1.496×10^{8}	29.786
400	1.43493×10^{8}	4.23066×10^{7}	1.496×10^{8}	29.786
800	1.25672×10^{8}	8.11593×10^{7}	1.496×10^{8}	29.786
1200	9.75899×10^{7}	1.13386×10^{8}	1.496×10^{8}	29.786
1600	6.15408×10^{7}	1.36356×10^{8}	1.496×10^{8}	29.786
2000	2.04676×10^{7}	1.48193×10^{8}	1.496×10^{8}	29.786
2400	-2.22767×10^{7}	1.47932×10^{8}	1.496×10^{8}	29.786
2800	-6.32023×10^{7}	1.35594×10^{8}	1.496×10^{8}	29.786
3200	-9.8968×10^{7}	1.12185×10^{8}	1.496×10^{8}	29.786
3600	-1.26654×10^{8}	7.96178×10^{7}	1.496×10^{8}	29.786
4000	-1.43999×10^{8}	4.05503×10^{7}	1.496×10^{8}	29.786
4400	-1.49589×10^{8}	-1.82776×10^{6}	1.496×10^{8}	29.786
4800	-1.42966×10^{8}	-4.40566×10^{7}	1.496×10^{8}	29.786
5200	-1.24671×10^{8}	-8.26886×10^{7}	1.496×10^{8}	29.786
5600	-9.61974×10^{7}	-1.1457×10^{8}	1.496×10^{8}	29.786
6000	-5.98704×10^{7}	-1.37097×10^{8}	1.496×10^{8}	29.786
6400	-1.86556×10^{7}	-1.48432×10^{8}	1.496×10^{8}	29.786
6800	2.40823×10^{7}	-1.47649×10^{8}	1.496×10^{8}	29.786
7200	6.48541×10^{7}	-1.34811×10^{8}	1.496×10^{8}	29.786
7600	1.00331×10^{8}	-1.10968×10^{8}	1.496×10^{8}	29.786
8000	1.27617×10^{8}	-7.80645×10^{7}	1.496×10^{8}	29.786
8400	1.44484×10^{8}	-3.8788×10^{7}	1.496×10^{8}	29.786
8800	1.49555×10^{8}	3.65522×10^{6}	1.496×10^{8}	29.786

The above Table shows the Earth in a circular orbit about the Sun. For our choice of starting conditions, `r` and `v` remain constant.

Just how accurate is *Mathematica*? Notice that in the above **`Table`** that we have solved the Differential Equations for a time of one year, and, as expected, the radius of the orbit and the velocity have remained absolutely constant. This is a first test of the *Mathematica* numerical solver, and it is gratifying to see that the LSODA algorithm in **`NDSolve`** produces good, consistent results. (The Livermore Solver for Ordinary Differential equations Adaptive method was developed at Lawrence Livermore Labs and utilizes both an Adams method and a Gear backward differences method to obtain results of high accuracy.)

In terms of precision, *Mathematica* routinely carries at least 16 places of decimal accuracy. This is far more than what we will need in this text, where we shall report the results of computations to 6 places (1 part per million accuracy) except in those few cases where greater precision is required.

To this point, we have only modeled the Earth as traveling in a circular orbit at one Astronomical Unit (its average distance of 149.6 million kilometers) from the center of the sun. What is necessary next, is to use the computer to find all the parameters of the Earth's *elliptical* orbit about the sun, using only Newton's Law of Gravity, and a starting distance and velocity for the Earth.

1.1 Motion of the Planets about the Sun

Let's program the actual motion of the Earth about the sun. Astronomers tell us that the Earth is closest to the sun every January 3, at a distance of 147.1×10^6 km from the center of the sun, and the velocity of the Earth at that time is 30.288 km/s. Let us program in these two numbers as *initial conditions*, and see if we can find the complete orbit of the Earth about the sun.

From the plot, the orbit appears circular. However, if we **Table** the numbers, we see that the Earth actually slows down a small amount, and moves outward from the sun by a small amount, and halfway through the orbit, speeds up again, as it comes in closer to the sun. The orbit is an ellipse.

