

ODE2D_Quiver_Neuron.m

```
function ODE2D_Quiver_Neuron(low1,step1,high1,low2,step2,high2,w11,w12,w21,w22)
% function ODE2D_Quiver_Neuron(low1,step1,high1,low2,step2,high2,w11,w12,w21,w22)
%
% produce a direction field for the system
%
%  $dv_1/dt = -5*v_1(t) + w_{11}*atan(v_1(t)) + w_{12}*atan(v_2(t))$ 
%  $dv_2/dt = -1*v_2(t) + w_{21}*atan(v_1(t)) + w_{22}*atan(v_2(t))$ 
%
% on the rectangle [low1,high1] x [low2,high2] with grid spacing "step"

% Define the grid points:

[v1,v2] = meshgrid([low1:step1:high1],[low2:step2:high2]);

% Compute the derivatives at each grid point:

dv1 = -5*v1 + w11*atan(v1) + w12*atan(v2);
dv2 = -1*v2 + w21*atan(v1) + w22*atan(v2);

% Put the (dv1,dv2) arrow at each grid point (v1,v2)
% and scale (time units) to make the arrows pretty:

quiver(v1,v2,dv1,dv2);

return
```

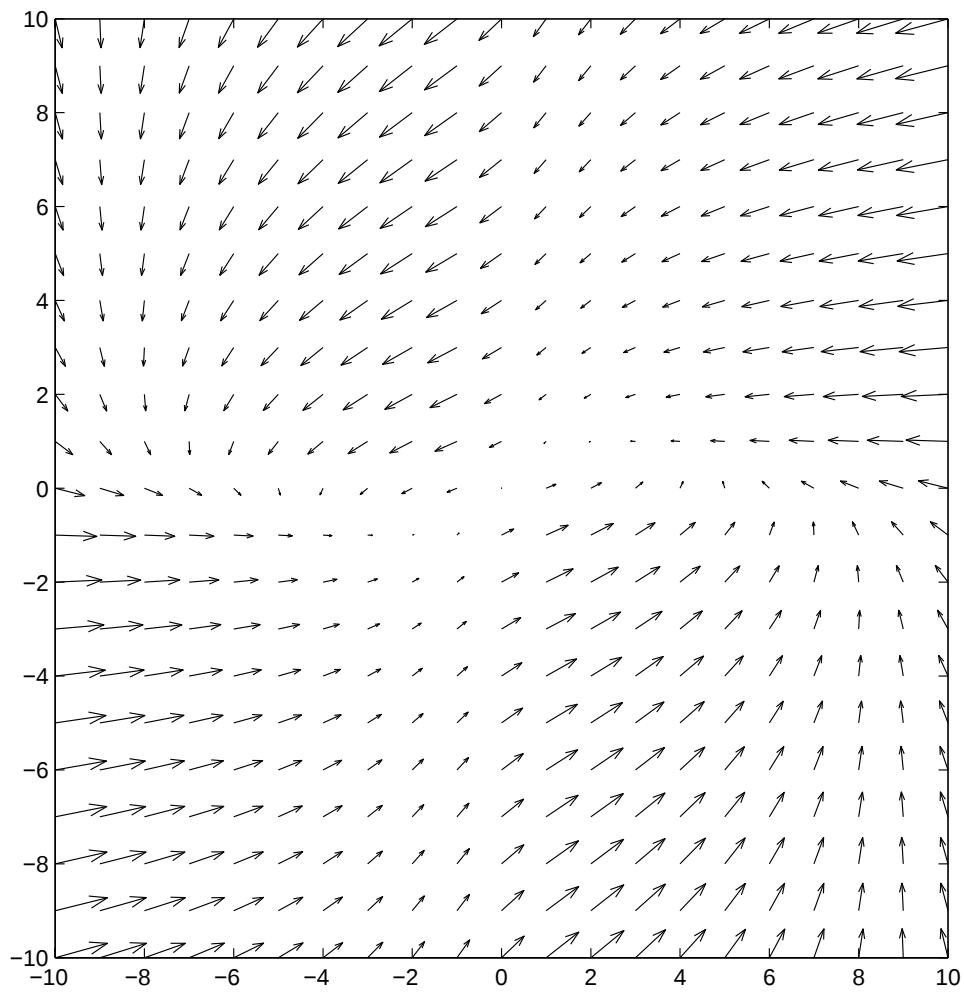


Figure 1: **Direction field** for 2 neurons using `ODE2D_Quiver_Neuron(-10,1,10,-10,1,10,17,-13,4,-5)`.

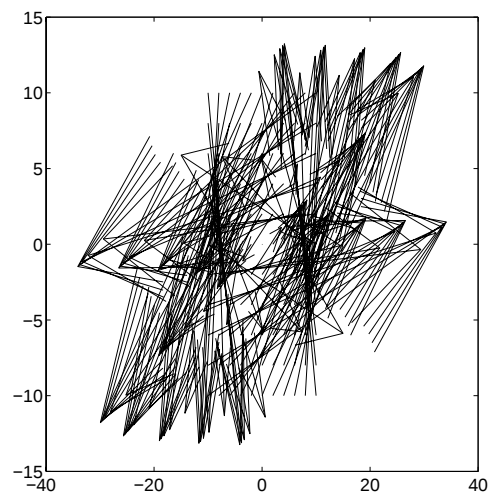


Figure 2: Just for fun... this is what would happen if `quiver` did not scale the arrows!