Contents

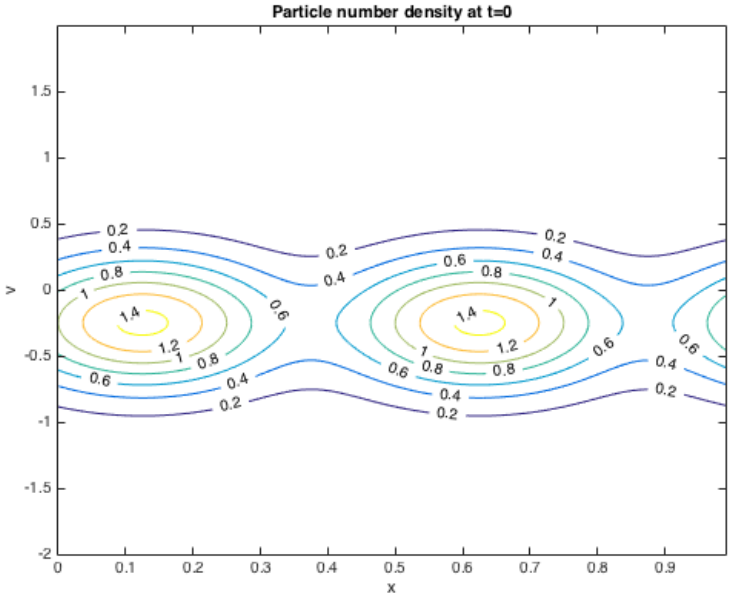
Carry out 1D Vlasov-Poisson simulations with PBCs

