THE USE OF COMPUTERIZED PROBLEM SOLVERS IN ENGINEERING SCIENCES

By

LEONARDO Q. LIONGSON, Ph.D.*

INTRODUCTION

Engineering sciences encompass the common basic engineering courses which are required in the curricula of the various undergraduate and graduate engineering degree programs. For the undergraduate programs, these courses are technical drawing, mechanics of rigid and deformable bodies, fluid mechanics, and engineering mathematics. The first engineering science courses taken by the students require as prerequisites several units in physics and college mathematics which may be as advanced as differential and integral calculus.

That the computer is a useful tool for instruction and research has been demonstrated by teachers and students who have been placed in the happy situation where computer installations are primarily intended for such uses. The teaching of engineering science courses, to the writer's opinion, may offer a common fresh and fertile ground where the full potential of the computer as a tool of the engineering educator can be explored. Engineering science students, at their stage of learning, possess the basic foundations in the physical sciences and should then be prepared for and receptive to the rigor and completeness in the treatment of topics covered by the engineering science subjects. At the same time, these students have yet to acquire the intermediate skills and knowledge which are prerequisites to and are, in effect, taken for granted in the design and other advanced courses which they have to take later. In short, the computer may be best utilized in that important step in the ladder of learning wherein students already attuned to the natural sciences are further molded to become the critical learners of designs, principles and criteria, which are significantly rooted in the engineering sciences.

COMPUTERIZED PROBLEM SOLVERS

We want to explore the potentials of the computer beyond its mere function as a fast and efficient modern version of the slide rule or as a heavy-duty "big brother" of the pocket calculator. By means of the com-

* Chairman, and Assistant Professor, Department of Engineering Sciences, U.P. College of Engineering.

puter, we want to bring the ideas contained in the lecture and textbooks closer and make the same more familiar to the students. We want to try and experiment with the facilities and features uniquely afforded by the computer which can supplement the teacher's lecture, the reading assignments, and the homeworks. We want to encourage the students to formulate and pose problems which they can solve interactively with the machine—the latter acquiring the identity of an accurate and smart but patient and non-intimidating partner in learning. We shall use the term computerized problem solvers to mean computer programs specifically designed to solve engineering problems and patterned to require a certain degree of student-machine interaction so that the learner begins to appreciate not only the desired answer but also the underlying physical concepts of the problem. The details of the mathematical steps and the so-called algorithms of the program may not be transparent to the user; what is more important is for the student to realize, appreciate, and understand the role and effect of different physical variables and parameters on the final answer or set answers.

A convenient and by no means mutually exclusive grouping of computerized problem solvers is given as follows:

1. Equation solver
2. Simulator
3. Tutor

EQUATION SOLVER

An equation solver yields the values of "unknown" variables which constitute the solutions of certain equations requiring as inputs the "known" variables and parameters of the problem. Several variations of choices of known and unknown variables may exist for a given total set of variables in a given generic problem; the user may opt for the particular choice of known and unknown variables. The process of solution may be straightforward, based on an explicit or closed-form formula, or it may be an iterative technique involving the application of numerical methods.

A simple example is an equation solver based on Manning's formula for uniform open-channel flow:

$$Q = \frac{1.49}{n} AR^{2/3}S^{1/2}$$

where Q is the flow rate, n is Manning's constant (a property of the channel bed material), A is the wetted area of flow, R is the hydraulic radius, functions of the flow depth y for a given cross-section geometry of the

channel. Other parameters which describe the shape of the cross section appear in the functions for A and R.

Take a trapezoidal cross section which is parameterized by the bottom width B and the bank slope 1:m, as shown in Figure 1. It follows that

$$A = By + 1/2\ my^2$$

$$R = \frac{By + 1/2\ my^2}{B + 2\sqrt{1 + m^2}\ y}$$

If $B = 0$, and $m \neq 0$, the trapezoid reduces to a triangle. On the other hand, if $m = 0$ and $B \neq 0$, it becomes a retangle. Thus, the triangular and rectangular cross sections are special cases.

At least three variations of choices of unknown and known variables are identifiable:

(a) Unknown Q; the rest are specified.
The solution for Q involves the straightforward use of Manning's formula.

(b) Unknown y; the rest are specified.
The solution for y generally involves the application of numerical iterative methods since y appears in A and R in a complicated manner.

(c) Unknown S; the rest are specified.
The solution for S is again straightforward, based on an explicit expression for S derivable from Manning's formula.

In each of these choices, the user can investigate the relative effects of the design parameters n, B, and m.

SIMULATOR

A simulator computes and displays the propagation in time and/or space (in one, two or three dimensions) of the solutions of certain algebraic equations or, more commonly in practice, of ordinary and partial differential equations, subject to specified initial and boundary conditions and prescribed values of governing parameters. The unknown or dependent variables in a simulation environment are sometimes called state variables. They change and propagate with respect to time and space. The nature of the initial and boundary conditions as well as the values of the governing parameters determine the particular behavior of the solution in an actual run of the simulator.

A classical example is the simulator for the vibration of a mass-spring-damper system (Figure 2). The governing ordinary differential equation is

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

subject to the initial conditions that $x = x_0$ and $\frac{dx}{dt} = v_0$ at $t = 0$.

Here x is the displacement of the mass from the equilibrium position, and it is the dependent variable. Time t is the independent variable. The mass m, damping coefficient c, and spring constant k are the governing parameters. There are three qualitatively different types of solutions:

(a) If $c^2 < 4mk$, the vibration is said to be underdamped.
(b) If $c^2 = 4mk$, the vibration is critically damped.
(c) If $c^2 > 4mk$, the vibration is overdamped.

