Computational Physics I

Instructor: Prof. Ulrich Kleinekathöfer, Research III, room 61
email: u.kleinekathoefer@jacobs-university.de

Topics (preliminary):
1) Classical dynamics
2) Electrostatics
3) Cellular Automata and Lattice Gas Models
4) Wave Dynamics
5) Dynamical motion in chaotic systems

NOT a programming course!
Knowledge of Python on level of "Numerical Software Lab" is expected.

Computational Physics I & II constitute a Module. Only one grade for the Module!
(both terms => one Module grade)

Computational Physics II in Spring 2023

Grading based on "Project(s)":
In class, we will consider a certain type of physics problem and discuss how to solve it using computational methods. Based on these discussions, projects will be assigned in which you are asked to solve a specific problem.
You will have to (i) analyze the problem, (ii) select an algorithm (if not specified), (iii) write a Python program, (iv) run the program, (v) visualize the data numerical data, and (vi) extract an answer to the physics question from the data.

For each project you will submit a short report describing the physics problem, your way of attacking it, and the results you obtained. Each report will be worth 100 pts. A total of roughly 6 projects (per term) will be assigned. Your lowest score (per term) will be dropped (if submitting in the given time frame). When working on the projects and reports, discussions among colleagues are allowed and encouraged. However, the reports you hand in should be based on your efforts (i.e., your code !! and your simulation runs) and not that of a group. You should document any reference material which you directly use.

You should behave as responsible scholars. Academic dishonesty such as plagiarism and cheating is unethical and unacceptable and will be dealt with accordingly.

Book (for parts of the course): "Introduction to Computer Simulation Methods"
by Harvey Gould, Jan Tobochnik, and Wolfgang Christian
free pdf version at: http://physics.clarku.edu/sip/

Moodle
Communication, material and homework submission using Moodle
https://elearning.jacobs-university.de

TA: Tejal Bhattarai

1) Classical dynamics

1.1 Motion of a falling

idealized particle: no rotation, no internal motion
position: y(t), velocity v(t), acceleration a(t)

kinematic equations:

$$v(t) = \frac{dy(t)}{dt} \qquad a(t) = \frac{dv(t)}{dt} = \frac{d^2y(t)}{dt^2}$$

Newton's second law: $F = m \cdot a$ $\qquad a(t) = \frac{F(y,v,t)}{m}$

F force, m inertial m

"free fall": all objects have the same acceleration regardless of size, mass, compostion, ...

- acceleration on earth: $g=9.81\ \text{m/s}^2$ $\qquad \uparrow y$

$a(t)=-g$, $v(t)= v_0 - g \cdot t$

$$y(t) = y_0 + v_0 t - \frac{1}{2} g t^2$$

Variation of the earth's gravitational field

Newon's law of gravitation

$$F = \frac{GMm}{(R+y)^2} = \frac{GMm}{R^2\left(1+\frac{y}{R}\right)^2} = mg\left(1 - 2\frac{y}{R} + ..\right)$$

y: distance from the surface of the earth
R=6370 km radius of the earth, M= $5.99 \cdot 10^{24}$ kg

G=6.67$\cdot 10^{-11}\frac{m^3}{kg\,s^2}$ gravitational constant

$$g=\frac{GM}{R^2}$$

Velocity-dependent friction (drag force)

$F_d = k_1 v$ or $F_d = k_2 v^2$

example: spherical pebble, m=10 g, r =1cm

$F_d = k_2 v^2$ $\quad k_2 = 10^{-4}\frac{kg}{m}$

terminal velocity: $F_d = mg$ $\quad k_2 v^2 = mg$

$$v_t = \sqrt{\frac{m\cdot g}{k_2}} \approx 30\frac{m}{s}$$

reached in a fall of 50m in about 3 s

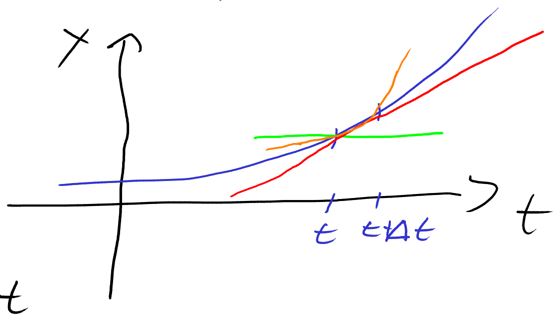
1.2 Euler algorithm

differential equation => finite differences

$$\frac{dy}{dt} = v(t)$$

$$y_1 = y(t+\Delta t) = y(t) + v(t)\Delta t + \frac{1}{2}a(t)(\Delta t)^2 + O((\Delta t)^3)$$

Taylor expansion assuming small Δt



keep only terms of order Δt

$y_n = y(t_0 + n\cdot\Delta t)$, $t_n = n\cdot\Delta t$

Euler scheme: $y_{n+1} = y_n + v_n \Delta t$

equivalent: $v_{n+1} = v_n + a_n \Delta t$ $\quad a_n = \frac{F(y_n, v_n, t)}{m}$

(time-)local error: order of $(\Delta t)^2$

(time-)global error: number of steps: $T = N\cdot\Delta t$ $\quad N = \frac{T}{\Delta t} \propto \frac{1}{\Delta t} = (\Delta t)^{-1}$

accumulation of errors => global error $\quad (\Delta t)^2\cdot\frac{1}{\Delta t} = \Delta t = (\Delta t)^1$

