Theory Computing Project Notebook

Tingyu Chen

School of Physics and Astronomy

University of Manchester

January 2025

Week 1 - Finite Difference method:

Task 1: Estimating Derivatives:

The derivative is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}.$$

Computationally, we have the forward derivative:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+\delta x) - f(x)}{\delta x}.$$

Example 1:

$f(x) = x^3$ and $f'(x) = 3x^2$ at x=10.

How small should we make δx to get the right derivative at the second decimal place? $\delta x = 10$**-4

Repeat the same process at $x = 5$. Can you explain the difference? Consider the Taylor expansion around x,

$$f(x+\delta x) = f(x) + f'(x)\,\delta x + \frac{1}{2} f''(\xi)\,\delta x^2 \quad \text{, where } \xi \in [x,\, x+\delta x],$$

$$\text{so} \quad \frac{f(x+\delta x) - f(x)}{\delta x} = f'(x) + \frac{1}{2} f''(\xi)\,\delta x.$$

The difference comes from the truncation error in finite-difference approximations. Since the second derivative of the function is $f''(x) = 6x$, the truncation error is proportional to $6x\delta x$. Thus:

At x=10, the second derivative is 60, leading to a larger numerical error.

At x=5, the second derivative is 30, which leads to a half-smaller numerical error.

Use the definition of midpoint derivative,

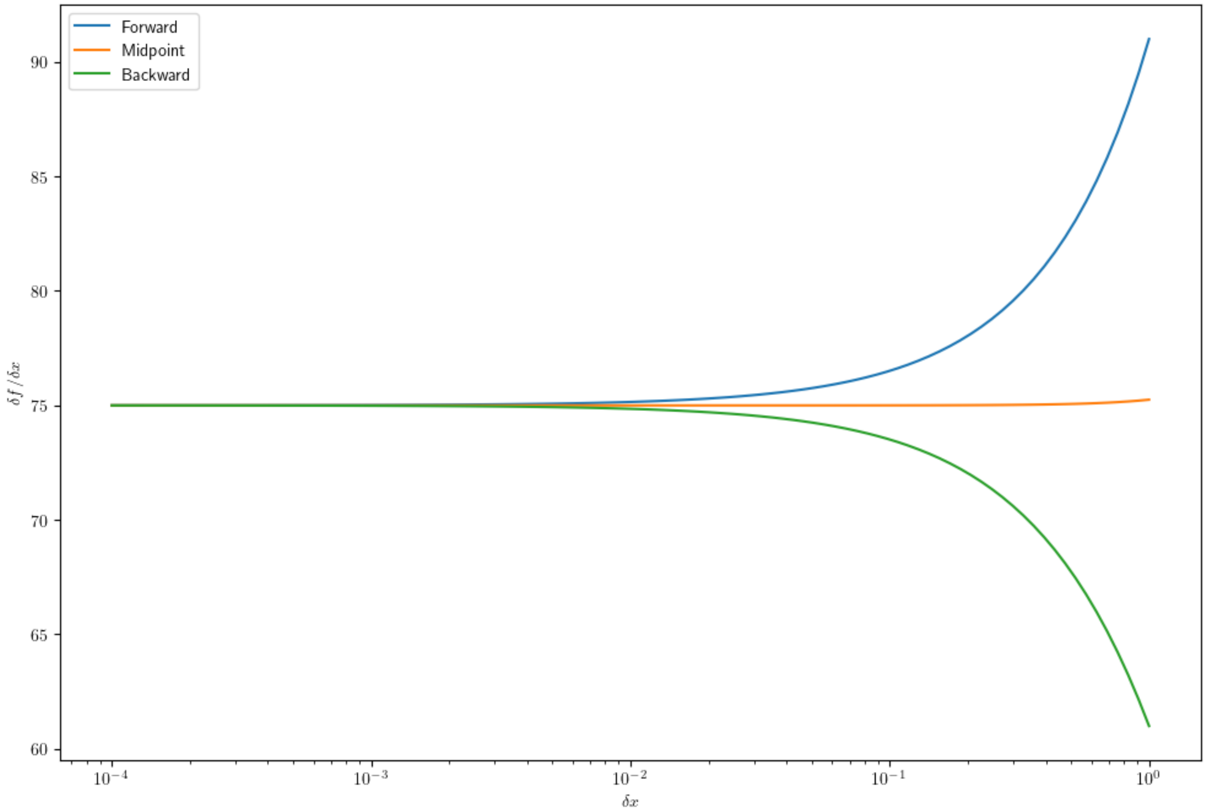
$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+\delta x/2) - f(x-\delta x/2)}{\delta x},$$

that would give an error of order $\mathrm{O}(\delta x^3)$ to repeat the above tasks.

