A NUMERICAL METHOD TO DESCRIBE UNLOADING. IN DYNAMIC PLASTICITY

C. CRISTESCU
Academy of Romanian Socialist Republic, Bucharest, Roumania

N. CRISTESCU
Institut de Mathématiques, University of Bucharest, Bucharest, Roumania

SUMMARY

The paper presents a program to be used in order to describe dynamic plastic deformation followed by elastic unloading. Two kinds of such programs are in fact presented, which correspond to two basic plastic models used in plasticity theory: the classical non-viscous model and the rate-dependent one.

The main difficulty of the problem is that in the plastic region a quasi-linear system of equations is used, while in the elastic region a linear one. The finding of the boundary between the two regions is the main objective of the program. Together with this boundary, which is obtained with the computer (for the first time in the literature), are obtained the solutions in both elastic and plastic regions. For the first time one has available a method allowing the solving of this problem for any initial or boundary conditions and any constitutive law as well.

Since in the plastic region the slopes of the characteristic lines are variable and since one has to change abruptly the slope of the characteristic line when passing from the elastic to the plastic region and vice-versa, one introduces the following (new) method: in every single point the first iteration of the solution is the local "elastic" solution in that particular point. The true solution is then obtained by iteration. The method is very efficient, few iterations being necessary in each point.

The method has been used since 1969 for the study of dynamic plastic deformation of aluminium assumed to satisfy a plastic non-viscous constitutive equation. Further on the method has been adopted for plastic rate-dependent models which may describe dynamic relaxation and creep as well.

is the so-called elastic "bar velocity." For physical reasons one always has $0 \leq c \leq c_o$.

The differential relations satisfied along these characteristic lines are

$$d\sigma = \mp \rho c(\sigma,\epsilon)\, dv + \rho c^2(\sigma,\epsilon)\Psi(\sigma,\epsilon)\, dt \qquad (15)$$

and

$$d\epsilon^E = \frac{1}{E}\, d\sigma \qquad (16)$$
$$d\epsilon^P = \Phi(\sigma,\epsilon)\, d\sigma + \Psi(\sigma,\epsilon)\, dt,$$

respectively.

It is important to observe that, although the main difficulty of the problem is the passing during integration from a quasilinear system to another semilinear and vice versa, by introducing the parameters (5) and (8) for both systems of equations the characteristics are (11) and (12) and the differential relations are (15) and (16).

Before using the previous equations in a program, they have been rewritten in dimensionless coordinates:

$$S = \frac{\sigma}{E}, \quad V = \frac{v}{c_o}, \quad T = Kt, \quad X = \frac{K}{c_o}\, x \qquad (17)$$

where K is a constant with the dimension t^{-1}. K was sometimes changed during integration in order to change the size of the characteristic mesh.

In dimensionless variables the equations of the characteristic lines are

$$\frac{dX}{dT} = \pm \frac{c}{c_o} = \pm \sqrt{\frac{1}{1+\bar{\Phi}}} \qquad (18)$$

$$dX = 0 \quad \text{(twice)}, \qquad (19)$$

while the differential relations satisfied along these lines are

$$dS = \mp \frac{c}{c_o}\, dV - \left(\frac{c}{c_o}\right)^2 \bar{\Psi}\, dT \qquad (20)$$

$$d\epsilon^P = \bar{\Phi}\, dS + \bar{\Psi}\, dT \qquad (21)$$
$$d\epsilon^E = dS$$

where the notations $\bar{\Psi} = \frac{\Psi}{K}$ and $\bar{\Phi} = E\Phi$ are also used. The upper and lower signs in (20) and (18) correspond to each other.

For system (1), a mixed boundary value problem will be solved. The "initial" data are:

$t = 0$, $a \leq x \leq b$; $v(x,0)$, $\sigma(x,0)$, $\epsilon(x,0)$ are all prescribed, (22)

while the "boundary" conditions are:

$$\left.\begin{array}{l} x = a \\ t \geq 0 \end{array}\right\}, \quad \text{prescribed either} \quad v(a,t) \text{ or } \sigma(a,t)$$
$$\qquad (23)$$
$$\left.\begin{array}{l} x = b \\ t \geq 0 \end{array}\right\}, \quad \text{prescribed either} \quad v(b,t) \text{ or } \sigma(b,t)$$

Note that the data for $x = a$, $t = 0$ and $x = b$, $t = 0$, as furnished by (22) and (23), may or may not coincide. Both possibilities have been considered in writing the program. In the first case, only smooth or acceleration waves are involved, while in the second case, the so-called shock waves are considered as well.

