TB 13-15 Kinematic Vorticity

03-05-18

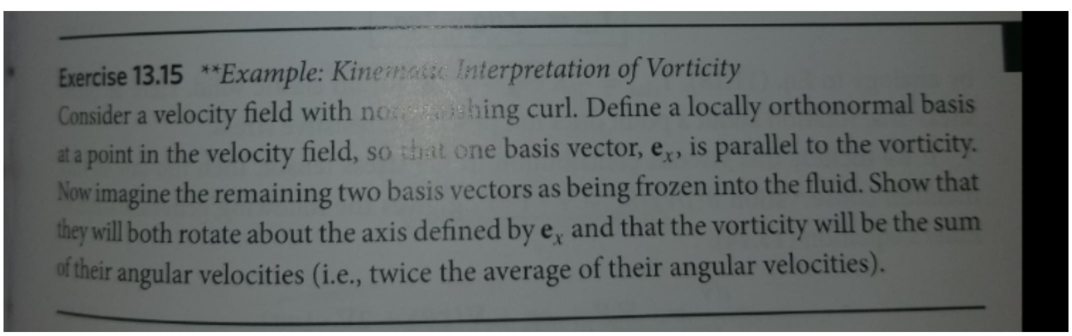
N. T. Gladd

Initialization: Be sure the file *NTGUtilityFunctions.m* is in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Purpose

This is the 9th in a series of notebooks in which I work through material and exercises in the magisterial new book *Modern Classical Physics* by Kip S. Thorne and Roger D. Blandford. If you are a physicist of any ilk, BUY THIS BOOK. You will learn from a close reading and from solving the exercises.

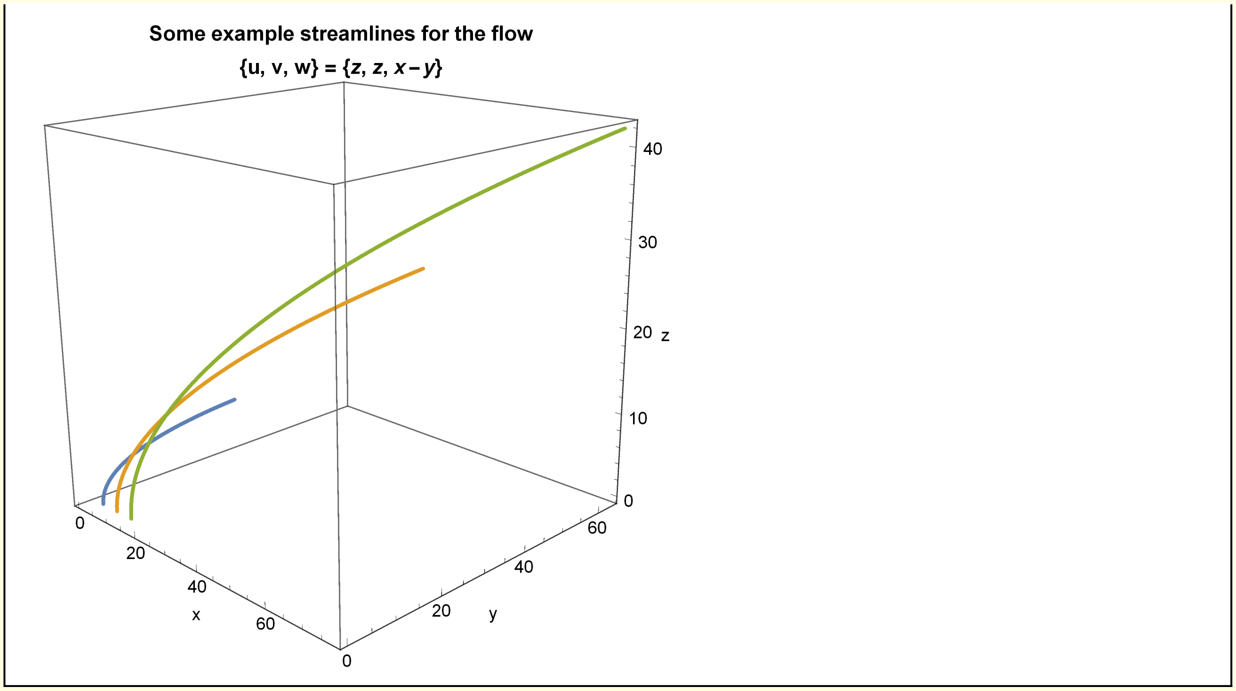


Exercise 13.15 ***Example: Kinematic Interpretation of Vorticity***
Consider a velocity field with nonvanishing curl. Define a locally orthonormal basis at a point in the velocity field, so that one basis vector, $\mathbf{e}_x$, is parallel to the vorticity. Now imagine the remaining two basis vectors as being frozen into the fluid. Show that they will both rotate about the axis defined by $\mathbf{e}_x$ and that the vorticity will be the sum of their angular velocities (i.e., twice the average of their angular velocities).

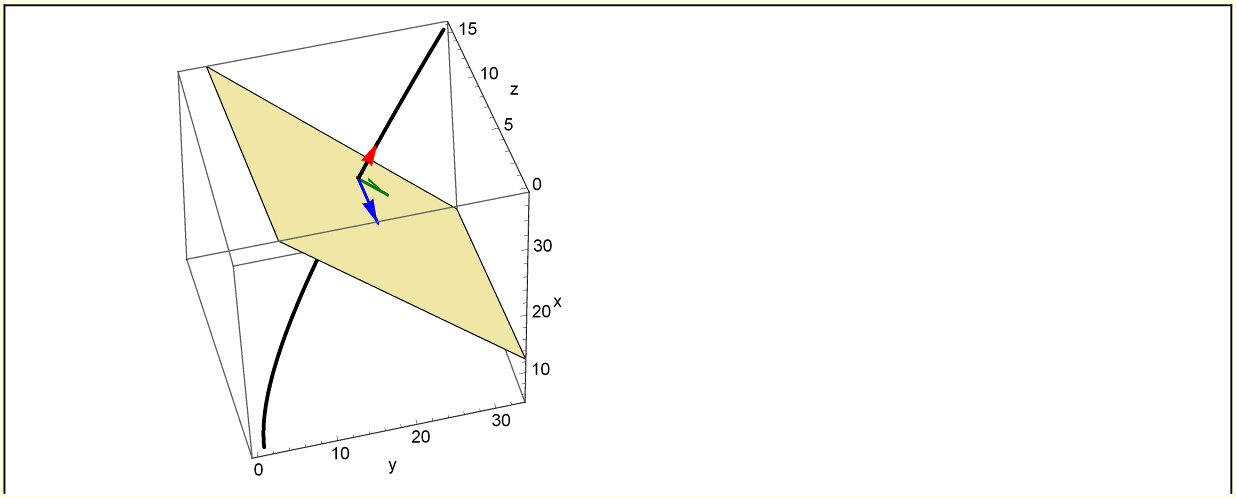
Analysis and solution

The statement of the problem seems straightforward. However, I found the solution required quite a bit of background work and exposed some weak spots in my background concerning fluids. Although I was a plasma physicist and published a number of papers in the Physics of Fluids, I never had a course in fluid mechanics. While many plasma physicists work with magnetohydrodynamic models, that was not my situation. Early in my graduate research, at the consistent with their research interests of my teachers, I focused on the kinetic theory of plasma as it applies to waves and instabilities. I encountered very little of classical fluid mechanics. So - while solving this problem, I took the opportunity to work out some details of aspects of fluid kinematics that were unfamiliar.

I start by visualizing the problem. In section *E Illustration of 3-D streamline* below, I derive an analytical expression for the streamlines associated with an example flow consistent with the requirements of the problem.



The next step is to construct an orthonormal coordinate system at some point s_0 along one of the streamlines. Using techniques from differential geometry, I construct a Frenet-Serret frame at s_0. The three colored vectors are the tangent (red), normal(blue) and binormal(green). The yellow plane containing the normal and binormal is locally perpendicular to the streamline.



The example flow is not curl free. There is vorticity and the unit vectors perpendicular to the streamline will deform and rotate for s > s_0. For the purposes of working out the details of the rotation called for in the problem, it is sufficient to consider the instantaneous rotation using a 2-D Cartesian coordinate system with the flow in the x-y plane and the vorticity oriented in the z-direction.

I find it convenient to work with a specific example of a flow while discussing the background topics below. I choose the flow

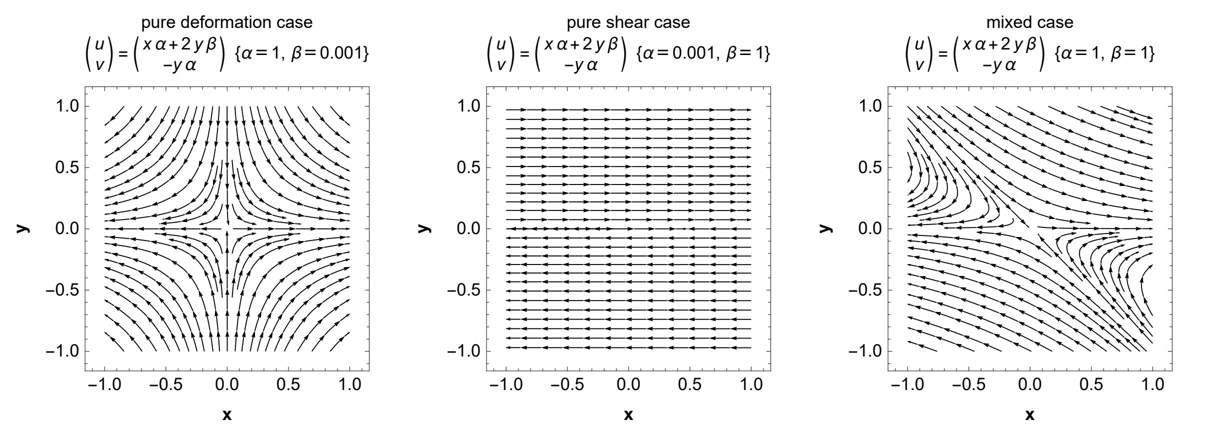
$$
\begin{aligned}
u &= \alpha x + 2\beta y \\
v &= 2\beta y \\
w &= 0
\end{aligned}
\tag{1}
$$

This flow is incompressible, but does have vorticity (rotation).

```
{Div[{α x + 2 β y, -α y}, {x, y}], Curl[{α x + 2 β y, -α y}, {x, y}]}
```

```
{0, -2 β}
```

This particular flow is discussed in Section 3.2.4 of *Physical Hydrodynamics*, E. Guyon, J-P Hulin, L. Petit, C Mitescu. For different parameter choices the flow can be either pure deformation or pure shear. See Section *A Streamlines, pathlines and streaklines* for details of the calculations leading to the following figures.



I used the Mathematica function StreamPlot to generate the stream lines.

