DOING PHYSICS WITH PYTHON

COMPLEX SYSTEMS

[2D] NONLINEAR DYNAMICAL SYSTEMS

Ian Cooper

Please email me any corrections, comments, suggestions or additions: **matlabvisualphysics@gmail.com**

DOWNLOAD DIRECTORIES FOR PYTHON CODE

[Google drive]

[GitHub]

cs210.py cs211.py

This Python Code solves a pair of nonlinear ODEs in x and y. The solution gives the time evolution of the two variables and the phase portrait (quiver plot and streamplot) using nullclines, vector fields, and eigenvectors. One can find and classify critical points in the phase plane.

Reference

Stephen Lynch
Dynamical Systems with Applications using Python

INTRODUCTION

This article considers how Python can be used to solve [2D] nonlinear dynamical systems. The [2D] systems are described by a pair of ordinary differential equations (ODEs) in x and y. The ODEs are solved numerically using the Python function **odeint**. The solutions for x and y are displayed graphically as time evolution plots and phase portrait plots. For a dynamical system in two-dimensions the ODEs can be expressed as

$$\dot{x} = f(x, y) \qquad \dot{y} = g(x, y)$$

The phase portrait is a [2D] figure showing how the qualitative behaviour of system is determined as x and y vary with t. With the appropriate number of trajectories plotted, it should be possible to determine where any trajectory will end up from any given initial condition.

The direction field or **vector field** gives the gradients dy dx. The slope of the trajectories at each point in the vector field is given by

$$\frac{dy}{dx} = \frac{\dot{y}}{\dot{x}}$$

The contour lines for which dy/dx is a constant are called **isoclines**. The contour lines for which $dy/dt = 0$ and $d/xdt = 0$ are called ***nullclines***. Isoclines may be used to help with the construction of the phase portrait. For example, the nullclines for which $\dot{x} = 0$ and $\dot{y} = 0$ are used to determine where the trajectories have vertical and horizontal tangent lines, respectively. If $\dot{x} = 0$, then there is no motion horizontally, and trajectories are either stationary or move vertically. When $\dot{y} = 0$, then there is no motion vertically, and trajectories are either stationary or move horizontally.

An **equilibrium** occurs at **critical points** or **fixed points** (x_C, y_C) of a dynamical system generated by system of ordinary differential equations (ODEs) where a solution that does not change with time.

$$\dot{x} = f(x_C, y_C) = 0 \qquad \dot{y} = g(x_C, y_C) = 0$$

The stability of typical equilibria of smooth ODEs is determined by the sign of real part of eigenvalues of the Jacobian matrix. These eigenvalues are often referred to as the eigenvalues of the equilibrium. In [2D] systems the Jacobian matrix is

$$\mathbf{J}(x_C, y_C) = \begin{pmatrix} \partial f / \partial x & \partial f / \partial y \\ \partial g / \partial x & \partial g / \partial y \end{pmatrix}_{|x=x_C, y=y_C}$$

and has two eigenvalues, which are either both real or complex-conjugates. The eigenvalues and eigenfunctions can be found using the Python function **eig**. A critical point is called *hyperbolic* if the real part of the eigenvalues of the Jacobian matrix $\mathbf{J}(x_C, y_C)$ is nonzero. If the real part of either of the eigenvalues of the Jacobian is equal to zero, then the critical point is called *nonhyperbolic*.

Eigenvalues λ_1, λ_2	Stability of critical point (equilibrium or fixed points)
distinct, real, and positive $\lambda_1 \neq \lambda_2 \quad \lambda_1 > 0 \quad \lambda_2 > 0$	Unstable node
Both eigenvalues zero $\lambda_1 = \lambda_2 = 0$	Unstable
distinct, real, and negative $\lambda_1 \neq \lambda_2 \quad \lambda_1 < 0 \quad \lambda_2 < 0$	Stable node
One eigenvalue is positive and the other negative $\lambda_1 > 0 \quad \lambda_2 < 0$	Saddle point
Repeated eigenvalues $\lambda_1 = \lambda_2 > 0$	Unstable If there are two linearly independent eigenvectors, then the critical point is called a singular node. If there is one linearly independent eigenvector, then the critical point is called a degenerate node.
Repeated eigenvalues $\lambda_1 = \lambda_2 < 0$	Stable
Complex eigenvalues $\lambda = a + b\,j \quad b \neq 0$	
$a > 0$	Unstable oscillator: amplitude grows with time
$a = 0$	Stable: undamped oscillator
$a < 0$	Stable: damped oscillator

Examples are presented to illustrate the main concepts of [2D] dynamics systems.