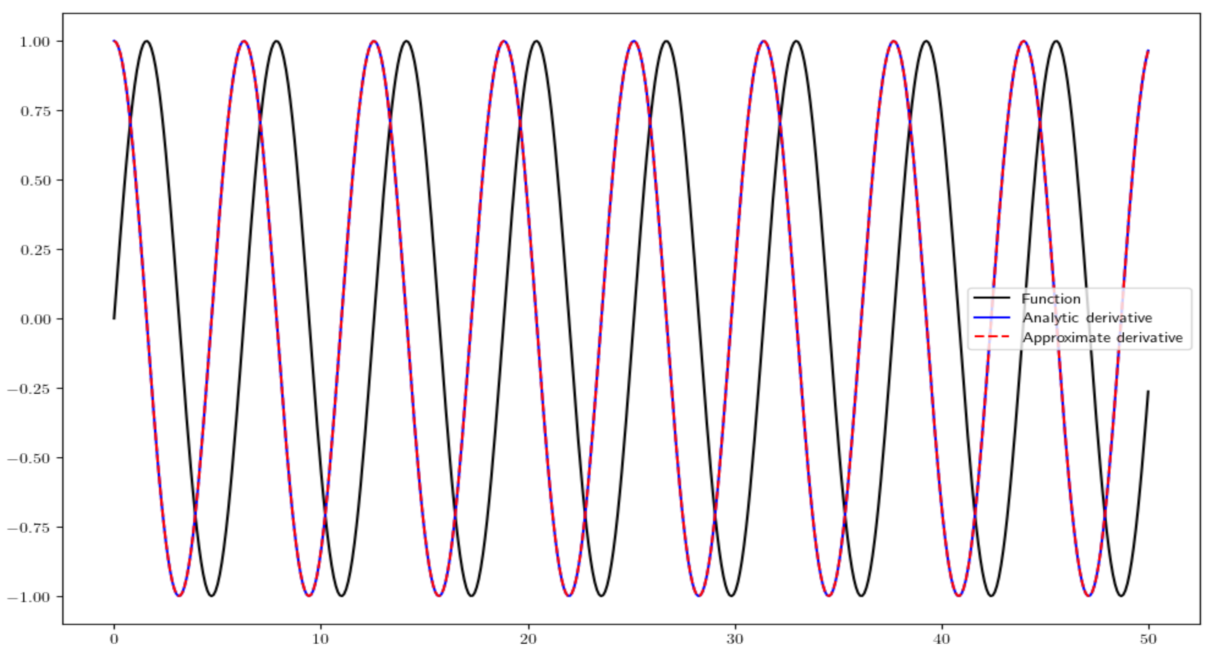
In 9
```python
x = 10
delta_x = 10**-4

print(f"Analytic derivative:    {func_prime(x)}")
print(f"Approximate derivative: {(func(x + delta_x/2) - func(x - delta_x/2)) / delta_x}")
```
Executed at 2025.01.31 15:42:40 in 49ms

```
Analytic derivative:    300
Approximate derivative: 300.0000000020009
```

Below shows the visualization of $\frac{\delta f}{\delta x}$ v.s. δx for the forward, backward and midpoint derivatives:

