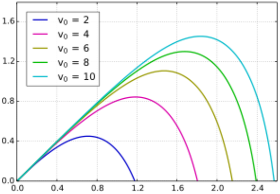
Lecture 2:

Projectile motion

Euler and higher-order methods

Why do golf balls have dimples? Credit: Penn State.



Ballistic motion, Credit: wikipedia

Ballistic motion: Mathematical model & analytical solution

- Newtonian dynamics:

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = 0\,; \qquad \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -g$$

$g \approx 9.8$ m s^{-2} is the acceleration due to gravity, x is horizontal distance travelled, y is height

- Analytical solution:

$$x = x_0 + \dot{x}_0\, t\,; \qquad y = y_0 + \dot{y}_0\, t - \frac{g}{2} t^2\,.$$

$(x_0\,,\,y_0)$ is initial position, $(\dot{x}_0\,,\,\dot{y}_0)$ is initial velocity in x and y direction

- In terms of launch angle, θ_0, and launch speed, v_0,

$$\dot{x}_0 = v_0 \cos(\theta_0)\,; \qquad \dot{y}_0 = v_0 \sin(\theta_0)\,.$$

Ballistic motion: Mathematical model & analytical solution

- Exercise: show that for $\dot{y}_0 > 0$ assume $(x_0, y_0) = (0, 0)$ and a flat terrain:
  - maximum height is reached at time $t_{\max}$

$$t_{\max} = \frac{\dot{y}_0}{g}$$

  - maximum distance travelled when $\theta_0 = \frac{\pi}{4}$
- Particle's energy, $E = \frac{1}{2}mv^2 + mgy$, is conserved

$\dot{E} = m(v_x \dot{v}_x + v_y \dot{v}_y + g v_y) = m v_y(\dot{v}_y + g) = 0$, since $\dot{v}_x = 0$ and $\dot{v}_y = -g$

Good test for numerical solution!

Ballistic motion: Numerical solution

- Euler's method (see lecture on radioactive decay)
  - Solution for differential equations of the type

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x},\, t)\,.$$

  - Discretise time t and coordinate x with time-step Δt:

$$\mathbf{x}(t^{n+1}) \equiv \mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{f}(\mathbf{x}^n,\, t^n)\Delta t\,.$$

- Euler method won't work directly first order differential equations only
- We will massage the equations!

Ballistic motion: Numerical solution

- Problem: Euler's method not directly applicable, because equations are second order
- Solution: use velocities as well (generally applicable)
  - Original second-order equation: $f_y = -g$ in previous slide

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = f_y$$

  - Rewrite as two, first-order equations:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = v_y\,; \quad \frac{\mathrm{d}v_y}{\mathrm{d}t} = f_y\,.$$

and similarly for x (and z, etc)

- Solve first-order equations using Euler's method

Ballistic problem: Numerical solution (cont'd)I

- Mathematical model: $\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = 0\,; \quad \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -g$
- Initial conditions: Launch angle θ_0, launch speed v_0, $(x_0, y_0) = (0, 0)$, $(v_{x,0}, v_{y,0}) = v_0(\cos(\theta_0), sin(\theta_0))$
- Euler's method: $t = 0$: $(x^0, y^0) = (0, 0)$, $(v_x^0, v_y^0) = v_0(\cos(\theta_0), sin(\theta_0))$

$$
\begin{aligned}
x(t^{n+1}) \equiv x^{n+1} &= x^n + v_x^n \Delta t\,; \quad v_x^{n+1} = v_x^n + 0\,\Delta t \\
y(t^{n+1}) \equiv y^{n+1} &= y^n + v_y^n \Delta t\,; \quad v_y^{n+1} = v_y^n - g\Delta t \\
t^{n+1} &= t^n + \Delta t
\end{aligned}
$$

Exercise: does this conserve energy? Answer: NO!

Ballistic motion: Numerical solution (cont'd)

- As in previous lecture: need to choose Δt carefully
  - time-scale in this problem: $t_{\max} = \frac{v_{y,0}}{g}$ is time to reach maximum height

    therefore take $\Delta t \ll t_{\max}$
  - the flight duration is $t_f = 2t_{\max} \rightarrow$ equivalently take $\Delta t \ll t_f$
- Analytical solution known: good test of implementation and choice of Δt

Air resistance: mathematical model

Projectile suffers from air resistance, which depends on speed. No known analytical solution.

