An Introductory Plasma simulation by Ｐａｒｔｉｃｌｅ-in-cell (PIC) code

プラズマシミュレーション事始め

Πλ_σμα，ατο_，__(_Greek_:________)

－From Debye shelter to Laser Wakefield Acceleration－

1. Simulation gets more easy for science and technology
2. Algorithms for PIC codes
3. Finite difference method for Poisson equation
4. Results of 1D & 2D PIC simulations
5. Towards a simulation for Laser Wakefield Acceleration

Nanotechnology center, ISIR, Osaka University

Yoshida Akira 吉田亮

6 September, 2006 Workshop SAD2006 at KEK

Personal schedule:スケジュール

１． Make a **1 D Electro-static Particle-in-cell** ：

～April,2006.

２． **2 D Electro-static Particle-in-cell**:

~August,2006…

３． Interaction between floating electromagnetic field and
charged particles : **Complete electromagnetic field**

４． **Towards a simulation for Laser Wakefield Acceleration**

５． Mathematical analysis for unstable phenomena by discretized solutions used in **Finite difference**

☆ We want to simulate Plasma Physics by PC to understand Electromagnetism with 3D charts or movies.

∵ Note PC with 2GHz clock & 2GB memory (\ a quarter of a million) is far superior to twenty-years-old super computer (\ three billion)

∵ Note PC with 2GHz clock & 2GB memory (\ a quarter of a million) is far superior to twenty-years-old super computer with 70MHz clock & 256MB memory (\ three billion) *by 1~2 figures ! Cost/performance ~10^5 or 10^6*

●1980's Pipelined Super computer VP series (VP100/200/400) consists of a Scalar processor （Main frame ：M380) and a Vector processor (vector registers and pipelined arithmetic unit) ：

Clock : 15ns = 1.5×10^{-8}sec ; Hz : 1 /clock = $1/1.5 \times 10^{-8} = 0.67 \times 10^{8}$ = 67MHz

Clock of pipelined arithmetic unit was 7.5ns (2 floating operations/15ns)

Two pipelined arithmetic unit calculate simultaneously ： 268 MFLOPS (VP200：Add.+Mult. 134 MFLOPS x2 ; Winter,1983) (**M**ega **Fl**oating **O**perations **P**er **S**econd)

◎Pentium4 (2GHz) : if one floating operation/clock ：~ 2GFLOPS**?** If two floating operation/clock ： 2GFLOPS ×2= ~4GFLOPS ？ (**G**iga **Fl**oating **O**perations **P**er **S**econd)

Xenon dual core processor (2.7GHz) : Linpac record : 1.93GFLOPS (May2006)

Algorithm for Particle-in-cell code : calculate movement of charged particles in a electromagnetic fields (Shigeo kawata川田重夫 Simulation Physics 1, 1990)

Initial conditions setting

↓

Place charged particles in a mesh

↓

(+Δt)

Solve Poisson eq. $\nabla^2 \phi = -\frac{\rho}{\varepsilon}$ to calculate potential & electric field

Laplacian $\nabla^2 = \nabla \cdot \nabla = (\Delta) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$

↓

Calculate the electric field at each particle and calculate the movement for a particle

↓

Output results

$E_p=(S1E_{i+1/2} + S2E_{i+3/2})/(S1 + S2)$, S1 : the distance between particle and (i+3/2),

S2 : the distance between (i+1/2) and particle.

←S2→←---- S1 ---→ @:the position of a particle

|------ - - - ----|-----------|------@----|-----------|-------------------- - - - ---------------|

i=1 i (i+1/2) i+1 (i+3/2) ….. Total mesh number

Algorithms for Kawata’s Particle-in-cell code

初期データ設定：粒子、ポテンシャルetcを初期化する

↓

荷電粒子の配置に基ずきメッシュごとに電荷密度を計算する

↓

ポアソン方程式を中心差分法で解きメッシュ毎の静電ポテンシャルを求める

↓

メッシュ毎の電場を計算する

↓

荷電粒子の運動を計算し新しい位置を求める

↓

メッシュ毎の電荷密度、静電ポテンシャル、電場、全粒子の位置と速度の出力

$t = t + \Delta t$

｛時間の終了条件まで回る｝

Fundamentals of the particle-in-cell plasma simulation method

J. P. Verboncoeur, UCB-NE

This seminar is the first in a series on the fundamentals of the particle-in-cell (PIC) technique of plasma simulation [1-3]. In the PIC method, point particles with continuum phase variables are tracked within a discrete spatial mesh on which electromagnetic fields are defined. We will cover the essence of the PIC method as outlined in Figure 1, including the integration of the equations of motion, fundamental particle boundary conditions, the interpolation of charge and current source terms, ρ and J, to the field mesh, and the interpolation of the fields E and B from the mesh to the continuum particle locations.

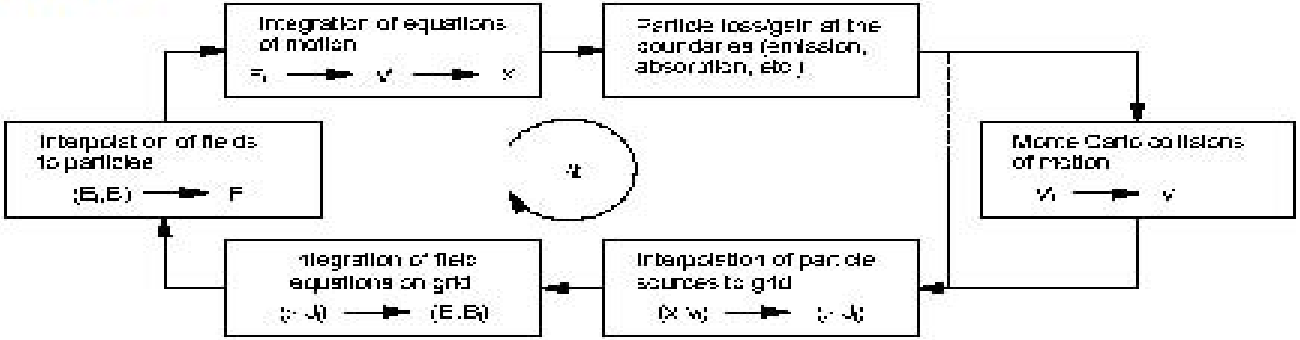


Figure 1 PIC simulation flowchart.

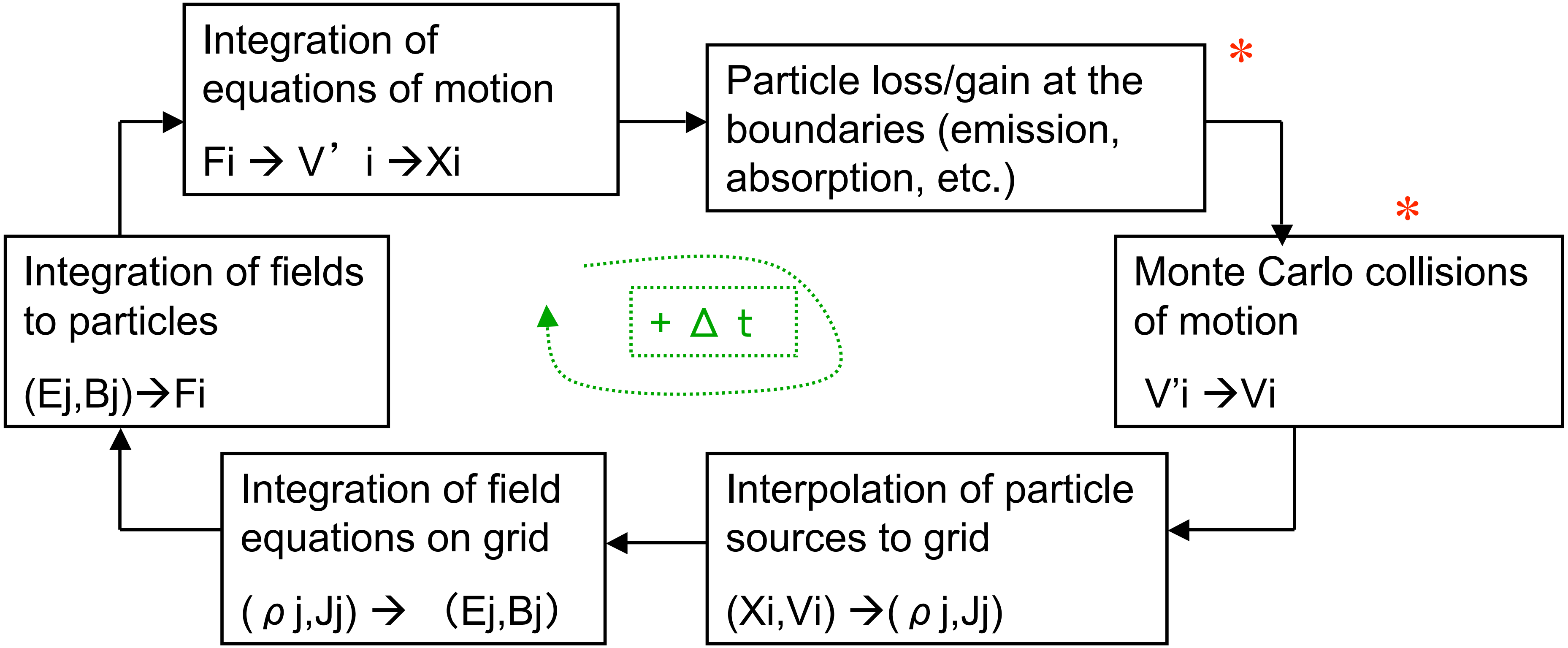
[1] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, IOP Publishing Ltd. ~~(2005)~~ (1991)
[2] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, IOP Publishing Ltd. (1988).
[3] J. P. Verboncoeur, "Particle simulation of plasmas: review and advances", *Plasma Phys. Control. Fusion* 47 (2005) A231-A260.

Formerly KEK, Honorary professor Ogata said

"The P I C simulation is Birdsall's ! " at spring 2006, but

It's FORTRAN77 and a little older…? ➔ C++ is cool.

C.K.Birdsall, "1D Electro Static code : ES1. Algorithm"



C.K.Birdsall and A.B.Langdon, "Plasma Physics via Computer Simulation", 1991

* These are more precise than Kawata's algoritms?

Approximation by finite difference method for Poisson equation

A. 3 dimension Poisson equation：

$$\nabla^2 \phi = -\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}\right) = -\frac{\rho}{\varepsilon} \ldots (1)$$

where、 ϕ is static potential、 ρ is charge density、 ε is dielectric constant

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\rho}{\varepsilon} \ldots (1')$$

B. 1dimension finite element method: FDM

$$\frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{\Delta x^2} = -\frac{\rho_i}{\varepsilon} \ldots (2)$$

: ϕ_{i+1}can be calculated with ϕ_{i-1} and ϕ_i

we solve n（number of mesh）simultaneous linear equations with two boundary conditior

$$\begin{aligned}
&\phi_1 = o \\
&\phi_1 + \phi_3 - 2\phi_2 = \frac{\rho_2}{\varepsilon}\Delta x^2 \\
&\phi_2 + \phi_4 - 2\phi_3 = \frac{\rho_3}{\varepsilon}\Delta x^2 \\
&\ldots \\
&\phi_n + \phi_{n+2} - 2\phi_{n+1} = \frac{\rho_{n+1}}{\varepsilon}\Delta x^2 \\
&\phi_n = V
\end{aligned} \ldots (3)$$

boundary condition 1: $\phi_{i=1}$ =0

boundary condition 2: $\phi_{i=n}$ =V

We solve (3) by Gaussian elimination. (3) can be converted into a triple diagonal matrix, and n lines and 4 lows array is used to solve it to save memory of computer if it's a large simultaneous linear equations.