Example 1 **cs210.py**

$$\dot{x} = x \qquad \dot{y} = x^2 + y^2 - 1$$

Critical point 0(xC, yC) **(0 , 1.000)**

Jacobian matrix J0

[[1 0]
 [0 2]]

Eigenvalues J0 **(1.000 , 2.000)**

Eigenfunctions J0

[[1. 0.]
 [0. 1.]]

Eigenvalues: two distinct positive eigenvalues and hence the critical point is an **unstable node**.

Critical point 1(xC, yC) **(0 , -1.000)**

Jacobian matrix J1

[[1 0]
 [0 -2]]

Eigenvalues J1 **(1.000 , -2.000)**

Eigenfunctions J1

[[1. 0.]
 [0. 1.]]

Eigenvalues: one positive and one negative eigenvalue, and so this critical point is a **saddle point**.

The Jacobian matrices for each critical point are in diagonal form, and the eigenvectors are $(1, 0)^T$ and $(0, 1)^T$ for both critical points. Thus, in a small neighbourhood around each critical point, the stable and unstable manifolds are tangent to the lines generated by the eigenvectors through each critical point. Therefore, near each critical point the manifolds are horizontal $(1, 0)^T$ and vertical $(0, 1)^T$.

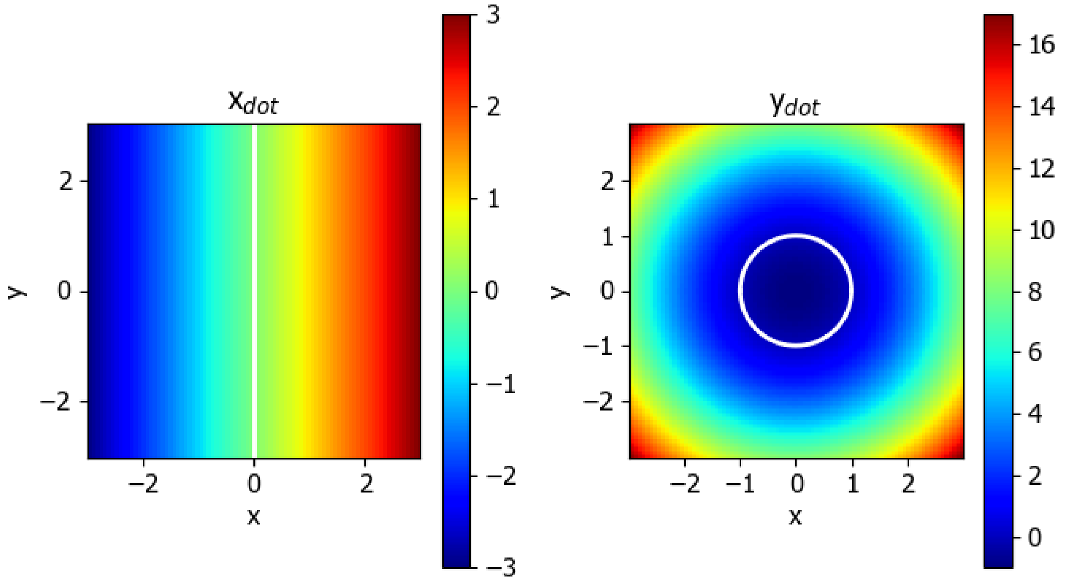


Fig 1.1. [2D] view of the system equations

$$\dot{x} = x \qquad \dot{y} = x^2 + y^2 - 1$$

x-nullcline: $\dot{x} = 0 \Rightarrow x = 0$ (white line)

$$x > 0 \Rightarrow \dot{x} > 0 \qquad x < 0 \Rightarrow \dot{x} < 0$$

y-nullcline: $\dot{y} = 0 \Rightarrow x^2 + y^2 = 1$ (white circle)