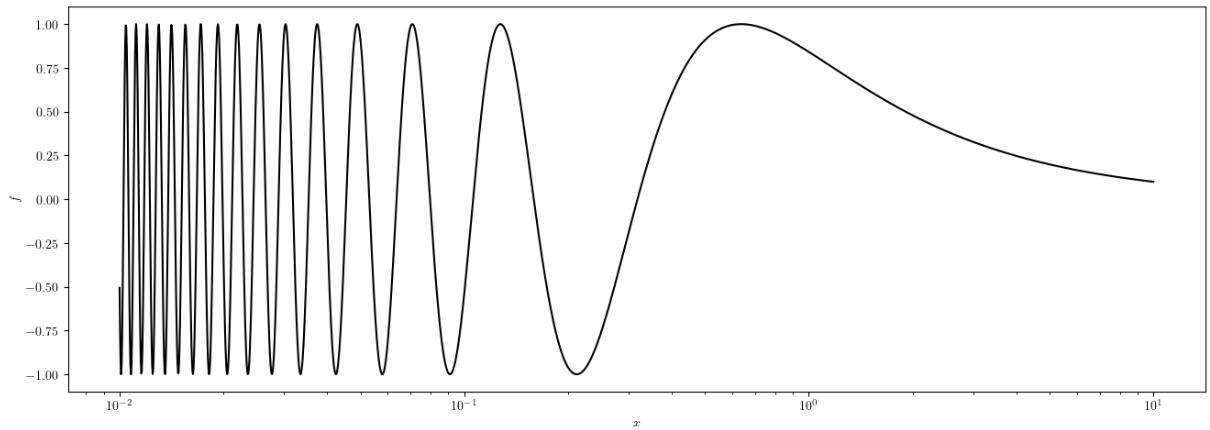


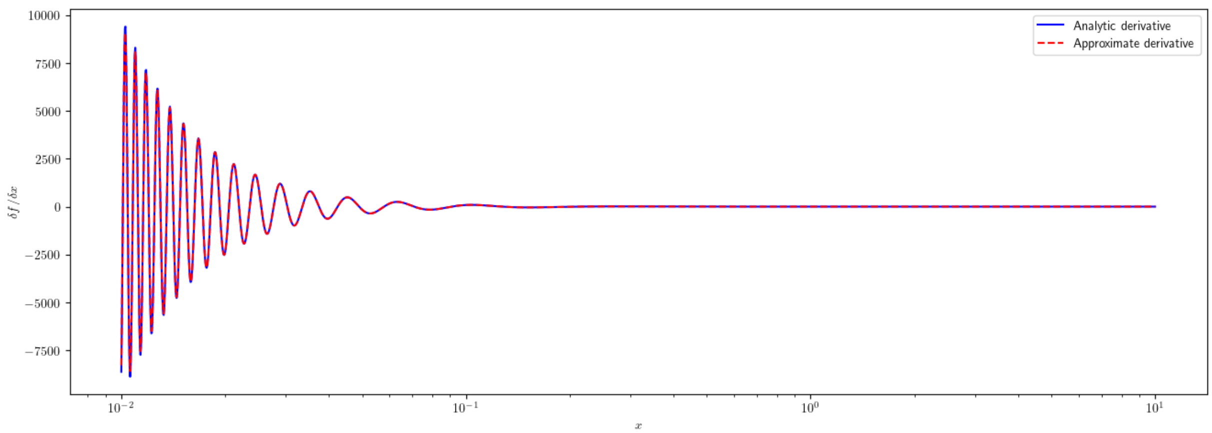
It can be seen that the midpoint derivative converges to the correct answer much faster than the other two. Using this method to visualize derivatives across an entire domain:



Example 2:

f(x)= $x\sin(\frac{1}{x})$ at dx = 10**4.





The main challenge is that sin(1/x) is very oscillatory near zero, and its derivative grows unboundedly, thus a fixed step size $\delta x = 10^{-}4$ may not be "small enough" in that region to capture the rapid changes, especially on a log-scale from x=0.01 all the way to 10.

Task 2: Estimating Integrals:

One possible approximate finite difference formula is the following:

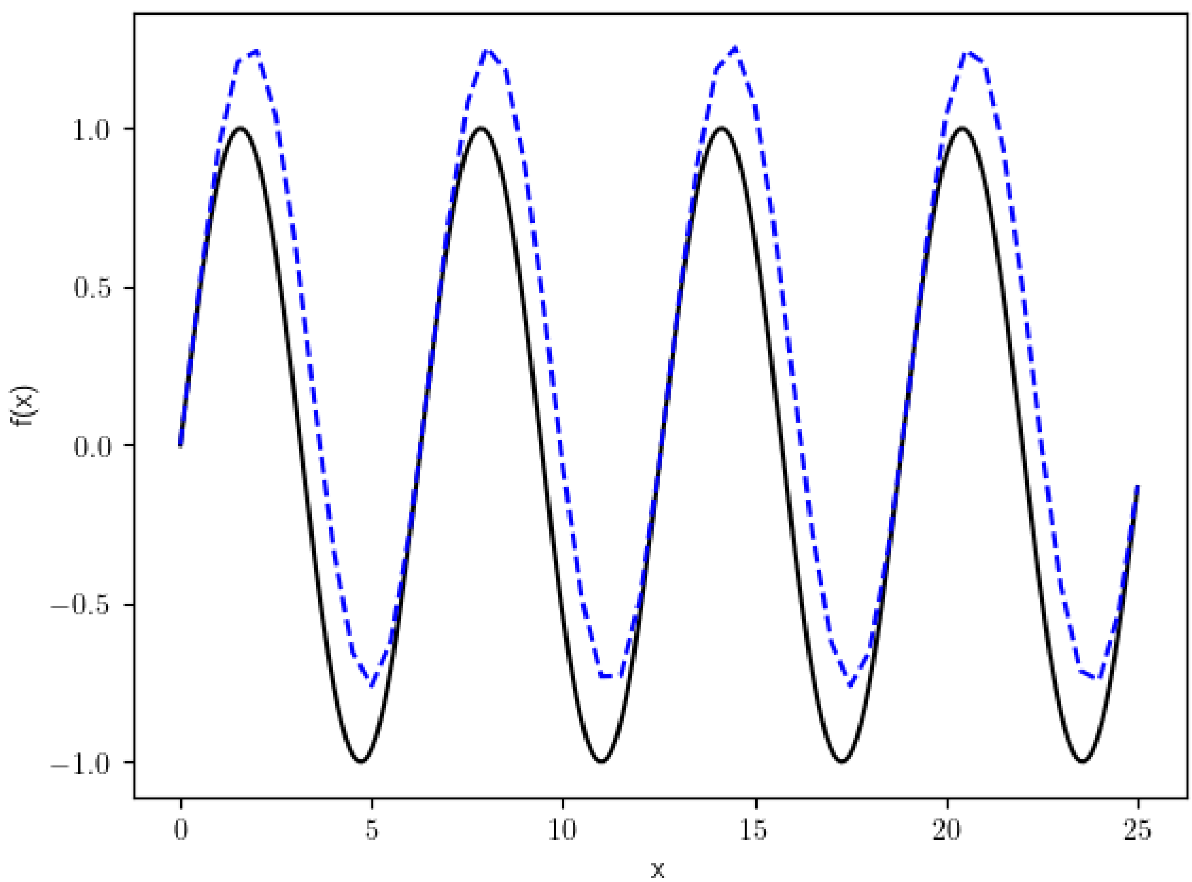
$$\frac{\mathrm{d}f}{\mathrm{d}x} = f'(x) \approx \frac{f(x+\delta x) - f(x)}{\delta x}. \implies f(x+\delta x) = f(x) + f'(x)\delta x.$$

