

P3501 - 2 Streamlines, Pathlines, Streaklines

Claus-Dieter Ohl

October 13, 2016

1 Streamlines, Pathlines, Streaklines

In this lecture we discuss three ways to describe and experimentally visualize a flow field, namely the streamline, the pathline, and the streakline. A point in lab space (not moving with the fluid) is called \mathbf{x} , the velocity field is a function of space and time, i.e. $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$. Suppose we inject dye at a point in the flow over a period of time. At any given time (after we start the dye injection), we will see the locus of points in space that are the positions of particles that were at the injection position at previous times. In unsteady flow (flow that is a function of time) these dye streaks are called *streaklines*. In contrast if we mark a single fluid particle, for example we color it and then follow the path of this particle over time we obtain the *pathline* of this fluid particle. The *streamline* is the shape a thin filament would form at some instant of time in the flow, e.g. a wool thread pictured with strobe photography. This filament aligns with flow (we will learn later why) such that it is tangential to the instantaneous velocity field. In steady flow, all three kind of lines coincide, while for unsteady flow they are distinct. In this lecture we derive the ordinary differential equations for the three lines and solve them exemplarily using the appropriate initial conditions. Also you will find a simple python script plotting the lines.

Why?

Above discussion neglects the effect of diffusion or non-neutral buoyancy which would cause particles not to follow the fluid particles. Also, we ignore any inertia of the particle.

1.1 Streamline:

The streamline is a curve tangential to the instantaneous velocity vector; the trajectory of a particle on this “frozen” velocity field. Let $d\mathbf{x}$ be the tangent vector to the streamline of a velocity field $\mathbf{u}(\mathbf{x}; t)$ the velocity field.

$$d\mathbf{x} \times \mathbf{u} = \mathbf{0}$$

This equation can be transformed into three separate equations (*Show!*):

$$\frac{dx_1}{u_1} = \frac{dx_2}{u_2} = \frac{dx_3}{u_3} \quad .$$

For each of the terms the quotient is the same and has the unit of time; let's name this $d\tau$. Above we mentioned a streamline is only valid for a fixed time (if the flow field is unsteady), let's call this time $t = t_0$. We can thus rewrite above equation:

$$\frac{dx_1}{u_1(\mathbf{x}; t = t_0)} = \frac{dx_2}{u_2(\mathbf{x}; t = t_0)} = \frac{dx_3}{u_3(\mathbf{x}; t = t_0)} = d\tau$$

Which can be written as three separate equations

$$\frac{dx_1}{d\tau} = u_1(\mathbf{x}; t = t_0) \quad (1)$$

$$\frac{dx_2}{d\tau} = u_2(\mathbf{x}; t = t_0) \quad (2)$$

$$\frac{dx_3}{d\tau} = u_3(\mathbf{x}; t = t_0) \quad (3)$$

with

$$\mathbf{x}(\tau = 0) = \begin{pmatrix} x_{10} \\ x_{20} \\ x_{30} \end{pmatrix}$$

Then we integrate above equation and if possible eliminate τ by describing it as a function of one of the coordinates (x_1, x_2, x_3) . If not, the streamlines remains being parameterized by τ .

1.1.1 Example

Find the streamline which goes through $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ at time $t = 1$ of the velocity field $\mathbf{u} = \begin{pmatrix} 2x_1 + t \\ x_2 - 2t \end{pmatrix}$. Please ignore the inconsistency in the units.

First we solve for x_1 :

$$\frac{dx_1}{d\tau} = 2x_1 + t \Big|_{t=1} \text{ with } x_1(\tau = 0) = 1 \text{ integrate and insert initial condition leads to } x_1 = \frac{3}{2}e^{2\tau} - \frac{1}{2}.$$

And then we solve for x_2 :

$$\frac{dx_2}{d\tau} = x_2 + 2t \Big|_{t=1} \text{ with } x_2(\tau = 0) = 1 \text{ and again integrate and insert initial conditions gives } x_2 = -e^\tau + 2.$$

Above equations are simple enough to eliminate the parameter τ by solving for τ in the first equation and inserting it into the second, i.e. thus we obtain:

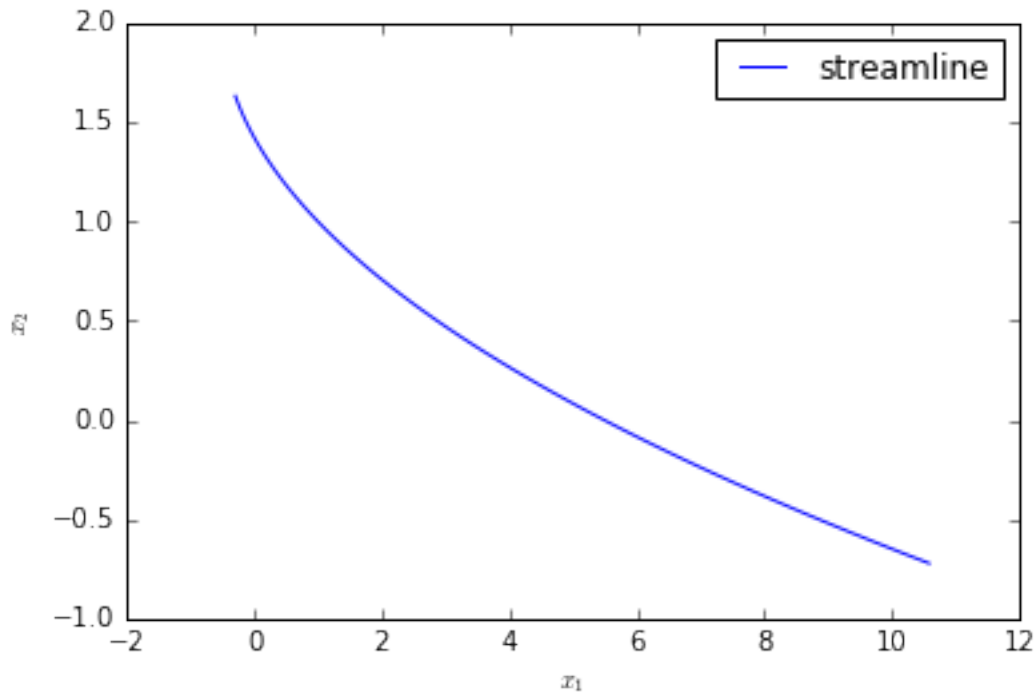
$$x_2 = 2 - \sqrt{\frac{2}{3} \left(x_1 + \frac{1}{2} \right)}$$

.

```

1 #standard boilerplate for Python
2 %matplotlib inline
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import math
6
7 tau=np.linspace(-1.,1.,100)
8 X1_stream=3./2.*np.exp(2.*tau)-.5
9 X2_stream=-np.exp(tau)+2.
10 plt.plot(X1_stream,X2_stream,label='streamline')
11 plt.legend()
12 plt.xlabel('$x_1$')
13 plt.ylabel('$x_2$');
```

We can add a vector field to this plot using the function `quiver()`. This function needs the velocity field which is calculated using two arrays, one for the x_1 -coordinate and one for the x_2 -coordinate through the function `mgrid()`.



```

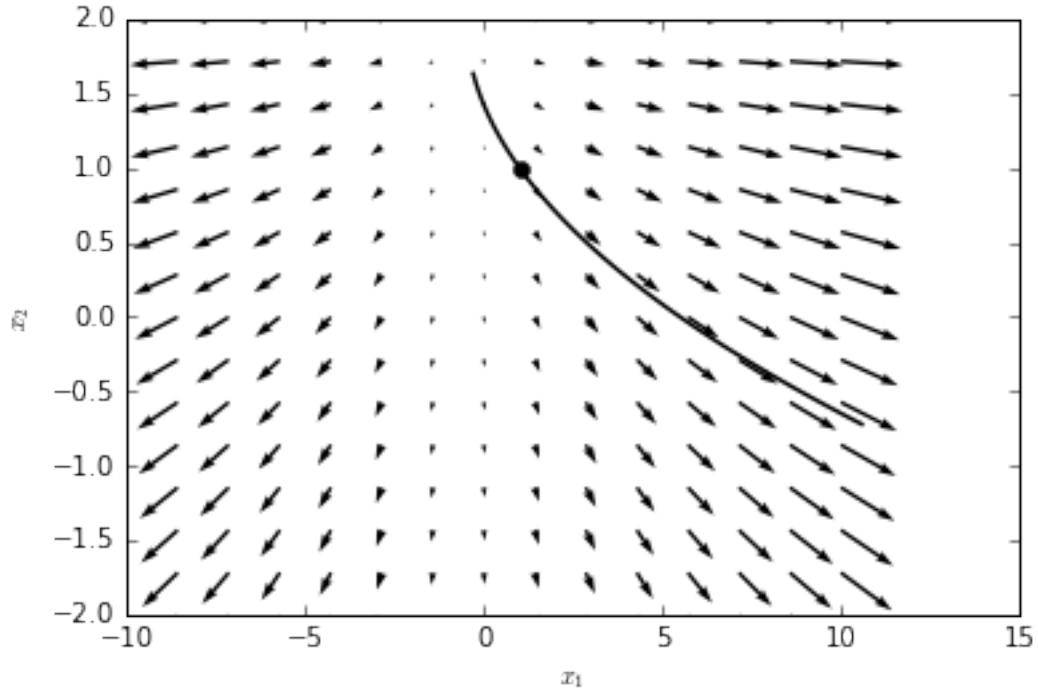
1 #generate two arrays with 15x15 elements
2 #containing the x1 and x2 coordinates respectively
3 x1,x2 = np.mgrid[-10.:10.:15j, -2.:2.:15j]
4 #Calculate the velocity field for each element of the
5 #arrays x1 and x2
6 u1=2.*x1+1.
7 u2=x2-2.
8 plt.quiver(x1,x2,u1,u2,angles='xy') #plot the velocity field
9 plt.hold(True)
10 #streamline through (1,1) at time t=1
11 plt.plot(X1_stream,X2_stream,'k',linewidth=1.5)
12 plt.plot(1.,1., 'ko')
13 plt.xlabel('$x_1$')
14 plt.ylabel('$x_2$');