The user of the simulator has at his disposal the manipulation of the values of m, c, k, x_0 and v_0 and the subsequent examination of the responses of the solution to changes in those parameters and initial conditions.

A second example which we shall study with a certain amount of detail is a simulator of the steady-state pattern of streamlines in two-dimensional, incompressible, either irrotational or laminar, flow. The program is called VORTICES-15 — acronym for Visually Oriented Routines for Two-dimensional-flow Identification and Calculation in E.S. 15. E.S. 15 is the engineering science course on mechanics of fluids. The simulator features a flow tunnel (we can call it a numerical wind tunnel) whose exact dimensions are user-specified (see Appendix I). The total amount of flow is also user-specified. (see Appendix I). A very important feature is the facility for the user to fashion the shape of the tunnel walls in any desired manner and to introduce one or more immersed solid bodies of arbitrary shapes in the flow. The user can also provide the entrance conditions of flow, remove or slide a wall, and qualify whether the flow is irrotational or laminar. The range of capabilities is illustrated by the set of commands available to the user listed in Appendix I. Briefly, the simulator derives by numerical methods the solutions in terms of the stream function, which is actually a means of tagging the streamlines. One value of the stream function identifies one streamline; the value of that function, at the same time, is the amount of flow contained between its corresponding streamline and a certain base streamline, usually a boundary wall. The governing equations for laminar flow which are converted to numerical algorithms are two coupled partial differential equations — in two dependent variables, stream function and vorticity, and in two independent variables, coordinates x and y. Only the stream function is of immediate interest.

For irrotational flow, vorticity simply disappears and only one equation for the stream function applies.

Appendix II presents the coding of commands which achieve the simulation of various flow phenomena involving walls or immersed bodies with arbitrary shapes (see accompanying figures in Appendix III).

The writer utilized the simulator in elucidating the nature of and contrast between irrotational and laminar flows before two sections of E.S. 15. VORTICES-15 has been very helpful in providing to the class visible examples of flows resulting from boundary layer formation and separation, a particularly difficult topic for many students. Due to limitations of time and computer funds, however, the students were not able to use VORTICES-15 extensively. The program runs in an IBM 360 computer with a core memory size of 256 k bytes.

TUTOR

A tutor may contain the features of an equation solver or a simulator but an important and necessary property is that it can conduct the semblance of a conversation with the user — that is, it is an interactive program. It can solicit answers from the user by means of queries. These queries may be solicitations for choices by the user among several options of problem types or combinations of variables. They are questions answerable by numbers, by yes or no, or by categorical literal answers. These queries are put forward at the start of the run and generated during the course of the computations. The computer provides an immediate feedback to the answers or interpretations yielded by the user. The computer via the tutor can affirm or negate the user's response, can suggest or guide as to subsequent options open to the user, and can evaluate or rate the user at the end of one session.

Initial efforts to this end are exemplified by two short computer programs developed by Dr. Edgardo S. Pacheco, Professor of Engineering Sciences, with the TRS-80 microcomputer. One is a matrix multiplication tutor and the other is a vibration analysis program for a spring-mass-damper system in series. The latter exhibits the dual properties of tutor and simulator.

CONCLUSION

It is believed that with the advent of the microcomputer as well as the continued availability of the bigger main frame computers, we can meet the challenge posed by the recognized need to supplement text and lecture materials with other media of learning wherein the student can participate when he is most ready and receptive, wherein time pressure and other constraints are absent to inhibit the learning process, wherein work becomes fun, and wherein the satisfaction of immediate feedback on

his performance is guaranteed. These are prescriptions we hope to carry out if we can devote some of our time and resources to rethinking our classroom topics, translating them to computerized problem solvers, and developing, testing, using, and refining the same. We may acquire in the end, for ourselves and for our students, a magic wand that can replace the pedant's stick.

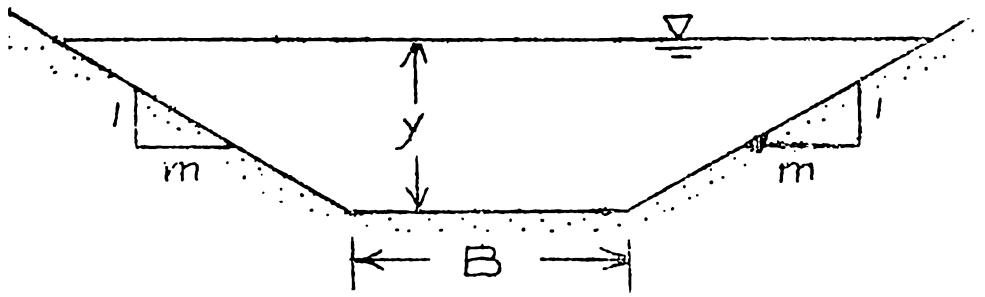

FIG. 1. TRAPEZOIDAL CROSS SECTION

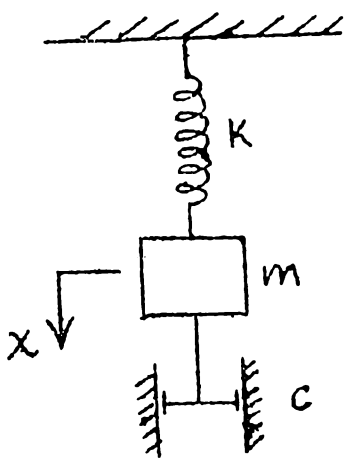

FIG. 2. SPRING-MASS DAMPER SYSTEM

APPENDIX I

COMMANDS IN VORTICES–15

1. TASK 'taskname'
   Initiates a task and defines the taskname or title.

2. SIZE lt wt q
   Prescribes the length lt and width wt of the tunnel and the total flowrate q per unit thickness.

3. L. WALL s x1 x2 y1 y2
   Defines a segment of the left (or lower) wall as a line connecting the points (x1, y1) and (x2, y2) with stream function value s.