3. The numerical integration

3.1 Principles of the iteration method. Since it is very difficult to perform an integration with a computer using a non-regular grid of characteristic lines, and because two of the characteristic lines (eq. (18)) have variable slopes, a procedure which uses a regular grid has been employed, but in each vertex an iterative method is also necessary. The procedure is the following. First the maximum value taken by characteristic slope in the whole domain of the characteristic plane under consideration is to be found. With this slope a basic net of characteristic lines of constant slope is built. This net in turn will be used to find in each vertex the real slope of the characteristic line in that particular vertex and the first iteration of the solution at the same point.

To give a "physical" explanation of the method, the first iteration in each vertex is considered as being the "elastic" one. In other words, a regular grid of characteristic lines (18) is built with $c = c_o$ (see Fig. 1). This is obtained by making $\varkappa = 0$ (see (5)) or $\chi = 0$ (see (8)) in all equations. Therefore the grid consists of equally spaced lines

$$dX = \pm\, dT$$

with the spacing $\Delta X = \Delta T$ chosen conveniently for each specific problem and which is sometimes changed even during the integration of a single problem.

The coordinates assigned to the vertices have been established as follows: The vertical characteristics $dX = 0$ have been designated by $0\Delta X$, $1\Delta X$, $2\Delta X, \ldots, M\Delta X, \ldots, M_F\Delta X = \ell$ (see Fig. 1), while the successive characteristics of positive slope $dX = dT$ have been designated by $0\Delta T$, $2\Delta T, \ldots, 2N\Delta T, \ldots$.

Let us now consider a loop of these regular characteristic lines (Fig. 2). The generic vertex where the solution is to be found is the point T ["top" or (M, 2N+M)]. It is assumed that in the other three vertices of the loop, L ["left," or (M-1, 2N+M-1)], R ["right," or (M+1, 2N+M-1)] and B ["bottom," or (M, 2N+M-2)], the solution is already known from previous integration or boundary and initial conditions. To find the solution in T, first, with $\chi = 0$, equation (20) is integrated along (18). For this purpose, equation (20) is written in finite differences between L and T, and between R and T, respectively. Thus, S_T and V_T are obtained in the first ("elastic") approximation. Then (21) is written in finite differences and these equations are integrated between B and T to obtain in a first approximation ϵ^E and ϵ^P Always before computing ϵ^P, it is necessary to check if $S_T > S_B$. If this condition is not satisfied, then $\epsilon^P_T = \epsilon^P_B$ in the case RI when $\Psi = 0$, while in the case RD if $\Psi \neq 0$, $(21)_1$ is written as

$$\epsilon^P_T = \epsilon^P_B + 2\Psi_B \Delta T$$

These conditions express the physical requirement that always $\dot{\epsilon}^P \geq 0$

Inequalities (6) and (8) are now checked. To do that, in the memory of the computer are kept the maximum value of S for any X including the value S_B just under consideration for a certain characteristic loop. If both $\Phi = 0$ and $\Psi = 0$, then the first approximation of the solution just obtained is "the solution" in that particular top of the loop and the results are printed. If $\Psi \neq 0$ or $\Phi \neq 0$, or both, then additional iterations are necessary.

If $\Phi \neq 0$ and $\Psi \neq 0$, then Φ and Ψ at T are computed with the first approximation just obtained for S and ϵ. Therefore, starting from T the characteristics (18) are drawn backwards (dotted lines in Fig. 2) up to the intersections with sides of the initial mesh in the points BL (bottom left) and BR (bottom right). After finding the coordinates of BL and BR by linear interpolation between B and L and between B and R, the values of all required unknown functions in the point BL and BR are found. The interpolation formulae used are of the form

$$S^{(i)}_{BL} = \frac{1}{\Delta T}\left\{(X^{(i)}_{BL} - (M-1)\Delta T)\, S_B + (M\Delta T - X^{(i)}_{BL})\, S_L\right\} \qquad (24)$$

symmetric, at B and T the values of all required functions are equal on the two sides of the interface $X = 0$. Besides this, by symmetry, it is assumed that $S_R = S_L$, $\epsilon_R = \epsilon_L$, etc., and $V_R = -V_L$, since $V_B = V_T = 0$. However, in the physical problem considered, the initial condition for the specimen are, in fact, (27), i.e., the specimen is at rest, while the hitter is initially moving to the right with the velocity v_{max}. Thus, besides the symmetry conditions assumed for S, ϵ, etc., the velocity condition used is:

$$V_L = 2V_B - V_R \tag{31}$$

Thus, the half loop from Fig. 3 is completed with the symmetric other half (dotted lines). The values of all required functions in L are obtained from those in R by the symmetry law just described. From now on the procedure follows that of the main program.

