CALCODE

Contents

Notes

[1]J. L. Synge and B. L. Griffith, *Principles of Mechanics*, 3rd ed., McGraw Hill, New York, 1959, pp. 141–146.

[2]Synge and Griffith, cited above, p. 334.

[3]e.g. Synge and Griffith, cited above, p. 330.

[4]F. P. Byrd, *Handbook of Elliptic Integrals for Engineers and Scientists*, Springer, Berlin, 1971.

[5]For a more detailed account of this episode, see my article, "Time: what is it that it can be measured?" *Science and Education*, **15**(6) (2006) DOI: 10.1007/s11191-005-5287-z.

[6]E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations*, Springer, Berlin, 1991.

$$f(x) = f_1(x) * (x < 0) \quad (40)$$
$$+ f_2(x) * ((x > 0)\&(x < 2))$$
$$+ f_3(x) * (x > 2).$$

Note that $>$ stands for $\geq$, while $<$ stands for strict inequality, so that both sorts of inequalities can be handled.

4. **Function names**. abs, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, exp, ln, log, sqrt. Names are case insensitive. For the power function, see arithmetic symbols.

5. **Variable names**. The "independent" variable is denoted by x. The "dependent" variables are denoted by y1, y2, y3, ..., or by y[1], y[2], y[3], ... (case insensitive). Parameters are denoted by k1, k2, k3, ..., or by k[1], k[2], k[3] .... Omitted y (or k) index defaults to y1 (respectively k1).

6. **Numbers**. When defining the RHS of equations, numbers MUST be entered in mathematical notation: 10∧−6 and NOT 1e-6. Exactly the reverse holds good for numbers entered elsewhere. Thus, permissible error = 1.0E-4 and NOT

Introduction

What is CALCODE?

CALCODE is an educational toy that I developed for/with my children to teach them physics without fearing the calculus.

CALCODE makes ordinary differential equations (ODEs) as easy as arithmetic with a calculator. Anyone who can set up an ODE can immediately see the solution (animated in 3 dimensions and from any perspective).

Who will find CALCODE useful?

Newtonian physics is all about solving ordinary differential equations. But, because ODEs are regarded as difficult they are avoided in elementary school (K-12) courses in Newtonian physics (and calculus). Even many undergraduate courses stop with linear ODEs.

This results in various problems. For example, a cricket ball is heavier than a tennis ball. So, according to school-level physics, one should be able to throw the tennis ball farther. But this directly contradicts the child's experience.

Similarly, the motion of the simple pendulum is often confounded with simple harmonic motion—just because the real motion of the simple pendulum is described by nonlinear ODEs regarded as difficult. Hence, they are replaced by simpler linear ODEs, according to which the time period is independent of the amplitude. But the variation in the time period of a simple pendulum with changes in its amplitude can be observed, and this again contradicts the child's experience.

It is a sad sort of science which is taught in a way divorced from experience—just because of a perceived difficulty in teaching the underlying mathematics related to the calculus. As CALCODE demonstrates, that difficulty no longer exists.

Hence, I expect that students and teachers of calculus and physics at school (K-12) and college level will find CALCODE useful.

CALCODE has other useful features: it accepts ODEs defined symbolically. It outputs the solution numerically and connects this to 2d graphs. It enables phase plots and comparisons with various functions. It also enables easy study of how the solutions and their properties vary with parameters.

All this enables one to teach realistic physics without fearing the difficulty of the underlying mathematics.

Symbolic equation syntax

The right hand sides of the ODEs must be defined symbolically, using a restricted set of symbols. Available symbols are from the following.

1. **Parentheses**. (), {}, [ ]. Curly brackets may be used interchangeably (in pairs) with ordinary brackets, for visual relief. Square brackets are reserved for arrays.

2. **Arithmetic symbols**. $\wedge$, * and /, + and $-$, in that order of precedence.

3. **Logical symbols**. $<$, $>$, | (or), & (and). These are provided for conditional function definitions. E.g. to define

$$f(x) = \begin{cases} f_1(x) & : \quad x < 0 \\ f_2(x) & : \quad 0 \leq x < 2 \\ f_3(x) & : \quad x \geq 2 \end{cases}$$

use

the splash screen for CALCODE. Such a smooth plot requires a large number of interpolation points.

