REVIEWS OF BOOKS AND PAPERS IN THE COMPUTER FIELD

etc.). Thus, there would be fifteen additional parameters to describe a "compile, load, and go" job. There are an additional eleven parameters associated with every data set that is to be referenced by a job step (volume of data, device number or device class, variance of data requirement rate, etc.). Unfortunately, there is little mention of the input data which the user must provide to the job generator in order for the desired job stream to be constructed for the simulation. Hence, the reader is unable to determine whether or not he could supply the information necessary for application of the model in his own particular problem situation.

Although the paper does not report any experimental or test case results, it does contain illustrations of the twelve sections found in the simulation's output or summary report. Each section reflects one of the basic classes of statistics that are collected during the run (turnaround times, throughput, queue development, device utilization, etc.). In addition to the report at the end of each simulation run, there is a facility for obtaining reports at any desired point (or points) during the course of a run. It is also possible to obtain a report showing a time-stamped trace of the movements of each job.

This experimental model of System/300 represents another step in our endeavor to understand more about the operation of some of today's complex general purpose computer systems. Although the model does have its limitations, it has the ability to produce a substantial amount of information at relatively low cost. Thus, as a research tool it can help present a general picture, isolating certain areas for further study via other means. Although the job generator is not emphasized in the paper, it could well be of benefit to other researchers attempting to develop computer simulation models.

NORMAN R. NIELSEN
Stanford University
Stanford, Calif.

B. GRAPHIC TERMINALS


This paper is an introduction which presents only the first layer of the makeup of graphics. While this layer is explained sufficiently and clearly, as in many tutorial papers, value judgements are studiously avoided. The primary purpose, therefore, of this review will be to place in perspective some of the alternatives presented.

The first point to be noted is the dispatch with which the author dismisses TV-like raster scan displays. It readily becomes apparent in almost any useful application that the amount of information that can be drawn with the hardware described is very limiting primarily because of flicker. Raster scan displays eliminate this problem by plotting about 10^6 points per frame compared to about 10^4 points for random scan displays. Also, TV displays are capable of halftone intensities which can greatly enhance comprehension of complex drawings. The two problems inherent in TV displays are the large amount of storage required for the video and the necessity of transforming picture descriptions from the form used by the program to a scan sequence. The latter is called scan conversion, and one way to accomplish it is with a conventional TV camera viewing a random scan display of the type described. Since most pictures do not change frame by frame, one scan converter may service many displays provided each display has its own video storage. Low-cost, high-density video storage devices in the form of serial magnetic memories will be available soon to solve the storage problem.

This paper carefully distinguishes between the picking or pointing capability of light pens and the writing or drawing capability of tablet devices. It is not made clear, however, that in most applications it is necessary to do both. Furthermore, it is becoming increasingly clear that the comparator hardware needed to allow pointing on a tablet is far less objectionable than the tracking program needed to provide a drawing capability for a light pen.

A shortcoming of this paper is that the function of the various processors and memories was not clearly described. Display controllers and local display buffers may be considered as processors and memory regardless of their simplicity or special purpose nature. The partitioning of computation and storage in most computer systems with displays is an ad hoc arrangement of the equipment which happened to be available. Given a choice, one of the most attractive arrangements is to have a single central memory available (through mapping hardware) to all processors, with the display processors having sufficient power to handle complex associatively structured (perhaps by lists) display files. The advantages of a central memory are that the memory management problems are simplified when more than one processor gets into the act and as much memory as possible is available to the time-sharing or central computing processor (Lewin's "processor").

One explanation for the slow evolution of graphics may be the lack of good tutorial papers. If so, this paper is a step in the right direction. Second, few installations have been in the graphics business long enough to develop the necessary foundations for productive software generation. These include such things as a good operating system, languages for generating and modifying complex data structures, and a knowledge of projective geometry for treating curve intersections and three-dimensional objects. Third, the most difficult task in creating an applications program is the determination of what computer dialogue the user or designer wishes. Suggestions such as, "I want the computer to help me design integrated circuit masks" are frequently heard, but to pin down what is desired in sufficient detail to write a program is usually difficult. The actual programming (in the right installation environment) may be quite simple, especially when the display instructions are imbedded in high-level languages. (Several examples are ALGOL-based LEAP, the list-processing language LISP, PL/I-based APL, and FORTRAN-based GPARK.) However, a fundamental problem still remains in being able to put together separately written programs into one package. This problem is not unique to graphic software, and high-level languages only slightly ease the agony.

In short, the paper is a good introduction to currently popular computer-driven displays. Although there have been few important innovations in recent years (two interesting recent hardware developments are described4), one can look forward to high-capacity TV displays and much more computing power in processors with high-level machine languages measured by today's standards.

G. D. HORNBUCKLE
M.I.T. Lincoln Lab.
Lexington, Mass.

C. ANALOG COMPUTER SIMULATION


Hawryszkiewycz describes a dual-mode problem-oriented simulation language. He utilizes the microprogram capability of the CIRRUS computer to simulate the function of certain analog components in the solution of differential equations. These equations are presented to the analyst in block-diagram form.
The language consists of four simulation elements and two executive commands. The simulation elements represent the analog integrator, summer, multiplier, and a device which Hawryszkiewycz calls an accumulator, to be used as the equivalent of the integrator in the solution of a nonlinear differential equation. The executive commands signify the conclusion of input and a mode change from a linear problem to a nonlinear problem.