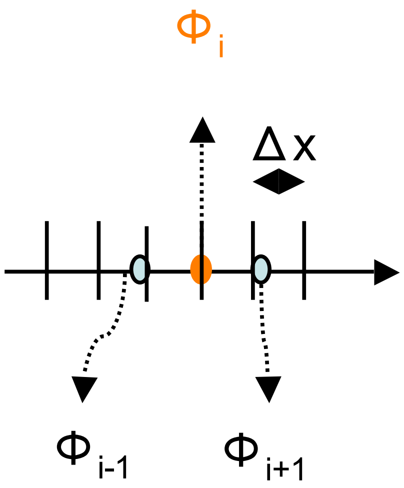
C. 2 dimension finite difference method (FDM)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \frac{\rho}{\varepsilon} \ldots (1'')$$

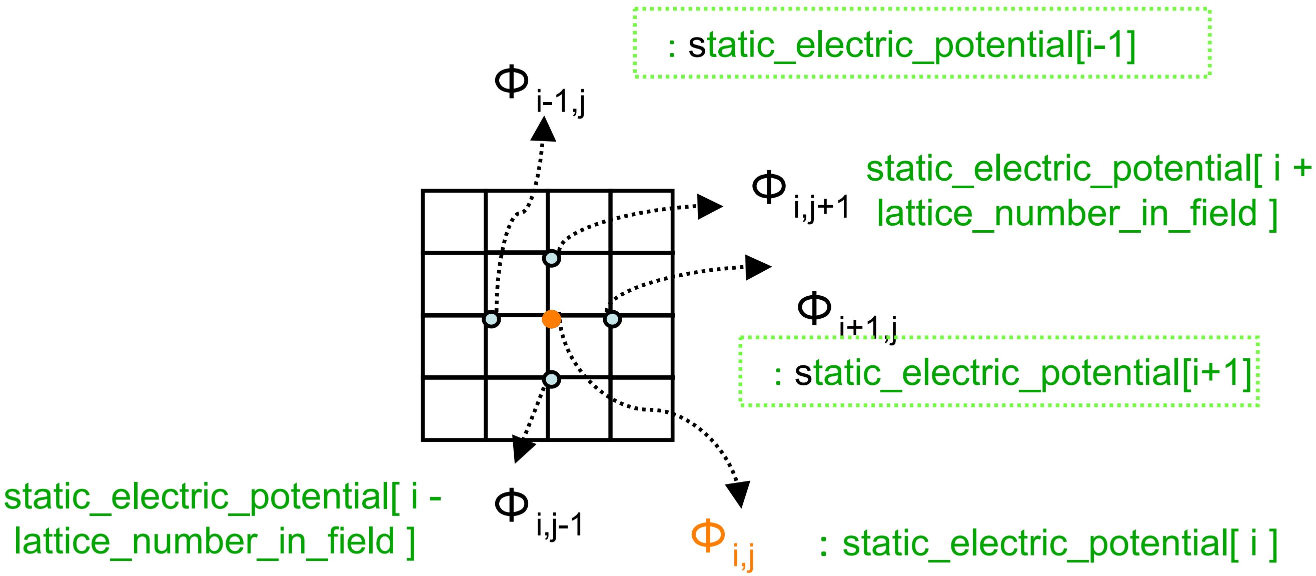
Central finite difference Method of 2D Poisson equation:

$$\frac{\phi_{i+1,j} + \phi_{i,j+1} + \phi_{i-1,j} + \phi_{i,j-1} - 4\phi_{i,j}}{\Delta x^2} = -\frac{\rho_i}{\varepsilon} \ldots (2')$$

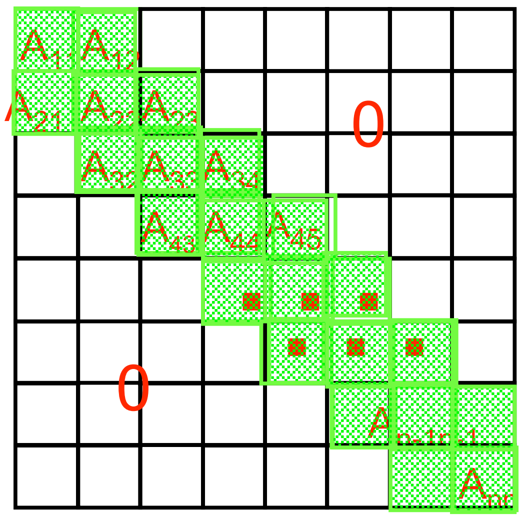
(2') is same as Laplace equation, and it is also solved by Gaussian elimination. Here we use quintuple diagonal matrix and solve n lines and 6 lows array to solve it.



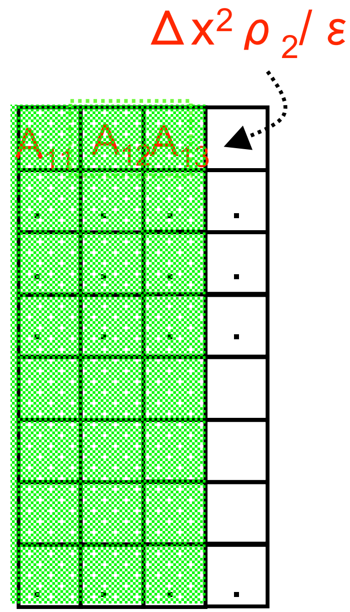
1Dimension (eq.2)



2Dimension (eq.2’)



Block triple diagonal matrix
ブロック3重対角行列の例
（Only 3 lines upper & lower of the diagonal element have nonzero elements）



n-lines,4-colums array is stored to save memory

C++ code for computation of electric field for 2D Poisson equation with Central finite difference method (ポアソン方程式の中心差分による電場の計算部分

```
// calculate electric field :  2D poisson equation for electrostatic field
void electric_field()         // f(x,y)=1/(h*h) * [(Ui+1,j)+(Ui,j+1)+(Ui-1,j)+(Ui,j-1)-4Uij]
                              //   6                  1        2        3         4        5
{                          // finite difference method
   for (int i=1; i<=( lattice_number_x * lattice_number_y -1 ); i++)  {
      static_electric_field[ i ] =
                         ( static_electric_potential[ i + 1 ]     +
                           static_electric_potential[ i - 1 ]     +
                           static_electric_potential[ i + lattice_number_in_field ]   +
                                 static_electric_potential[ i - lattice_number_in_field ]       -
                          4 * static_electric_potential[ i ]  )     /
                                   (mesh_width*mesh_width*mesh_width);  *
                         }
      static_electric_field[ lattice_number_x * lattice_number_y ]= 0.0;
}
```

*static_electric_potential[i] is an 1D array arranged from original 2D mesh array:

static_electric_potential[i +1]+static_electric_potential[i-1]+static_electric_potential[i+lattice_number]+
static_electric_potential[i-lattice_number]- 4static_electric_potential[i]

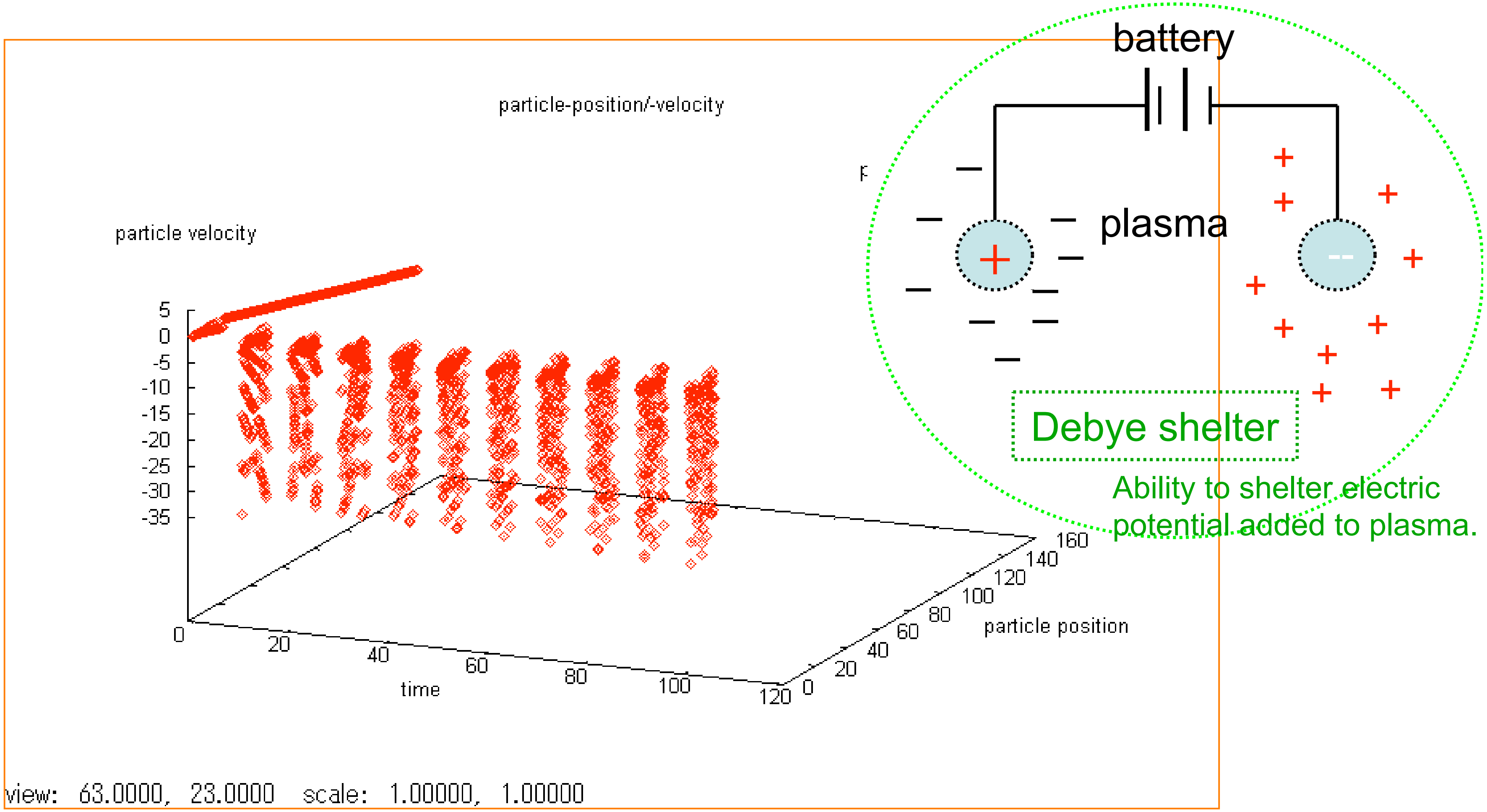


Fig.1: A simulation of (maybe) Debye shelter by 1D Electro-static particle-in-cell code :
Particles around the position 0~140 uniformly at time=0 move to the position 0~20 at time=10~110, and the velocity about 0 at time=0 spread abruptly to -35~0 during time=10~110 in this simulation.