The subroutine just described is used as long as $S > 0$. If at a certain time $S_B \leq 0$ for physical reasons one makes $S_B = 0$ and the subroutine at $x = 0$ is again changed: $S_T = 0$, $\epsilon^P_T = \epsilon^P_B$, while V is obtained from $(20)_2$ which for this case, when $\Psi = 0$ and $c = c_o$ becomes simply $V_T = V_R - S_R$. When $S = 0$, one has $\Phi = \Psi = 0$, and no iterations are necessary. The moment when for the first time $S = 0$ is called "time of contact" and is denoted by T_c. The computations for $t \geq T_c$ are continued as long as it is considered necessary.

Thus, all three conditions (30) have been written in the form of three subroutines. The passing from $(30)_1$ to $(30)_2$ is arbitrary and depends on the choosing of t_m, while the passing from $(30)_2$ to $(30)_3$ is done automatically by the computer when $S = 0$.

The subroutine for the boundary $x = \ell$ is very similar to the subroutine for $x = 0$ when $S = 0$. Now the procedure is very simple since at $x = \ell$, $S = \epsilon^E = \epsilon^P = 0$, and the only quantity to be computed is V. This is done with a formula of type $(20)_1$.

The boundary conditions (30) just described have been chosen since for this case a great deal of experimental data were available. It is not difficult, however, to contemplate the writing of some other subroutines which may correspond to some other kind of boundary conditions, if necessary.

4.2 Flowchart. A scheme of the flowchart for the RD case is given below (Fig. 4). Though the flowchart is self-explanatory, some additional details will be given.

The computation starts with the vertex $N = 1$, $M = 0$. The subroutine used in connection with the boundary condition $(30)_1$ is denoted by II, i.e., for the boundary conditions used in the first stage of the loading process.

The computations are then continued at successive vertices lying on the same characteristic $N = 1$, as long as $M < M_F$. Now the main body of the program is used, which was denoted by I, and was described schematiclly above in § 3.

When $M = M_F$, that is, the boundary $x = \ell$ is reached, where the boundary conditions are (29), the subroutine IV is used. This subroutine is quite similar to II and therefore will not be described further here.

At this stage the computations on the next characteristic line ($N = N + 1$) are started, and so on. A maximum number of characteristic lines N_{max} for the problem considered is appreciated in order to stop the calculations if something happens to be wrong in the computations.

Now, the role of the characteristic N is replaced in the memory of the computer by $N+1$, i.e., the operation is

DO $\quad S_B = S_T, \; V_B = V_T, \ldots, \Psi_B = \Psi_T$

for all required unknown functions. Thus, the data on two successive characteristics only are kept in the memory of the computer.

At this stage the condition $T > T_M$ is checked, where T_M corresponds to t_m from $(30)_1$, which as already mentioned is arbitrarily chosen in connection with a certain particular boundary condition. If $T < T_M$ the subroutine II is continued. If however, $T > T_M$, one checks to see if $S_B > 0$. If $S_B > 0$, then subroutine III is applied corresponding to $(30)_2$ and which was explained in § 4.1.

When for the first time at $X = 0$, $S_T = 0$ is obtained (if already $S_T < 0$ though $S_B > 0$, one makes $S_T = 0$), the corresponding time is denoted by T_c and another subroutine IV_2 corresponding to $(30)_3$ is used. Starting from this moment, the computation will be carried out on a certain additional number N_1 of positive characteristic lines after which the computations are stopped when $N_1 > N^*$. The number N^* is chosen conveniently from physical reasons. It is to be observed that the problem may be considered finished when at $X = 0$, $T = T_c$ and the computations are continued just to get the variation of various required functions after $T = T_c$.

The mesh size generally chosen was $\Delta X = 5 \times 10^{-4}$. If, however, the boundary conditions $(30)_1$ prescribe a very fast variation of $V(0,t)$, then in the neighborhood of the point $X=0$, $T=0$, the mesh size was gradually reduced up to $\Delta X = 10^{-5}$.