4. R. WALL s x1 x2 y1 y2
   Defines a segment of the right (or upper) wall as a line connecting the points (x1, y1) and (x2, y2) with stream function value s.

5. BODY s x1 x2 y1 y2 y3 y4
   Defines a segment of an immersed solid body as a trapezoid with vertices (x1, y1), (x1, y3), (x2, y2) and (x2, y4) with stream function value s.

6. OVAL s e ar br xc yc
   Defines a generalized oval body with equation

$$\left|\frac{x-xc}{ar}\right|^{e}+\left|\frac{y-yc}{br}\right|^{e}=1$$

   and with stream function value s.

7. TRIANGLE s x1 y1 x2 y2 x3 y3
   Defines a triangular body with vertices (x1, y1), (x2, y2), and (x3, y3) and with stream function value s.

8. LINE s x1 y1 x2 y2
   Defines a straight initializing streamline which connects the points (x1, y1) and (x2, y2) and with stream function value s.

9. ENTRANCE y1 y2 u1 u2 s1 s2 a
   Prescribes the entrance flow conditions at $x = 0$ in terms of the line connecting points (0, y1) and (0, y2) as entrance flow area, flow speeds u1 and u2, and stream function values s1 and s2, at those two points, respectively, and the angle a made by the entrance flow with respect to the horizontal axis.

10. VISCO.NU v
Specifies the coefficient of kinematic viscosity v of the fluid.

11. L. FREE
Removes the left (or lower) wall and renders that boundary shear-tree.

12. R. FREE
Removes the right (or upper) wall and renders that boundary shear-free.

13. L. SPEED u
Assigns a horizontal speed u to the left (or lower) wall.

14. R. SPEED u
Assigns a horizontal speed u to the right (or upper) wall.

15. IRROTNAL
Qualifies the flow as irrotational.

16. LAMINAR
Qualifies the flow as laminar.

17. EXECUTE
Initializes the distribution of stream function and vorticity, applies the numerical algorithm for the steady-state stream function and vorticity, and displays entire plots of streamlines and isovorticity lines.

18. MASK s1 s2
Selectively plots the streamlines with stream function values which fall within the range from s1 to s2.

19. RELATE u v
Derives and plots the streamlines as viewed from a moving reference frame with uniform velocity (u, v), based on the previously computed stream function.

20. SAVE
Saves the current stream function against the transformations which follow the command RELATE.

21. RECALL
Recalls the saved stream function.

22. STOP
Terminates the task.

23. END
Terminates the run of a series of tasks.

APPENDIX II

EXAMPLES OF CODED INPUTS TO VORTICES–15

1. TASK IRROTATIONAL FLOW PAST A TRIANGLE
   SIZE 120. 100. 100.
   TRIANGLE 50. 20. 50. 50. 80. 100. 20.
   IRROTNAL
   EXECUTE
   STOP

2. TASK IRROTATIONAL FLOW PAST INCLINED PLATE W/O CIRCULATION
   SIZE 200. 100. 100.
   BODY 50. 50. 150. 70. 25. 75. 30.
   IRROTNAL
   EXECUTE
   STOP

3. TASK IRROTATIONAL FLOW PAST INCLINED PLATE WITH CIRCULATION
   SIZE 200. 100. 100.
   BODY 25. 50. 150. 70. 25. 75. 30.
   IRROTNAL
   EXECUTE
   STOP

4. TASK LAMINAR FLOW PAST AN INCLINED PLATE
   SIZE 120. 100. 100.
   BODY 50. 25. 75. 72. 22. 78. 28.
   LINE 50. 25. 78. 120. 78.
   LINE 50. 75. 22. 120. 22.
   L. FREE
   R. FREE
   VISCO. NU 0.10
   LAMINAR
   EXECUTE
   MASK 30. 70.
   STOP

5. TASK LAMINAR FLOW PAST A NORMAL FLAT PLATE
   SIZE 120. 100. 100.
   VISCO. NU 1.25
   BODY 50. 50. 55. 25. 25. 75. 75.
   LINE 50. 55. 25. 120. 25.
   LINE 50. 55. 75. 120. 75.
   L. SPEED 1.0
   R. SPEED 1.0

```
LAMINAR
EXECUTE
MASK.       40.     60.
STOP
```

6. TASK LAMINAR FLOW PAST A NORMAL FLAT PLATE (HALF-FIELD)

```
SIZE        120.    100.    100.
VISCO. NU   2.5
L. WALL     0.0     50.     55.     50.     50.
LINE        0.0     55.     50.     120.    50.
L. FREE
R. FREE
LAMINAR
EXECUTE
MASK        -100.   0.0
SAVE
RELATE      1.0
RECALL
MASK        0.0     100.
STOP
```

7. TASK LAMINAR FLOW PAST A CIRCULAR CYLINDER (HALF-FIELD)

```
SIZE        120.    100.    100.
VISCO. NU   0.5
OVAL        0.0     2.0     25.     25.     45.     0.0
L. FREE
R. FREE
LAMINAR
EXECUTE
MASK        -100.   0.0
RELATE      1.0
STOP
```

8. TASK LAMINAR FLOW PAST A SLIT

```
SIZE        120.    100.    100.
VISCO. NU   0.10
L. WALL     0.0     35.     40.     40.     40.
R. WALL     100.    35.     40.     60.     60.
LINE        0.0     40.     40.     120.    40.
LINE        100.    40.     60.     120.    60.
LAMINAR
EXECUTE
STOP
```

```
9. TASK       CONFINED OBLIQUE LAMINAR JET
   SIZE       120.  100.  100.
   ENTRANCE   45.   55.   0.0   0.0   0.0   100.  45.
   L. WALL    0.0   0.0   10.   45.   55.
   R. WALL    100.  0.0   10.   55.   65.
   LINE       0.0   10.   55.   120.  55.
   VISCO. NU  1.0
   LAMINAR
   EXECUTE
   STOP
```

