DOING PHYSICS WITH PYTHON

[2D] DYNAMICAL SYSTEMS

LINEAR PLANAR SYSTEMS

Mass / Spring System

Ian Cooper
matlabvisualphysics@gmail.com

DOWNLOAD DIRECTORIES FOR PYTHON CODE

Google drive

GitHub

ds1401.py

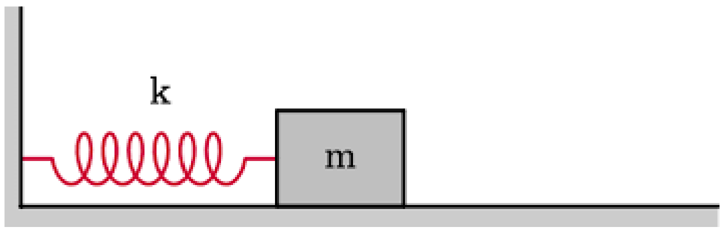
Jason Bramburger
Linear Planar Systems - Dynamical Systems | Lecture 14

https://www.youtube.com/watch?v=b8eJb5uwNZI

INTRODUCTION

In this paper, the basics of [2D] linear planar dynamical systems is discussed using the example of a mass – spring system. One can solve the ODE for the system numerically and compare it with a linear algebra approach where the solution is expressed in terms of eigenvalues and eigenvectors of the matrices defining the system. From my point of view the numerical solution is a much better approach than the more traditional linear algebra method.

A mass-spring system is a fundamental mechanical oscillator consisting of a mass attached to a spring.



displacement $x = x_1$ velocity $v \equiv \dot{x}_1 = x_2$ acceleration $a \equiv \ddot{x}_1 = \dot{x}_2$

It demonstrates the principle that the restoring force from the spring is proportional to the displacement, a concept described by Hooke's Law if $F = m\ddot{x} = -k\,x$.

These systems are crucial for understanding oscillating motion and have wide-ranging real-world applications, from vehicle suspension systems to simulating complex motions in computer graphics.

If there is no damping then the system will exhibit simple harmonic motion when displaced from its equilibrium position. The period *T* of the oscillation is

$$\omega^2 = \frac{k}{m} \quad \omega = \sqrt{\frac{k}{m}} \quad T = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{m}{k}}$$

In our model, we can introduce a damping term where the damping force is proportional to the velocity of the mass. The constant of proportionality is called the damping constant q. If the motion is damped, the oscillations will decay to zero. The ODE for the damped mass – spring is

$$F = m\ddot{x} = -k\,x - q\,\dot{x}$$

The second order ODE for the damped mass – spring system is

$$\ddot{x} = -\frac{k}{m}x - \frac{q}{m}\dot{x}$$

This needs to be expressed as two first order ODE

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1 - \frac{q}{m}x_2$$

where x_1 is the displacement *x*, and x_2 is the velocity *v*.

The system equations are solved in Python using the function **odeint**.

Let $K_1 = -k/m$ and $K_2 = -q/m$ then the ODEs become

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= K_1 x_1 + K_2\, x_2\end{aligned}$$

In a linear algebraic approach, the system equations are expressed in matrix form

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}0 & 1\\ K_1 & K_2\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix}0 & 1\\ K_1 & K_2\end{pmatrix} \quad \dot{\mathbf{x}} = \begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} \quad \mathbf{x} = \begin{pmatrix}x_1\\ x_2\end{pmatrix}$$

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x}$$

and the solution is

$$\begin{pmatrix}x(t)\\ v(t)\end{pmatrix} = \begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} = c_1\begin{pmatrix}f_{00}\\ f_{10}\end{pmatrix}e^{L_1 t} + c_2\begin{pmatrix}f_{10}\\ f_{11}\end{pmatrix}e^{L_2 t}$$

$$x(t) = c_1\, f_{00}\, e^{L_1 t} + c_2\, f_{10}\, e^{L_2 t}$$

$$v(t) = c_1\, f_{10}\, e^{L_1 t} + c_2\, f_{11}\, e^{L_2 t}$$

where L_1 and L_2 are the eigenvalues of the matrix **A** and **F** is the eigenvector matrix

$$\mathbf{F} = \begin{pmatrix}f_{00} & f_{01}\\ f_{10} & f_{11}\end{pmatrix} \quad \mathbf{F}_1 = \begin{pmatrix}f_{00}\\ f_{10}\end{pmatrix} \quad \mathbf{F}_2 = \begin{pmatrix}f_{10}\\ f_{11}\end{pmatrix}$$

The eigenvalues and eigenvector matrix are found using the Python function **eig(A)**. The *c* coefficients *c* found by solving the equations for *x*(*t*) and *y*(*t*) using the Python function **solve**.

