QUESTIONS :

Do BVP for SHO HW

midterms in 2wk

general ODE strategy

↓ gives rate of change @ each configuration

$$\frac{d\vec{x}}{dt} = f(\vec{x}, t)$$

how much $\vec{x}$ (STATE VECTOR) changes ...

... depends on it's current configuration and the time

ALGORITHMS : some variant of

$$x_{i+1} = x_i + \Delta t \, f(x_i, t_i)$$

- EULER METHOD $\mathcal{O}(\Delta t)$
- RK2 $\mathcal{O}(\Delta t^2)$

↓ can go on (RK4, ...)

PROBLEM : WE CODED THE (nonlinear) simple harmonic osc.

observed : amplitude grows

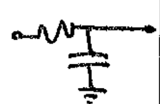
claim: you can use RK4, RK6, etc.

↳ amplitude will keep growing

physics!

WE SHOWED : THIS IS BECAUSE ENERGY GROWS

↳ approx error is s.t. small positive term is added each step



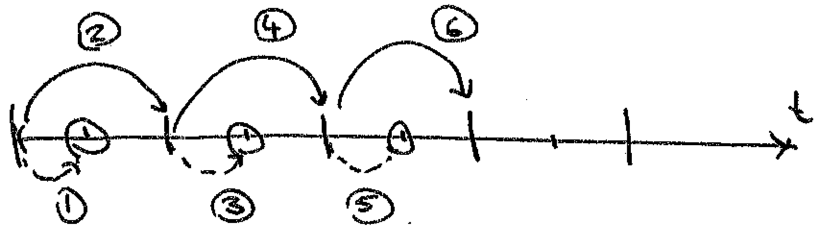
why is this a problem for SHO, but not for RC circ?

EULER METHOD:

arrow: evolve state $\vec{x}$

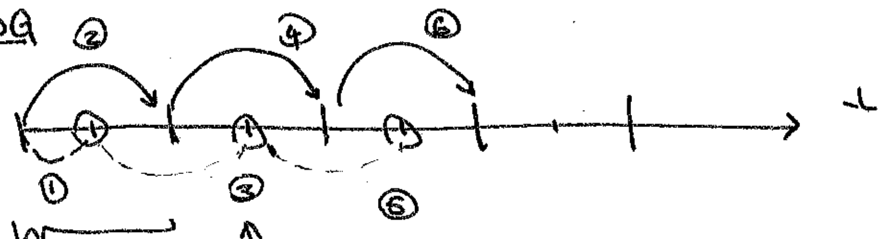
circle: sample velocity

RK2



estimate "halfway point" velocities

LEAPFROG



start like RK1

"big jump"

each halfway point calculated from previous halfway pt.

$$x_{i+1/2} = x_i + \tfrac{1}{2}\Delta t\, f(x_i, t_i) \quad \leftarrow \text{EULER}$$

integer: $x_{i+1} = x_i + \Delta t\, f(x_{i+1/2}, t_{i+1/2})$

1/2 integer: $x_{(i+1/2)+1} = x_{i+1/2} + \Delta t\, f(x_{i+1}, t_{i+1})$

EXAMPLE: CENTRIPETAL FORCE

$$F = -\frac{mv^2}{r}\hat{r} = m\ddot{\underline{x}}$$

$r = \sqrt{x^2+y^2}$ $\qquad$ $\hat{r} = \cos\theta\,\hat{x} + \sin\theta\,\hat{y} = \frac{x}{\sqrt{x^2+y^2}}\hat{x} + \frac{y}{\sqrt{x^2+y^2}}\hat{y}$

$$\dot{V}_x = -\frac{V_x^2+V_y^2}{\sqrt{x^2+y^2}}\cdot\frac{x}{\sqrt{x^2+y^2}} = -\frac{x(V_x^2+V_y^2)}{x^2+y^2} = f$$

$$\dot{V}_y = -\frac{y(V_x^2+V_y^2)}{x^2+y^2} = g$$

$$\dot{x} = V_x$$
$$\dot{y} = V_y$$

INTERESTING OBSERVATION

E seems to be CONSERVED w/ RK
if you update simultaneously ie

$$\dot{x} = f(x,V,t) = V$$
$$\dot{V} = g(x,V,t)$$

incl. this

$K1x = dt \times f(x,V,t)$
$K1V = dt \times g(x,V,t)$
$K2X = dt \times f(x + 0.5\cdot K1x,\ V\boxed{+0.5\cdot K2x}, t)$

why is this the case? I REALLY DON'T KNOW.

<u>How does it work?</u> [~~COMPUTATIONAL~~ PHYSICS SIDE of "COMPUTATIONAL PHYSICS"]

[hah. I got it backwards...

