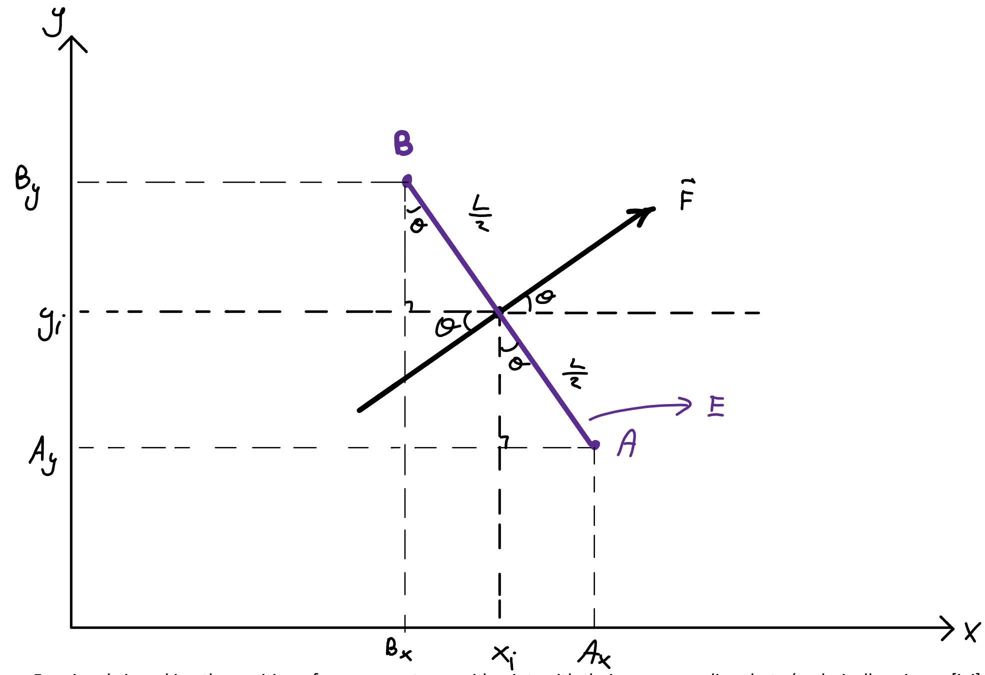
Geometry used in defining stacks from quiver arrows and their arrowheads

05 July 2021 11:28

To create a stacks. We require sheets (line in 2D), perpendicular to arrow from quiver whose density is determined by arrow magnitude.

Having the positions of the grid in x_grid and y_grid from np.meshgrid and having the magnitudes at each point in x and y (or r and theta etc.). Can define the stacks as:

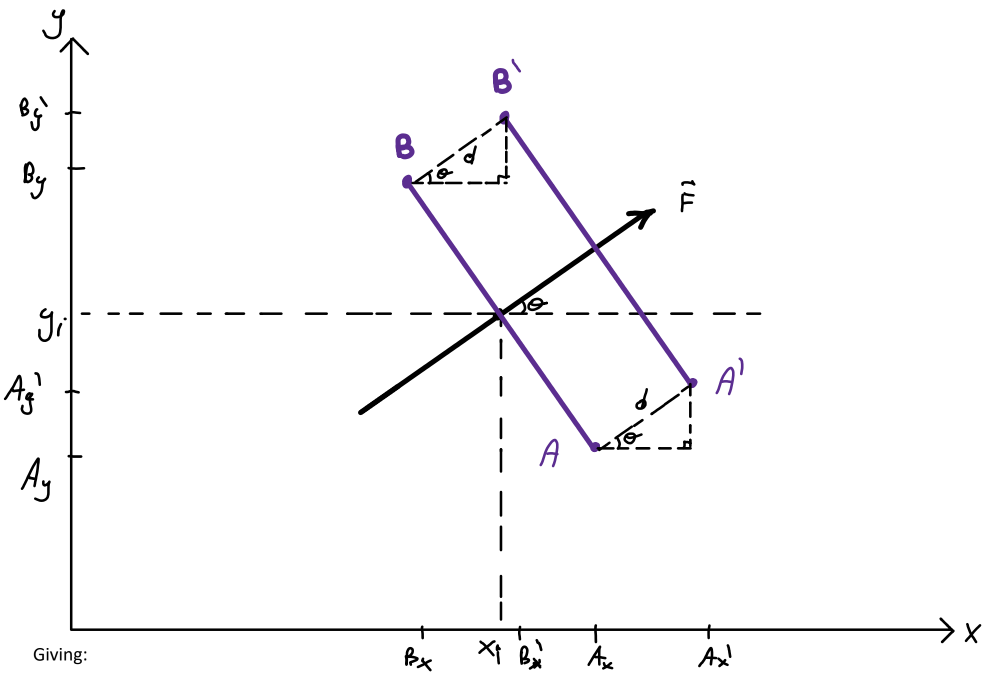


For yi and xi marking the position of an arrow, at any grid point, with their corresponding theta (technically x_i -= xg[i, j], y_i = yg[i, j], theta_i = theta[i, j] and **F** = **F**[i, j])
The stack is shown in purple, with its endpoints, A and B, at a perp. distance to the arrow L (defined as a fraction of graph scale).

