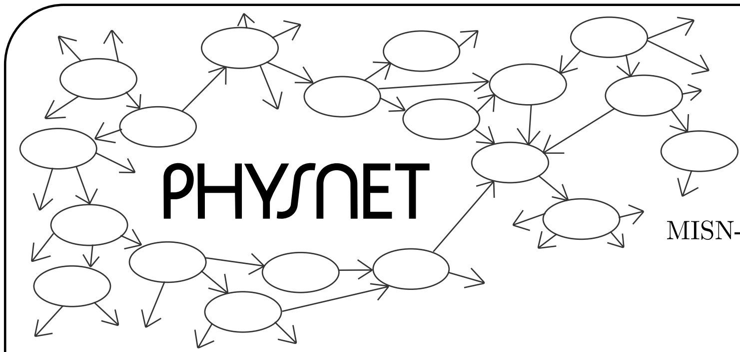
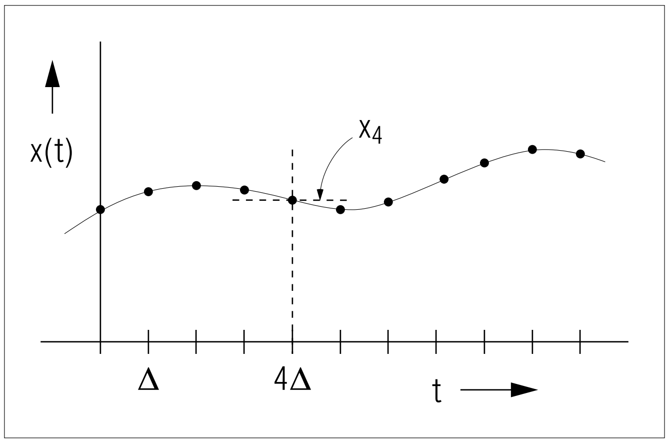


PHYSNET

MISN-0-104

THE NUMEROV ALGORITHM FOR SATELLITE ORBITS



$$x_{n+1} = a_n x_n + b_n x_{n-1}$$

Project PHYSNET Physics Bldg. Michigan State University East Lansing, MI

THE NUMEROV ALGORITHM FOR SATELLITE ORBITS

by

Peter Signell

Title: **The Numerov Algorithm for Satellite Orbits**

Author: Peter Signell, Michigan State University

Version: 2/1/2000 Evaluation: Stage 0

Length: 1 hr; 12 pages

Input Skills:

1. State the 3-point finite difference approximation to a second derivative (MISN-0-4).
2. State Newton's Second Law (MISN-0-14).
3. State Newton's Law of Gravitation (MISN-0-101).

Output Skills (Knowledge):

K1. Derive the recurrence relation for the Numerov Algorithm, to second order and in two dimensions, in a form suitable for use in obtaining satellite trajectories numerically and showing all steps in the derivation.

K2. Derive equations for insertion of initial position and velocity in the Numerov Algorithm and communicate a method of obtaining a particular desired accuracy.

Post-Options:

1. "Orbits in an Inverse Square Law Force Field: A Computer Project" (MISN-0-105).
2. "Computer Algorithm For the Damped Driven Oscillator" (MISN-0-39).

THE NUMEROV ALGORITHM FOR SATELLITE ORBITS

by

Peter Signell

1. Introduction

Many problems in science and engineering cannot be solved in terms of known functions,even when the underlying equation is known. Such a problem is the position as a function of time along a satellite orbit or, for that matter, along the earth's orbit around the sun. For such cases, one must resort to approximate numerical techniques; one such technique is examined in this module.

2. The Equation of Motion

2a. Newton's Second Law Plus the Law of Gravitation. We will numerically determine the trajectory of a satellite in the field of a single gravitational source, such as the earth. The equation governing such motion can be obtained by combining Newton's Second Law,[1]

$$\vec F = m\frac{d^2\vec r(t)}{dt^2} \equiv m\vec r\,''(t),$$

with Newton's Law of Gravitation,[2]

$$\vec F = -\frac{Gmm_E}{r^2(t)}\,\hat r(t).$$

We get:

$$\vec r\,''(t) = -\frac{Gm_E}{r^2(t)}\,\hat r(t) = -\frac{Gm_E}{r^3(t)}\,\vec r(t).$$

Here $r(t)$ is the time changing position vector of the mass m and $r(t)$ is the radially outward pointing unit vector at the position of the mass at

[1]The number of primes will indicate the order of the derivative with respect to time in this module.
[2]See "Newton's Law of Gravitation" (MISN-O-101).

