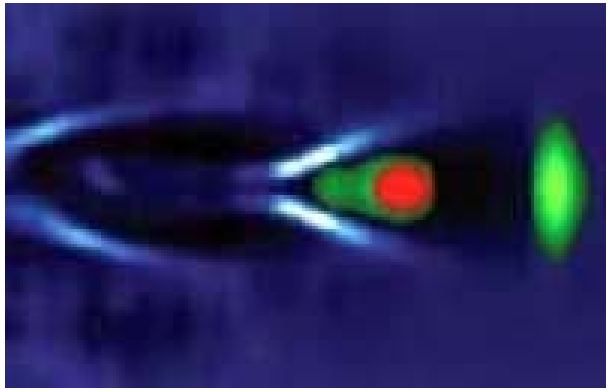
Terascale Accelerator School

10-14 March 2008, DESY Hamburg

Novel Accelerator Concepts - Exercise

S. Khan, University of Hamburg

Make your own 1d plasma code



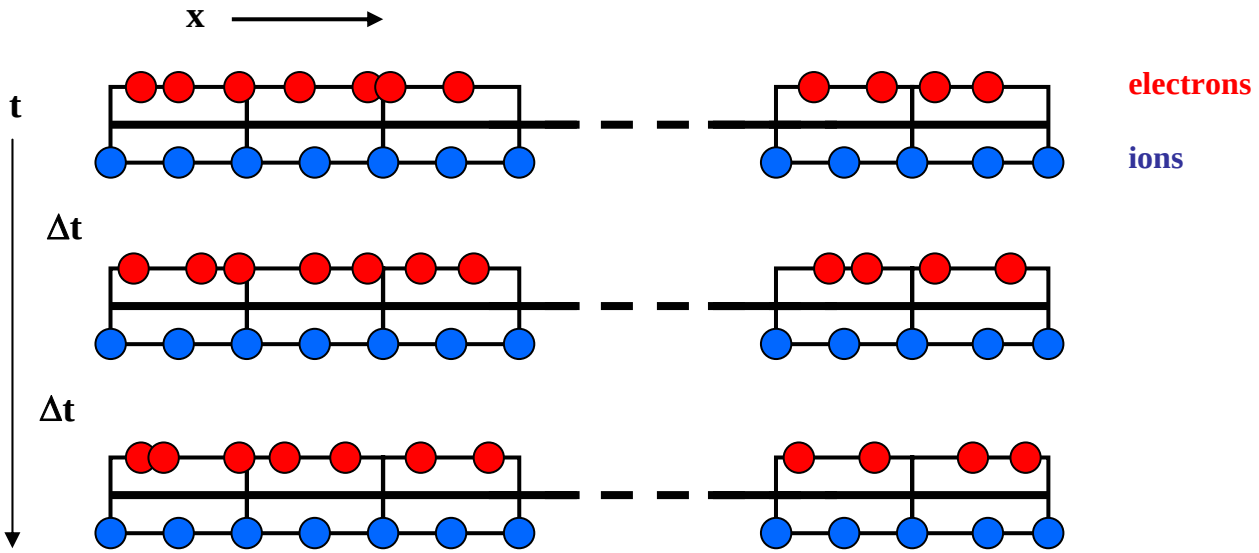
Literature: **W. Kruer, *The Physics of Laser Plasma Interactions***
(Addison Wesley, Frontiers in Physics 1988)

C. Birdsall, A. Langdon, *Plasma Physics via Computer Simulation*
(Taylor & Francis, Series in Plasma Physics 2005)

The model

- circular 1d geometry
- two charged fluids (electrons and ions), zero total charge
- ions are heavy and form a uniform background
- particles (electrons) with continuous positions at discrete time steps
- particle density and electric field evaluated on a discrete grid
- no scattering, no external field, light pulse

} PIC*



typically np=10000 particles
nc=100 cells

*) PIC = particle-in-cell

The physics

$$\frac{dv}{dt} = -\frac{eE}{m}$$
$$\frac{dx}{dt} = v$$
$$\frac{dE}{dx} = -\frac{e}{\varepsilon_0}\left(n - \bar{n}\right)$$

