Differential Equations, The Language of Science,

a brief introduction to scientific simulation

by Charles W. Neville

Part 1. Differential Equations, The Language of Science

0. Introduction

Differential equations are equations involving derivatives. As an example, consider the differential equation

$$d^2x/dt^2 + 9x = 0 \tag{1}$$

This is an equation for a function $x(t)$ in terms of the function x and its second derivative, d^2x/dt^2. If this equation seems mysterious, don't worry. In the next section, we will *solve it*, that is we will find a function $x = x(t)$ which, when substituted into equation 1, satisfies the equation. We will also review the necessary calculus.

Differential equations are the language of science. The reason is easy to understand, Newton's second law of motion,

$$F = ma \tag{2}$$

where F is the force action on a body, m is the mass of the body, and a is the acceleration of the body.

To see why Newton's second law implies that differential equations are the language of science, you need to recall from calculus or physics courses the relation between position x, velocity v and acceleration a,

$$v = \text{rate of change of position} = dx/dt$$
$$a = \text{rate of change of velocity} = dv/dt = d^2x/dt^2$$

Thus the velocity of a body is the first derivative of its position with respect to time, and the acceleration of a body is the second derivative of its position with respect to time. Hence, we may rewrite Newton's second law of motion, equation 2 above, as

$$F = md^2x/dt^2 \tag{3}$$

. If the force F depends on the position x and the time t, as it usually does, then we may rewrite equation 2 as

$$F(x,t) = md^2x/dt^2 \tag{4}$$

This is a differential equation for the position x in terms of the function $F(x,t)$.

We have just considered the motion of a single body in one dimension subject to designated forces. We can generalize these observations to handle systems of n bodies in three dimensions interacting through forces given as functions of position and time. Suppose the masses of the n bodies are $m_1, m_2, \ldots, m_n$, the vector positions of the bodies at time t are $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, and the vector forces acting on the bodies are

$$\mathbf{F}_1(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t), \quad \mathbf{F}_2(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t), \quad \ldots, \quad \mathbf{F}_n(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t)$$

Then we may apply Newton's second law of motion to obtain the system of vector differential equations,

$$\begin{aligned}
\mathbf{F}_1(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t) &= m_1 d^2\mathbf{x}_1/dt^2 \\
\mathbf{F}_2(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t) &= m_2 d^2\mathbf{x}_2/dt^2 \\
&\vdots \\
\mathbf{F}_n(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, t) &= m_n d^2\mathbf{x}_n/dt^2
\end{aligned} \tag{5}$$

In a later section, we shall give a concrete example of such a system where there are 3 bodies (Earth, Moon and a rocket coasting through space from the Earth to the Moon) interacting through gravitational attraction.

1. The Simple Harmonic Oscillator

All of this probably seems quite abstract. So let us consider a concrete case, a vibrating spring. We idealize the spring as having no mass. It is anchored at one end to a wall, and it has a small metal ball of mass m attached to the other end, which is free to move. To keep things simple, we suppose that the free end of the spring only moves along the x axis, so we have a simple concrete case of a single body with mass m (the small ball) moving in one dimension. When the spring is stretched, Hooke's law gives us the force F acting on the small ball,

$$F = -kx \tag{1}$$

where x is the displacement of the free end of the spring from its rest (unstretched) position. Thus x also measures the position of the small ball, where the origin of the x coordinate system is taken to be the rest position of the free end of the spring.

The negative sign in equation 1 arises from the fact that the force on the small ball on the free end of the spring is directed in the opposite direction from the displacement, and k is a constant depending on the material composition and thickness of the spring. Naturally, k is called the *spring constant* of the spring.

Now substitute Newton's second law into the force equation arising from Hooke's law,

$$\begin{aligned}
F &= md^2x/dt^2 \quad \text{from Newton's second law} \\
F &= -kx \quad \text{from Hooke's law}
\end{aligned}$$

to get

$$md^2x/dt^2 = -kx$$

This is usually put into the form,

$$dx^2/dt^2 + (k/m)x = 0 \tag{2}$$

which is just the simple harmonic oscillator equation (equation 0.1) with which we began the introduction.