Python Code from **ds1401.py** to find eigenvalues and analytical solutions:

```
A = array([[0,1],[K,0]])
L, F = eig(A)
F0 = [[F[0,0],F[1,0]]]
F1 = [[F[0,1],F[1,1]]]
X = np.transpose(u0)
c = np.linalg.solve(F,X)
xA = c[0]*exp(L[0]*t)*F[0,0] + c[1]*exp(L[1]*t)*F[0,1]
vA = c[0]*exp(L[0]*t)*F[1,0] + c[1]*exp(L[1]*t)*F[1,1]
```

SIMULATIONS

The solutions using the numerical method by solving the ODEs and the linear algebra method produce identical results.

SIMPLE HARMONIC MOTION (No damping)

$$m = 1.0 \quad k = 4\pi^2 \quad q = 0 \quad x_0 = 0.8 \quad y_0 = 0 \Rightarrow T = 1.00 \text{ s}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -39.48 & 0 \end{pmatrix}$$

$$\lambda_0 = 0 + 6.28\,j \qquad \lambda_1 = 0 - 6.28\,j$$

$$\mathbf{F}_0 = \begin{pmatrix} 0 - 0.1572\,j \\ 0.988 + 0\,j \end{pmatrix} \qquad \mathbf{F}_1 = \begin{pmatrix} 0 + 0.1572\,j \\ 0.988 + 0\,j \end{pmatrix}$$

$\lambda = a + b\,j \quad a = 0 \quad b \neq 0 \;\Rightarrow$ motion is purely oscillatory without any decay in the motion towards the fixed point at the Origin in the phase portrait (*x* vs *v*).

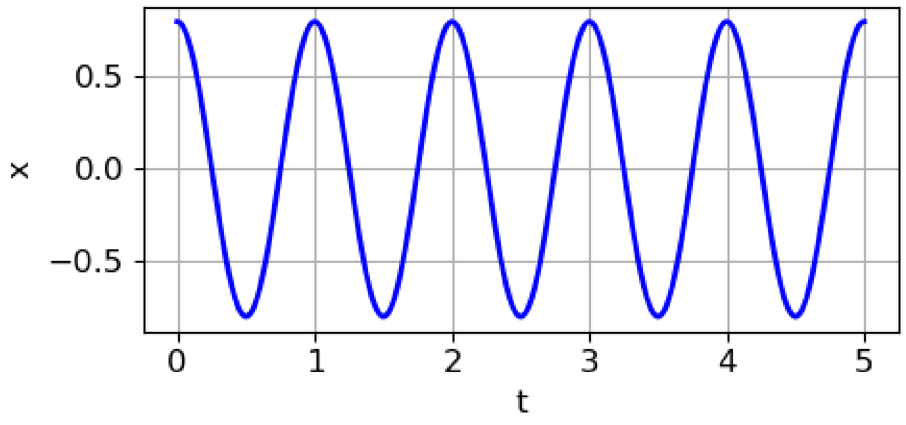


Fig. 1.1. Displacement: Simple harmonic motion with period $T = 1.0$.

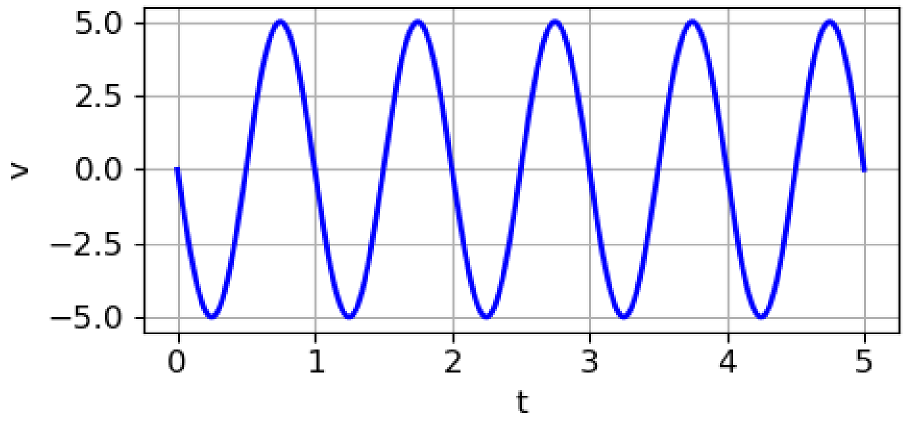


Fig. 1.2. Velocity.