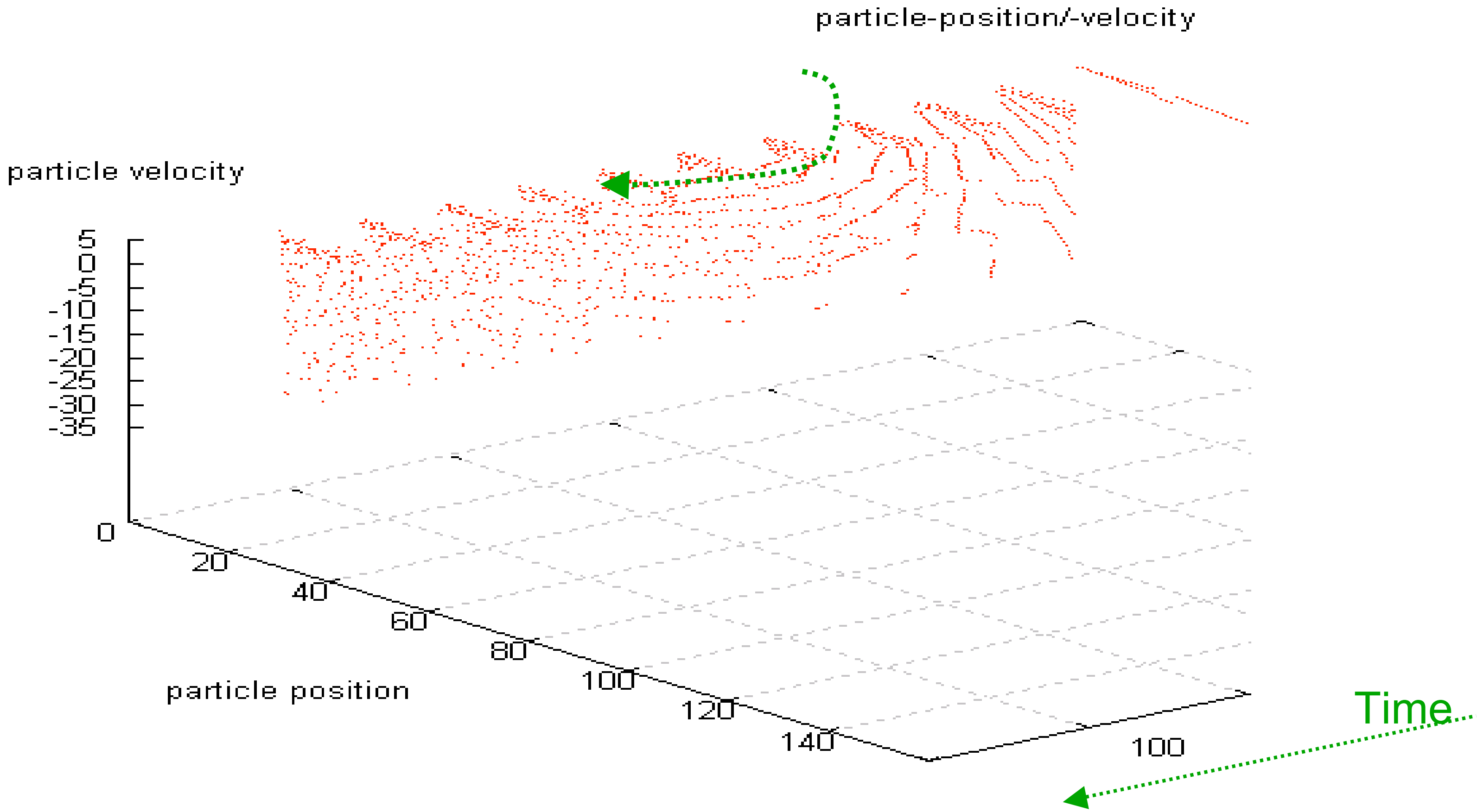


Fig.1’: A simulation of (maybe) Debye shelter by 1D Electro-static particle-in-cell code :
Particles are circling and accelerated against the charge (the initial boundary condition) due to the generated magnetic field by the movement of the charged particles.

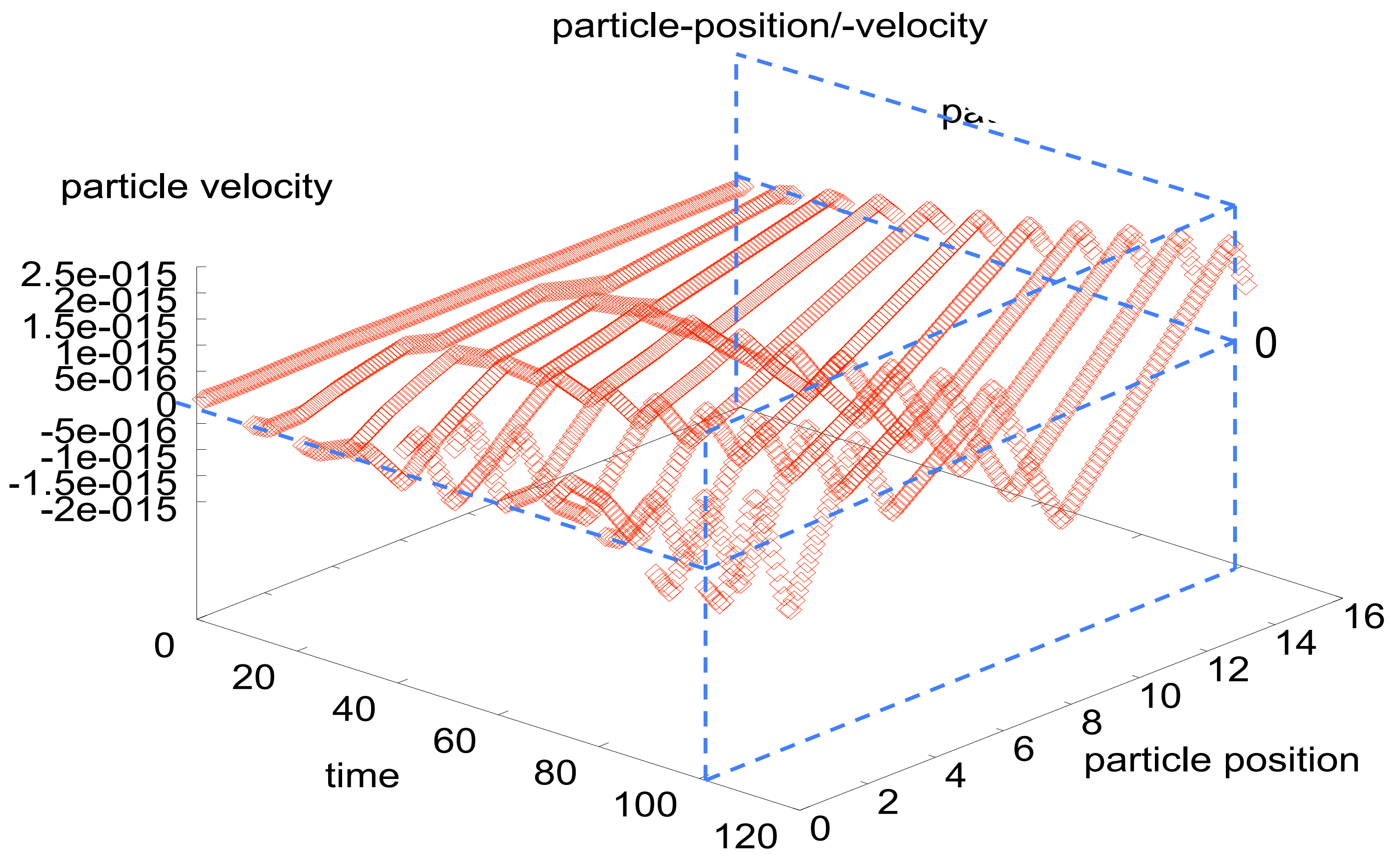


Fig.2 : A simulation of 1D Electro-static particle-in-cell code (it=101) : ?
Particles at the position 0~16 with particle velocity~0 uniformly at time=0 change to particle velocity +2.0~ -1.5. It's irregular wavy shape, and gradually raising the velocity difference as time goes by from 0 to 101.

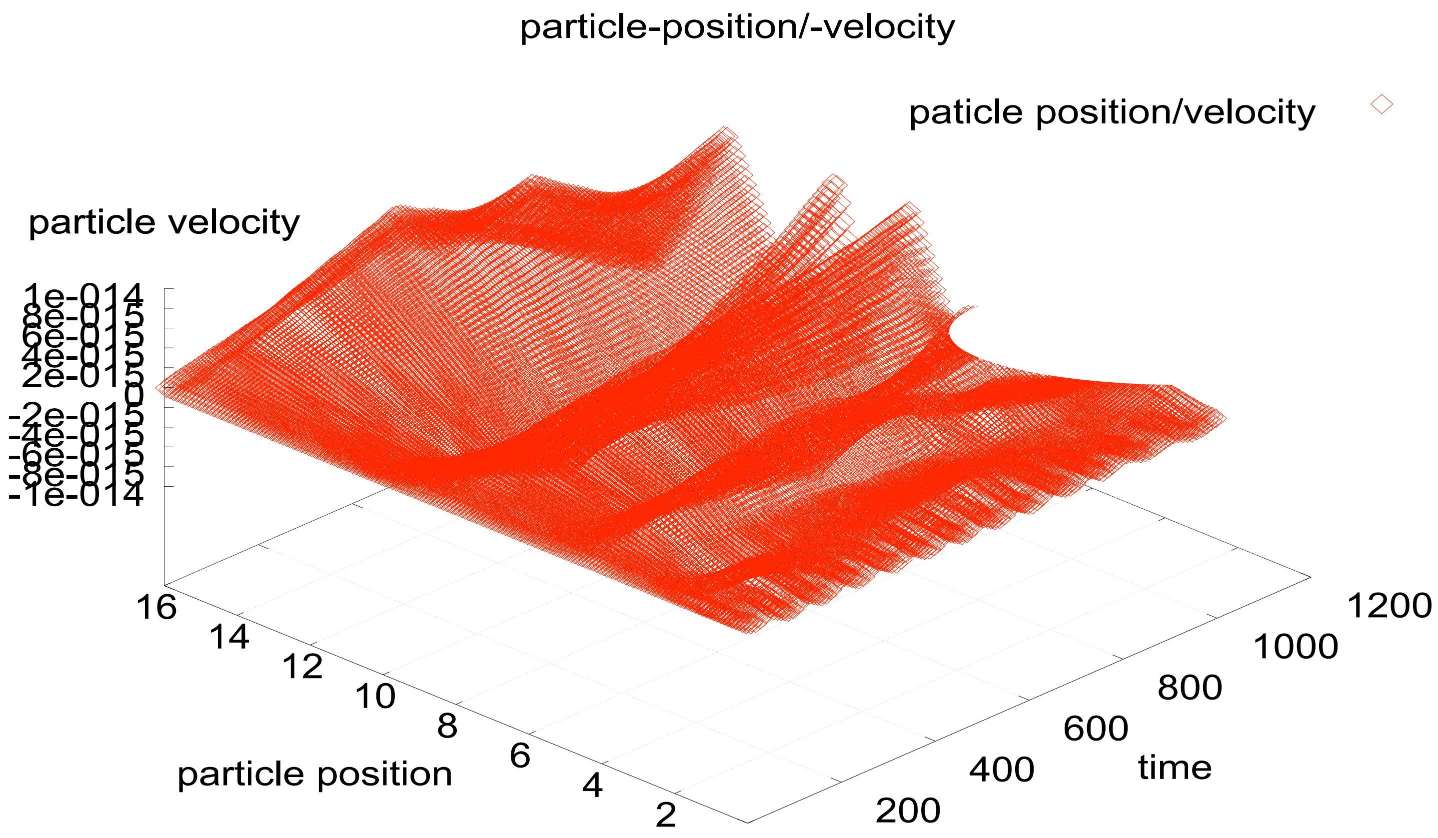


Fig.2’ : A simulation of 1D Electro-static particle-in-cell code (it=1001) : Particles at the position 0~16 with particle velocity~0 uniformly at time=0 change to particle velocity +2.~ -1 periodically, gradually raising the difference as time goes by 0~1001.

