The 2D Vector Line Equation: **p • o =** ℓ

Dataflow Geometry 2D

Most Mathematics educators consider 2D line equations a settled matter. Why should this topic be revisited as fertile ground for innovation?

In a nutshell, working with 2D lines *in a human-computer partnership* was never factored in -- all the line equations in the textbook are holdovers from the pre-computational era. Not one of the existing forms optimally meets *both* human and computational needs.

These are the criteria deemed important for 21st century problem-solvers working with 2D lines:

- the *numerical features* of unique line L be easy to comprehend, visualize, sketch and display
- all possible 2D lines are representable in this form (representational completeness)
- 1:1 representation -- a unique 2D line has only 1 numerical representation, and visa versa
- two nearly identical 2D lines have nearly identical numerics (continuity)
- a broad array of problems involving 2D lines are solveable with reasonable effort
- the computations run fast, dependably and as exception-free as possible
- theory utilizes vector math concepts that will scale up in 3D to represent planes and 3D lines

Let's take a quick look at how we build up to, and introduce 2D line theory:

The instructor suggests this line of inquiry: **What is invariant numerically about all the points on a 2D line?** What do all those points share in common mathematically? **y = mx + b** is respectfully passed over because it cannot handle vertical lines. Standard form **ax + by = c** satisfies representational completeness, but how do you sketch **a**, **b** and **c**? Can we morph the standard equation into something with meaningful *geometric features*? Yes, in the following way (Fig. 1):

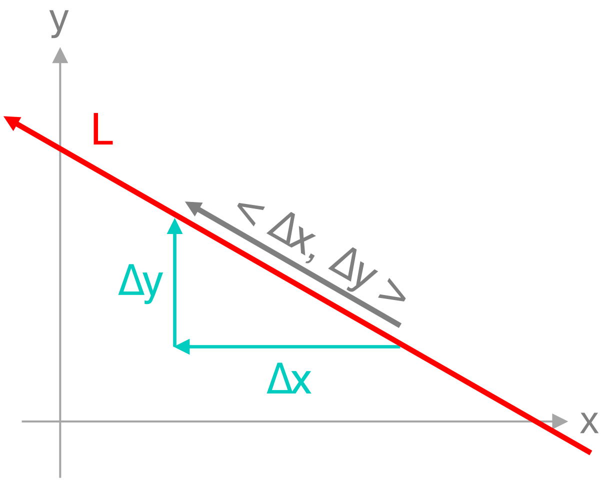
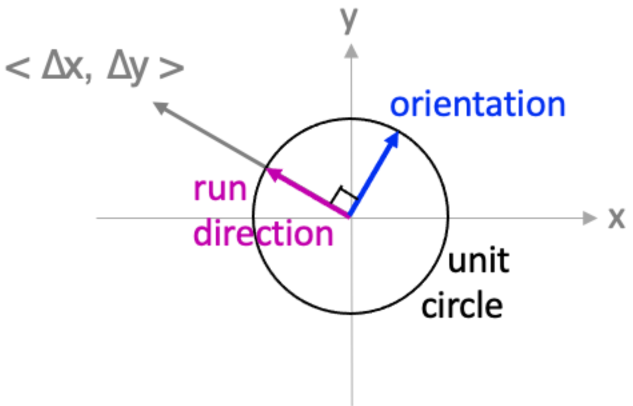


Fig 1. The information about line L's tilt is vectorized (instead of divided into slope m)

Applying vector normalization to <Δx, Δy>, students compute the *run direction* of the line expressed as a direction vector. Students are already comfortable representing 2D directions this way. Next, rotating the run direction 90° clockwise, the line's *orientation* is obtained (Fig 2):

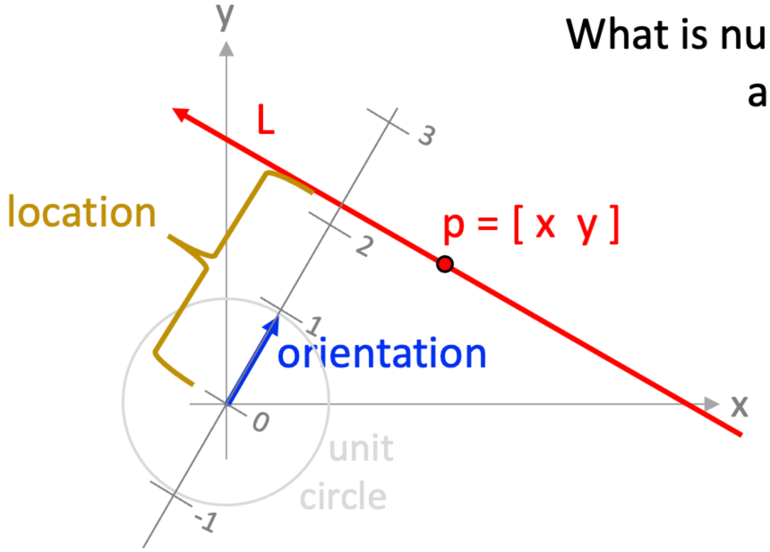


$$\text{run direction} = rd = \frac{< \Delta x,\ \Delta y >}{\sqrt{\Delta x^2 + \Delta y^2}}$$

$$\text{orientation} = < rd_y,\ -rd_x >$$

Fig. 2 The orientation of line L points 90° clockwise from its run direction (both unit vectors)