4.3 Remarks concerning the RI case. In tackling the RI case (i.e., when $\Psi = 0$ everywhere) it was possible, due to the peculiarity of the problem to economize on computing time, by finding the solution by analytic formulas in the lower part of the strip of Fig. 1, under the straight line

$$X = -(T - 2M_F \Delta T). \qquad (32)$$

In other words, for any boundary conditions prescribed at $X = 0$, the solution along (32) is found. These data are then used as "initial conditions" for the computations to be done above this line (see Cristescu [3]). Some details concerning the main algorithm and the flowchart will be given below.

The main subroutine is in some respect close to the one described in connection with Fig. 2. The first "elastic" approximation is computed as before. Then the inequalities (5) are checked. For this purpose for all cross sections X considered, $S_{max}(X)$ is kept in the memory of the computer. If one of the inequalities $(5)_2$ is satisfied, then the first iteration just obtained is the final solution for the particular vertex under consideration. If, however, $(5)_1$ is satisfied, then successive iterations, which generally follow those described above in connection with Fig. 2, are necessary.

The flowchart is given below (Fig. 5). Now the computations have started from the vertex $N = 0$, $M = M_F$ (see Fig. 1). The computations are made on successive characteristics of positive slope, starting from the vertex lying on the straight line (32). In the flowchart, V denotes the subroutine by which the solution along (32) is obtained by elementary analytic formulae. Subroutine IV is used for the boundary conditions (29), subrouting IV_2 is used in conjunction with the boundary condition $(30)_3$, and subroutine III corresponds to $(30)_2$. Other details of the flowchart are self-explanatory.

5. Examples

Various examples for both RI and RD cases have been considered (see Cristescu [3], Cristescu and Bell [1], Cristescu [2]). The results of the computations have been compared with experimental data, the agreement being generally very good. Further, a few examples will be given. The aim is to show the way in which the previously described program has been applied in a specific problem. The problem considered was the longitudinal impact of two identical aluminum bars, i.e., the plastic dynamic deformation of those bars, including the elastic unloading process.

For the RI cases, it has been assumed that the stress-strain relation (3) for the considered material (aluminum) can be written in the form

$$\begin{aligned} \sigma &= E\varepsilon && \text{if} \quad \sigma \le \sigma_Y \\ \sigma &= \beta(\varepsilon + \varepsilon_o)^{\frac{1}{2}} && \text{if} \quad \sigma \ge \sigma_Y \end{aligned} \tag{33}$$

where E, σ_Y, β and ε_o are material constants.

For the RD cases, the following expressions for the coefficients entering in $(1)_3$ have been chosen. In the coefficient $\Psi(\sigma,\varepsilon)$ as defined by (6), H was assumed a linear function with

$$f(\varepsilon) = \begin{cases} \sigma_Y & \text{if} \quad \varepsilon \le \varepsilon_Y \\ \sigma_Y + \frac{\beta}{2}\,\varepsilon_z^{-\frac{1}{2}}(\varepsilon - \varepsilon_Y) & \text{if} \quad \varepsilon_Y \le \varepsilon \le \varepsilon_z \\ \beta\varepsilon^{\frac{1}{2}} & \text{if} \quad \varepsilon_z \le \varepsilon \end{cases} \tag{34}$$

with

$$\varepsilon_z = \left(\frac{\beta\varepsilon_Y}{\sigma_Y - \sqrt{\sigma_Y^2 - \varepsilon_Y \beta^2}} \right)^2$$

and ε_Y, ε_z are material constants. For $k(\varepsilon)$ the following formulae were used:

$$k(\varepsilon) = \begin{cases} k_1 & \text{if} \quad \varepsilon \le \varepsilon_1 \\ k_2 + \dfrac{k_1 - k_2}{\varepsilon_1 - \varepsilon_2}(\varepsilon - \varepsilon_2) & \text{if} \quad \varepsilon_1 \le \varepsilon \le \varepsilon_2 \\ k_2 & \text{if} \quad \varepsilon_2 \le \varepsilon \end{cases} \tag{35}$$

where k_1, k_2, ε_1, ε_2 are experimentally found constants.

The coefficient $\Phi(\sigma,\varepsilon)$, as defined by (7), has been chosen in the form

$$\Phi(\sigma,\varepsilon) = \chi \left\{ \frac{3\left[\varepsilon - \frac{\sigma}{E} - \underset{\sim}{\varepsilon} - \left(\frac{a}{3E}\right)^{3/2}\right]^{2/3}}{a} - \frac{1}{E} \right\} \tag{36}$$

with $a = m + n\sqrt{\varepsilon}$ and $\underset{\sim}{\varepsilon}$, m, n constants.