The point is that for the simple physical system we have just considered, Newton's second law of motion plus a simple force law (Hooke's law in this case) give rise to a *differential equation* which completely describes the motion of the system. Of course, to convince you of this, we have to *solve* the simple harmonic oscillator equation. We shall procede to do this now.

2. A Very Brief Calculus Review

Before we can solve the simple harmonic oscillator equation, we have to very briefly review a few facts from calculus.

First, let us take a few derivatives:

$$\begin{aligned} (1) \qquad & d\sin(x)/dx = \cos(x), \quad d\sin(3x)/dx = 3\cos(3x) \\ & d\cos(x)/dx = -\sin(x), \quad d\cos(3x)/dx = -3\sin(3x) \quad d3\cos(3x)/dx = -3^2\sin(3x) \end{aligned}$$

The derivatives of $\sin(3x)$ and $\cos(3x)$ are obtained by using the infamous *chain rule*, $(f(g(x))' = f'(g(x) \cdot g'(x)$. The chain rule is one of the harder parts of elementary differential calculus to master, and misapplications of the chain rule are responsible for 70% of all errors by freshmen and sophomores on calculus tests. However, I am sure that *you* have thoroughly mastered the chain rule by now.

We can combine the top and bottom equations in 1 to see that

$$d^2\sin(3x)/dx = -3^2\sin(3x)/dx$$

or, in words, the second derivative of $\sin(3x)$ is $-3^2\sin(3x)$. Similarly, we can show that

$$\begin{aligned} (2) \qquad & d^2\sin(kx)/dx = -k^2\sin(kx), \quad \text{and} \\ & d^2\cos(kx)/dx = -k^2\cos(kx) \end{aligned}$$

3. Solving the Simple Harmonic Oscillator Equation

Now that we have reviewed a bit of calculus, we can solve the simple harmonic oscillator equation which describes the motion of the small ball of mass m attached to the free end of a vibrating spring,

$$(1) \qquad d^2x/dt^2 + (k/m)x = 0$$

We can substitute the function $x(t) = \sin(\sqrt{k/m}\,t)$ into the simple harmonic oscillator equation and use equations 2 of the brief calculus review (equations 2.2) to see that

$$\begin{aligned} & d^2\sin(\sqrt{k/m}\,t)/dt^2 + (k/m)\sin(\sqrt{k/m}\,t) \\ & = -(k/m)\sin(\sqrt{k/m}\,t) + (k/m)\sin(\sqrt{k/m}\,t) = 0 \end{aligned}$$

. In other words, the function $x(t) = \sin(\sqrt{k/m}\,t)$ is a solution to the simple harmonic oscillator equation. Similarly, we can show that the function $x(t) = \cos(\sqrt{k/m}\,t)$ is also a solution. Both of these solutions are physically realistic, because the graphs of x vs t are oscillating sine and cosine waves, each with a period of $2\pi/\sqrt{k/m}$. (Remember your basic trigonometry!). This agrees with our physical intuition, and with our observations, about the way real springs vibrate.

We can combine these two solutions to obtain the general solution of the simple harmonic oscillator equation, equation 1,

$$(2) \qquad x(t) = A\cos(\sqrt{k/m}\,t) + B\sin(\sqrt{k/m}\,t)$$

The constants A and B are determined by the position and velocity of the small metal ball attached to the free end of the spring at time $t = 0$.

Let us do a numerical example. Suppose $k/m = 9$, as in equation 1 of the introduction (equation 0.1), so that $\sqrt{k/m} = 3$. Then the general solution is

$$x(t) = A\cos(3t) + B\sin(3t) \tag{3}$$

Suppose that the position of the small metal ball at time $t = 0$ is $x(0) = 2$, and that the velocity of the small metal ball at time $t = 0$ is $x'(0) = 12$. (Here we are using Newton's prime notation to designate derivatives.) If we substitute $t = 0$ into the general solution, equation 3, and use the facts that $\cos(0) = 1$ and $\sin(0) = 0$, we obtain

