in terms of the original function $f(x)$. This rule contains in general one more point than the corresponding Gauss rule since it contains a nonzero weight for $f(q)$.

### C. Numerical examples

The simple rules given here comprise convenient methods for evaluating the integrals (1). We illustrate these rules with the numerical examples shown in **Table 2**; in this table, we have set $a = 0$, $b = 1$, $q = \sqrt{\tfrac{1}{2}}$, and we compare the two-point and three-point rules (6) and (7) at a number of step sizes with the modified Gaussian rule (12).

### References


HETHERINGTON, J. H., and SCHICK, L. H. (1965). *Phys. Rev.*, Vol. 137, p. 935.
PHILLIPS, A. C. (1966). *Phys. Rev.*, Vol. 142, p. 984.


## Book Review

*Computation: Finite and Infinite Machines*, by Marvin Minsky, 1967; 317 pages. (London: *Prentice Hall International*, 84s.)

Marvin Minsky is a superb teacher and his new book a marvel of lucid exposition which one can hardly recommend too highly to anybody wishing to understand the concept of computation and the abilities and disabilities of computing machinery. The technical ideas involved are simple enough and, given an average amount of patience and the capacity to follow straightforward reasoning, we cannot fail to follow where Minsky leads, impelled by such infectious enthusiasm.

Indeed, machine and computability theory is not the abstruse subject it is often made out to be, and made. If the significance of the questions at issue is not always immediately obvious, much can be done to hasten understanding. There is all the difference in the world, for instance, between presenting the fully fledged concept of finite automaton on a plate, as it were, take it or leave it, and the careful build-up of the concept as a highly reasonable object for study which we find in the opening pages of this book. Again, when we relax the finite restriction, it is no use ignoring the heckler in the back row who naturally wants to know why we are wasting our time with infinity, although to give him an adequate reply is not easy, even for Minsky.

The classical type of "infinite machine" is of course Turing's. One of the chief problems of exposition here arises from the fact that, while it is natural to define such a machine formally by means of a matrix, the matrix itself reveals nothing of the associated algorithmic structure. We need to devise some system of flow diagrams to convey what even the simplest Turing machine is doing. Minsky's neat resolution of the difficulty is to employ diagrams linking "left-moving" and "right-moving" states. (For every Turing machine there is an equivalent machine with such "directed" states.) Equally admirable is the way he then steers us through the crucial sequence of argument that shows how every action of a Turing machine may be mirrored in operations of a certain kind on the non-negative integers and precisely in what sense we may then deduce that a "universal" machine exists.

Since Turing's day, much consideration has been given to simple universal machines of quite different form from his. Some of these—Minsky terms them "program machines"—are closer to "real" computers in their types of instruction and use of separately accessible registers. Others have been developed from the symbol-manipulation systems of Emil Post. These, particularly "tag", have long been among Minsky's special research interests and their directness and simplicity which "steps around arithmetic" makes them particularly illuminating.

It is now possible to demonstrate, as Minsky does, that many of these systems are in some sense equivalent, but this very equivalence involves their sharing essential limitations which render various classes of problem for all time insoluble. The solubility status of *specific* problems, on the other hand, remains an unsatisfactorily "grey area" which Minsky tentatively explores in some of the most interesting sections of the book. Could, for example, Fermat's Last Theorem be actually proved insoluble? This is a field in which ingenious arguments abound but it is particularly difficult to access their logical validity.

The volume is elegantly produced, and a real pleasure to use. If printing errors are common (Why?) in no case do they handicap the reader to any serious extent except possibly in the definition and illustration of "Jump unless equal" (page 208). The author sets us problems of all grades of difficulty, includes solutions of a selection, and adds wise words on human problem-solving in general. It is all excellently carried out, though I should query the solutions of problems 4.2–1, 12.3–1 and 12.3–2.

According to the blurb, this book is "an essential tool in the hands of anyone whose professional range encompasses computers and their use". One wonders how far Minsky himself would endorse that statement. It is of course true that many of the geniuses of the computer from Turing to von Neumann have been deeply concerned with the ideas here discussed. Yet in the present state of the art the practical impact of these ideas is surely negligible. Only at some distant date may, as von Neumann believed, the development of computers come to depend crucially on developments in the theory of computability. Only then will the concepts of this book at last be seen to lie in the mainstream of computer development itself.

MICHAEL BELL (London)