$$|y| > 1 \Rightarrow \dot{y} > 0 \qquad |y| < 1 \Rightarrow \dot{y} < 0$$

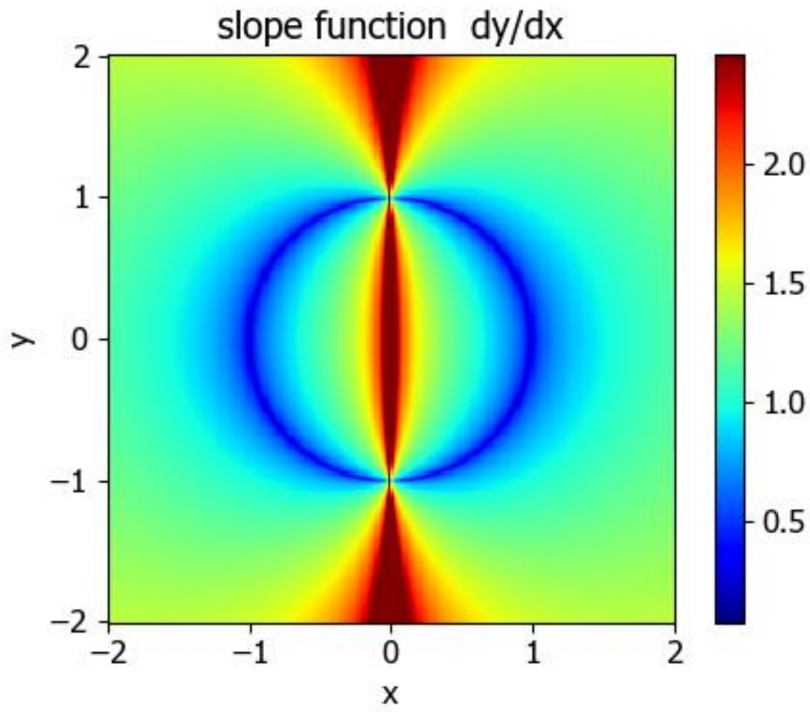


Fig. 1.2. Slope of the trajectories dy/dx

(maximum slope set to 20)

$$\frac{dy}{dx} = \frac{x^2 + y^2 - 1}{x} \qquad x \to 0 \Rightarrow \left|\frac{dy}{dx}\right| \to \infty$$

Critical points (0,1) and (1,0)

The colour coding in figure 1.2 gives an indication of the motion either horizontally (x-direction) or vertically (y-motion). From the colorbar = 0 corresponds to purely horizontal motion, whole colorbar > 2 corresponds to vertical motion. Note: it is very easy to now identify the two critical points

Below are a set of plots with different initial conditions.

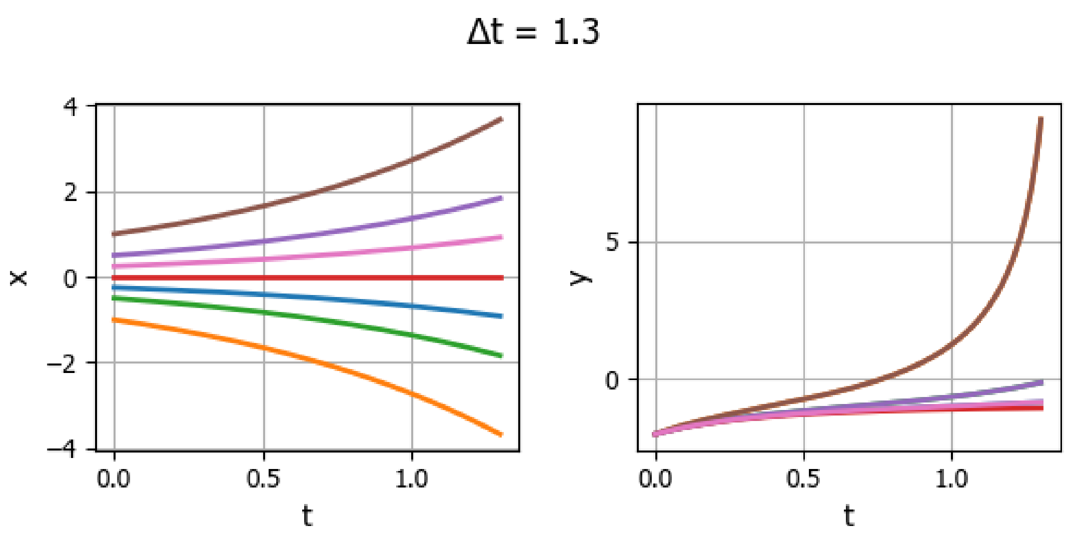


Fig. 1.3. Trajectories for different initial conditions in the time interval $\Delta t = 1.30$. For the initial condition (1,-2), the trajectory rapidly diverges to infinity when $\Delta t > 1.30$,