Sir I. Newton

S. D. Poisson

define $t' = \omega_p t$ with $\omega_p = \sqrt{\frac{\bar{n}\, e^2}{\varepsilon_0 m}}$

$$x' = \frac{x}{\delta}$$
$$v' = \frac{v}{\omega_p \delta}$$
$$E' = \frac{\varepsilon_0 E}{e\bar{n}\delta}$$

$$t = \frac{t'}{\omega_p}$$
$$x = x'\delta$$
$$v = v'\omega_p \delta$$
$$E = E'\frac{e\bar{n}\delta}{\varepsilon_0}$$

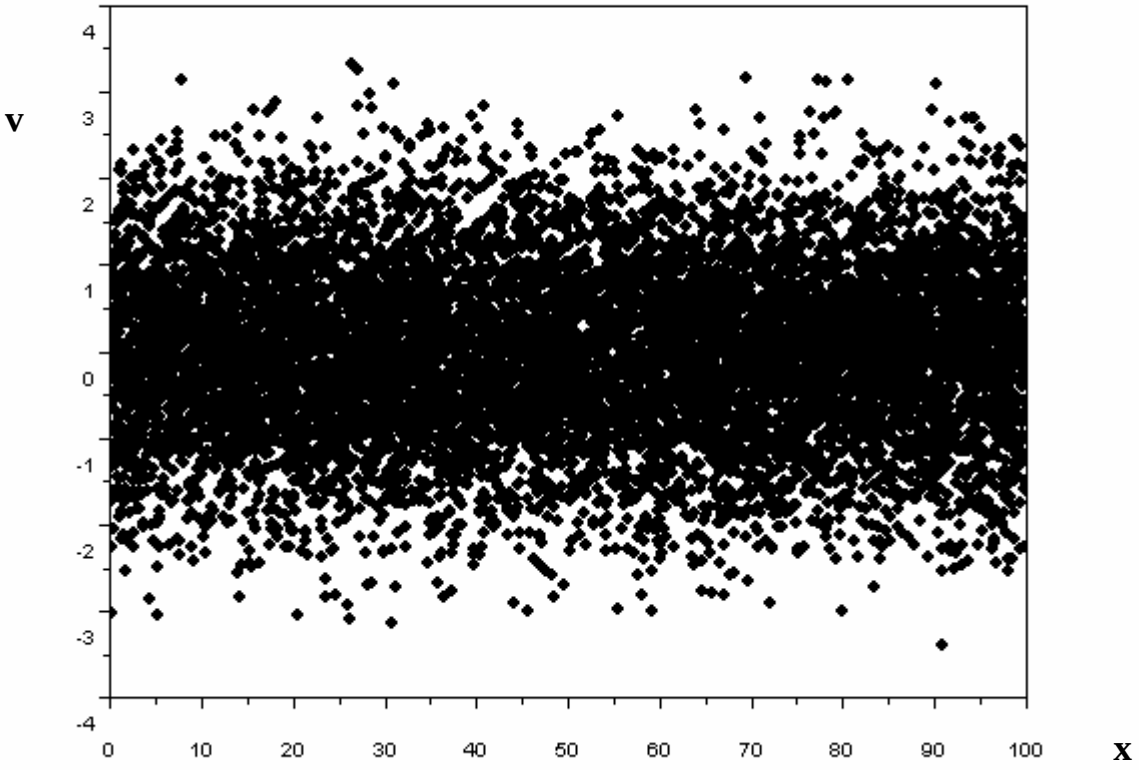
$$\frac{dv'}{dt'} = -E'$$
$$\frac{dx'}{dt'} = v'$$
$$\frac{dE'}{dx'} = -\frac{1}{\bar{n}}\left(n - \bar{n}\right) = -\frac{n\delta}{\bar{n}\delta} + 1$$

$$\bar{n}\delta = \frac{\mathrm{np}}{\mathrm{nc}}$$

First step: initialize the particle distribution

x(i) (i=1,np) even distribution
v(i) (i=1,np) Gaussian distribution*

*) e.g.
```
rand()+rand()+rand()+rand()...
+rand()+rand()+rand()+rand()...
+rand()+rand()+rand()+rand()-6;
```



First step: initialize the particle distribution

```
nc=100
np=10000

norm=nc/np

vamp=0.05

dt=0.05

//  ============================================
//  initialize particle positions and velocities
//  ============================================

for i=1:np
    x(i)=(i-0.5)*norm;
    v(i)=vamp*(rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()-6.);
end

plot(x,v,,'linest','none','marker','.',`markers',5)
```