It is useful to derive the functional form of the streamlines for the example flow. From the requirement that a differential element is tangent to the flow —
dr × **v** = 0, it follows that the differential equations for the streamline are

$$
\frac{dx}{u(x,\ y)} = \frac{dy}{v(x,\ y)}
\quad \text{or} \quad
\frac{dy}{dx} = \frac{v(x,\ y)}{u(x,\ y)}
\tag{2}
$$

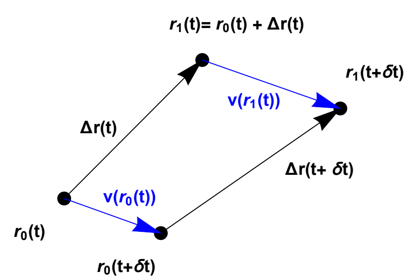
In section A, I explicitly derive the following parametric representation for the streamlines associated with the flow in equation (1).

$$\{x[s],\ y[s]\} = \left\{\frac{e^{-s\,\alpha}\left(e^{2\,s\,\alpha}\,x0\,\alpha - y0\,\beta + e^{2\,s\,\alpha}\,y0\,\beta\right)}{\alpha},\ e^{-s\,\alpha}\,y0\right\}$$

I check this derivation by overlaying some streamlines generated with this expression on those generated by StreamPlot (see figure below).

When visualizing fluid motion, pathlines and streaklines are used in addition to streamlines. For the purpose of comparing these entities I also consider an example of an unsteady flow in Section A. A comparative plot of the three types of lines is presented.

An infinitesimal fluid element will propagate along a stream line but the specific interest here is how a small Cartesian coordinate system will deform and rotate as it flows. It’s easier to first consider how a general vector will be modified by the flow and then specialize the result to vectors representing the x and y-axes. Consider the figure below. I show how the vector $\boldsymbol{r_1}(t)$ - $\boldsymbol{r_0}(t)$ changes during a time interval δt. The flow in the figure is actually generated using the the flow in equation (1) — see Section *B Deformation tensor: Introduction* for details.



Note

$$\boldsymbol{r_0}(t+\delta t) = \boldsymbol{r_0}(t) + \boldsymbol{v}(\boldsymbol{r_0}(t))\,\delta t$$
$$\boldsymbol{r_1(t)} = \boldsymbol{r}_0(t) + \boldsymbol{\Delta r}(t) \tag{3}$$
$$\boldsymbol{r_1}(t+\delta t) = \boldsymbol{r_1}(t) + \boldsymbol{v}(\boldsymbol{r_1}(t))\,\delta t$$

There are two small quantities here – the initial separation of the two spatial points which I denote as **Δr**(t), and the time interval δt over which the fluid elements move.

Since $\mathbf{\Delta r}(t)$ is considered small, the velocity at t is approximately

$$\boldsymbol{v}(\boldsymbol{r}_1(t+\delta t) \cong \boldsymbol{v}(\boldsymbol{r_0}(t)) + \nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t) \tag{4}$$

The separation at time t+δt is

$$\begin{aligned}\mathbf{\Delta r}(t+\delta t) &= \boldsymbol{r_1}(t+\delta t) - \boldsymbol{r}_0(t+\delta t)\\ &\cong \boldsymbol{r_1}(t) + (\boldsymbol{v}(r_0(t)) + \nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t))\,\delta t - (\boldsymbol{r}_0(t) + \boldsymbol{v}(\boldsymbol{r}_0(t))\,\delta t)\\ &= (\boldsymbol{r_1}(t) - \boldsymbol{r}_0(t)) + (\boldsymbol{v}(r_0(t))\,\boldsymbol{r_0(t)}\,\nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t))\,\delta t - (\boldsymbol{r}_0(t) + \boldsymbol{v}(\boldsymbol{r}_0(t))\,\delta t)\\ &= (\boldsymbol{r_1}(t) - \boldsymbol{r}_0(t)) + \nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t)\,\delta t\\ &= \mathbf{\Delta r}(t) + \nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t)\,\delta t\end{aligned} \tag{5}$$

Thus,

$$\mathbf{\Delta r}(t+\delta t) - \mathbf{\Delta r}(t) = \nabla \boldsymbol{v}\,|_0 \cdot \mathbf{\Delta r}(t)\,\delta t \tag{6}$$

The quantity $\nabla \mathbf{v}\,|_0$ can be represented as a matrix

$$\boldsymbol{J} \equiv \nabla \boldsymbol{v}\,|_0 = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{pmatrix}_{(x_0,y_0)} \tag{7}$$

J is called the rate of deformation tensor. Actually this tensor seems to have different names in different areas of fluid physics and engineering.

In Section B, I also carry out the derivation just performed using Mathematica.

Decomposition of the rate of transformation matrix: The **J** tensor can be decomposed into components that can be associated with different geometric effects. Split **J** into symmetric and anti-symmetric parts (T denotes matrix transform)

$$\boldsymbol{J} = \frac{1}{2}\left(\boldsymbol{J} + \boldsymbol{J}^T\right) + \frac{1}{2}\left(\boldsymbol{J} - \boldsymbol{J}^T\right) \equiv \boldsymbol{\epsilon} + \boldsymbol{\omega} \tag{8}$$

where $\boldsymbol{\epsilon}$ is the rate of strain tensor and $\boldsymbol{\omega}$ is the vorticity tensor. The $\boldsymbol{\epsilon}$ tensor is further decomposed. The diagonal of the $\boldsymbol{\epsilon}$ tensor is identified as the rate of translation tensor.

$$\boldsymbol{t} = \frac{1}{2}\,\mathrm{Tr}(\boldsymbol{\epsilon})\,\boldsymbol{I} \tag{9}$$

where Tr denotes the trace and **I** is the identity tensor. Finally, $\boldsymbol{\sigma}$, the rate of shear tensor, is the off-diagonal component of $\boldsymbol{\epsilon}$.

$$\boldsymbol{\sigma} = \boldsymbol{\epsilon} - \frac{1}{2}\,\mathrm{Tr}(\boldsymbol{\epsilon})\,\boldsymbol{I} \tag{10}$$

In summary,

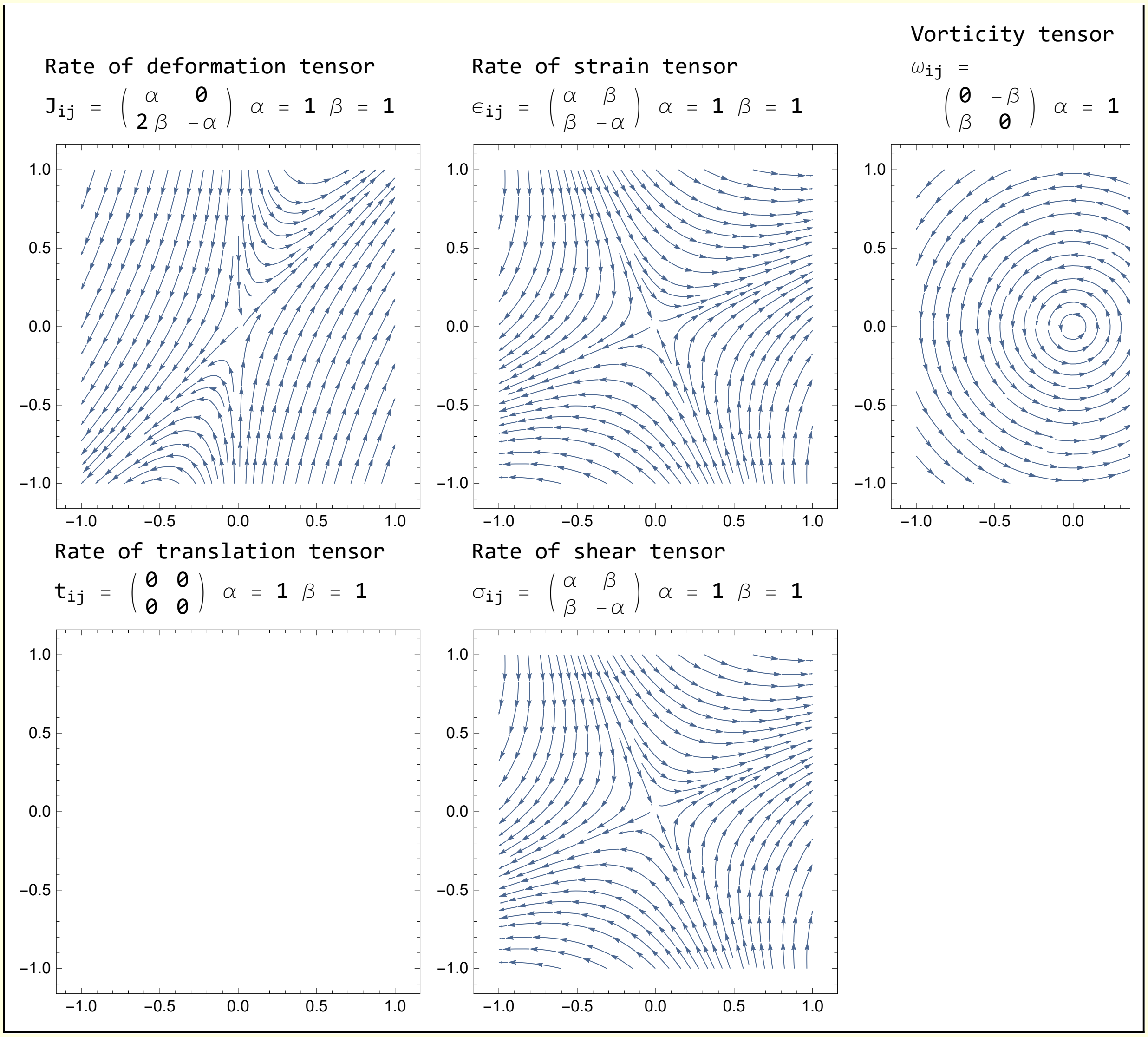
$$\boldsymbol{J} = \boldsymbol{t} \;\; \text{(translation)} + \boldsymbol{\sigma} \;\text{(shear)} \; + \boldsymbol{\omega} \;\text{(rotation)} \tag{11}$$

Further, note that the trace term is

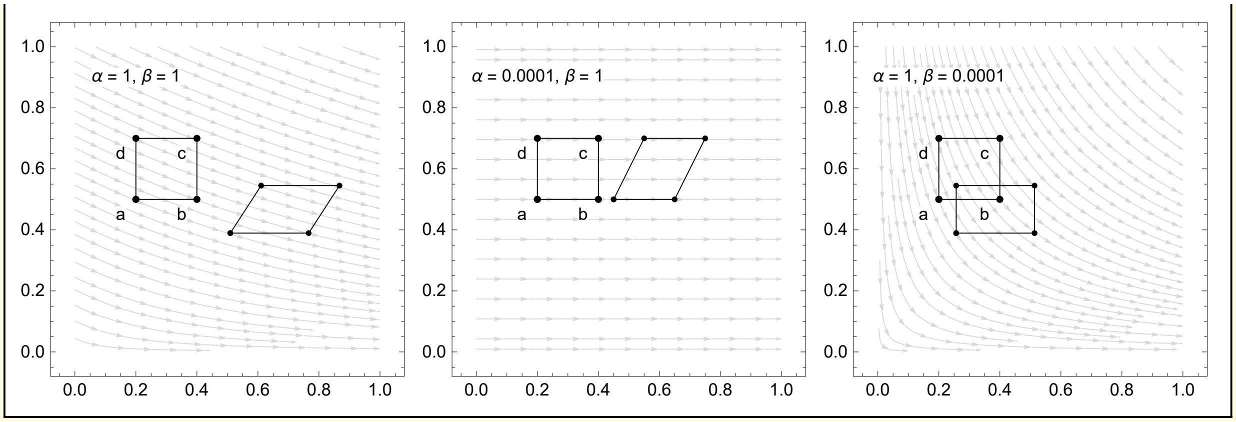
$$\mathrm{Tr}(\boldsymbol{\epsilon}) \;=\; \frac{\partial u}{\partial x}\Big|_0 + \frac{\partial v}{\partial x}\Big|_0 \;=\nabla\cdot\boldsymbol{v} \tag{12}$$

so a vanishing trace corresponds to an incompressible flow. In Section B below, I implement a function that perform the decomposition of F for a given flow.