The computational or simulation elements may be interconnected by means of the equal sign and the Hollerith symbols +, -. Additionally, the "primary" or independent variable of integration may be assigned to any integrator/accumulator by means of the asterisk (*).

The problem-oriented source program is then converted to machine-language code by means of direct translation to blocks of microprograms which carry out the functions of the simulation elements and are wired into the fixed store. The significance of the wired functions is seen in the fact that this permits the use of a straight translation rather than an interpretable or compiler scheme.

Internal computation is performed by means of forward stepwise sequential methods based on polynomial approximation of extrapolated values of the variables from known functional values and derivatives. Integration is accomplished by Runge-Kutta (presumably fourth order) with fixed-step size which must be an exact multiple of two. Partial double precision is also used in the integration process.

During problem execution there is a functional section and an operational section of the processor. One step of numerical integration is carried out in the functional section and then control is transferred to the operational section which is scanned for input/output orders and manual interrupts. It becomes apparent that machine-assembly language may be used in conjunction with the problem-oriented language in order to allow the user to extend the scope of his simulation.

The use of a microprogrammed control unit and the physical alteration of the machine structure in order to enable a digital computer to solve analog problems with more efficiency is a valuable contribution to simulation techniques, and, for that class of problem which can be easily modified structurally, offers a genuine contribution to existing techniques.

One minor deficiency in the presentation was noted, and this was the lack of any explicit means of arbitrary function generation elements in the language. Presumably, Hawryszkiewycz uses his multipliers and summers to generate polynomial approximations of some of the more common functions, and the machine language capability of CIRCUS for the more difficult ones.

FRANK G. CURL
Martin Marietta Corp.
Orlando, Fla.

D. ANALOG AND HYBRID COMPUTATION

The paper by Smith is representative of a class of spacecraft-guided simulation problems wherein the body dynamics of the spacecraft are eliminated by assuming that the inertia of the body is zero. Thus rotational accelerations become infinite and attitude system response times are taken to be zero. The simulation of such problems is most generally performed entirely on digital equipment because the rate of change of the guidance function is generally of the same order as the rate of change of the trajectory parameters.

The approach presented by Smith represents a departure from the all-digital solution of this problem by utilizing hybrid equipment for the trajectory computation. A set of differential equations are formulated for particle motion in a central force field under the qualified statement that "since the distances from the center of the earth for the abort runs will be great, a spherical earth potential gravity field will be used." The potential-gravity field of the moon is not mentioned, but Smith states that "for aborts close to the earth, the gravitational harmonics should be added. . . . " The avoidance of treating the moon's gravitational potential in the force model certainly deserves a few qualifying remarks by the author.

The analysis is continued and the thrust acceleration acting on the point mass is treated as a perturbing-acceleration vector which, after integration, produces a perturbed-velocity vector and, with a second integration, produces a perturbed-position vector. The perturbed-state vector is solved for with the analog computer. The reference-state vector and the guidance equations are solved for with the digital computer.

**COMMENT ON BASIC ASSUMPTIONS**

The validity of the assumption of a central force field centered at the earth center is highly dependent on the position of the spacecraft in cislunar space. For burning times which are short and thrust accelerations which are high, this is a completely adequate gravitational model. To suggest as Smith does that this assumption be used in cislunar space with low-thrust engines of the ion class is completely erroneous.

**COMMENT ON MATHEMATICAL DEVELOPMENT**

Smith's differential equations (1), (2), and (3) do not admit of a thrust acceleration acting on the stated point mass. Therefore, (4) cannot follow from the given form of (1). Equations (1), (2), and (3) should be generalized to admit of arbitrary accelerations in addition to those provided by the central force field. The constant C as given by Smith is not the universal gravity constant but it is the product of the universal gravitational constant times the sum of the masses of the gravitating bodies.

It would be appropriate if, in the development of the perturbed-acceleration equations, given in Smith's paper as (9), (10), and (11), credit were given to J. F. Encke who originated this particular technique of trajectory computation around 1820.

The development of the relation for \( \ddot{r} \) as given in Appendix B of this paper requires the evaluation of many products. It would be most informative if the amount and types of equipment used to solve for this single parameter in terms of \( \dot{x}_1 \), \( \dot{y}_1 \), and \( \dot{z}_1 \) were listed. The accuracy of the function derived in Appendix B appears to be primarily dependent on the central angle between the reference vector and the perturbed vector.

The presentation of the reference-trajectory solution in Appendix D requires the use of an iteration technique to obtain the solution of Kepler's equation for time. The repetitive use of this iteration at each update of the reference state could be avoided completely. This would be done by expanding the solution for eccentric anomaly about the initial value of eccentric anomaly instead of about the time of perigee passage. The net result would be a trigonometric series whose argument was less than 2/10 of a radian. This converges rapidly and can be truncated after a very few terms to provide an explicit function for eccentric anomaly in terms of time.

**CONCLUSION**

It is the considered conclusion of the reviewer that the implementation of Smith's problem could be considerably simplified in formulation if it were performed entirely on digital equipment. Particular care is necessary in the interpretation of this technique when utilizing thrust-acceleration levels less than ten feet per second\(^2\) or with burning times in excess of ten minutes.

EDWARD E. MARKSON
Geonautics, Inc.
Falls Church, Va. 22046