$$2 = x(0) = A\cos(0) + B\sin(0) = A$$

Thus $A = 2$. Similarly, we can take the derivative of the general solution, equation 3, to obtain

$$x'(t) = -3A\sin(3t) + 3B\cos(3t)$$

so

$$12 = x'(0) = -3A\sin(0) + 3B\cos(0) = 3B$$

Hence $B = 4$. Thus the solution is

$$x(t) = 2\cos(3t) + 4\sin(3t)$$

To recapitulate, the solution to the differential equation

$$d^2x/dt^2 + 9x = 0 \tag{4}$$

with initial conditions $x(0) = 2$ and $x'(0) = 12$ is

$$x(t) = 2\cos(3t) + 3\sin(3t) \tag{5}$$

Now we have done all we promised at the beginning of the introduction and at the end of section 1. The general solution to the differential equation 4 is given by equation 3. The constants A and B are determined by the values of x and x' at $t = 0$. In the particular case where $x(0) = 2$ and $x'(0) = 12$, the particular solution is given by equation 5.

4. The Scope of Differential Equations

The simple harmonic oscillator equation describes much more than vibrating springs. It also describes (1) the flow of current in an electrical circuit consisting of a coil and a capacitor, (2) the motion of an atom or molecule vibrating around its rest position in a crystal lattice, and even (3) the oscillations of a vibrating electromagnetic field.

Each example we have given so far comes from classical physics, that is physics as it was understood before the advent of quantum mechanics. But it turns out that differential equations also describe quantum mechanical systems. (For the *cognoscenti*, these differential equations often involve partial derivatives, so they are called *partial differential equations*.) In particular, the quantum mechanical harmonic oscillator plays a very important role in many areas of quantum mechanics.

Let us consider the implications of the last paragraph. The applicability of differential equations goes far beyond the description of simple systems in classical physics, such as the vibrating spring. Indeed, differential equations (possibly involving partial derivatives) describe virtually ALL

systems in nature, whether classical or quantum mechanical. Here is a short list of some of the systems so described:

1 The solar system (understood by Newton in the 17th century).
2 Supernova explosions (understood only in the past 20 years).
3 Chemical reactions (understood in principle by the 1930's, understood in detail only in the past 20 years).
4 The folding and configuration of DNA molecules (only partially understood today).

Note that only the first two could be considered as part of physics (of astronomy or astrophysics, actually). The remaining two involve chemistry and biochemistry! The reason that differential equations describe the behavior of systems in such widely differering areas of science is that ALL natural phenomena are reducible to the basic laws of physics, whether of classical physics or quantum mechanics, and the basic laws of physics are phrased in terms of differential equations. This has been understood since the time of Laplace in the late 18th century, although our appreciation of the depth of this insight has expanded remarkably in the intervening centuries as our understanding of physics has improved. (We also understand now that Laplace's arguments were wrong, although his conclusions were correct.)

The largely successful attempt to reduce all natural phenomena to the first principles of physics is called "reductionism" by those who don't like it, and "the unity of science" by those who do. As you can tell, I am a "unity of science" man.

But one must be a bit cautious here. There are some systems, involving the aggregate behavior of multitudes of particles, which are so complex that it will probably never be possible to derive all of their behavior from first principles and differential equations. A good example of such systems are living, biological organisms. Though we can describe many important details of a given organism in terms of biochemistry, thus in terms of chemistry, and thus ultimately in terms of the first principles of physics, we will probably never understand the overall behavior of any organism this way. We will not understand why kittens purr, or are cuddly, nor will we understand their uncanny skill at play and hunting. And we will not understand why we perceive kittens as cute and cuddly, nor why we admire their ability to play and hunt, at least we will not in terms of differential equations.

There are other sciences which are ultimately historical and so cannot be reduced to the first principles of physics and differential equations. Take evolution for example. We can demonstrate evolution with simple computer simulations; after all, that is what genetic programming is all about. But, as Stephen J. Gould has so eloquently pointed out, evolution in nature is both contingent and accidental. To survive long enough to pass your genes on, you have to be both capable and lucky. The same holds for species. Because chance plays such a large role, we will never repeat the evolution of dinosaurs. Thus paleontology must remain a largely historical and descriptive science, based on careful observation.