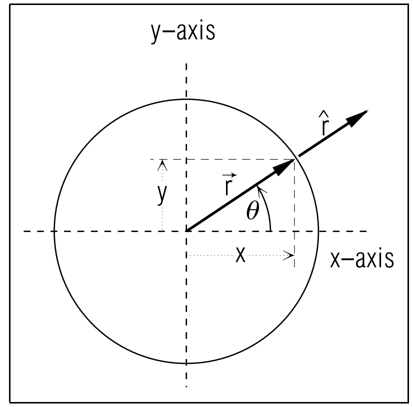


Figure 1. The vectors used to define a particle's position.

time t.[3] The origin of the coordinate system has been placed at the force center.[4]

2b. The Equation in Cartesian Coordinates. We can write the above vector equation in Cartesian component form by writing $\vec r$ in terms of Cartesian unit vectors $\hat x$ and $\hat y$:

$$\vec r(t) = x(t)\,\hat x + y(t)\,\hat y.$$

The reason for going over to Cartesian coordinates is that $\hat x$ and $\hat y$ are independent of time whereas $r(t)$ is not.That is, $\hat x$ and $\hat y$ form a time-independent reference system. Then taking components and writing the magnitude of $\vec r$ in terms of $\hat x$ and $\hat y$,

$$|\vec r(t)|^2 = x^2(t) + y^2(t),$$

we get the $\hat x$ equation:

$$x'' = -\frac{G\,m_E}{\left[x^2(t)+y^2(t)\right]^{3/2}}\,x(t) \equiv f(t)\,x(t).$$

Similarly for the y equation:

$$y\,''(t) = f(t)\,y(t).$$

[3]See "Kinematics: Circular Motion" (MISN-O-9).
[4]For most satellite problems the center of the Earth can be taken to be a fixed force center. For the motion of the moon, however, that is insufficient due to its large mass. The methods presented here still apply, but the Earth's mass becomes replaced by the mass of the total system. Other quantities require careful interpretation. For further details, see "Two Body Kinematics and Dynamics" (MISN-O-45).

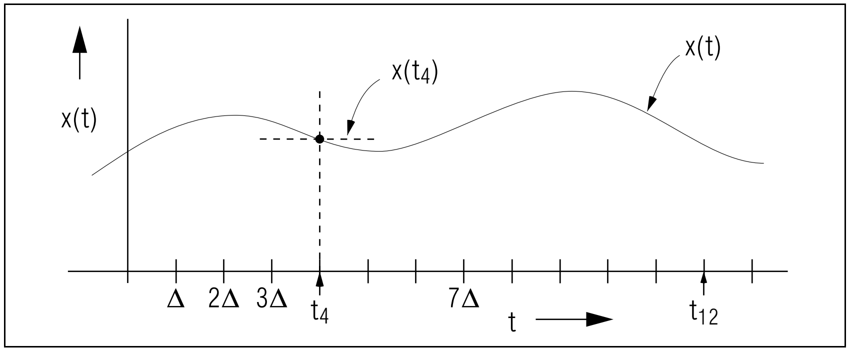


Figure 2. A function $x(t)$, specified at equally-spaced values of t.

3. Second-Order Numerov Solution

3a. "Net-Point" Notation. We will here use the Numerov method of solving equations involving derivatives. This method has the advantage of being easily understandable, although it is less appropriate for orbit problems than are some other less understandable methods.[5] In the Numerov method we deal with the solution functions, $x(t)$ and $y(t)$, as a series of numbers at "net-point" times that are integrally spaced:

$$t_n \equiv n\,\Delta.$$

This is illustrated in Figure 2 for $x(t)$.

We now introduce a more succinct notation for the coordinate positions at the net-point times:

$$x_n \equiv x(t_n) \equiv x(n\Delta),$$

$$y_n \equiv y(t_n) \equiv y(n\Delta),$$

and our equations become:

$$x_n{}'' = f_n x_n, \tag{1}$$

$$y_n{}'' = f_n y_n,$$

where:

$$f_n \equiv -G\,m_E\,\left(x_n^2 + y_n^2\right)^{-3/2}.$$

[5]The Numerov method tends to allow errors to accumulate. This is all right if the errors are both positive and negative so they tend to cancel out. However, for orbit problems the errors tend to all have the same sign so a method like the Runge-Kutta is better.