Brusselator

The name derives from the Brussels school of Prigongine et al. The system was proposed as a model for the mythical emergence of order from chaos.

The system models a chaotic chemical reaction involving two substances coloured red and blue, say. Instead of a uniform indigo mixture, one gets a chemical clock which suddenly switches colour from red to blue and back: the density of chemicals in the reaction oscillates.

The system has become a well-known test problem for numerical ODEs.[6] The system is given by

$$y_1' = A + y_1^2 y_2 - (B+1)y_1 \tag{38}$$

$$y_2' = By_1 - y_1^2 y_2. \tag{39}$$

Typical values for the reaction rates A and B are $A = 2$, $B = 8.533$. Typical initial data are $y_1(0) = 1$ and $y_2(0) = 4.2665$.

Other examples are included in the example files.

Getting started

Simple harmonic motion

As a first example, consider simple harmonic motion, described by the ODEs

$$\frac{d^2y}{dx^2} = -k^2 y. \tag{1}$$

Using primes to denote derivatives, one could write this more compactly as

$$y'' = -k^2 y. \tag{2}$$

The standard form

CALCODE uses the equivalent formulation of the above equation as a system of two first order equations. This is the "standard form". In practice, this amounts to little more than a change of notation. The primary variable y is denoted by y_1 (entered into the computer as plain y1). The first derivative y' is denoted by y_2 (written as y2). The above equations are then equivalent to the two simultaneous equations

$$y_1' = y_2, \quad (3)$$
$$y_2' = -k^2 y_1. \quad (4)$$

The right hand sides (RHS) of these equations are described symbolically to CALCODE as respectively y2 and -k*k*y1.

The range

Next, one must specify a range, such as $[0, 20]$ over which the equations are to be solved.

Initial data

Then, one needs to specify the initial data. For the equation (2) this require us to specify the values of y and y' at the starting point:

$$y(0) = 0, \quad (5)$$
$$y'(0) = 1.$$

Recall that, in our notation, y is written as y_1, and y' is written as y_2, so this corresponds to specifying the values

$$y_1(0) = 0, \quad (6)$$
$$y_2(0) = 1.$$

Finally, one must fix a numerical value for the parameter k. (There can, of course, be more than one parameter in an equation.)

Van der Pol's equation

CALCODE can solve a variety of other non-linear ODEs. One such ODE is the van der Pol equation. To the equation of simple harmonic motion, van der Pol added *ad hoc* a term which gives damping when the amplitude is large, and anti-damping when it is small to obtain:

$$y'' + \epsilon(y^2 - 1)y' + k^2 y = 0, \quad (32)$$

where $\epsilon > 0$. Small oscillations are amplified, while large oscillations are damped. Putting $k = 1$ we get two equations in standard form

$$y_1' = y_2 \quad (33)$$
$$y_2' = \epsilon(1 - y_1^2)y_2 + y_1. \quad (34)$$

Lorenz model

The system of 3 ODEs defining the Lorenz model is already in standard form and is given by

$$y_1' = -\sigma y_1 + \sigma y_2, \quad (35)$$
$$y_2' = -y_1 y_3 + r y_1 - y_2, \quad (36)$$
$$y_3' = y_1 y_2 - b y_3. \quad (37)$$

Saltzman's values for the parameters are $b = \frac{8}{3}$, $\sigma = 10$, $r = 28$, and the initial data are $y_1 = 8$, $y_2 = -8$, $y_3 = 27$ for a typical range such as $[-2, 2]$. The resulting phase plot (the Lorenz "butterfly") obtained using CALCODEis used as

$$1 - y^2 = \mathrm{cn}(x) = y_2 \quad (27)$$
$$1 - k^2 y^2 = \mathrm{dn}(x) = y_3. \quad (28)$$

From this, one immediately obtains a system of 3 equations in standard form for the Jacobian elliptic functions sn, cn, and dn.

$$y_1' = y_2 y_3, \quad (29)$$
$$y_2' = -y_3 y_1, \quad (30)$$
$$y_3' = -k^2 y_1 y_2, \quad (31)$$

with usual initial data being $y_1(0) = 0$, $y_2(0) = 1$, $y_3(0) = 1$.