As a result of the computation, in all the considered vertices in the characteristic plane, the values of σ, ε^E, ε^P, v, Φ, Ψ, σ_{max} have been obtained. Besides that, for all vertices the number of necessary iterations have been registered (generally three iterations were enough). At each exit for WRITE a flag has been placed in order to mention the subroutine and the section of the subroutine which was used for that particular vertex. This has proved to be useful for an immediate global checking of the obtained solution and for the tracing of the boundaries in the characteristic planes.

In the computations, approximately 140,000 vertices have been used. However, the results have been printed at every fifth vertex situated on every fifth characteristic line. With an IBM-360 model 65 the computation time for one case was about twelve minutes.

The results have been plotted on various graphs, showing the variation either in time or space of different functions. First in Fig. 6a and Fig. 6b are shown various significant domains in two characteristic planes, one corresponding to an RI case (using the formulae (33), (5)) and one to an RD case (using the formulae (34), (35), (36), (8)). Thus, for the RD case, for instance, one has

in L: $\Phi > 0$ and $\Psi > 0$

in R: $\Phi = 0$ and $\Psi > 0$

in U: $\Phi = \Psi = 0$

All these boundaries have been traced simultaneously with the finding of the solution. It is to be remembered that in plasticity theory, the main difficulty in solving any problem is to find the boundary between the loading and unloading domains.

In Fig. 7a and Fig. 7b are shown the variations in time of σ at various cross sections along the bar for an RI case (formulae (33), (5)) and RD (formulae (34), (35), (36), (8)), respectively. The two solutions obtained for the same initial and boundary conditions are completely distinct for cross sections close to $x = 0$ for small t, but are somehow close for large t.

Finally, in Fig. 8a and Fig. 8b are given the variations in time of the function v at various cross sections of the bar for RI and RD solutions. Again the two solutions are distinct mainly at cross sections close to the impacted end and for short times.

For more details concerning various solutions obtained, see Cristescu [2,3,4]. Shown there is an excellent agreement in the comparison of experimental data with computed solutions.

The computations have been stopped soon after the time of contact has been reached. No experimental data were available for $t > T_c$ so that the program was prepared to describe the phenomena which may occur for $t < T_c$ only.

Finally, in preparing the program, some of the subroutines have been constructed so that it would be quite easy to generalize them for the two-dimensional case.

REFERENCES

[1] CRISTESCU, N., BELL, J. F., "On unloading in the symmetrical impact of two aluminum bars," Inelastic Behavior of Solids, McGraw-Hill Book Co., pp. 397-419 (1970).

[2] CRISTESCU, N., "A procedure for determining the constitutive equations for materials exhibiting both time-dependent and time-independent plasticity," Int. J. Solids Structures 8, 511 (1972).

[3] CRISTESCU, N., "The unloading in symmetric longitudinal impact of two elastic-plastic bars," Int. J. Mech. Sci. 12, 723 (1970).

[4] CRISTESCU, N., "Rate-type constitutive equations in dynamic plasticity," Problems of Plasticity, Vol. II, Nordhoff International Publishing, 1973 (in press).

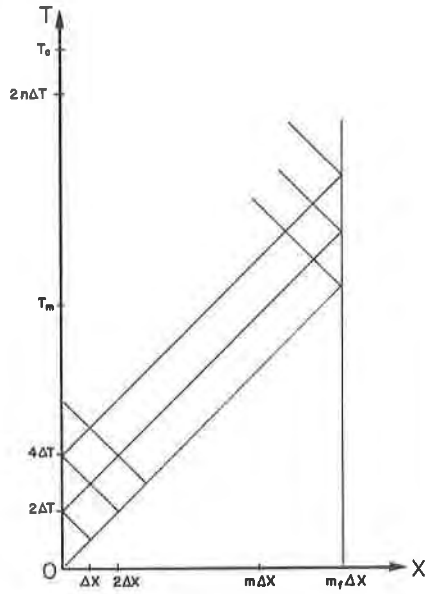
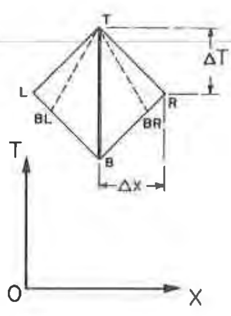


Fig. 1. Characteristic plane showing the spacing of the basic grid of characteristic lines.