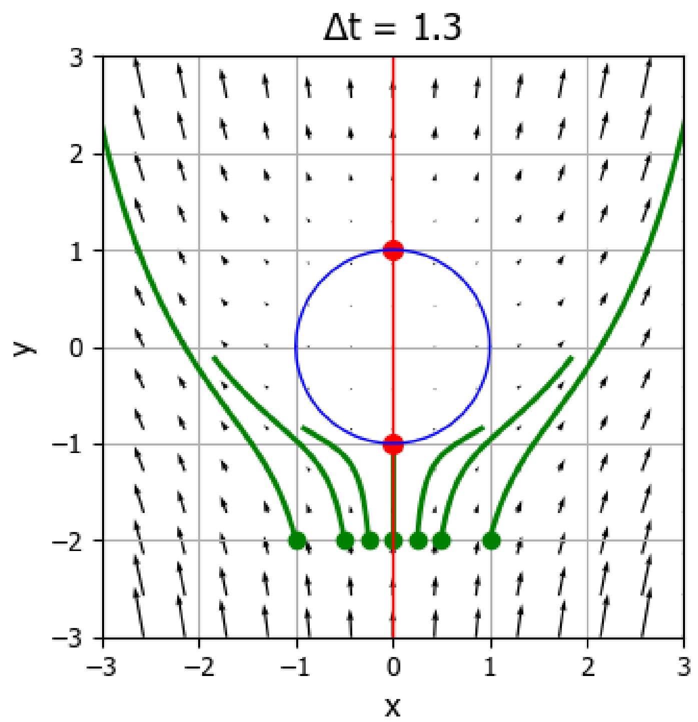


Fig. 1.4. Phase portrait (quiver plot). The red dots show the critical points (0,1) and (0,-1). The red vertical line is the x-nullcline and the blue circle is the y-nullcline.

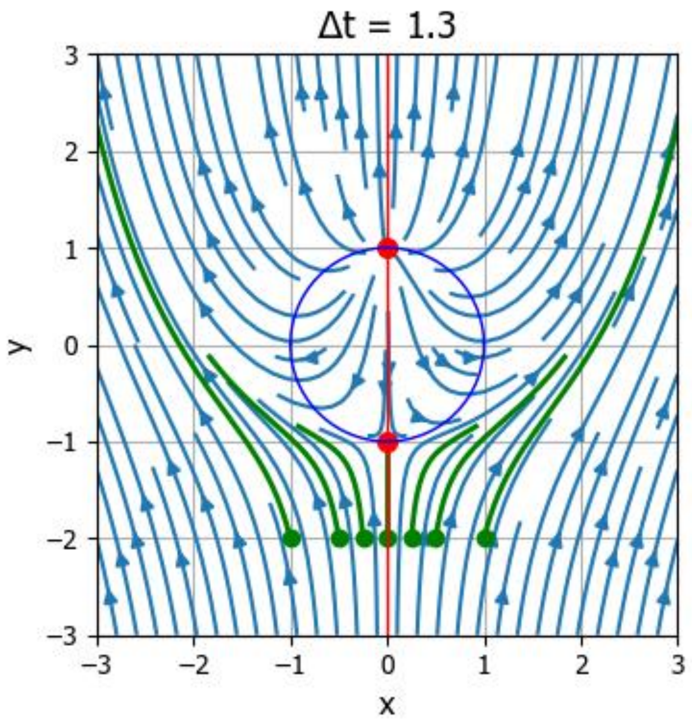


Fig. 1.5. Phase portrait (streamplot). The red dots show the critical points (0,1) and (0,-1). The red vertical line is the x-nullcline and the blue circle is the y-nullcline. The streamplot makes it very easy to predict the trajectory from any staring point.

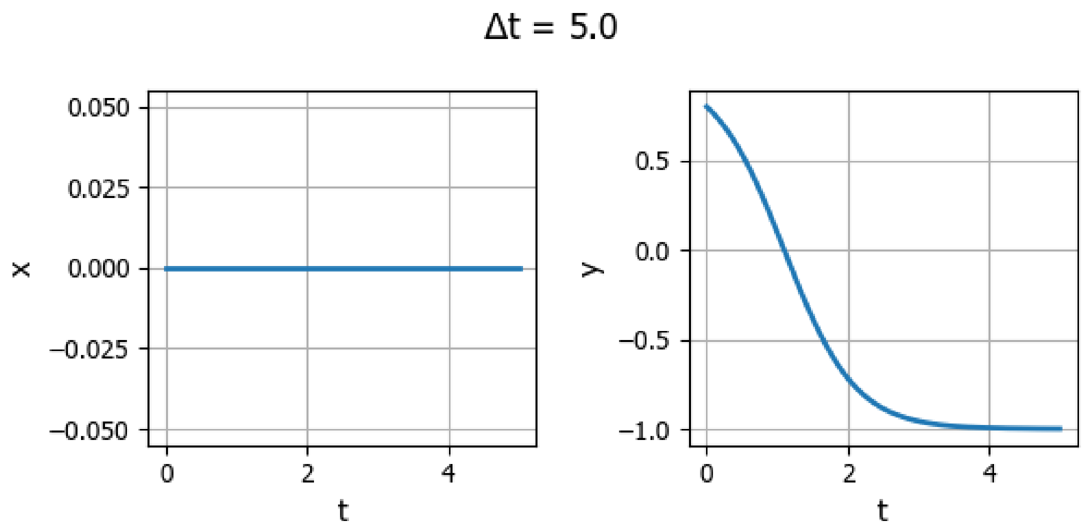


Fig. 1.6.