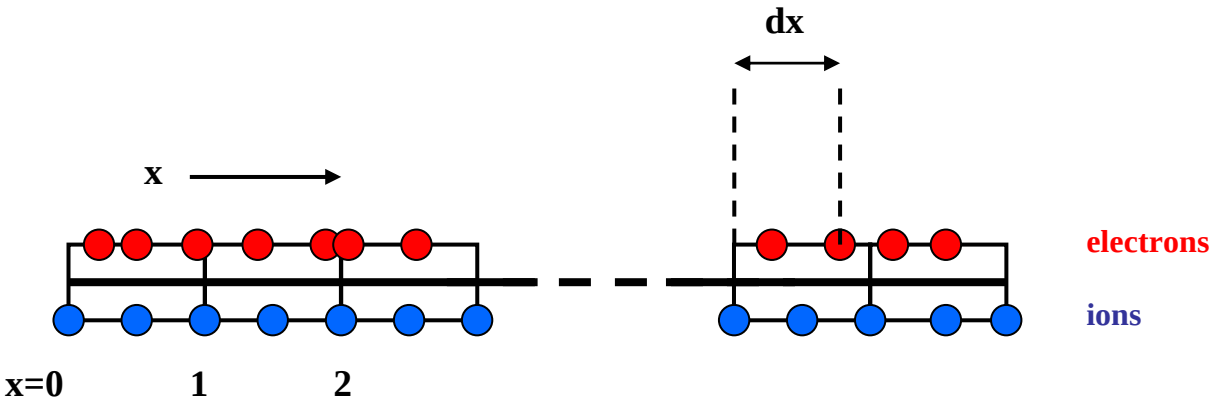
Second step: start the loop over time steps

1, 2, 3, 4, ... nc

```
for k=1:nc
    cell(k)=k;
end

loop=0
while loop<10
    loop=loop+1;
    printf('%d\n',loop)


    ...


end
```

```
...
[More (y or n)?]
1
2
3
4
5
6
7
8
9
...
```

Third step: compute the charge density from the electron positions



k=int(x)
k1=k+1
k2=k+2

dx = x – k
ρ(k1) increases by – (1 – dx)
ρ(k2) increases by – dx

finally ρ(k)=ρ(k)*nc/np+1

normalize to charge 1 per cell

positive ions

Third step: compute the charge density from the electron positions

```
//  ======================
//  compute charge density
//  ======================

    for k=1:nc
        rho(k)=0.0;
    end

    for i=1:np
        k=int(x(i));
        k1=k+1;
        k2=k+2;
        if k2>nc then
           k2=1;
        end

        dx=x(i)-k;
        rho(k1)=rho(k1)-(1.-dx);
        rho(k2)=rho(k2)-dx;
    end

    for k=1:nc
        rho(k)=norm*rho(k)+1.;
    end
```

negative (electrons)

normalize and add ions

Fourth step: compute the electric field from the charge density

Poisson: get E(k+1) from E(k)

$$E(k+1)-E(k)=\frac{1}{2}\{\rho(k+1)+\rho(k)\}$$ and $$E(nc+1)=E(1);\qquad \sum_{k=1}^{nc}E(k)=0$$

for j>1

$$E(j)=E(1)+E(2)-E(1)+E(3)-E(2)\ldots+E(j)-E(j-1)$$

$$=E(1)+\sum_{k=1}^{j-1}\frac{1}{2}\{\rho(k)+\rho(k+1)\}$$

$$E(1)+\sum_{j=2}^{nc}E(j)=0=E(1)+\sum_{j=2}^{nc}\left[E(1)+\sum_{k=1}^{j-1}\frac{1}{2}\{\rho(k)+\rho(k+1)\}\right]$$

$$nc\times E(1)=-\sum_{j=2}^{nc}\sum_{k=1}^{j-1}\frac{1}{2}\{\rho(k)+\rho(k+1)\}$$

$$j'=j-1$$

$$nc\times E(1)=-\sum_{j'=1}^{nc}\sum_{k=1}^{j'}\frac{1}{2}\{\rho(k)+\rho(k+1)\}$$

find E(1) first,
then E(2), E(3) ... E(nc) can be calculated

Fourth step: compute the electric field from the charge density

```
//  ======================
//  compute electric field
//  ======================

    esum=0.;
    for k=1:nc
        for k1=1:k
            k2=k1+1;
            if k2>nc then
               k2=1;
            end

            esum=esum+(rho(k1)+rho(k2))/2.;
        end
    end

    e(1)=-esum/nc;

    for k=2:nc
        e(k)=e(k-1)+(rho(k)+rho(k-1))/2.;
    end
```