```

You can see that the streamline is following nicely the velocity vectors, i.e. the streamline is tangential to the velocity vector.

Homework:

Now you add a plot of the streamlines using the python function `streamplot()` and color the lines as a function of the velocity magnitude. You may wonder how the function `streamplot()` calculates the streamlines. It does so by integrating ODEs numerically. We will see in the coming lecture how this can be achieved with an algorithm named “Euler Method”.



1.2 Pathlines

The pathline is the trajectory of a specific fluid particle. We specify the fluid particle by choosing its position \mathbf{x}_0 at a specific time t_0 .

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}; t)$$

with $\mathbf{x}(t = t_0) = \mathbf{x}_0$. For most of the cases no analytical solution exists and the a numerical integration is done. The solution is given as a vector function $\mathbf{x} = \mathbf{x}(\mathbf{x}_0; t)$.

1.2.1 Example

For above example of the flow field we can calculate analytically the flow field. We first specify the fluid particle $\mathbf{x}_0(t = 1) = (1, 1)^T$.

Thus we have to solve

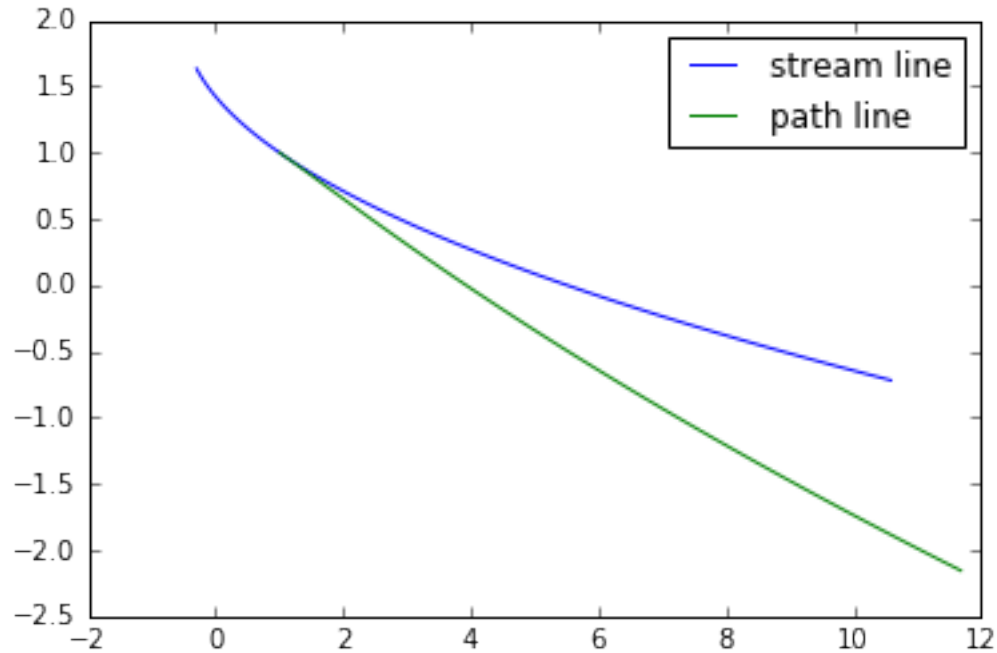
$$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} 2x_1 + t \\ x_2 - 2t \end{pmatrix}$$

Both equations are linear ODEs which can be solved with the [integrating factor](#) and give

$$\mathbf{x} = \begin{pmatrix} \frac{7}{4}e^{2(t-1)} - \frac{t}{2} - \frac{1}{4} \\ -3e^{t-1} + 2t + 2 \end{pmatrix} .$$

Here we can't eliminate the parameter t , yet for simpler solutions that may be possible and explicit results of the kind $x_2 = f(x_1)$ can be given.