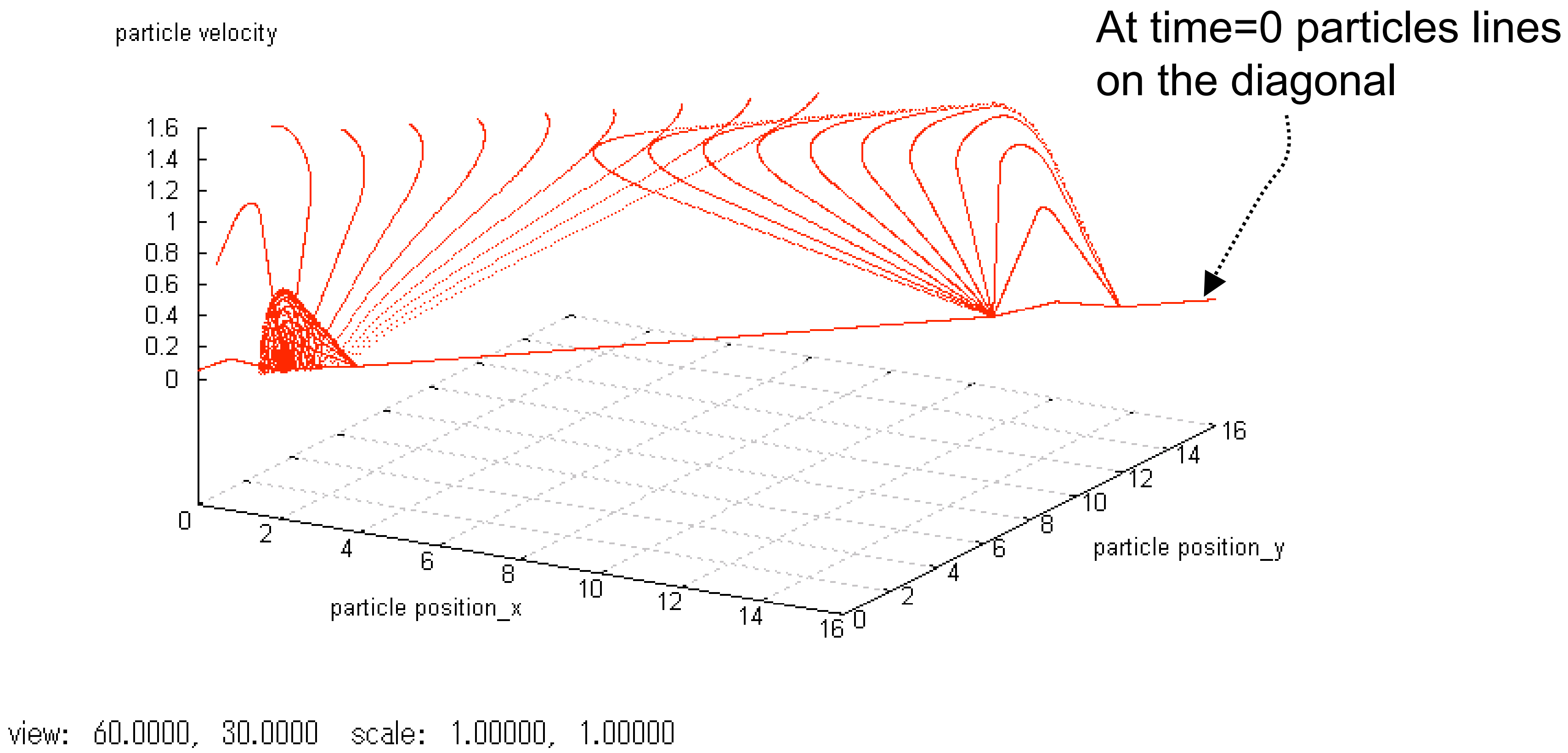


Fig.3 : A simulation of 2D Electro-static particle-in-cell code (it=101) :
Particles on the diagonal line with particle velocity~0 uniformly at time=0 change to particle velocity ~+1.6 drawing spirals as time goes by from 0 to 101.

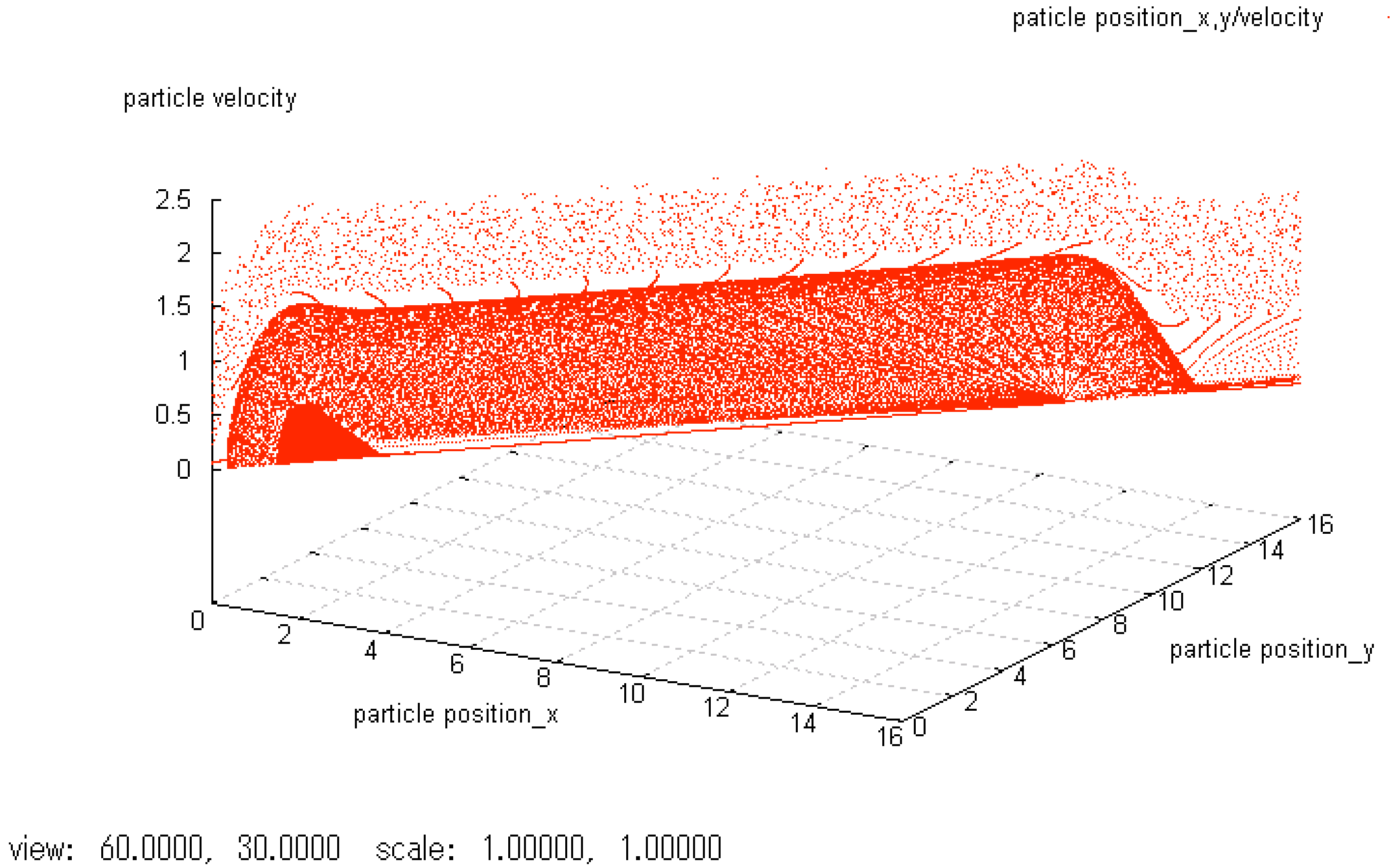


Fig.3' : A simulation of 2D Electro-static particle-in-cell code (it=1001) : Particles on the diagonal line with particle velocity~0 uniformly at time=0 change to particle velocity ~+1.6 drawing spirals as time goes by from 0 to 101.

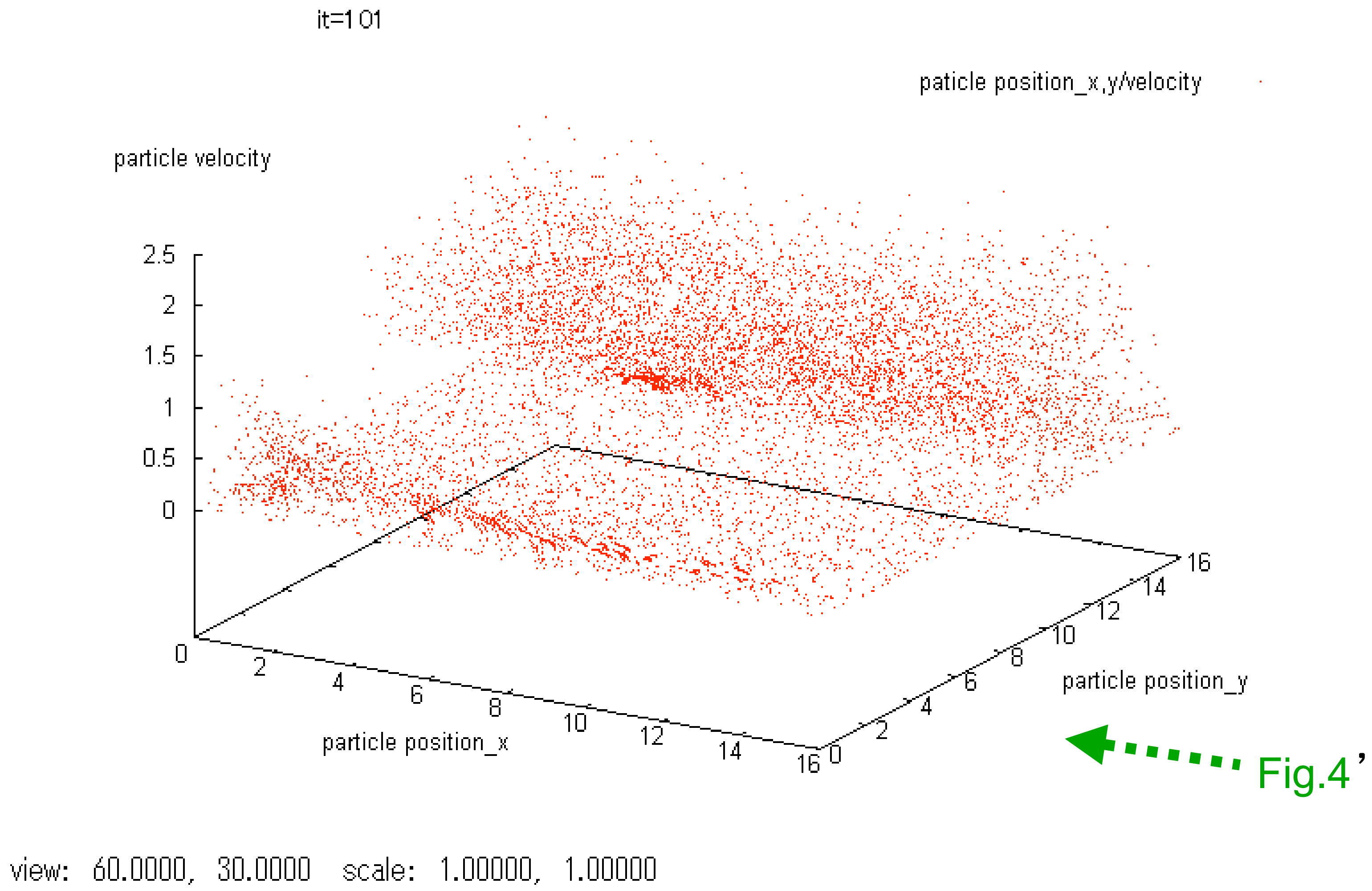


Fig.4 : A simulation of 2D Electro-static particle-in-cell code (it=101) : Particles on the x-y plane with velocity~0 uniformly at time=0 change to particle velocity ~+2.5 drawing spirals as time goes by from 0 to 101.