Example 1:

For $f'(x) = \cos x$ with the known integral $f(x) = \sin x$, dx = 0.5, we can determine the approximated integral value by:

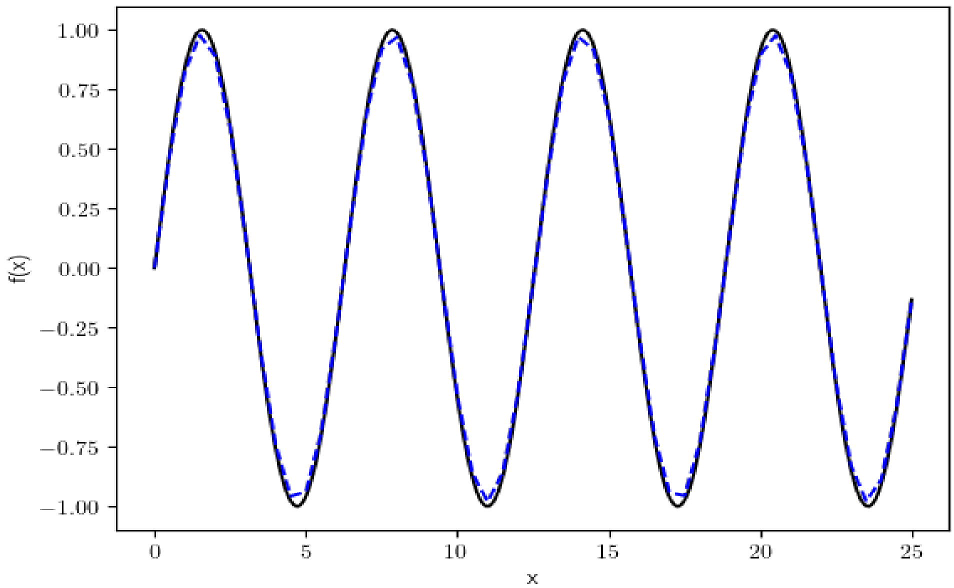
$$I_{n+1} = I_n + f(x_n)\,\delta x$$

the Euler's method, which is only first-order accurate in δx.



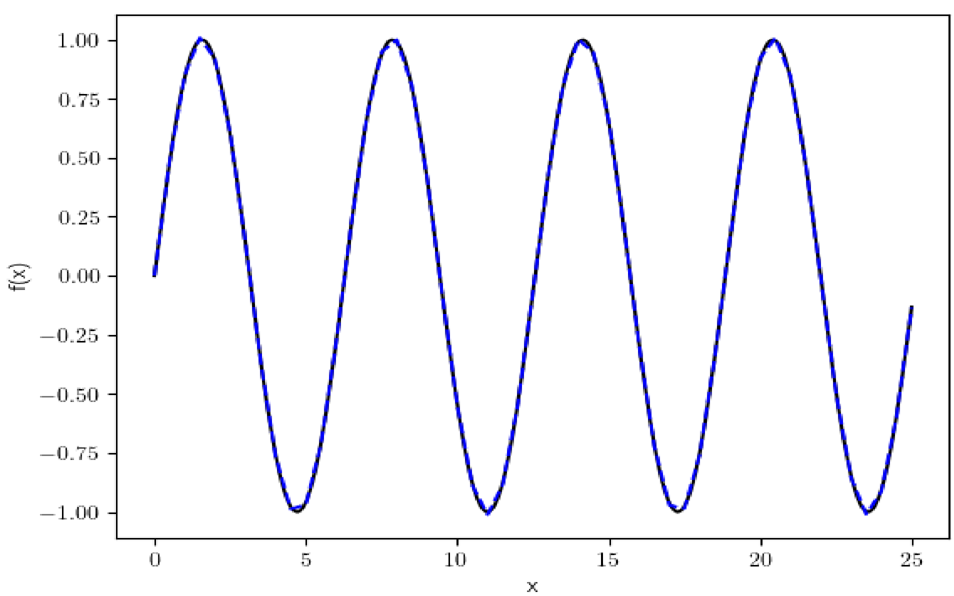
However, by using **the trapezoidal rule**, where we repeat the above taylor expansion to second order, we have:

$$I_{n+1} = I_n + \frac{1}{2}\big(f(x_n) + f\big(x_{n+1}\big)\big)\,\delta x.$$



However, we can do even better, using **the midpoint method (RK2 method)**:

$$I_{n+1} = I_n + f\big(x_n + \frac{\delta x}{2}\big)\delta x$$



Task 3: Solving ODEs:

