Vlasov Description of a Supercritical Magneto-acoustic Compression Pulse.

R. Chodura

K. Graf Finck v. Finckenstein

IPP 1/113
6/ 92

December 1970

INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MUNCHEN

INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

Vlasov Description of a Supercritical
Magneto-acoustic Compression Pulse.

R. Chodura
K. Graf Finck v. Finckenstein

IPP 1/113
6/ 92 December 1970

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Institut für Plasmaphysik GmbH und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

IPP 1/113 R. Chodura
6/ 92 K. Graf Finck v. Finckenstein

Vlasov Description of a Supercritical Magneto-acoustic Compression Pulse.

(in English)

December 1970

Abstract:

The time evolution of a compression pulse driven by a fast rising magnetic field and propagating perpendicularly to a homogeneous bias magnetic field is investigated numerically. The compression parameters are chosen such that overturning occurs in the pulse (supercritical pulse). The fluid description of the ions then breaks down and is replaced by a Vlasov description.

The ion distribution develops a two-stream structure giving rise to a strong broadening of the spatial density profile of the pulse, the occurrence of an upstream "foot" in the magnetic field profile, and strong heating of the ions.

Part I

1. Introduction

Acceleration of a plasma by a "magnetic piston", i.e. a current layer at the interface between plasma and vacuum magnetic field, was investigated in the past by numerically solving fluid equations at various levels of complexity [1]. These fluid models showed the expected behaviour of the formation of a current sheath as given by the resistivity η of the plasma or, for $\eta \rightarrow 0$, by electron inertia. After this build-up the magnetic piston begins to move, collecting plasma in a growing density pulse and steepening the flank of the transition of the magnetic field from plasma to vacuum. This steepening process may be limited by resistive dissipation of the current in the piston. If, however, the piston is sufficiently strong, the density pulse becomes infinite in a cold plasma [2] or, in a hot plasma, develops a sharp peak with a flank of infinite steepness [3,4], thus indicating overturning of fluid elements from behind over their predecessors. This overturning process is more familiar in the form of water-waves breaking in shallow water.

The occurrence of the density peak in the "supercritical wave" indicates failure of fluid description of the acceleration process. The fluid description is applicable as long as the velocity distribution of the plasma constituents can be characterized by a few low moments, i.e. density, mean velocity, pressure, viscosity, and heat flow. It is necessary that so-called transport effects, i.e. viscosity and heat flow, be small corrections. As soon as the density tends to peak up these transport effects become large, thus indicating that higher moments of the velocity distribution come into play which are not allowed for in the fluid description.

The applicability of the fluid description is violated for the ion fluid first: In order to neglect higher moments in the fluid description of the collisionless plasma considered here, one has to postulate a small ratio of gyro-radius to typical pulse dimension. This condition is obviously more restrictive for ions than for electrons.

Thus, in order to get some insight into the acceleration process in a supercritical compression wave, one has to dispense with a fluid description at least for the ions and investigate their behaviour by means of a Vlasov equation.

Electrons are treated as a cold fluid transmitting the Lorentz forces of the magnetic piston by means of electrostatic fields acting on the ions.

2. Model

An infinitely extended, homogeneous plasma slab is initially confined by two walls, separated by a distance 2L, and immersed in an uniform magnetic field B_o parallel to the walls (Fig. 1). At a time $t = 0$ the walls are removed and the vacuum field outside the plasma is changed in magnitude according to some growing time function, thus launching a magneto-acoustic compression wave into the plasma.

As stated above, the electrons are treated as a cold fluid. In order to allow for some field penetration into the plasma, the electrons are assumed to be involved with the ions in a friction process described by a constant friction coefficient ν. Electron inertia is neglected.

The ions are assumed to move in the x-direction only, i.e. Lorentz forces acting on the ions are neglected. This condition restricts the time of applicability of our model to a small fraction of an ion gyro-period. But this time-interval is long enough to include all essential features of steepening and overturning of the wave to be described here.

The electric space charge field is determined by a quasi-neutrality condition rather than by solving Poisson's equation. Putting F(x,w,t) for the number of ions at the point x with the x-component of the velocity w at time t, we get Vlasov's equation

$$\frac{\partial F}{\partial t} + w \frac{\partial F}{\partial x} - v B \frac{\partial F}{\partial w} = 0 \tag{1}$$

where v is the fluid velocity of the electrons in the y-direction as given by their momentum equation

$$-E + u B - \nu v = 0 . \tag{2}$$

The magnetic field B and the y-component of the electric field E are given by Maxwell-equations

$$\frac{\partial B}{\partial x} = \varepsilon n v \tag{3}$$

$$\frac{\partial E}{\partial x} = - \frac{\partial B}{\partial t} . \tag{4}$$

The variables n and u are the first two moments of F, i.e.

$$n = \int_{-\infty}^{\infty} F \, dw \tag{5}$$

$$u = \frac{1}{n} \int_{-\infty}^{\infty} F w \, dw . \tag{6}$$

In order to get rid of constant parameters like the ion and electron mass, M and m, and charge, Ze and e respectively, we have introduced normalized variables in (1) to (6):

x is normalized by L, t by a value t_o, to be specified later, w and u by L/t_o, v by $(M/Zm)^{1/2} L/t_o$, B by $(Mm/Z)^{1/2}/(e\, t_o)$, and E by this normalization value of B times L/t_o. The (dimensionless) parameter ε is given (in e.m.u.) by:

$$\varepsilon = 4\pi e^2 n_o L^2/m \qquad (7)$$

where n_o is the normalization for the number density and c the velocity of light. As initial conditions we take

$$B(t=0) = \text{const} = B_o\ ,\ v(t=0) = 0,\ E(t=0) = 0$$

and a Maxwell distribution for F ,

$$F(t=0) = \frac{1}{\sqrt{2\pi T_o}} \exp(-\, w^2/2\, T_o) \qquad (8)$$

where T_o is a homogeneous temperature, normalized by $M\, L^2/t_o^{\,2}$.

The compression pulse is initiated by rising the external magnetic field according to a boundary condition

$$B_e(t) = B(x \rightarrow \pm\infty\ ,\ t) = B_o + \quad t\ . \qquad (9)$$

Either B_o or can be made equal to 1 by appropriately choosing t_o .

A model similar in some respects to ours was treated by AUER, HURWITZ, and KILB [5]. These authors investigated a supercritical pulse by following the motion of discrete plasma sheets. Quasineutrality was achieved by loading each sheet with equal electron and ion charges. This, apparently, is too strong a constraint on the relative motion of the electrons and ions because, when two sheets overturn, it leads to the ambiguity whether electrons will keep moving with their original ion charges or change place with electrons of the overturned sheet.

This ambiguity is avoided in our model by requiring equal mean velocity u for electrons and ions only.

3. Results

As was stated in Sect. 2 our model involves four dimensionless parameters: the number of particles in the slab ε , electron friction coefficient ν , initial ion temperature T_o , and either the initial magnetic field B_o or the rate of change of the external magnetic field α. It was shown in [4] by a fluid description that rising ε or α results in faster steepening of the compression pulse, while rising ν or B_o makes it more smooth.

We are interested here in the transition from a subcritical to a supercritical pulse (i.e. a pulse with overturning). We, therefore, investigated the behaviour of the pulse for rising values of ε, keeping the other parameters constant.

Fig. 2 shows results for a weakly supercritical pulse with $\varepsilon = 1$, $\nu = 1$, $T_o = 0.4$, $B_o = 1$ (by choosing $t_o = (M\, m)^{1/2} / (e\, B_o\, Z^{1/2})$), $\alpha = 100$.

The profiles in Fig. 2a for density n, magnetic field B and ion pressure p for different times t (normalized by t_o) are calculated from a Lagrange fluid description. At the beginning the profiles are fairly smooth, but after a time $t \sim 0.1$ the density and pressure develop the steep flank characteristic of overturning.

Fig. 2b gives results from the Vlasov description for the same set of parameters. The microscopic view of the ion-velocity distribution at two different points shows fairly regular behaviour in the outer part of the plasma at $x = 0.82$: the distribution remains Maxwellian, being shifted inwards only and broadened by adiabatic compression. But, considering now the development of the velocity distribution at a point where the steep flank develops in the fluid description, $x = 0.614$, one finds shifted Maxwellians only for a time interval $t \sim 0.08$. Then the distribution function develops a "foot" toward negative (inward) velocities, thus showing a property which is not resolved in the fluid description. As a consequence of this behaviour of the ion distribution, the moments of zeroth and second order, n and $p = \int (w - u)^2\, F\, dw$, show a smoother shape than in the fluid description, while B is essentially unchanged.

The same comparison is done for a strong supercritical pulse with $\varepsilon = 100$ in Fig. 3. As can be seen, the fluid description breaks down very soon, practically at the outset of inward motion. This is understandable if one looks at the time development of the ion distribution at $x = 0.82$: the Maxwellian distribution is distorted at the same time as the inward motion sets in. Nevertheless, after some time,

$t \sim 0.15$, a (shifted) Maxwellian is restored. Looking now at the time development of the ion distribution at an inner point of the plasma ($x = 0.614$), we find a Maxwellian that is practically constant during the whole computation time and describes plasma particles not yet reached by the magnetic piston at this position. This Maxwellian is accompanied by a shifted Maxwellian growing in time which describes particles accelerated in the outer regions of the plasma by the magnetic piston and now streaming through the remaining part of the plasma. Thus a two-stream structure of the ion-distribution with two shifted Maxwellians is established.

Again forming moments of the ion-distribution, we find a smooth profile for number density n . The steepness of the upstream flank of the density pulse is nearly constant in time, the thickness of the pulse being larger than in case $\varepsilon = 1$.

The second moment, i.e. ion pressure p, shows a similar broad profile. The development of the two-stream structure of the ions due to overturning shows up in their strong non-adiabatic heating (up to a factor 25 in our calculation).

The magnetic field profile of the strong supercritical pulse is composed of two parts: a steep jump preceeded by a long "foot". This foot is apparently connected with the overturning process: the overturning ion stream flushes electrons with it which in turn compress some magnetic flux ahead of the main magnetic piston. The occurrence of such a foot in the magnetic field profile was observed in several supercritical collisionless shock experiments [6].

4. Conclusions

Description of overturning ions in supercritical magneto-acoustic compression pulse by a Vlasov equation reveals the development of a two-stream structure of the ion flow which cannot be gained by a one-fluid description of the ions, nor by taking transport effects such as viscosity into account. (Perhaps a two-fluid description of the ions would work.)

Together with a cold electron component the weighted velocity distribution $f = f_e + m_e/m_i\ f_i$ (see Fig. 4) should exhibit a three-stream instability.

Without allowance for spatial inhomogeneity and the temperature in the two ion beams the dispersion relation is given by

$$1 = \frac{\omega_{pi1}^2}{(\omega - k\, u_1)^2} + \frac{\omega_{pi2}^2}{(\omega - k\, u_2)^2} + \frac{\omega_{pe}^{\ 2}}{(\omega - k\, u\ )^2} \qquad (10)$$

where $\omega_{pe}^2 = \dfrac{4\pi\, n\ e^2\ c^2}{m}$

$$\omega_{pi1} = \frac{4\pi n_1\ e^2 c^2}{M},\ \omega_{pi2} = \frac{4\pi n_2\ e^2 c^2}{M}$$

$$n_{1,2} = \int f_{i1,2}\ dw\ ,\quad u_{1,2} = \int w\ f_{i1,2}\ dw$$

together with the conditions

$$n_1 + n_2 = n\ ,\quad n_1\ u_1 + n_2\ u_2 = n\ u\ .$$

As is well known from the two-stream case, the dispersion relation (10) has complex roots for small enough k, thus indicating instability.

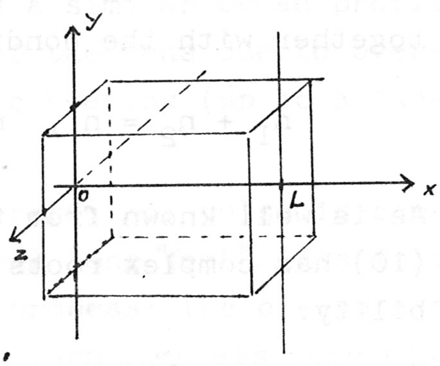
In a more realistic case we had to take into account the thermal spread of the electrons, which may be large compared with u_1, u_2, u. This will reduce the growth rate of the instability or even quench it [7].