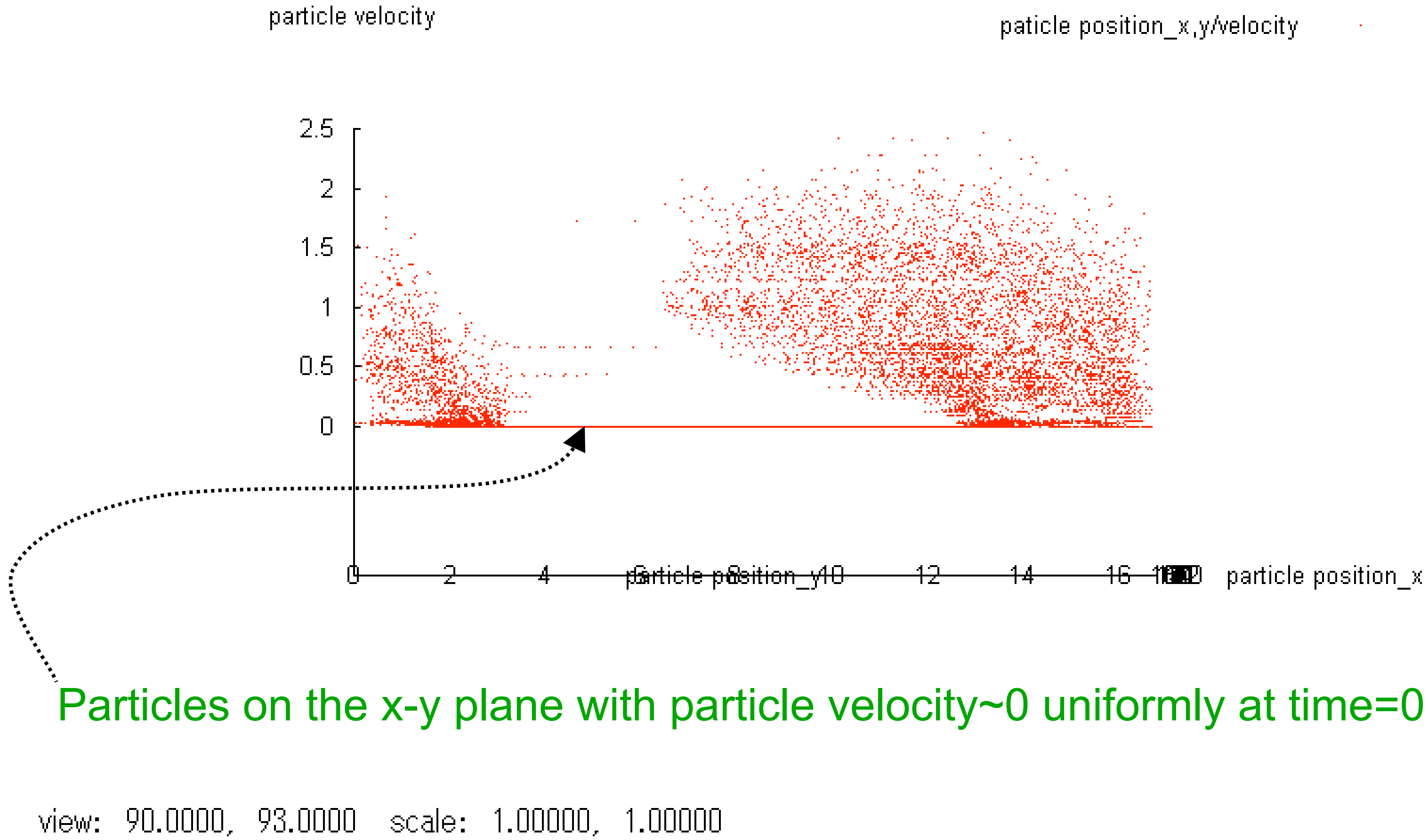


Fig.4’ : Side view from the X-axis of Fig.4 (it=101) :
Particles on the x-y plane with particle velocity~0 uniformly at time=0 change to particle velocity ~+2.5 on both sides as time goes by from 0 to 101.

Precision of finite difference method: Analysis of Numerical methods

1. An initial value problem of ordinary differential equation:

$$\frac{dy}{dt} = \alpha y, y(0) = y_0 \ldots..(1)$$

Usual way to use finite difference method for (1) is :

$$y_{n+1} - y_n = \alpha \Delta t y_n, y_0 = \beta \ldots..(2)$$

Truncated error of (2) is the order more than Δt (order 1) if Y' is replaced with $((Y_{n+1}-Y_n)/\Delta t)$ and it can be Taylor expanded at Yn :

$$y' \sim ((y_{n+1} - y_n)/\Delta t) + O(\Delta t) \ldots...(3)$$

If Y' is replaced by $((Y_{n+1}-Y_{n-1})/2\Delta t)$ and it is Taylor expanded at Yn, the truncated error is:

$$(y_{n+1} - y_{n-1})/2\Delta t \sim y' + O((\Delta t)^2) \ldots..(4)$$

This shows central difference method has higher precision (order 2). But, central difference method needs to have a value of y_1 : $y_1 = y_0 + \Delta t \alpha y_0$ (Euler's method). //

Chaos generated by Discretization for a differential equation

1. Logistic equation : Linear differential equation representing increase of an organism

$$\frac{du}{dt} = \varepsilon u ......(1)$$

u means the number of organisms, ε is a positive constant. A solution of (1) is:

$$u(t) = u_0 e^{\varepsilon t} ....(2),$$

U_0 is the initial value at t=0, and it is suitable for describing the increase of, for example, a bacterium. But a little bigger life, for ex., a drosophilia's increase is said to decrease as the function of the population u, and become saturated.

$$\frac{du}{dt} = (\varepsilon - hu)u ......(3)$$

(3) Is called Logistic equation, ε and h are positive constants, and it is made by modifying (1). The solution for of (3) is:

$$u(t) = \frac{\varepsilon C e^{\varepsilon t}}{1 + hCe^{\varepsilon t}}, C = \frac{u_0}{\varepsilon - hu_0} ......(4),$$

Fig. a shows (4) : It monotonously increasing as t, pass a inflection point, and asymptotically gets closer to the saturation point ε /h.

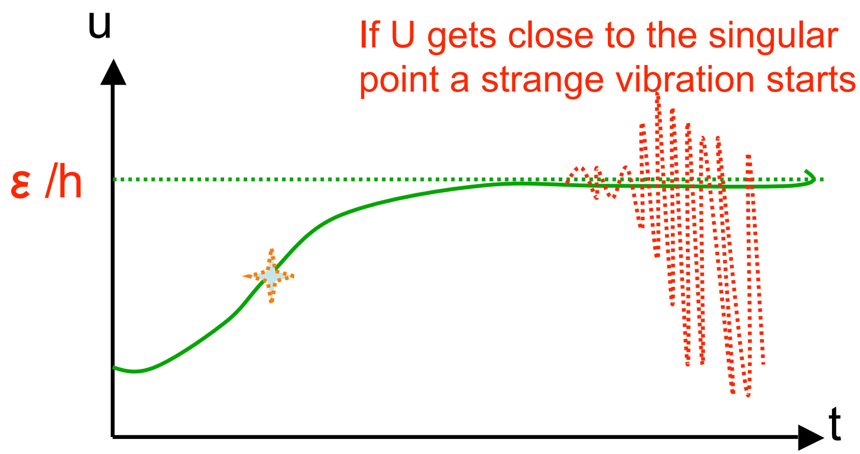


Fig. a

2. Discretization of Logistic equation

We have many methods to make a difference equation by discretyzing (3).

The best known is Euler's finite difference method [u(nΔt)＝u_n]:

$$\frac{u_{n+1}-u_n}{\Delta t}=(\varepsilon-hu_n)u_n........(5)$$

The others are :

$$\frac{u_{n+1}-u_n}{\Delta t}=(\varepsilon-hu_{n+1})u_n........(6)$$

$$\frac{\varepsilon(u_{n+1}-u_n)}{e^{\varepsilon\Delta t}-1}=(\varepsilon-hu_{n+1})u_n........(7)$$

3. Robert May's study :

Mathematics proved that "a solution of (5) approximates the solution of (3) by making Δt small enough in a finite time 0<t<T". But the infinite case (nΔt→＋∞) of the solution for (5) have remains unknown.

To rewrite (5) with a=1+ε Δt, (hΔtu_n)/(1+ε Δt)=X_n , make a finite difference equation with X_n :

$$x_{n+1}=ax_n(1-x_n)........(8)$$

(8) is a quadratic function and has max value a/4 at x_n=1/2. Then, if 0<a<4 & 0<x_n<1, it follows 0<x_{n+1}<1. So we think only 0<a<4 & 0<=x_0<=1

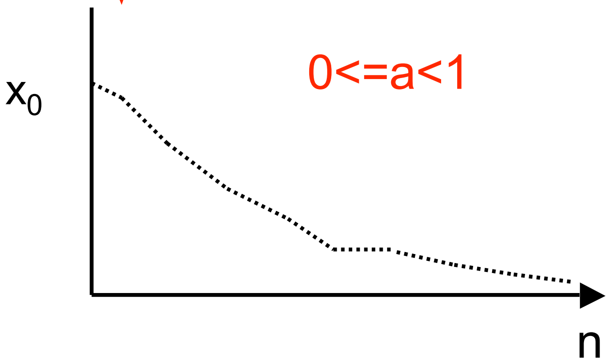
[Extracted from Sugaku seminar "Nonlinear phenomena & analysis",1981]

To change a means to change **ε** or Δt.

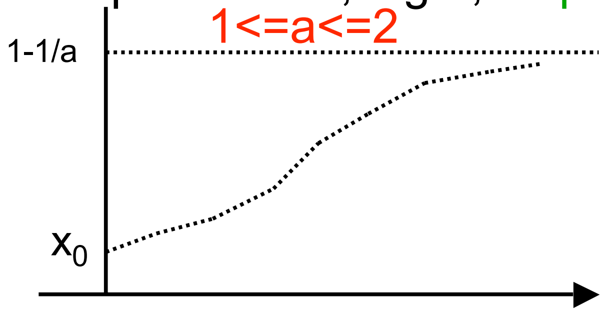
$$x_{n+1} = ax_n(1 - x_n)........(8)$$

The behavior of the solutions of (8) depend on the value of a.