<u>Physics</u>: E conserv $\longleftrightarrow$ time transl. invariance

BTW: <u>not</u> true in our universe

<u>ODE solving</u>: turns out to also be similar

$\rightarrow$ I think it's mostly poetry.

<u>PROBLEM</u>: something that should be <u>constant</u> is changing w/ <u>time</u>

each time step.

... in such a way that there is an accumulated drift.

one way to try to combat this: tweak the algorithm s.t. it works the same way going forward / backward in time!

eg. <u>LEAP FROG</u>:

~~$x(t+\Delta t) = x(t) + \Delta t \, f(x(t)$~~

$$x_{i+1} = x_i + \Delta t \, f(x_{i+1/2}, t_{i+1/2}) \quad \text{(A)}$$

$$x_{(i+1/2)+1} = x_{i+1/2} + \Delta t \, f(x_{i+1}, t_{i+1}) \quad \text{(B)}$$

now "time reverse" by changing $\Delta t \to -\Delta t$

↳ eg $x_{i+1} = x(t+\Delta t) \longrightarrow x_{i-1} = x(t-\Delta t)$

<u>then</u>:

$$x_{i-1} = x_i - \Delta t\, f(x_{i-1/2}, t_{i-1/2}) \quad \text{A}'$$

$$x_{(i-1/2)-1} = x_{i-1/2} - \Delta t\, f(x_{i-1}, t_{i-1}) \quad \text{B}'$$

now shift $\boxed{i \to i + 3/2}$

still minus

$$x_{i+1/2} = x_{(i+1/2)+1} - \Delta t\, f(x_{i+1}, t_{i+1})$$

$$x_i = x_{i+1} - \Delta t\, f(x_{i+1/2}, t_{i+1/2})$$

exactly the same as A + B

---

by comparison: RK2

(a) $x_{i+1/2} = x_i + \frac{1}{2}\Delta t\, f(x_i, t_i)$

(b) $x_{i+1} = x_i + \Delta t\, f(x_{i+1/2}, t_{i+1/2})$

$$x_{i-1/2} = x_i - \tfrac{1}{2}\Delta t\, f(x_i, t_i)$$

$$x_{i-1} = x_i - \Delta t\, f(x_{i-1/2}, t_{i-1/2})$$

$$x_i = x_{i+1/2} - \tfrac{1}{2}\Delta t\, f(x_{i+1/2}, t_{i+1/2})$$

$$x_i = x_{i+1} - \Delta t\, f(x_{i+1/2}, t_{i+1/2}) \longrightarrow \text{(b)}$$

$$\boxed{x_{i+1/2} = x_i + \tfrac{1}{2}\Delta t\, f(x_{i+1/2}, t_{i+1/2})} \neq \text{(a)}$$

Other types of Differential Eq:

ODE → initial value

eg $\ddot{x} = f(x, v, t)$

$x(0)$ & $\dot{x}(0)$ given.

WHAT IS $x(T)$?

ODE → boundary value

eg. $\ddot{x} = f(x, v, t)$

$x(0)$ & $x(T)$ given

WHAT IS REQ. v_0?

eg: $\boxed{\ddot{x} = -g}$

$x(0)$ = shoulder height

$x(T)$ = 10 feet

REGULATION BBALL RIM

problem: this is @ the <u>END</u> of the integration!

we like initial conditions like x_0 & v_0!

BASIC TECHNIQUE: scan over V_0 values, integrate,

see if $x(T)$ is what you want.

See: Newman. P 390-391 ex. 8.8

with that initial velocity. Our goal in solving the boundary value problem is to find the value of v that makes the function zero. That is, we want to solve the equation $f(v) = 0$. But this is simply a matter of finding the root of the function $f(v)$ and we already know how to do that. We saw a number of methods for finding roots in Section 6.3, including binary search and the secant method. Either of these methods would work for the current problem.

RELAXATION
BINARY SEARCH
NEWTON
GOLDEN RATIO

So the shooting method involves using one of the standard methods for solving differential equations, such as the fourth-order Runge–Kutta method, to calculate the value of the function $f(v)$, which relates the unknown initial conditions to the final boundary condition(s). Then we use a root finding method such as binary search to find the value of this function that matches the given value of the boundary condition(s).