The instability just mentioned does not show up in our numerical calculations because we have replaced Poisson's equation by a quasineutrality condition.

Part II: Numerical Treatment

1. The problem

The problem will be treated in one-dimensional space, i.e. with respect to the other two dimensions there is homogeneity. A plasma composed of ions and electrons between two material walls at $x = 0$ and $x = L$ is considered. The plasma can be regarded as a finite mass by considering a square prism with a cross section 1×1 between the two walls. A magnetic field B that is uniform in x at the time $t = 0$ is applied in the z-direction. As boundary condition this magnetic field grows linearly with time at the outer boundary $(x = L)$. This results in the formation of a compression wave that moves towards the plane of symmetry $(x = 0)$. This effect is simulated by numerically solving the Vlasov equation.



2. Equations

Notation: Let

x = space coordinate, $0 \leq x \leq L$,
w = velocity coordinate of the ions in the x-direction, $-\infty < w < \infty$,
t = time, $0 \leq t < \infty$,
F = distribution function of the ions
($F = F(x,w,t)$ is the ions density in the x-w space at time t),
$B = B(x,t)$ magnetic field in the z-direction,
$E = E(x,t)$ electric field in the y-direction,
$v = v(x,t)$ electron velocity in the y-direction (ion velocity in the y-direction ignored).

In addition, let

$n = n(x,t) = \int_{-\infty}^{\infty} F\,dw$ = ion density at point x at time t,

$u = u(x,t) = \frac{1}{n} \int_{-\infty}^{\infty} F\,w\,dw$ = mean velocity of the ions in the x-direction at point x at time t,

$p = p(x,t) = \int_{-\infty}^{\infty} F(w - u)^2\,dw$ = pressure at the point x at time t,

$T = T(x,t) = \frac{p}{n}$ = temperature of the ions at point x at the time t.

The equations treated are as follows:

$$\frac{\partial F}{\partial t} + w \frac{\partial F}{\partial x} - v \cdot B \cdot \frac{\partial F}{\partial w} = 0 \qquad \text{(Vlasov equation)} \quad (1)$$

$$E = uB - \nu \cdot v, \quad \nu \geq 0 \text{ constant} \qquad \text{(Ohm's law)} \quad (2)$$

$$\frac{\partial B}{\partial t} = -\frac{\partial E}{\partial x} \qquad \text{(law of induction)} \quad (3)$$

$$v = \frac{1}{\varepsilon \cdot n} \cdot \frac{\partial B}{\partial x}, \quad \varepsilon \geq 0, \text{ constant} \qquad \text{(Ampere's law)} \quad (4)$$

Equations (2), (3), (4) are combined in one equation, E and v being eliminated:

$$\frac{\partial B}{\partial t} = \frac{\partial}{\partial x} \left(\frac{\nu}{\varepsilon \cdot n} \cdot \frac{\partial B}{\partial x} - uB \right) \quad (5)$$

Integrating eq. (1) with respect to w yields the equation of continuity:

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x} (n \cdot u) = 0 \quad (6)$$

By integrating eq. (6) over x one gets for the finite mass

$$N(t) = \int_0^L n(x,t)\, dx :$$

$$\frac{dN}{dt} = n(o,t) \cdot u(o,t) - n(L,t) \cdot u(L,t) \quad (7)$$

3. Initial and boundary conditions

a) Initial conditions:

$$n(x,o) = 1$$
$$B(x,o) = 1$$
$$T(x,o) = To = \text{const} \quad (I\ 1)$$

A Maxwellian is taken for the distribution function F:

$$F\ (x,\ w,\ o)\ =\ (2\pi T_o)^{-\frac{1}{2}} \cdot exp\left(-\frac{w^2}{2T_o}\right) \qquad \text{(I 2)}$$

From eqs. (I 1) and (I2) it immediately follows that

$$u\ (x,o)\ =\ 0$$
$$v\ (x,\ o)\ =\ 0$$
$$v\ (x,\ o) \cdot B\ (x,\ o)\ =\ 0 \qquad \text{(I 3)}$$

b) Boundary conditions:

In order that mass be conserved during the calculation, the right-hand side of eq. (7) has to vanish. For this purpose it is sufficient that

$$u\ (o,\ t)\ =\ u\ (L,\ t)\ =\ 0$$

These conditions are satisfied if the particles are subjected to elastic reflection at the walls x = 0 and x = L. We, therefore, take

$$F\ (o,\ w,\ t)\ =\ F\ (o,\ -w,\ t)$$
$$F\ (L,\ w,\ t)\ =\ F\ (L,\ -w,\ t) \qquad \text{(B 1)}$$
$$\text{for } -\infty < w < \infty$$

Furthermore, the ion velocity w has to be limited to allow numerical calculation:

$$-w_{max} \leq w \leq w_{max}$$

where $w_{max} > 0$ has, of course, to be chosen sufficiently large so that the resulting truncation error does not appreciably affect the accuracy of the calculation. We thus take

$$F\ (x,\ w_{max},\ t)\ =\ F\ (x,\ -w_{max},\ t)\ =\ 0 \qquad \text{(B 2)}$$

In addition, no mass should be present to the right of the right-hand boundary:

$$F(x, w, t) = 0 \qquad \text{for } x > L \tag{B 3}$$

As already stated, the external magnetic field should grow linearly with time, thus giving

$$B(x, t) = 1 + \alpha t, \tag{B 4}$$

α constant, non-negative
for $x > L$.

It is assumed, moreover, that no current flows at $x = 0$ (plane of symmetry), and so we have

$$v(o, t) = 0$$

or $$\frac{\partial B}{\partial x}(o, t) = 0 \tag{B 5}$$

From eq. (B 1) it follows that

$$u(o, t) = u(L, t) = 0 \tag{B 6}$$

Equations (B 3) and (B 4) yield

$$\left.\begin{aligned} n(x, t) &= 0 \\ u(x, t) &= 0 \\ v(x, t) &= 0 \end{aligned}\right\} \quad \text{for } x > L \tag{B 7}$$

4. Discretization methods

Notation:

Let

i = index in the x-direction, $i = 0, 1, \dots N$

j = index in the w-direction, $j = -M, \dots\dots M$

n = index in the t-direction, $n = 0, 1, 2\dots$

$h = \Delta t,$ $\quad n \cdot h = t,$

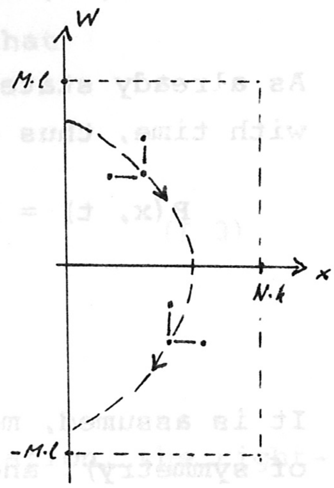
$k = \Delta x,$ $\quad i \cdot k = x,$ $\quad N \cdot k = L$

$l = \Delta w,$ $\quad j \cdot l = w,$ $\quad M \cdot l = w_{max}$

In addition, let F_{ij}^{n} be the approximation solution of F (x, w, t) and B_i^n the approximation of B (x, t) and so on.

a) Approximation of the Vlasov equation

Equation (1) is approximated by means of a positivity preserving first-order difference scheme that describes fairly well the paths of the particles (characteristics). As the accelaration $v \cdot B$ will be ≥ 0, the projections of the characteristics onto the x – w plane will qualitatively follow the path of the curve shown in the sketch. It is, therefore, reasonable to use the following approximation:



$$\frac{1}{h}\left(F_{ij}^{n+1}-F_{ij}^{n}\right) = \begin{cases} -jl\,\frac{1}{k}\left(F_{ij}^{n}-F_{i-1,j}^{n}\right)+v_i^n B_i^n \cdot \frac{1}{l}\left(F_{i,j+1}^{n}-F_{ij}^{n}\right) & \text{for } j \geq 0 \\ -jl\,\frac{1}{k}\left(F_{i+1,j}^{n}-F_{ij}^{n}\right)+v_i^n B_i^n \cdot \frac{1}{l}\left(F_{i,j+1}^{n}-F_{ij}^{n}\right) & \text{for } j < 0 \end{cases}$$

Abbreviating thus

$$\frac{j \cdot l \cdot h}{k} = a_j, \quad j = -M, \ldots, M$$

$$\frac{h}{l}\, v_i^n \cdot B_i^n = b_i^n, \quad i = 1, \ldots, N, \quad n \geq 0$$

yields

$$F_{ij}^{n+1} = \left(1 - a_j - b_i^n\right)\cdot F_{ij}^{n} + a_j F_{i-1,j}^{n} + b_i^n F_{i,j+1}^{n} \quad \text{for } j \geq 0 \qquad (8a)$$

$$F_{ij}^{n+1} = \left(1 + a_j - b_i^n\right)\cdot F_{ij}^{n} - a_j F_{i+1,j}^{n} + b_i^n \cdot F_{i,j+1}^{n} \quad \text{for } j < 0 \qquad (8b)$$

From eq. (B 1) it follows that

$$F^n_{oj} = F^n_{o,-j} \qquad j = 1, \dots M, \qquad n \geqslant 0$$

$$F^n_{N,j} = F^n_{N,-j} \qquad j = -M \dots -1, \qquad n \geqslant 0 \qquad (9)$$

Furthermore, it immediately follows from the Vlasov equation that

$F\,(o,\ o,\ t) = \text{const},$ i.e.

$$F^n_{o,o} = F^o_{o,o} \quad \text{for } n \geqslant 0 \qquad (10)$$

Because of eq. (B2) we get

$$F^n_{i,M+1} = F^n_{i,-(M+1)} = 0, \quad i = 0, \dots N, \quad n \geqslant 0 \qquad (11)$$

By means of the initial conditions (I 1), (I 2), (I 3) it is now possible to calculate with formulae (8) to (11) all values F^1_{ij} at the time $h = \Delta t$ a.s.o.

b) Detailed study of the difference equations (8a), (8b)

In order that scheme (8) be positivity preserving, the coefficients have to be non-negative. This imposes the condition

$$\max_j |a_j| + \max_i |b_i^n| \leq 1 \qquad \text{for } n \geqslant 0$$

from which it follows that the time step $h = \Delta t$ from n to n + 1 is given by the following inequality:

$$h \leq \frac{l \cdot k}{\max_i |v_i^n B_i^n| \cdot k + l^2 \cdot M} \qquad (12)$$

This is, therefore, a necessary condition on the choice of the time step to guarantee preservation of positivity.
If the inequality (12) is satisfied, it follows from (8) that

$$|F^{n+1}_{ij}| \leq (1 - |a_j| - |b_i^n|) \cdot |F^n_{i,j}| + |a_j| \cdot |F^n_{i\pm 1,j}| + |b_i^n| \cdot |F^n_{i,j+1}| \leq$$

$$\leq \max \{ |F^n_{i,j}|, |F^n_{i\pm 1,j}|, |F^n_{i,j+1}| \} .$$

This applies to every pair i,j from the lattice, except for certain boundary points. By forming maxima over all i,j it follows that

$$\| F^{n+1} \| \leq \| F^n \| \quad , \qquad n = 0, 1, 2 \ldots$$

from which it follows, in particular, that

$$\| F^n \| \leq \| F^o \| \qquad \text{for } n = 0, 1, 2 \ldots \tag{13}$$

For the linear case the inequality (13) means that the scheme (8) is globally stable. The fact that the maximum of the initial distribution can nowhere be exceeded at any time is also demanded from the physical point of view.