From these:
A_x = x + (L/2) * sin(theta)
A_y = y - (L/2) * cos(theta)
B_x = x - (L/2) * sin(theta)
B_y = y + (L/2) * cos(theta)

These also function generally as operations that displace a point on the vector in the direction perp. to the arrow, by a corr. length - here (L/2)

To then displace the stack sheet in the direction parallel to arrow:

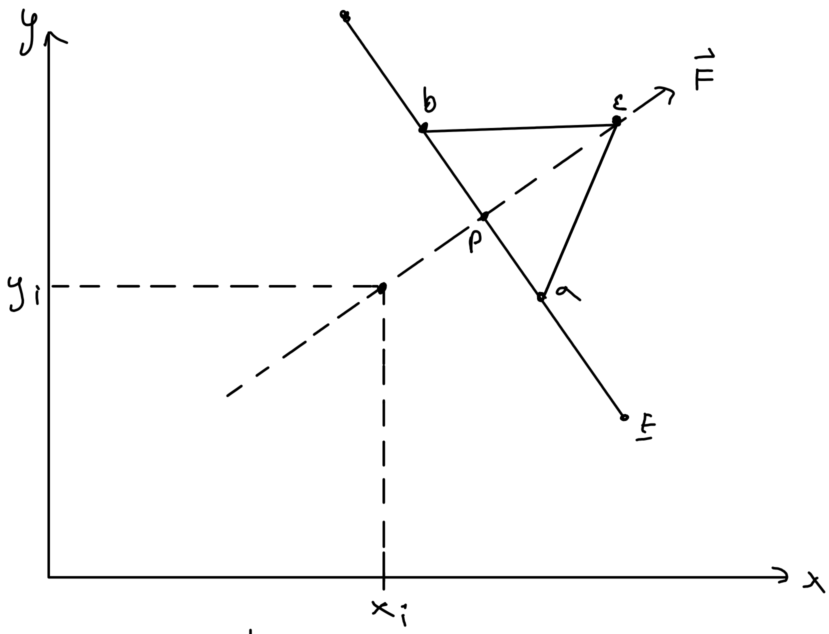


Giving:

A_x' = A_x + d*cos(theta)
A_y' = A_y + d*sin(theta)
B_x' = B_x + d * cos(theta)
B_y' = B_y + d * sin(theta)

Which again function as general operations for such parallel displacements.

Defining the arrowhead works using these 2 operations:



$$p = \Big(\underbrace{x_i + \frac{L_s}{2}\cos\theta}_{p_x},\ \underbrace{y_i + \frac{L_s}{2}\sin\theta}_{p_y}\Big)$$

$$a = \left(p_x + \frac{L}{8}\sin\theta,\ p_y - \frac{L}{8}\cos\theta\right)$$

$$b = \left(p_x - \frac{L}{8}\sin\theta,\ p_y + \frac{L}{8}\cos\theta\right)$$

$$c = \left(x + \frac{L_s}{4}\cos\theta,\ y + \frac{L_s}{4}\sin\theta\right)$$

With previously defined parameters. Note - for the single sheet stacks, the displacement from (x_i, y_i) to p needs to be omitted.

General from of stack positions

There probably exists a nicer way of doing this, but We approached the problem by considering odd and even number of stacks. We defined them by displacing stack sheets a certain lengths (with above equations), from the middle position.

For the **odd number of stack sheets**, the following pattern was noticed:
For 1 sheet: no displacing (+- 0) ----------> from the middle, therefore no displacing
For 3 sheets: (+-0) and (+- 0.5) ---------> technically 0.5*(total possible limiting length of stack, determined by user, as fraction of graph size)
For 5 sheets: (+-0), (+-0.5) and (+-0.25)
For 7 sheets: (+-0), (+-0.5), (+-2/6) and (+- 1/6) -----------> At this point, note that 3/6 = 0.5 0.5 (obvious but important realisation)
For 9 sheets: (+-0), (+-0.5), (+-3/8), (+- 1/4) and (+-1/8) ---------> Again, 1/4 = 2/8 and 1/2 = 4/8
For 11 sheets: (+-0), (+-0.5), (+-4/10), (+- 3/10), (+-2/10) and (+-1/10)
Etc.

These display the following recursion:
The displacement along the arrow as a fraction of total stack length of the s(th) sheet, when drawing n stack sheets overall, for odd n is:

$$\frac{\pm s}{n-1}$$

Where we require the fractional displacement to not exceed half of the total length therefore:
$1 < s < \frac{1}{2}(n-1)$

For the **even number of stack sheets** this pattern emerges:

2 sheets: +-1/2
4 sheets: +-1/2 and +-1/6
6 sheets: +-1/2, +-3/10 and +-1/10 ----------> Note again, 1/2 = 5/10, continue with those not reduced fractions:
8 sheets: +-1/14, +-3/14, +-5/14 and +-7/14 (1/2)
10 sheets: +-1/18, +-3/18, +-5/18, +-7/18, +-9/18
Etc.

These display the following recursion:

The displacement along the arrow as a fraction of total stack length of the s(th) sheet, when drawing n stack sheets overall, for odd n is:

$$\pm\frac{2s+1}{2(n-1)}$$

Where we require the fractional displacement to not exceed half of the total length therefore:

$$1 < s < \frac{1}{2}(n-2)$$

Alternatively, each can be defined separately, manually, if a small amount of sheets is needed.

Code

This has been implemented in code. The parameters for this are as follows:

L is the length of axis in each x and y, from origin
pt_den is the number of points along each axis.
a is a linear scaling of the field
u is the x component -----------------------> Alternatively:
v is the y component ------------------------> Fr is the radial component and Ftheta is the theta component
orientation is a sting that defined how arrows pivot
scale is a linear scale on the quiver plot arrows
delta is the extra length along the axis to show, past the defined grid to show full emerging arrows from border points
fract is the fraction of graph length equal to stack sheet in direction perp. to arrow
s_max is the maximum number of stacks to use, note - this rescales the magnitudes
sheet_L is the length of stack perp. to arrow
s_L is the maximum length of stack sheet parallel to arrow