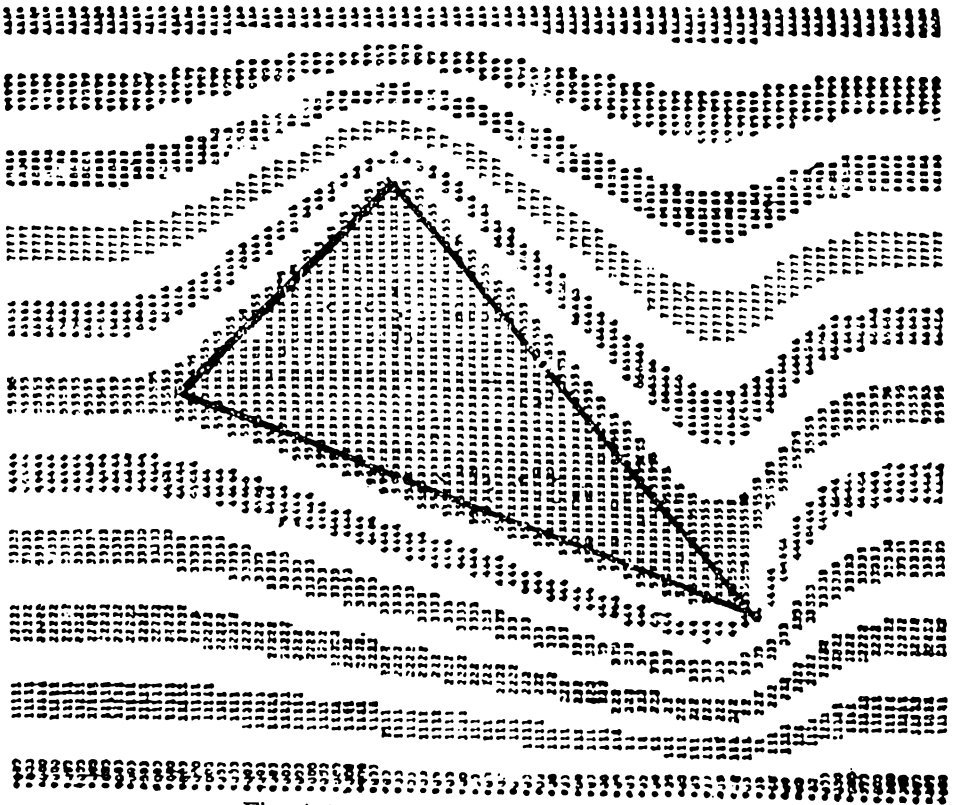

Fig. A.1 Irrotational flow past a triangle

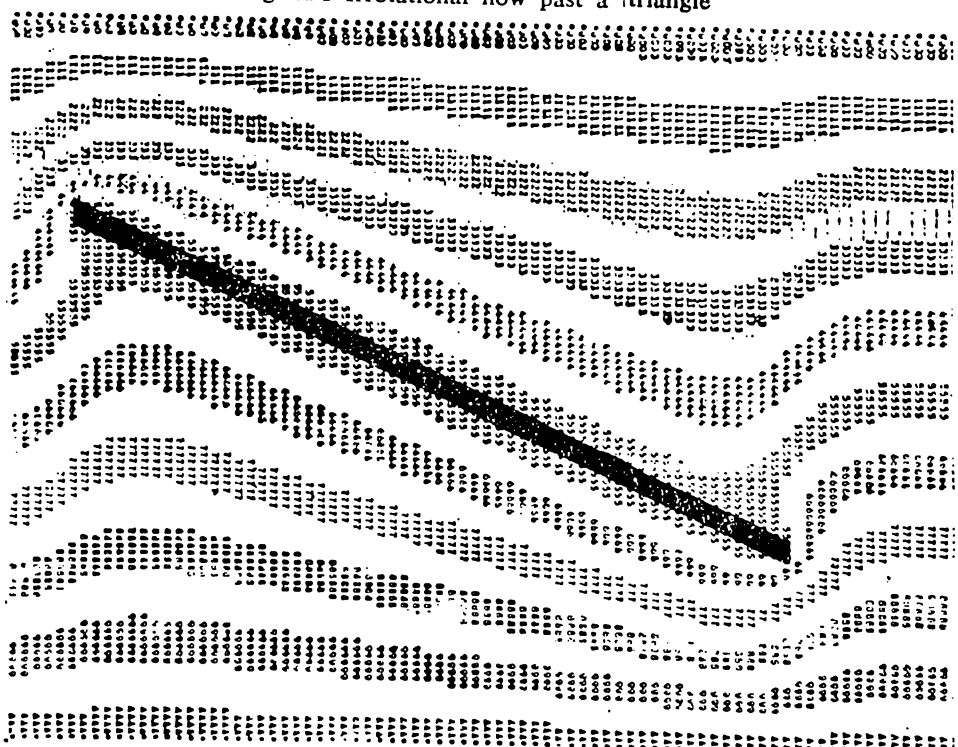

Fig. A.2 Irrotational flow past an inclined plate without circulation

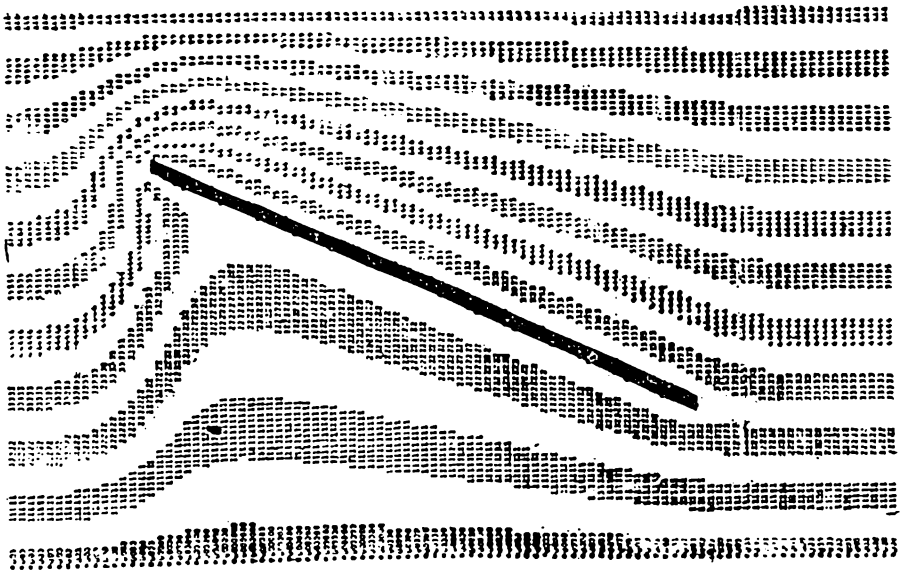

