

```

function traj2(arg1,arg2,arg3,arg4,arg5,arg6)
%TRAJ2    Trajectory plot.  Modified version of QUIVER version
4.1
%    TRAJ2(X,Y,DX,DY) draws little headless arrows at every
(X,Y)
%    pair in matrices X and Y.  The (DX,DY) pairs in matrices DX
%    and DY determine the direction and magnitude of the arrows.
%    In order for this function to work properly, DX and DY
%    MUST be equal to the cosine and sine, respectively, of
the
%    orientation of the headless arrows.
%
%    TRAJ2(x,y,DX,DY), with two vector arguments replacing the
%    first two matrix arguments, must have length(x) = n and
%    length(y) = m where [m,n] = size(DX) = size(DY).  In this
%    case, the arrows are the quads (x(j), y(i), DX(i,j), %
DY(i,j)).
%    Note that x corresponds to the columns of DX and DY and y
%    corresponds to the rows.
%
%    TRAJ2(DX,DY) uses x = 1:n and y = 1:m.  In this case DX and
DY
%    are defined over a geometrically rectangular grid.
%
%    TRAJ2(X,Y,DX,DY,S) and TRAJ2(DX,DY,S) apply scalar S as a
%    scale factor to the lengths of the arrow.  For example, S =
2 % doubles their relative length and S = 0.5 halves them.
%
%    A final trailing string argument specifies linetype and
color
%    using any legal line specification as described under the
%    plot command.
%
%    For example, try
%    xord = -2:.2:2;
%    yord = -2:.2:2;
%    [x,y] = meshgrid(xord,yord);
%    z = x .* exp(-x.^2 - y.^2);
%    [px,py] = gradient(z,.2,.2);
%    contour(x,y,z),hold on, quiver(x,y,px,py), hold off
%
%    See also GRADIENT, COMPASS, FEATHER, ROSE.
%
%    Charles R. Denham, MathWorks 3-20-89
%    Modified 12-19-91, LS.
%    Modified by SJM 10/22/93
%    Copyright (c) 1984-93 by The MathWorks, Inc.

```

```

xx = [0 1 NaN NaN NaN].';
yy = [0 0 NaN NaN NaN].';
arrow = xx + yy.*sqrt(-1);

eval(['last = arg' int2str(nargin) ';'']);
if isstr(last)
    eval(['style = arg' int2str(nargin), ';'']);
    narg = nargin - 1;
else
    style = 'r-';
    narg = nargin;
    eval(['last = arg' int2str(narg-1) ';'']);
    if isstr(last)
        error('Only the final argument can be a string.');
```

```

    end
end
if narg == 0
    error('First 2 or 4 arguments must be numeric.')
```

```

end
eval(['lastdim = min(size(arg' int2str(narg) '));'']);
if lastdim == 1
    if isstr(eval(['arg' int2str(narg)]))
        error('Scalar scale argument expected.')
```

```

    end
    eval(['scale = arg' int2str(narg) ';'']);
    narg = narg - 1;
else
    scale = 1;
end
if narg == 0
    error('First 2 or 4 arguments must be numeric.')
```

```

end
if min(size(arg1)) > 1
    [m,n] = size(arg1);
else
    m = max(size(arg1));
    n = max(size(arg2));
end
if narg == 2
    if isstr(arg1) | isstr(arg2)
        error('Input must be numeric.')
```

```

    end
    [xx,yy] = meshgrid(1:n, 1:m);
    px = arg1;
    py = arg2;
else
    if isstr(arg1) | isstr(arg2) | isstr(arg3) | isstr(arg4)
        error('Input must be numeric.')
```

```

end
if min(size(arg1)) == 1 & min(size(arg2)) == 1
    [xx,yy] = meshgrid(arg1,arg2);
else
    xx = arg1;
    yy = arg2;
end
px = arg3;
py = arg4;
end

% figure out delx and dely so spacing is accounted for in z
delx = xx(1,2)-xx(1,1);
dely = yy(2,1)-yy(1,1);

grid = xx + yy.*sqrt(-1); grid = grid(:);
px = px(:); py = py(:);
maxlen = max(sqrt((px/delx).^2+(py/dely).^2));
z = (px + py.*sqrt(-1)).';
% The next active line was modified 10/22/93 by SJM
% The scaling factor of 0.90 was reset to 0.40
scale = scale*0.40 ./ maxlen;
% The next line is a mathworks suggestion
%a = scale * arrowU * z + ones(5,1) * grid.U;
a = scale * arrow * z + ones(5,1) * grid.';
% append nan's so we get one handle
a = [a; nan*ones(1,size(a,2))];
a = a(:);
cax = newplot;
plot(real(a), imag(a), style);
next = lower(get(cax,'NextPlot'));
if ~ishold
    minx = min(min(xx));
    miny = min(min(yy));
    maxx = max(max(xx));
    maxy = max(max(yy));
    axis([minx maxx miny maxy]);
    view(0,90);
end

```