Another desirable requirement stipulated for the distribution function is mass conservation. The condition (B 1) is only necessary for keeping the approximation of the integral

$$N(t) = \int_0^L \int_{-\infty}^{\infty} F \, dw \, dx \qquad \text{constant.}$$

The approximation of $N(t)$ according to the trapezoidal rule is of the form:

$$\sum_{ij}{}' F_{ij}^n = k \cdot l \cdot \sum_j \left\{ \sum_{i=1}^{N-1} F_{ij}^n + \frac{1}{2} (F_{oj}^n + F_{Nj}^n) \right\} .$$

A longer calculation using the formulae (8) and the initial and boundary conditions yields

$$\sum_{ij}{}' F_{ij}^{n+1} = \sum_{ij}{}' F_{ij}^n - \frac{1}{2} b_N^n \cdot (F_{N,0}^n + F_{N,1}^n) \cdot l \cdot k$$

or

$$\sum_{ij}{}' F_{ij}^{n+1} - \sum_{ij}{}' F_{ij}^n = -l \cdot k \cdot \frac{v_N^n \cdot B_N^n}{2} \cdot (F_{N,0}^n + F_{N,1}^n) \tag{14}$$

The scheme (8) thus leads to loss of mass. This, however, can be avoided by letting the initial density drop to zero towards the boundary L, so that there the magnetic field B can be assumed to be uniform. The right-hand side of eq. (14) then vanishes.

c) Approximation of density and mean velocity

The density and mean velocity of the ions are calculated with integrals over the distribution function. They are approximated by means of the trapezoidal rule:

$$n_i^n = \ell \cdot \left(\frac{F_{iM}^n + F_{i-M}^n}{2} + \sum_{j=-M+1}^{M-1} F_{ij}^n \right) \qquad (15)$$

for i = 0, 1 ... N

$$u_i^n = \frac{\ell^2}{n_i^n} \cdot \left(\frac{M}{2} \cdot \left(F_{iM}^n - F_{i-M}^n \right) + \sum_{j=-M+1}^{M-1} j \cdot F_{ij}^n \right) \qquad (16)$$

for i = 0, 1, ... N

Formula (16) presents difficulties if the density becomes very low. This in fact happens at the outer boundary during the calculation since the gas streams to the left.

A scanning procedure is therefore enlisted in the numerical calculation. If the density drops below a certain value, n and u are set equal to zero there, the magnetic field B is assumed to be uniform, and this region is excluded from the magnetic field calculation.

d) Approximation, pressure, and temperature

These quantities are also calculated according to the trapezoidal rule. For the pressure and temperature the same approach is adopted as in c).

e) Approximation of eq. (5)

In order to calculate the approximation solutions for B_i^n, the equation $\frac{\partial B}{\partial t} = \frac{\partial}{\partial x}\left(\frac{\nu}{\varepsilon\cdot n}\cdot\frac{\partial B}{\partial x} - uB\right)$ is approximated by the Crank-Nicholson method which is semi-implicit.

This yields

$$\frac{1}{h}\cdot(B_i^{n+1}-B_i^n) = \frac{1}{k}\cdot\frac{\nu}{\varepsilon}\cdot\left\{\frac{1}{2}\cdot\frac{B_{i+1}^n-B_i^n}{n_{i+\frac{1}{2}}^n\cdot k} + \frac{1}{2}\cdot\frac{B_{i+1}^{n+1}-B_i^{n+1}}{n_{i+\frac{1}{2}}^{n+1}\cdot k} - \frac{1}{2}\cdot\frac{B_i^n-B_{i-1}^n}{n_{i-\frac{1}{2}}^n\cdot k} - \frac{1}{2}\cdot\frac{B_i^{n+1}-B_{i-1}^{n+1}}{n_{i-\frac{1}{2}}^{n+1}\cdot k}\right\}$$

$$-\frac{1}{2k}\left\{\frac{1}{2}u_{i+1}^{n+1}B_{i+1}^{n+1} + \frac{1}{2}u_{i+1}^n\cdot B_{i+1}^n - \frac{1}{2}u_{i-1}^{n+1}B_{i-1}^{n+1} - \frac{1}{2}u_{i-1}^n\cdot B_{i-1}^n\right\}$$

with

$$n_{i+\frac{1}{2}}^n = \frac{1}{2}\cdot(n_i^n + n_{i+1}^n)$$

$$n_{i-\frac{1}{2}}^n = \frac{1}{2}\cdot(n_i^n + n_{i-1}^n)$$

for $i = 1, 2 \ldots N, \quad n \geq 0$.

An approximation is still required for the left-hand boundary, i.e. $i = 0$; from $\frac{\partial B}{\partial x}(x = 0, t) = 0$ it follows that

$$\frac{\partial B}{\partial t}(x = 0) = \frac{\nu}{\varepsilon\cdot n(x = 0)}\cdot\frac{\partial^2 B}{\partial x^2}(x = 0) - B(x = 0)\cdot\frac{\partial u}{\partial x}(x = 0).$$

With $B_1 = B_{-1}$ it follows from second order that

$$\frac{\partial^2 B}{\partial x^2}(0) \approx \frac{2}{k^2}\cdot(B_1 - B_0)$$

$$\frac{\partial u}{\partial x}(0) \approx \frac{1}{2k}\cdot(4u_1 - u_2)$$

from which we get

$$\frac{1}{h}(B_0^{n+1}-B_0^n) = \frac{\nu}{\varepsilon k^2\cdot n_0^{n+1}}\cdot(B_1^{n+1}-B_0^{n+1}) + \frac{\nu}{\varepsilon k^2 n_0^n}\cdot(B_1^n-B_0^n) - \frac{4u_1^{n+1}-u_2^{n+1}}{4k}\cdot B_0^{n+1} - \frac{4u_1^n-u_2^n}{4k}\cdot B_0^n .$$

For $i = 1, \ldots N, \quad n \geq 0$ we set :

$$d_i^n = \frac{\nu\cdot h}{k^2\varepsilon\cdot(n_i^n + n_{i+1}^n)} \quad ; \quad \bar{d}_i^n = \frac{\nu\cdot h}{k^2\varepsilon\,(n_{i-1}^n + n_i^n)}$$

$$e_i^n = \frac{h}{4k}\cdot u_{i+1}^n \quad ; \quad \bar{e}_i^n = \frac{h}{4k}\,u_{i-1}^n$$

$$\alpha_i^n = -\bar{d}_i^n - \bar{e}_i^n \quad ; \quad \beta_i^n = 1 + d_i^n + \bar{d}_i^n \quad ; \quad \gamma_i^n = -d_i^n + e_i^n$$

$$\varphi_i^n = (\bar{d}_i^n + \bar{e}_i^n) \cdot B_{i-1}^n + (1 - d_i^n - \bar{d}_i^n) \cdot B_i^n + (d_i^n - e_i^n) B_{i+1}^n$$

and also

$$\beta_0^n = 1 + \frac{\nu \cdot h}{k^2 \varepsilon n_0^n} + \frac{h}{4k} (4u_1^n - u_2^n)$$

$$\gamma_0^n = -\frac{\nu \cdot h}{k^2 \varepsilon n_0^n}$$

$$\varphi_0^n = \left[1 - \frac{\nu \cdot h}{k^2 \varepsilon n_0^n} - \frac{h}{4k}(4u_1^n - u_2^n)\right] \cdot B_0^n + \frac{\nu \cdot h}{k^2 \varepsilon n_0^n} \cdot B_1^n \; .$$

The system of equations can now be written in the following form:

$$\begin{aligned} \beta_0^{n+1} \cdot B_0^{n+1} + \gamma_0^{n+1} \cdot B_1^{n+1} &= \varphi_0^n \\ \alpha_i^{n+1} \cdot B_{i-1}^{n+1} + \beta_i^{n+1} \cdot B_i^{n+1} + \gamma_i^{n+1} \cdot B_{i+1}^{n+1} &= \varphi_i^n \end{aligned} \qquad (17)$$

for i = 1, ... N, $n \geq 0$,

or in matrix form:

$$\begin{pmatrix} \beta_0^{n+1} & \gamma_0^{n+1} & & & 0 \\ \alpha_1^{n+1} & \beta_1^{n+1} & \gamma_1^{n+1} & & \\ & \ddots & \ddots & \ddots & \\ & & & & \gamma_{N-1}^{n+1} \\ 0 & & & \alpha_N^{n+1} & \beta_N^{n+1} \end{pmatrix} \begin{pmatrix} B_0^{n+1} \\ B_1^{n+1} \\ \vdots \\ \vdots \\ B_N^{n+1} \end{pmatrix} = \begin{pmatrix} \varphi_0^n \\ \varphi_1^n \\ \vdots \\ \vdots \\ \varphi_N^n - \gamma_N^{n+1} B_{N+1}^{n+1} \end{pmatrix} .$$

With a suitable choice of the step sizes h, k and the parameters ν, ε it is possible to satisfy the following conditions:

1. The matrix is diagonal dominant
2. $\beta_i^{n+1} > 0$, i = 0, ... N
3. $\alpha_i^{n+1} \leq 0$, $\gamma_i^{n+1} \leq 0$, i = 0, ... N

4. $\varphi_i^n > 0, \qquad i = 0, \ldots, N$

From these conditions it follows that the system (17) can be uniquely solved while preserving the positivity of the solution vector $(B_0^{n+1}, \ldots, B_N^{n+1})$.

This linear system is solved by the usual method:
The following substitutions are made:

$$\text{I.} \begin{cases} \lambda_0 = \beta_0^{n+1} , & \mu_0 = \gamma_0^{n+1} \cdot \lambda_0^{-1} \\ \lambda_i = \beta_i^{n+1} - \alpha_i^{n+1} \cdot \mu_{i-1} & i = 1, \ldots, N \\ \mu_i = \gamma_i^{n+1} \cdot \lambda_i^{-1} & i = 1, \ldots, N-1 \end{cases}$$

$$\text{II} \begin{cases} g_0 = \varphi_0^n \cdot \lambda_0^{-1} \\ g_i = (\varphi_i^n - \alpha_i^{n+1} \cdot g_{i-1}) \cdot \lambda_i^{-1} , & i = 1, \ldots, N-1 \\ g_N = (\varphi_N^n - \gamma_N^{n+1} \cdot B_{N+1}^{n+1} - \alpha_N^{n+1} \cdot g_{N-1}) \cdot \lambda_N^{-1} \end{cases}$$

$$\text{III} \begin{cases} B_N^{n+1} = g_N \\ B_i^{n+1} = g_i - \mu_i B_{i+1}^{n+1} , & i = N-1, \ldots, 1, 0 \end{cases} .$$

It is readily seen that under conditions 1,2,3, and 4 the solutions are $B_i^{n+1} > 0$. It is now briefly explained how the recursion formulae I, II, III are arrived at.

The system (17) is written in the form

$$A \cdot b = \varphi$$

and the following ansatz is made:

$$A = L \cdot U = \begin{pmatrix} \lambda_0 & & & 0 \\ \alpha_1^{n+1} & \lambda_1 & & \\ & \ddots & \ddots & \\ 0 & & \alpha_N^{n+1} & \lambda_N \end{pmatrix} \cdot \begin{pmatrix} 1 & \mu_0 & & 0 \\ & 1 & \ddots & \\ & & \ddots & \mu_{N-1} \\ 0 & & & 1 \end{pmatrix} .$$

Comparison of the coefficients yields formula I. The system

$$L \cdot g = \varphi$$

is then solved by recursion formulae, the result being expressed by II.

Finally, the system

$$U \cdot b = g$$

is solved, as described by formulae III.

The boundary conditions (B 4) to (B 7) can then be used to calculate the magnetic field B by the method described.

f) Approximation of the acceleration v . B

v_i^n is calculated according to eq. (4):

$$V_i^n = \frac{1}{\varepsilon \cdot n_i^n} \cdot \frac{B_{i+1}^n - B_{i-1}^n}{2k} \qquad (18)$$

for $i = 1, \ldots N; \quad n \geq 0$.

Because of the boundary condition (B 5) we have

$$v_o^n = 0 \text{ for } n \geq 0 \quad . \qquad (19)$$

With

$$b_i^n = \frac{h}{I} \cdot V_i^n \cdot B_i^n \quad \text{for } i = 0, \ldots, N \; ; \quad n \geq 0 \qquad (20)$$

We thus calculated the coefficients for the equations (8).

