WIS-CS-73-171
Computer Sciences Department
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

Received: January 29, 1973

AN ARITHMETIC, PARTICLE THEORY OF FLUID DYNAMICS

by

Donald Greenspan

Appendix: FORTRAN Program for Discrete Fluids
by S. T. Jones

Technical Report #171

March 1973

AN ARITHMETIC, PARTICLE THEORY OF FLUID DYNAMICS

by

Donald Greenspan

1. INTRODUCTION

It is characteristic in fluid model theory to separate flows into two categories, laminar and turbulent (see, e.g., refs. [3,4,6,10, 14-16] and the numerous additional references contained therein.) Laminar flows are usually studied by means of deterministic, continuous models, and, though a modicum of successful mathematical analysis has emerged, there is still, nevertheless, no analytical or numerical method for solving the full Navier-Stokes equations for arbitrary Reynolds number [3,16]. Turbulent flows are usually studied by means of probabilistic models and, except for homogeneous formulations, have resisted all mathematical analysis [15,16]. Unfortunately, our level of understanding seems even further diminished by the observation that most flows in nature and in technological devices are turbulent [5, p. 83] and inhomogeneous [15, p. 492].

In this paper we will initiate a new, deterministic approach to the theory of fluid behavior which will be of such generality that both laminar and turbulent flows will correspond merely to different degrees of flow irregularity. The model will be developed in the spirit recommended by von Neumann [11] and will be computer oriented in that (a) a fluid will be thought of as consisting of a finite number of particles, (b) the motions of these particles will be determined by dynamical difference equations, or, in effect, by recursion formulas, and (c) the difference equations will be solvable in a constructive fashion by means of modern digital computers. Moreover, not only will this high-speed arithmetic approach be simpler and more comprehensive than the specialized methods developed by Lax [7] and Popov and Samarskii [13], who neglect both viscosity and heat conduction, but it too will be conservative.

2. n-Particle Fluid Dynamics

For positive time step Δt, let $t_k = k\Delta t$, $k = 0,1,2,\ldots$. At time t_k, let particle P_i of mass m_i be located at $\vec{x}_{i,k} = (x_{i,k}, y_{i,k}, z_{i,k})$, have velocity $\vec{v}_{i,k} = (v_{i,k,x}, v_{i,k,y}, v_{i,k,z})$, and have acceleration $\vec{a}_{i,k} = (a_{i,k,x}, a_{i,k,y}, a_{i,k,z})$, for $i = 1,2,\ldots,n$. Position, velocity, and acceleration are assumed to be related by the fundamental formulas [2]:

$$(2.1) \qquad (\vec{v}_{i,k+1} + \vec{v}_{i,k})/2 = (\vec{x}_{i,k+1} - \vec{x}_{i,k}) \,/\, (\Delta t)$$

$$(2.2) \qquad \vec{a}_{i,k} = (\vec{v}_{i,k+1} - \vec{v}_{i,k}) \,/\, (\Delta t) .$$

If $\vec{F}_{i,k} = (F_{i,k,x}, F_{i,k,y}, F_{i,k,z})$ is the force acting on P_i at time t_k, then force and acceleration are assumed to be related by the discrete equation

$$(2.3) \qquad \vec{F}_{i,k} = m_i \vec{a}_{i,k} .$$

In particular, in addition to gravity, we will structure the force between each pair of particles to simulate classical fluid molecular forces, so that there will be a component of attraction, which behaves like $p/(r^{\alpha})$, and a component of repulsion, which behaves like $q/(r^{\beta})$, where p, q, α, and β are nonnegative parameters which depend on the particles, and where r is the distance between the particles. Thus, if $r_{ij,k}$ is the distance between P_i and P_j at time t_k, while p_{ij}, q_{ij} are constants determined by the pair of particles under consideration, we define $\vec{F}_{i,k}$, the force exerted on P_i by gravity and by the remaining particles P_j, $j = 1,2,\ldots,i-2,i-1,i+1,i+2,\ldots,n$, of the system is

$$(2.4)\qquad \vec{F}_{i,k} = -\vec{\gamma} m_i g + m_i \sum_{\substack{j=1\\ j\neq i}}^{n} \left\{ m_j \left(-\frac{p_{ij} \sum_{\xi=0}^{\alpha-2} (r_{ij,k}^{\xi}\, r_{ij,k+1}^{\alpha-\xi-2})}{r_{ij,k}^{\alpha-1}\, r_{ij,k+1}^{\alpha-1}\, (r_{ij,k+1}+r_{ij,k})} + \frac{q_{ij} \sum_{\xi=0}^{\beta-2} (r_{ij,k}^{\xi}\, r_{ij,k+1}^{\beta-\xi-2})}{r_{ij,k}^{\beta-1}\, r_{ij,k+1}^{\beta-1}\, (r_{ij,k+1}+r_{ij,k})} \right) \left(\vec{x}_{i,k+1} + \vec{x}_{i,k} - \vec{x}_{j,k+1} - \vec{x}_{j,k} \right) \right\},$$

where $\vec{\gamma} = (0,0,1)$.

In place of approximations like the Navier-Stokes equations [4], our dynamical equations will be the primitives (2.1)-(2.4).

Let us show now that the arithmetic formulation (2.1)-(2.4) is energy conserving. To do this, define the quantity W by

$$(2.5)\qquad W = \sum_{i=1}^{n} W_i,$$

where

$$(2.6)\qquad W_i = \sum_{k=0}^{N-1} [(\vec{x}_{i,k+1} - \vec{x}_{i,k}) \cdot \vec{F}_{i,k}] .$$

Then, (2.1)-(2.3) and (2.6) imply

$$W_i = m_i \sum_{k=0}^{N-1} [(x_{i,k+1} - x_{i,k})a_{i,k,x} + (y_{i,k+1} - y_{i,k})a_{i,k,y}$$

$$+ (z_{i,k+1} - z_{i,k})a_{i,k,z}]$$

$$= \frac{1}{2} m_i \sum_{k=0}^{n-1} [(v^2_{i,k+1,x} - v^2_{i,k,x}) + (v^2_{i,k+1,y} - v^2_{i,k,y})$$

$$+ (v^2_{i,k+1,z} - v^2_{i,k,z})]$$

$$= \frac{1}{2} m_i (v^2_{i,N,x} + v^2_{i,N,y} + v^2_{i,N,z}) - \frac{1}{2} m_i (v^2_{i,0,x} + v^2_{i,0,y} + v^2_{i,0,z}).$$

If kinetic energy $K_{i,k}$ of P_i at t_k is defined by

$$K_{i,k} = \frac{1}{2} m_i (v^2_{i,k,x} + v^2_{i,k,y} + v^2_{i,k,z}) ,$$

then

$$W_i = K_{i,N} - K_{i,0} . \tag{2.7}$$

If then the kinetic energy K_k of the system at t_k is defined by

$$K_k = \sum_{i=1}^{n} K_{i,k} , \tag{2.8}$$

then (2.5), (2.7) and (2.8) yield

$$W = K_N - K_0 , \tag{2.9}$$

which, interestingly enough, is independent of the particular structure of the forces involved.

For (2.4), in particular, set

$$\vec{F}_{i,k} = \vec{F}^{(1)}_{i,k} + \vec{F}^{(2)}_{i,k} \,, \tag{2.10}$$

where

$$\vec{F}^{(1)}_{i,k} = -\,\vec{\gamma}\, m_i g \tag{2.11}$$

$$\vec{F}^{(2)}_{i,k} = m_i \sum_{\substack{j=1\\ j\neq i}}^{n} \left\{ m_j \left(- \frac{p_{ij} \sum\limits_{\xi=0}^{\alpha-2} (r^{\xi}_{ij,k}\, r^{\alpha-\xi-2}_{ij,k+1})}{r^{\alpha-1}_{ij,k}\, r^{\alpha-1}_{ij,k+1}\, (r_{ij,k+1} + r_{ij,k})} + \frac{q_{ij} \sum\limits_{\xi=0}^{\beta-2} (r^{\xi}_{ij,k}\, r^{\beta-\xi-2}_{ij,k+1})}{r^{\beta-1}_{ij,k}\, r^{\beta-1}_{ij,k+1}\, (r_{ij,k+1} + r_{ij,k})} \right) \left(\vec{x}_{i,k+1} + \vec{x}_{i,k} - \vec{x}_{j,k+1} - \vec{x}_{j,k} \right) \right\}. \tag{2.12}$$

Also, define $W^{(1)}_i$ and $W^{(2)}_i$ by

$$W^{(j)}_i = \sum_{k=0}^{N-1} [(\vec{x}_{i,k+1} - \vec{x}_{i,k}) \cdot \vec{F}^{(j)}_{i,k}], \quad j = 1,2. \tag{2.13}$$

Then,

$$W^{(1)}_i = - \sum_{k=0}^{N-1} (z_{i,k+1} - z_{i,k})\, m_i g = - m_i g z_{i,N} + m_i g z_{i,0}.$$

If $V^{(1)}_k$, the potential energy of the system at t_k due to gravity is defined by

$$(2.14) \qquad V_k^{(1)} = \sum_{i=1}^{n} m_i g z_{i,k} ,$$

then

$$(2.15) \qquad \sum_{i=1}^{n} W_i^{(1)} = - V_N^{(1)} + V_0^{(1)} .$$

In considering $W_i^{(2)}$, we note first that [2]

$$(2.16) \quad \sum_{\substack{i,j=1 \\ i<j}}^{n} \left\{ m_i m_j \left(\frac{r_{ij,k+1}^{\alpha-1} - r_{ij,k}^{\alpha-1}}{r_{ij,k}^{\alpha-1} \, r_{ij,k+1}^{\alpha-1}} \right) \right\} \equiv$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} \sum_{i=1}^{n} \left[\frac{m_i m_j \sum\limits_{\xi=0}^{\alpha-2} (r_{ij,k}^{\xi} \, r_{ij,k+1}^{\alpha-\xi-2})}{r_{ij,k}^{\alpha-1} \, r_{ij,k+1}^{\alpha-2} (r_{ij,k+1} + r_{ij,k})} \left(\vec{x}_{i,k+1} + \vec{x}_{i,k} - \vec{x}_{j,k+1} - \vec{x}_{j,k}\right) \cdot \left(\vec{x}_{i,k+1} - \vec{x}_{i,k}\right) \right] .$$