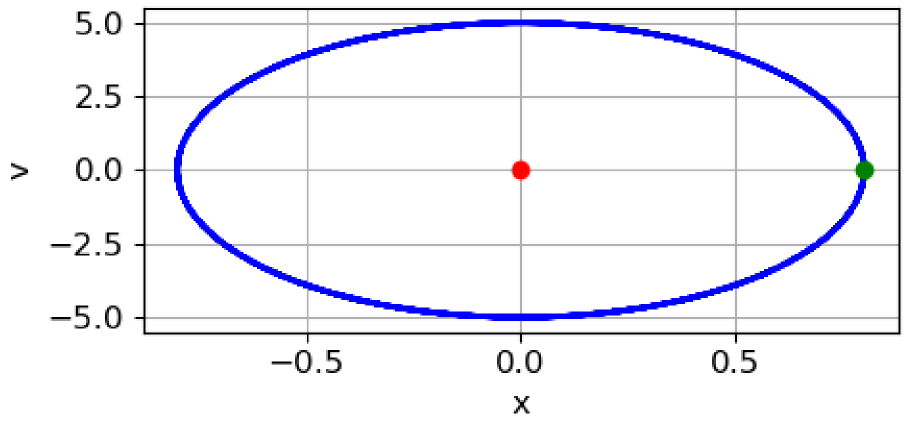


Fig. 1.3. Phase plane: the Origin (0, 0) is the fixed point (centre).

The Origin is the centre of the orbit in phase space. The trajectory is a closed orbit (limit cycle).

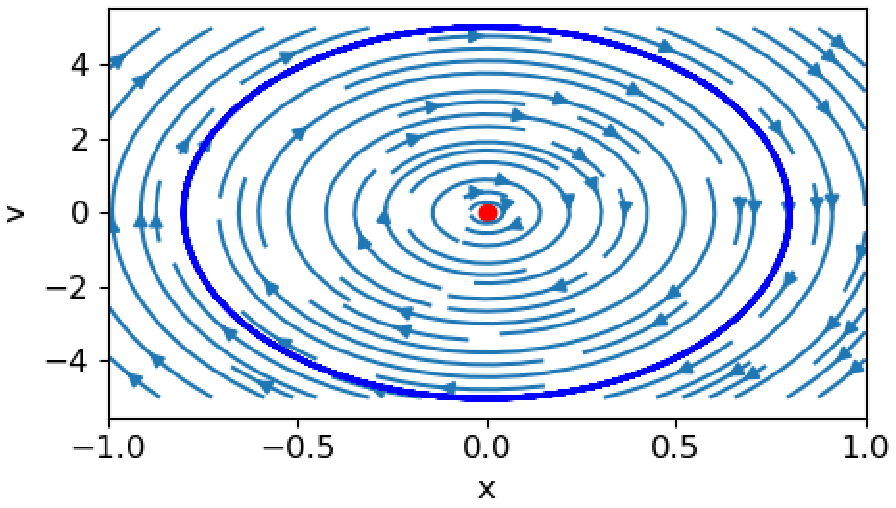


Fig. 1.4. Phase plane: streamline plot (vector field) and trajectory. The trajectory in the phase plan is an ellipse. The flow is clockwise. Using this plot you can to predict the orbit for any initial condition.

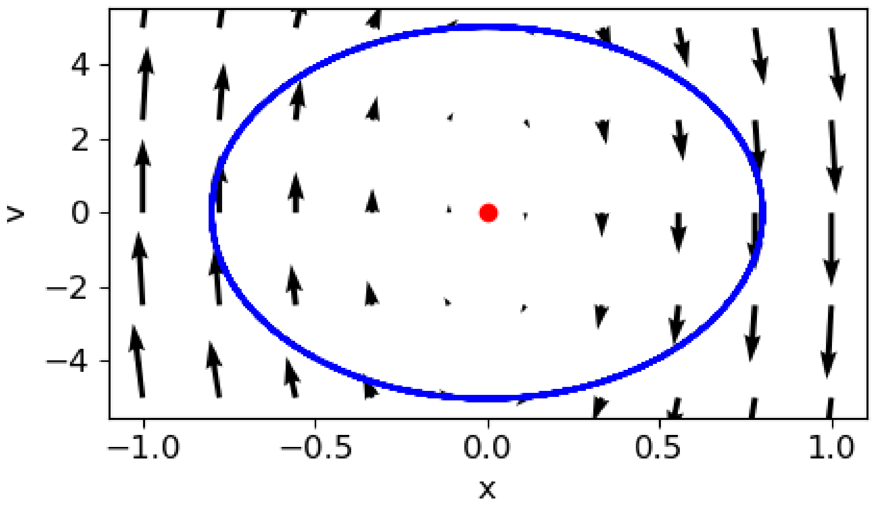


Fig. 1.5. Phase plane: quiver plot. The streamplot gives a much better view of the vector field.