5) Numerical calculations

The calculation now proceeds as follows:

1. Calculation of the distribution function F by the method described in 4 a) at the time h_o, where h_o is the time step size initially chosen.

2. Calculation of the ion density n and the mean ion velocity u at the time $t = h_o$ by the method described in 4 c).

3. Calculation of the magnetic field B at the time $t = h_o$ by the Crank-Nicholson method (see 4 e)).

4. Calculation of the acceleration $v \cdot B$ at the time $t = h_o$, according to 4 f).

5. Determination of the new time step size h_1 by the scanning method (12) (preservation of positivity and stability).

On completion of these five steps the distribution function F is again calculated according to 4 a) at the time $t = h_o + h_1$, and the cycle is then repeated.

a) Data

The following fixed data were used for the calculations (see (I 1), (B 4)):

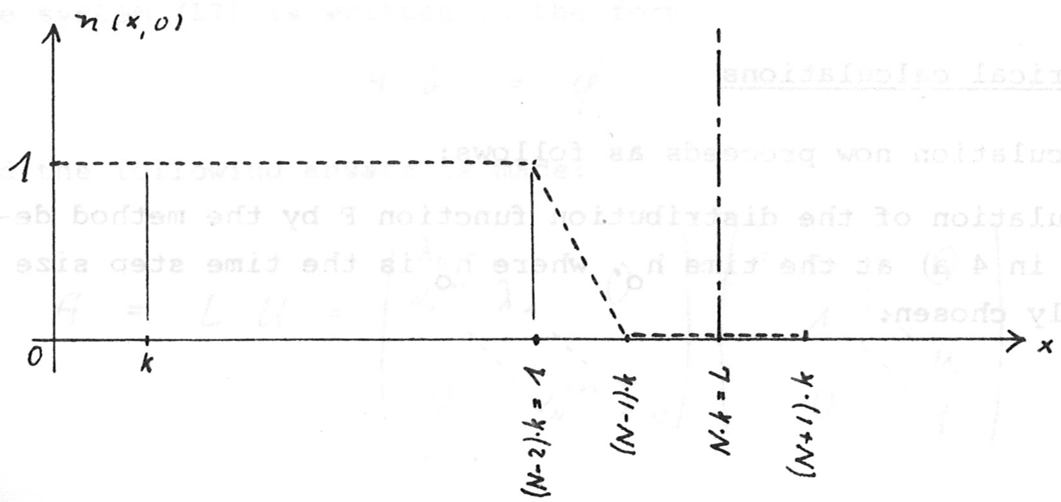
$$B_o = 1$$

$$T_o = 0{,}4$$

$$\alpha = 100$$

$$n_i^o = \begin{cases} 1 & \text{for } i = 0,1 \ldots N-2 \\ 0 & \text{for } i = N-1,\ N,\ N+1 \end{cases}$$

The initial density thus has the following profile:



Choice of step sizes:

$$w = \ell = 0{,}1, \quad M = 200$$

from which it follows that

$$M\ell = w_{max} = 20$$

An estimate then yields (see (12)):

$$F^{0}_{i,M} = F^{0}_{i,-M} = (2\pi T_0)^{-\frac{1}{2}} \cdot \exp\left(-\frac{\ell^2 M^2}{2T_0}\right) < 10^{-200} .$$

The w-region, however, is intentionally chosen so large that under certain circumstances a strong shift of the distribution function in the w-direction may occur during the calculation.

The step size $\Delta x = k$ was varied to dertermine the accuracy of the calculation (as a function of Δx). Calculations were made for

1. $k = 1/8, \quad N = 10$
2. $k = 1/18, \quad N = 20$
3. $k = 1/38, \quad N = 40$
4. $k = 1/78, \quad N = 80$.

A good degree of accuracy is abviously already obtained for $k = 1/78$.

The step size h_o and the parameters ν, ε were also varied. The choice of h_o must satisfy the condition

$$h_o \leq \frac{k}{M \cdot 1} = \frac{k}{20}$$

(see formula (12) at the time $t = 0$).

Furthermore, the conditions 1 to 4 imposed on the tridiagonal matrix of the system (17) have to be satisfied. Each time 2000 time steps were computed; the results of every hundredth time step were printed out and some of them plotted. The total job time was about 32 minutes.