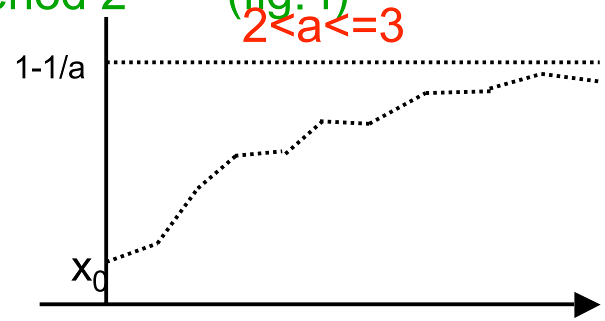
1. 0<=a<1 : X_n shows monotone decreasing and $x_n \to 0$.   (fig. b)
2. 1<=a<=2 : X_n shows monotone and $x_n \to 1-(1/a)$.   (fig. c)
3. 2<a<=3 : X_n shows not monotone, but attenuated vibration and $x_n \to 1-(1/a)$.   (fig. d)
4. $3 < a <= 1+\sqrt{6} = 3.449...$: X_n shows period two vibration.   (fig. e)
5. $1+\sqrt{6} = 3.449... < a$: we can see period four, eight, … period 2^n (fig. f)



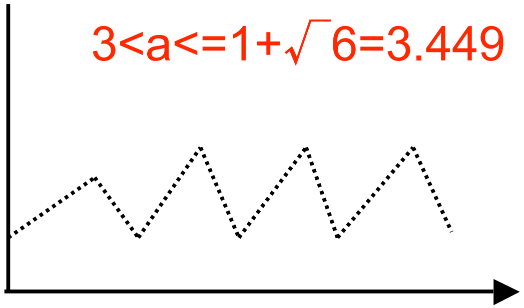
(fig. b)



(fig. c)

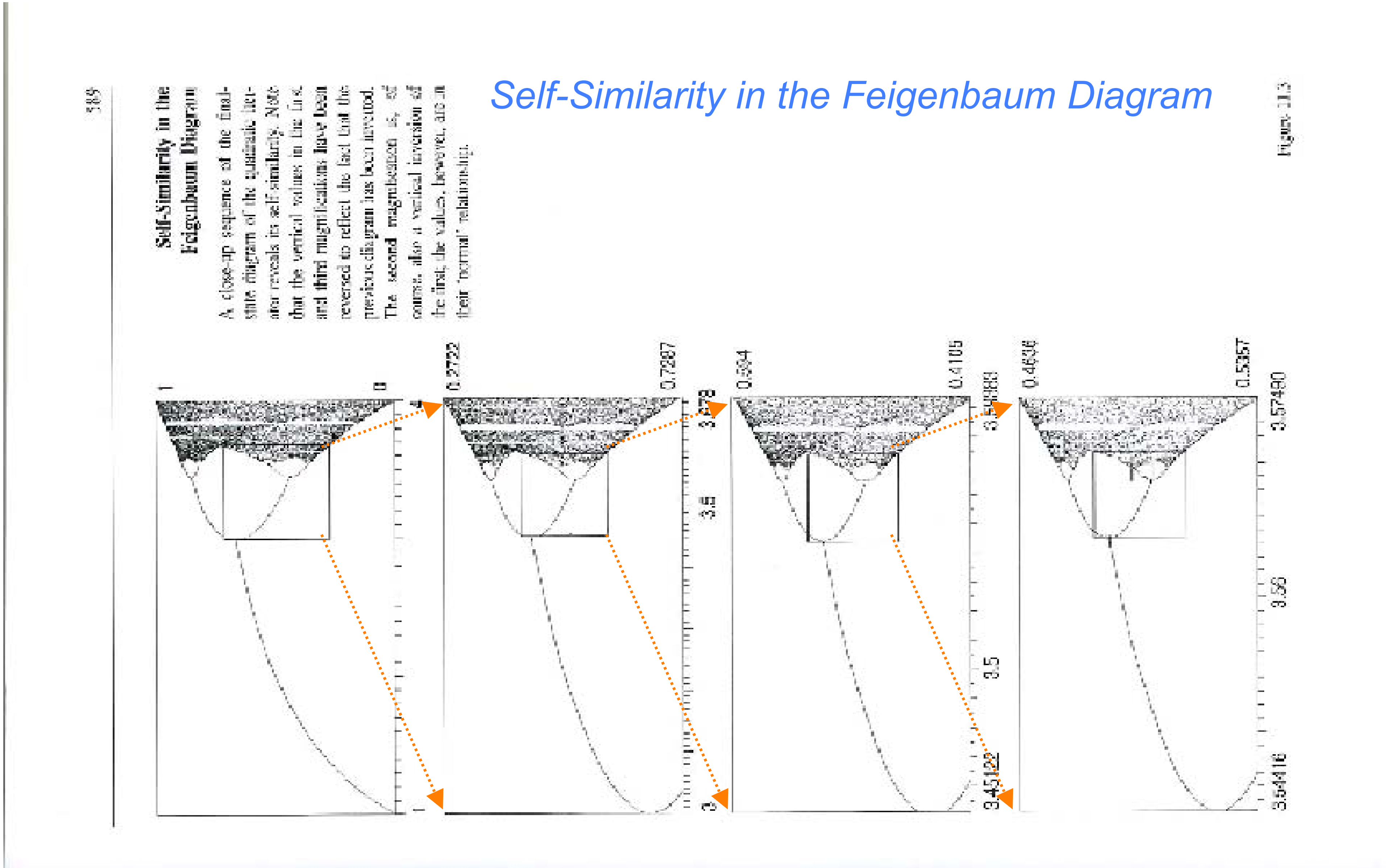


(fig. d)



(fig. e)

Self-Similarity in the Feigenbaum Diagram

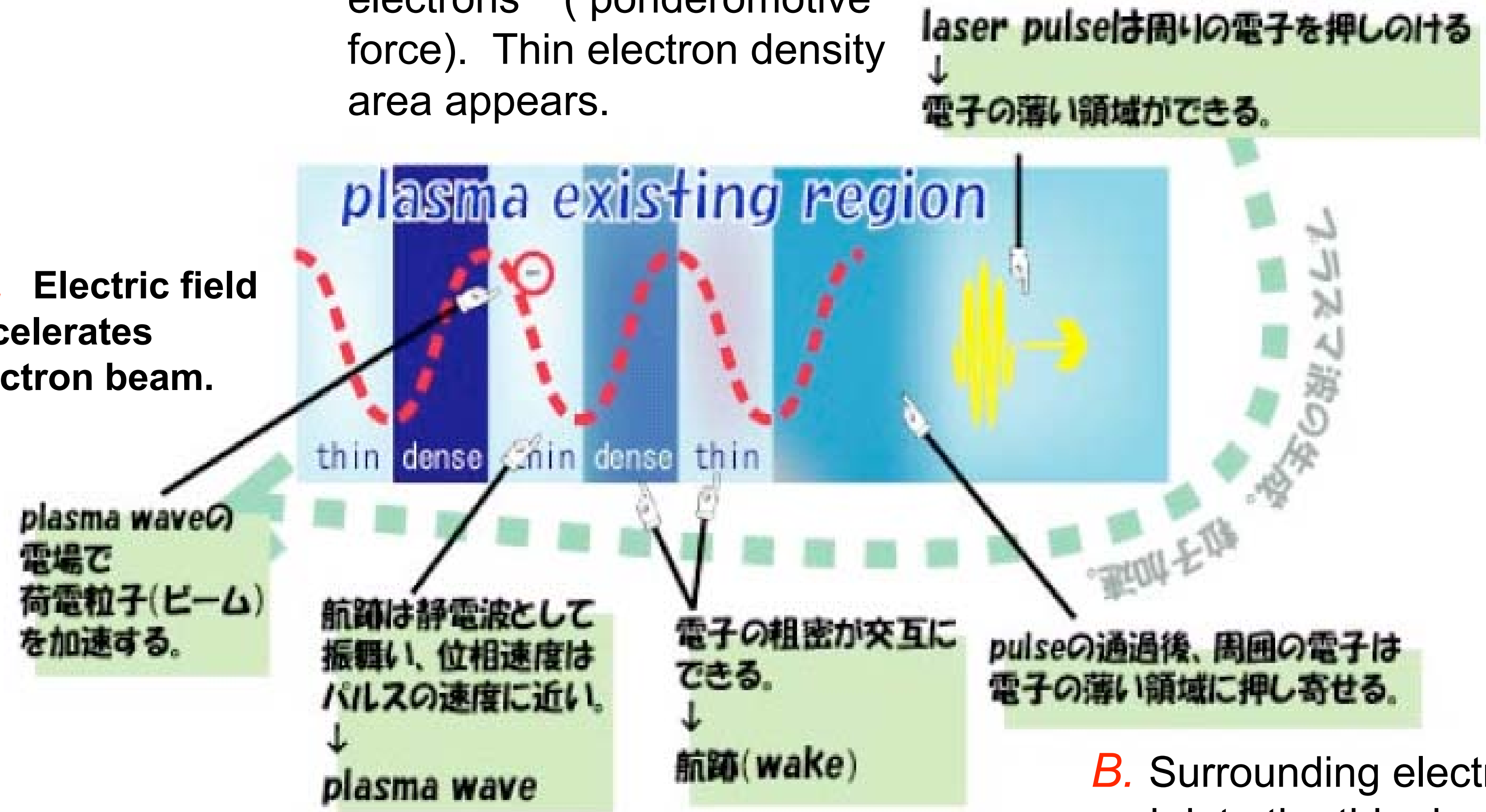


period 2n : fig. f

Extracted from “Chaos and Fractals, Peitgen,Jurgens,Saup” ; p.589

Ionized Plasma consists of ions and electrons. Plasma wave is a compression wave of plasma electrons, and it is called Electrostatic wave, or Langmuir wave.

A. Laser pulse push surrounding electrons (ponderomotive force). Thin electron density area appears.



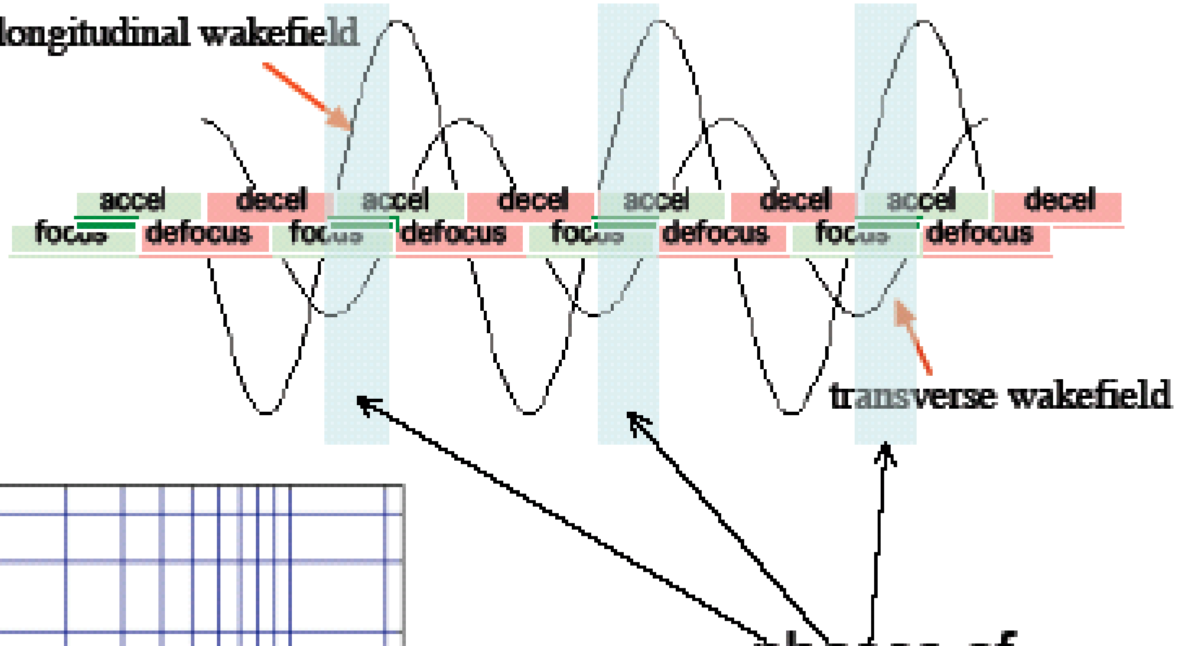
E. Electric field accelerates electron beam.