Hence, (2.12), (2.13) and (2.16) imply

$$\sum_{i=1}^{n} W_i^{(2)} = \sum_{k=0}^{N-1} \sum_{\substack{i,j=1 \\ i<j}}^{n} \left\{ m_i m_j \left(- p_{ij} \frac{r_{ij,k+1}^{\alpha-1} - r_{ij,k}^{\alpha-1}}{r_{ij,k}^{\alpha-1} \, r_{ij,k+1}^{\alpha-1}} + q_{ij} \frac{r_{ij,k+1}^{\beta-1} - r_{ij,k}^{\beta-1}}{r_{ij,k}^{\beta-1} \, r_{ij,k+1}^{\beta-1}} \right) \right\}$$

$$= \sum_{\substack{i,j=1 \\ i<j}}^{n} \left\{ m_i m_j \left[-p_{ij} \left(\frac{1}{r_{ij,0}^{\alpha-1}} - \frac{1}{r_{ij,N}^{\alpha-1}} \right) + q_{ij} \left(\frac{1}{r_{ij,0}^{\beta-1}} - \frac{1}{r_{ij,N}^{\beta-1}} \right) \right] \right\} .$$

Defining the potential energy component $V_{ij,k}^{(2)}$ of the pair P_i and P_j

at time t_k by

$$(2.17) \qquad V^{(2)}_{ij,k} = \left(- \frac{p_{ij}}{r^{\alpha-1}_{ij,k}} + \frac{q_{ij}}{r^{\beta-1}_{ij,k}} \right) m_i \, m_j$$

then yields

$$\sum_{i=1}^{n} W^{(2)}_i = \sum_{\substack{i,j=1 \\ i<j}}^{n} V^{(2)}_{ij,0} - \sum_{\substack{i,j=1 \\ i<j}}^{n} V_{ij,N} \; .$$

Defining the potential energy component $V^{(2)}_k$ of the system at t_k by

$$V^{(2)}_k = \sum_{\substack{i,j=1 \\ i<j}}^{n} V^{(2)}_{ij,k}$$

then yields

$$\sum_{i=1}^{n} W^{(2)}_i = V^{(2)}_0 - V^{(2)}_N \; .$$

Finally, defining the system potential energy V_k at t_k by

$$V_k = V^{(1)}_k + V^{(2)}_k$$

yields

$$\begin{aligned} W &= \sum_{i=1}^{n} [W^{(1)}_i + W^{(2)}_i] \\ &= - V^{(1)}_N + V^{(1)}_0 - V^{(2)}_N + V^{(2)}_0 \; , \end{aligned}$$

so that

$$W = -V_N + V_0 \quad . \tag{2.18}$$

Thus, (2.9) and (2.18) yield

$$K_N + V_N = K_0 + V_0, \; N = 0,1,2,\ldots ,$$

which is the classical law of conservation of energy.

3. Examples

We show now, by some simple examples, how fluid flow determined by (2.1)-(2.4) and given initial conditions can be affected radically simply by varying a velocity parameter.

Consider, for economy, a two-dimensional fluid in motion, a small portion of which is shown in Figure 3.1. Let particles $P_1 - P_{11}$ be called the first row, $P_{12} - P_{23}$ the second row, and $P_{24} - P_{34}$ the third row. Let $p_{ij} \equiv q_{ij} \equiv m_i \equiv 1$, $i = 1, 2, \ldots, 34$, and let us simulate a discrete Lennard-Jones type potential by setting $\alpha = 7$, $\beta = 10$. The initial positions of $P_1 - P_{34}$ are set so that $P_{13} - P_{22}$ are centers of regular hexagons of radii $r = \sqrt[3]{1.5}$, while the remaining particles are centered of the vertices of the hexagons. This choice of r is one of relative configuration stability, since, by (2.12), a force $\vec{F}^{(2)}_{i,0} = \vec{0}$ exerted by a single particle P_j on P_i implies $r_{ij,0} = \sqrt[3]{1.5}$.

Now, for fixed positive constants V and J, assume that the initial velocity of each P_i is given by

$$v_{i,0,x} = V + \varepsilon_{i,1}, \quad v_{i,0,y} = \varepsilon_{i,2},$$

where $\varepsilon_{i,1}$ and $\varepsilon_{i,2}$ are random numbers, independent of V and J, which satisfy

$$|\varepsilon_{i,j}| \leq (J\%)V = \frac{JV}{100}.$$

Fixing the parameters J and V then determines the motions of $P_1 - P_{34}$ in the following recursive fashion. For fixed k, $\vec{a}_{i,k}$ is determined from (2.3) and (2.4); $\vec{x}_{i,k+1}$ and $\vec{v}_{i,k+1}$ are then determined by solving the system (2.1) - (2.2) by Newton's method.

In the examples which follow, our interest will center on increasing values of V and on initial time steps only. For these reasons, we will

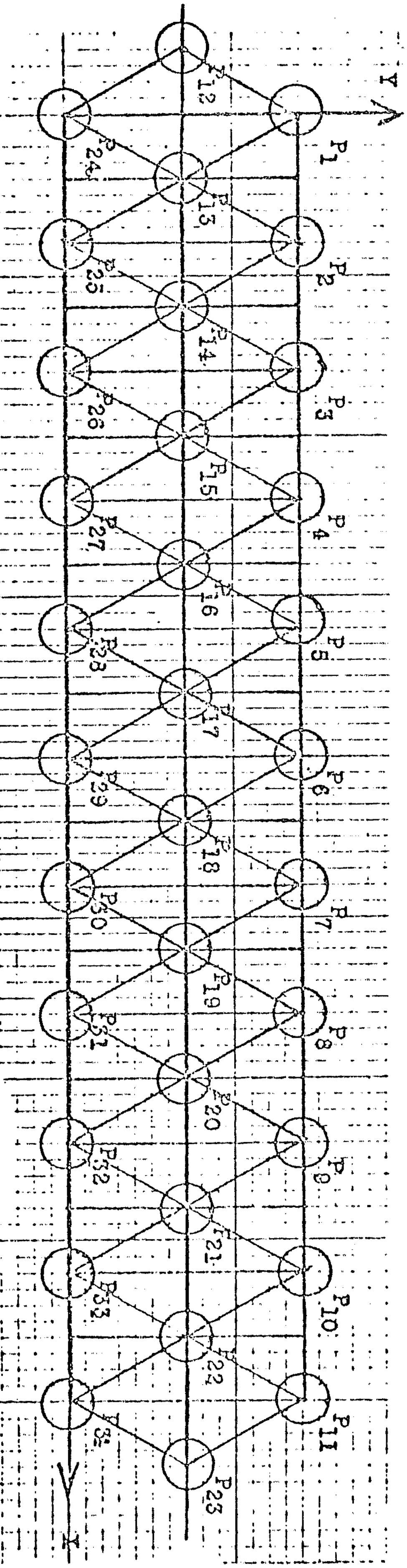
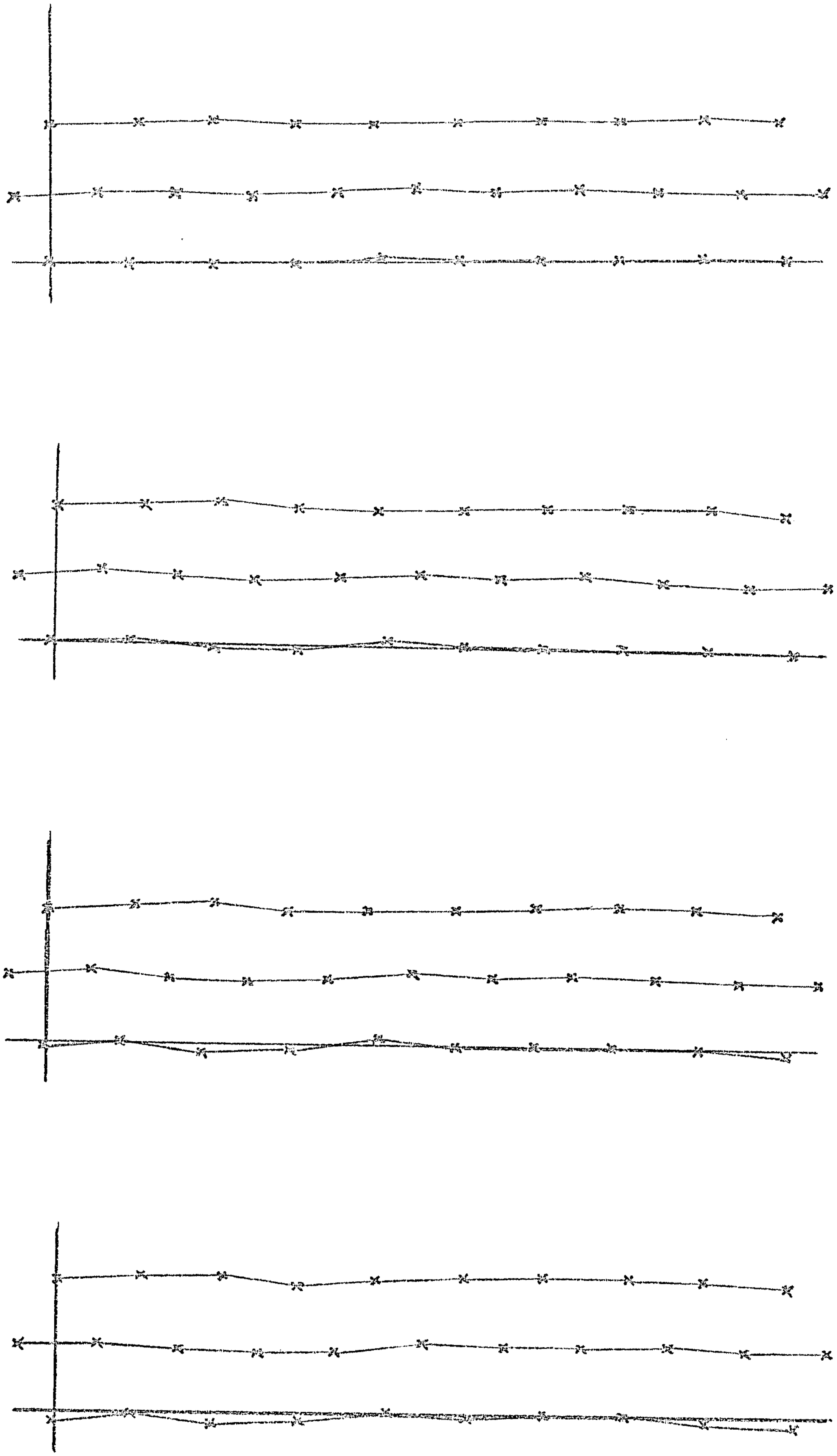