Part 2. Scientific Simulation and Differential Equations

0. Introduction

Virtually all scientific simulations involve the numerical solution by computer of systems of differential equations, because differential equations describe the behavior of almost all systems occuring in nature. In part 2, we will learn how to solve differential equations numerically, and set up some simple scientific simulations.

First, we need to understand the two basic ways to solve systems of differential equations. The first, which we have already illustrated in part 1 with the simple harmonic oscillator equation, is to

use analytic methods. Of course, there are actually many different analytic methods, but they all have this in common: They produce a function or set of functions, given by formulas, which solve the given system of differential equations. Thus, an *analytic solution* to a system of differential equations is given by a formula or set of formulas. It is then possible to graph the functions given by these formulas, or to subject them to other sorts of analysis, to obtain fundamental information about the underlying physical system. For example, from our analytic solution to the simple harmonic oscillator equation, we were able to compute the period of the spring vibration as $2\pi/\sqrt{k/m}$.

The second basic way to solve systems of differential equations is to use numerical methods. Again, there are many different numerical methods, but they all have this in common: They produce a table of approximate numerical values for the function or set of functions which solve the given system of differential equations. Thus a *numerical solution* to a system of differential equations is given by a table or set of tables. It is then possible to graph the solution functions from the tablulated data, or to subject the tabulated data to other sorts of analysis, to obtain fundamental information about the underlying physical system.

1. Scientific Simulation, a Revolution in Science

Before the advent of widely available powerful digital computers, scientists attempting to predict the behavior of systems in nature from the first principles of classical physics or quantum mechanics had to try to solve the relevant systems of differential equations analytically. Unfortunately though, most systems of differential equations are too complicated to solve analytically, so scientists had to make various approximations and simplifications, or perhaps just consider parts of the system in isolation, in order to obtain approximate analytic solutions to use in deducing fundamental properties of the system. For example, the use of the simple harmonic oscillator equation to describe the motion of an atom or molecule vibrating around its rest position in a crystal lattice arises in just this way. (For the *cognoscenti*, it only describes small vibrations accurately, because it is based on a Taylor series approximation to the potential.)

This is a very powerful method, and is still, even in our computer age, regarded by theoretical physicists as the best method for many purposes. But the method has inherent limitations. It cannot be applied to understand supernova explosions, or to predict chemical reactions, because the systems of differential equations are simply too complex.

The wide availability of powerful digital computers within the past 20 years has created a REVOLUTION IN SCIENCE, because it has made it possible to solve even very complex systems of differential equations numerically. Thus, it has made it possible to numerically simulate even very complex systems in nature, and to investigate their behavior under various combinations of inputs.

For example, it is numerical simulation by computer that has made it possible to predict chemical reactions from the first principles of physics. These *ab initio* simulations, as they are called, are now so accurate that their results are taken as the definitive values for various numerical parameters associated with simple chemical reactions and even moderately complex organic molecules. In other words, the results of *ab initio* simulations are now taken to be more accurate than experimental measurements in many cases.

Of course, there are still chemical and biochemical systems which are too complex to simulate, yet. For example, two of the "grand challenge" problems in computer science are (1) the simulation of chemical reactions involving really large organic molecules, and (2) the simulation of the folding of DNA molecules. There are Nobel prizes waiting there for the bright and adventurous.

Here are two other examples: (1) Supernova explosions are now understood through numerical scientific simulations. (2) Short range weather prediction is now possible (and highly accurate)

through numerical scientific simulations. There is a grand challenge problem associated with numerical weather prediction: accurate long range weather prediction. Considerable progress has been made on this in the past few years. For example, the long range effect of *el nino* and *la nina* events on weather was first discovered by observation of climate data, but is now much better understood through numerical scientific simulation.

If you are interested in participating in this tremendously exciting scientific revolution, you should consider enrolling in one of the many doctoral programs in computational science offered at the great universities in North America and Europe.