B. Surrounding electrons rush into the thin density area.

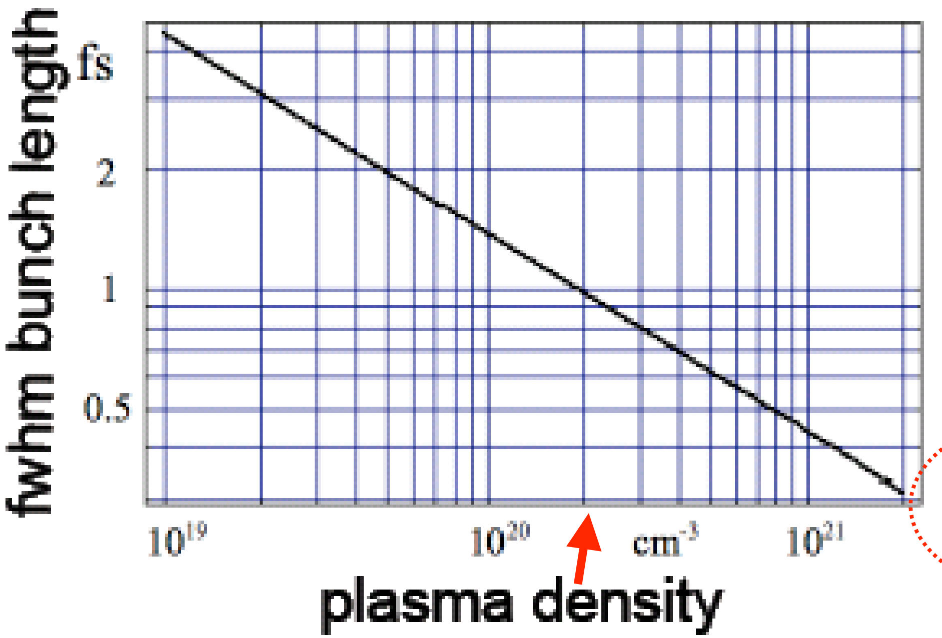
D. Wake behaves as an electrostatic wave = plasma wave.

C. Thin and dense areas appear = wake.

LWFA Linear model says fwhm bunch length < plasma wavelength/~~4~~8



phases of acceleration and focusing



$2\ 10^{20} cm^{-3}$ plasma density gives bunch length below 1fs.

Plasma Vibration (Simple harmonic O s c i l l a t i o n)

Angular frequency : $\omega=(k/m)^{1/2}$; constant: k, mass: m,

$F=kL$; force proportional to distance: F, distance: L（単振動）

Coulomb force : $F=e^2/4\pi\varepsilon_0 r^2$

$L\sim r$; the distance between electrons

$$k= e^2/4\pi\varepsilon_0 r^3= e^2 n/4\pi\varepsilon_0 \leftarrow \quad n=1/V=1/r^3: \text{Electron density}$$

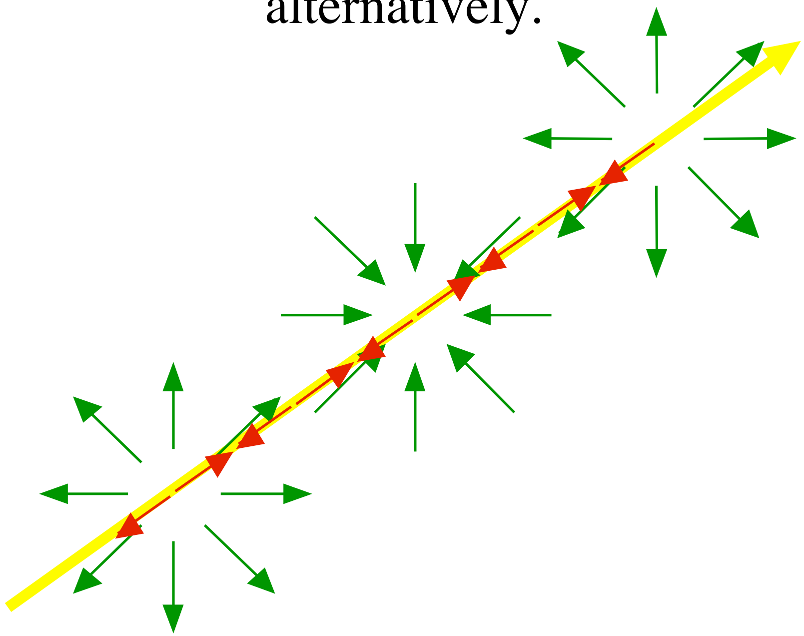
$$\omega= (k/m)^{1/2}\sim (e^2 n/4m\pi\varepsilon_0)^{1/2}$$

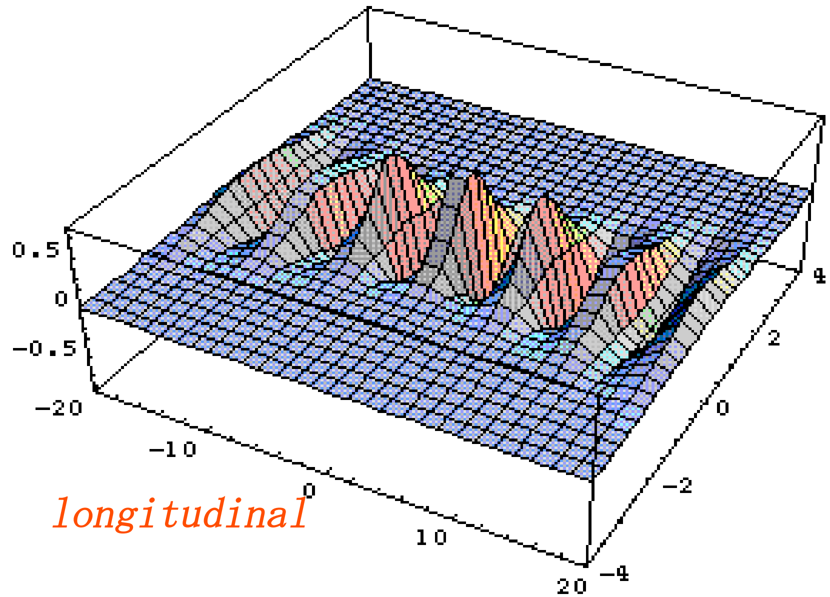
Plasma wave is excited in both

longitudinal & transverse directions

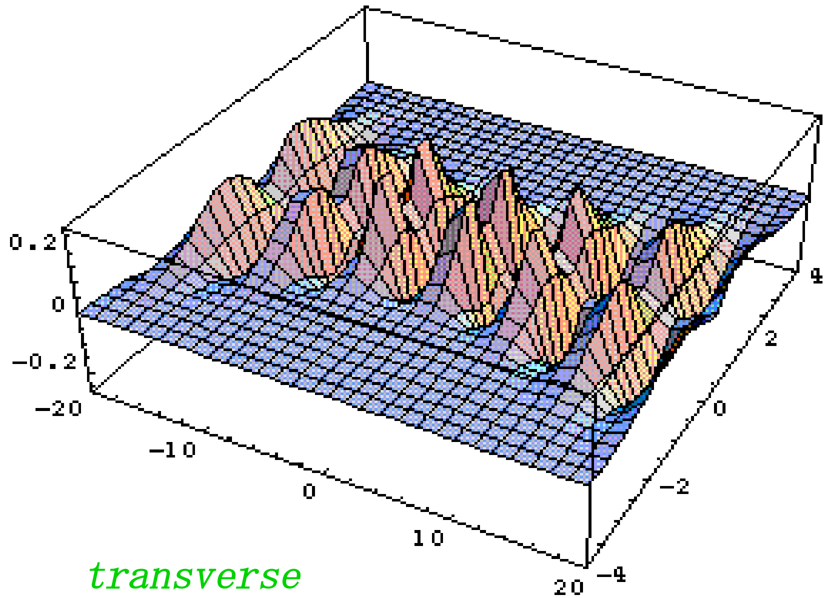
in longitudinal direction we have acceleration and deceleration phases alternatively.

in transverse direction we have focusing and defocusing phases alternatively.





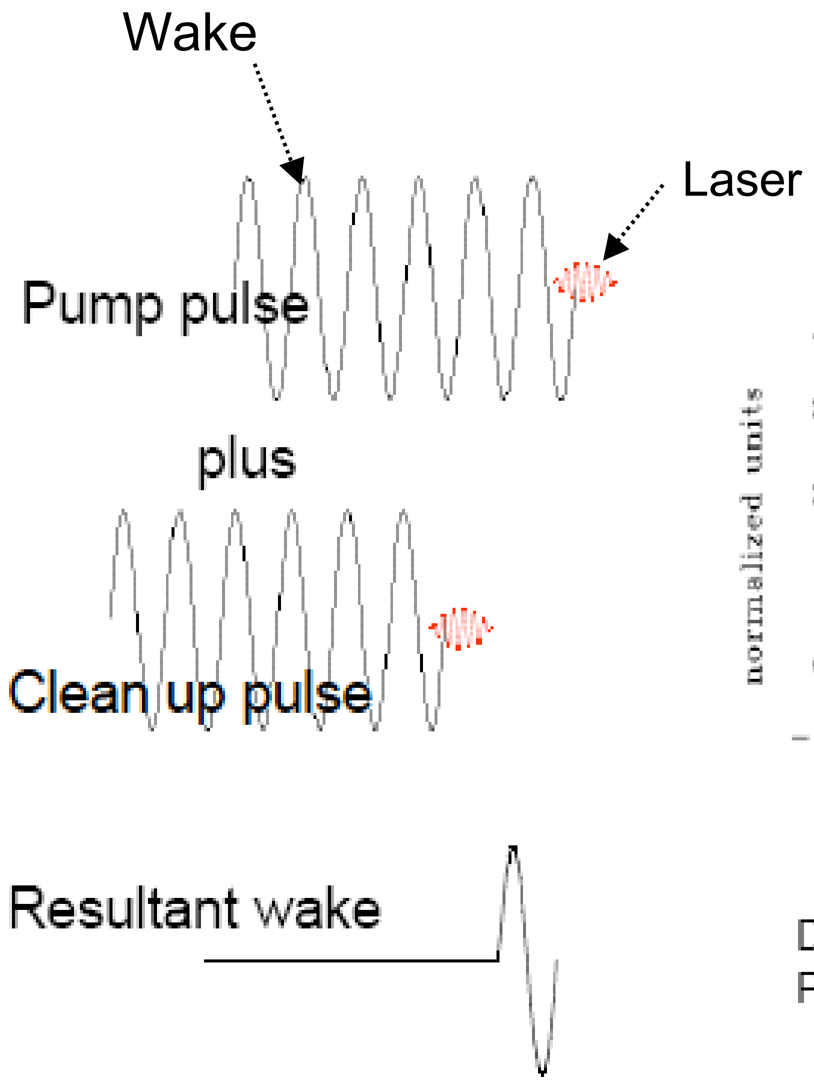
longitudinal

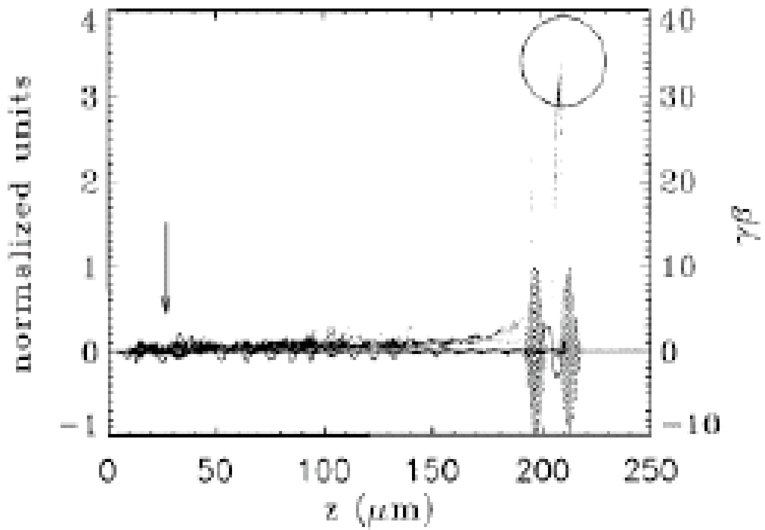


transverse

Plasma Wakefield Acceleration（ＰＷＦＡ or LWFA)