Fig. A.3 Irrotational flow past an inclined plate with circulation

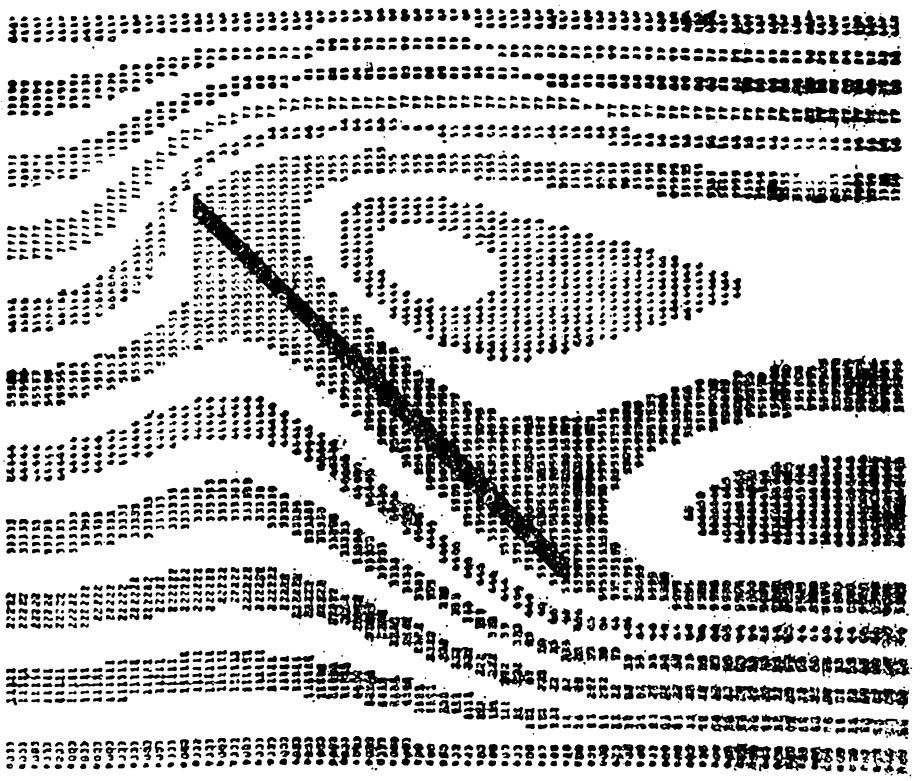

Fig. A.4(a) Laminar flow past an inclined plate after command EXECUTE.

Fig. A.4(b) Laminar flow past an inclined plate after command MASK.