The "Compare" feature of CALCODE enables one to compare the solution with the sin function that one gets in the case of simple harmonic motion.

If we do not confound the simple pendulum with simple harmonic oscillation, there is no simple formula for the time period of the simple pendulum. Calculating the time period requires the computation of non-elementary elliptic integrals.[3] This once was (at least until the 1970's) an appropriate subject matter for a scholarly handbook.[4] Accordingly, this was a left out of most elementary texts as a topic fit for graduate students.

However, the "Sections" feature of CALCODE enables any schoolchild to compute zeros of the solution: the Jacobian elliptic function sn. One can now use the "Repeat" feature to determine how the zeros change with the amplitude of the pendulum, for instance, or with damping.

For a documented project in this direction done by a school child see the project at `http://11picsoftime.com/pendulum.pdf`.[5]

The Wizard

The CALCODE Wizard conducts you step by step through these tasks, and also takes down some more preferences. (For instructions on how to use the Wizard, start CALCODE and click the "Getting Started" book in the CALCODE help contents or see the user guide.)

Output and visualisation

After solving the equation, CALCODE displays the solution in 3 different ways.

1. As numbers in report view.
2. As 2d graphs in graph view.
3. As animated particle motion in 3d view.

The CALCODE analysis menu provides various ways to study the solution. It can be compared with, say, the function $\sin(x)$. One can repeat the solution with a changed value of the parameter k, which specifies the stiffness of the spring. The 3d animation helps to see exactly what is meant by saying that the frequency of oscillation increases with the stiffness of the spring.

Damping

As a prelude to more advanced problems it is a good idea to teach/study the case of damped harmonic motion. In simple harmonic motion the force is proportional and opposite to

the displacement (y or y1). In the damped case, there is an additional force proportional to the velocity (y' or y2), and acting in the opposite direction. So this force corresponds to a term of $-k_2 y'$ (or -k2*y2 in our notation). The two forces are to be added. Thus, the ODEs for damped harmonic motion can be written as

$$y_1' = y_2, \quad (7)$$
$$y_2' = -k_1^2 y_1 - k_2 * y_2. \quad (8)$$

We do not force the second parameter k_2 to be positive, since one may be interested also in anti-damping. The equation can be solved and visualised exactly as in the above case.

once the equations have been entered, one can play with them in a variety of cases. Although the cricket ball goes further, the shot put does not, so there is no simple rule, except on the moon where what goes furthest is the balloon.

Again, one may want to know the optimum angle of throw. To determine this, one uses the "Repeat" feature of CALCODE to repeat the above solution for different initial data $\dot{x}(0) = v_0 \cos\theta$, $\dot{y}0 = v_0 \sin\theta$ for different angles of throw θ. In each case, the zeros of y1 can be calculated (using "Sections" in the Analyse menu) and the resulting data can be stored and plotted. Incidentally, the optimum angle of throw works out to be quite different for a javelin compared to a cannon ball (and neither is 45°)!

Different models of air resistance can be tried out: even conditional function definitions are possible (see the section on equation syntax), so that we may use a model which switches from linear to quadratic air resistance at higher velocities.

The simple pendulum

The equation of motion of a simple pendulum is[2]

$$y'^2 = (1 - y^2)(1 - k^2 y^2). \quad (25)$$

This form of the equations is not suited to numerical calculations because as y crosses 1, y' must turn negative. The sign of the equation is more readily specified by putting the equations in Jacobi's form

$$y = \mathrm{sn}(x) = y_1 \quad (26)$$

jectiles to projectiles with air resistance exactly as we moved from simple harmonic motion to damped harmonic motion.

Specifically, the total force $\boldsymbol{F}$ on the projectile can be regarded as the sum of two forces.

$$\boldsymbol{F} = \boldsymbol{F}_1 + \boldsymbol{F}_2 \tag{19}$$

where $\boldsymbol{F}_1 = m(0, -g)$ is just the earlier force due to gravity, and $\boldsymbol{F}_2$ is the new force due to air resistance. As in the case of damped harmonic motion, let us take this force to act in the opposite direction to velocity. Further, let us suppose its magnitude depends linearly or quadratically (or cubically) on the velocity. That is,

$$\boldsymbol{F}_2 = -k_1 v^{k_2} \hat{\boldsymbol{v}} \tag{20}$$

Here the magnitude of the velocity $v = \sqrt{\dot{x}^2 + \dot{y}^2}$, and $\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{v}$ is a unit vector in the direction of $\boldsymbol{v}$, and k_1 and k_2 are parameters which we can vary to suit the exact model of air resistance that is used.