Fifth step: move the electrons in their electric field

k = int(x)
k1 = k+1
k2 = k+2

dx = x – k

F = – E(k1) (1 – dx) – E(k2) dx

v = v + F dt
x = x + v dt

Fifth step: move the electrons in their electric field

```
//  ==================
//  move the particles
//  ==================

    for i=1:np
        k=int(x(i));
        k1=k+1;
        k2=k+2;
        if k2>nc then
           k2=1;
        end

        dx=x(i)-k;
        force=-e(k1)*(1.-dx)-e(k2)*dx;

        v(i)=v(i)+force*dt;
        x(i)=x(i)+v(i)*dt;

        while x(i)<0., x(i)=x(i)+nc; end
        while x(i)>nc, x(i)=x(i)-nc; end
    end
```

negative (electrons)

stay within the model

Sixth step: have fun

a) Follow induvidual electron trajectories

b) Two-Stream instability

c) Electron beam

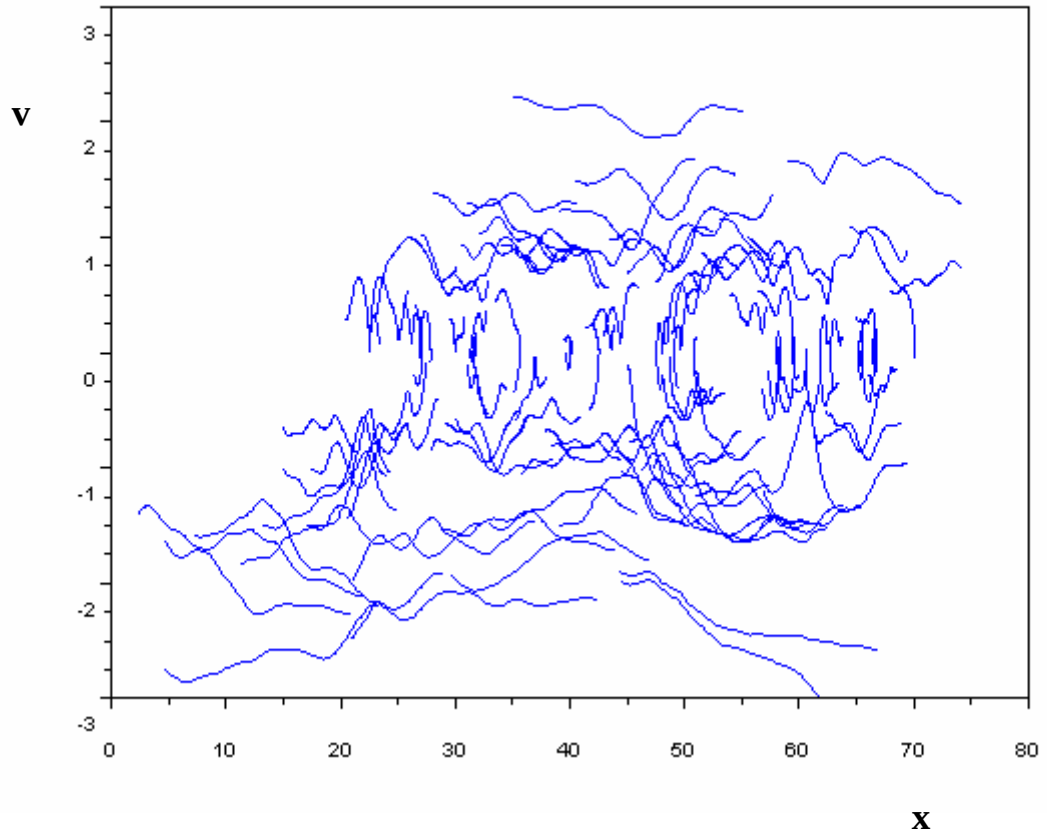
a) Follow individual electron trajectories

```
loop=0
while loop<100
    loop=loop+1;

      ...

    for j=1:100
        i=5*j+200
        xmon(j,loop)=x(i);
        vmon(j,loop)=v(i);
    end
end

for j=1:100
    for l=1:99
        xx(l)=xmon(j,l);
        vv(l)=vmon(j,l);
    end
    plot(xx,vv)
end
```