We will state now without proof the formula for RK4 now for an entire vector $\vec{x}$ of variables:

$$\vec{k_1} \equiv f(t,\, \vec{x})\delta t,$$

$$\vec{k_2} \equiv f\left(t + \frac{\delta t}{2},\, \vec{x} + \frac{\vec{k_1}}{2}\right)\delta t,$$

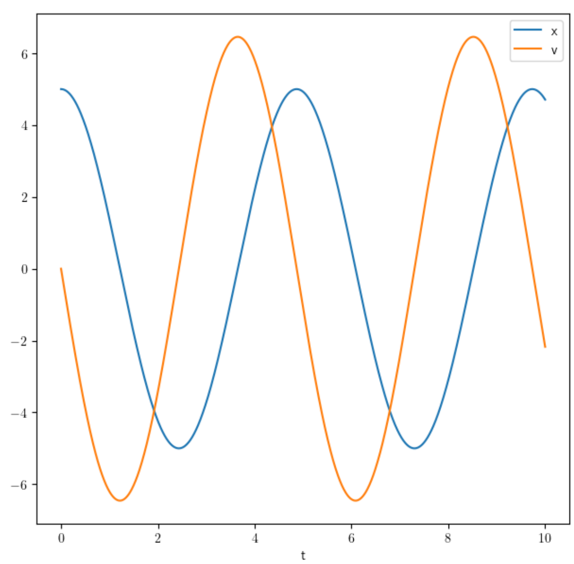
$$\vec{k_3} \equiv f\left(t + \frac{\delta t}{2},\, \vec{x} + \frac{\vec{k_2}}{2}\right)\delta t,$$

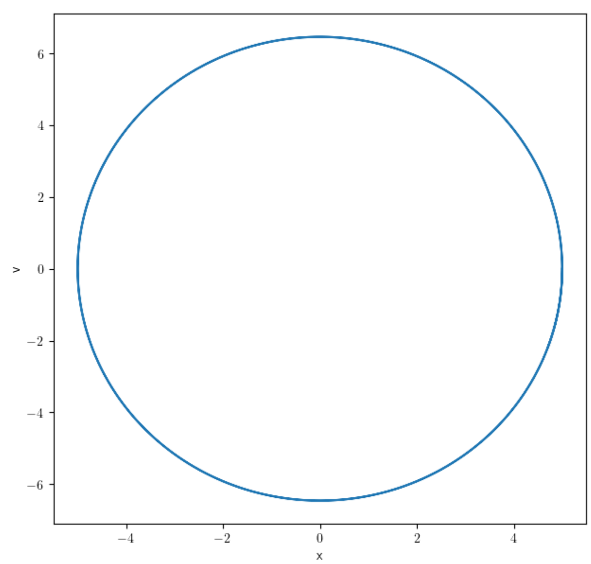
$$\vec{k_4} \equiv f(t + \delta t,\, \vec{x} + \vec{k_3})\delta t,$$

$$\vec{x}(t + \delta t) = \vec{x}(t) + \frac{1}{3}\left(\frac{\vec{k_1}}{2} + \vec{k_2} + \vec{k_3} + \frac{\vec{k_4}}{2}\right).$$

Example 1: The spring-mass system:

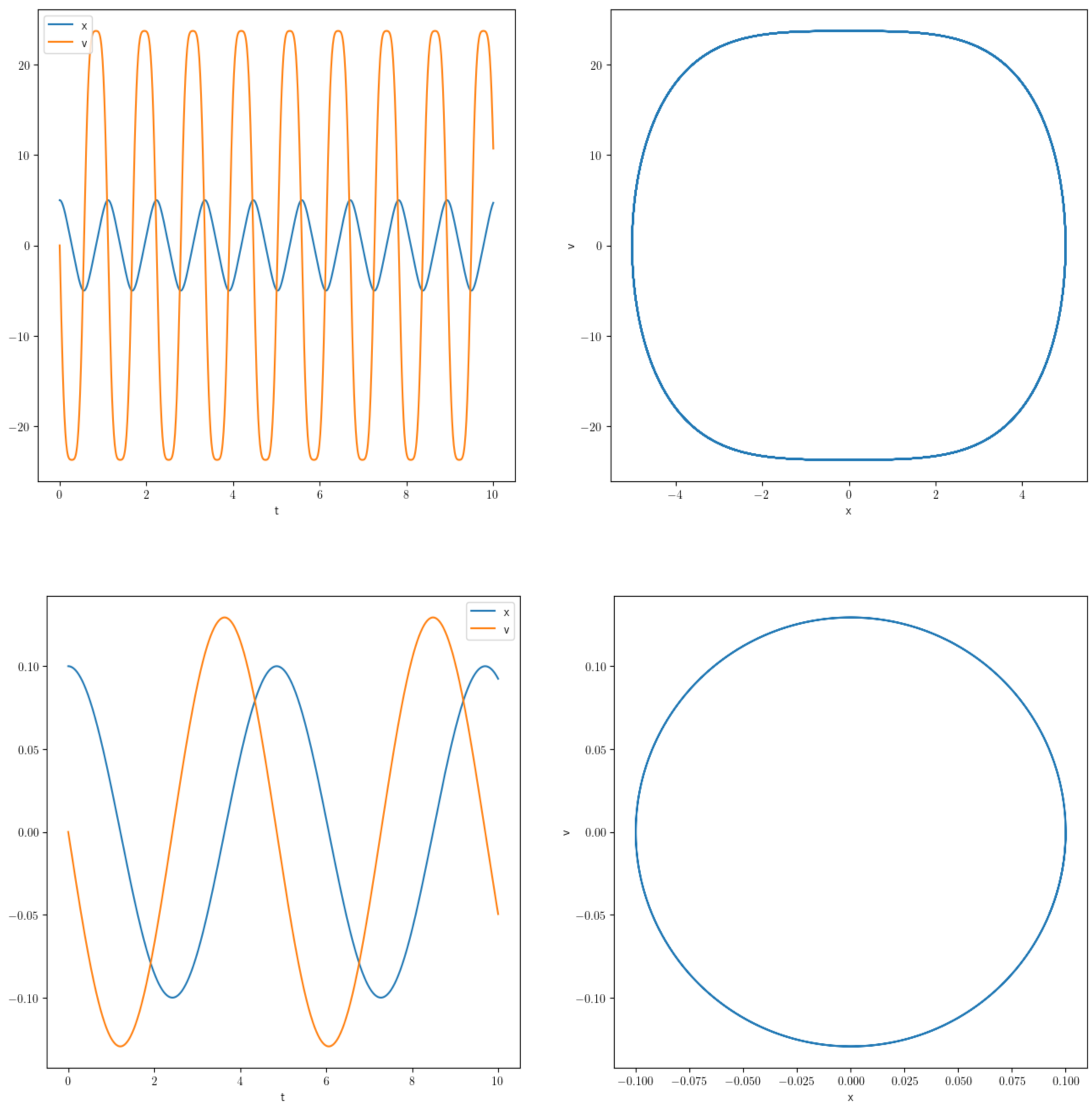
The sinusoidal motion of x(t) and v(t) is shown below with a phase diagram that plots the x-v plane where the Initial state is [x, v] = [5, 0] at time = 0.





If we add a cubic perturbation to the spring force,

$$F = -kx - kx^3$$

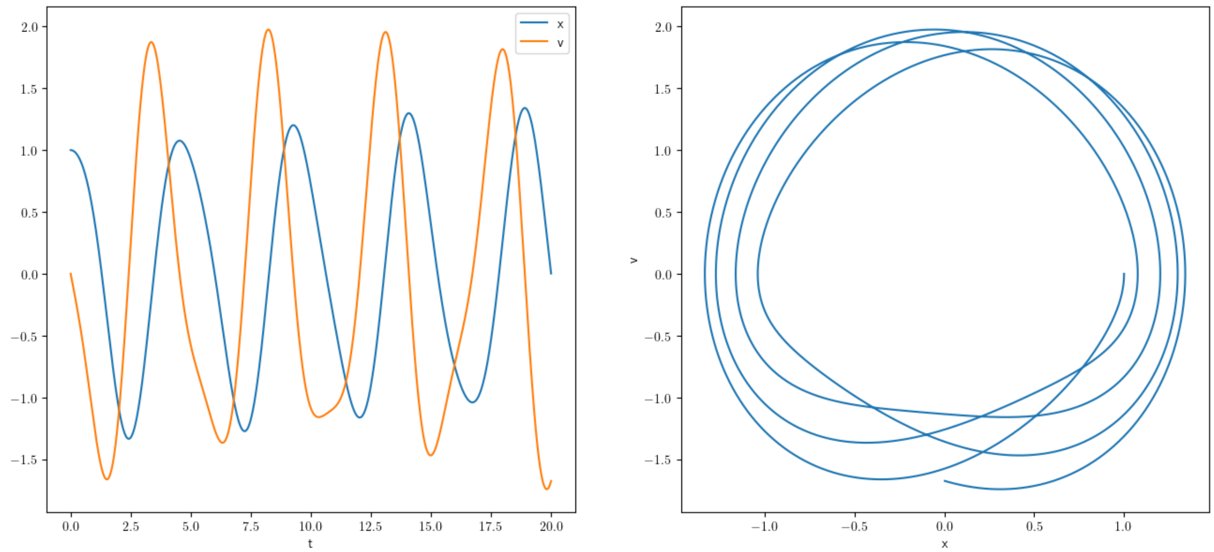


With the extra cubic term, the restoring force is no longer purely linear in x, and a distorted waveform compared to the pure harmonic oscillator can be seen. Also, in phase plane, a linear spring shows closed elliptical orbits with a cubic term, these orbits deviate from perfect ellipses. For $|x| << 1$, The cubic term $(-kx^3)$ is relatively small compared to the linear term (-kx). Consequently, the system behaves nearly like a linear mass-spring oscillator, with only minor differences in period or phase-plane shape, as shown above. Whereas, for large-amplitude regime, The cubic term becomes significant, and noticeable differences in the oscillation frequency, waveform shape, and phase-plane trajectories can be observed.