For the example flow of equation (1), the various rate tensors are

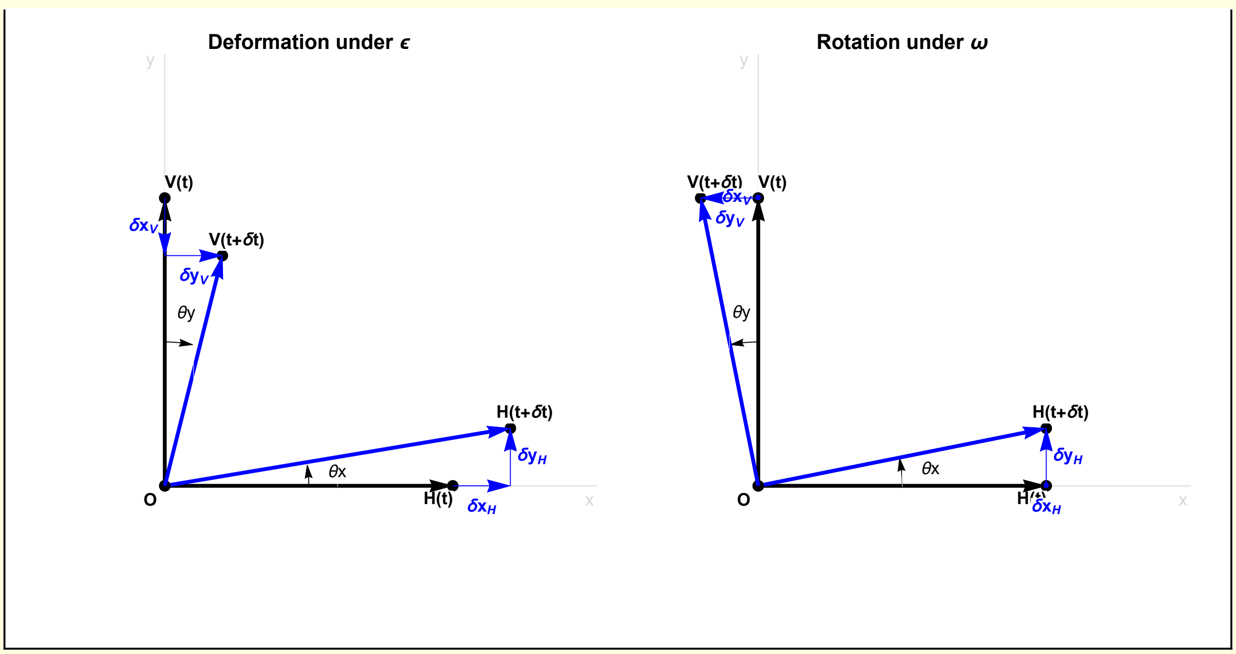


To illustrate the geometrical deformations I calculate the deformation of the square abcd under the flow described by equation (1).



For the special case of a shear flow (α = 0), the square is sheared into a parallelogram. For the special case of a deformation flow (β = 0), the square is distorted into a rectangle. Note that since the flow is incompressible, the area of the rectangle is the same as the area of the square. For the more general flow, α = β = 1, both distortion and shear occur.

With these preliminary steps completed, I calculate the distortion of a Cartesian coordinate system. For the x-axis, choose **Δr**(t) = {Δx(t), 0}. For the y-axis, choose **Δr**(t) = {0, Δx(t).}. As detailed below, the deformation and rotation under the example flow is



where I have shown the change in the axes over a small interval of time δt for the example flow of equation (1).

Remember **J** = $\boldsymbol{\epsilon}$ + $\boldsymbol{\omega}$. The effect of rate of strain tensor $\boldsymbol{\epsilon}$ is to shear the coordinate axes — the x-axis is rotated counterclockwise by θ_x, the y-axis is rotated clockwise by θ_y. The effect of the vorticity tensor is to rotate both the x-axis and the y-axis in the counterclockwise direction.

These two axes correspond the basis axes mentioned in the problem statement. The average rate of rotation is

$$\dot{\theta}_{\text{ave}} = \frac{1}{2} \frac{(\theta_x + \theta_y)}{\delta t} \tag{13}$$

The vorticity is

$$\boldsymbol{\omega} = \frac{(\theta_x + \theta_y)}{\delta t} \mathbb{1}_z = 2\, \dot{\theta}_{\text{av}}\, \mathbb{1}_z \tag{14}$$

which is the desired result.

Supporting details follow below.

A Streamlines, pathlines and streaklines

Visualization is useful when considering kinematical fluid flows and the consideration of representative examples is helpful.

Example 1: A stylized 2-D flow

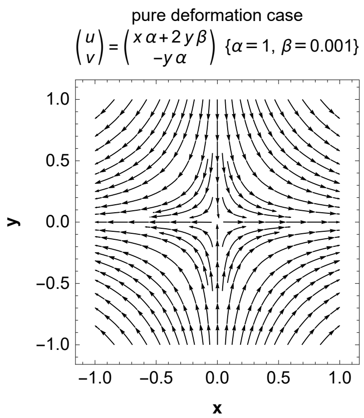
In Section 3.2.4 of *Physical Hydrodynamics*, E. Guyon, J-P Hulin, L. Petit, C Mitescu, the following flow is considered

$$u = \alpha\, x + 2\, \beta\, y$$
$$v = -\alpha\, y$$
$$w = 0$$

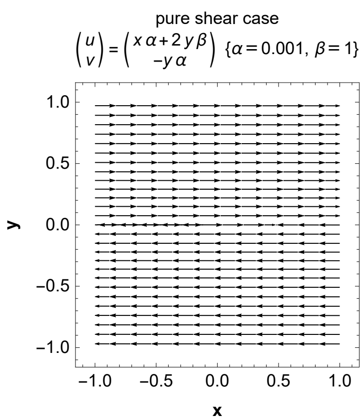
where the common notation {u, v, w} $\Longleftrightarrow$ $\{v_x, v_y, v_z\}$ is adopted. This flow field has some descriptive limiting cases.

Using Mathematica's StreamPlot, I can immediately generate the streamlines associated with this flow.

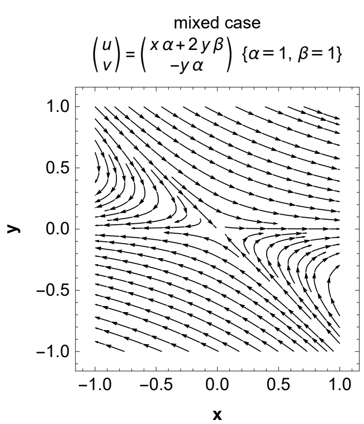
```
G[1] = With[{α = 1, β = 0.001, description = "pure deformation case"},
  ShowStreamPlot[α, β, description]]
```



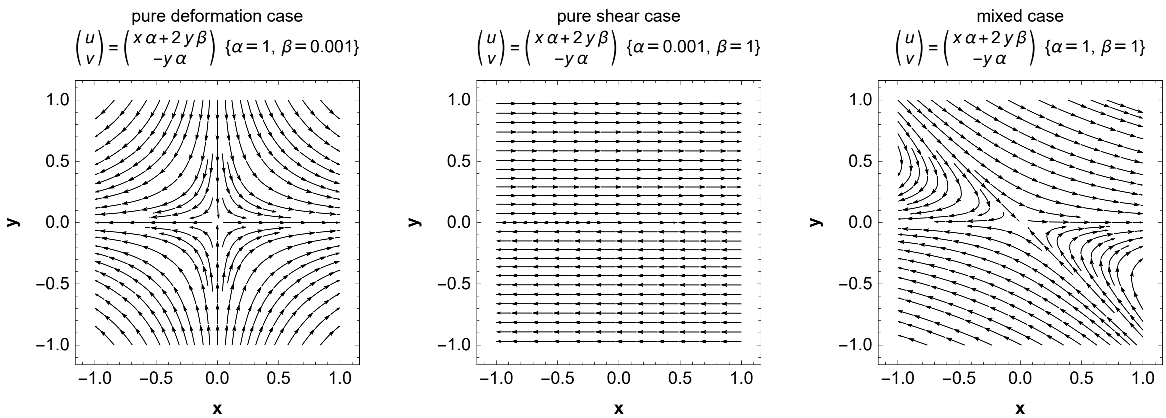

```
G[2] = With[{α = 0.001, β = 1, description = "pure shear case"},
  ShowStreamPlot[α, β, description]]
```




```
G[3] = With[{α = 1, β = 1, description = "mixed case"},
  ShowStreamPlot[α, β, description]]
```



```
Grid[{{G[1], G[2], G[3]}}]
```




```
Clear[ShowStreamPlot];
ShowStreamPlot[αα_, ββ_, description_] :=
 Module[{vFlow, lab},
  vFlow[x_, y_, α_, β_] := {α x + 2 β y, -α y};
  lab = StringForm["``\n`` = ``  ``", description,
    TraditionalForm[(u
                     v)], TraditionalForm[(α x + 2 β y
                                              -α y)], {α == αα, β == ββ}];
  Module[{α = αα, β = ββ, R = 1},
   StreamPlot[vFlow[x, y, α, β], {x, -R, R}, {y, -R, R}, ImageSize → 200,
    StreamStyle → Black, FrameLabel → {{Stl["y"], ""}, {Stl["x"], lab}}]]]
```

Streamline equations: Streamlines are determined by the condition that the streamline be tangent to the flow.

$$\mathbf{dr} \times \boldsymbol{v} = 0 \qquad (15)$$

I illustrate the derivation of the differential equations describing the streamlines.

```
wA[1] = Cross[{dx, dy, dz}, {u, v, w}] == 0
```

```
{-dz v + dy w, dz u - dx w, -dy u + dx v} == 0
```

Each component must be zero

```
wA[2] = (# == 0) & /@ wA[1][[1]]
```

```
{-dz v + dy w == 0, dz u - dx w == 0, -dy u + dx v == 0}
```

```
wA[3] = Solve[wA[2], {dx, dy, dz}][[1]] // RE
```

Solve: Equations may not give solutions for all "solve" variables.