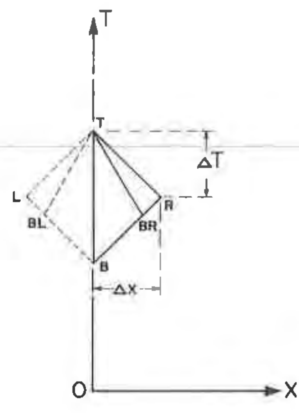


Fig. 2. Loop of regularly spaced characteristic lines of constant slope (full lines) and to approximate positions of variable slope characteristic lines.

Fig. 3. Half characteristic loop used in the subroutine at the end of the body.

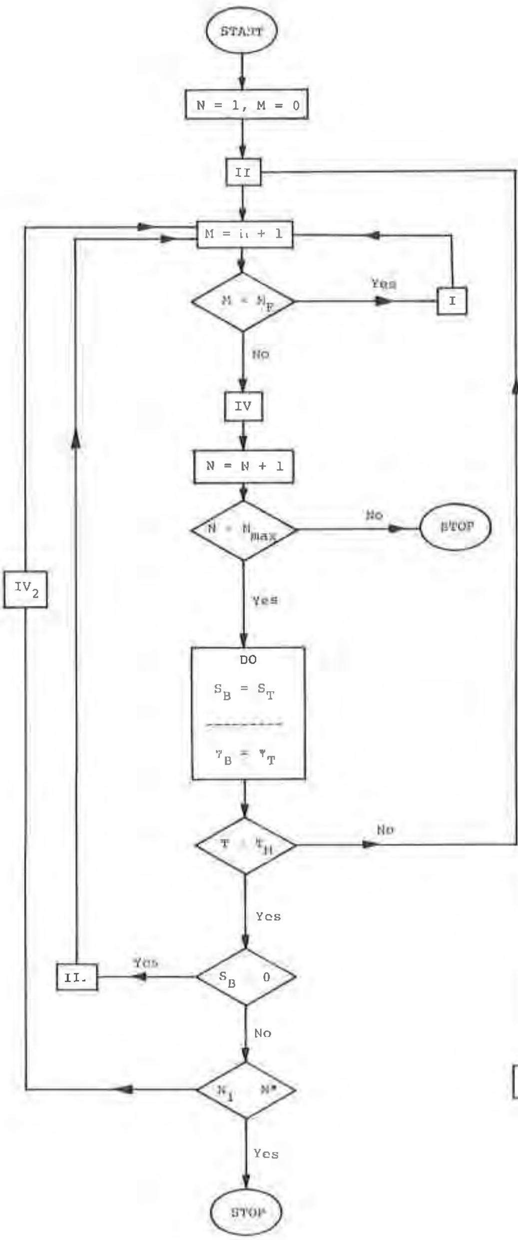


Fig. 4. Schematic flowchart used for the RD cases.