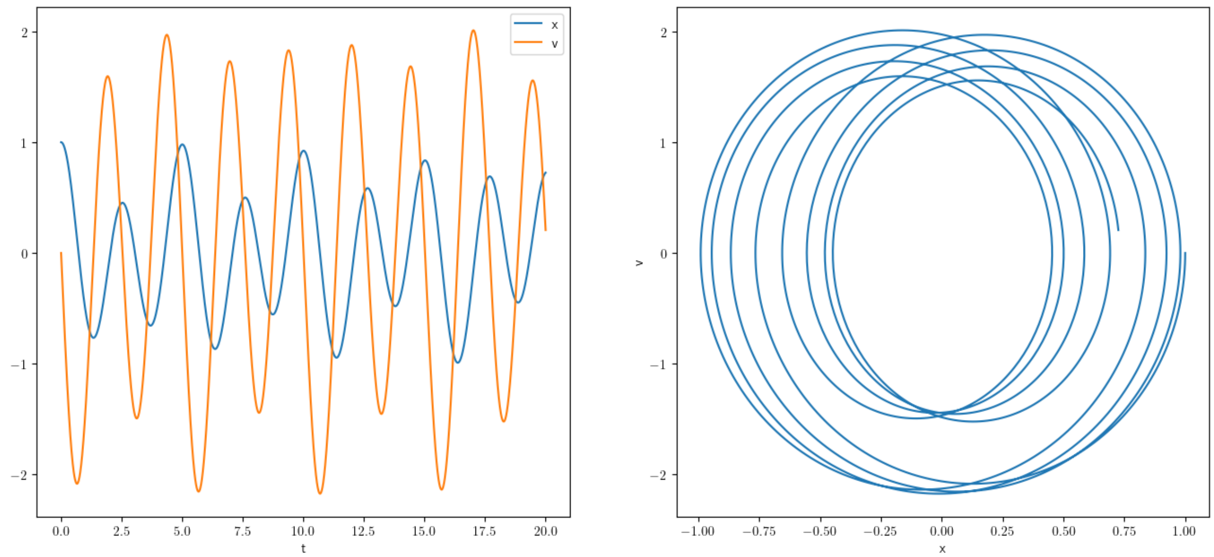
For the two-mass, three-spring system, we have:

$$\begin{cases} \dot{x}_1 = v_1, \\ \dot{v}_1 = \dfrac{-k_1 x_1 + k_2(x_2 - x_1)}{m}, \\ \dot{x}_2 = v_2, \\ \dot{v}_2 = \dfrac{k_2(x_1 - x_2) - k_3 x_2}{m}. \end{cases}$$

For the symmetric case, e.g. both masses displaced to +1 positions:



For the asymmetric case, e.g. first mass +1, second mass -1 positions:



For the symmetric initial condition, both masses move in phase and the motion has a certain natural frequency, corresponding to the symmetric normal mode. While for the asymmetric initial condition, the masses move out of phase, so the middle spring stretches and compresses significantly that correspond to the antisymmetric normal mode. In a linear system, any arbitrary motion can be seen as a superposition of these two modes. This can also be seen from the graph, in x-t plot, they will be in sync for symmetric case and $x_2(t)$ might track $-x_1$(t) for asymmetric case. While for the phase diagram, the pure modes will form clean elliptical loops as expected.

Challenge Question:

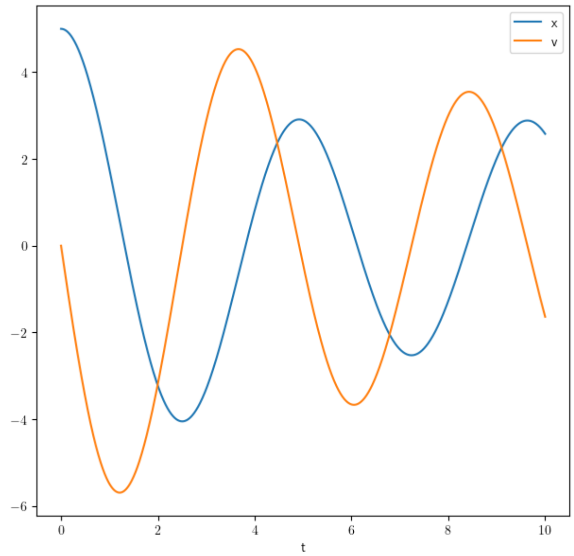
Considering the full forced damped oscillator:

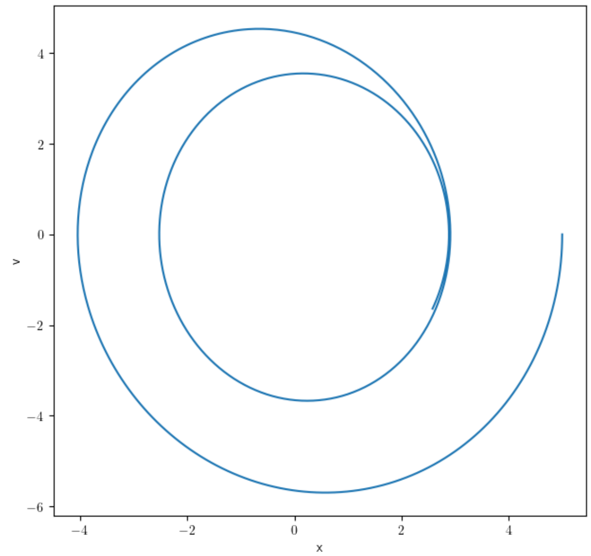
$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = F_0 \cos(\omega t),$$

Thus we have:

$$\begin{cases} x'(t) = v(t), \\ v'(t) = -\dfrac{b}{m}v(t) - \dfrac{k}{m}x(t) + \dfrac{F_0}{m}\cos(\omega t). \end{cases}$$

Substituted to the RK4, with assuming $b = 0.3$, $F_0 = 2$, and $\omega = \pi/4$, we have computationally:





While analytically, we can rewrite the equation as

$$\ddot{x}(t) + 2\alpha\dot{x}(t) + \omega_0^2 x(t) = \frac{F_0}{m}\cos(\omega t),$$

where

$$\alpha = \frac{b}{2m}, \quad \omega_0 = \sqrt{\frac{k}{m}}.$$

By substituting the numbers, we have $\alpha = \frac{0.3}{2\times 3} = 0.05$ and $\omega_0 = \sqrt{\frac{5}{3}} = 1.29099$

The analytic solution is composed of a complementary function and a particular integral by $\cos\omega t$.

For the CF,

$$x_c(t) = e^{-\alpha t}\left[A\cos(\beta t) + B\sin(\beta t)\right],$$

with

$$\beta = \sqrt{\omega_0^2 - \alpha^2} \quad \text{for } \omega_0 > \alpha \text{ (underdamped case).}$$

and the PI,

$$x_p(t) = A(\omega)\cos(\omega t - \phi),$$

where

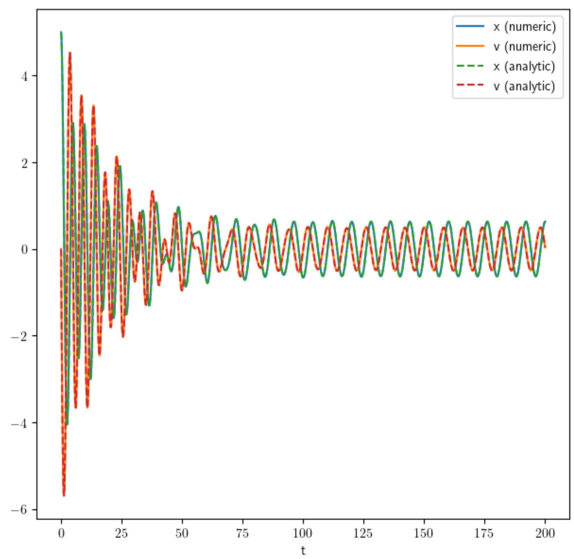
$$A(\omega) = \frac{F_0/m}{\sqrt{(\omega_0^2 - \omega^2)^2 + (2\alpha\omega)^2}}, \quad \tan(\phi) = \frac{2\alpha\omega}{\omega_0^2 - \omega^2}.$$

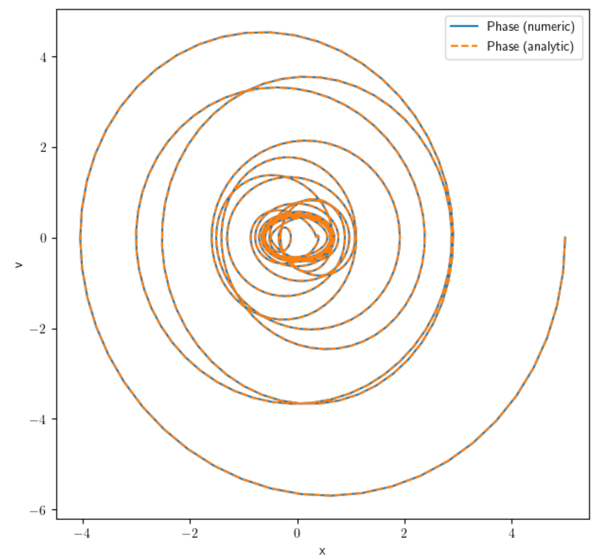
We can determine A and B from the initial conditions, where $x(0) = 5, \dot{x}(0) = 0$.

It can be found that

$$A = 5 - A(\omega)\cos\phi \text{ and } B = \frac{\alpha A - A(\omega)\omega\sin(\phi)}{\beta}.$$

Therefore, by using Python, we can acquire the graphs that compare the analytical and computational methods:





The step size is 0.1 and step number is 2000 for this case, we can clearly see the close agreement for the two methods. And after a long period, the motion reaches the steady-state regime as we expected.