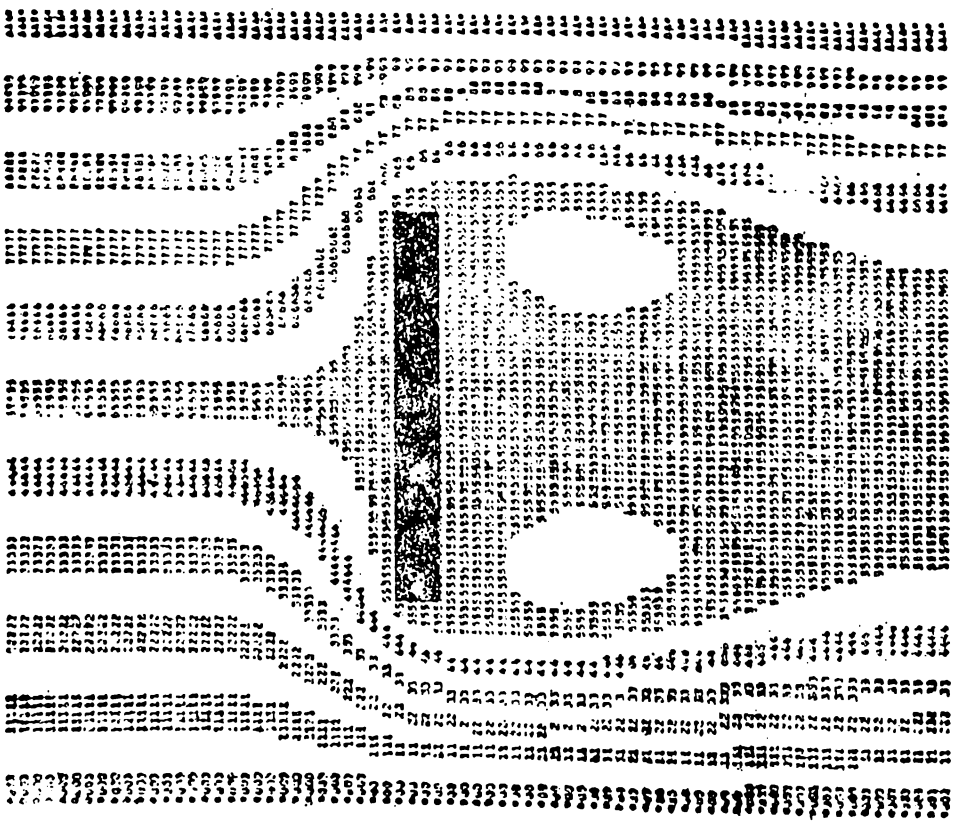

Fig. A.5(a) Laminar flow past a normal flat plate—after command EXECUTE.

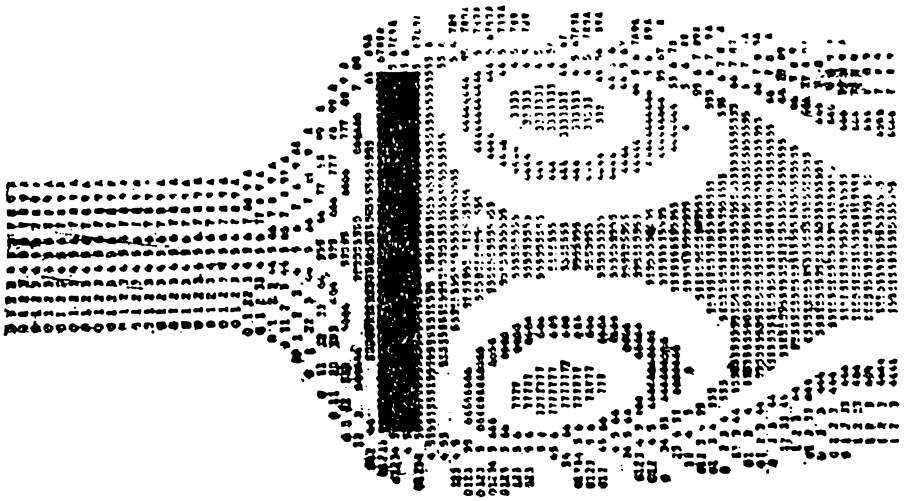

Fig. A.5(b) Laminar flow past a normal flat plate—after command MASK.

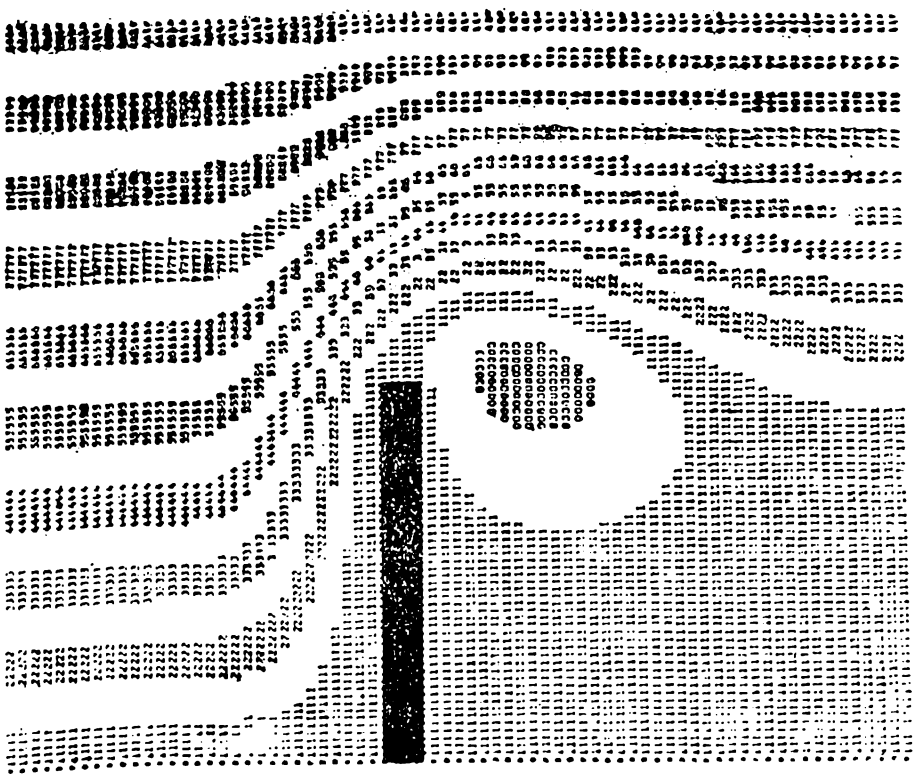

Fig. A.6(a) Laminar flow past a normal flat plate (half-field)—after command EXECUTE.

