Spreadsheets in Education (eJSiE)

Volume 3 | Issue 2 Article 3

10-9-2009

Simulating Projectile Motion in the Air with Spreadsheets

Jan Benacka
Constantine the Philosopher University, Nitra, Slovakia, jbenacka@ukf.sk

Follow this and additional works at: http://epublications.bond.edu.au/ejsie

Recommended Citation
Benacka, Jan (2009) "Simulating Projectile Motion in the Air with Spreadsheets," *Spreadsheets in Education (eJSiE)*: Vol. 3: Iss. 2, Article 3.
Available at: http://epublications.bond.edu.au/ejsie/vol3/iss2/3

This In the Classroom Article is brought to you by the Faculty of Business at ePublications@bond. It has been accepted for inclusion in Spreadsheets in Education (eJSiE) by an authorized administrator of ePublications@bond. For more information, please contact Bond University's Repository Coordinator.

Simulating Projectile Motion in the Air with Spreadsheets

Abstract
The paper gives two examples of simulating motion with spreadsheets, which are projectile motion in vacuum, and projectile motion in the air with quadratic drag.

Keywords
spreadsheet simulation, projectile motion, quadratic drag

This in the classroom article is available in Spreadsheets in Education (eJSiE): http://epublications.bond.edu.au/ejsie/vol3/iss2/3

1. Introduction

It is well known that the trajectory of a projectile in a vacuum is a parabola and the maximum range is reached with launch angle 45°. In the air, resistance force (drag) acts, and the trajectory is a ballistic curve. If the projectile is moving slowly, the drag is proportional to the velocity (linear drag). In this case, the equation of motion is linear and solvable analytically [1]. At higher velocities, provided that the projectile speed does not exceed 270 m/s which is 80% of the sonic speed [2], the magnitude of the drag is proportional to the speed squared [3]

$$\mathbf{F}_\mathrm{D} = 0.5\, CS\rho_\mathrm{a} v^2 \mathbf{v}^0 , \tag{1}$$

where C is the drag coefficient of the body [4], S is the maximum cross-section area of the body perpendicular to the velocity vector, ρ_a is the air density, v is the speed, and $\mathbf{v}^0$ is the velocity unit vector. This drag formula applies to football, cricket, golf, and other balls, as well as stones, darts, arrows, and other projectiles used in various mechanical shooting weapons [5].

The problem of projectile motion with quadratic drag was analytically solved first by Johann Bernoulli in 1711 [6]. The solution comprises definite integrals that require numerical integration performed by a computer program.

Neuwirth and Arganbright in [7] present an interactive spreadsheet model of projectile motion with quadratic drag that is based upon solving the governing differential equations by Euler's method. The great advantage of the model is that it uses standard spreadsheet functions only, that is, no programming skills are required to build it up. The model enables the user to experiment with the inputs and make investigations into the shape of the trajectory, and the horizontal and vertical ranges. A model for motion in a vacuum is also presented.

Neuwirth and Arganbright in [7] and [8] present two methods of moving a point along a curve. The point is a ball-shaped one-point xy graph. In the first method, the coordinates of the point are returned by the OFFSET function from the numbered list of the points of the graph. The ordinal number of the point is taken from a cell that is linked with a scrollbar. Hence, the value of the scrollbar determines the position of the point on the curve. The second method uses a macro with two loops. The first one goes over the list of graph points. The second one sums numbers from 1 to 100000, and it is embedded into the first loop to slow down the calculation and make the motion of the ball visible. Wischniewsky in [9] presents another method of moving a point along a graph where the iteration, that is, getting the ball into the next position, is governed by a circular reference, and the automatic regime is performed by a macro.

In this paper, a variant of the model in [7] is presented that simulates the projectile motion. At the simulation, another method of moving a point along a graph is used. The model works with spherical projectiles of various smoothness, radius, and material, and it allows investigation into the effect of the inputs on the motion of the projectile. The trajectory is graphed using Euler's method over 5000 points, which makes the accuracy of the model very good. The horizontal and vertical ranges and the duration of the flight