Y
X
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_{10}
P_{11}
P_{12}
P_{13}
P_{14}
P_{15}
P_{16}
P_{17}
P_{18}
P_{19}
P_{20}
P_{21}
P_{22}
P_{23}
P_{24}
P_{25}
P_{26}
P_{27}
P_{28}
P_{29}
P_{30}
P_{31}
P_{32}
P_{33}
P_{34}

FIGURE 3.1

neglect the effect of gravity by setting $g = 0$ in (2.4). Also, only one set of random numbers is generated for the velocity components of each particle. These are normalized for each example so that the maximum $|\varepsilon_{ij}|$ always gives equality in (3.1) with $J = 1$, which is also used throughout. Finally, one can view the examples as the portion of a gas emitted from a jet when the particle velocities, upon emission, are perturbed $\varepsilon_{i,j}$ by various factors, like collision with a wall, previous to emission.

Figure 3.2 shows the particle's motion for $V = 50$ and $\Delta t = 0.02$ at $t = 0.2$, 0.4, 0.6, 0.8, 1.0, 1.2 and 1.4. A gentle wave motion develops in each row while the rows maintain their relative positions. The flow is essentially of a classical laminar nature. Figure 3.3 shows the motion for $V = 300$ and $\Delta t = 0.02$ at $t = 0.2$, 0.4, 0.6, 0.8, and 1.0. Repulsion between the particles has assumed a greater significance and, though the rows still maintain their relative positions, the motion is becoming more chaotic. Figure 3.4 shows this same motion from a direction field point of view and exhibits quite clearly the strong effect of repulsion. For example, following particle pairs like P_1 and P_{11}, or P_2 and P_{12}, or P_{17} and P_{26} through the time steps shows almost complete reversals of motion due to repulsion. The rotational effects evident in Figure 3.3 also become more reasonable when viewed from this direction field point of view. Figure 3.5 shows the motion for $V = 1000$ with $\Delta t = 0.1$ at $t = 0.2$, 0.4, 0.6, 0.8 and 1.0. So much motion results that the choice $\Delta t = 0.1$ was necessary for the convergence of Newton's method in solving (2.1) - (2.2). Here, the laminar character of the flow has disappeared in that the rows no longer maintain their relative positions, and the motion becomes relatively chaotic. Thus, with the increase in velocity, particles can come nearer to other particles, which results in increased repulsive forces and more complex motion.



