Examples of Initial-Value ODE Problems

By Gilberto E. Urroz, September 2004

The following are examples of initial-value ODE problems (IVP) solved using Matlab predefined functions *ode23* and *ode45*.

Example 1 – Free-falling sphere in a fluid

- Script: *Falling_Sphere_Script.m*
- Functions: *vel.m*, *acc.m*

A sphere of diameter D and density ρ_s falling in a fluid of density ρ_w is subject to its own weight, $W = \rho_s gV$, which acts as a driving force, and to a buoyancy force $B = \rho_w gV$, which acts as a resisting force. The variable V represents the volume of the sphere, namely, $V = 4/3\pi D^3$, while g is the acceleration of gravity. The sphere is also subjected to a resisting viscous force, $F_v = kv$, where v is the sphere velocity. The mass of the sphere is given by $m = \rho_s V$. With the acceleration given by $a = dv/dt$, Newton's second law can be written as:

$$\rho_s gV - \rho_w gV - kv = \rho_s V (dv/dt).$$

Thus, in terms of the velocity, the governing ODE can be written as:

$$dv/dt = g(1-\rho_w/\rho_s) - (k/(\rho_s V))\, v.$$

In terms of the position, x, the governing equation becomes:

$$d^2x/dt^2 = g(1-\rho_w/\rho_s) - (k/(\rho_s V))\,(dx/dt).$$

The latter ODE can be transformed into a system of first order equations by taking $u_1 = x$, $u_2 = dx/dt = du_1/dt$, i.e.,

$$du_1/dt = u_2$$
$$du_2/dt = g(1-\rho_w/\rho_s) - (k/(\rho_s V))\, u_2.$$

(a) Use the following values of the parameters: $\rho_w = 1000\ kg/m^3$, $\rho_s = 2500\ kg/m^3$, $g = 9.806\ m/$, $D = 0.05\ m$, $k = 1.0\ N\cdot s/m$, and solve for the velocity v as a function of time t in the interval $0 < t < 30\ s$. (b) Also, solve the system of equations in $u_1(t)$ and $u_2(t)$ so as to calculate the position x as a function of t.

Example 2 – Lotka-Volterra equations

- Script: *LotkaVolterra_Script.m*
- Function: *fLV.m*

From *Heath, M.T.*, 1997, *Scientific Computing – An Introductory Survey*, WCB McGraw-Hill, New York – Problem 9.1:

The populations of two species, a prey denoted by y_1 and predator denoted by y_2, can be modeled by a system of ODEs:

$$y_1' = by_1 - cy_1y_2$$

$$y_2' = -dy_2 + cy_1y_2$$

due to Lotka and Volterra. The parameters *b* and *d* govern the birth rate of prey and death rate of predators, respectively, and the parameter *c* governs the interaction of the two populations. With the parameter values $b = 1$, $d = 10$, and $c = 1$, and initial conditions $y_1(0) = 0.5$ and $y_2(0) = 1$ (the populations are normalized, and we treat them as continuous variables), use Matlab function *ode45* to solve this system numerically, integrating to $t = 10$. Plot each of the two populations as a function of time, and on a separate graph plot the trajectory of the point $(y_1(t),y_2(t))$ in the plane as function of time. The latter is sometimes called "a phase portrait." Give a physical interpretation of the behavior you observe. Can you find nonzero initial populations such that either of the populations eventually become extinct?

Example 3 – Lorentz equations

- Script: *Lorentz_Script.m*
- Function: *fLorentz.m*

The Lorentz equations are used to simulate the convection of a layer of fluid of infinite horizontal extent heated from below. The model is a simplified version of the heating of the atmosphere. The equations are obtained by expanding the terms for temperature and pressure involved in the problem with their Fourier series expansion and simplifying the expansion to the first three modes represented by the variables *x, y,* and *z*. The resulting system of equations is

$$dx/dt = \sigma(-x+y)$$
$$dy/dt = rx - y - xz$$
$$dz/dt = xy - bz$$

where σ, r, and b are constants that result from combining physical parameters of the problem. (For a detailed derivation refer to Berge, P., Y. Pomeau, and C. Vidal, 1984, "*Order within Chaos - Towards a deterministic approach to turbulence,*" John Wiley & Sons, New York).