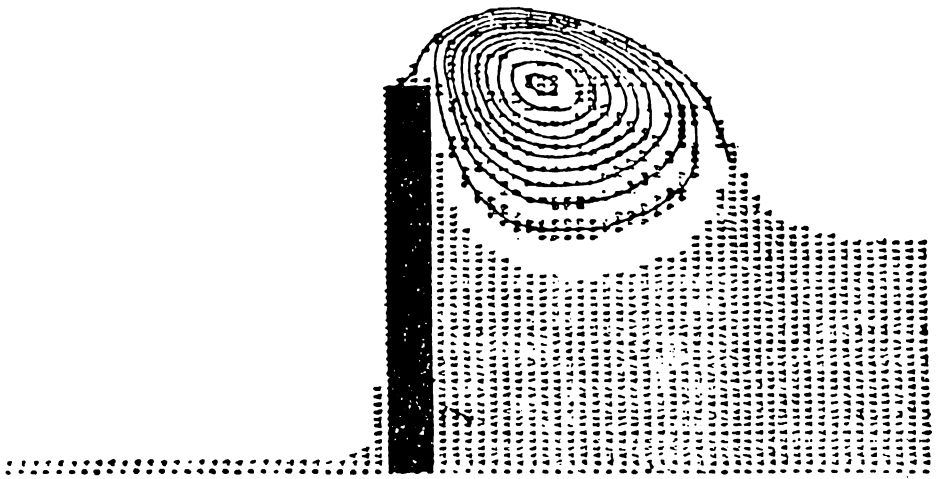

Fig. A.6(b) Laminar flow past a normal flat plate (half-field)—after the first command MASK.

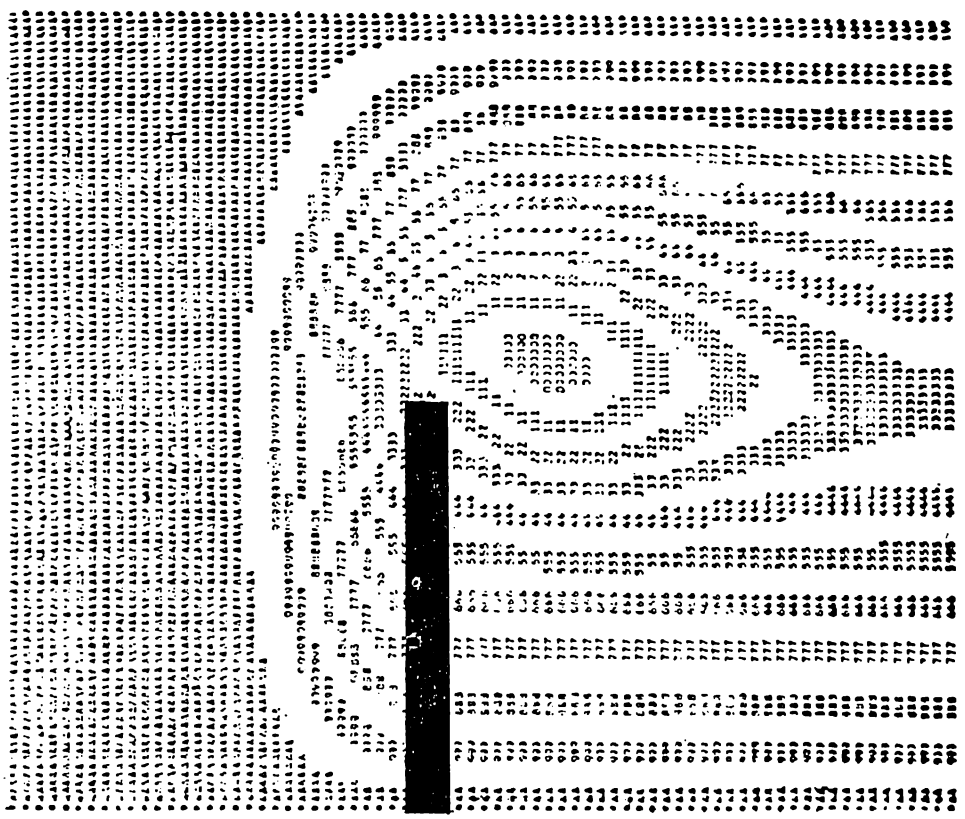

Fig. A.6(c) Laminar flow past a normal flat plate (half-field)—after command RELATE.

Fig. A.7(a) Laminar flow past a circular cylinder (half-field)—after command MASK.

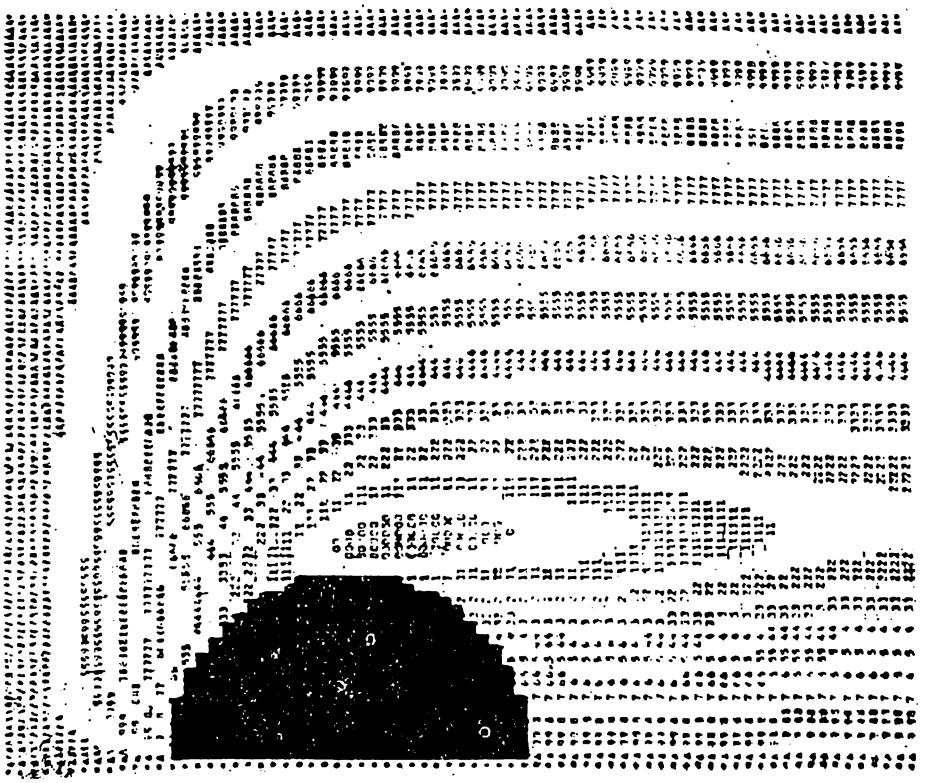

Fig. A.7(b) Laminar flow past a circular cylinder (half-field)—after command RELATE.