DAMPED HARMONIC MOTION

$$m = 1.0 \quad k = 4\pi^2 \quad q = 1.20 \quad x_0 = 0.8 \quad y_0 = 1.20 \;\Rightarrow\; T = 1.00 \text{ s}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -39.48 & -1.20 \end{pmatrix}$$

$$\lambda_0 = -0.6 + 6.25\,j \qquad \lambda_1 = -0.6 - 6.25\,j$$

$$\mathbf{F}_0 = \begin{pmatrix} -0.015 - 0.1565\,j \\ 0.988 + 0\,j \end{pmatrix}$$

$$\mathbf{F}_1 = \begin{pmatrix} -0.015 + 0.1565\,j \\ 0.988 + 0\,j \end{pmatrix}$$

$\lambda = a + b\,j \quad a < 0 \quad b \ne 0 \;\Rightarrow$ motion is oscillatory where the orbit is drawn to the fixed point at the Origin with the amplitude of the oscillation decaying exponentially to zero.

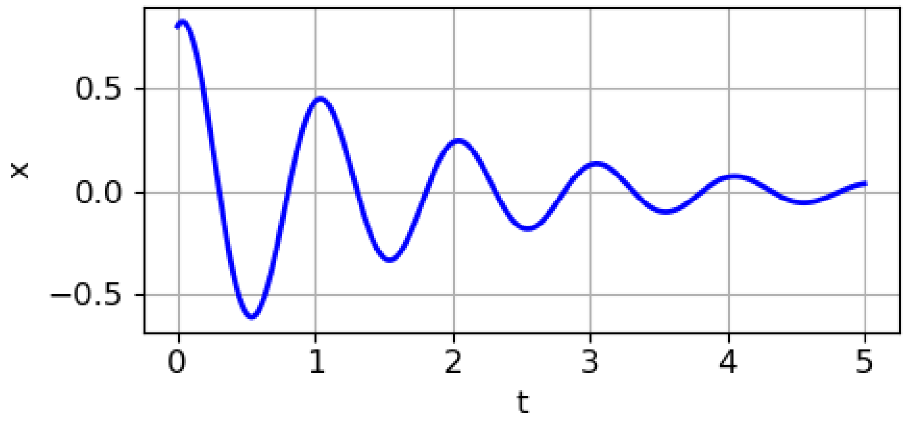


Fig. 2.1. Displacement: Simple harmonic motion with period $T = 1.0$.

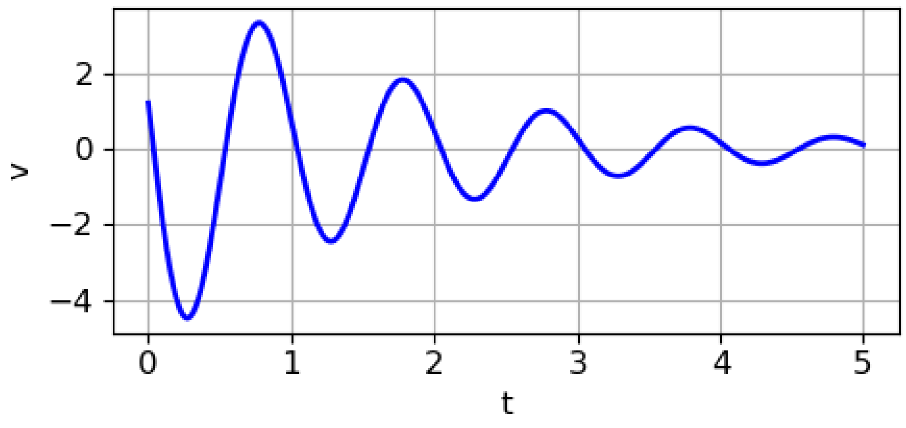


Fig. 2.2. Velocity.