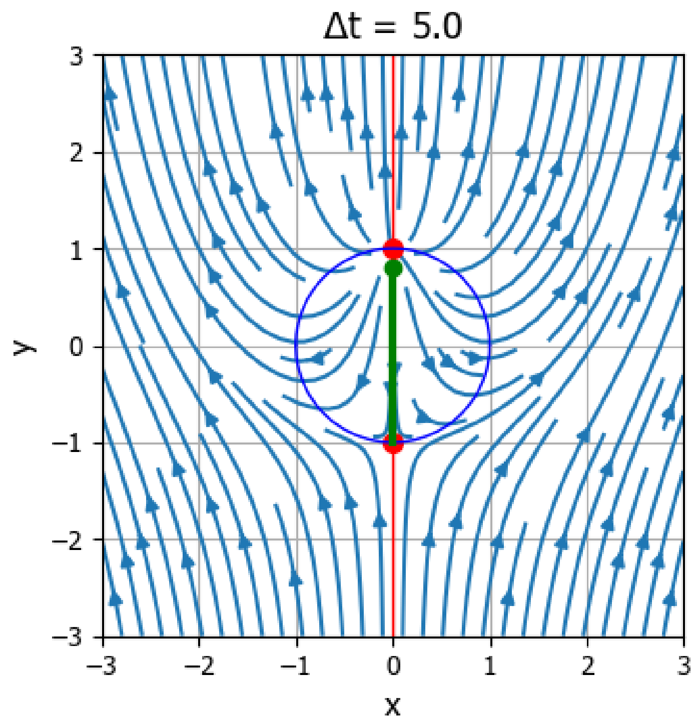
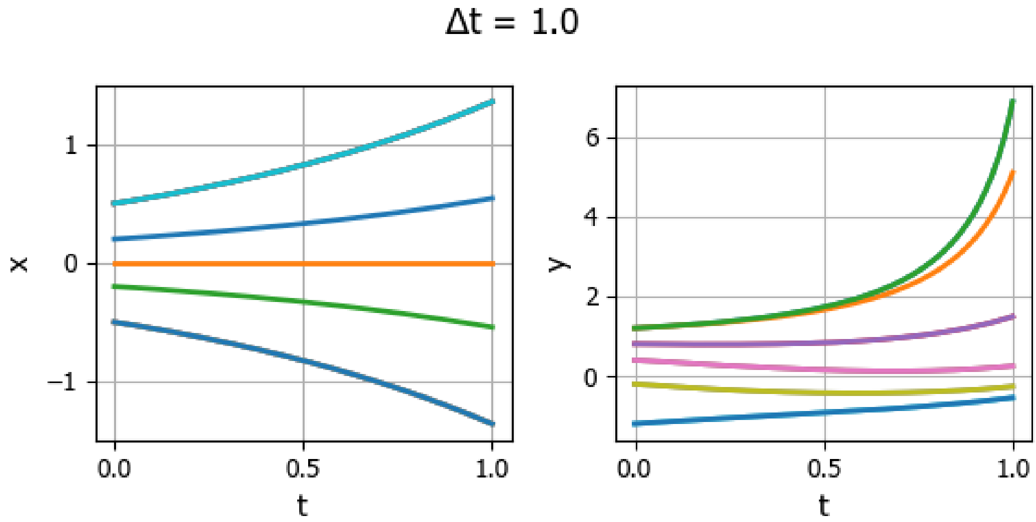


Fig. 1.6. Trajectory for the initial condition (0,0.8).