$$\left\{dy == \frac{dx\, v}{u},\ dz == \frac{dx\, w}{u}\right\}$$

Thus, for the 2-D case the streamlines involve solving the differential equations

$$\frac{dx}{u(x,\ y)} = \frac{dy}{v(x,\ y)}$$

or

$$\frac{dy}{dx} = \frac{v(x,\ y)}{u(x,\ y)}$$

Formulae defining streamlines

For the example flow, start by solving the 2nd equation for y[s]

```
wA[4] = DSolve[{D[y[s], s]/(- α y[s]) == 1, y[0] == y0}, y[s], s][[1, 1]]
```

$$y[s] \to e^{-s\alpha}\, y0$$

The first equation would then be

```
wA[5] = D[x[s], s]/(α x[s] + 2 β y[s]) == 1 /. wA[4]
```

$$\frac{x'[s]}{2\, e^{-s\alpha}\, y0\, \beta + \alpha\, x[s]} == 1$$

Hence

```
wA[6] = DSolve[{wA[5], x[0] == x0}, x[s], s][[1, 1]]
```

$$x[s] \to \frac{e^{-s\alpha}\left(e^{2 s\alpha}\, x0\, \alpha - y0\, \beta + e^{2 s\alpha}\, y0\, \beta\right)}{\alpha}$$

This approach provides parametric representation of the streamline

```
wA[7] = {x[s], y[s]} = ({x[s], y[s]} /. {wA[4], wA[6]})
```

$$\left\{\frac{e^{-s\alpha}\left(e^{2 s\alpha}\, x0\, \alpha - y0\, \beta + e^{2 s\alpha}\, y0\, \beta\right)}{\alpha},\ e^{-s\alpha}\, y0\right\}$$

An alternative approach is to solve for y(x)

```
wA[7] = DSolve[D[y[x], x] == (- α y[x])/(α x + 2 β y[x]), y[x], x]
```

$$\left\{\left\{y[x] \to \frac{-x\,\alpha - \sqrt{x^2\,\alpha^2 + 4\, e^{C[1]}\,\beta}}{2\,\beta}\right\},\ \left\{y[x] \to \frac{-x\,\alpha + \sqrt{x^2\,\alpha^2 + 4\, e^{C[1]}\,\beta}}{2\,\beta}\right\}\right\}$$

Determine the constant of integration.

```
wA[8] = wA[7][[1, 1]] /. x → x0 /. y[x0] → y0 // RE
```

$$y0 == \frac{-x0\,\alpha - \sqrt{x0^2\,\alpha^2 + 4\,e^{C[1]}\,\beta}}{2\,\beta}$$

In this example, choice of branch will determine whether the streamline is appropriate for one half plane or other

```
wA[9] = Solve[wA[8], C[1]][[1, 1]]
```

... Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

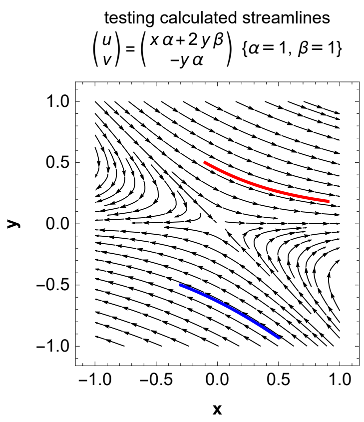
$$C[1] \to \mathrm{Log}\left[-\frac{-4\,x0\,y0\,\alpha\,\beta - 4\,y0^2\,\beta^2}{4\,\beta}\right]$$

```
wA[10] = wA[7][[1, 1]] /. wA[9] // Simplify
```

$$y[x] \to -\frac{x\,\alpha + \sqrt{x^2\,\alpha^2 + 4\,y0\,\beta\,(x0\,\alpha + y0\,\beta)}}{2\,\beta}$$

I illustrate that the derived forms are consistent with the streamlines generated by StreamPlot.

```
ShowStreamPlotWithCalculatedStreamlines[1, 1, "testing calculated streamlines"]
```



calculated streamlines
RED {x(s), y(s)}, BLUE y(x)

```
Clear[ShowStreamPlotWithCalculatedStreamlines];
ShowStreamPlotWithCalculatedStreamlines[αα_, ββ_, description_] :=
 Module[{vFlow, lab},
  vFlow[x_, y_, α_, β_] := {α x + 2 β y, -α y};
  lab = StringForm["``\n`` = ``  ``", description,
    TraditionalForm[(u
                     v)], TraditionalForm[(α x + 2 β y
                                           -α y)], {α == αα, β == ββ}];
  Module[{α = αα, β = ββ, R = 1, range, G},
   range = {{-R, R}, {-R, R}};
   (* streamline calculated with parametric form *)
   With[{x0 = -0.1, y0 = 0.5},
    G[1] = ParametricPlot[{1/α e^(-s α) (e^(2 s α) x0 α - y0 β + e^(2 s α) y0 β), e^(-s α) y0},
      {s, 0, 1}, PlotStyle → Red, PlotRange → range]];
   (* streamline calculated with y(x) form *)
   With[{x0 = -0.3, y0 = -0.5},
    G[2] = Plot[-1/(2 β) (x α + Sqrt[x^2 α^2 + 4 y0 β (x0 α + y0 β)]),
      {x, x0, 0.5}, PlotStyle → Blue, PlotRange → range]];
   G[3] = StreamPlot[vFlow[x, y, α, β], {x, -R, R}, {y, -R, R}, ImageSize → 200,
     StreamStyle → Black, FrameLabel → {{Stl["y"], ""}, {Stl["x"], lab}}];
   Labeled[Show[G[3], G[1], G[2]], Stl@StringForm[
      "calculated streamlines \n RED {x(s), y(s)}, BLUE y(x)"]]]]
```

Example 2: Streamlines, pathlines (trajectories) and streaklines for an unsteady flow
I work through Example (4-5) found at http://www.kau.edu.sa/Files/0057863/Subjects/Chapter%204.pdf

I define a function for the velocity flow

```
Clear[vFlowExample2];
vFlowExample2[x_, y_, ω_, t_] := {1/2 + 4/5 x, 3/2 + 5/2 Sin[ω t] - 4/5 y}
```

This flow is incompressible and irrotational but is unsteady (involves time t)

```
{Div[vFlowExample2[x, y, ω, t], {x, y}], Curl[vFlowExample2[x, y, ω, t], {x, y}]}
```

```
{0, 0}
```

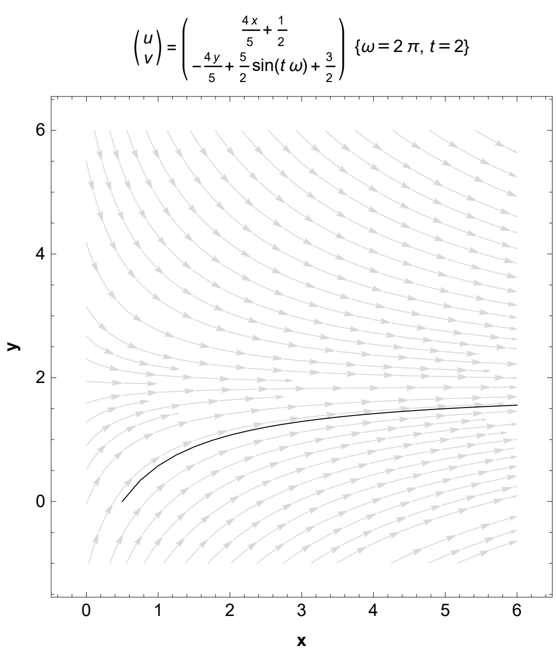
Streamlines can be obtained by solving

```
wA2[1] = DSolve[{D[y[x], x] == (3/2 + 5/2 Sin[ω t] - (4/5) y[x]) / (1/2 + (4/5) x),
    y[x0] == y0}, y[x], x][[1, 1]]
```

$$y[x] \to \frac{1}{5+8x}\left(15x - 15x0 + 5y0 + 8x0\,y0 + 25x\,\mathrm{Sin}[t\,\omega] - 25x0\,\mathrm{Sin}[t\,\omega]\right)$$

I check this formula by overlaying a representative streamline (starting at {1/2, 0}) on the figure generated by Mathematica’s StreamPlot

```
LAB = StringForm["`` = ``  ``", TraditionalForm[(u
                                                 v)],
  TraditionalForm[(          1/2 + 4/5 x
                   3/2 + 5/2 Sin[ω t] - 4/5 y)], {ω == 2 π, t == 2}];
Module[{ω = 2 π, t = 2, x0 = 0.5, y0 = 0, exampleStreamLine},
 exampleStreamLine =
  Line@Table[{x, 1/(5 + 8 x) (15 x - 15 x0 + 5 y0 + 8 x0 y0 + 25 x Sin[t ω] - 25 x0 Sin[t ω])},
    {x, 0.5, 6, 0.25}];
 GSTREAMLINE = StreamPlot[vFlowExample2[x, y, ω, t], {x, 0, 6},
   {y, -1, 6}, ImageSize → 300, StreamStyle → LightGray,
   FrameLabel → {{Stl["y"], ""}, {Stl["x"], LAB}}, Epilog → exampleStreamLine]]
```



Calculation of Pathlines: Pathlines are obtained by solving

$$\frac{\mathrm{dx}(t)}{\mathrm{dt}} = u(t)$$

$$\frac{\mathrm{dy}(t)}{\mathrm{dt}} = v(t)$$

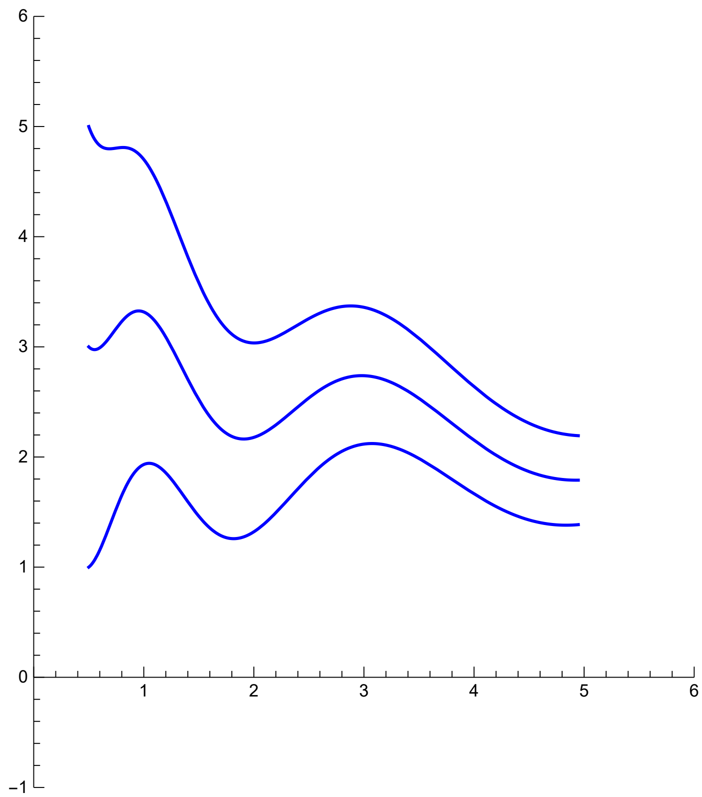
Equations for pathlines

While these equations can be solved analytically for this example, that would not generally be the case. I illustrate a numerical approach that would be broadly useful.

```
Clear[GeneratePathLine];
GeneratePathLine[x0_, y0_, tMax_, ω_] := NDSolve[
   {D[x[t], t] == 1/2 + 4/5 x[t], D[y[t], t] == 3/2 + 5/2 Sin[ω t] - (4/5) y[t],
    x[0] == x0, y[0] == y0}, {x[t], y[t]}, {t, 0, tMax}][[1]];
```