an old idea

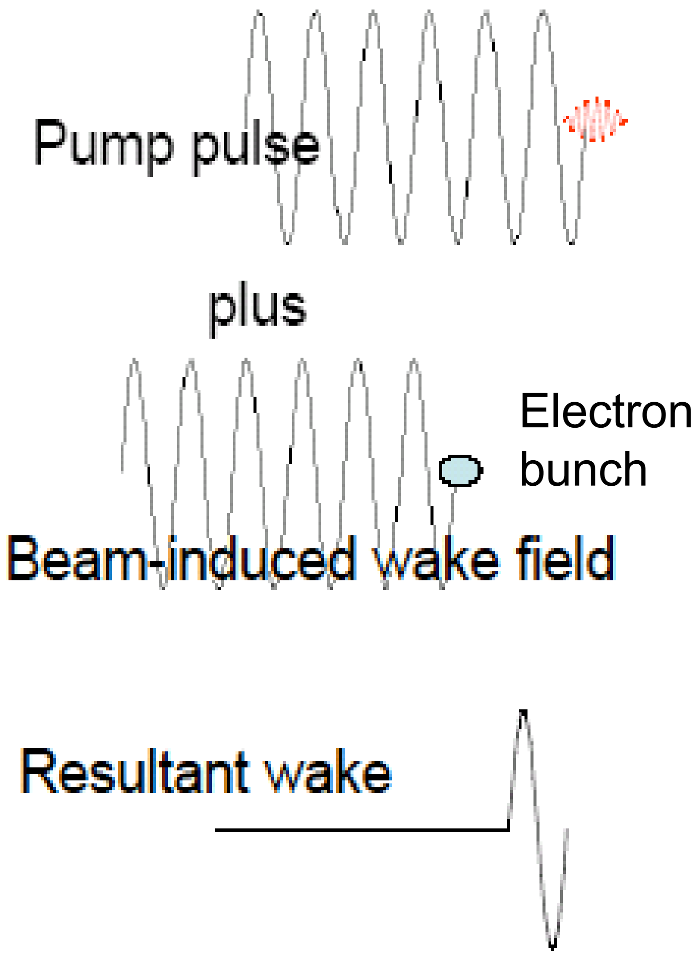


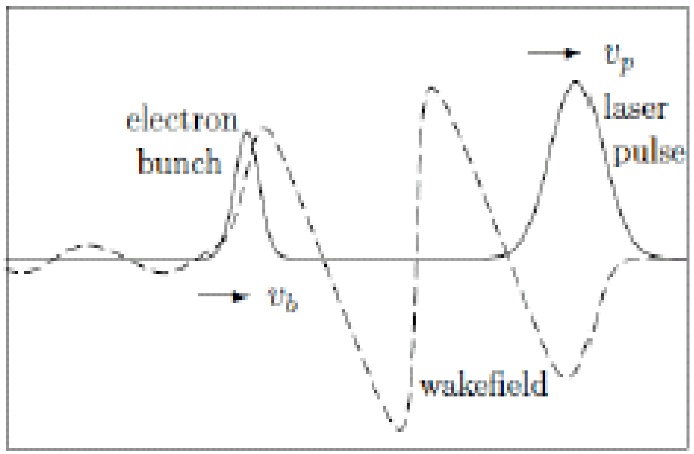


D.Umstadter, et al.,
Phys.~Rev.~Lett. 76 (1996) 2073.

another old idea: beam loading

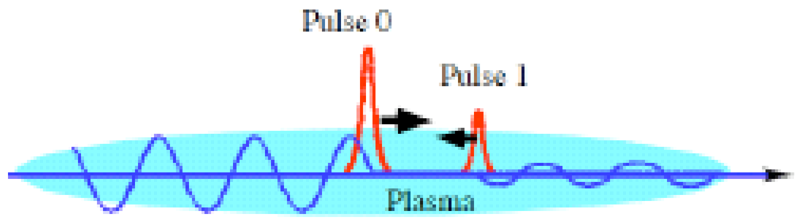
Heavy beam loading cleans up the wake created by a laser.





A.J.W.Reiysma et al.,
PRL94(2005)085004

New idea : Create just a single pulse from the very beginning



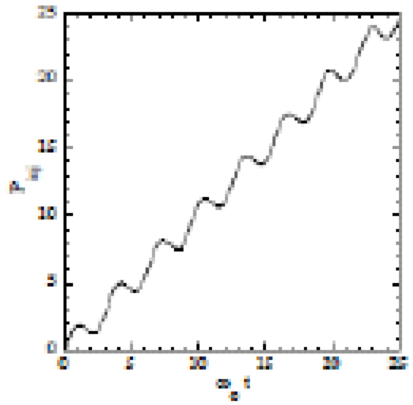
Two counter-injected laser pulses with equal frequencies produces stationary wave, in which electrons gets energy large enough to be trapped in a wake produced by the pump.

対向する2個のｆｓレーザビーム：

レーザ継続中だけ**定在波ができ**、

レーザ領域に高温プラズマが生成され、

高温のプラズマ電子は航跡場に補足される



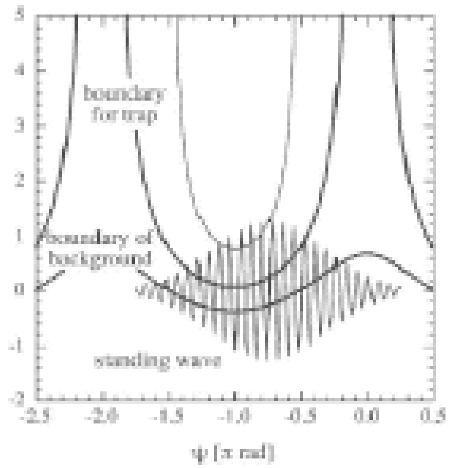
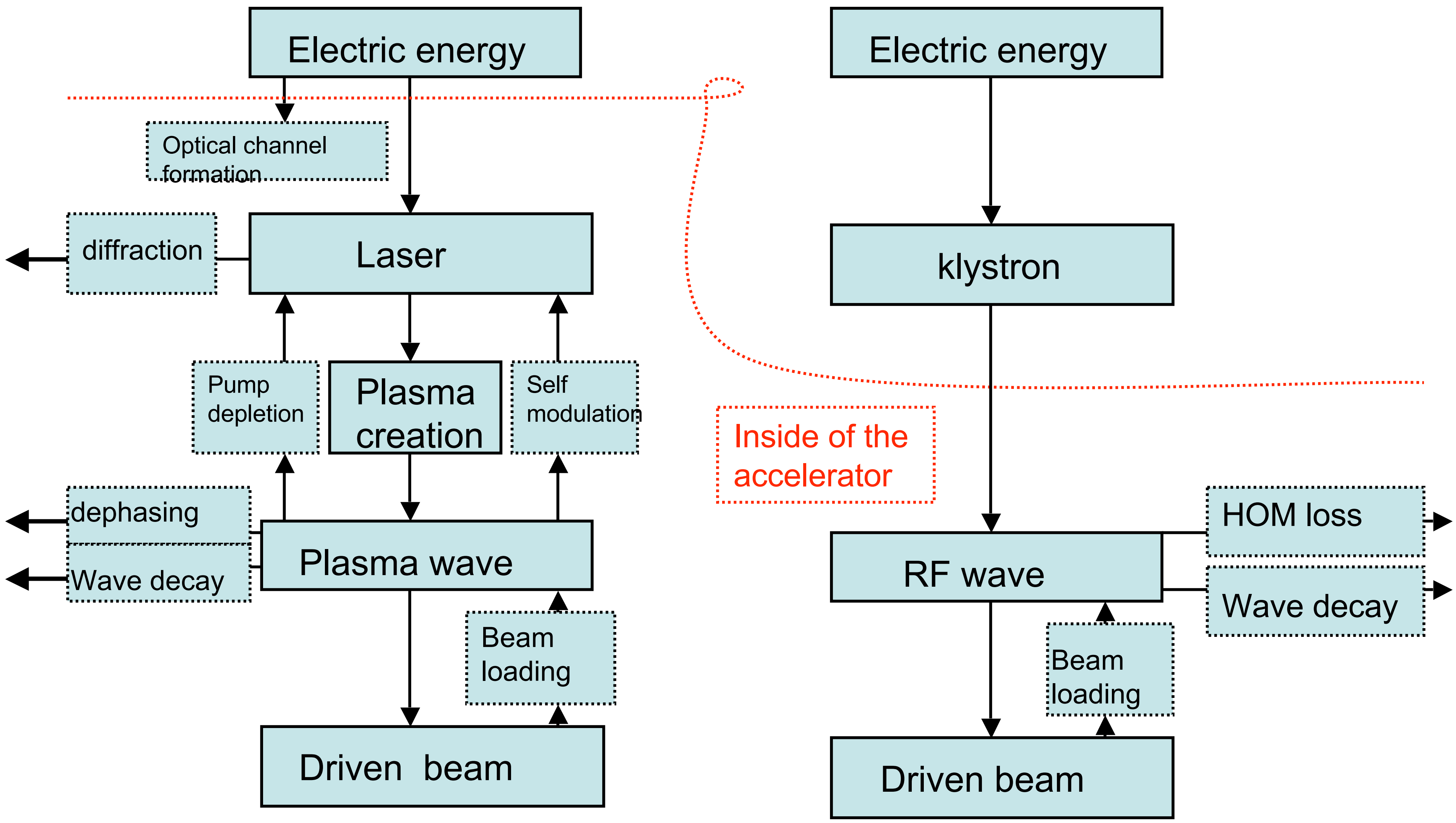


FIG. 2. Longitudinal acceptance of a standing wave and a traveling wave with a phase velocity at the speed of light for the condition of $k_p = \pi\sigma_z$ for $a_0 = 1.0$ and $a_1 = 0.4$. Electrons beyond the trapped orbit are injected into the wakefield.

H.Kotaki, et al., Phys.Plasmas 11(2004)3296.



Laser wakefield accelerator; electric energy changes into laser (light), and plasma wave.

RF linear accelerator

Flow of energy [A.Ogata, Nucl. Instr.Meth.410(1998)]

◆Towards Laser Plasma Accelerator
（　～１００as？　）

・Plasma density　$\propto$　Short wave length
$\propto$　Big accelerating gradient
$\propto$　Short bunch length (∝:proportional to)

・Tera watt Laser appears and makes Laser plasma accelerator a reality (~2010)

Summary

1. Made a 1 **D Electro-static Particle-in-cell** :

2. Making 2 **D Electro-static Particle-in-cell**:

3. Starting Mathematical analysis for unstable phenomena by discretized solutions used in **Finite difference method (2D Chaos)**

Next step

4. **Complete electromagneticfield** : Interaction between floating electromagnetic field and charged particles

5. **Realize a simulation for Laser Wakefield Acceleration**