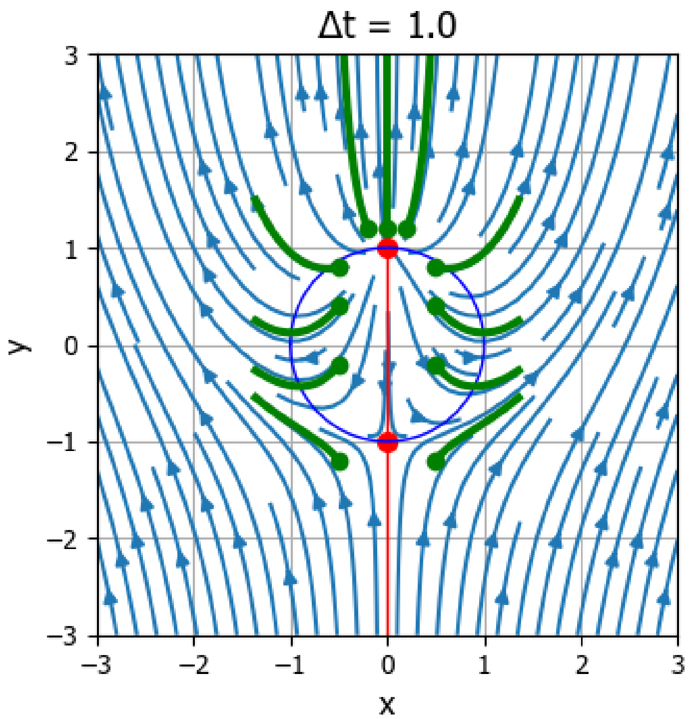


Fig. 1.7. The critical point (0,1) is **unstable** whereas the critical point (0,-1) is a **saddle**.

Example 2 **cs211.py**

$$\dot{x} = y \qquad \dot{y} = x\left(1 - x^2\right) + y$$

Jacobian matrix $\mathbf{J} = \begin{pmatrix} 0 & 1 \\ 1 - 3x^2 & 1 \end{pmatrix}$

x-nullcline $\dot{x} = 0 \Rightarrow y = 0$

y-nullcline $\dot{y} = 0 \Rightarrow y = x\left(x^2 - 1\right)$

Critical points xC = [0 1 -1] yC = [0 0 0]

Critical point (0, 0)

Jacobian matrix J0

[[0 1]
[1 1]]

Eigenvalues J0 **[-0.618 1.618]**

Eigenvectors J0

[[-0.851 -0.526]
[0.526 -0.851]]

The critical point at the Origin **(0, 0)** is a **saddle point** as both eigenvalues are real, one positive and one negative.

Critical point (1, 0)

Jacobian matrix J1

[[0 1]
[-2 1]]

Eigenvalues J1 **[0.5+1.323j 0.5-1.323j]**

Eigenvectors J1

[[0.204-0.54j 0.204+0.54j]
[0.816+0.j 0.816-0.j]]

The critical point at **(1, 0)** is an **unstable focus (spiral point)** since the eigenvalues are complex with the real parts greater zero.

Critical point (-1, 0)
Jacobian matrix J2
 [[0 1]
 [-2 1]]
Eigenvalues J2   **[0.5+1.323j 0.5-1.323j]**
Eigenvectors J2
 [[0.204-0.54j 0.204+0.54j]
 [0.816+0.j   0.816-0.j]]

The critical point at **(-1, 0)** is an **unstable focus (spiral point)** since the eigenvalues are complex with the real parts greater zero.

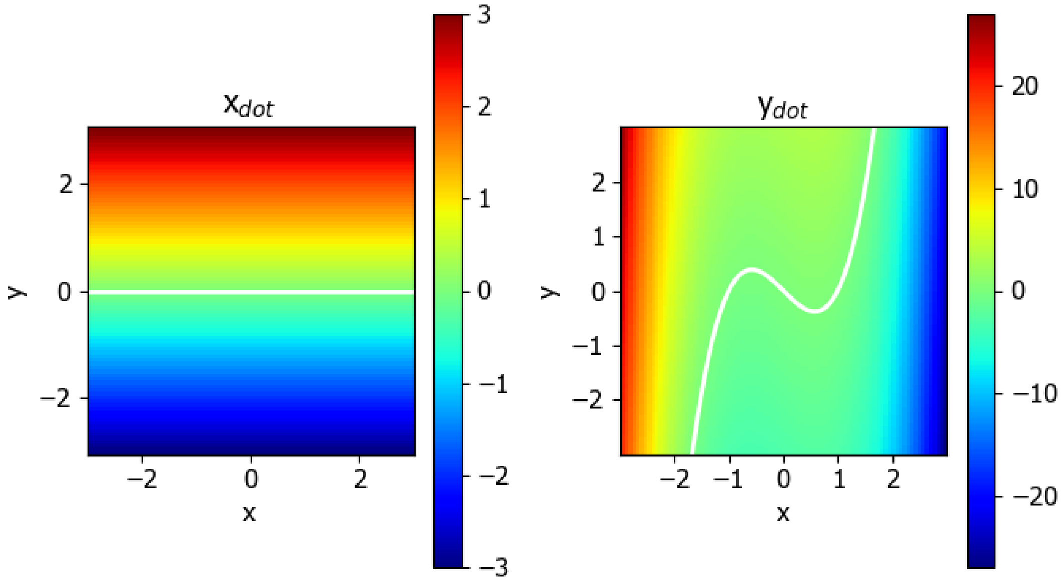


Fig 2.1. [2D] view of the system equations (nullclines: white lines)