Solve the Lorentz equations for the following combination of parameters:

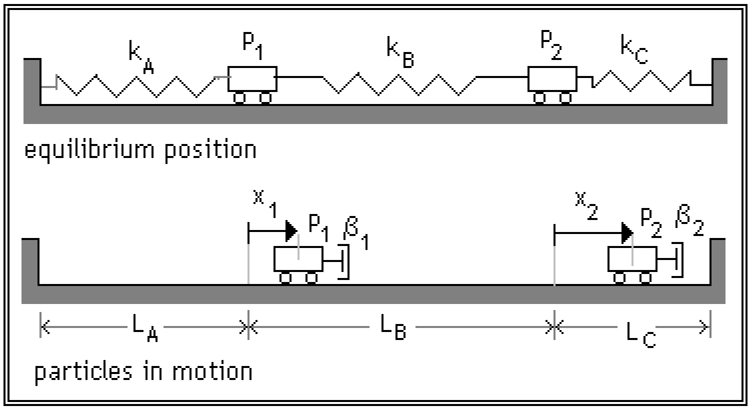
(a) $\sigma = 10$, $r = 25$, $b = 2.666$, $x_0 = 1$, $y_0 = 1$, $z_0 = 1$
(b) $\sigma = 10$, $r = 75$, $b = 2.666$, $x_0 = 1$, $y_0 = 1$, $z_0 = 1$
(c) $\sigma = 10$, $r = 25$, $b = 2.666$, $x_0 = 1$, $y_0 = 1$, $z_0 = 1$
(d) $\sigma = 10$, $r = 25$, $b = 2.666$, $x_0 = 1$, $y_0 = 1$, $z_0 = 1$

Plot the signals *x-vs-t, y-vs-t, z-vs-t*, as well as the phase portraits *x-vs-y*, *x-vs-z*, and *y-vs-z*.

Example 4 – Two masses and three springs

- Script: *Lorentz_Script.m*
- Function: *fLorentz.m*

The figure below shows two particles P_1 and P_2, of mass m_1 and m_2, respectively, linked by three springs (k_A, k_B, k_C). The figure at the top represents the system in their state of equilibrium, while the one at the bottom shows the system at any generic point at time t>0. The displacement of particle P1 with respect to its equilibrium position is given by $x_1(t)$ while that of particle P2 is given by $x_2(t)$. The corresponding velocities are $v_1 = dx_1/dt$ and $v_2 = dx_2/dt$. The magnitude of the forces applied by the springs on the particles are given by Hooke's law, $F = k(L-L_0)$, where L is the stretched length of the spring, L_0 is the unstretched length of the spring, and k is the spring constant. The particles are also provided by dashpots that produce a viscous damping force whose magnitude is given by $F = \beta v$, where β is a damping constant and v is the speed of the particle, i.e., $F = \beta(dx/dt)$, where x = position of the particle.



(a) Write down the differential equations describing the motion of the two linked particles including spring and damping forces as shown in the figure above.

(b) Solve for $x_1(t)$ and $x_2(t)$ if $m_1 = 10$ kg, $m_2 = 20$ kg, $k_A = 80$ N/m, $k_B = 120$ N/m, $k_C = 100$ N/m, $\beta_1 = 1$ N·s/m, $\beta_2 = 5$ N·s/m. The initial conditions are given by $x_1(0) = 0.5$ m, $x_2(0) = 0.25$ m, $v_1(0) = 0$, $v_2(0) = 1.0$ m/s. Plot the signals and the velocity versus time for $0 < t < 5$ s.

(c) Solve for $x_1(t)$ and $x_2(t)$ if $m_1 = 10$ kg, $m_2 = 20$ kg, $k_A = 80$ N/m, $k_B = 120$ N/m, $k_C = 100$ N/m, $\beta_1 = 0$, for values of β_2 = 0, 0.1, 0.5, 1, and 5 N·s/m. The initial conditions are given by $x_1(0) = 0.5$ m, $x_2(0) = 0.25$ m, $v_1(0) = 0$, $v_2(0) = 1.0$ m/s. Plot the signals and the velocity versus time for $0 < t < 5$ s for the different values of β_2.

(d) Write the differential equations for $x_1(t)$ and $x_2(t)$ if an external force $F_1 = F_0 \sin(\omega_0 t + \phi_0)$ is applied to particle P_1 in addition to the spring and damping forces.

(e) Using the conditions of part (a) of this problem solve for $x_1(t)$ and $x_2(t)$ for the case in which particle P_1 is subject to the external force $F_1 = F_0 \sin(\omega_0 t + \phi_0)$ with $F_0 = 20$ N, $\omega_0 = 2.5$ rad/s, and $\phi_0 = 1.5$ rad. Plot the signals, velocity, and acceleration versus time for $0 < t < 5$ s for the different values of β_2.

Example 5 – Two bodies under mutual gravitational attraction

- Script: *TwoBodies_Script.m*
- Function: *fTBz.m*

From *Heath, M.T.*, 1997, *Scientific Computing – An Introductory Survey*, WCB McGraw-Hill, New York – Problem 9.7:

An important problem in classical mechanics is the motion of two bodies under mutual gravitational attraction. Suppose that a body of mass *m* is orbiting a second body of much larger mass *M*, such as the earth orbiting the sun. From Newton's law of motion and gravitation, the orbital trajectory $(x(t),y(t))$ is described by the system of second-order ODEs

$$x'' = -GMx/r^3$$

$$y'' = -GMy/r^3$$

where G is the gravitational constant and $r = (x^2+y^2)^{1/2}$ is the distance of the orbiting body from the center of mass of the two bodies. For this exercise, we choose units such that $GM = 1$.