Next we plot the pathline (blue) and compare it with the streamline (green).



```

1 t=np.linspace(1,2,100)
2 X1_path=7./4.*np.exp(2.*(t-1))-t/2.-0.25
3 X2_path=-3.*np.exp(t-1.)+2.*t+2.
4 #one can add multiple lines with a single plot command
5 plt.plot(X1_stream,X2_stream,X1_path,X2_path)
6 #and then add legends to the plot.
7 plt.legend(('stream line','path line'));

```

1.3 Streaklines

Streaklines are the locus of points at time t_0 which have passed through a particular point \mathbf{x}_0 at some past time $\hat{t} \leq t$.

The streamlines can be obtained by calculating all pathlines through a point \mathbf{x}_0 at some time $t = \hat{t}$, i.e.

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}; t)$$

with $\mathbf{x}(t = \hat{t}) = \mathbf{x}_0$.

We therefore obtain a solution $\mathbf{x}(\mathbf{x}_0; t; \hat{t})$. Next we specify the time the streakline is observed t and fix \mathbf{x}_0 to obtain the positions $\mathbf{x}(\hat{t})$ with $\hat{t} \leq t$.

1.3.1 Example

Let's find the streakline at time $t = 1$ through the point $\mathbf{x}_0 = (1, 1)^T$. We therefore have to solve the ODEs with the initial condition $\mathbf{x}(t = \hat{t}) = \mathbf{x}_0$. This means, that we allow for all pathlines passing through point \mathbf{x}_0 .

$$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} 2x_1 + t \\ x_2 - 2t \end{pmatrix}$$

If we then select a specific time, we obtain the locus of all the pathlines for that time, parameterized by \hat{t} .

Let's do it, first we find the solution for $\mathbf{x}(t = \hat{t}) = 1$, which can be again obtained with the integrating factor. The constant of integration now includes t and \hat{t} and the result is

$$\mathbf{x} = \begin{pmatrix} \frac{5+2\hat{t}}{4}e^{2(t-\hat{t})} - \frac{t}{2} - \frac{1}{4} \\ -(1+2\hat{t})e^{t-\hat{t}} + 2t + 2 \end{pmatrix}.$$

To obtain the streakline at time $t = 1$, we just insert this into above parametrization and obtain

$$\mathbf{x} = \begin{pmatrix} \frac{5+2\hat{t}}{4}e^{2(1-\hat{t})} - \frac{3}{4} \\ -(1+2\hat{t})e^{1-\hat{t}} + 4 \end{pmatrix}.$$

Again, the expression is too complicated to obtain an explicit solution of the kind $x_2 = f(x_1)$, yet we can plot it using \hat{t} as parameter, $\hat{t} < t$ which is $\hat{t} < 1$.

```

1 #t_hat goes from 0. to 1.
2 #and shows the locus of all pathlines
3 #going through point (0.,0) at time t=1.
4 th=np.linspace(0.,1.,100)
5 X1_streak=(5.+2.*th)/4.*np.exp(2.*(1-th))-0.75
6 X2_streak=-(1.+2*th)*np.exp(1.-th)+4
7 #Let's plot all three lines together
8 plt.plot(X1_stream,X2_stream,X1_path,X2_path,X1_streak,X2_streak)
9 plt.legend(('stream line','path line','streak line'));
10 plt.xlabel('$x_1$')
11 plt.ylabel('$x_2$');
```

Let's have a look at above graph. All graphs go through point $(1, 1)^T$. From the velocity field

$$\mathbf{u} = \begin{pmatrix} 2x_1 + t \\ x_2 - 2t \end{pmatrix}$$

it is evident that the vertical velocity component u_2 becomes more negative with time. Thus the pathline of a particle passing through point $(1, 1)^T$ at $t = 1$ will be more negative in x_2 -direction than the streamline through this point and at this time. The streakline is the locus of all pathlines who have passed through $(1, 1)^T$ at time $t = 1$; then the velocity field in x_2 -direction was less negative. Therefore, the streakline is above both other lines.

Homework:

Have a look at the next three videos and explain what is that what you see. Are these streamlines, streaklines, and/or pathlines?

Out [13]:

Out [14]:

Out [15]:

Out [16]: <IPython.core.display.HTML object>