are calculated promptly. The trajectory is drawn, and the projectile flight along it is simulated. The simulation works with a constant time interval, which makes the simulated motion consistent with the physical principles – the ball slows down going up and speeds up going down. The actual time is calculated as a proportion of the total flight time, where the multiplier (≤ 1) is a fraction whose numerator and denominator are taken from two cells. The denominator is an empirical value; setting the value changes the duration of the simulated flight. Thus, a real flight time can be achieved, which makes the model an accurate diminishment of the real motion. The numerator changes from zero up to the denominator by one, which is performed by a macro. The ball coordinates that correspond to the actual time are returned from the graph points list by the VLOOKUP function. The trajectory in a vacuum can be graphed in the same chart to compare (switched on or off by a checkbox). To make the application a kind of computer game, a target with random position is set into the chart to be shot down. The aim of the paper is to show the reader another method of moving a point along the graph along with the possibilities that Excel offers in simulating for school physics.
Pedagogical remark: 35 years ago when I was in the first class of secondary grammar school, our physics teacher performed the theory of projectile motion, which is a most interesting topic for me so far [10], [11]. To my great disappointment, she finished with the note that the derived formulas are only valid in a vacuum. Hence, the nearest place where the theory works is on the Moon. The solution on the Earth was said to be too complicated for grammar school physics. The ballistic curve that she sketched on the blackboard by chalk just increased my interests. When computers came, making a model of projectile motion in the air was the reason why I learned BASIC. However, with the advent of spreadsheets, making numeric models of dynamic systems that are given by differential equations "is a cinch." In particular, if Euler's method is applicable directly, the mathematics of the model only comprises rewriting the differential equation into the difference one. The obtained recursive equations are ideal for performing the calculation in spreadsheets. If the students are familiar with the basics of calculus, the use of Euler's method is clear. 35 years ago, I would have appreciated the possibility to make such model or just to use it and experiment with it. However, no spreadsheet was in schools that time (even no computer). Despite the fact that more than thirty years have passed since spreadsheets sprung up [12], many maths and physics teachers do not know how wonderful a tool spreadsheets are. That is why spreadsheets still stay a rather "overlooked technology" [13].

2. Simulating projectile motion in vacuum

If the projectile is launched in a vacuum from the origin $(0,0)$ of an xy coordinate system at angle α and speed v_0, then the velocity components v_x, v_y are [7]

$$v_x = v_0 \cos\alpha, \tag{2}$$

$$v_y = v_0 \sin\alpha - gt, \tag{3}$$

and the coordinates x, y are

$$x = v_0 t \cos\alpha\,, \tag{4}$$

$$y = v_0 t \sin\alpha - \frac{1}{2} g t^2\,. \tag{5}$$

Substitution for time t from Eq. (4) to Eq. (5) gives the trajectory parabola

$$y = x \tan\alpha - \frac{g}{2 v_0^2 \cos^2\alpha} x^2\,. \tag{6}$$

Substituting $y = 0$ in Eq. (5) gives the duration of the flight

$$t_{\text{imp}} = \frac{2 v_0 \sin\alpha}{g}\,. \tag{7}$$

Substituting t_{imp} for t in Eq. (4) gives the x-coordinate at impact i.e. the x-range

$$x_{\text{imp}} = \frac{{v_0}^2 \sin 2\alpha}{g}\,. \tag{8}$$

This is a maximum at $\alpha = 45°$ with

$$x_{\max} = \frac{{v_0}^2}{g}\,. \tag{9}$$

Substituting $v_{\text{y}} = 0$ in Eq. (3) gives the duration of the flight to the vertex

$$t_{\text{ver}} = \frac{v_0 \sin\alpha}{g}\,. \tag{10}$$

Substituting t_{ver} for t in Eq. (5) gives the y-range

$$y_{\text{ver}} = \frac{{v_0}^2 \sin^2\alpha}{2g}\,. \tag{11}$$

This is a maximum at $\alpha = 90°$ with

$$y_{\max} = \frac{{v_0}^2}{2g}\,. \tag{12}$$

The application is in Fig. 1. The inputs are in cells Q2:Q4. The grey ones are only changeable by the scrollbars. The graph reacts interactively to the changes. First, duration t_{imp}, x-range x_{imp}, and y-range y_{ver} are calculated in cells T11:T13 using Eqs (7), (8), and (11). Then, the x-range is divided to 100 points x, and coordinate y is computed at each x according to Eq. (6) (range M19:N120). The trajectory parabola is an

xy line graphed over the 100 points (x, y). The fact that the x-range is a maximum at 45° can be revealed immediately.