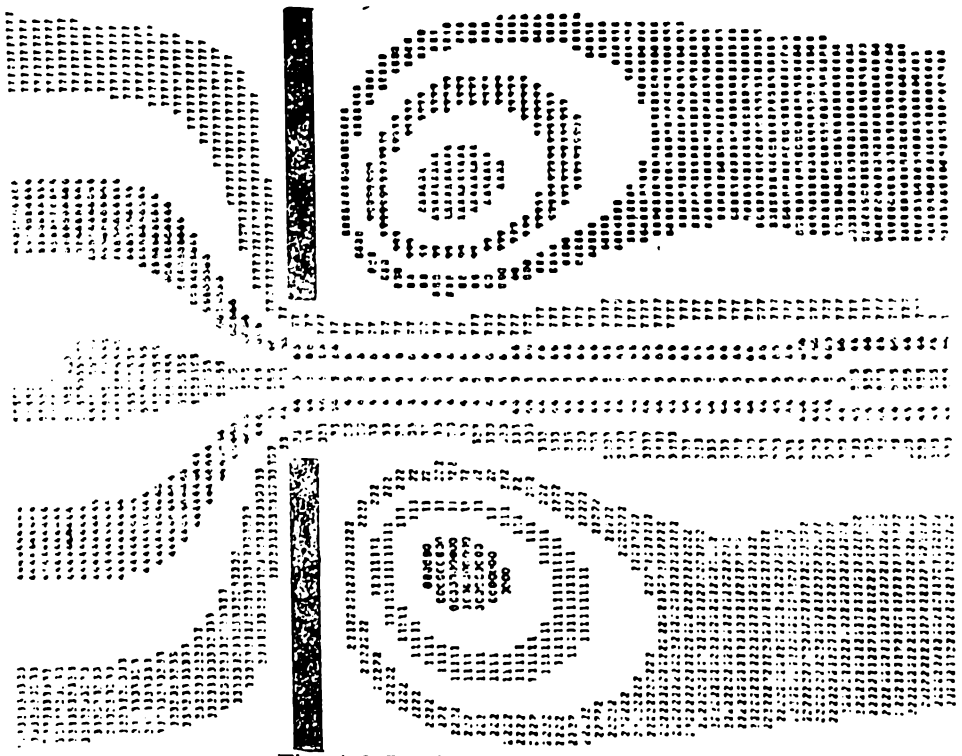

Fig. A.8 Laminar flow past a slit

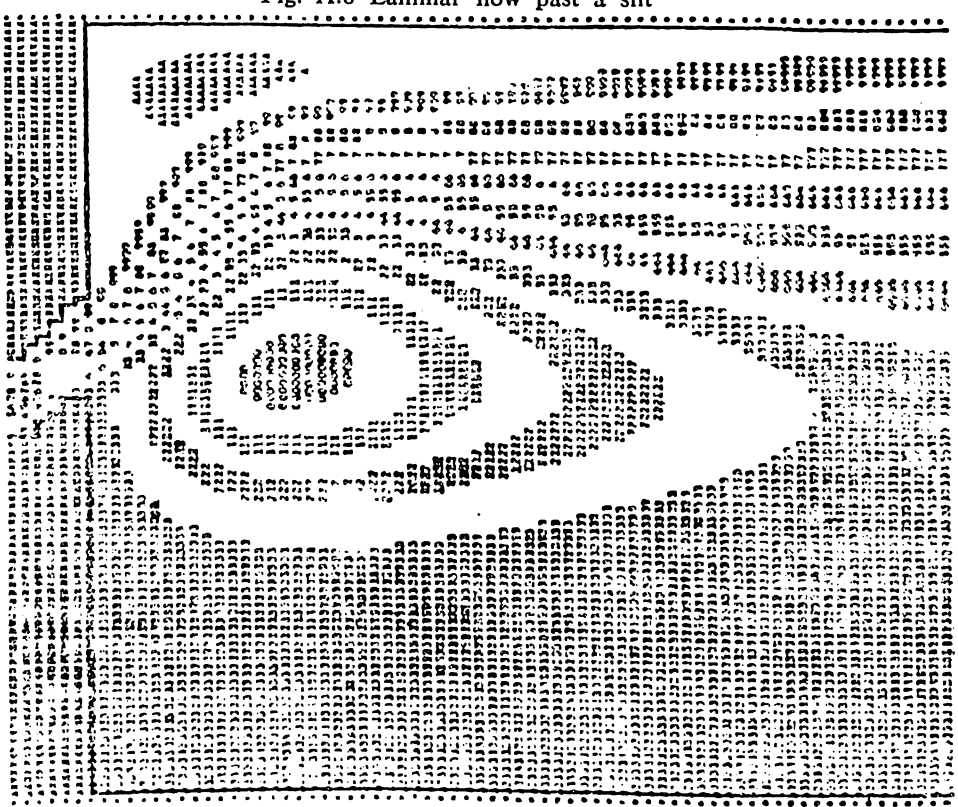

Fig. A.9 Confined oblique laminar jet