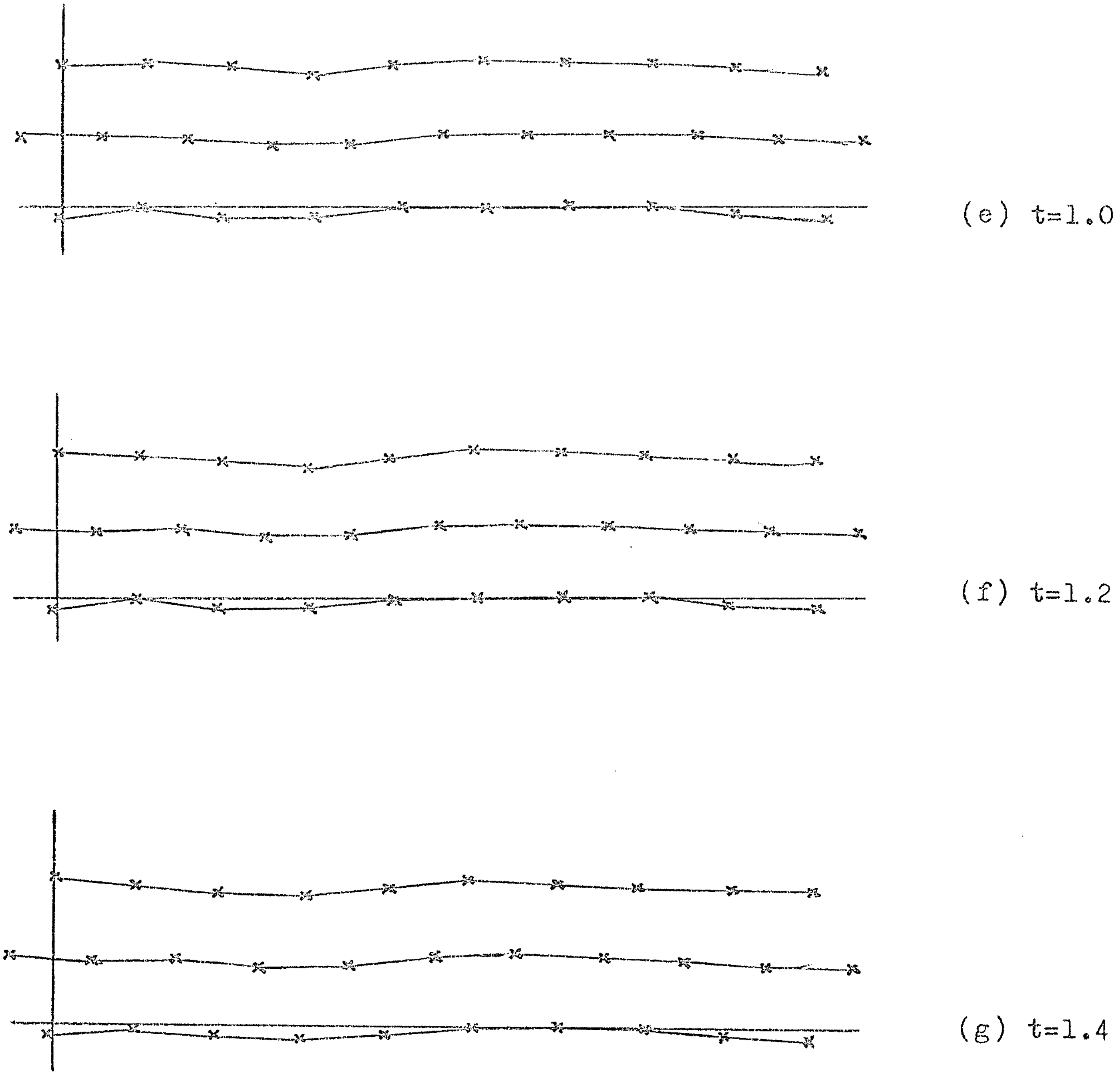
(a) t=0.2

(b) t=0.4

(c) t=0.6

(d) t=0.8

Figure 3.2

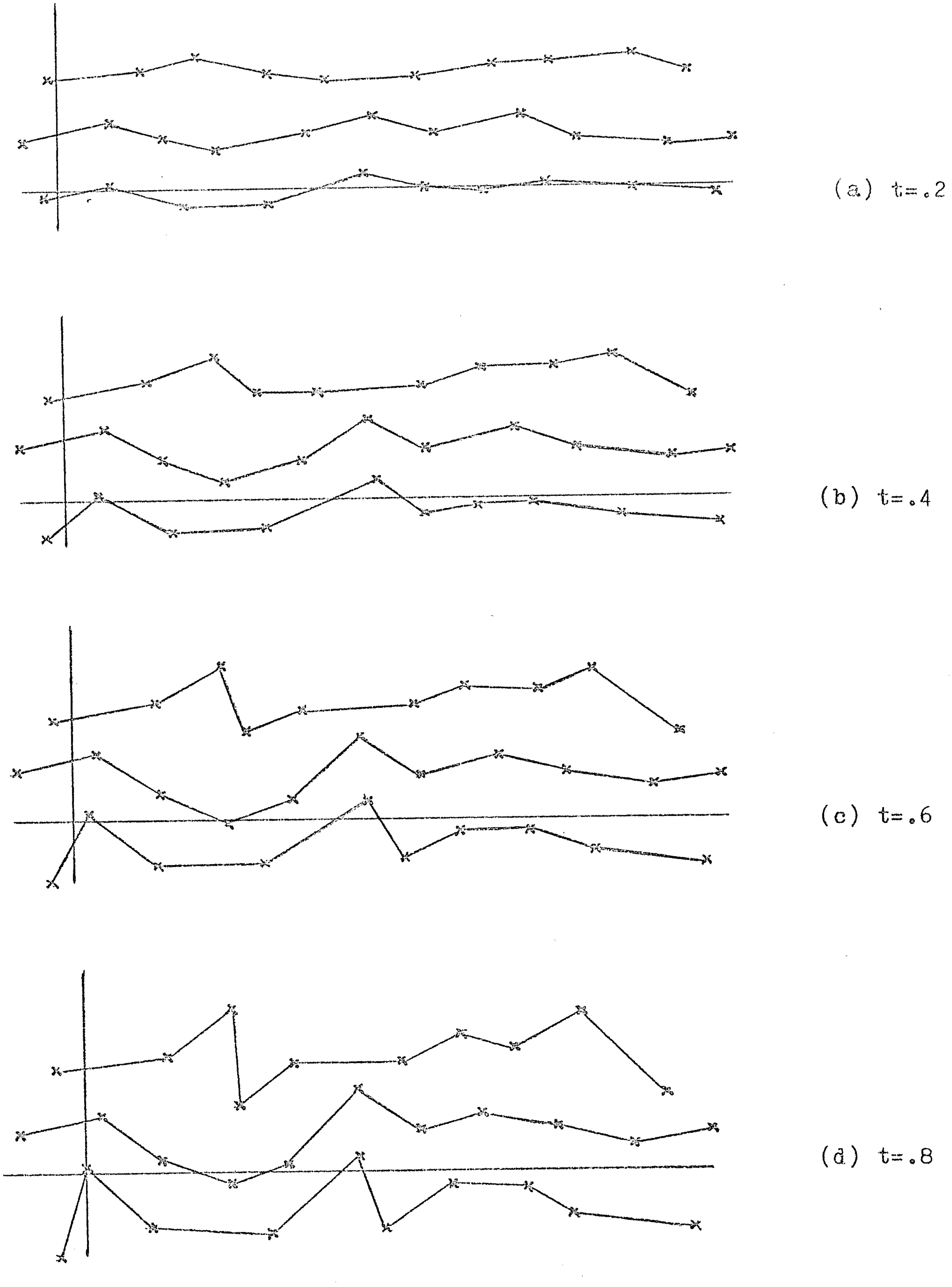


(e) t=1.0

(f) t=1.2

(g) t=1.4

Fig. 3.2



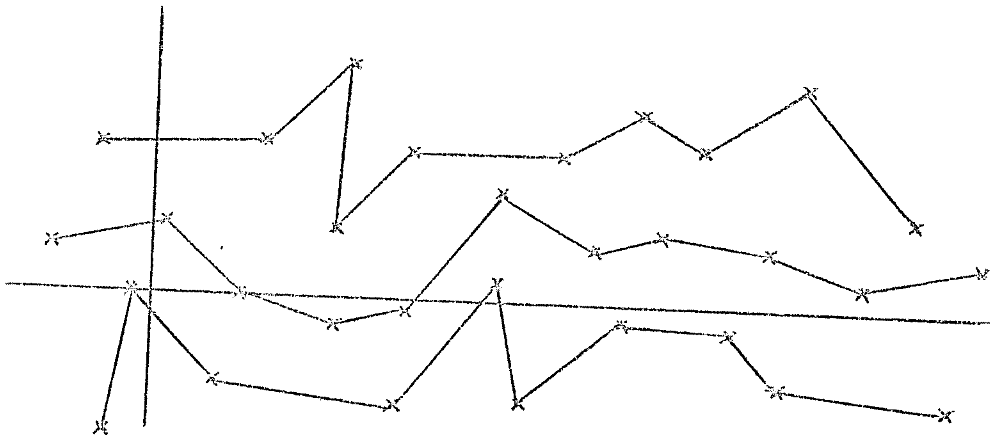
(a) t=.2

(b) t=.4

(c) t=.6

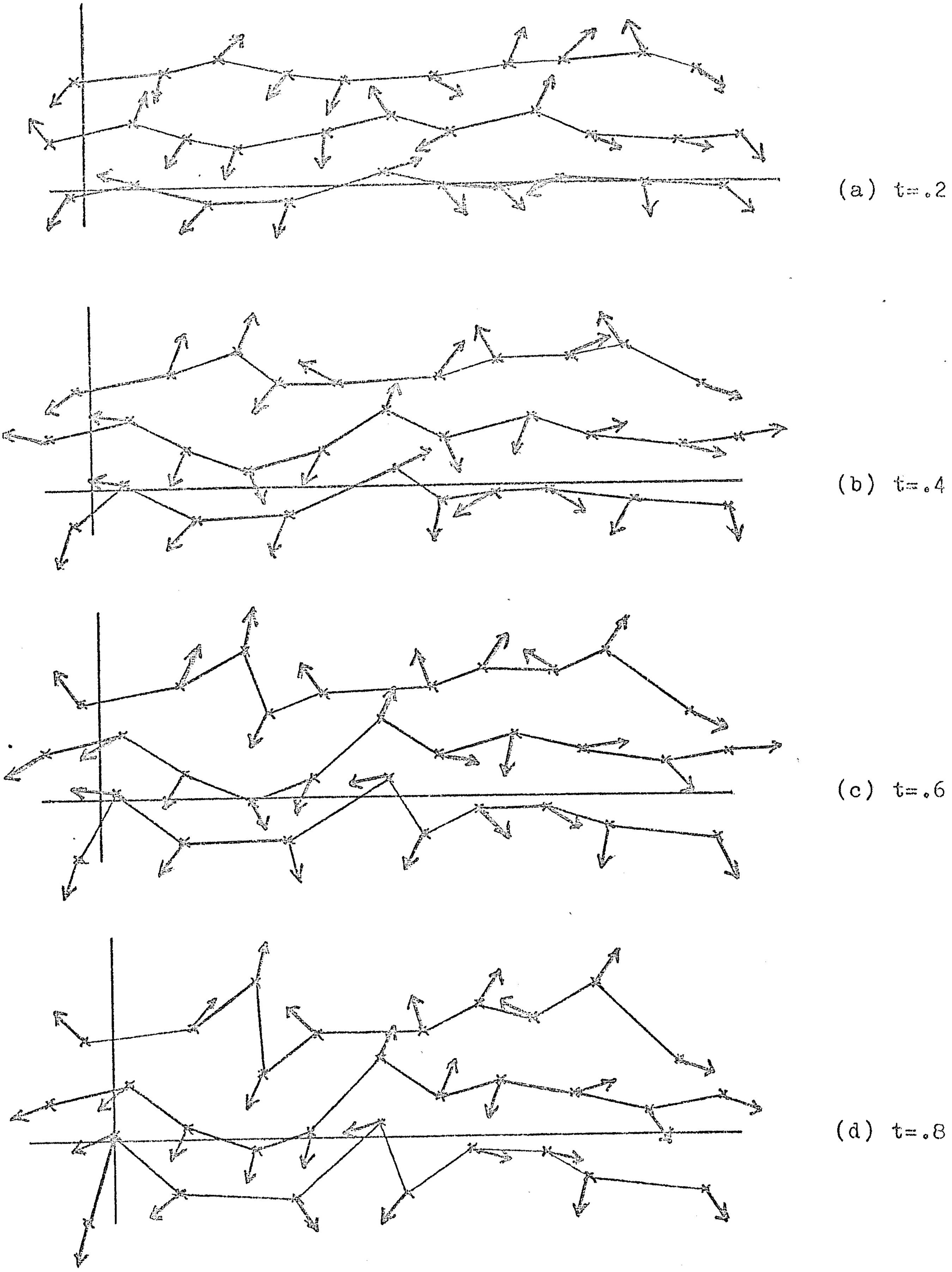
(d) t=.8

Fig. 3.3



(e) t=1.0

Fig. 3.3



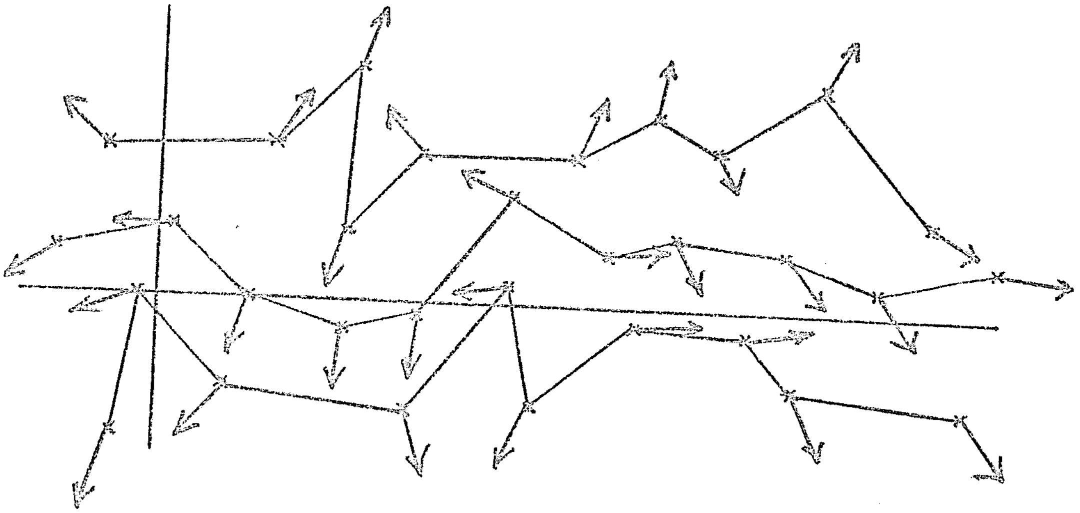
(a) t=.2

(b) t=.4

(c) t=.6

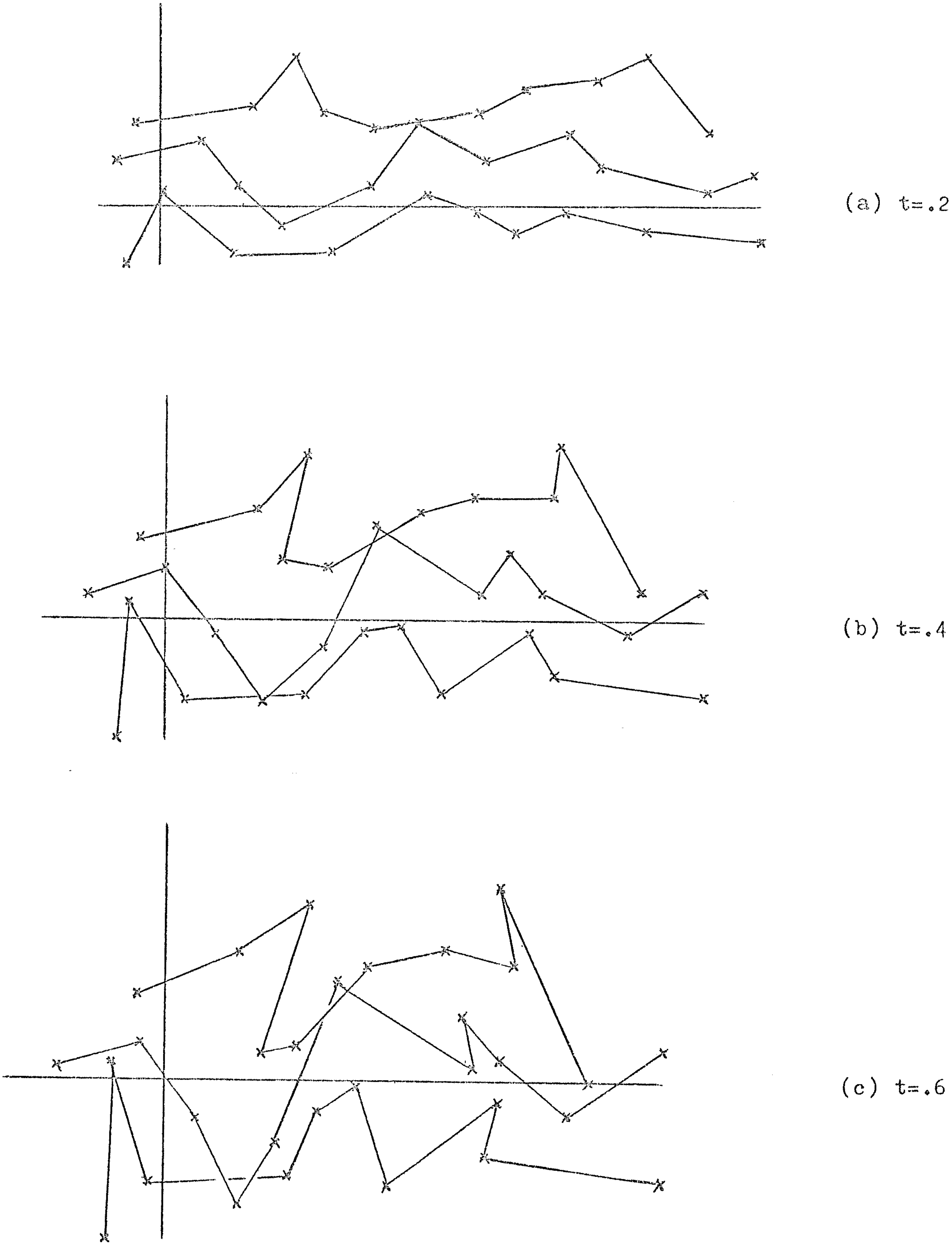
(d) t=.8

Fig. 3.4



(f) t=1.0

Fig. 3.4

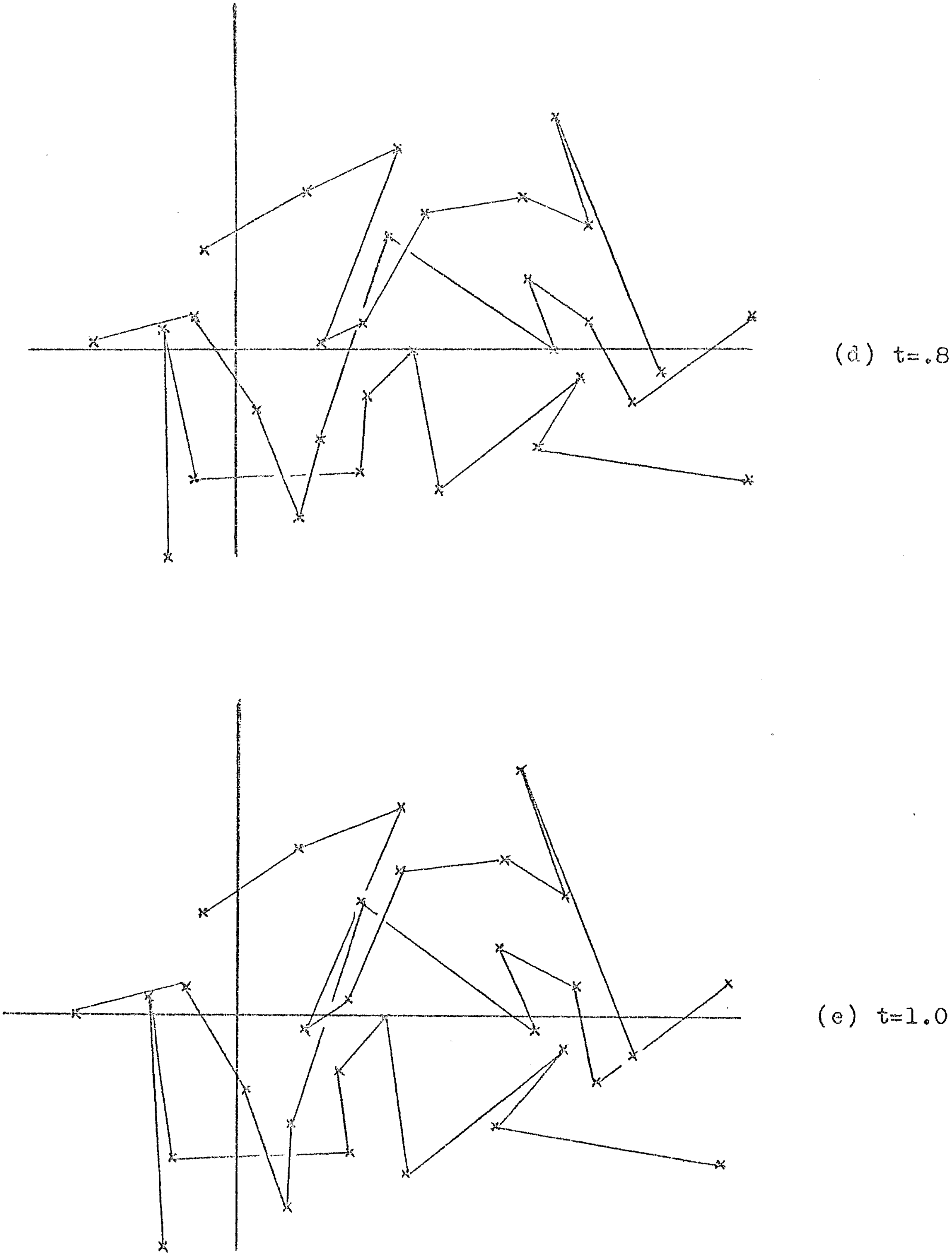


(a) t=.2

(b) t=.4

(c) t=.6

Fig. 3.5



(d) t=.8

(e) t=1.0

Fig. 3.5

Now, in the examples presented thus far, we have assumed that velocities of particles had been perturbed by, say, collisions with a wall. Let us then look at some simple examples which explore such interaction, since, for example, the action of fluid particles which are near the particles of a solid boundary can result in a transfer of kinetic energy for that boundary, thus yielding a viable model of friction [6], which, in turn, suggests a mechanism for boundary layer development in a liquid and shock wave development in a gas.

In Figure 3.6 is shown a fluid particle P_1 of mass m_1 approaching wall particles $P_2 - P_9$, each of mass m_2. The speed of P_1 is $|v|$ and its initial motion is in a direction determined by the angle θ, as shown in the figure, measured from the velocity vector $\vec{v}$ to the horizontal. The distance between any two consecutive wall particles is set at $\sqrt[3]{1.5}$, and P_2 and P_8 are allowed no motion at all. Then, for P_1 positioned initially at $(0.1717, 0.9914)$, and for the parameter choices $m_1 = 0.1$, $m_2 = 10$, $|v| = 10$, $\theta = 30^o$, $\alpha = 