b) Program

```
C HAUPTPROGRAMM ZU VLASSOWGLEICHUNG
C GARCHING,DEN 11.8.1969
      INTEGER T,TMAX
      REAL MAGN
      REAL NN,K,L
      COMMON /INDEX/ N,M,NI,MJ,KI,KJ,NIP1,NIM1,KIP1,MJM1,MJP1,KJP1,KJM1
      COMMON /CONST/ K,L,H,ACON,EPS,BE,TO,C,PI,BMAX
      COMMON /AB/ HO,REST
      COMMON/FI/F(100,405),VERT1(405,20),VERT2(405,20),VERT3(405,20)
      COMMON/BUNV/BKL(100)   ,U(100)   ,B(100)   ,NN(100)   ,V(100)   ,A(405)
      COMMON/PL/DICHTE(100,20),MAGN(100,20),GESCH(100,20),ZEIT(20),
     1          DRUCK(100,20)
       COMMON/A/ AJJ(405),AII(100)
       DIMENSION AN(100)
      READ(5,100)  IF17,IF18
       READ(5,100) KONTRO,IP,IT
      READ(5,100)NK1,NK2,NK3,NK4,NK5
      READ(5,100)KI,KJ,KT
      READ(5,100)N,M,TMAX
      READ(5,101)K,L,HO
      READ(5,101)ACON,EPS,BE,TO,C,BMAX
      READ(5,101)REST
      WRITE(6,150)KI,KJ,KT
      WRITE(6,151)N,M,TMAX
      WRITE(6,152)K,L,HO
      WRITE(6,153)ACON,EPS,BE,TO,C,BMAX,REST
      WRITE(6,154)   NK1,NK2,NK3 ,NK4,NK5
      WRITE(6,999)KONTRO,IP,IT
       NNK=NK/2
         NDIM=100
         MDIM=405
         PI=3.1415927
         NI=N+1
         NIP1=NI+1
         NIM1=NI-1
         NIM2=NI-2
         KIP1=KI+1
         MJ=2*M+2
         MJP1=MJ+1
         MJM1=MJ-1
         KJM1=KJ-1
         KJP1=KJ+1
         KTMAX=TMAX+1
         H=HO
         S=0.
      REWIND 17
      REWIND 18
      IF(IF17.EQ.0)   GO TO 32
       READ(17)     IPP,ITER,ITER1,TT,T,H,S,
     1   AJJ,AII,F,BKL,U,B,NN,A,BMAX,
     2   ZEIT,DICHTE,MAGN,GESCH,DRUCK,VERT1,VERT2,VERT3,VERT4,VERT5
      REWIND 17
         KT=T+1
      GO TO 36
   32 IF(IF18.EQ.0)   GO TO 35
       READ(18)     IPP,ITER,ITER1,TT,T,H,S,
     1   AJJ,AII,F,BKL,U,B,NN,A,BMAX,
     2   ZEIT,DICHTE,MAGN,GESCH,DRUCK,VERT1,VERT2,VERT3,VERT4,VERT5
```

```
      REWIND 18
         KT=T+1
      GO TO 36
   35 CONTINUE
         JJ=-M-1
      DO 33 J=KJ,MJ
         JJ=JJ+1
   33    AJJ(J)=JJ*L
         II=-1
      DO 34 I=KI,NI
         II=II+1
   34    AII(I)=II*K
      CALL ANF(F,BKL,B,U,NN,A,NDIM,MDIM)
      WRITE(6,170)
      CALL OUT(F     ,NI,MJ,KI  ,KJ,NDIM,MDIM,AJJ,AII)
         IPP=0
         ITER=0
         ITER1=0
         TT=0
   36 CONTINUE
      DO 1 T=KT,TMAX
         TT=TT+H
      CALL FIJ(A,F,NDIM,MDIM,BKL)
         S=S+C*H
      CALL BKLI(B,NN,U,BKL,NDIM,S,F,MDIM)
         NZEIT=T-1
      IF(NZEIT.NE.IT*ITER) GO TO 50
      WRITE(6,160) TT
      CALL OUT(F     ,NI,MJ,KI  ,KJ,NDIM,MDIM,AJJ,AII)
      WRITE(6,1010)(NN(I),I=KI,NI,8)
         SUM=0
      DO 981 I=KIP1,NIM1
  981    SUM=SUM+NN(I)
         SUM=K*((NN(KI)+NN(NI))*0.5+SUM)
      WRITE(6,1199) SUM
      WRITE(6,1011)(B(I),I=KI,NI,8)
      WRITE(6,1099)(U(I),I=KI,NI,8)
      DO 717 I=KI,NI
  717    AN(I)=BKL(I)/H*L
      WRITE (6,1999)(AN(I),I=KI,NI,8)
      DO 800 I=KI,NIM2
         AN(I)=0.5*(F(I,KJ)*(M*L+U(I))**2+F(I,MJ)*(M*L-U(I))**2)
         SSS=0
         JJ=-M
      DO 801 J=KJP1,MJM1
         JJ=JJ+1
  801    SSS=SSS+F(I,J)*(JJ*L-U(I))**2
         AN(I)=(AN(I)+SSS)*L
C        AN(I)=AN(I)/NN(I)
  800 CONTINUE
      AN(NIM1)=AN(NIM2)
      AN(NI)=AN(NIM2)
      WRITE (6,1089)(AN(I),I=KI,NI,8)
         ITER=ITER+1
      IF(IF17.EQ.0) GO TO 48
       WRITE(17)  IPP,ITER,ITER1,TT,T,H,S,
     1   AJJ,AII,F,BKL,U,B,NN,A,BMAX,
     2   ZEIT,DICHTE,MAGN,GESCH,DRUCK,VERT1,VERT2,VERT3,VERT4,VERT5
```

```
      REWIND 17
         IF17=0
      GO TO 50
   48  WRITE(18)  IPP,ITER,ITER1,TT,T,H,S,
     1   AJJ,AII,F,BKL,U,B,NN,A,BMAX,
     2   ZEIT,DICHTE,MAGN,GESCH,DRUCK,VERT1,VERT2,VERT3,VERT4,VERT5
      REWIND 18
         IF17=1
   50 CONTINUE
      IF(NZEIT.NE.IP*ITER1) GO TO 77
      IF(IPP.GE.20) GO TO 77
         IPP=IPP+1
         ZEIT(IPP)=TT
      DO 80  I=KI,NI
      DRUCK(I,IPP)=AN(I)
   80 CONTINUE
      DO 78 I=KI,NI
         DICHTE(I,IPP)=NN(I)
         MAGN(I,IPP)=B(I)
         GESCH(I,IPP)=U(I)
   78 CONTINUE
      DO 79  I=KJ,MJ
         VERT1(I,IPP)=F(NK1,I)
         VERT2(I,IPP)=F(NK2,I)
         VERT3(I,IPP)=F(NK3,I)
   79 CONTINUE
         ITER1=ITER1+1
   77 CONTINUE
    1 CONTINUE
       IF(KONTRO.EQ.0)STOP
      WRITE(6,1299) (ZEIT(I),I=1,IPP)
      CALLPLOTDM(DICHTE,MAGN,AII,KI,NI,IPP,N)
      CALL PLOTDM(DICHTE,DRUCK,AII,KI,NI,IPP,N)
      CALL VERT(VERT1,NI,MJ,KI,KJ,NDIM,MDIM,AII,AJJ,TO,IPP)
      CALL VERT(VERT2,NI,MJ,KI,KJ,NDIM,MDIM,AII,AJJ,TO,IPP)
      CALL VERT(VERT3,NI,MJ,KI,KJ,NDIM,MDIM,AII,AJJ,TO,IPP)
C FORMATE
C ********
  100 FORMAT(6I12)
  101 FORMAT(6E12.4)
  150 FORMAT(1H1,10X,4HKI =I2,10X,4HKJ =I2,10X,4HKT =I2/)
  151 FORMAT(11X,3HN =I4,10X,3HM =I4,10X,6HTMAX =I4/)
  152 FORMAT(11X,3HK =1PE12.4,5X,3HL =E12.4,5X,3HH =E12.4/)
  153 FORMAT( 2X,3HA =1PE12.4,5X,5HEPS =E12.4,5X,4HBE =E12.4,5X,4HTO =
     1      E12.4,5X,3HC =E12.4 ,3X,'BMAX='1PE12.4/
     2    ' REST='1PE12.4///)
  154 FORMAT(11X, 'NK1='I4,10X,'NK2='I4,10X,'NK3='I4,10X,'NK4=',I4,
     1   10X,'NK5='I4/)
  160 FORMAT(1H1,'T ='1PE12.4/,1X,'---------------'///)
  170 FORMAT(///,1X,'ANFANGSVERTEILUNG'/,
     1          1X,'-----------------'//)
 1010 FORMAT(///,' DICHTE'/,1P11E12.4)
 1011 FORMAT(///,' MAGNETFELD'/,1P11E12.4)
 1099 FORMAT(///' GESCHWINDIGKEIT'/,1P11E12.4)
 1999 FORMAT(///' BESCHLEUNIGUNG'/,1P11E12.4)
 1199 FORMAT(/,' GESAMTSUBSTANZ ='1PE12.4)
 1299 FORMAT(1H0,'  ZEITEN DER KURVEN'/,1P11E12.4)
  999 FORMAT(10X,' KONTR =',I3,10X,' IP =',I3,10X,' IT =',I3)
```

```
      SUBROUTINE ANF(F,BKL,B,U,NN,A,NDIM,MDIM)
      REAL NN,L,K
      COMMON /INDEX/ N,M,NI,MJ,KI,KJ,NIP1,NIM1,KIP1,MJM1,MJP1,KJP1,KJM1
      COMMON /CONST/ K,L,H,ACON,EPS,BE,TO,C,PI,BMAX
      DIMENSION BKL(NDIM),NN(NDIM)
      DIMENSION U(NDIM),A(MDIM),F(NDIM,MDIM)
      DIMENSION B(NDIM)
C        FAK1=SQRT(BE*BE-2.*EPS*TO)
         FAK2=1./SQRT(2.*PI*TO)
         FAK3=-L*L/(2.*TO)
         FAK5=L*H/K
         FAK6=BE-FAK1
         FAK7=2.*EPS*TO
         II=-1
      DO 1 I=KI,NIP1
         II=II+1
         BKL(I)=0.
         B(I)=BE
C        B(I)=FAK1
C       IF(II.GE.N-2) B(I)=B(I)+0.5 *(II-N+3)*FAK6
C       IF(II.GE.N) B(I)=BE
         NN(I)=1.
         U(I)=0.
C        NN(I)=(BE*BE-B(I)*B(I))/FAK7
    1 CONTINUE
      NN(NIM1)=0.
      NN(NI)=0.
         NN(NIP1)=0.
         JJ=-M-1
      DO 2 J=KJ,MJ
         JJ=JJ+1
         A(J)=JJ*FAK5
         FAK1=FAK2*EXP(JJ*JJ*FAK3)
         II=-1
      DO 3 I=KI,NI
         II=II+1
         F(I,J)=FAK1*NN(I)
    3 CONTINUE
    2 CONTINUE
      DO 6 J=KJM1,MJP1
    6 F(NIP1,J)=0.
      DO 7 I=KI,NIP1
         F(I,KJM1)=0.
         F(I,MJP1)=0.
    7 CONTINUE
      WRITE(6,100)
  100 FORMAT(1H1,'ANF')
      WRITE(6,101)(BKL(I),I=KI,NIP1)
      WRITE(6,101)(NN(I),I=KI,NIP1)
      WRITE(6,101)(U(I),I=KI,NIP1)
      WRITE(6,101)(B(I),I=KI,NIP1)
      WRITE(6,101)(A(J),J=KJM1,MJP1)
      WRITE(6,101)(F(KI,J),J=KJM1,MJP1)
  101 FORMAT(1P11E12.4)
      RETURN
      END
```

```
      SUBROUTINE FIJ(A,F,NDIM,MDIM,BKL)
      REAL NN,L,K
      COMMON /INDEX/ N,M,NI,MJ,KI,KJ,NIP1,NIM1,KIP1,MJM1,MJP1,KJP1,KJM1
      COMMON /CONST/ K,L,H,ACON,EPS,BE,TO,C,PI,BMAX
       DIMENSION A(MDIM),F(NDIM,MDIM),BKL(NDIM)
       JJ=-M-1
       DO 99 J=KJ,MJ
       JJ=JJ+1
       A(J)=JJ*L*H/K
   99 CONTINUE
          NULL =M+2
         MEINS=NULL-1
         NEINS=NULL+1
      DO 3 I=KI,NIM1
      IF(BKL(I).LT.0.)  GO TO 5
      DO 7 J=KJ,MEINS
         F(I,J)=F(I,J)+BKL(I)*(F(I,J+1)-F(I,J))
     1   +A(J)*(F(I,J)-F(I+1,J))
    7 CONTINUE
      GO TO 3
    5 DO 6 J=KJ,MEINS
         F(I,J)=F(I,J)+BKL(I)*(F(I,J)-F(I,J-1))
     1   +A(J)*(F(I,J)-F(I+1,J))
    6 CONTINUE
    3 CONTINUE
       DO 1 I=KIP1,NI
      IF(BKL(I).LT.0.)  GO TO 8
      DO 10 J=NULL,MJ
         F(I,J)=F(I,J)+BKL(I)*(F(I,J+1)-F(I,J))
     1   +A(J)*(F(I-1,J)-F(I,J))
   10 CONTINUE
      GO TO 1
    8 DO 9 J=NULL,MJ
         F(I,J)=F(I,J)+BKL(I)*(F(I,J)-F(I,J-1))
     1   +A(J)*(F(I-1,J)-F(I,J))
    9 CONTINUE
    1 CONTINUE
         MINJ=MEINS
      DO 2 J=NEINS,MJ
         F(NI,MINJ)=F(NI,J)
         MINJ=MINJ-1
    2 CONTINUE
         MINJ=MJ
      DO 4 J=KJ,MEINS
         F(KI,MINJ)=F(KI,J)
         MINJ=MINJ-1
    4 CONTINUE
C     F(NIM1,KJ)=F(NIM1,KJ)+0.5*BKL(NI)*(FST(NI,KJ)-FST(NI,KJP1))
C     MINJ=MEINS
C     DO 10 J=NEINS,MJ
C     F(NIM1,MINJ)=F(NIM1,MINJ)+BKL(NI)*(FST(NI,J)-FST(NI,J+1))
C     MINJ=MINJ-1
C  10 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE UINNI(F,NN,U,NDIM,MDIM)
      REAL NN,L,K
      COMMON /INDEX/ N,M,NI,MJ,KI,KJ,NIP1,NIM1,KIP1,MJM1,MJP1,KJP1,KJM1
      COMMON /CONST/ K,L,H,ACON,EPS,BE,TO,C,PI,BMAX
      COMMON /AB/ HO,REST
      DIMENSION F(NDIM,MDIM),NN(NDIM),U(NDIM)
        F1=0.5*M
        II=0
      DO 1 I=KI,NI
        II=II+1
        S1=(F(I,MJ)+F(I,KJ))*0.5
        S2=(F(I,MJ)-F(I,KJ))*F1
        JJ=-M+1
      DO 2 J=KJP1,MJM1
        S1=S1+F(I,J)
        S2=S2+JJ*F(I,J)
        JJ=JJ+1
    2 CONTINUE
        NN(I)=L*S1
    3   U(I)=L/S1*S2
    1 CONTINUE
       NN(NIP1)=0.
      RETURN
      END
```

```
      SUBROUTINE BKLI(B,NN,U,BKL,NDIM,S,F,MDIM)
     REAL NN,L,K
     COMMON /INDEX/ N,M,NI,MJ,KI,KJ,NIP1,NIM1,KIP1,MJM1,MJP1,KJP1,KJM1
     COMMON /CONST/ K,L,H,ACON,EPS,BE,TO,C,PI,BMAX
     COMMON /AB/ HO,REST
     DIMENSION PHI(1000),F(NDIM,MDIM)
      DIMENSION B(NDIM),NN(NDIM),U(NDIM),BKL(NDIM)
      DIMENSION ALF(1000),BET(1000),GAM(1000)
      KONTRO=0
      III=0
 199 IF(NN(NI).GE.REST) GO TO 100
     NI=NI-1
     NIP1=NI+1
     NIM1=NI-1
     III=III+1
     GO TO 199
 100 CONTINUE
         B(NIP1)=BE+S
         IF (B(NIP1).GT.BMAX) B(NIP1)=BMAX
         FAK1=  ACON*H/(K**2*EPS)
         FAK2=H/(4.*K)
         FAK3=1./NN(KI)
         FAK4=4.*U(KIP1)-U(KI+2)
     IF(KONTRO.EQ.0)  GO TO 3
         BET(KI)=1.+FAK2*FAK4+FAK1*FAK3
         GAM(KI)=-FAK1*FAK3
   3  IF(KONTRO.EQ.0)
    1    PHI(KI)=(1.-FAK1*FAK3-FAK2*FAK4)*B(KI)+FAK1*FAK3*B(KIP1)
         ANM=       NN(KI)+NN(KIP1)
         II=1
         U(NIP1)=0.
      DO 1 I=KIP1,NI
         ANP=       NN(I)+NN(I+1)
         DI=FAK1/ANP
         DQI=FAK1/ANM
         EI=FAK2*U(I+1)
         EQI=FAK2*U(I-1)
     IF(KONTRO.EQ.0)  GO TO 4
         ALF(I)=-DQI-EQI
         BET(I)=1.+DI+DQI
         GAM(I)=-DI+EI
   4  IF(KONTRO.EQ.0)
    1    PHI(I)=(DQI+EQI)*B(I-1)+(1.-DI-DQI)*B(I)+(DI-EI)*B(I+1)
         II=II+1
         ANM=ANP
   1 CONTINUE
     IF(KONTRO.EQ.1) GO TO 99
     CALL UINNI(F,NN,U,NDIM,MDIM)
     KONTRO=1
     GOTO 100
  99 CONTINUE
         PHI(NI)=PHI(NI)-GAM(NI)*B(NIP1)
      CALL SYSNOR(BET,ALF,GAM,B,PHI,KI,NI)
      DO 2 I=KIP1,NIM1
         VI=1./(EPS*NN(I))*(B(I+1)-B(I-1))/K *0.5
         BKL(I)=VI*B(I)
   2 CONTINUE
     VNI=1./(EPS*NN(NI))*(B(NI+1)-B(NI-1))/K*0.5
```

```
      BKL(NI)=VNI*B(NI)
      AMAX=ABS(BKL(I))
      DO 98 I=KIP1,NIM1
      IF(AMAX.GT.ABS(BKL(I+1)))GO TO 98
      AMAX=ABS(BKL(I+1))
   98 CONTINUE
      H=H0
      IF (H.GT.L*K/(K*AMAX+L*L*M)) H=L*K/(K*AMAX+L*L*M)
      FAKTOR=H/L
      DO 97 I=KIP1,NI
   97 BKL(I)=BKL(I)*FAKTOR
        IIIP1=III+1
      DO 198 I=1,IIIP1
        NIPI=NI+I
        U(NIPI)=0.
        BKL(NIPI)=0.
        B(NIPI)=B(NIP1)
  198 CONTINUE
        NI=NI+III
        NIP1=NI+1
        NIM1=NI-1
       RETURN
       END
```

```
      SUBROUTINE SYSNOR(A,B,C,X,F,KI,NI)
      DIMENSION A(1),B(1),C(1),X(1),F(1)
      DIMENSION GAM(1000),G(1000)
         KIP1=KI+1
         NIM1=NI-1
         ALF=A(KI)
         GAM(KI)=C(KI)/ALF
         G(KI)=F(KI)/ALF
      DO 1 I=KIP1,NI
         ALF=A(I)-B(I)*GAM(I-1)
         GAM(I)=C(I)/ALF
         G(I)=(F(I)-B(I)*G(I-1))/ALF
    1 CONTINUE
         X(NI)=G(NI)
      DO 2 II=KI,NIM1
         I=NIM1+KI-II
         X(I)=G(I)-GAM(I)*X(I+1)
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OUT(A,N,M,KI,KJ,NDIM,MDIM,B,C)
      DIMENSION A(NDIM,MDIM)
      DIMENSION B(MDIM),C(NDIM)
         KONTR=0
         K=1
         NANF=KI
         NEND=NANF+80
      IF(N.LE.NEND) NEND=N
    2 WRITE(6,1009)(C(I),I=NANF,NEND,8)
       WRITE(6,1010)
      DO 1 J=KJ,M,4
      WRITE(6,1007)B(J),(A(I,J),I=NANF,NEND,8)
    1 CONTINUE
      IF(NEND.EQ.N) RETURN
         K=K+1
      IF(KONTR.EQ.1) RETURN
         N1=N-K*80
      IF(N1.LE.80) KONTR=1
         NANF=NANF+11
      IF(KONTR.EQ.1) GO TO 3
         NEND=NANF+80
      WRITE(6,1008)
      GO TO 2
    3    NEND=N
      WRITE(6,1008)
      GO TO 2
 1009 FORMAT(9X,11F10.4)
 1010 FORMAT(///)
 1007 FORMAT(F7.2,3X,11F10.5)
 1008 FORMAT(1H1)
      END
```

```
      SUBROUTINE PLOTDM(DICHTE,MAGN,AII,KI,NI,IPP,N)
      REAL MAGN
      DIMENSION X(2),Y(2),D(100),B(100)
      DIMENSION DICHTE(100,20),MAGN(100,20),AII(100)
         XMAXD=0.
         XMAXM=0.
      DO 7  J=1,IPP
      DO 6 I=KI   ,NI
      IF(XMAXM.GE.MAGN(I,IPP))GO TO 5
         XMAXM=MAGN(I,IPP)
    5 IF(XMAXD.GE.DICHTE(I,IPP)) GO TO 6
         XMAXD=DICHTE(I,IPP)
    6 CONTINUE
    7 CONTINUE
      CALL FRAME(0.,0.,270.,180.)
C PLOTTEN DICHTE
      CALL SCALE(0.,0.,2.2,XMAXD)
         X(1)=0.
         X(2)=1.
         Y(1)=0.
         Y(2)=0.
      CALL PLOTL(X,Y,2)
         X(2)=0.
         Y(2)=XMAXD
      CALL PLOTL(X,Y,2)
      DO 1 K=1,IPP
      DO 2 I=KI,NI
    2    D(I)=DICHTE(I,K)
      CALL PLOTL(AII(KI),D(KI),N)
    1 CONTINUE
C PLOTTEN DES MAGNETFELDES
      CALL SCALE(-1.1,0.,1.1,XMAXM)
         X(1)=0.
         X(2)=1.
         Y(1)=0.
         Y(2)=0.
      CALL PLOTL(X,Y,2)
         X(2)=0.
         Y(2)=XMAXM
      CALL PLOTL(X,Y,2)
      DO 3 K=1,IPP
      DO 4 I=KI,NI
    4    B(I)=MAGN(I,K)
      CALL PLOTL(AII(KI),B(KI),N)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VERT(F,NI,MJ,KI,KJ,NDIM,MDIM,AII,AJJ,TO,IPP)
      DIMENSION F(MDIM,20)   ,AII(NDIM),AJJ(MDIM)
      DIMENSION ZEICH(1000),XX(2),YY(2)
         FMAX=0.4/SQRT(TO)
      CALL FRAME(AJJ(KJ),0.,AJJ(MJ),FMAX)
         XX(1)=AJJ(KJ)
         XX(2)=AJJ(MJ)
         YY(1)=0.
         YY(2)=0.
      CALL PLOTL( XX,YY,2)
         XX(1)=0.
         XX(2)=0.
         YY(2)=FMAX
      CALL PLOTL(XX,YY,2)
         M=MJ-KJ+1
      DO 5  I=1,IPP
      DO 3 J=KJ,MJ
         ZEICH(J)=F(J,I)
    3 CONTINUE
      CALL PLOTL(AJJ(KJ),ZEICH(KJ),M)
    5 CONTINUE
      RETURN
      END
```