- Drag force:

$$\mathbf{F}_{\mathrm{drag}} = -B_{1,\mathrm{drag}}\, v \frac{\mathbf{v}}{v} - B_{2,\mathrm{drag}}\, v^2 \frac{\mathbf{v}}{v} + \ldots$$

- drag force is parallel to velocity, $\mathbf{F} \parallel \mathbf{v}$
  $\frac{\mathbf{v}}{v}$ is unit vector in the direction of motion
- drag coefficients $B_{1,\mathrm{drag}} > 0$ and $B_{2,\mathrm{drag}} > 0$ since drag *slows projectile down*

Air resistance: mathematical model (cont'd)

- Dimensional analysis: $|\mathbf{F}_{\rm drag}|$ depends on density of air (ρ), speed (v) and size of projectile (r): $F_{\rm drag} \propto \rho^\alpha v^\beta r^\gamma$

  $[A]$ means dimension of A

$$\begin{aligned} [F_{\rm drag}] &= \mathrm{kg\ m\ s}^{-2} = [\rho]^\alpha [v]^\beta [r]^\gamma = (\mathrm{kg\ m}^{-3})^\alpha\,(\mathrm{m\ s}^{-1})^\beta\,\mathrm{m}^\gamma \\ \to \alpha &= 1\,; \quad \beta = 2\,; \quad \gamma = 2 \end{aligned}$$

- Therefore take

$$\mathbf{F}_{\rm drag} \approx -B_{2,\rm drag}\, v^2 \frac{\mathbf{v}}{v} = -B_{2,\rm drag}\, v \begin{pmatrix} v_x \\ v_y \end{pmatrix},$$

  where, of course, $v^2 = v_x^2 + v_y^2$, and $B_{2,\rm drag} \propto \rho r^2$ depends on projectile's size and density of air

- Homework: $B_{2,\rm drag} = B_{2,\rm drag}(y) = B_{2,\rm drag}(y=0)\frac{\rho(y)}{\rho(y=0)}$

Air resistance: Numerical solution

- Mathematical model: : m is mass of projectile, B is drag coefficient
$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} = -\frac{B(y) v\, v_x}{m}$, $\quad \frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = -g - \frac{B(y) v\, v_y}{m}$
- Euler's method: $t = 0$: $(x^0, y^0) = (0, 0)$, $(v_x^0, v_y^0) = v_0(\cos(\theta_0), sin(\theta_0))$

$$
\begin{aligned}
x^{n+1} &= x^n + v_x^n \Delta t \, ; \quad v_x^{n+1} = v_x^n - \frac{B(y^n) v^n v_x^n}{m} \Delta t \\
y^{n+1} &= y^n + v_y^n \Delta t \, ; \quad v_y^{n+1} = v_y^n - g\Delta t - \frac{B(y^n) v^n v_y^n}{m} \Delta t \\
t^{n+1} &= t^n + \Delta t \\
v^n &= \left((v_x^n)^2 + (v_y^n)^2 \right)^{1/2}
\end{aligned}
$$

taking $\Delta t << v_{y,0}/g$

Pseudo-code

Main program

- Initial conditions.
- Calculate the trajectory.
- Print/plot the result.
- Calculate range.

Initialisation

- Fix x_0, y_0, t_0, fix/read in v_0, θ_0 (in degrees).

Calculation

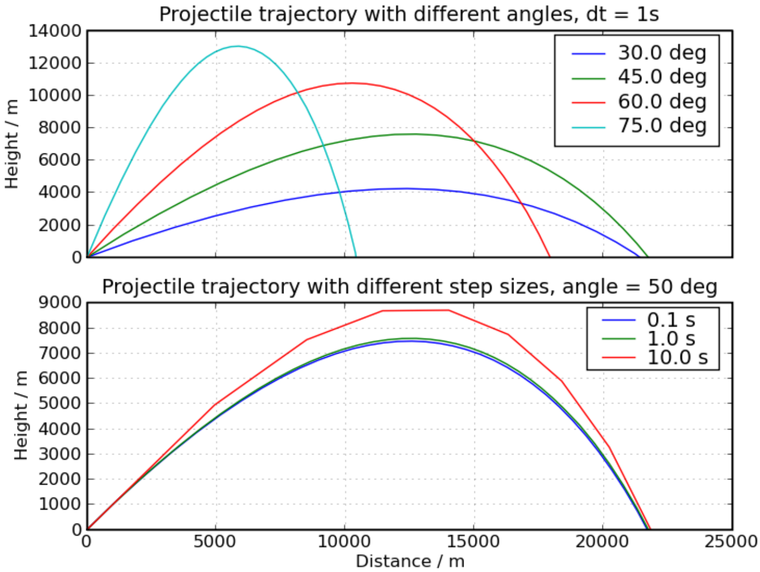
- Iterate eqn's above, stop when $y_i < 0$, $n_{\text{end}} = n = i$.

Calculate range

- Range from interpolation between (x_n, y_n) and (x_{n-1}, y_{n-1}):

$$x_{\text{range}} = \frac{y_n x_{n-1} - y_{n-1} x_n}{y_n - y_{n-1}}\,.$$

Results for trajectories



Higher-order methods Improving the Euler's method

- Euler method simple to implement, but correct only to $\mathcal{O}(\Delta t)$. Can we improve this?
- Yes, we can!
  Remember origin of Euler's method: Taylor expansion

$$x(t+\Delta t) = x(t) + \frac{\mathrm{d}x}{\mathrm{d}t}\Delta t + \dots$$

  According to the mean value theorem:

$$\exists t' \in [t, t+\Delta t] \; : \; x(t+\Delta t) \equiv x(t) + \left.\frac{\mathrm{d}x}{\mathrm{d}t}\right|_{t=t'} \Delta t$$