$\dot{x} = y \qquad \dot{y} = x\left(1 - x^2\right) + y$

x-nullcline $\qquad \dot{x} = 0 \Rightarrow y = 0$

y-nullcline $\qquad \dot{y} = 0 \Rightarrow y = x\left(x^2 - 1\right)$

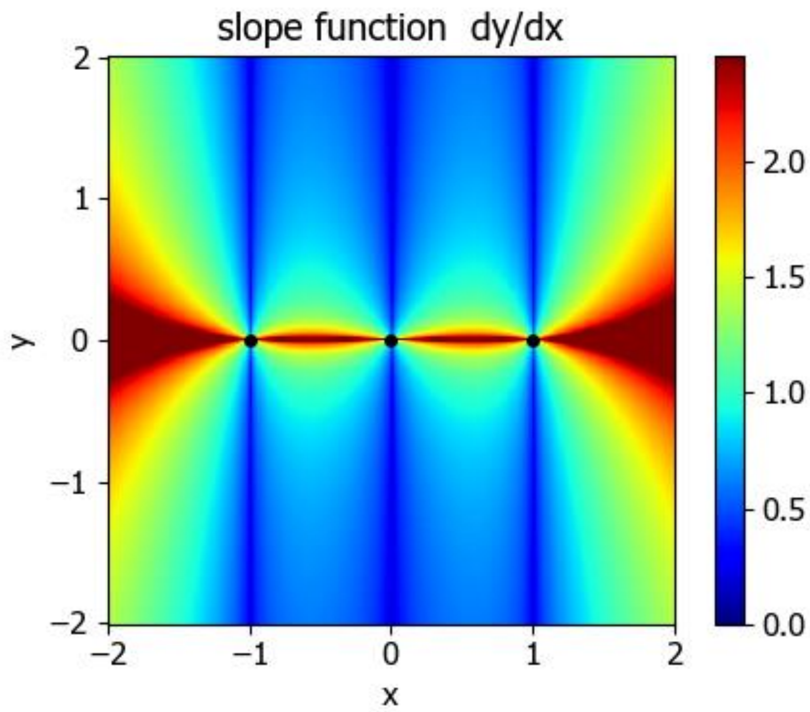


Fig. 2.2. Slope of the trajectories dy/dx

(maximum slope set to 20)

$$\frac{dy}{dx} = \frac{x\left(1 - x^2\right) + y}{y} \qquad y \to 0 \Rightarrow \left|\frac{dy}{dx}\right| \to \infty$$

Critical points (0,1) and (1,0)

The colour coding in figure 1.2 gives an indication of the motion either horizontally (x-direction) or vertically (y-motion). From the colorbar = 0 corresponds to purely horizontal motion, whole colorbar > 2 corresponds to vertical motion.

Below are a set of plots with different initial conditions.