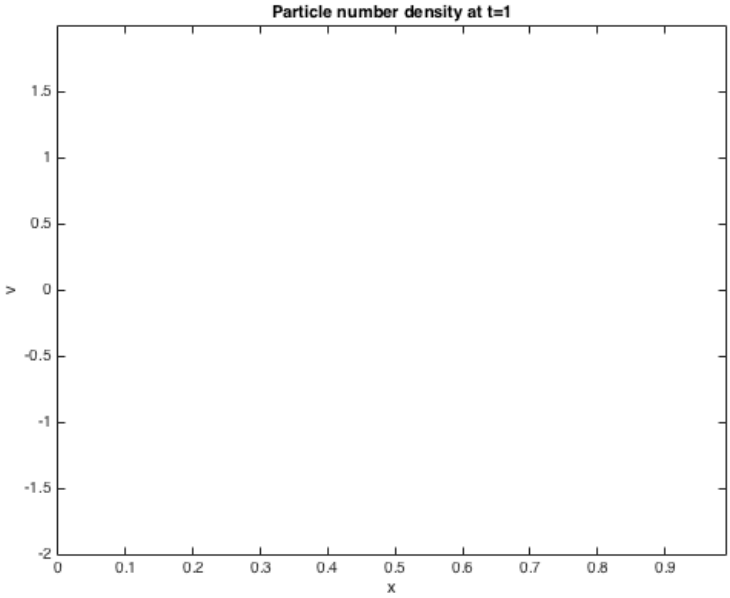
Use the Lax-Friedrichs finite-difference scheme Phase space is x-v, with counter plots iso-temporal lines

```
function[]=Math671_HW3_p3()

% Physical constants
m = 1;
q = -1;

% Simulation settings
dx = 0.01;
dv = 0.01;
vmax = 2;
dt = 0.005;
t = 1;
nsteps = t/dt;
step = 1;

% Setting up mesh grid
vgrid = -vmax:dv:vmax-dv;
xgrid = 0:dx:1-dx;
N = length(vgrid)*length(xgrid);
[X,V] = meshgrid(xgrid,vgrid);

% Calculating f at t=0
for i=1:(1/dx)
    for j=1:2*(vmax/dv)
        f(i,j) = exp(-4*(vgrid(j)+0.25)^2)*(1+0.45*sin(4*pi*xgrid(i)))+exp(-4*(vgrid(i)-0.25)^2)*(1+0.45*sin((4*pi*xgrid(i)-pi)));
    end
end
figure
contour(X,V,f','ShowText','on')
title('Particle number density at t=0')
xlabel('x')
ylabel('v')

% Numerically calculating f at t=1
f_new = transpose(zeros(length(vgrid),length(xgrid)));
for t=2:nsteps
    % Set up density calculations
    rho = q*sum(f,2)*dv;
    rho_avg = sum(rho)/length(rho);
    rho = rho + rho_avg;
    phi = calcPhi(rho,dx,length(rho));
    for i=2:(1/dx)-1
        for j=2:2*(vmax/dv)-1
            Dx = (f(i+1,j)-f(i-1,j))/(2*dx);
            Dv = (f(i,j+1)-f(i,j-1))/(2*dv);
            F = q*(-1)*(Dx/f(i,j))*phi(i);
            f_new(i,j) = -(vgrid(j)*Dx + F*f(i,j)/m*Dv)*dt + 0.25*(f(i+1,j)+f(i-1,j)+f(i,j+1)+f(i,j-1));
        end
    end
    % PBCs for first and last rows/columns
    % PBCs for i
    i=1/dx;
    iplus = 2;
    for j=2:2*(vmax/dv)-1
        Dx = (f(iplus,j)-f(i-1,j))/(2*dx);
        Dv = (f(i,j+1)-f(i,j-1))/(2*dv);
        F = q*(-1)*(Dx/f(i,j))*phi(i);
        f_new(i,j) = -(vgrid(j)*Dx + F*f(i,j)/m*Dv)*dt + 0.25*(f(iplus,j)+f(i-1,j)+f(i,j+1)+f(i,j-1));
    end
    f_new(1,:)=f_new((1/dx),:);
    % PBCs for j
    j=2*(vmax/dv)-1;
    jplus=2;
    for i=2:(1/dx)-1
        Dx = (f(i+1,j)-f(i-1,j))/(2*dx);
        Dv = (f(i,jplus)-f(i,j-1))/(2*dv);
        F = q*(-1)*(Dx/f(i,j))*phi(i);
        f_new(i,j) = -(vgrid(j)*Dx + F*f(i,j)/m*Dv)*dt + 0.25*(f(i+1,j)+f(i-1,j)+f(i,jplus)+f(i,j-1));
```

```
    end
    f_new(:,1)=f_new(:,2*(vmax/dv));
    % PBCS for corners
    iminus = (1/dx)-1;
    jminus = (2*(vmax/dv)-1);
    Dx = (f(2,1)-f(iminus,1))/(2*dx);
    Dv = (f(1,2)-f(1,jminus))/(2*dv);
    F = q*(-1)*(Dx/f(i,j))*phi(i);
    f_new(1,1) = -(vgrid(1)*Dx + F*f(1,1)/m*Dv)*dt + 0.25*(f(2,1)+f(iminus,1)+f(1,2)+f(1,jminus));
    % Set corner values
    f_new(iminus,jminus) = f_new(1,1);
    f_new(1,jminus) = f_new(1,1);
    f_new(iminus,1) = f_new(1,1);
    % Reset f,f_new for next timepoint
    f = f_new;
    f_new = transpose(zeros(length(vgrid),length(xgrid)));
end
figure
contour(X,V,f','ShowText','on')
title('Particle number density at t=1')
xlabel('x')
ylabel('v')

end

% Currently returns phi for a number of particles
% Need to make for grid
function[phi] = calcPhi(rho,dx, N)

% Generate linear algebra
Diag = -ones(N,N);
Diag(logical(eye(size(Diag)))) = 2;

% Calculate phi the lin alg method laid out in lecture notes
phi = inv(Diag)*rho.*(dx)^2;

end

function[D]=calcD(array,d)
D = -(circshift(array,-1)-circshift(array,1))/(2*d).*array;
end
```

```
Warning: Contour not rendered for constant ZData
```





Published with MATLAB® R2015a