We have programmed in a starting velocity a little greater than that required for circular orbit. Therefore our starting point is the perihelion and the point of furthest excursion is the aphelion. What is worth noting is that we only specified the location and velocity of the Earth at ONE POINT.

And this, with Newton's Law of Gravity, is all you need to describe the motion of any planet or comet about the sun, or any satellite about a planet.

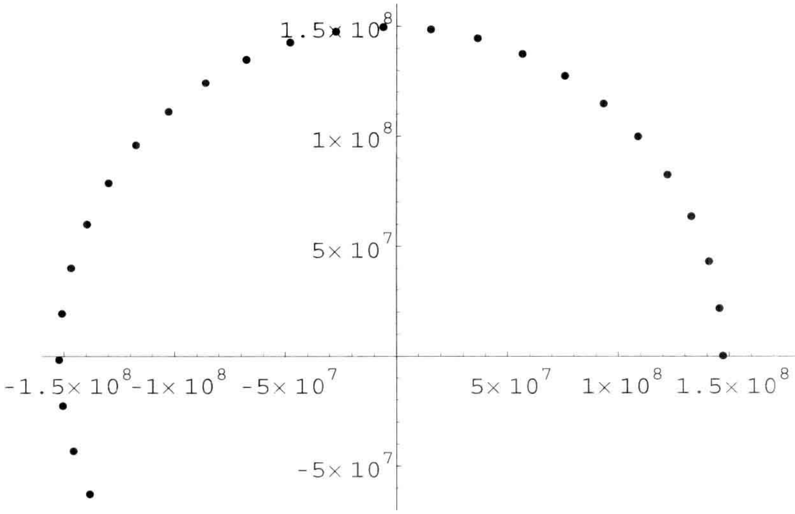
Now, the correct equations of motion for the Earth about the sun are

$$\ddot{x} = -\frac{GM}{x^2+y^2}\,\mathrm{Cos}\,\theta = -\frac{GM\,x}{(x^2+y^2)^{3/2}} \qquad x_o = 147.1\times 10^6\ \mathrm{km} \qquad \dot{x}_o = 0$$

$$\ddot{y} = -\frac{GM}{x^2+y^2}\,\mathrm{Sin}\,\theta = -\frac{GM\,y}{(x^2+y^2)^{3/2}} \qquad y_o = 0 \qquad \dot{y}_o = 30.288\ \mathrm{km/s}$$

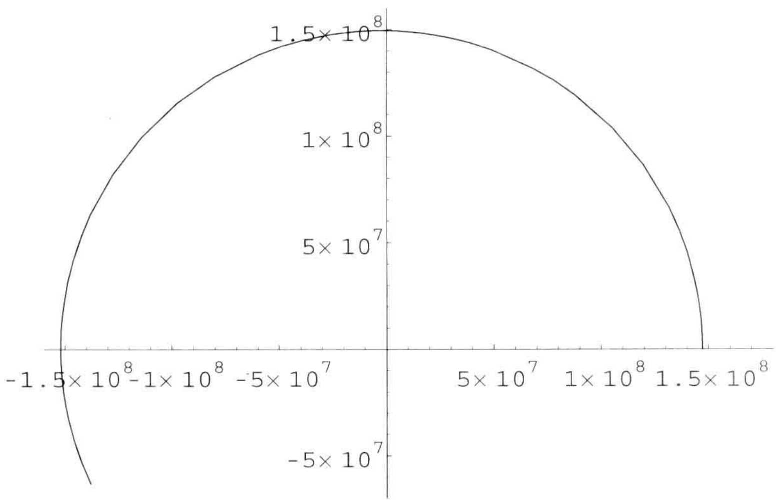
```
G = 3600^2 * 6.673 * 10^-20; M = 1.989 * 10^30;
sol1 = NDSolve[
  {x''[t] == G * M * (-x[t]) / (x[t]^2 + y[t]^2)^1.5,
   y''[t] == G * M * (-y[t]) / (x[t]^2 + y[t]^2)^1.5, x[0] == 147.1 * 10^6,
   y[0] == 0, x'[0] == 0, y'[0] == 30.288 * 3600}, {x, y}, {t, 0, 9000}]

InterpFunc1 = x /. sol1[[1]]; InterpFunc2 = y /. sol1[[1]];
tbl = Table[{InterpFunc1[t], InterpFunc2[t]}, {t, 0, 5000, 200}];
ListPlot[tbl, AspectRatio → Automatic, Prolog → AbsolutePointSize[3]]
```