To simulate the motion, a ball-shaped one-point xy graph is embedded into the chart. The main time interval t_{imp} is divided into N equal subintervals, and the actual time t of the projectile is calculated in cell P17 by the formula

$$t = \frac{n}{N} t_{\text{imp}}, \tag{13}$$

where n is in cell S17, and N is in cell S18. Number n goes from 0 to N by unit steps performed by the following macro of button "GO":

```
Private Sub btGo_Click()
 Range("S17") = 0
 For i = 1 To Range("S18")
  Range("S17") = Range("S17") + 1
  Calculate
 Next i
End Sub
```

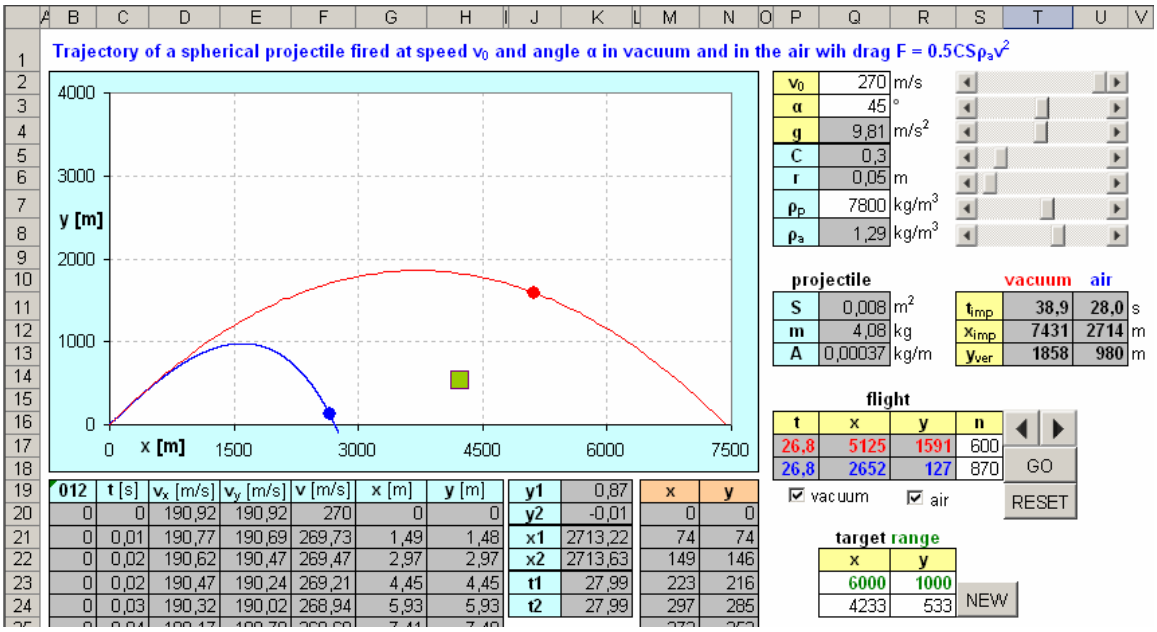

Figure 1: Projectile motion in vacuum (red) and in the air (blue)

Command "Calculate" is necessary for picturing the ball. To move the ball step by step, a spinbutton is used that is linked with cell S17. The maximum of the spinbutton is set at each click according to cell S18. Button "RESET" puts 0 in cell S17. The (new) xy coordinates of the projectile are calculated in range Q17:R17 using Eqs. (4), (5), which causes moving the ball to the new position. The model matches reality – the ball slows down if going up and it speeds up if going down. Increasing N causes an increase in the duration of the simulated flight. For the inputs in Fig. 1, if N = 870, then the simulated flight lasts 39 seconds, which is the real time of the flight. This makes the model an accurate diminishment of the real motion, however, slowing down and speeding up are not observable in this case (put N = 100 and $\alpha = 80°$ for that). To make the application

a kind of computer game, a square-shaped one-point xy graph is embedded into the chart as a target to be shot down. The coordinates of the target are random within the range set in cells Q23:R23, and the change is performed by a two-line macro assigned to the button "NEW" (put 0 in R23 for a target on the ground).

3. Simulating projectile motion in the air

Let the resistance force be given by Eq. (1). Let the body be launched at angle $0 < \alpha < 90°$, velocity $\mathbf{v}_0$, and acceleration of gravity g (Fig. 1). The equation of motion is

$$m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -mg\mathbf{j} - \frac{1}{2}CS\rho_\mathrm{a}v^2\mathbf{v}^0,\ \mathbf{v}(0) = \mathbf{v}_0,\ \mathbf{r}(0) = 0, \tag{14}$$