7$, $\beta = 10$, $g = 980$, $p = 10^{-2}$, $q = 10^{-5}$, and $\Delta t = 10^{-4}$, Figure 3.7 shows the motion of P_1 from t_0 to t_{200}. The interplay of gravity and repulsion results in a relatively gradual fall of the particle, due to gravity, interspersed with several small rises, due to repulsion. In Figure 3.7 is shown the graph of the x-component of velocity of P_1 from t_0 to t_{200}. The initial value $v_{1,0,x} = 8.66$ is maximal, while subsequent values indicate relatively nonuniform behavior. Were fluid particles above P_1 to maintain horizontal velocity components of approximately 8.66, then Figure 3.8 implies that fluid motion near the wall would be relatively slower.

Finally, let us examine the possibility of P_1, in the above example, moving at an exceptionally high velocity. To do this, P_1 is positioned at (.40065, .09914) and an additional wall particle, P_{10}, is positioned at (.40065, −.09914), as shown in Figure 3.9. Again, P_2 and P_8 are allowed no motion at all, and the only changes in parameters from the above example are $|v| = 1000$, $\theta = 90^{\circ}$, $\Delta t = 10^{-5}$. Figure 3.10 shows the resulting motion of P_1, P_4, P_5, P_6, P_7 and P_{10} at t_{10}, t_{15}, t_{20} and t_{25}. The velocity of P_1 was chosen sufficiently large this time to break the bonding of the boundary particles, and P_1 has penetrated the wall. At t_{25}, P_1 has arrived at the stationary position (.40065, −.05244), has transferred its kinetic energy to the wall particles, and has "dented" the surface.

Note, with regard to all the examples of this section, that the Fortran program is available in [5], while no example ever exceeded 10 minutes of running time on the UNIVAC 1108.