I generate some representative pathlines starting at {1/2,1}, {1/2,3}, {1/2,5}

```
Module[{ω = 2 π, tMax = 2, cases, soln, range},
 range = {{0, 6}, {-1, 6}};
 cases = {{1, {1/2, 1}}, {2, {1/2, 3 }}, {3, {1/2, 5 }}};
 (soln[#[[1]]] = GeneratePathLine[#[[2, 1]], #[[2, 2]], tMax, ω]) & /@ cases;
 GPATHLINES = ParametricPlot[{{x[t], y[t]} /. soln[1], {x[t], y[t]} /. soln[2],
    {x[t], y[t]} /. soln[3]}, {t, 0, 2}, PlotRange → range, PlotStyle → Blue] ]
```

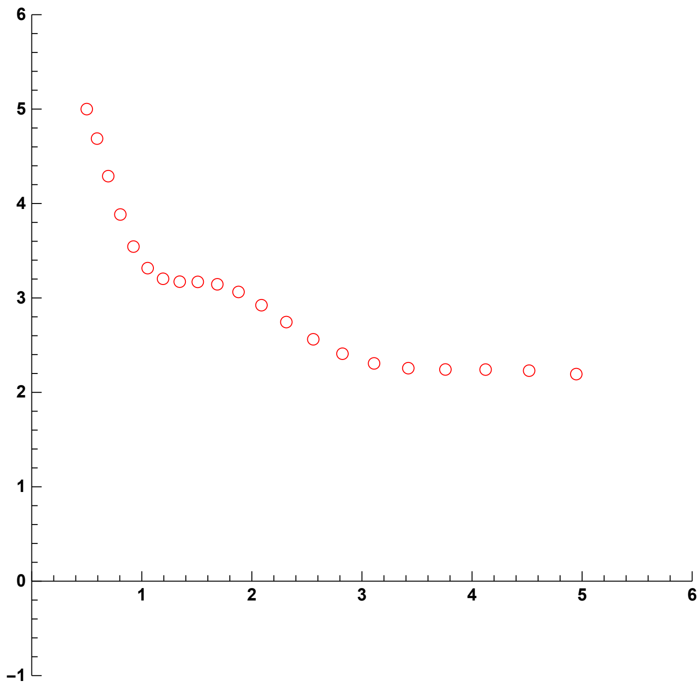


Calculation of streaklines: I define a numerical function that generates the streakline from {x(t0) = x0, y(t0) = y0} to {x(tMax), y(tMax)}. Remember that streaklines can be thought of as the locus of particle markers starting at {x0, y0} at different times.

```
Clear[GenerateStreakPoint];
GenerateStreakPoint[x0_, y0_, t0_, tMax_, ω_] := NDSolve[
   {D[x[t], t] == 1/2 + 4/5 x[t], D[y[t], t] == 3/2 + 5/2 Sin[ω t] - (4/5) y[t],
    x[t0] == x0, y[t0] == y0}, {x, y}, {t, t0, tMax}][[1]];
```

I generate points on the streakline corresponding to starting marker particles at {x0 = 1/2, y0 = 5} at times t0 = {0, 2.0, 0.25}

```
Module[{ω = 2 π, tMax = 2, soln, x0 = 1/2, y0 = 5, streakPoints, circles, range, G},
 range = {{0, 6}, {-1, 6}};

 streakPoints =
  Table[soln = GenerateStreakPoint[x0, y0, t0, tMax, ω];
   {t0, {x[tMax], y[tMax]} /. soln }, {t0, 0, tMax, 0.1}];
 circles =
  Tooltip[OC[#[[2]], Red], ToString@StringForm["t0 = ``", #[[1]]]] & /@ streakPoints;
 GSTREAKPOINTS = Graphics[circles, Axes → Automatic,
   AspectRatio → 1, PlotRange → range]
]
```

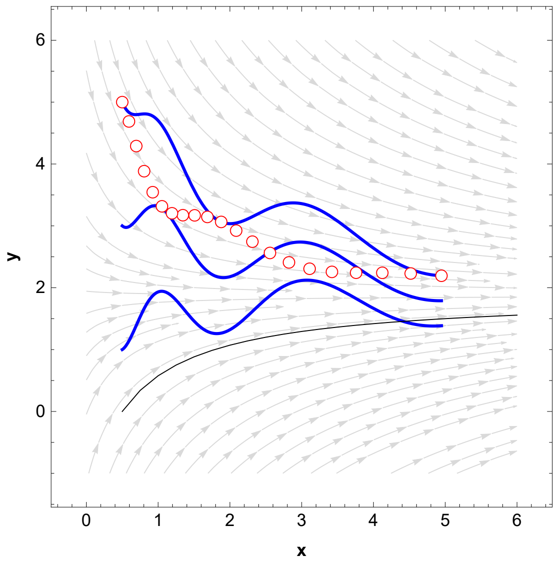


Note that mousing a red point pops up a “Tooltip” shows the time that the corresponding streak particle started at {x0, y0}.

Finally, I overlay the different types of lines and points on the original StreamPlot

```
Show[GSTREAMLINE, GPATHLINES, GSTREAKPOINTS]
```

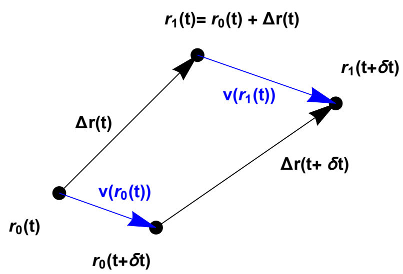
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{4x}{5} + \frac{1}{2} \\ -\frac{4y}{5} + \frac{5}{2}\sin(t\,\omega) + \frac{3}{2} \end{pmatrix} \{\omega = 2\pi,\ t = 2\}$$



B Rate of deformation tensor: Introduction

Introduction of the deformation tensor J

Consider how velocity changes with position



The velocity a small distance $\boldsymbol{\delta r}$ from a reference point $\boldsymbol{r_0}$ is approximately given by

$$\boldsymbol{v}(\boldsymbol{r_0} + \boldsymbol{\delta}\mathbf{r}) \simeq \boldsymbol{v}(\boldsymbol{r}_0) + \nabla \boldsymbol{v}(\boldsymbol{r}_0)\, \boldsymbol{\delta}\mathbf{r}$$

In the main section, I sketched the derivation of δr(t+δt) using vector notation that was coordinate independent. Just for fun, I carry out this expansion in Cartesian coordinates using Mathematica.

```
wB[1] = {u[{x0 + δx, y0 + δy}], v[{x0 + δx, y0 + δy}] }
```

```
{u[{x0 + δx, y0 + δy}], v[{x0 + δx, y0 + δy}]}
```

Expand in a Taylor series

```
wB[2] = wB[1] == (Normal@Series[wB[1], {δx, 0, 1}, {δy, 0, 1}] // Expand)
```

$$\{u[\{x0+\delta x,\ y0+\delta y\}],\ v[\{x0+\delta x,\ y0+\delta y\}]\} == \{u[\{x0,\ y0\}] + \delta y\, u^{(\{0,1\})}[\{x0,\ y0\}] + \delta x\, u^{(\{1,0\})}[\{x0,\ y0\}] + \delta x\, \delta y\, u^{(\{1,1\})}[\{x0,\ y0\}],\ v[\{x0,\ y0\}] + \delta y\, v^{(\{0,1\})}[\{x0,\ y0\}] + \delta x\, v^{(\{1,0\})}[\{x0,\ y0\}] + \delta x\, \delta y\, v^{(\{1,1\})}[\{x0,\ y0\}]\}$$

Truncate the expansion at first order in perturbed quantities

```
wB[3] = wB[2] /. {δx → ϵ δx, δy → ϵ δy} /. ϵ^2 → 0 /. ϵ → 1
```

$$\{u[\{x0+\delta x,\ y0+\delta y\}],\ v[\{x0+\delta x,\ y0+\delta y\}]\} == \{u[\{x0,\ y0\}] + \delta y\, u^{(\{0,1\})}[\{x0,\ y0\}] + \delta x\, u^{(\{1,0\})}[\{x0,\ y0\}],\ v[\{x0,\ y0\}] + \delta y\, v^{(\{0,1\})}[\{x0,\ y0\}] + \delta x\, v^{(\{1,0\})}[\{x0,\ y0\}]\}$$

Construct the explicit change in velocity

```
wB[4] = MapEqn[({#[[1]] - u[{x0, y0}], #[[2]] - v[{x0, y0}]}) &, wB[3]] /.
   -u[{x0, y0}] + u[{x0 + δx, y0 + δy}] → δu /. -v[{x0, y0}] + v[{x0 + δx, y0 + δy}] → δv
```

$$\{\delta u, \delta v\} == \left\{\delta y\, u^{(\{0,1\})}[\{x0, y0\}] + \delta x\, u^{(\{1,0\})}[\{x0, y0\}], \delta y\, v^{(\{0,1\})}[\{x0, y0\}] + \delta x\, v^{(\{1,0\})}[\{x0, y0\}]\right\}$$

The right hand side can be expressed as a matrix that I represent by **J**

```
wB[5] = J == wB[4][[2]] /. Plus → List /. {δx → 1, δy → 1};
wB[5][[2]] // MatrixForm
```

$$\begin{pmatrix} u^{(\{0,1\})}[\{x0, y0\}] & u^{(\{1,0\})}[\{x0, y0\}] \\ v^{(\{0,1\})}[\{x0, y0\}] & v^{(\{1,0\})}[\{x0, y0\}] \end{pmatrix}$$

```
wB[6] = wB[4][[1]] == Inactive[Dot][J, {δx, δy}] /. (wB[5] // ER)
```

$$\{\delta u, \delta v\} == \left\{\left\{u^{(\{0,1\})}[\{x0, y0\}], u^{(\{1,0\})}[\{x0, y0\}]\right\}, \left\{v^{(\{0,1\})}[\{x0, y0\}], v^{(\{1,0\})}[\{x0, y0\}]\right\}\right\}.\{\delta x, \delta y\}$$

In standard notation

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}_{x0,y0} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix} \equiv J_{ij}\, \delta r_j$$