where $\mathbf{r}$ is the position vector $\mathbf{r} = x\mathbf{i} + y\mathbf{j}$. As $v^2\mathbf{v}^0 = v\mathbf{v}$, $\mathbf{v} = v_\mathrm{x}\mathbf{i} + v_\mathrm{y}\mathbf{j}$ (note that $v = \sqrt{v_\mathrm{x}^2 + v_\mathrm{y}^2}$), Eq. (14) can be recast into the coupled equations

$$\frac{\mathrm{d}v_\mathrm{x}}{\mathrm{d}t} = -Av_\mathrm{x}v,\ v_\mathrm{x}(0) = v_0\cos\alpha,\ x(0) = 0, \tag{15}$$

$$\frac{\mathrm{d}v_\mathrm{y}}{\mathrm{d}t} = -g - Av_\mathrm{y}v,\ v_\mathrm{y}(0) = v_0\sin\alpha,\ y(0) = 0, \tag{16}$$

where $A = \dfrac{CS\rho_\mathrm{a}}{2m}$.

This differential equation system is solvable numerically by Euler's method that is simple enough for the students who are familiar with the basics of calculus. Replacing the differentials with differences gives

$$\Delta v_\mathrm{x} = -Av_\mathrm{x}v\Delta t,\ v_{\mathrm{x}0} = v_0\cos\alpha, \tag{17}$$

$$\Delta v_\mathrm{y} = -(g + Av_\mathrm{y}v)\Delta t,\ v_{\mathrm{y}0} = v_0\sin\alpha. \tag{18}$$

First, the duration of the flight is estimated from above by the duration in a vacuum given by Eq. (7). Then, this time is divided to 5000 equal intervals Δt. Eqs. (17), (18) enable discrete values of v_x, v_y to be computed by the recursive formulae

$$v_{\mathrm{x}i} = v_{\mathrm{x}(i-1)} - Av_{\mathrm{x}(i-1)}v_{i-1}\Delta t,\ i = 1,2,...,5000,\ v_{\mathrm{x}0} = v_0\cos\alpha, \tag{19}$$

$$v_{\mathrm{y}i} = v_{\mathrm{y}(i-1)} - (g + Av_{\mathrm{y}(i-1)}v_{i-1})\Delta t,\ i = 1,2,...5000,\ v_{\mathrm{y}0} = v_0\sin\alpha, \tag{20}$$

where $v_{i-1} = \sqrt{v_{\mathrm{x}(i-1)}^2 + v_{\mathrm{y}(i-1)}^2}$.

The model is in Fig. 1. The inputs are in cells Q2:Q8. Parameter A is calculated in cell Q13, Δt is calculated in hidden cell Y11. Values of $v_{\mathrm{x}i}$, $v_{\mathrm{y}i}$, v_i are calculated in

columns D, E, F, starting at v_{x0}, v_{y0}, v_0 in row 20. Their values at $i = 1, 2, ..., 5000$ calculated from Eqs. (19) – (20) are in the next rows.
As $v_x = \Delta x / \Delta t$ and $v_y = \Delta y / \Delta t$, then