Of course, this is a "brief introduction to scientific simulation", so the scientific simulations we shall consider are going to be much simpler than the great achievements and grand challenge problems just described. In the remaining sections, we are going to introduce several numerical methods for solving differential equations, and then we are going to apply these methods to simulate two much more modest systems, a vibrating spring and the orbit of a rocket coasting through space from the Earth to the Moon.

2. The Forward Euler Method

The simplest numerical method for solving differential equations is the Forward Euler Method, also called Solution by Forward Differences. This is also the easiest numerical method to understand, so it is a good place to begin. Unfortunately, it is also the least accurate of the standard numerical solution methods, so it is seldom used alone. However, it forms the starting point for several more accurate methods, and so it finds frequent use as a part of other methods.

Consider the differential equation,

$$dx/dt = f(x,t), \quad x(t_0) = x_0 \tag{1}$$

Now recall the definition of the derivative of the function $x(t)$ from calculus,

$$dx/dt = \lim_{\Delta t \to 0} \Delta x/\Delta t = \lim_{h \to 0} (x(t+h) - x(t))/h$$

The expression $\Delta x/\Delta t$ is called a *difference quotient.* If we make $\Delta t = h$ small, and replace the derivative dx/dt by the difference quotient $\Delta x/\Delta t$, we get a good approximation to the numerical value of the derivative. The forward difference method is based on this very simple idea. In detail, we choose a small value for $\Delta t = h$ and write

$$\begin{aligned} \Delta x/\Delta t &= f(x(t),t) \\ \Delta x &= f(x(t),t)\Delta x \\ x(t+h) - x(t) &= f(x(t),t)h \\ x(t+h) &= x(t) + f(x(t),t)h \end{aligned}$$

In summary, directly from the definition of the derivative and the differential equation 1, we obtain the approximation

$$x(t+h) = x(t) + f(x(t),t)h \tag{2}$$

When we construct a table of approximate numerical values for the solution function $x(t)$ using equation 2, we select a small value for the increment h and then consider a number N of evenly spaced points on the t interval separated by h, beginning with the given initial value of t, that is we consider

$$t_0, t_1 = t_0 + h, t_2 = t_1 + h, \ldots, t_{n+1} = t_n + h, \ldots, t_N$$

We also consider the corresponding values of the solution function $x(t)$ beginning with the given inital value of x,

$$x_0 = x(t_0), x_1 = x(t_1), x_2 = x(t_2), \ldots, x_{n+1} = x(t_{n+1}), \ldots, x_N = x(t_N)$$

Then equation 2 becomes

$$x_{n+1} = x_n + f(x_n, t_n)h \tag{3}$$

Of course, the starting value x_0 is given by the initial condition $x(t_0) = x_0$ in equation 1. Equation 3 is often called the *forward difference scheme*.

All of this will be clearer if we do an illustrative example, so consider the specific case

$$dx/dt = -2x, \quad x(0) = 16, \tag{4}$$

and suppose we want to find solutions on the interval $0 <= t <= 2$. This equation is the specific case of equation 1 where $f(x,t) = -2x, x_0 = 16$, and $t_0 = 0$. If we chose $h = 0.1$, much too large a value for h for accuracy but good for purposes of illustration, we would choose $N = 2/h = 20$ so that $t_N = 2$. The forward difference scheme 3 becomes

$$x_{n+1} = x_n + -2x_n h$$

and if we were to do our computations by hand, we could arrange then in tabular form this way,

n	t_n	x_n	$-2x_n h$	$x_n + 2x_n h = x_{n+1}$
0	0.0	16.00	-3.20	$16.00 - 3.20 = 12.80$
1	0.1	12.80	-2.56	$12.80 - 2.56 = 10.24$
2	0.2	10.24	-2.048	$10.24 - 2.048 = 8.192$
3	0.3	8.192		

where the table is continued up to $N = 20$. The last value of t_n is $t_N = 2$, and the last value of x_n, that is x_N, is $x(2) = 0.1845$. This is rather far from the correct value of $x(2)$ which is $x(2) = 0.2931$ to four decimal places.