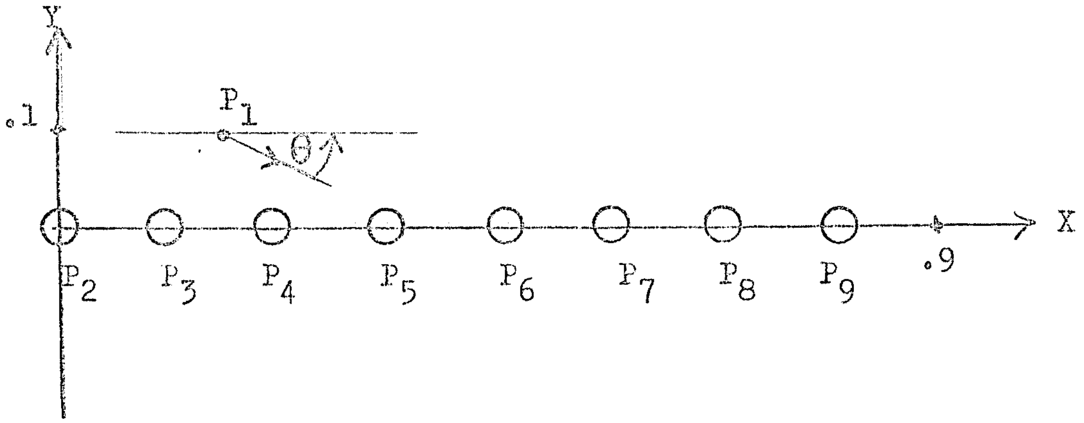


Figure 3.6

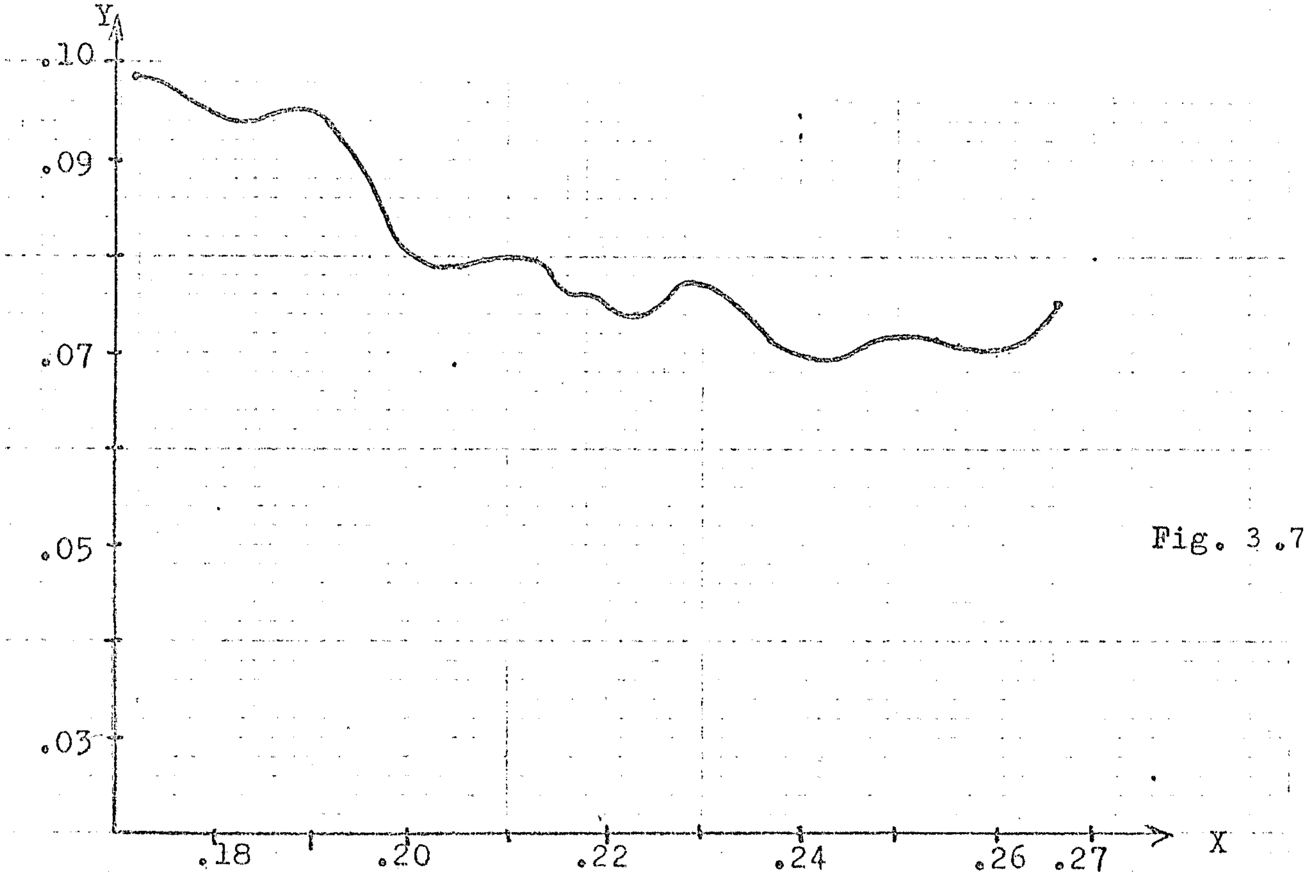


Fig. 3.7

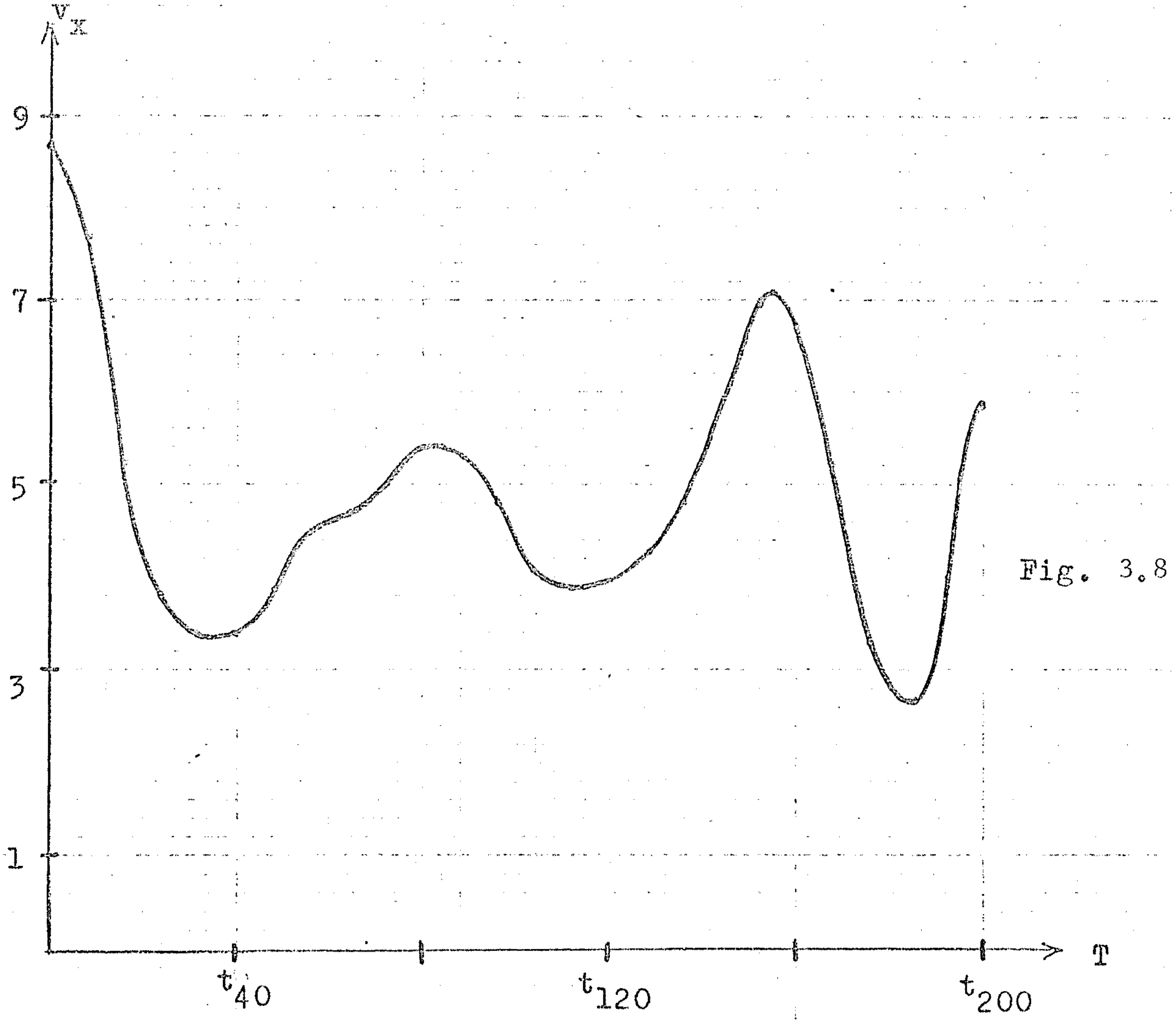


Fig. 3.8

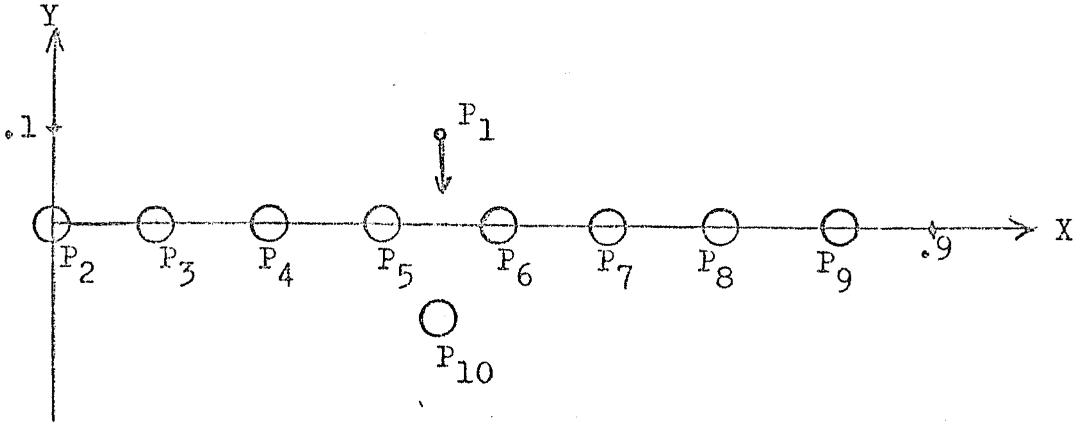


Figure 3.9

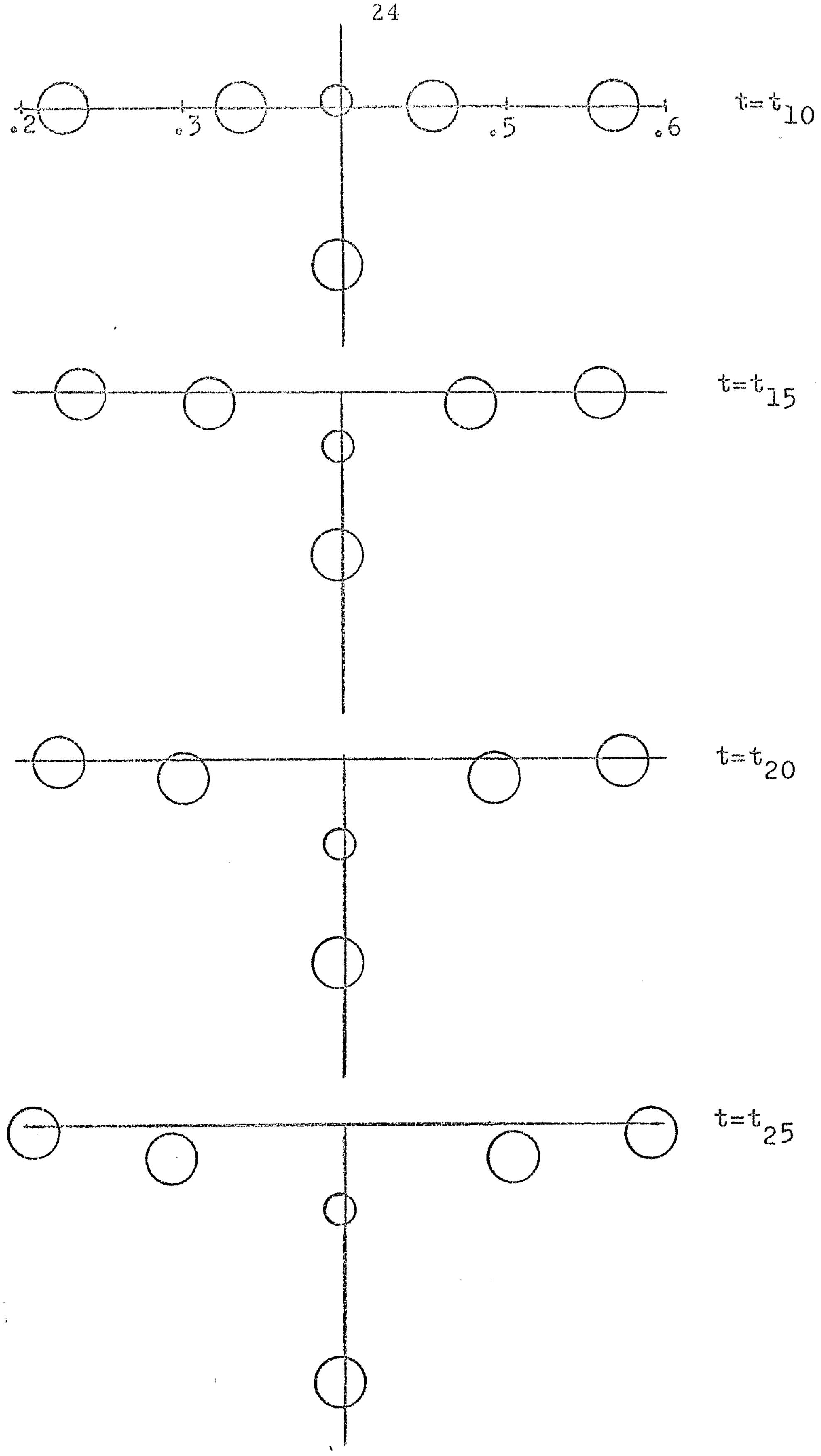


Fig. 3.10

4. Remarks.

Though a variety of examples were run with parameter values different from those given in Section 3, insufficient funding actually prevented the study of large scale examples. It would be of interest to generate computer examples to test the apparently contradictory conclusions of von Neuman [11] and of Pasta and Ulam [12]. Von Neuman concluded that one could simulate the action of 10^{24} molecules of a hydrodynamical model by considering only 10-100 particles provided the intramolecular forces were scaled up. Pasta and Ulam concluded that one would need a minimum of 200,000 particles in order to assure an error of the order of magnitude of 5%. The theory of this paper does allow simulation as suggested by von Neuman. Fluid dynamicists would have to pressure computer manufacturers to develop specialized computers in order to apply the theory of this paper as recommended by Pasta and Ulam. However, if one wants to modify the methods described here, it is possible, using present day computers, to simulate models with 120,000 particles [9] in reasonable computing times. Such modifications, however, are nonconservative.

It would also be of interest to compare the methods of this paper with the nonconservative, molecular-dynamics approach of MacPherson [8] and with the popular particle-in-cell method [1], which is nonconservative and which also restricts particles to nodal positions.

References

1. A. A. Amsden, "The particle-in-cell method for the calculation of the dynamics of compressible fluids," TR3406, Los Alamos Sci. Lab., L. A., N. M., 1966.