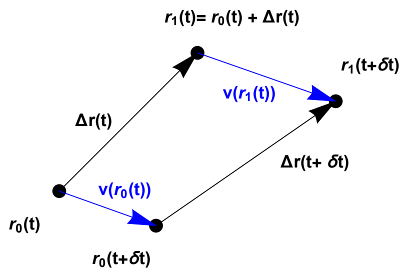
The matrix J is called the rate of deformation tensor. The term “rate” is used because the displacement tensor X_{ij} is related to J by

$$dX_{ij} = J_{ij}\, dt$$

Creation of visualization of the deformation of a vector.

```
Clear[ShowPoint, ShowVector, vFlow];
ShowPoint[pt_, text_, off_] :=
  {Black, PointSize[0.02], Point[pt], Text[Style[text, Bold], pt + off]};
ShowVector[ptStart_, ptFin_, text_, off_] :=
  {Arrow[{ptStart, ptFin}], Text[Style[text, Bold], (ptStart + ptFin)/2 + off]};
```

```
vFlow[{x_, y_}, α_, β_] := {α x + 2 β y, - α y};
Module[{α = 1, β = 1, x0 = 0.4, y0 = 0.5, δx = 0.2, δy = 0.2,
  δt = 0.1, r0, δr, r1, v0, r0tpδt, r1tpδt, rObjects, vObjects},
 (* starting position of vector *)
 r0 = {x0, y0};
 δr = {δx, δy};
 r1 = r0 + δr;
 (* vector after flow over interval δt *)
 r0tpδt = r0 + vFlow[r0, α, β] δt;
 r1tpδt = r1 + vFlow[r1, α, β] δt;

 rObjects = {ShowPoint[r0, "r0(t)", {-0.05, -0.05}],
   ShowPoint[r1, "r1(t)= r0(t) + Δr(t)", {0.05, 0.05}],
   ShowPoint[r0tpδt , "r0(t+δt)", {-0.05, -0.05}],
   ShowPoint[r1tpδt , "r1(t+δt)", {0.05, 0.05}],
   ShowVector[r0, r1, "Δr(t)", {-0.05, 0}],
   ShowVector[r0tpδt, r1tpδt, "Δr(t+ δt)", {0.1, 0}]};
 vObjects = {Blue, ShowVector[r0, r0tpδt, "v(r0(t))", {0.025, 0.025}],
   ShowVector[r1, r1tpδt, "v(r1(t))", {-0.025, -0.025}]};
 Graphics[{rObjects, vObjects},
  Axes → None, AspectRatio → 1, PlotRange → {{0, 1}, {0, 1}}]]
```



C Rate of deformation tensor: Decomposition

The rate of deformation tensor can be decomposed into parts that have different physical effects on the fluid.

```
J = {{D[vx[x, y], x], D[vy[x, y], x]}, {D[vx[x, y], y], D[vy[x, y], y]}};
J // MatrixForm
```

$$\begin{pmatrix} vx^{(1,0)}[x,y] & vy^{(1,0)}[x,y] \\ vx^{(0,1)}[x,y] & vy^{(0,1)}[x,y] \end{pmatrix}$$

This is decomposed into symmetry and anti-symmetric forms. The symmetric part is the rate of strain tensor

```
wC[2] = ϵij == 1/2 (J + Transpose[J]);
wC[2][[2]] // MatrixForm
```

$$\begin{pmatrix} vx^{(1,0)}[x,y] & \frac{1}{2}\left(vx^{(0,1)}[x,y]+vy^{(1,0)}[x,y]\right) \\ \frac{1}{2}\left(vx^{(0,1)}[x,y]+vy^{(1,0)}[x,y]\right) & vy^{(0,1)}[x,y] \end{pmatrix}$$

The anti-symmetric part is the rate of rotation tensor

```
wC[3] = ωij == 1/2 (J - Transpose[J]);
wC[3][[2]] // MatrixForm
```

$$\begin{pmatrix} 0 & \frac{1}{2}\left(-vx^{(0,1)}[x,y]+vy^{(1,0)}[x,y]\right) \\ \frac{1}{2}\left(vx^{(0,1)}[x,y]-vy^{(1,0)}[x,y]\right) & 0 \end{pmatrix}$$

The diagonal part of the strain tensor is the rate of expansion tensor

```
wC[4] = tij == 1/2 Tr[wC[2][[2]]] IdentityMatrix[2];
wC[4][[2]] // MatrixForm
```

$$\begin{pmatrix} \frac{1}{2}\left(vy^{(0,1)}[x,y]+vx^{(1,0)}[x,y]\right) & 0 \\ 0 & \frac{1}{2}\left(vy^{(0,1)}[x,y]+vx^{(1,0)}[x,y]\right) \end{pmatrix}$$

The off-diagonal part of the strain tensor is the rate of shear tensor

```
wC[5] = σij == wC[2][[2]] - 1/2 Tr[wC[2][[2]]] IdentityMatrix[2];
wC[5][[2]] // MatrixForm
```

$$\begin{pmatrix} \frac{1}{2}\left(-vy^{(0,1)}[x,y]-vx^{(1,0)}[x,y]\right)+vx^{(1,0)}[x,y] & \frac{1}{2}\left(vx^{(0,1)}[x,y]+vy^{(1,0)}[x,y]\right) \\ \frac{1}{2}\left(vx^{(0,1)}[x,y]+vy^{(1,0)}[x,y]\right) & vy^{(0,1)}[x,y]+\frac{1}{2}\left(-vy^{(0,1)}[x,y]-vx^{(1,0)}[x,\right. \end{pmatrix}$$

I define functions for constructing and decomposing the deformation matrix

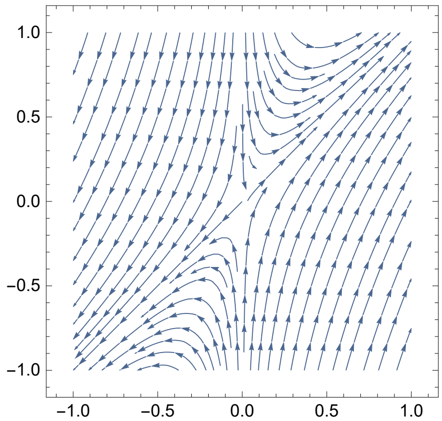
```
Clear[ConstructDeformationMatrix, DecomposeMatrix];
ConstructDeformationMatrix[{vx_, vy_}] :=
  Module[{J},
   J = {{D[vx, x], D[vy, x]}, {D[vx, y], D[vy, y]}}];


DecomposeMatrix[J_] :=
 Module[{ϵij, ωij, tij, σij},
  (* symmetric part *)
  ϵij = 1/2 (J + Transpose[J]);
  (* anti- symmetric part *)
  ωij = 1/2 (J - Transpose[J]);
  (* trace *)
  tij = 1/2 Tr[ϵij] {{1, 0}, {0, 1}};
  (* off diagonal part *)
  σij = ϵij - tij;
  {ϵij, ωij, tij, σij}]
```

I illustrate the motions associated with these tensors using the example flow considered above.

```
Clear[vFlow];
vFlow[x_, y_, α_, β_] := {α x + 2 β y, - α y};

Module[{J, ϵ, ω, t, σ, lab, G},
 J = ConstructDeformationMatrix[vFlow[x, y, α, β]];
 {ϵ, ω, t, σ} = DecomposeMatrix[J];

 Module[{αα = 1, ββ = 1, size = 225, JNum, ϵNum, ωNum, tNum, σNum},

  JNum = ConstructDeformationMatrix[vFlow[x, y, αα, ββ]];
  {ϵNum, ωNum, tNum, σNum} = {ϵ, ω, t, σ} /. {α → αα, β → ββ};

  lab["J"] = StringForm["Rate of deformation tensor\n`` = `` α = `` β = ``",
    "Jij", MatrixForm[J], αα, ββ];
  G["J"] = Labeled[StreamPlot[JNum . {δx, δy}, {δx, -1, 1},
     {δy , -1, 1}, ImageSize → size], lab["J"], Top];
  lab["ϵ"] = StringForm["Rate of strain tensor\n`` = `` α = `` β = ``",
    "ϵij", MatrixForm[ϵ], αα, ββ];
  G["ϵ"] = Labeled[StreamPlot[ϵNum . {δx, δy}, {δx, -1, 1},
     {δy , -1, 1}, ImageSize → size], lab["ϵ"], Top];
  lab["ω"] = StringForm["Vorticity tensor\n`` = `` α = `` β = ``",
    "ωij", MatrixForm[ω], αα, ββ];
  G["ω"] = Labeled[StreamPlot[ωNum . {δx, δy}, {δx, -1, 1},
     {δy , -1, 1}, ImageSize → size], lab["ω"], Top];
  lab["t"] = StringForm["Rate of translation tensor\n`` = `` α = `` β = ``",
    "tij", MatrixForm[t], αα, ββ];
  G["t"] = Labeled[StreamPlot[tNum . {δx, δy}, {δx, -1, 1},
     {δy , -1, 1}, ImageSize → size], lab["t"], Top];
  lab["σ"] = StringForm["Rate of shear tensor\n`` = `` α = `` β = ``",
    "σij", MatrixForm[σ], αα, ββ];
  G["σ"] = Labeled[StreamPlot[σNum . {δx, δy}, {δx, -1, 1},
     {δy , -1, 1}, ImageSize → size], lab["σ"], Top];
  Grid[{{G["J"], G["ϵ"], G["ω"]}, {G["t"], G["σ"]}}]]]
```

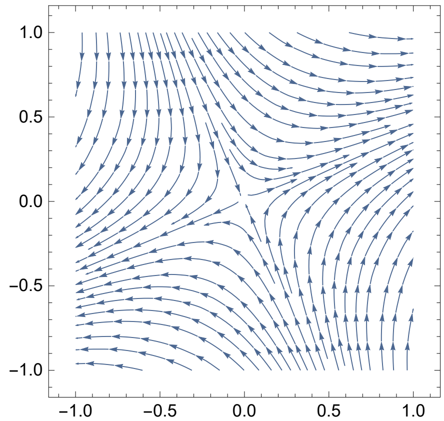
Rate of deformation tensor

$J_{ij} = \begin{pmatrix} \alpha & 0 \\ 2\beta & -\alpha \end{pmatrix}$ $\alpha = 1$ $\beta = 1$



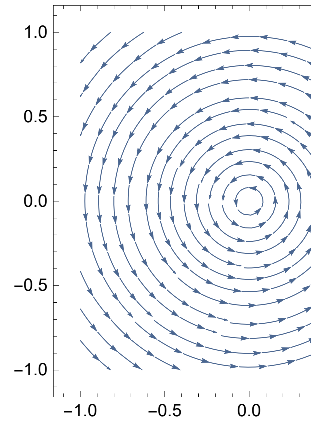
Rate of strain tensor

$\epsilon_{ij} = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix}$ $\alpha = 1$ $\beta = 1$



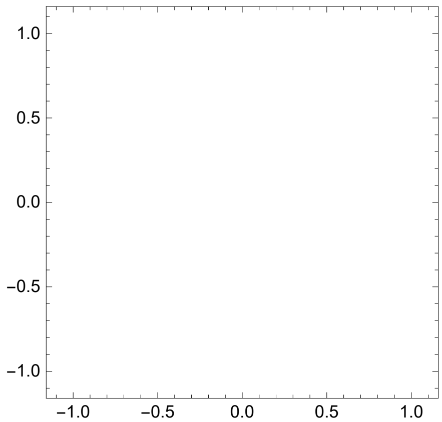
Vorticity tensor

$\omega_{ij} = \begin{pmatrix} 0 & -\beta \\ \beta & 0 \end{pmatrix}$ $\alpha = 1$



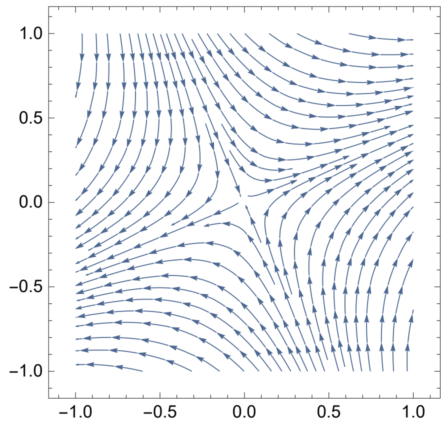
Rate of translation tensor

$t_{ij} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ $\alpha = 1$ $\beta = 1$