EXAMPLE 8.8: VERTICAL POSITION OF A THROWN BALL

Let us solve the problem above with the thrown ball for the case where the ball lands back at $x = 0$ after $t = 10$ seconds. The first step, as is normal for second-order equations, is to convert Eq. (8.105) into two first-order equations:

$$\frac{dx}{dt} = y, \qquad \frac{dy}{dt} = -g. \tag{8.106}$$

We will solve these using fourth-order Runge–Kutta, then perform a binary search to find the value of the initial velocity that matches the boundary conditions. Here is a program to accomplish the calculation:

File: throw.py

```
from numpy import array,arange

g = 9.81          # Acceleration due to gravity
a = 0.0           # Initial time
b = 10.0          # Final time
N = 1000          # Number of Runge-Kutta steps
h = (b-a)/N       # Size of Runge-Kutta steps
target = 1e-10    # Target accuracy for binary search

def f(r):
    x = r[0]
    y = r[1]
    fx = y
    fy = -g
    return array([fx,fy],float)
```

```
# Function to solve the equation and calculate the final height
def height(v):
    r = array([0.0,v],float)
    for t in arange(a,b,h):
        k1 = h*f(r)
        k2 = h*f(r+0.5*k1)
        k3 = h*f(r+0.5*k2)
        k4 = h*f(r+k3)
        r += (k1+2*k2+2*k3+k4)/6
    return r[0]

# Main program performs a binary search
v1 = 0.01
v2 = 1000.0
h1 = height(v1)
h2 = height(v2)

while abs(h2-h1)>target:
    vp = (v1+v2)/2
    hp = height(vp)
    if h1*hp>0:
        v1 = vp
        h1 = hp
    else:
        v2 = vp
        h2 = hp

v = (v1+v2)/2
print("The required initial velocity is",v,"m/s")
```

One point to notice about this program is that the condition for the binary search to stop is a condition on the accuracy of the height of the ball at the final time $t = 10$, not a condition on the initial velocity. In most cases we care about matching the boundary conditions accurately, not calculating the initial conditions accurately.

If we run the program it prints the following:

```
The required initial velocity is 49.05 m/s
```

In principle, we could now take this value and use it to solve the differential equation once again, to compute the actual trajectory that the ball follows, verifying in the process that indeed it lands back on the ground at the allotted time $t = 10$.

PARTIAL DIFFERENTIAL EQUATIONS

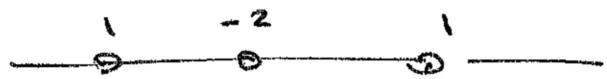
partial derivatives — b/c many directions

... can't just linearly evolve!

PARTIAL DERIVATIVE :
$$\frac{\partial f(x,y,\ldots)}{\partial x} = \frac{f(x+a/2) - f(x-a/2)}{a}$$

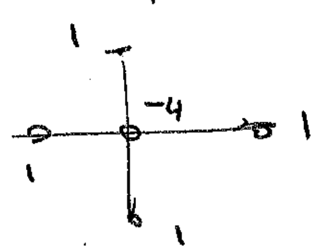
$$\Rightarrow \underbrace{\frac{\partial}{\partial x}\frac{\partial f(x,y,\ldots)}{\partial x}} = \frac{f(x+a)-f(x) - [f(x)-f(x-a)]}{a^2}$$

$$\frac{\partial^2}{\partial x^2} f(x,\ldots) = \frac{f(x+a) - 2f(x) + f(x-a)}{a^2}$$



Usually, by invariance (rot. sym)

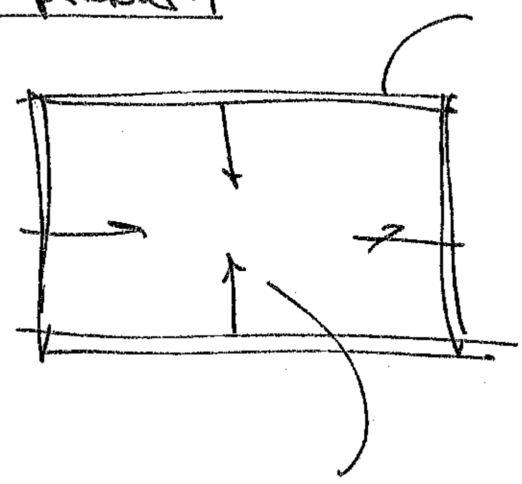
$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\phi(x,y) = \frac{1}{a^2}\Big(\phi(x+a,y) - 2\phi(x,y) + \phi(x-a,y)$$
$$\phi(x,y+a) - 2\phi(x,y) + \phi(x,y-a)\Big)$$



{ similarly in higher dim.

Something deep: you've turned calculus into linear alg.

typical problem

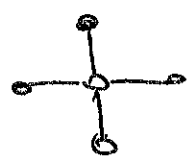


boundary has some fixed values

some PDE (eg $\nabla^2 f = 0$) determines values inside

reminiscent of holographic principle

how to solve?

1. guess
2. use  like "ouija board" to minimize.

relaxation.