Acknowledgements

Finally we wish to express our thanks to H. Fisser, R. Gorenflo, K. v. Hagenow and M. Kaufmann for stimulating discussions.

Furthermore, we thank Mrs. H. Wilhelm for working out the numerical program

Literature

[1] K. HAIN, in Proc. of the APS Topical Conference on Pulsed High-Density Plasmas (Los Alamos Scientific Lab., 1967) p. F1-1.

[2] I.H. ADLAM and I.E. ALLEN, Proc. Phys. Soc., 75, 640 (1960).

[3] K.W. MORTON, J. Fluid Mech., 14, 369 (1962).

[4] R. CHODURA, Phys. Fluids, 11, 400 (1968).

[5] P.L. AUER, H. HURWITZ, and R.W. KILB, Phys. Fluids, 5, 298 (1962).

[6] I.W.M. PAUL, L.S. HOLMES, M.J. PARKINSON, and I. SHEFFIELD, Nature, 208, 133 (1965).

[7] B.D. FRIED and A.Y. WONG, Phys Fluids 9, 1084 (1966).

[8] E. ISAACSON, H.B. KELLER, "Analysis of Numerical Methods", John Wiley, New York (1966)

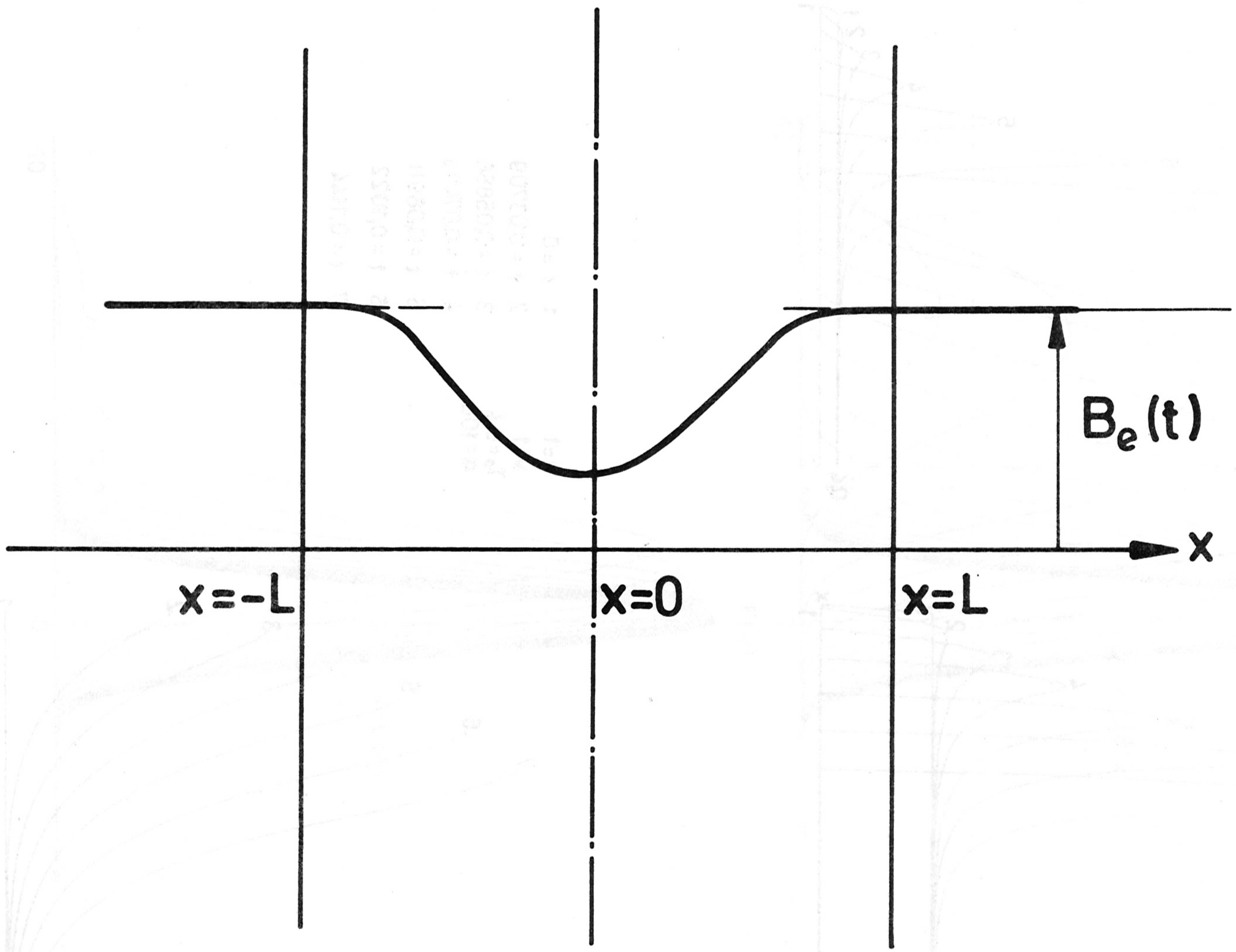


Fig. 1

Initial extent of the plasma slab and boundary value of magnetic field.

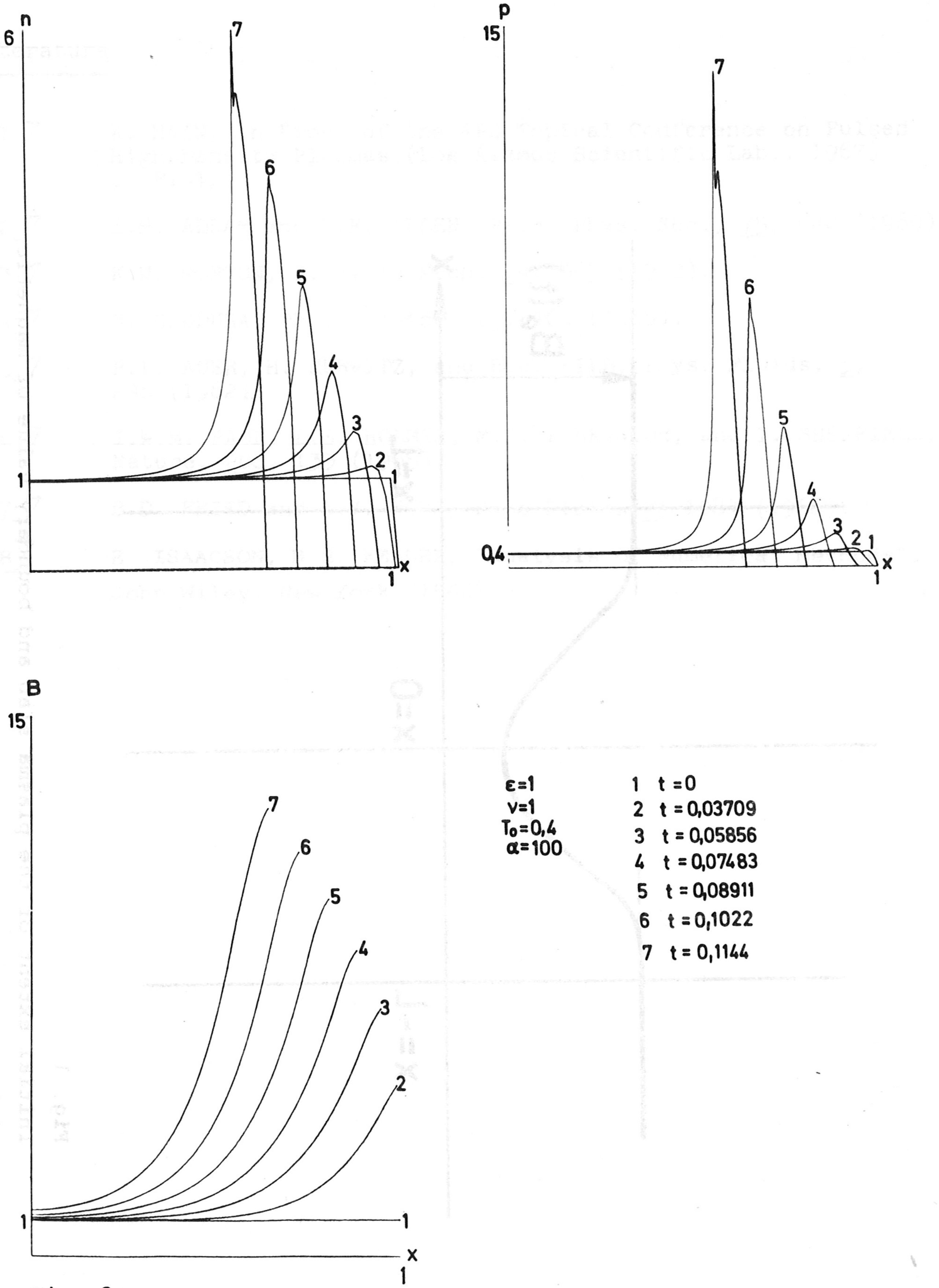


Fig. 2a

Time-behaviour of density n, ion pressure p and magnetic field B of the compression pulse in fluid description. The time is normalized by $t_o = \sqrt{M\ m/Z} \ / \ (eB_o)$, n and B by their initial values, p by $n_o ML^2/t_o^2$.