Rate of shear tensor

$\sigma_{ij} = \begin{pmatrix} \alpha & \beta \\ \beta & -\alpha \end{pmatrix}$ $\alpha = 1$ $\beta = 1$

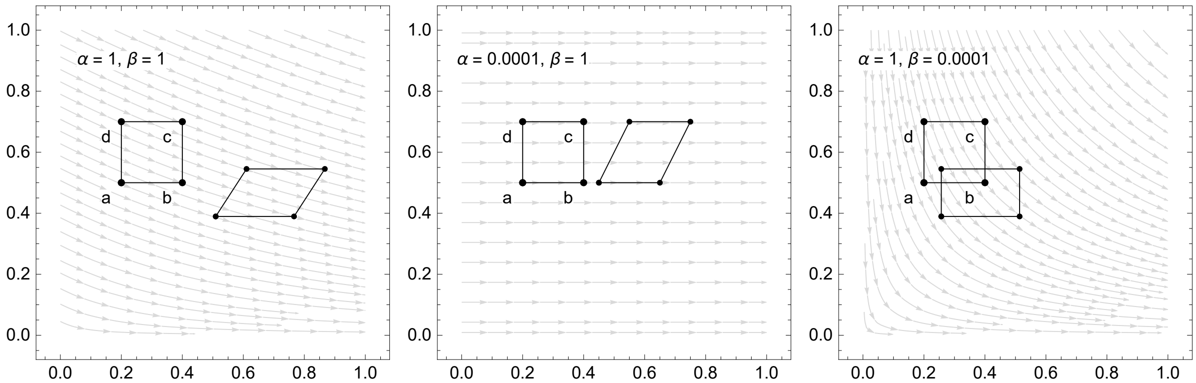


D Illustrations of deformation

To illustrate deformation in a flow field, I use the same example flow field and associated streamlines derived in previous sections. However, it is relatively straightforward to modify the code below to incorporate other flow models.

The first objective is to just visualize the deformation of the representative square as it moves in the flow field. Details of the deformations will follow.

```
Grid[{{ShowReferenceRectangle[1, 1],
   ShowReferenceRectangle[0.0001, 1], ShowReferenceRectangle[1, 0.0001]}}]
```




```
Clear[ShowReferenceRectangle];
ShowReferenceRectangle[α_, β_] := Module[{x0 = 0.2, y0 = 0.5, Δx = 0.2,
    Δy = 0.2, refPoints, movedRefPoints, DisplayRefPoint, text, lab, G},
  DisplayRefPoint[{x_, y_, lab_}] := {PointSize[0.02],
    Point[{x, y}], Text[lab, {x, y} + {-0.05, -0.05}]};

  refPoints = {{x0, y0, "a"}, {x0 + Δx, y0, "b"},
    {x0 + Δx, y0 + Δy, "c"}, {x0, y0 + Δy, "d"}, {x0, y0, "a"}};
  movedRefPoints = With[{s = 0.25},
    StreamLineExample[s, #, α, β] & /@ refPoints ⟦All, {1, 2}⟧];
  text = StringForm["α = ``, β = ``", α, β];
  lab = Text[text, {0.2, 0.9}];
  G[1] =
   Graphics[{DisplayRefPoint /@ refPoints, Line[refPoints ⟦All, {1, 2}⟧ ], Line@
      movedRefPoints , Point /@ movedRefPoints, lab}, PlotRange → {{0, 1}, {0, 1}}];
  G[2] = StreamPlotBackground[α, β];
  Show[G[2], G[1]]]
```

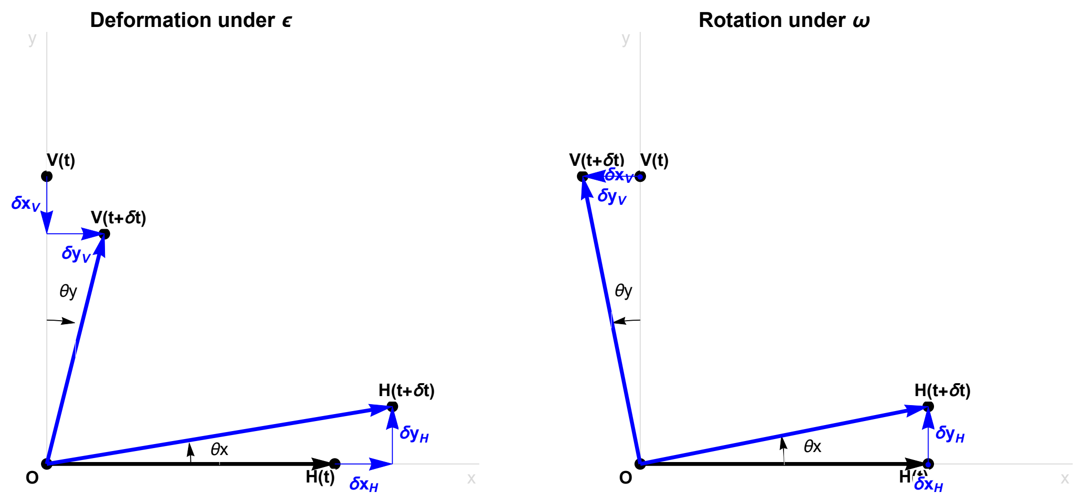
```
Clear[vFlowExample, StreamLineExample, StreamPlotBackground];
vFlowExample[x_, y_, α_, β_] := {α x + 2 β y, -α y};
StreamLineExample[s_, {x0_, y0_}, α_, β_] :=
  {(1/α) e^(-s α) (e^(2 s α) x0 α - y0 β + e^(2 s α) y0 β), e^(-s α) y0};
StreamPlotBackground[αα_, ββ_] :=
  Module[{α = αα, β = ββ, R = 1},
   StreamPlot[vFlowExample[x, y, α, β],
    {x, 0, R}, {y, 0, R}, ImageSize → 200, StreamStyle → LightGray]];
```

Distortion of the coordinate system

```
Clear[vFlow];
vFlow[x_, y_, α_, β_] := {α x + 2 β y, - α y};
```

```
Clear[ShowPoint, ShowVector, ShowArc];
ShowPoint[{pt_, text_, off_}] :=
  {Black, PointSize[0.02], Point[pt], Text[Style[text, Bold], pt + off]};
ShowVector[ptStart_, ptFin_, text_, off_] :=
  {Arrow[{ptStart, ptFin}], Text[Style[text, Bold], (ptStart + ptFin)/2 + off]};
ShowArc[O_, r_, θStart_, θFin_] :=
 Module[{PtoC},
  PtoC[rr_, θ_] := rr {Cos[θ], Sin[θ]};
   {Arrowheads[Small] ,
    Arrow@Table[O + PtoC[r, θ], {θ, θStart, θFin, Sign[θFin - θStart] π/64}]}]
```

```
Module[{α = 1, β = 1, vx, vy, J, ϵ, ω, lab, G},
 {vx[x, y], vy[x, y]} = vFlow[x, y, α, β];
 J = {{D[vx[x, y], x], D[vy[x, y], x]}, {D[vx[x, y], y], D[vy[x, y], y]}};
 ϵ = 1/2 (J + Transpose[J]);
 ω = 1/2 (J - Transpose[J]);
 lab = Stl@StringForm["Deformation under ϵ"];
 G[1] = ShowDeformationRotation[ϵ, lab];
 lab = Stl@StringForm["Rotation under ω"];
 G[2] = ShowDeformationRotation[ω, lab];
 Grid[{{G[1], G[2]}}]]
```



```
Clear[ShowDeformationRotation];
ShowDeformationRotation[Amat_, lab_] :=
 Module[{Δx = 1, Δy = 1, δt = 0.2, O = {0, 0}, R = 1.5, r0, r1, J, ϵ, ω, H
   (*H(t)*), Hp (*H(t+δt)*), OHLine, OHpLine, δxHLine, δyHLine, V, Vp,
   OVLine, OVpLine, δxVLine, δyVLine , axes, range, points, θxArc, θyArc, θPolar},
  θPolar[{xx_, yy_}] := ToPolarCoordinates[{xx, yy}][[2]];
  range = {{-R/3, R}, {-R/3, R}};
  axes = {LightGray, Line /@ {{O, {R, 0}}, {O, {0, R}}},
    Text["x", {R, 0} + {-0.025, -0.05}], Text["y", {0, R} + {-0.05, -0.025}]};

  (* transformation of horizontal vector *)
  H = {Δx, 0};
  Hp = H + Dot[Amat, H] δt;
  OHLine = {Black, Thick, ShowVector[O, H, "", {0, -0.075}]};
  OHpLine = {Blue, Thick, ShowVector[O, Hp, "", {0, -0.075}]};
  δxHLine = {Blue, ShowVector[H, {Hp[[1]], 0}, "δxH", {0, -0.075}]};
  δyHLine = {Blue, ShowVector[{Hp[[1]], 0}, Hp, "δyH", {0.075, 0.0}]};
  (* transformation of vertical vector *)
  V = {0, Δy};
  Vp = V + Dot[Amat, V] δt;
  OVLine = {Black, Thick, ShowVector[O, V, "", {0, 0}]};
  OVpLine = {Blue, Thick, ShowVector[O, Vp, "", {0, 0}]};
  δxVLine = {Blue, ShowVector[V, {0, Vp[[2]]}, "δxV", {-0.075, 0}]};
  δyVLine = {Blue, ShowVector[{0, Vp[[2]]}, Vp, "δyV", {0, -0.075}]};

  points = ShowPoint /@
    {{O, "O", {-0.05, -0.05}}, {H, "H(t)", {-0.05, -0.05}}, {V, "V(t)",
      {0.05, 0.05}}, {Hp, "H(t+δt)", {0.05, 0.05}}, {Vp, "V(t+δt)", {0.05, 0.05}}};

  θxArc = With[{r = Δx/2, θStart = θPolar[H] , θFin = θPolar[Hp]},
    {ShowArc[O, r, θStart, θFin],
     Text["θx", O + 1.2 r {Cos[(θStart + θFin)/2], Sin[(θStart + θFin)/2]}]}];
  θyArc = With[{r = Δx/2, θStart = θPolar[V] , θFin = θPolar[Vp]},
    {ShowArc[O, r, θStart, θFin],
     Text["θy", O + 1.2 r {Cos[(θStart + θFin)/2], Sin[(θStart + θFin)/2]}]}];

  Graphics[{axes, points , θxArc, θyArc, OHLine, OHpLine,
    δxHLine, δyHLine, OVpLine, δxVLine, δyVLine}, Axes → None,
   AspectRatio → 1, PlotRange → range, PlotLabel → lab, ImageSize → 300]]
```