The orbit of the Earth, from January to July.

```
ParametricPlot[{x[t], y[t]} /. sol1[[1]],
  {t, 0, 5000}, AspectRatio → Automatic];
```



The orbit of the Earth about the sun (at 0, 0). Although the orbit appears circular, the following Table shows that it is an ellipse.

```
In[210]:= InterpFunc3 = x' /. sol1[[1]]; InterpFunc4 = y' /. sol1[[1]];
          r[t_] = √(InterpFunc1[t]^2 + InterpFunc2[t]^2)
          Vel[t_] := √(InterpFunc3[t]^2 + InterpFunc4[t]^2) /3600
          Table[{t, InterpFunc1[t], Chop[InterpFunc2[t]],
             r[t], Vel[t]}, {t, 0, 8800, 400}] // TableForm
```

t (hr)	x (km)	y (km)	r (km)	v (km / s)
0	1.471×10^{8}	0	1.471×10^{8}	30.288
400	1.40788×10^{8}	4.29893×10^{7}	1.47205×10^{8}	30.2666
800	1.2242×10^{8}	8.22999×10^{7}	1.47512×10^{8}	30.2046
1200	9.36301×10^{7}	1.1461×10^{8}	1.47994×10^{8}	30.1076
1600	5.69426×10^{7}	1.37264×10^{8}	1.48607×10^{8}	29.9844
2000	1.55064×10^{7}	1.48492×10^{8}	1.49299×10^{8}	29.846
2400	-2.72117×10^{7}	1.47524×10^{8}	1.50013×10^{8}	29.7039
2800	-6.77342×10^{7}	1.34609×10^{8}	1.5069×10^{8}	29.5697
3200	-1.02851×10^{8}	1.10935×10^{8}	1.51277×10^{8}	29.4539
3600	-1.29847×10^{8}	7.84934×10^{7}	1.51728×10^{8}	29.3652
4000	-1.46679×10^{8}	3.99028×10^{7}	1.5201×10^{8}	29.31
4400	-1.52089×10^{8}	-1.79697×10^{6}	1.521×10^{8}	29.2923
4800	-1.45678×10^{8}	-4.3357×10^{7}	1.51993×10^{8}	29.3133
5200	-1.27919×10^{8}	-8.15376×10^{7}	1.51696×10^{8}	29.3715
5600	-1.00141×10^{8}	-1.13326×10^{8}	1.51232×10^{8}	29.4628
6000	-6.44497×10^{7}	-1.36152×10^{8}	1.50635×10^{8}	29.5806
6400	-2.36095×10^{7}	-1.48083×10^{8}	1.49953×10^{8}	29.7158
6800	1.91367×10^{7}	-1.48007×10^{8}	1.49239×10^{8}	29.8581
7200	6.03011×10^{7}	-1.35761×10^{8}	1.48551×10^{8}	29.9957
7600	9.64315×10^{7}	-1.12202×10^{8}	1.47947×10^{8}	30.117
8000	1.24421×10^{8}	-7.91801×10^{7}	1.47479×10^{8}	30.2113
8400	1.41812×10^{8}	-3.94194×10^{7}	1.47188×10^{8}	30.2701
8800	1.47054×10^{8}	3.71581×10^{6}	1.47101×10^{8}	30.2878

The above Table shows the motion of the Earth in its elliptical orbit about the sun. Notice that the maximum distance r (aphelion) is near t = 4400 hours, and the minimum distance (perihelion) at t = 0, and again at t = 8800 hours.