```
function hh = quiveralt(varargin)
%QUIVER Quiver plot.
%   QUIVER(X,Y,U,V) plots velocity vectors as arrows with components (u,v)
%   at the points (x,y).  The matrices X,Y,U,V must all be the same size
%   and contain corresponding position and vecocity components (X and Y
%   can also be vectors to specify a uniform grid).  QUIVER automatically
%   scales the arrows to fit within the grid.
%
%   QUIVER(U,V) plots velocity vectors at equally spaced points in
%   the x-y plane.
%
%   QUIVER(U,V,S) or QUIVER(X,Y,U,V,S) automatically scales the
%   arrows to fit within the grid and then stretches them by S.  Use
%   S=0 to plot the arrows without the automatic scaling.
%
%   QUIVER(...,LINESPEC) uses the plot linestyle specified for
%   the velocity vectors.  Any marker in LINESPEC is drawn at the base
%   instead of an arrow on the tip.  Use a marker of '.' to specify
%   no marker at all.  See PLOT for other possibilities.
%
%   QUIVER(...,'filled') fills any markers specified.
%
%   H = QUIVER(...) returns a vector of line handles.
%
%   Example:
%      [x,y] = meshgrid(-2:.2:2,-1:.15:1);
%      z = x .* exp(-x.^2 - y.^2); [px,py] = gradient(z,.2,.15);
%      contour(x,y,z), hold on
%      quiver(x,y,px,py), hold off, axis image
%
%   See also FEATHER, QUIVER3, PLOT.

%   Clay M. Thompson 3-3-94
%   Copyright (c) 1984-98 by The MathWorks, Inc.
%   $Revision: 5.15 $  $Date: 1997/11/21 23:46:38 $

% Arrow head parameters
alpha = 0.10; % Size of arrow head relative to the length of the vector
beta = 0.10;  % Width of the base of the arrow head relative to the length
%autoscale = 1; % Autoscale if ~= 0 then scale by this.
autoscale = 0;
plotarrows = 1; % Plot arrows
sym = '';

filled = 0;
ls = '-';
ms = '';
col = '';

nin = nargin;
% Parse the string inputs
while isstr(varargin{nin}),
  vv = varargin{nin};
  if ~isempty(vv) & strcmp(lower(vv(1)),'f')
    filled = 1;
    nin = nin-1;
  else
    [l,c,m,msg] = colstyle(vv);
    if ~isempty(msg),
      error(sprintf('Unknown option "%s".',vv));
    end
```

```
    if ~isempty(l), ls = l; end
    if ~isempty(c), col = c; end
    if ~isempty(m), ms = m; plotarrows = 0; end
    if isequal(m,'.'), ms = ''; end % Don't plot '.'
    nin = nin-1;
  end
end

error(nargchk(2,5,nin));

% Check numeric input arguments
if nin<4, % quiver(u,v) or quiver(u,v,s)
  [msg,x,y,u,v] = xyzchk(varargin{1:2});
else
  [msg,x,y,u,v] = xyzchk(varargin{1:4});
end
if ~isempty(msg), error(msg); end

if nin==3 | nin==5, % quiver(u,v,s) or quiver(x,y,u,v,s)
  autoscale = varargin{nin};
end

% Scalar expand u,v
if prod(size(u))==1, u = u(ones(size(x))); end
if prod(size(v))==1, v = v(ones(size(u))); end

if autoscale,
  % Base autoscale value on average spacing in the x and y
  % directions.  Estimate number of points in each direction as
  % either the size of the input arrays or the effective square
  % spacing if x and y are vectors.
  if min(size(x))==1, n=sqrt(prod(size(x))); m=n; else [m,n]=size(x); end
  delx = diff([min(x(:)) max(x(:))])/n;
  dely = diff([min(y(:)) max(y(:))])/m;
  len = sqrt((u.^2 + v.^2)/(delx.^2 + dely.^2));
  autoscale = autoscale*0.9 / max(len(:));
  u = u*autoscale; v = v*autoscale;
end

ax = newplot;
next = lower(get(ax,'NextPlot'));
hold_state = ishold;

% Make velocity vectors
x = x(:).'; y = y(:).';
u = u(:).'; v = v(:).';
uu = [x;x+u;repmat(NaN,size(u))];
vv = [y;y+v;repmat(NaN,size(u))];

h1 = plot(uu(:),vv(:),[col ls]);

if plotarrows,
  % Make arrow heads and plot them
  hu = [x+u-alpha*(u+beta*(v+eps));x+u; ...
        x+u-alpha*(u-beta*(v+eps));repmat(NaN,size(u))];
  hv = [y+v-alpha*(v-beta*(u+eps));y+v; ...
        y+v-alpha*(v+beta*(u+eps));repmat(NaN,size(v))];
  hold on
  h2 = plot(hu(:),hv(:),[col ls]);
else
  h2 = [];
end
```

```
if ~isempty(ms), % Plot marker on base
  hu = x; hv = y;
  hold on
  h3 = plot(hu(:),hv(:),[col ms]);
  if filled, set(h3,'markerfacecolor',get(h1,'color')); end
else
  h3 = [];
end

if ~hold_state, hold off, view(2); set(ax,'NextPlot',next); end

if nargout>0, hh = [h1;h2;h3]; end
```