Contents

Solve the vortex-blob equations

```
function[]=Math671_HW3_p4()

N = 200;
delta = 2e-1; % smoothing parameter
dt = 1e-1;

% change for different plots
trun = 16;
nsteps = trun/dt;
t = [0,1,2,4,8,16];
timepoints = t./dt;

% Solve using the Lagrangian particle method on p33 of class notes
x = zeros(N,nsteps+1);
y = zeros(N,nsteps+1);
theta = zeros(N,1);
gamma = zeros(N,1);
step = 1;

% Initialize systems
for j = 1:N
    theta(j) = (-1 + j/(N+1))*pi;
    gamma(j) = (pi/(N+1))*cos(theta(j));
    x(j,step) = cos(theta(j));
    y(j,step) = 0;
end

% Run for specified number of time steps
for step = 2:nsteps+1
    dx = dxdt(x(:,step-1),y(:,step-1),delta,gamma,N).*dt;
    dy = dydt(x(:,step-1),y(:,step-1),delta,gamma,N).*dt;
    x(:,step) = x(:,step-1) + dx;
    y(:,step) = y(:,step-1) + dy;
end

% Plot location of vortex blobs at t = 0,1,2,4,8,16
for i=1:length(timepoints)
    hold on
    plot(x(:,timepoints(i)+1),y(:,timepoints(i)+1),'-o','DisplayName',['t = ' num2str(t(i))])
    legend('-DynamicLegend','Location','southeast')
end
title('4b: Numerical solution of vortex-blob equations')
xlabel('x position')
ylabel('y position')
hold off

end
```

Differential equations for x positions

```
function[dx]=dxdt(x,y,delta,gamma,N)

dx = zeros(N,1);
for i=1:N
    directsum = 0;
    for j=1:N
        directsum = directsum + (-1)*(y(i)-y(j))/((x(i)-x(j))^2+(y(i)-y(j))^2+delta^2)*gamma(j);
    end
    dx(i) = directsum/(2*pi);
end

end
```

Differential equations for y positions

```
function[dy]=dydt(x,y,delta,gamma,N)

dy = zeros(N,1);
for i=1:N
    directsum = 0;
    for j=1:N
        directsum = directsum + (x(i)-x(j))/((x(i)-x(j))^2+(y(i)-y(j))^2+delta^2)*gamma(j);
    end
    dy(i) = directsum/(2*pi);
end

end
```

Published with MATLAB® R2015a