save position and velocity in these arrays

one plot per particle (there may be more elegant ways to do this)

a) Follow induvidual electron trajectories

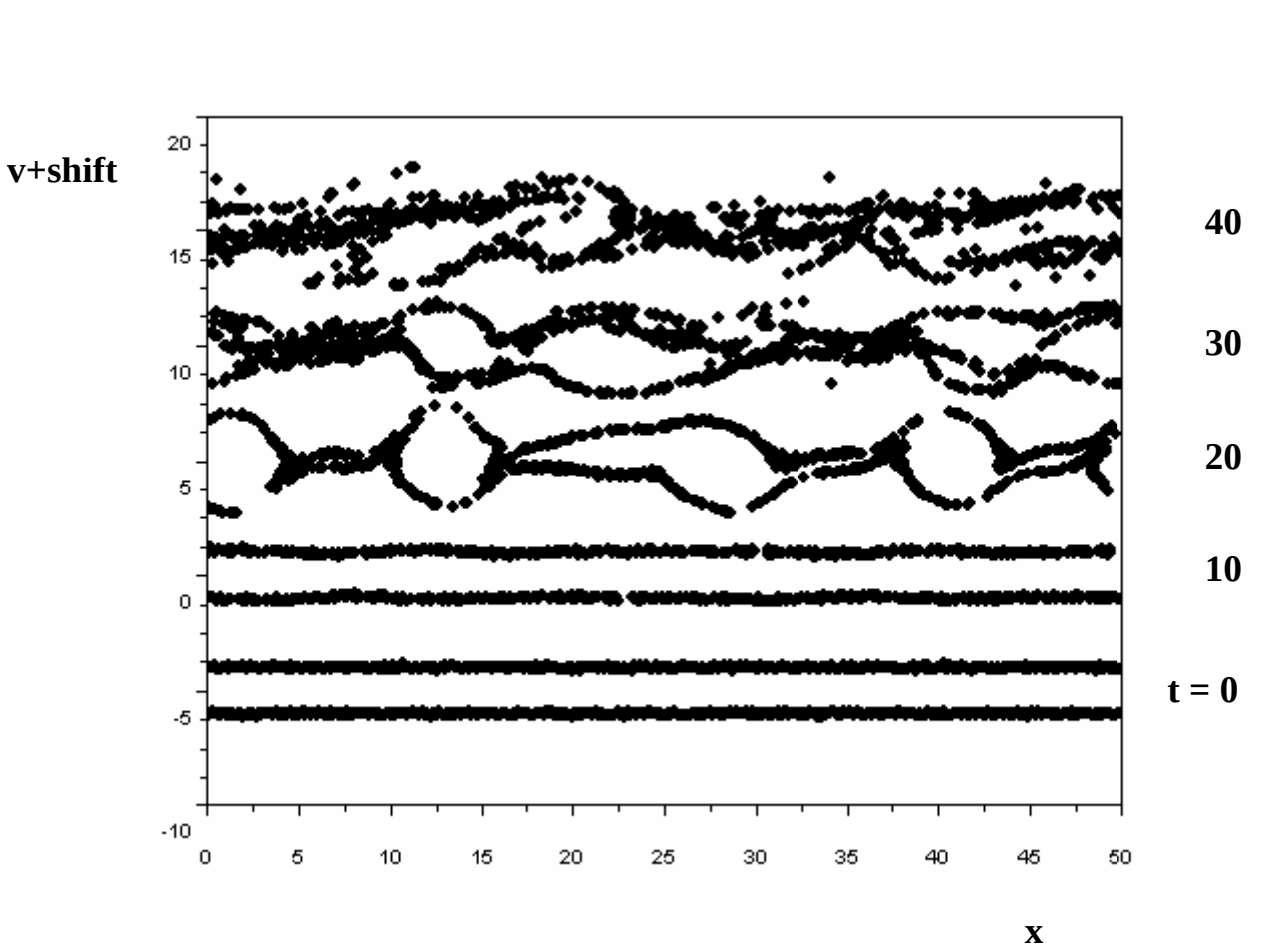


b) Two-stream instability

```
vamp=0.05
voff=1.
vsig=-1.
vshift=0;

dt=0.05

for i=1:np
    voff=voff*vsig;
    x(i)=(i-0.5)*norm;
    v(i)=vamp*(rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()-6.)+voff;
end
```

```
...
if modulo(loop-1,200) == 0 then
   plot(x,v+vshift,,'linest','none','marker','.','Markers',5)
   vshift=vshift+5.;
end
```