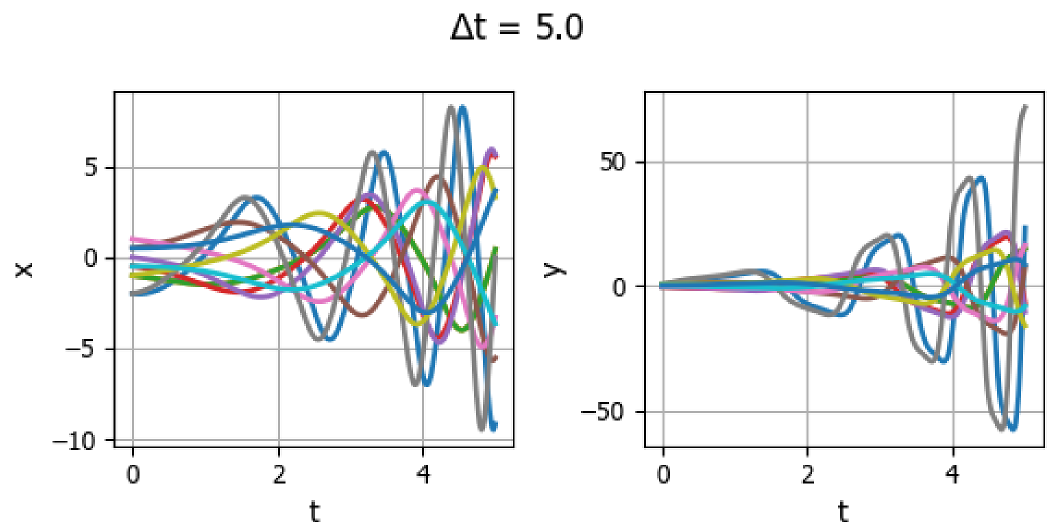


Fig. 2.3. Trajectories for different initial conditions in the time interval $\Delta t = 5.0$.

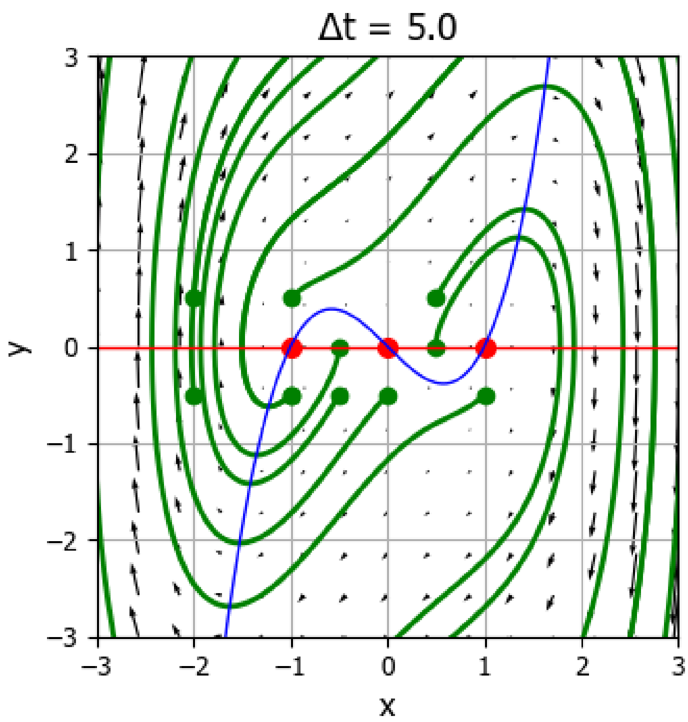


Fig. 2.4. Phase portrait (quiver plot). The red dots show the critical points (0, 0), (1,0) and (-1,0). The red line is the x-nullcline and the blue line is the y-nullcline.

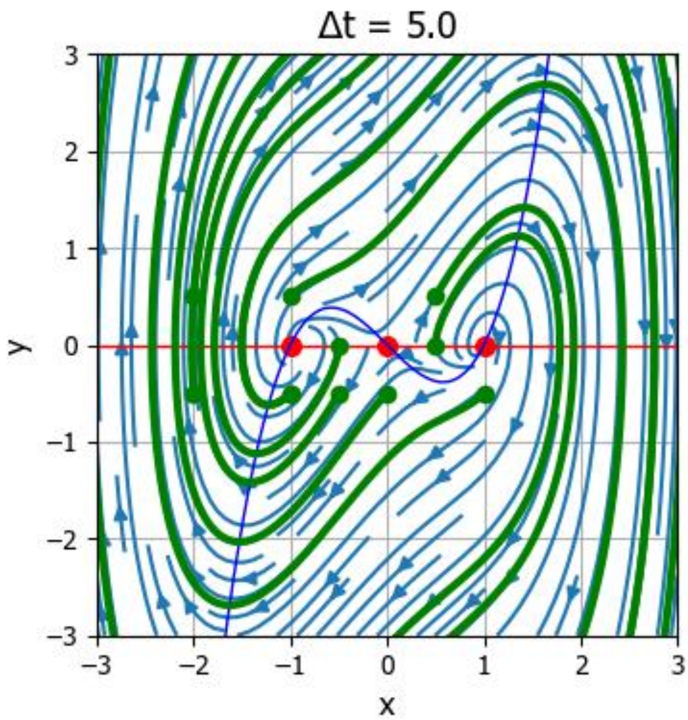


Fig. 2.5. Phase portrait (streamplot). The red dots show the critical points (0,1) and (0,-1). The red dots show the critical points (0, 0), (1,0) and (-1,0). The red line is the x-nullcline and the blue line is the y-nullcline. The streamplot makes it very easy to predict the trajectory from any staring point. One can observe the **spiral patterns** around the critical points (-1,0) and (1,0).