In CALCODE notation, the projectile equations can now be modified to

$$\begin{aligned}
y_1' &= y_3, && (21)\\
y_2' &= y_4, && (22)\\
y_3' &= -\frac{k1}{m} * \left(y_3^2 + y_4^2\right)^{\frac{k_2-1}{2}}, && (23)\\
y_4' &= -\frac{k1}{m} * \left(y_3^2 + y_4^2\right)^{\frac{k_2-1}{2}} - g. && (24)
\end{aligned}$$

The equations look quite horrible, but there is now no difficulty in understanding their origin or in solving them. Also,

Examples

We can now move on to more serious examples.

Projectile motion

Projectile motion is often taught in schools just as a Galilean parabola. We can however proceed as follows. Let the position of the particle (mass point) be given by the vector $\boldsymbol{r} = (x, y)$ where x and y are the x and y coordinates of the particle at any time. Let its velocity be $\boldsymbol{v} = (\dot{x}, \dot{y})$ and its acceleration $\boldsymbol{a} = (\ddot{x}, \ddot{y})$, where dots denote derivatives with respect to time. The only force acting on the particle is that of gravitation. Taking the y direction to be pointing upwards, this force is given by $\boldsymbol{F} = m(0, -g)$, where m is the mass of the particle, and g is the acceleration due to gravity. By Newton's second "law" of motion, we have $\boldsymbol{F} = m\boldsymbol{a}$, which corresponds to the following two second order differential equations.

$$\begin{aligned}
\ddot{x} &= 0, && (9)\\
\ddot{y} &= -g. && (10)
\end{aligned}$$

To convert these to the standard form used by CALCODE, we need the following change of notation.

- y_1 (y1) denotes the x coordinate
- y_2 (y2) denotes the y coordinate
- y_3 (y3) denotes $\dot{x}$
- y_4 (y4) denotes $\dot{y}$

(There is some merit to this numbering scheme which is assumed by convention in 3d animation.) With this change of notation, the 4 equations in standard form can be written as

$$\begin{aligned} y_1' &= y_3, && (11)\\ y_2' &= y_4, && (12)\\ y_3' &= 0, && (13)\\ y_4' &= -g. && (14) \end{aligned}$$

The quantity g should be replaced by its numerical value in the units being used. The (implicit) time variable is the dummy variable denoted by x in CALCODE.

The graphs of the solution, and the resulting animation help to understand what a parabola is. (Since we have specified equations for only two physical coordinates x and y of the particle, in the 3d animation dialogue we must specify 2 dimensions and 1 particle.)

Let us now see how this can be used to study the question of why the cricket ball can be thrown farther than a lighter tennis ball.

Ballistics with air resistance

The traditional treatment of projectile motion with air resistance is complex and counter-intuitive, even in a good text on classical mechanics like that of Synge and Griffith.[1] The usual equations (in CALCODE notation) are:

$$\begin{aligned} y_1' &= y_1 \left\{ \frac{\phi(y_1) + \sin x}{\cos x} \right\}, && (15)\\ y_2' &= -\frac{y_1^2}{g}, && (16)\\ y_3' &= -\frac{y_1^2 \tan x}{g}, && (17)\\ y_4' &= -\frac{y_1 \sec x}{g}. && (18) \end{aligned}$$

Here, y_1 is the velocity, x is the angle in radians measured along the trajectory, $\phi(y_1)$ is a function (linear or quadratic) which models the dependence of air-resistance on velocity, y_2 is the horizontal displacement, y_3 is the height or vertical displacement, and y_4 is the time elapsed since the throw.

CALCODE can, of course, solve these equations. (See the example on "Ballistics with Air Resistance (Polar Coordinates.ode"; g should be replaced by its numerical value in the units being used.) But the derivation of these equations is complex for a school child.

However, let us recognize that the equations were formulated in the above fashion just to ease the difficulty in solving them—assuming older symbolic methods of solution—which difficulty no longer exists. Therefore, we can move from pro-