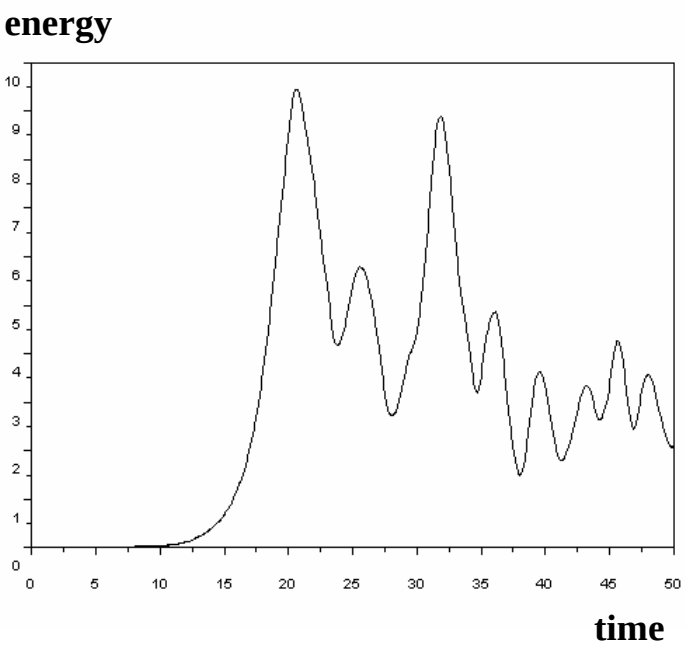
b) Two-stream instability

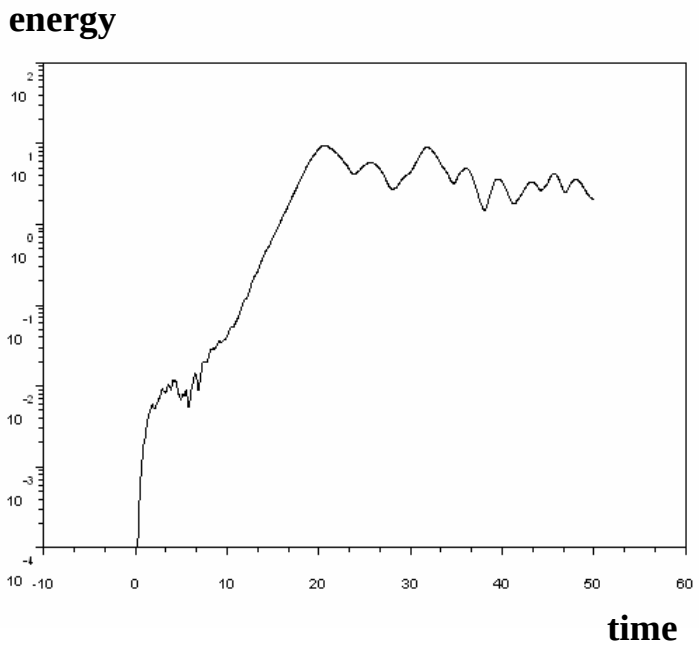


b) Two-stream instability

```
//    if modulo(loop-1,200) == 0 then
//       plot(x,v+vshift,,'linest','none','marker','.','Markers',5)
//       vshift=vshift+5.;
//    end
    esum=0.;
    for k=1:nc
        esum=esum+e(k)*e(k);
    end
    time(loop)=dt*loop;
    energy(loop)=esum;
end

plot(time,energy)

plot2d(time,energy,logflag='nl')
```

b) Two-stream instability





c) Electron beam

```
for i=1:np-3
    voff=voff*vsig;
    x(i)=(i-0.5)*norm;
    v(i)=vamp*(rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()...
              +rand()+rand()+rand()+rand()-6.)+voff;
end

/// =============
/// electron beam
/// =============

bunchpos=100;
for i=np-2:np
    x(i)=bunchpos;
    v(i)=5.;
end
```

c) Electron beam