To illustrate the effect of decreasing the size of h on accuracy, here are some values for $x(2)$ obtained by computer using this scheme for various values of h,

h	$x(2)$ approximate	$x(2)$ exact
10^{-1}	0.18447	0.29305
10^{-2}	0.28141	
10^{-3}	0.29188	
10^{-4}	0.29293	
10^{-5}	0.29304	

For this particular differential equation, decreasing the value of h by a factor of 10 gives us one additional decimal digit of accuracy, so it takes $h = 10^{-4}$ to give us 4 decimal digits of accuracy for the value of $x(2)$.

From this, you might conclude that smaller is always better as far as h is concerned. Within resonable limits, this is true, but it is certainly not true if carried to extremes. The reason is *roundoff error*. The computer program we used to obtain the approximate values for $x(2)$ listed above does its computations using double precision floating point numbers. Each arithmetic calculation is accurate to approximately 16 decimal places. This may seem like a lot of accuracy, but if we

multiply two 16 digit numbers together, we get a number approximately $2 \cdot 16 = 32$ decimal digits in length. (You can verify this for yourself by multiplying two 2 digit numbers together and observing that you typically get 4 digit numbers. For example, $40 \cdot 40 = 1600$.) There is no room to store the extra 16 digits, so we have to round off the 32 digit answer to 16 digits of accuracy. Each time we round off, we introduce error. This error is called *roundoff error*. The roundoff error for an individual calculation is insignificant, but it grows as the number of calculations grows.

The importance of the phenomenon of roundoff error has been understood since the beginning days of digital computation. In their famous memorandum from 1946 where they first outlined the design of a stored program digital computer, Burks, Goldstein and Von Neumann discussed roundoff error and concluded, rather pessimistically, that there was no really good way to analyse or control roundoff error. The best one could do was subject it to a probabilistic analysis. About half the time, roundoff error decreases the answer, the other half the time it increases it. Hence roundoff error behaves like a random walk, and it grows as $\sqrt{N}$, where N is the number of computations performed. Even today, we can't do much better, so you should assume that the roundoff error in your calculations grows as $\sqrt{N}$.

To return to the effect of making the increment h too small, remember that N is inversely proportional to h. (In our example, $N = 2/h$ because the length of the interval was 2.) Now if you divide by a very small number, you get a very large number. For example, when h was $0.1 = 10^{-1}$, we found that N was 20, but when h is $0.00001 = 10^{-5}$, we can see that N will be $100,000 = 10^5$. Even 100,000 calculations will not produce appreciable roundoff error in double precision arithmetic, but billions or trillions of calculations will. So if you make h too small, roundoff error will totally destroy the precision of your calculations, to the extent that the numbers you get will be meaningless.

Next, let us answer a question you probably have already asked. How did we obtain the exact value for $x(2)$? The answer is that we solved the differential equation 4 analytically by a technique known as "separation of variables". Here are the details.

$$\begin{aligned} dx/dt &= -2x \quad \text{(equation 4 repeated)} \\ dx &= -2x \cdot dt \\ dx/x &= -2 \cdot dt \\ \ln x &= -2t + c \\ x &= e^{-2t}e^c \\ x &= Ae^{-2t}, \end{aligned}$$

where we have replaced e^c by the constant A. In summary, the general analytic solution to equation 4 is

$$x = Ae^{-2t} \tag{5}$$

Now we substitute $t = 0$ into the general solution, equation 5, and use the initial condition, $x(0) = 16$ to obtain

$$A = Ae^0 = 16$$

Thus the analytic solution to equation 4 is

$$x = 16e^{-2t} \tag{6}$$

.

Equation 4 is a special case of the equation of radioactive decay which is usually covered in calculus courses under the topic title, "Exponential growth and decay", so you have probably seen this before.

We shall close this section with a bit of Java code. Here is a Java method (the Java equivalent of a C or Pascal function) for solving equation 4 numerically on the interval $0 <= t <= 2$ by forward differences. The method takes as inputs the initial values of t and x, the final value of t, and the value of h, and it returns an array holding the approximate solution values $x_0, x_1, \ldots, x_n$.

```
// return an array of approximate numerical values of the solution
public static double[ ] solveIt(double t0, double x0,
                                double tN, double h) {
   double[ ] x;  int N;

   N = (int)((tN - t0)/h);
   x = new double[N + 1];  // actually allocate space for the array

   x[0] = x0;  // given
   for (int n = 0; n < N; n++) {
      x[n+1] = x[n] + -2*x[n]*h;
   }

   return x;
} // solveIt
```

You only have to modify one line of this code to solve any other given differential equation by forward differences. The line to be modified, is, of course, the one inside the `for` loop.