(a) Use Matlab function *ode45* to solve this system of ODEs with the initial conditions,

$$x(0) = 1 - e,\ y(0) = 0,\ x'(0) = 0,\ y'(0) = ((1+e)(1-e))^{1/2}$$

where *e* is the eccentricity of the resulting elliptical orbit, which has period 2π. Try the values *e = 0* (which should give a circular orbit), *e = 0.5*, and *e = 0.9*. For each case, solve the ODE for at least one period and obtain output at enough intermediate points to draw a smooth plot of the orbital trajectory. Make separate plots of *x* vs. *t*, *y* vs. *t*, and *y* vs. *x*. Experiment with different error tolerances to see how they affect the cost of the integration and how close the orbit comes to being closed. If you trace the trajectory through several periods, does the orbit tend to wander or remain steady?

(b) Check your numerical solutions in part (a) to see how well they conserve the following quantities, which should remain constant:

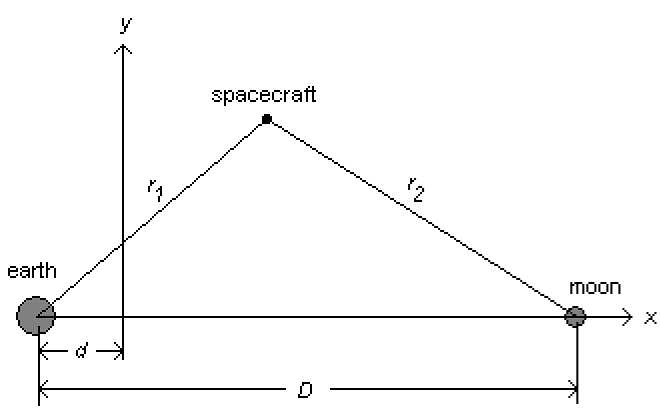
- Conservation of energy, $((x')^2+(y')^2)/2 - 1/r$
- Conservation of angular momentum, $xy'-yx'$

Example 6 – Three bodies in gravitational field – one body smaller than the other two

- Script: *ThreeBody_Script.m*
- Function: *f3B.m*

From *Heath, M.T.*, 1997, *Scientific Computing – An Introductory Survey*, WCB McGraw-Hill, New York – Problem 9.8:

Consider a restricted form of the three-body problem in which a body of small mass orbits two other bodies with much larger masses, such as an Apollo spacecraft orbiting the earth-moon system. We will use a two-dimensional coordinate system in the plane determined by the three bodies, with the origin at the center of mass of the two larger bodies, and the coordinate system rotating so that the two larger bodies appear fixed. The coordinate system is shown in the accompanying diagram, where *D* is the distance from earth to moon, *d* is the distance from the center of earth to the center of mass, r_1 is the distance from earth to spacecraft, and r_2 is the distance from moon to spacecraft. The mass of the spacecraft is assumed to be negligible compared to the other masses.



By using Newton's laws of motion and gravitation, and allowing for the centrifugal and Coriolis forces due to the rotating coordinate system, the motion of the spacecraft is described by the system of second-order ODEs:

$$x'' = - G[M(x+\mu D)/r_1^3 + m(x-\mu^* D)/r_2^3] + \Omega^2 x + 2\Omega y',$$

$$y'' = -G[My/r_1^3 + my/r_2^3] + \Omega^2 y - 2\Omega x',$$

where G is the gravitational constant, M and m are the masses of earth and moon, μ^* and μ are the mass fractions of earth and moon, and Ω is the angular velocity of rotation of the moon about the earth (and hence of the coordinate system). The numerical values of these quantities are given in the following table:

G	$6.67259 \times 10^{-11}\ m^3/(kg\ s^2)$
M	$5.974 \times 10^{24}\ kg$
M	$7.348 \times 10^{22}\ kg$
μ^*	$M/(m+M)$
μ	$m/(m+M)$
D	$3.844 \times 10^{8}\ m$
d	$4.669 \times 10^{6}\ m$
r_1	$[(x+d)^2+y^2]^{1/2}$
r_2	$[(D-d-x)^2+y^2]^{1/2}$
Ω	$2.661 \times 10^{-6}/s$

Use Matlab function *ode45* to solve this system of ODEs with the initial conditions

$$x(0) = 4.613 \times 10^8,\ y(0) = 0,\ x'(0) = 0,\ y'(0) = -1074.$$

Plot the resulting solution trajectory ($x(t),y(t)$) in the plane as a function of time. Indicate the positions of earth and moon on the graph. Compute the solution for at least one complete orbit (i.e., until the spacecraft returns to its original location), which is from $t = 0$ until approximately $t = 2.4 \times 10^6\ s$. Experiment with various error tolerances to see how much difference they make in whether the orbit is actually closed. Try to monitor the step size used by the ODE routine as the integration progresses. When does the step size become smaller or larger? How close does the spacecraft come to the surface of the earth? (Earth's radius is $6.378 \times 10^6\ m$, so the center of mass of the earth-moon system is actually *inside* the earth.)