3b. Finite Difference Approximation. We will now connect the consecutive values of x and y by using the finite difference approximation to any function's g's second derivative:

$$g_n{}'' \approx \left(g_{n+1} - 2g_n + g_{n-1}\right)/\Delta^2. \tag{2}$$

Equation (2) is the result of truncating the following series after the second term:[6]

$$g_{n+1} + g_{n-1} = 2g_n + \Delta^2 g_n'' + (\Delta^4/12)g_n^{iv} + \ldots \tag{3}$$

The truncation is justifiable to the extent that numerical values of *Delta* will be small so that Δ^4 will be insignificant compared to Δ^2. Keeping that in mind, we apply Eq. (2) to the x and y functions of Eq. (1) and get, with a little rearrangement:

$$x_{n+1} = (2 + \Delta^2 f_n)x_n - x_{n-1}. \tag{4}$$

Similarly, the y-coordinate relation is:

$$y_{n+1} = (2 + \Delta^2 f_n)y_n - y_{n-1}. \tag{5}$$

Such truncation of the series at the second term results in an algorithm that is referred to as "being of second order."

▷ Derive Eq. (5).

3c. The Recurrence Relations. Remembering that $f_n = -Gm_E(x_n^2 + y_n^2)^{3/2}$, we see that knowledge of x_0, y_0, x_1, and y_1 will enable us to compute x_2 and y_2. Then, using x_1, y_1, x_2, and y_2, we can compute x_3, and y_3. This process can be repeated until x and y are known at any time you wish. The relations in Eqs. (4)-(5) are thus called "recurrence relations." The only remaining problem is the specification of x_0, y_0, x_l and $y1$.

3d. Specifying Initial Position and Velocity. In order to specify the trajectory, we usually find it most practical to specify position and velocity at some particular time. This means that we specify: position: x_0, y_0. velocity: x_0', y_0'. In order to convert x_0, y_0, x_0', y_0', into $x0$, $y0$, x_1, and y_1, we subtract Eq.(2) at $t = 0$ to obtain:

$$x_1 - x_{-1} = 2\Delta x_0', \tag{6}$$

[6]See "Taylor's Series for the Expansion of a Function About a Point" (MISN-O-4).

while the $t = 0$ sum equations (4)-(5) are:

$$x_1 = (2 + \Delta^2 f_0)\, x_0 - x_{-1}. \tag{7}$$

The unknown x_{-1} can be eliminated from Eq. (6) to give:

$$x_1 = (1 + 2\Delta^2 f_0)\, x_0 + \Delta x_0{}'. \tag{8}$$

Similarly,

$$y_1 = (1 + 2\Delta^2 f_0)\, y_0 + \Delta y_0{}'. \tag{9}$$

To summarize, we specify the initial position and velocity components, x_0, y_0, $x_0{}'$, $y_0{}'$, and then use Eqs. (8) to obtain x_1 and y_1. These are combined with x_0 and y_0 to start the recurrence relations (4)-(5).[7]

4. Choice of Step Size

Finally, how do you know what size time interval Δ to use? You could attempt to assess the importance of successive terms in Eq. (3), but a more reliable method is to decrease Δ until the predicted trajectory stabilizes. That is, until it does not change significantly when Δ is made even smaller.

5. Summary

The algorithm, then, consists of:

1. a recurrence relation;
2. two initial conditions; and
3. a method of assuring desired accuracy.

Acknowledgments

I would like to thank Dr. N. R. Yoder for the many insights I gained in this area while collaborating with him. Preparation of this module was supported in part by the National Science Foundation, Division of Science Education Development and Research, through Grant #SED 74-20088 to Michigan State University.

[7] See "Orbits in an Inverse Square Law Force Field: A Computer Project" MISN-0-105), wherein the presently derived algorithm is used to calculate orbits and trajectories.

MODEL EXAM

1. See Output Skills K1-K2 in this module's *ID Sheet*. One or both of these skills may be on the actual exam.

Brief Answers:

1. See this module's *text*.