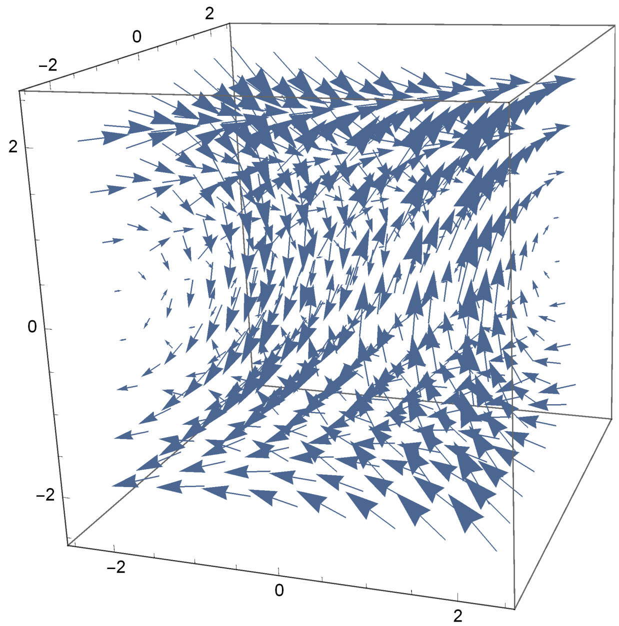
E Illustration of 3-D streamline

I construct a representative streamline for the 3D flow

$$\{u, v, w\} = \{z, z, x - y\}$$

StreamPlot only works for 3-D, so I visualize the 3-D flow using VectorPlot3D.

```
Module[{R = 2},
 VectorPlot3D[{z, z, x - y}, {x, -R, R}, {y, -R, R}, {z, -R, R}]]
```



I derive a closed form parametric expression for the streamline. Relate x and z with

```
Clear[x, y]
```

```
wE[1] = DSolve[{D[x[s], s]/z[s] == 1, D[y[s], s]/z[s] == 1}, {x[s], y[s]}, s][[1]] /.
   {K[1] → τ, K[2] → τ, C[1] → 𝒜, C[2] → ℬ} // RE
```

$$\left\{x[s] == \mathcal{A} + \int_1^s z[\tau]\, d\tau,\ y[s] == \mathcal{B} + \int_1^s z[\tau]\, d\tau\right\}$$

Eliminate the common integral expression

```
wE[2] = Solve[wE[1][[1]], ∫_1^s z[τ] dτ][[1, 1]]
```

$$\int_1^s z[\tau]\, d\tau \rightarrow -\mathcal{A} + x[s]$$

```
wE[3] = wE[1][[2]] /. wE[2]
```

$$y[s] == -\mathcal{A} + \mathcal{B} + x[s]$$

Redefine the constant of integration

```
wE[4] = wE[3] /. -𝒜 + ℬ → 𝒞
```

$y[s] == \mathcal{C} + x[s]$

Next relate x and z

```
wE[5] = DSolve[{D[x[s], s]/z[s] == 1, D[z[s], s]/(x[s] - (𝒞 + x[s])) == 1}, {x[s], z[s]}, s][[1]] // RE
```

$\left\{x[s] == -\frac{s^2\,\mathcal{C}}{2} + C[1] + s\,C[2],\ z[s] == -s\,\mathcal{C} + C[2]\right\}$

I want to express the constants of integration in terms of the starting point of a given streamline.

```
wE[6] = wE[4] /. s → 0 /. {x[0] → x0, y[0] → y0}
```

$y0 == x0 + \mathcal{C}$

```
wE[7] = Solve[wE[6], 𝒞][[1, 1]]
```

$\mathcal{C} \to -x0 + y0$

Thus, x and y are related by

```
wE[8] = wE[4] /. wE[7]
```

$y[s] == -x0 + y0 + x[s]$

Now the other constants of integration are determined

```
wE[9] = wE[5] /. wE[7]
```

$\left\{x[s] == -\frac{1}{2}\,s^2\,(-x0 + y0) + C[1] + s\,C[2],\ z[s] == -s\,(-x0 + y0) + C[2]\right\}$

```
wE[10] = wE[9] /. s → 0 /. {x[0] → x0, z[0] → z0}
```

$\{x0 == C[1],\ z0 == C[2]\}$

```
wE[11] = Solve[wE[10], {C[1], C[2]}][[1]]
```

$\{C[1] \to x0,\ C[2] \to z0\}$

x and z are given by

```
wE[12] = wE[9] /. wE[11]
```

$$\left\{x[s] == x0 - \frac{1}{2} s^2 \left(-x0 + y0\right) + s\, z0,\ z[s] == -s \left(-x0 + y0\right) + z0\right\}$$

and y was given by

```
wE[13] = wE[8] /. (wE[12][[1]] // ER)
```

$$y[s] == y0 - \frac{1}{2} s^2 \left(-x0 + y0\right) + s\, z0$$

Combining these

```
wE[14] = {wE[12][[1]], wE[13], wE[12][[2]]}
```

$$\left\{x[s] == x0 - \frac{1}{2} s^2 \left(-x0 + y0\right) + s\, z0,\ y[s] == y0 - \frac{1}{2} s^2 \left(-x0 + y0\right) + s\, z0,\ z[s] == -s \left(-x0 + y0\right) + z0\right\}$$

or, finally,

```
wE[15] = MapEqn[Simplify, #] & /@ wE[14];
wE[15] // ColumnForm
```

$$x[s] == x0 + \tfrac{1}{2} s^2 \left(x0 - y0\right) + s\, z0$$
$$y[s] == \tfrac{1}{2} s^2 \left(x0 - y0\right) + y0 + s\, z0$$
$$z[s] == s \left(x0 - y0\right) + z0$$

I collect the right hand sides for numerical purposes

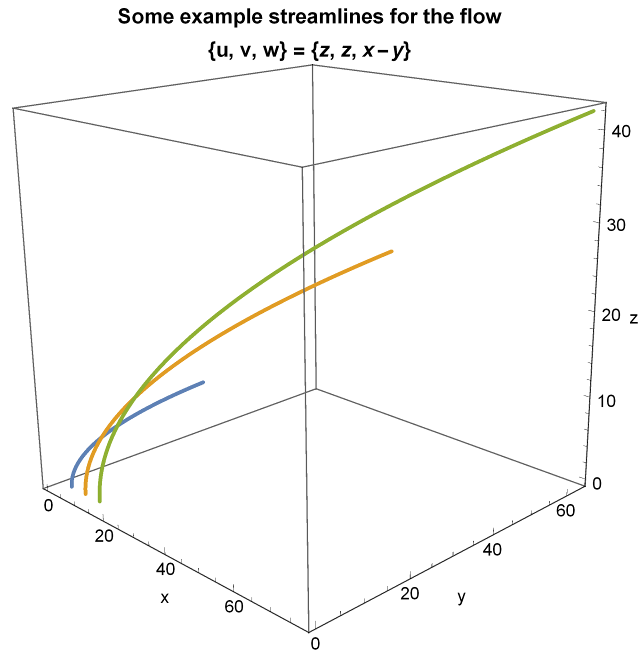
```
wE[16] = #[[2]] & /@ wE[15]
```

$$\left\{x0 + \frac{1}{2} s^2 \left(x0 - y0\right) + s\, z0,\ \frac{1}{2} s^2 \left(x0 - y0\right) + y0 + s\, z0,\ s \left(x0 - y0\right) + z0\right\}$$

```
Clear[ExampleStreamLine3D];
ExampleStreamLine3D[s_, {x0_, y0_, z0_}] :=
 {x0 + 1/2 s^2 (x0 - y0) + s z0, 1/2 s^2 (x0 - y0) + y0 + s z0, s (x0 - y0) + z0}
```

I visualize some representative streamlines

```
ParametricPlot3D[{ExampleStreamLine3D[s, {5, 1, 0}],
  ExampleStreamLine3D[s, {10, 1, 0}], ExampleStreamLine3D[s, {15, 1, 0}]},
 {s, 0, 3}, AxesLabel → {"x", "y", "z"}, BoxRatios → {1, 1, 1}, PlotLabel → Stl[
   StringForm["Some example streamlines for the flow\n{u, v, w} = ``", {z, z, x - y}] ]]
```



To illustrate flow deformation near some point, I need to generate a reference coordinate frame. I draw on differential geometry

From C : \NT G\2017\Analysis\Differential Geometry 01 - 15 - 17

```
Clear[FrenetSerretFrameAssociation];
FrenetSerretFrameAssociation[F_, s_] :=
 Module[{function, df, d2f, d3f, vTangent, vBinormal,
   vNormal, κ (*curvature*), τ(*torsion*), names, values},
  {function, df, d2f, d3f} = {F[s], D[F[s], s], D[F[s], {s, 2}], D[F[s], {s, 3}]};
  vTangent = Simplify[df / Norm[df], s ∈ Reals];
  vBinormal =
   With[{temp = Cross[df, d2f]}, Simplify[temp / Norm[temp], s ∈ Reals]];
  vNormal = With[{temp = Cross[vBinormal, vTangent]},
    Simplify[temp / Norm[temp], s ∈ Reals]];
  κ = Simplify[Norm[Cross[df, d2f]] / Norm[d2f]^3, s ∈ Reals];
  τ = Simplify[Dot[df, Cross[d2f, d3f]] / Norm[Cross[df, d2f]]^2, s ∈ Reals];
  names = {"function", "tangent", "normal", "binormal", "curvature", "torsion"};
  values = {function, vTangent, vNormal, vBinormal, κ, τ};
  AssociationThread[names, values]]
```

I generate the parametric equation for a particular set of initial conditions

```
ExampleStreamLine3D[s, {5, 1, 0}]
```

$\{5 + 2 s^2, 1 + 2 s^2, 4 s\}$

```
Clear[F, A];
F[s_] := {5 + 2 s^2, 1 + 2 s^2, 4 s};
A = FrenetSerretFrameAssociation[F, s]
```

$\langle|$ function $\to \{5 + 2 s^2, 1 + 2 s^2, 4 s\}$, tangent $\to \left\{\frac{s}{\sqrt{1+2 s^2}}, \frac{s}{\sqrt{1+2 s^2}}, \frac{1}{\sqrt{1+2 s^2}}\right\}$,

normal $\to \left\{\frac{1}{\sqrt{2+4 s^2}}, \frac{1}{\sqrt{2+4 s^2}}, -\frac{\sqrt{2}\, s}{\sqrt{1+2 s^2}}\right\}$,

binormal $\to \left\{-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right\}$, curvature $\to \frac{1}{8}$, torsion $\to 0 |\rangle$

I illustrate a Cartesian coordinate system at a particular point along the streamline. I also illustrate the plane normal to the streamline at that point.

```
Module[{ s0 = 3, scale = 5, P, ptP, vecTNB,  plane, g},
 P = F[s0];
 ptP = {Red, PointSize[0.02], Point[P]};
 vecTNB = {Arrowheads[Medium], {Red, Thick, Arrow[{P, P + scale A["tangent"]}]},
    {Blue, Thick, Arrow[{P, P + scale A["normal"]}]},
    {Darker[Green, 0.5], Thick, Arrow[{P, P + scale A["binormal"]}]}} /. s → s0 ;
 g[1] = ParametricPlot3D[F[s], {s, 0, 4}, PlotStyle → Black,
   BoxRatios → {1, 1, 1}, AxesLabel → {"x", "y", "z"}];
 plane = InfinitePlane[{P, P + scale A["normal"], P + scale A["binormal"]}] /. s → s0;
 g[2] = Graphics3D[{ptP, vecTNB, {LightYellow, plane}}, BoxRatios → {1, 1, 1}];
 Show[{g[1], g[2]}]]
```