2. D. Greenspan, "Discrete Newtonian gravitation and the n-body problem," Utilitas Math., 2, 1972, pp. 105-126.

3. H. P. Greenspan, The Theory of Rotating Fluids, Cambridge Univ. Press, Cambridge, 1968.

4. J. O. Hirschfelder,, C. F. Curtiss, and R. B. Bird, Molecular Theory of Gases and Liquids, Wiley, N. Y., 1954.

5. S. T. Jones, "FORTRAN program for discrete fluids," Appendix, TR#171, Dept. of Comp. Sci., Univ. Wis., Madison, 1973.

6. T. von Karman, Aerodynamics, McGraw-Hill, N. Y., 1963.

7. P. D. Lax, "Nonlinear partial differential equations and computing," SIAM Rev., 11, 1969, pp. 7-19.

8. A. K. MacPherson, "The formulation of shock waves in a dense gas using a molecular-dynamics type technique," J. Fluid Mech., 45, 1971, pp. 601-621.

9. R. H. Miller and K. H. Prendergast, "Stellar dynamics in a discrete phase space," Astrophys. J., 151, 1968, pp. 699-709.

10. R. von Mises, Mathematical Theory of Compressible Fluid Flow, Acad. Press, N. Y., 1958.

11. J. von Neumann, "Proposal and analysis of a new numerical method for the treatment of hydrodynamical shock problems," in The Collected Works of John von Neuman, vol VI, edited by A. H. Taub, Pergamon, N. Y., 1963, pp. 361-379.

12. J. R. Pasta and S. Ulam, "Heuristic numerical work in some problems of hydrodynamics," MTAC, 13, 1959, pp. 1-12.

13. Yu P. Popov and A. A. Samarskii, "Completely conservative difference schemes for the equations of gas dynamics in Euler's variables," USSR Comp. Math. and Math. Phys., 10, 1970, pp. 265-273.

14. L. Rosenhead (Ed.), Laminar Boundary Layers, Oxford Univ. Press, Oxford, 1963.

15. P. G. Saffman, "Lectures on homogeneous turbulence," in Topics in Nonlinear Physics, edited by N. J. Zabusky, Springer-Verlag, N. Y., 1968, pp. 485-614.