- Here t' includes higher order effects (curvature etc.).
  Drawback: Not known generally, but maybe better choices than $t' = t$ employed in Euler method

Higher-order methods: 2^{nd} order Runge-Kutta (RK2)

- Underlying idea: Estimate $t' = t + \Delta t/2$
- But: also need $\mathrm{d}x/\mathrm{d}t$ at $t = t'$.
  Estimate x' using the 'prediction'

$$x' = x + f(x,\ t)\frac{\Delta t}{2}\,.$$

- Second-order scheme (precision $\mathcal{O}[(\Delta t)^2]$):

$$\begin{aligned} x' &= x + f(x,\ t)\frac{\Delta t}{2} \\ x(t+\Delta t) &= x(t) + f(x', t')\Delta t \\ x^{n+1} &= x^n + f\left(x^n + \frac{\Delta t}{2} f(x^n, t^n),\ t^n + \frac{\Delta t}{2}\right)\Delta t \end{aligned}$$

$t^{n+1} = t^n + \Delta t$

Higher-order methods: 4^{th} order Runge-Kutta (RK4)

- Further improvement: More sampling points

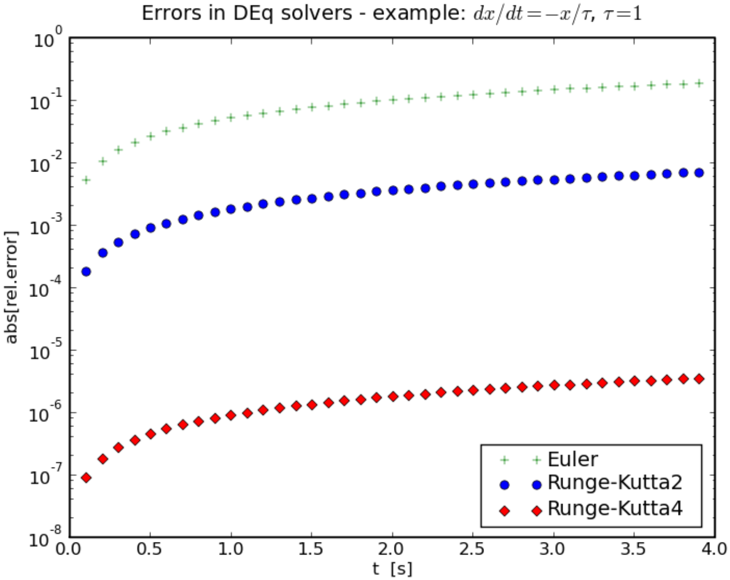
$$
\begin{aligned}
& x(t+\Delta t) = x(t) \\
+ \quad & \frac{\Delta t}{6}\left[f(x_1', t_1') + 2f(x_2', t_2') + 2f(x_3', t_3') + f(x_4', t_4')\right] .
\end{aligned}
$$

- Sampling points given by

$$
\begin{aligned}
x_1' &= x & t_1' &= t \\
x_2' &= x + f(x_1', t_1')\tfrac{\Delta t}{2} & t_2' &= t + \tfrac{\Delta t}{2} \\
x_3' &= x + f(x_2', t_2')\tfrac{\Delta t}{2} & t_3' &= t + \tfrac{\Delta t}{2} \\
x_4' &= x + f(x_3', t_3')\Delta t & t_4' &= t + \Delta t .
\end{aligned}
$$

- Fourth-order scheme (precision $\mathcal{O}[(\Delta t)^4]$)

Euler vs. Runge-Kutta(s) for Radioactive Decays



Integration of 2^{nd} order DEs - some more considerations

- Consider what is needed Value at $t = t_{\mathrm{end}}$? Or whole path?
- What is the accuracy required?
- Choice of Δt? Should Δt itself vary? How?

  How does that change the method/code?
- Higher-order methods 4^{th} order RK especially popular in computational physics
  - higher-order does not imply *higher accuracy*
  - more evaluations per step

    more computationally expensive unless step-size correspondingly larger
- Other methods exist e.g. predictor-corrector, see e.g. *Numerical Recipes*
- Method discussed here only works for smooth functions f

Summary

- Another example for numerical solutions of differential equations: trajectory of a particle
- Euler's method not directly applicable due to presence of 2^{nd} order derivatives

  Solution: Use velocities:

  one 2^{nd}-order DE $\rightarrow$ two 1^{st}-order DEs generally applicable
- This allows to use the Euler method (again).
- Improvement of the Euler method possible,

  higher-order methods: e.g. Runge-Kutta methods

  better accuracy for same step-size but more computations per step

Further physics extensions to projectile motion

- Value of drag coefficient depends on velocity

  underlying physics changes from laminar airflow at low speed to turbulent flow at high speed

  important aspect in describing the flight of a baseball!

- properties of the surafce of the projectile matter

  airflow, and hence drag force, depends significantly on smoothness of projectile's surface

- spin: making ball spin can dramatically affect flight path

  e.g. golf: strong back-spin dramatically increases range

  spin can make trajectory curved - e.g. football or tennis

  Exercise: use dimensional analysis to guess form of force to add