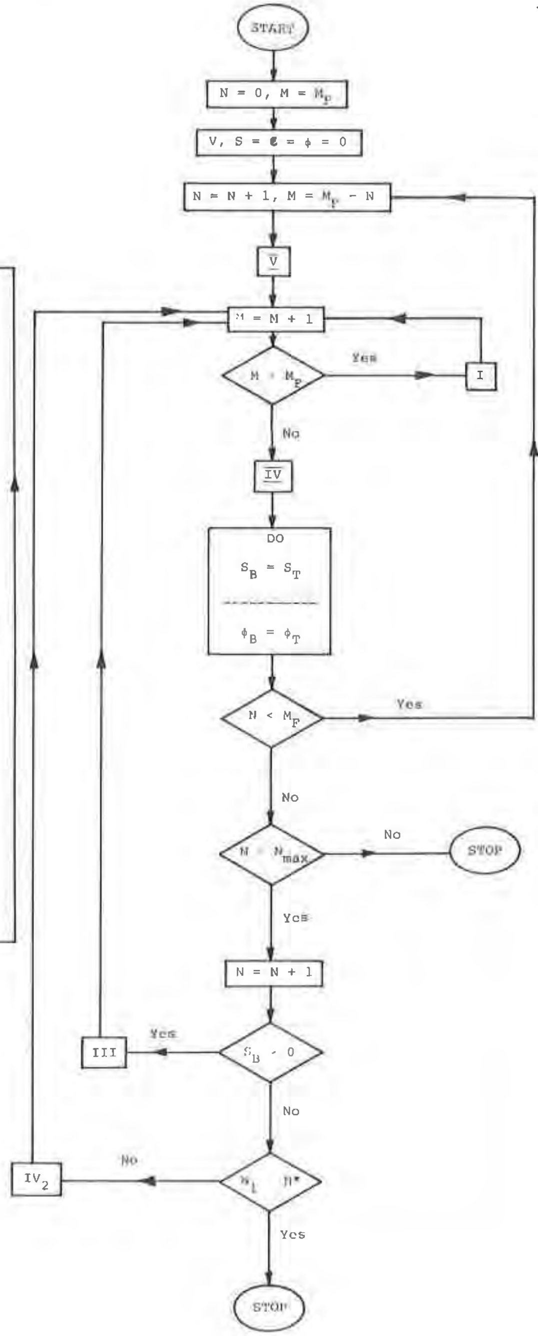


Fig. 5. Schematic flowchart used for the RI cases.

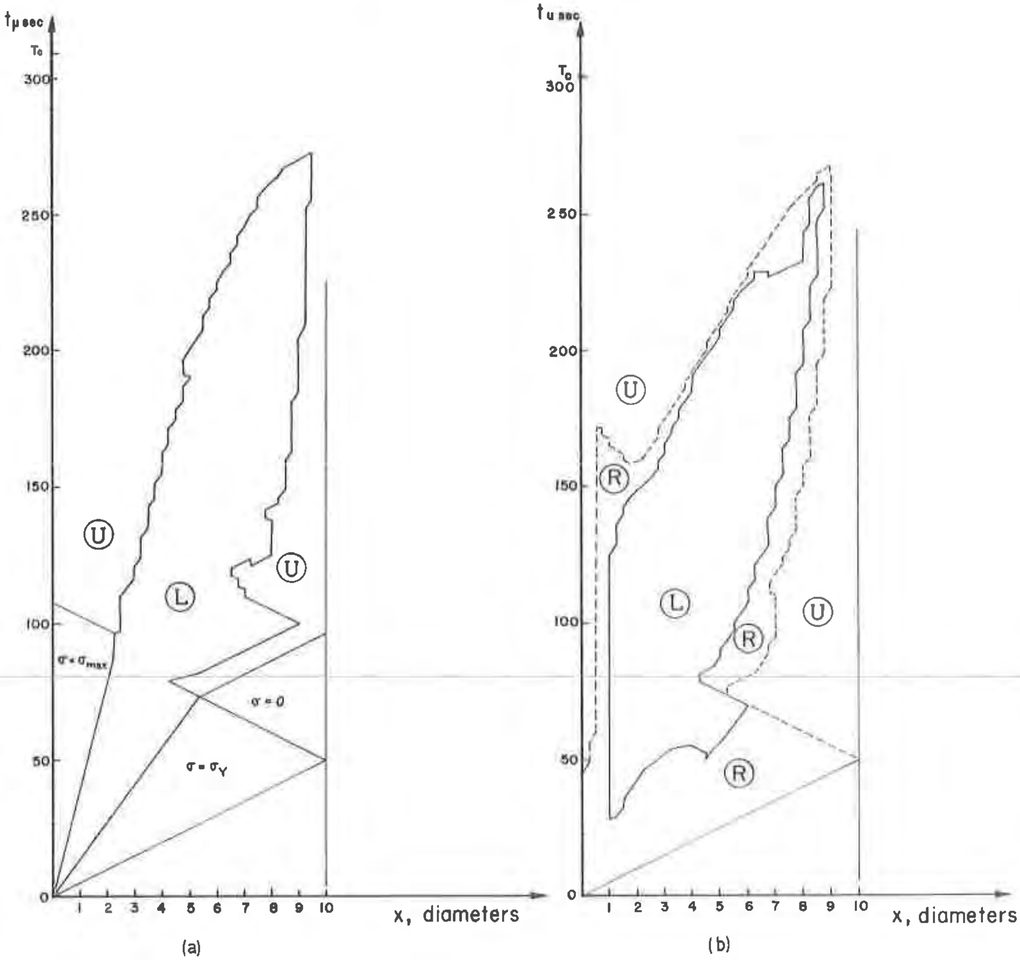


Fig. 6. (a) Characteristic plane for an RI case showing the computer obtained L/U boundary. (b) Characteristic plane for an RD case showing the L/R and R/U boundaries obtained with the computer.