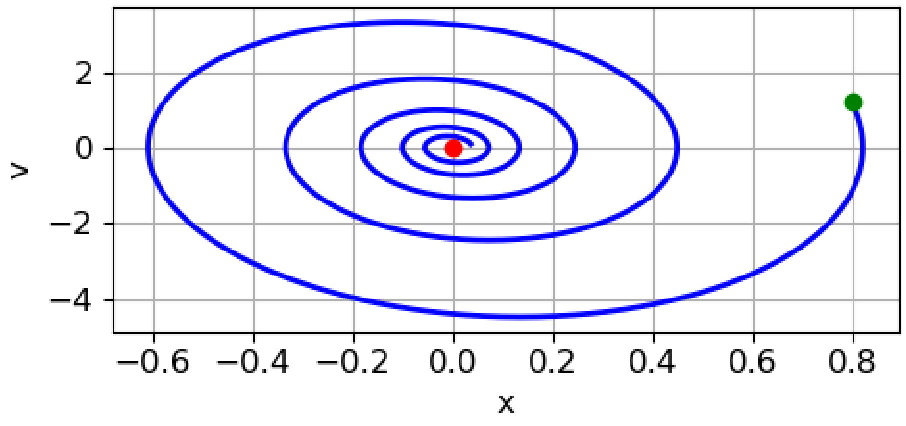


Fig. 2.3. Phase plane: the Origin (0, 0) is the fixed point (centre). The Origin is the centre of the orbit in phase space. The displacement and the velocity both decay to zero.

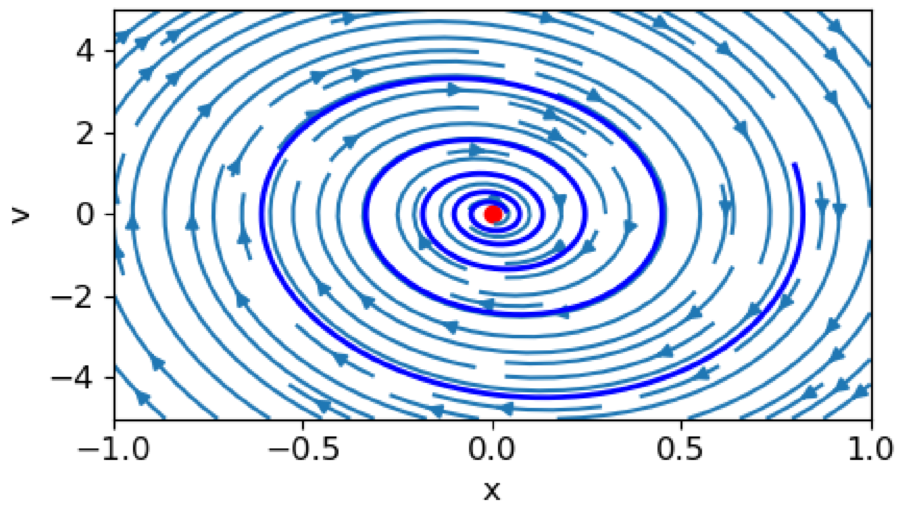


Fig. 2.4. Phase plane: streamline plot (vector field) and trajectory. The trajectory in the phase plan is an ellipse. The flow is clockwise. Using this plot you can to predict the orbit for any initial condition.

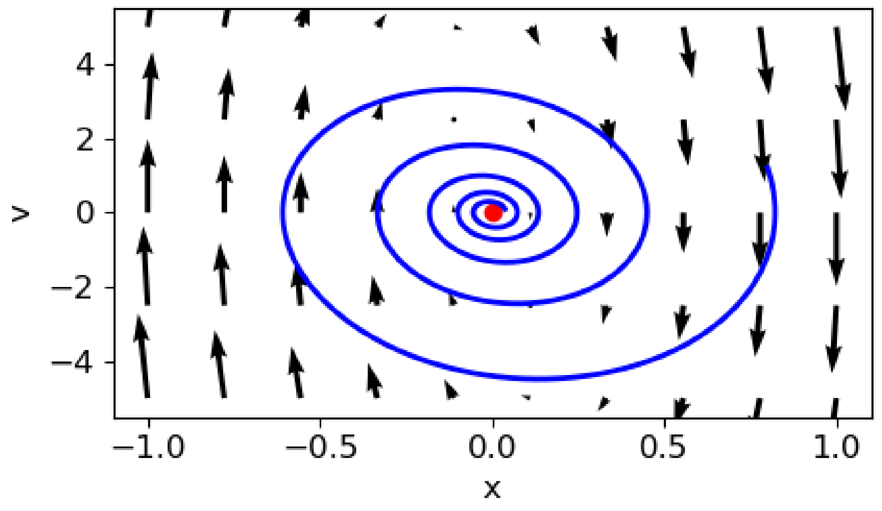


Fig. 2.5. Phase plane: quiver plot. The streamplot gives a much better view of the vector field.