Timebomb of approximation method in physics

vijay panchal

September 17, 2022

Contents

1 Introduction

Approximation method is yet essentensial topic in physics. Physicists love approximations, like in functional expansion for getting polynomials for their ease or may be specialized idealization in perticular topic. There is too much advantages of approximation methods like **doing physics** instead going maze of exactness in mathematics. Getting interpretation or more i say knowing system is sometime important then going for regorious mathematics. For example, famous equation of fluid dynamics **Navier-Stoke equation** can be imposible to solve but as physicist they know what it is.

Be aware, that approximation is just approximation. We should remember everytime we do that. Sometime we forgot actual system which is far from ideal. We should know that we are on mission to know nature not just building new theories.

Let's dive into one example, what do i imply by consequences of approximation methods. In classical mechanics, we have 3 or 4 major thoeries, like Centre forces oscillations. In Oscillation theory we studied **Simple Harmonic Osciallations**, but as we are going to see that simple harmonic oscillation is not exactly that simple.

2 Harmonic Oscillator

Classical example of oscillator as we studied in physics is pendulum. By understanding pendulum means understanding most of physics of oscillators, in sense atoms, dipoles or any periodic motion. Let's build our Pendulum. We take small bob (as we can structure) with mass m, which is attached to string of l with rigid wall. Mass m suspended vertically with gravity. We should take nonconservative force like damping to make our system feel more practical.

3 Simulations

Dfsadfsdfd

Sdf

3.1 Animations

I done animation with python's pygame library which is acurate in time. This library make sures that our system follows as real life as possible.

```
def bob1(window,x,y):
    image = pygame.image.load("bitmap2.png")
    window.blit(image, (x,y))

def bob2(window,x,y):
    image = pygame.image.load("bitmap.png")
    window.blit(image, (x,y))

def surface(window,x,y):
    image = pygame.image.load("drawings.png")
    window.blit(image, (x, y))

def position(l,theta):
    return origin_x-10+l*cos((1.5*pi)-theta),origin_y-10-l*sin((1.5*pi)-theta)

def mainloop(window,fps):
    global t
    run = True
    clock = pygame.time.Clock()
    c = 0
    while run:
for event in pygame.event.get():
    if event.type== pygame.QUIT:
run= False
break
clock.tick(fps)

x,y = position(l,exact[c])
x2,y2 = position(l,appro[c])
if c==len(exact):
    break
window.fill("#ffffff")
surface(window,175,origin_y-25)
pygame.draw.aaline(window,color="#5BDEC1",start_pos=(origin_x,origin_y),end_pos=(x+radi
pygame.draw.aaline(window,color="#5BDEC1",start_pos=(origin_x,origin_y),end_pos=(x2+10
blip(window,x,y)
blip2(window,x2,y2)
pygame.display.update()
c+=1
    pygame.quit()

if __name__ == "__main__":
    pygame.init()
    window = pygame.display.set_mode((width,height))
    exact = nonlinear(5,30)
    appro = linear(5,30)
    t = time.perf_counter()
    mainloop(window, 30)
```