Next, consider orientation as defining a *rotated axis*. Projective geometry tells us that every point **p** on line L projects onto the *same coordinate* on the gray axis. We call this invariant feature of the line its *location*. The vector dot product **p • orientation** elegantly computes these *point projections* (Fig. 3)



What is numerically invariant about
all points **p** on line L?

p • orientation = location

p • o = ℓ

vector line equation

Fig. 3. Vector line equation. All points on line L *project* onto an invariant coordinate on the gray axis, the *signed distance* of L from the origin. This *location* feature is always coupled with the line's *orientation* feature. Location will be negative if the line locates on the negative half of the gray axis, positive if it falls on the positive half (as is the case shown).

Now, let's relate this *vector line equation* back to the *standard equation* form (Fig. 4):

$$\mathbf{p} \bullet \mathbf{o} = \ell \qquad \textit{vector line equation}$$

$$x\, o_x + y\, o_y = \ell \qquad \text{(expanded dot product)}$$

$$o_x\, x + o_y\, y = \ell \qquad \text{(standard form imbued with geometric meaning)}$$

Fig 4. The vector line equation $\mathbf{p} \bullet \mathbf{o} = \ell$ is a *special case* of the standard form **ax + by = c** where the line-specifying constants **a b c** take on *geometric meanings* $\mathbf{o}_x$ $\mathbf{o}_y$ ℓ that can be visualized (see Fig. 5).

p being the point variable, the numerical feature pair able to pin down a unique line is: L = [**o** ℓ].

Moreover, by quantifying a line's tilt using *perpendicular direction* **o,** we are paving the way for representing the tilt of a plane when we get to 3D. The projective geometry using the dot product works the exact same way in 3D. Can you guess what will be invariant numerically about all the points on a plane? Can you guess what the vector plane equation will look like?

We leave this topic by showing computed numerics for example line L (Fig. 5). Don't worry about the extravagant precision -- students delegate all the number-crunching to the computer using symbolic variables. Most often, the numerics flow invisibly through the dataflow pipelines.

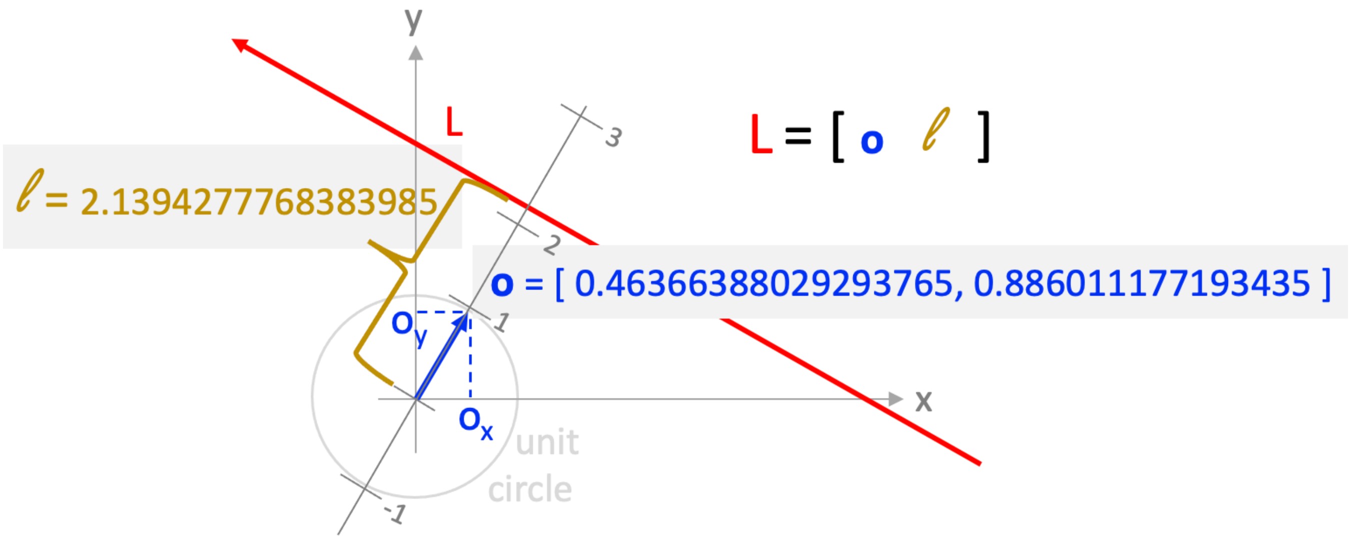


Fig 5. Numeric features **o** and ℓ representing line L and their spatial meanings

Are all the criteria listed at the outset met for this new form of the 2D line equation, $\mathbf{p} \bullet \mathbf{o} = \ell$?

Yes. One extra feature we pick up is adorning 2D lines with *run directions* (red arrow). This will become useful in many advanced applications in 2D and 3D.