Here is a variation on the Java method above which only returns the first and last values of x on the interval instead of all the values $x_0, x_1, \ldots, x_N$. It is useful in cases where the interval is very large or the value of h very small so you don't create too large an array and run out of memory. One thing we can learn from this example is that it is not always necessary to store every computed value in an array. Instead, to conserve memory, we may only save some computed values for later use, such as in a display.

```
// return first and last approximate numerical values of the solution
public static double[ ] solveIt(double t0, double x0,
                                double tN, double h) {
   double[ ] x; double xs; int N;

   N = (int)((tN - t0)/h);
   x = new double[2];  // actually allocate space for the array

   x[0] = x0;  // given
   for (int n = 0; n < N; n++) {
      xs = xs + -2*xs*h;
   }
   x[1] = xs;

   return x;
} // solveIt
```

3. Runge Kutta Methods

The second order Runge Kutta method, or midpoint method, is the simplest numerical method which is accurate enough for practical use. It also has the great virtue of being easy to understand.

The idea behind the second order Runge Kutta method is to use a "trial step" to the midpoint of the interval by forward differences. Then one uses the x and t values at the midpoint to estimate Δx and x across the whole interval. In detail,

$$\begin{aligned} r/2 &= f(x(t),t)h/2 \quad \text{trial step to midpoint by forward differences} \\ \Delta x &= f(x(t)+r/2, t+h/2)h \quad \text{estimate of } \Delta x \text{ across whole interval} \\ x(t+h)-x(t) &= f(x(t)+r/2, t+h/2)h \\ x(t+h) &= x(t)+f(x(t)+r/2, t+h/2)h \end{aligned}$$

In summary, we obtain the approximations

$$\begin{aligned} r &= f(x(t),t)h \\ x(t+h) &= x(t)+f(x(t)+r/2, t+h/2)h \end{aligned} \tag{1}$$

As in the case of forward differences, this is usually written in terms of t_n and x_n as

$$\begin{aligned} r &= f(x_n,t_n)h \\ x_{n+1} &= x_n+f(x_n+r/2, t_n+h/2)h \end{aligned} \tag{2}$$

The equations 2 are often called the *second order Runge Kutta scheme.*

Under reasonable assumptions on the function $f(x,t)$ which are almost always true in practice, the error term in the approximation of x_{n+1} from x_n is $O(h^3)$. For those unfamiliar with O notation, this simply means that the error in the approximation is bounded above by a constant times h^3. Thus the error across the $O(1/h)$ steps in the whole t interval is $O(h^2)$. The "second order" part of the name, "the second order Runge Kutta method", comes from this fact. Of course the "Runge Kutta" part comes from the fact that the method was invented by Runge and Kutta

There is also a fourth order Runge Kutta method. (In fact, there are Runge Kutta methods of arbitrarily high order, but we won't discuss them here.) We shall content ourselves with simply giving the fourth order Runge Kutta scheme,

$$\begin{aligned} r_1 &= f(x_n,t_n)h \\ r_2 &= f(x_n+r_1/2, t_n+h/2)h \\ r_3 &= f(x_n+r_2/2, t_n+h/2)h \\ r_4 &= f(x_n+r_3, t_n+h)h \\ x_{n+1} &= x_n+r_1/6+r_2/3+r_3/3+r_4/6 \end{aligned} \tag{3}$$

Under reasonable assumptions on the function $f(x,t)$ which are almost always true in practice, the approximation of x_{n+1} in terms of x_n is accurate to within a $O(h^5)$ error term, and the error across the $O(1/h)$ steps in the whole t interval is $O(h^4)$; hence the name "fourth order Runge Kutta method".