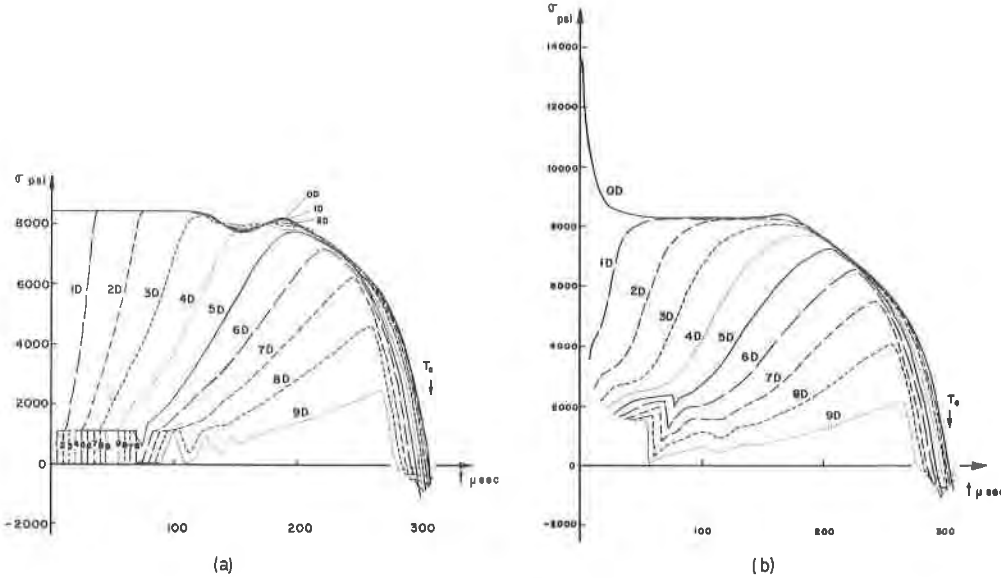


Fig. 7. Variation in time of σ at various cross sections along the bar: (a) an RI case, (b) an RD case.

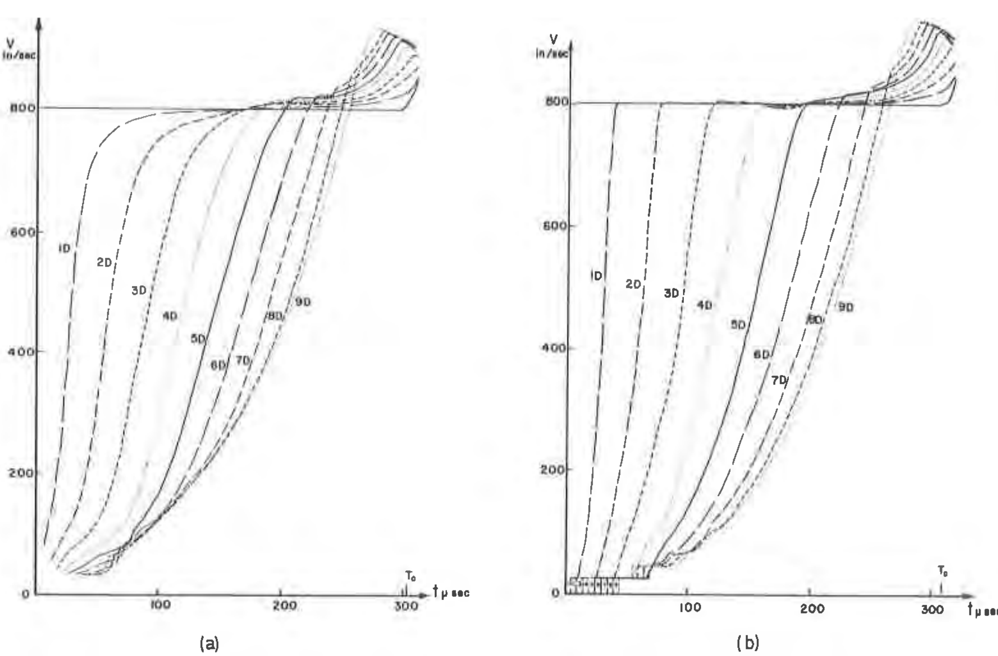


Fig. 8. Variation in time of v at various cross sections along the bar: (a) an RI case, (b) an RD case.