Euler scheme is a first-order scheme

in general: global error $(\Delta t)^n \Rightarrow n^{th}$ - order method

1.3 Euler-Richardson algorithm

split Δt in 2 steps: $\frac{\Delta t}{2}$

$$y\left(t+\frac{\Delta t}{2}\right) \simeq y(t) + v(t)\frac{\Delta t}{2} + \frac{1}{2}a(t)\left(\frac{\Delta t}{2}\right)^2$$

second half

$$y_2(t+\Delta t) \simeq y\left(t+\frac{\Delta t}{2}\right) + v\left(t+\frac{1}{2}\Delta t\right)\frac{\Delta t}{2} + a\left(t+\frac{\Delta t}{2}\right)\left(\frac{\Delta t}{2}\right)^2$$

$$= y(t) + \frac{1}{2}\left[v(t) + v\left(t+\frac{\Delta t}{2}\right)\right]\Delta t + \frac{1}{2}\left[a(t) + a\left(t+\frac{\Delta t}{2}\right)\right]\left(\frac{\Delta t}{2}\right)^2$$

use $a\left(t+\frac{1}{2}\Delta t\right) \simeq a(t) + \frac{1}{2}a'(t)\Delta t$

$$y_2(t+\Delta t) = y(t) + \frac{1}{2}\left[v(t) + v\left(t+\frac{\Delta t}{2}\right)\right]\Delta t + a(t)\frac{(\Delta t)^2}{4}$$

Euler-Richardson scheme

$$y_{ER}(t+\Delta t) = 2y_2(t+\Delta t) - y_1(t+\Delta t)$$
$$= y(t) + v\left(t+\frac{\Delta t}{2}\right)\Delta t + O((\Delta t)^3)$$

equivalent:

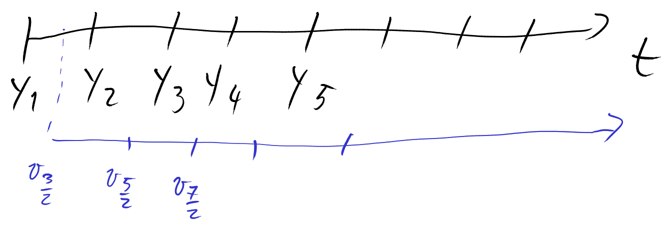
$$v_{ER}(t+\Delta t) = v(t) + a\left(t+\frac{\Delta t}{2}\right) + O((\Delta t)^3)$$

error estimation: $|y_2 - y_1|$ or $|v_2 - v_1|$

- second-order method

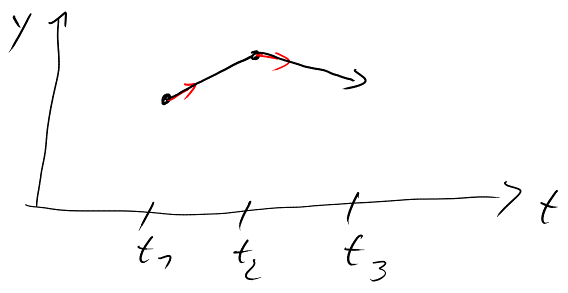
algorithm:

$$a_n = \frac{F(y_n, v_n, t_n)}{m}$$
$$v_{mid} = v_n + \frac{1}{2}a_n\Delta t$$
$$y_{mid} = y_n + \frac{1}{2}v_n\Delta t$$
$$a_{mid} = \frac{F(y_{mid}, v_{mid}, t+\frac{\Delta t}{2})}{m}$$
$$v_{n+1} = v_n + a_{mid}\Delta t$$
$$y_{n+1} = y_n + v_{mid}\Delta t$$



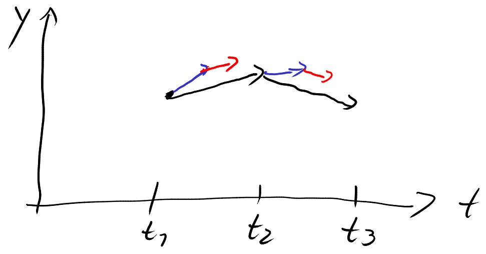
efficient if $a = $ const.

Euler scheme
(first order)

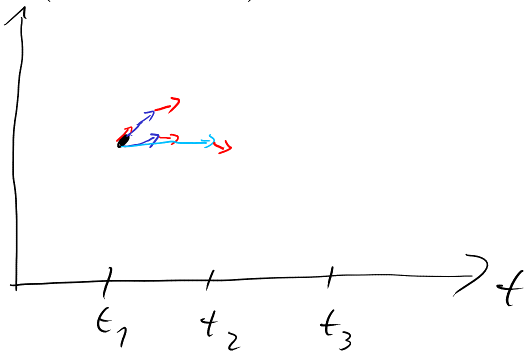


Euler-Richardson scheme
(second order)

derivative



Runge-Kutta scheme (fourth order)



in each time step calculate 4 derivatives

$$\frac{dy}{dt} = f(y,t)$$

$$r_1 = f(y_n, t_n)\Delta t$$

$$r_2 = f\left(y_n + \frac{r_1}{2}, t_n + \frac{\Delta t}{2}\right)\Delta t$$

$$r_3 = f\left(y_n + \frac{r_2}{2}, t_n + \frac{\Delta t}{2}\right)\Delta t$$

$$r_4 = f(y_n + r_3, t_n + \Delta t)\Delta t$$

$$\Rightarrow y_{n+1} = y_n + \frac{1}{6}(r_1 + 2r_2 + 2r_3 + r_4)$$