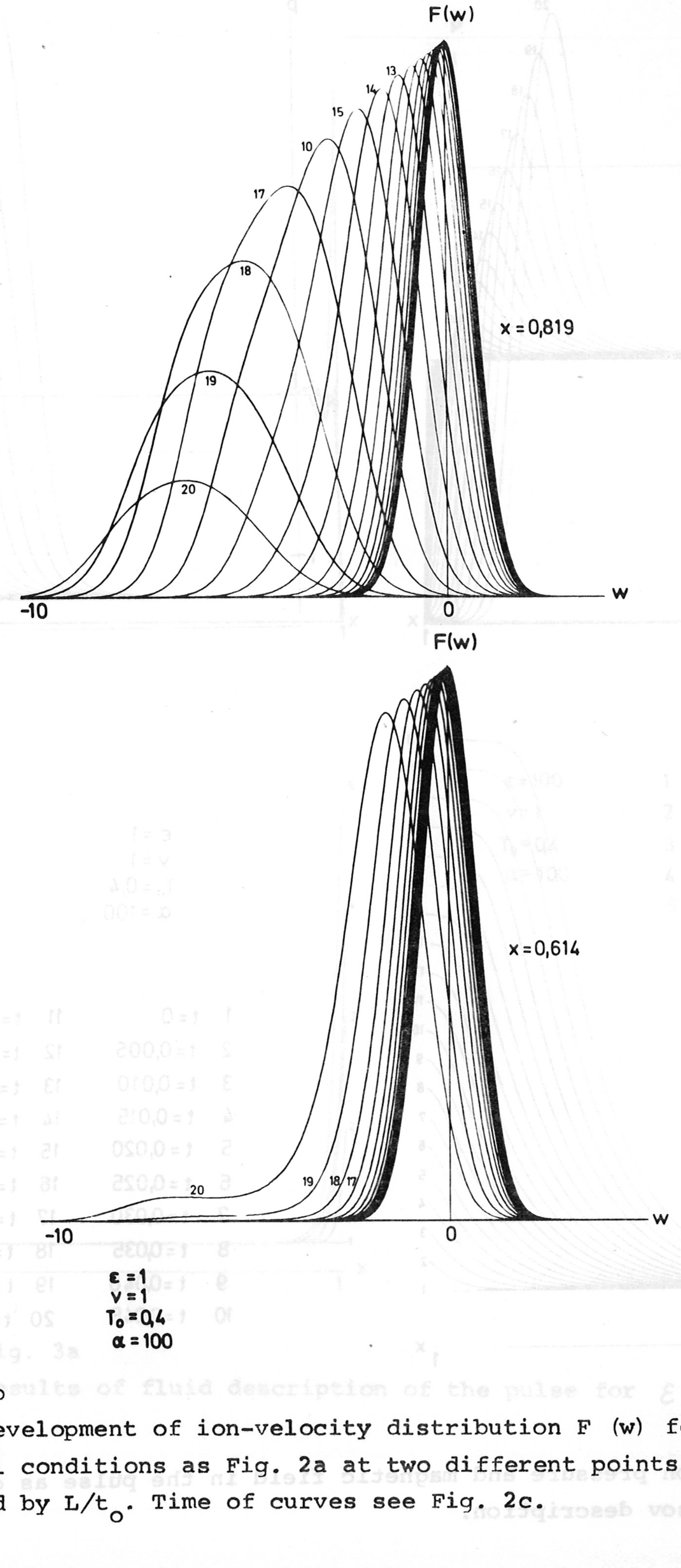


Fig. 2b

Time development of ion-velocity distribution F (w) for the same initial conditions as Fig. 2a at two different points X. w is normalized by L/t_o. Time of curves see Fig. 2c.

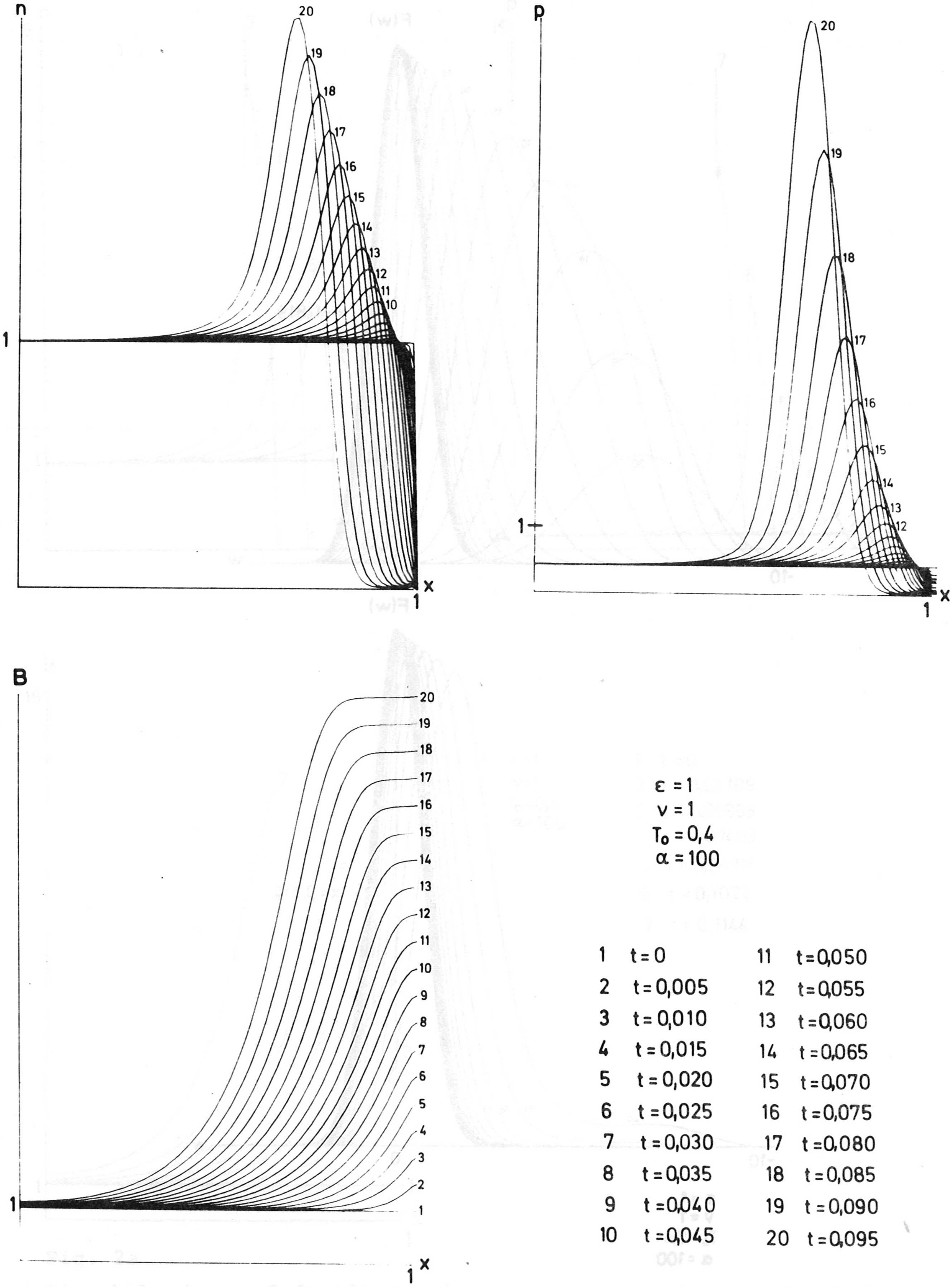


Fig. 2c
Density, ion pressure and magnetic field in the pulse as derived from a Vlasov description.

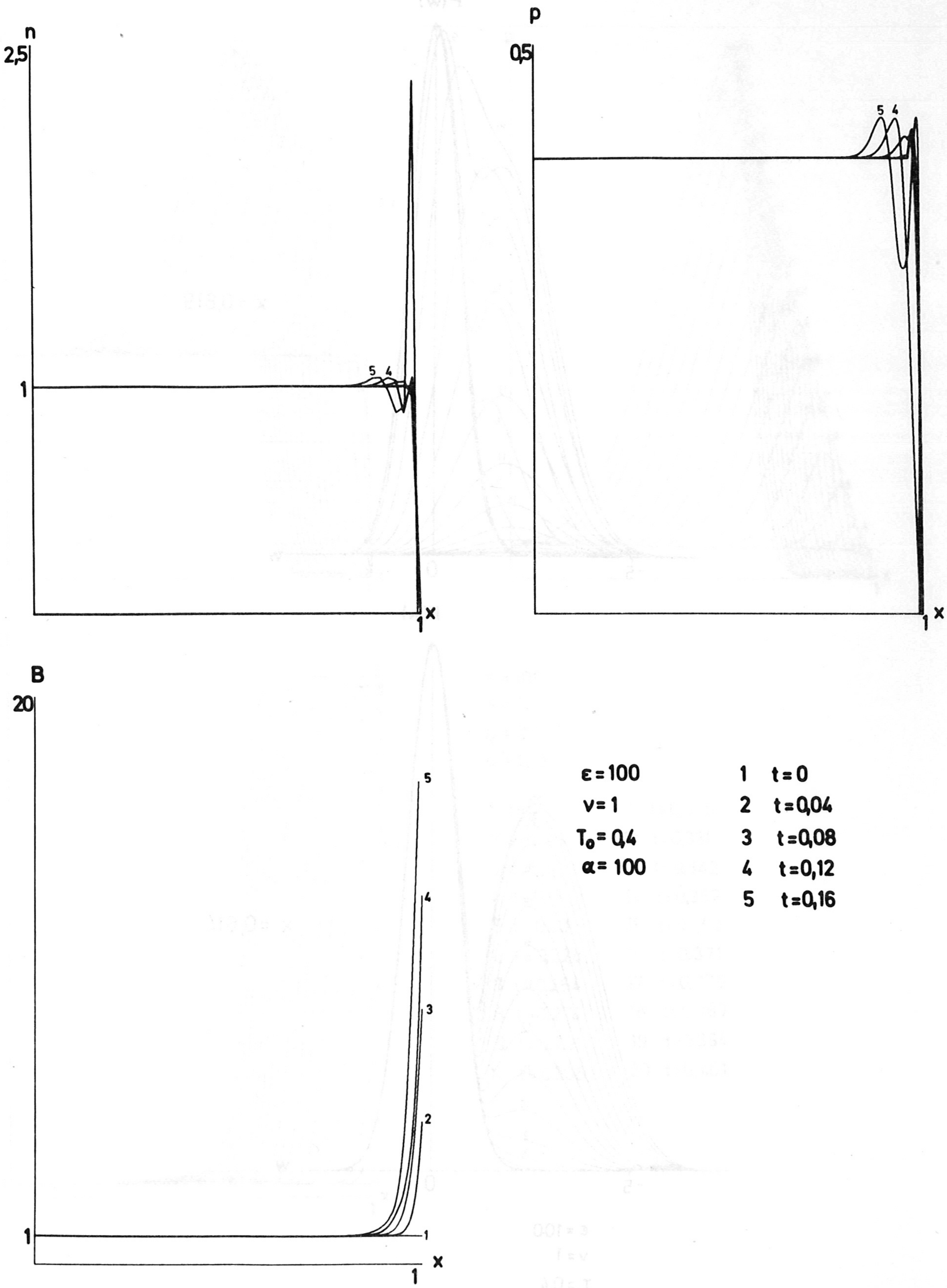


Fig. 3a

Results of fluid description of the pulse for ε = 100.