$$x_i = x_{i-1} + v_{xi} \Delta t, \; x_0 = 0, \tag{21}$$

$$y_i = y_{i-1} + v_{yi} \Delta t, \; y_0 = 0, \tag{22}$$

Values of x_i, y_i are calculated in columns G, H, starting at 0 in row 20. Their values at $i = 1, 2, ..., 5000$ calculated upon Eqs. (21), (22) are in the next rows.
The trajectory is an xy line graph made over the 5001 points (x, y) (see Fig. 1). The graph reacts interactively to the changes.
Values $t_i = t_{i-1} + \Delta t$, $t_0 = 0$ of the time are calculated in column B, starting at 0 in row 20. Column "012" contains 0 if $y > 0$, otherwise it contains the series 1, 2, 3,… that starts (at 1) in the last row where y is positive. The impact time t_{imp} and x-coordinate x_{imp} are calculated in cells U11:U12 using linear interpolation between the values of x and t that correspond to the last positive and first negative value of y – all the values are returned to cells K19:K24 by function VLOOKUP using column "012".
The simulation works upon the same principles as mentioned above for a vacuum. The actual time t of the projectile is calculated in cell P18 from Eq. (13) where t_{imp} is that for the air. The xy coordinates of the projectile that correspond to time t are returned to cells Q18:R18 by function VLOOKUP using column "012". However, if the checkbox "vacuum" is checked, i.e. the both trajectories are shown, then the simulation works with t_{imp} valid for vacuum, which allows comparing the positions of the both projectile at the same time points.
It can be easily shown that the maximum x-range is achieved in the air at a launch angle smaller than 45°. If the inputs are identical to those in Fig. 1, then it is 38° In this case, it is an iron ball of diameter 10 cm shot at 270 m/s. Enlarging the ball or increasing its density causes decreasing parameter A, which gets the trajectory closer to that one in a vacuum. Decreasing the launch speed decreases the influence of the drag, which gets the trajectory closer to that one in a vacuum, again. Consequently, the launch angle for achieving the maximum x-range is closer to 45°. For example, it is 40° for an iron ball of diameter 20 cm shot at 270 m/s, and it is 44° for an iron ball of diameter 20 cm shot at 80 m/s. For comparing, it is 39° for a led ball of diameter 10 cm shot at 270 m/s, it is 41° for a led ball of diameter 20 cm shot at 270 m/s, and it is 45° for a led ball of diameter 20 cm shot at 80 m/s.
The accuracy of the model is very good – if the inputs are set according to Fig. 1, then the x-range is $x_{\text{imp}} = 2714$ m, which makes a difference of 2 m only from the exact value $x_{\text{imp}} = 2716$ m that gives the integral solution in [6] (calculation performed by a compute program), i.e. the relative error is 0.074 %. Then, if air density ρ_a is set to zero (that is, the motion takes place in a vacuum), the x-range is $x_{\text{imp}} = 7430$ m, which makes a difference of 1 m only from the exact value 7431 m given by Eq. (8), that is, the relative error is 0.013 %.

4. Conclusions

The aim of the paper is to show a way of creating simulation models that are accurate diminishments of the real motion, including the duration. The crucial parts of the application do not require programming, except for the macros of button "GO" and the spinbutton (one can put manually 0 in S17 instead of clicking on button "RESET", or put values from the target range in Q24:Q2 instead of clicking on button "NEW"). However, a click and holding down on the spinbutton can substitute the function of button "GO", hence the line of VBA code in SpinButton1_Change routine is only necessary to implement. Even that can be bypassed by changing the spinbutton maximum manually, if necessary.

References

[1] Stewart, S.M. (2006). An analytic approach to projectile motion in a linear resisting medium. *Int. J. Math. Educ. Sci. Technol*. 37, 411− 431.

[2] http://en.wikipedia.org/wiki/Transonic_flow, retrieved 1 March 2009

[3] Marion, J.B. (1970). *Classical dynamics*. New York: Academic , 2nd edition.

[4] http://en.wikipedia.org/wiki/Drag_coefficient, retrieved 1 March 2009

[5] http://en.wikipedia.org/wiki/Projectile, retrieved 1 March 2009

[6] Hayen, J.C. (2003). Projectile motion in a resistant medium. *Int. J. Non-Linear Mech*. 38, 357-370.

[7] Neuwirth, E. and Arganbright, D. (2004). *The active modeler − Mathematical modeling with Microsoft Excel.* Belmont: Brooks-Cole.

[8] Arganbright, D. (2005). Enhancing Mathematical Graphical Displays in Excel through Animation. *Spreadsheets in Education* 2(1), 120-147. Available online at http://epublications.bond.edu.au/ejsie/vol2/iss1/8/

[9] Wischniewsky, W.A.L. (2008). Movie-like Animation with Excel's Single Step Iteration Exemplified by Lissajous Figures. *Spreadsheets in Education*, 3(1), 46-58. Available online at http://epublications.bond.edu.au/ejsie/vol3/iss1/4/

[10] Benacka, J. and Stubna, I. (2009). Ball launched against an inclined plane – an example of using recurrent sequences in school physics. *Int. J. Math. Educ. Sci. Technol*. 40 (5), 696 − 705. At http://pdfserve.informaworld.com/876421_905729217_910913813.pdf.

[11] Benacka, J. (2009). Solution to projectile motion with quadratic drag and graphing the trajectory in spreadsheets. *Int. J. Math. Educ. Sci. Technol*. On iFirst. Available at http://pdfserve.informaworld.com/872676_905729217_915022217.pdf.

[12] Baker, J. and Sugden, S. (2003). Spreadsheets in Education – The First 25 Years. *Spreadsheets in Education* (eJSiE), 1 (1), Article 2. Available online at http://epublications.bond.edu.au/ejsie/vol1/iss1/2

[13] Sugden, S. (2007). Spreadsheets: an overlooked technology for mathematics education. *Gazette of the Australian Mathematical Society*. Available online at http://www.austms.org.au/Publ/Gazette/2007/May07/068ClassroomNotes.pdf