16. H. Schlichting, Boundary Layer Theory, McGraw-Hill, N. Y., 1960.

```
C  FORTRAN PROGRAM FOR DISCRETE FLUIDS
       IMPLICIT DOUBLE PRECISION(A-H,M,O-Z)
       DIMENSION XO(10),YO(10),VXO(10),VYO(10),X(10,3),Y(10,3),VX(10,3),
      1VY(10,3),FX(10),FY(10)
       DIMENSION R(10,10,2),M(10)
       DIMENSION GAMMA(10)
       DIMENSION XKE(10)
 1001  FORMAT(8D10.0)
 1002  FORMAT(16I5)
 2000  FORMAT(1H1,' PROGRAM PARAMETERS')
 2001  FORMAT(5X,'OMEGA= ',D6.1,5X,'EPS= ',D6.1/5X,'A= ',D6.1,5X,'B= ',
      1D6.1/5X,'P= ',D6.1,5X,'Q= ',D6.1/5X,'DT= ',D5.1/5X,'NO. PARTICLES
      2THIS CASE= ',I2)
 2002  FORMAT(5X,'PARTICLE SPEED= ',D7.1,' AT ',D12.6,' RADIANS')
 2006  FORMAT(' NON-CONVERGENCE AFTER ',I3,' ITERATIONS FOR TIMESTEP ',
      1I5)
C  SPECIFY INPUT PARAMETERS
       READ 1001,OMEGA,EPS,A,B,G
       READ 1002,N,NMAX,IMAX,IPRINT,JPRINT,IPUNCH,JPUNCH,IPLOT,JPLOT
       READ 1001,P,Q,MASS,DT
       READ 1001,VEL,THETA
       BX=(Q*(B-1)/(P*(A-1)))**(1.0/(B-A))
       BY=DSQRT(BX**2-(0.5*BX)**2)
       OMW=1.0-OMEGA
       DT2=DT/2.0
       PI=3.14159260+00
     2 THETA=PI*THETA/180.
       CALL INP
       VXO(I)=VEL*DCOS(THETA)
       VYO(1)=VEL*DSIN(THETA)
C  PRINT PROGRAM PARAMETERS
       PRINT 2000
       PRINT 2001,OMEGA,EPS,A,B,P,Q,DT,N
       PRINT 2002,VEL,THETA
       NSTP=0
       KPLOT=2
       IPRT=-1
       IPNCH=0
       IPLT=-1
       IF(JPLOT.EQ.0)GO TO 10
C  INITIALIZE PLOT
       CALL INITPL(14,10.8)
    10 CALL OUTP
       IPRT=1
       IPNCH=1
       IPLT=1
C  SPECIFY INITIAL GUESS FOR NEWTON'S ITERATION,FIRST TIMESTEP
       DO 70 I=1,N
       X(I,3)=XO(I)
       Y(I,3)=YO(I)
       VX(I,3)=VXO(I)
    70 VY(I,3)=VYO(I)
       CALL RCALC
C  UPDATE POSITIONS,VELOCITIES,DISTANCES--ALL TIMESTEPS
```

```
   75 NSTP=NSTP+1
      DO 85 I=1,N
      X(I,1)=X(I,3)
      Y(I,1)=Y(I,3)
      VX(I,1)=VX(I,3)
      VY(I,1)=VY(I,3)
      DO 80 J=1,N
   80 R(I,J,1)=R(I,J,2)
   85 CONTINUE
C  BEGIN ITERATION LOOP
      DO 115 K=1,IMAX
C  UPDATE ALL VARIABLES,CURRENT TIMESTEP,PREVIOUS ITERATION
      DO 90 I=1,N
      X(I,2)=X(I,3)
      Y(I,2)=Y(I,3)
      VX(I,2)=VX(I,3)
   90 VY(I,2)=VY(I,3)
C  UPDATE ALL VARIABLES,CURRENT TIMESTEP,CURRENT ITERATION
      DO 95 I=1,N
      IF(I.EQ.2.OR.I.EQ.9)GO TO 95
      X(I,3)=OMW*X(I,2)+OMEGA*(DT2*(VX(I,2)+VX(I,1))+X(I,1))
      Y(I,3)=OMW*Y(I,2)+OMEGA*(DT2*(VY(I,2)+VY(I,1))+Y(I,1))
   95 CONTINUE
      CALL RCALC
      CALL FCALC
C  UPDATE VELOCITIES,CURRENT TIMESTEP,CURRENT ITERATION
      DO 100 I=1,N
      IF(I.EQ.2.OR.I.EQ.9)GO TO 100
      VX(I,3)=OMW*VX(I,2)+OMEGA*(DT*FX(I)+VX(I,1))
      VY(I,3)=OMW*VY(I,2)+OMEGA*(DT*FY(I)+VY(I,1))
  100 CONTINUE
C  TEST FOR CONVERGENCE
      DO 110 I=1,N
      IF(ABS(X(I,3)-X(I,2)).GT.EPS)GO TO 115
      IF(ABS(Y(I,3)-Y(I,2)).GT.EPS)GO TO 115
      IF(ABS(VX(I,3)-VX(I,2)).GT.EPS)GO TO 115
      IF(ABS(VY(I,3)-VY(I,2)).GT.EPS)GO TO 115
  110 CONTINUE
      GO TO 120
  115 CONTINUE
      PRINT 2006,K,NSTP
      GO TO 135
  120 CALL OUTP
      IF(NSTP.LT.NMAX)GO TO 75
  135 IF(JPLOT.EQ.0)STOP
      CALL ENDPLT
      STOP
      SUBROUTINE INP
      XO(1)=3.5*BX
      XO(2)=0.0
      DO 112 II=3,N
      XO(II)=XO(II-1)+BX
  112 CONTINUE
      DO 111 I=2,N
      Y0(I)=0.0
      VX0(I)=0.0
      VY0(I)=0.0
      GAMMA(I)=0.0
      M(I)=MASS
  111 CONTINUE
      Y0(I)=BY
      M(1)=0.1
      GAMMA(1)=1.0
      X0(10)=3.5*BX
      Y0(10)=-BY
      RETURN
C  INTERNAL SUBROUTINE TO COMPUTE DISTANCES BETWEEN PARTICLES
      SUBROUTINE RCALC
      NM1=N-1
      DO 250 II=1,NM1
      IP1=II+1
      DO 200 JJ=IP1,N
      R(II,JJ,2)=SQRT((X(II,3)-X(JJ,3))**2+(Y(II,3)-Y(JJ,3))**2)
      R(JJ,II,2)=R(II,JJ,2)
  200 CONTINUE
  250 CONTINUE
      RETURN
C  INTERNAL SUBROUTINE TO COMPUTE FORCES
      SUBROUTINE FCALC
      IA=A-1
      IB=B-1
      DO 690 II=1,N
      IF(II.EQ.2.OR.II.EQ.9)GO TO 690
      SUMX=0.0
      SUMY=0.0
      DO 660 JJ=1,N
      IF(II.EQ.JJ)GO TO 660
      SUMP=0.0
      SUMQ=0.0
      RIJ=R(II,JJ,1)+R(II,JJ,2)
      DO 600 I2=1,IA
      SUMP=SUMP+(R(II,JJ,1)**(I2-1))*(R(II,JJ,2)**(A-(I2-1)-2))
  600 CONTINUE
      DO 630 I2=1,IB
      SUMQ=SUMQ+(R(II,JJ,1)**(I2-1))*(R(II,JJ,2)**(B-(I2-1)-2))
  630 CONTINUE
      PD=R(II,JJ,1)**(A-1)*R(II,JJ,2)**(A-1)*RIJ
      SUMP=P*SUMP/PD
      QD=R(II,JJ,1)**(B-1)*R(II,JJ,2)**(B-1)*RIJ
      SUMQ=Q*SUMQ/QD
      SUMX=(SUMQ-SUMP)*M(JJ)*(X(II,3)+X(II,1)-X(JJ,3)-X(JJ,1))+SUMX
      SUMY=(SUMQ-SUMP)*M(JJ)*(Y(II,3)+Y(II,1)-Y(JJ,3)-Y(JJ,1))+SUMY
  660 CONTINUE
      FX(II)=M(II)*SUMX
      FY(II)=M(II)*(SUMY-GAMMA(II)*G)
  690 CONTINUE
      RETURN
C  INTERNAL OUTPUT SUBROUTINE
```

```
      SUBROUTINE OUTP
      REAL JSCALX(10),JSCALY(10),KXX,KYY
      DIMENSION XPL(100),YPL(100),VXPL(100),VYPL(100),AXLIM(2)
 3001 FORMAT(///' INITIAL CONDITIONS'/5X,'MASSES THIS CASE'/10X,3D10.3)
 3002 FORMAT(//' CONVERGENCE FOR TIMESTEP ',I3,' IN ',I3,' ITERATIONS')
 3003 FORMAT(5D24.16)
 3004 FORMAT(I5)
 3005 FORMAT(2D25.18)
      IF(NSTP.NE.0)GO TO 720
  700 DO 710 II=1,N
      XPL(II)=X0(II)
      YPL(II)=Y0(II)
      VXPL(II)=VX0(II)
      VYPL(II)=VY0(II)
  710 CONTINUE
      GO TO 740
  720 DO 730 II=1,N
      XPL(II)=X(II,3)
      YPL(II)=Y(II,3)
      VXPL(II)=VX(II,3)
      VYPL(II)=VY(II,3)
  730 CONTINUE
  740 IF(JPRINT.EQ.0)GO TO 800
      IF(IPRT)760,800,750
  750 IF(MOD(NSTP,IPRINT).NE.0)GO TO 800
      GO TO 761
  760 PRINT 3001,M(1),MASS
      GO TO 762
  761 PRINT 3002,NSTP,K
      DO 251 II=1,N
      XKE(II)=0.5*(VXPL(II)**2+VYPL(II)**2)
  251 CONTINUE
  762 DO 780 II=1,N
      PRINT 3003,XPL(II),YPL(II),VXPL(II),VYPL(II),XKE(II)
  780 CONTINUE
  800 IF(JPUNCH.EQ.0)GO TO 900
      IF(IPNCH)840,900,820
  820 IF(MOD(NSTP,IPUNCH).NE.0)GO TO 900
C  PUNCH REQUIRED THIS TIMESTEP
  840 WRITE(1,3004)NSTP
      WRITE(1,3005)(M(II),II=1,N)
      DO 860 II=1,N
      WRITE(1,3005)XPL(II),YPL(II)
  860 CONTINUE
      DO 880 II=1,N
      WRITE(1,3005)VXPL(II),VYPL(II)
  880 CONTINUE
  900 IF(JPLOT.EQ.0)RETURN
      IF(IPLT)920,970,910
  910 IF(MOD(NSTP,IPLOT).NE.0)RETURN
C  PLOT REQUIRED THIS TIMESTEP
C  COMPUTE MAXIMUM COORDINATE RANGE FOR AXES SCALE
  920 AXLIM(1)=XPL(1)
      AXLIM(2)=XPL(1)
      DO 930 II=1,N
      AXLIM(1)=DMIN1(AXLIM(1),XPL(II),YPL(II))
      AXLIM(2)=DMAX1(AXLIM(2),XPL(II),YPL(II))
  930 CONTINUE
      IF(MOD(KPLOT,2).NE.0)GO TO 940
      CALL PAGE(8.5,11.0,'NORMAL','NONE')
      KYY=5.75
      GO TO 950
  940 KYY=0.25
  950 KXX=1.25
      CALL SCALE(AXLIM,2,JSCALX,'DOUBLE',0,0)
      CALL SCALE(AXLIM,2,JSCALY,'DOUBLE',0,0)
      CALL BASIS(KXX,KYY,JSCALX,5.0,0.,JSCALY,5.0,90.)
      CALL AXLIN(JSCALX,JSCALY,0.00,0,0,0,0,'$$')
      CALL AXLIN(JSCALY,JSCALX,0.00,0,0,0,0,'$$')
C  PLOT POINTS
      DO 960 II=1,N
      CALL LOCATE(XPL(II),JSCALX,YPL(II),JSCALY,XP,YP,JDUM)
      CALL PLSYMB(XP,YP,4,0.04,0.,'UP')
  950 CONTINUE
      KPLOT=KPLOT+1
  970 RETURN
      END
```