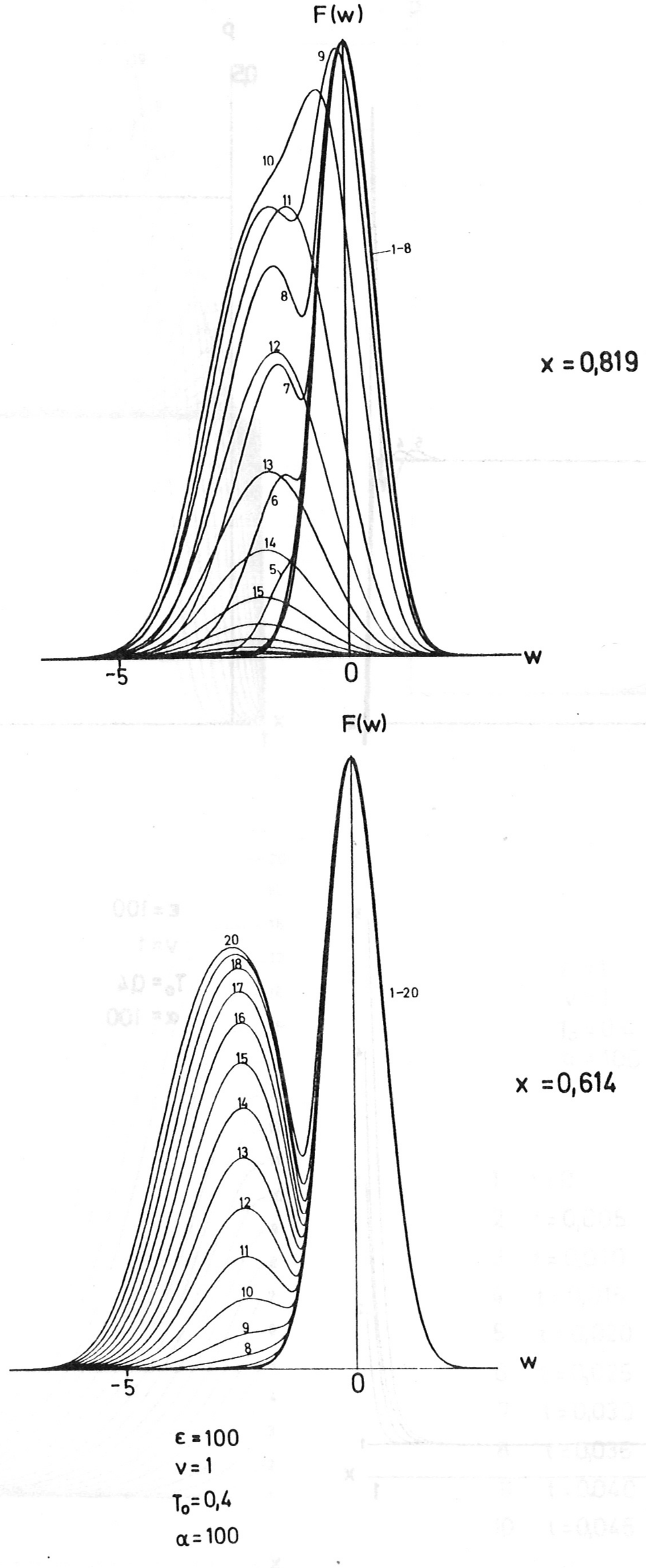


Fig. 3b

Ion-velocity destribution for $\mathcal{E}$ = 100. Time of curves see Fig. 3c.

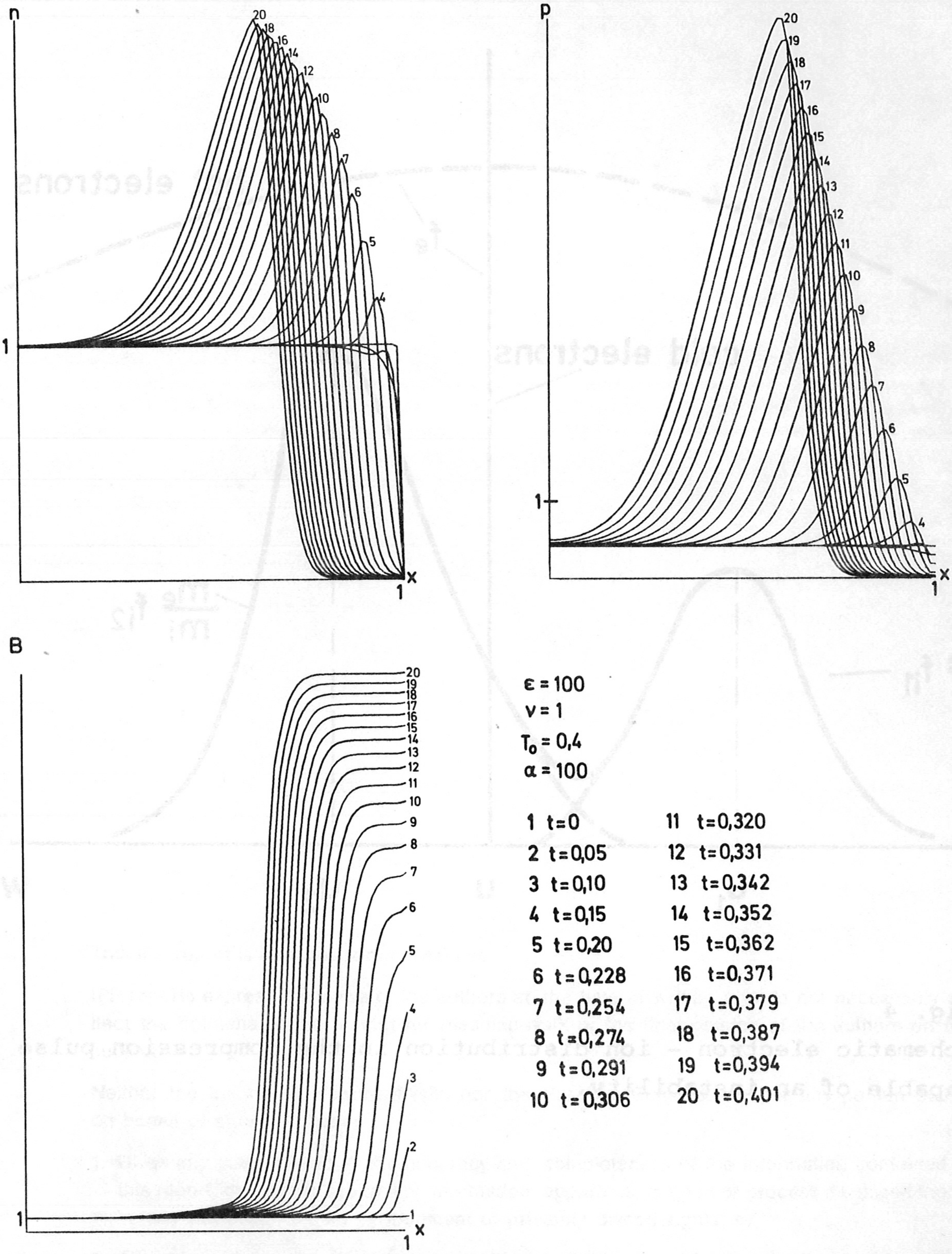


Fig. 3c
Density, ion-pressure and magnetic field from Vlasov description for ε = 100.

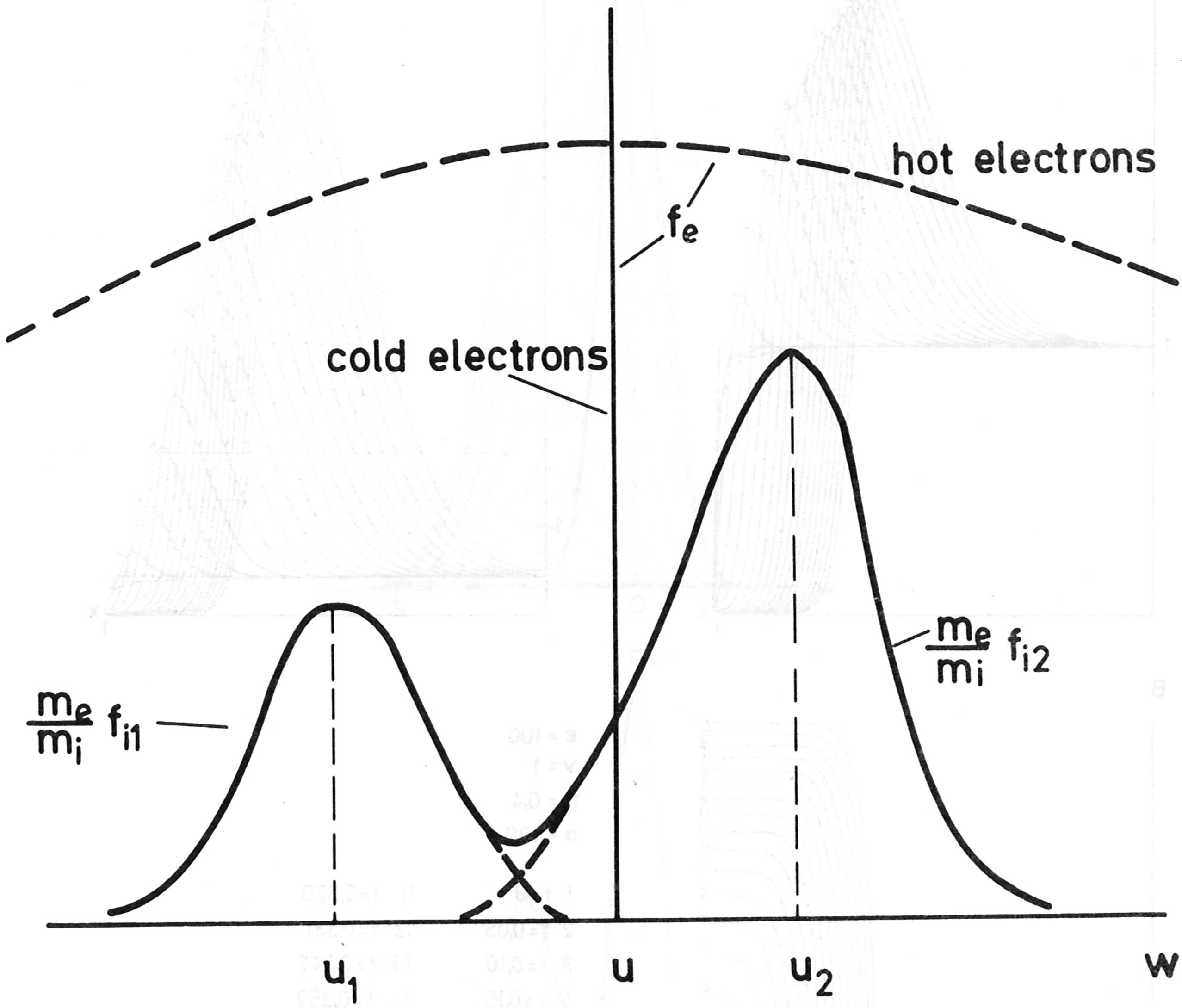


Fig. 4
Schematic electron - ion distribution in the compression pulse capable of an instability.