much slower logic and operating at 0.01-percent accuracy. The ADC incorporates a "look ahead" feature which automatically prepares the next subrange before it is actually required. In addition, the subranges are overlapping. These two features prevent the instabilities which are normally encountered when the input oscillates about a subrange boundary. Furthermore, an automatic "step size" selection feature adjusts the length of the bidirectional counter in inverse proportion to the rate of change of the input signal, resulting in a constant bandwidth-accuracy product. While the cost of these features is by no means negligible, the resulting improvement in performance is considerable. The PHENO ADC also provides, at no added cost, the analog equivalent of the low-order 10 bits, and a "change" pulse indicating a transition between subranges. Both are useful in hybrid function generation. It should be noted that the ability of the PHENO ADC to "fend for itself" in the presence of an oscillating argument removes one of the major objections to the use of analog breakpoint identification in hybrid function generation; namely, the danger of overloading the interrupt facilities of the digital computer.

The function generator (termed FGDAC for "function generating DAC") is based on the well-known fact that under certain circumstances, the digital representation of a function argument contains the correct lower breakpoint number in the most significant bits and the correct interpolating ratio \((x-x_i)/\Delta x\) in the least significant portion. The authors use potentiometers and an FET switching matrix driven by the breakpoint portion of the argument value to obtain the function value \(f_i\) at the lower breakpoint, as well as the difference \(\Delta f=f_{i+1}-f_i\). These quantities are fed into a summing MDAC whose digital input is the interpolating ratio. The output of this MDAC is the interpolated function value

\[ f_i + \frac{\Delta f}{\Delta x}(x - x_i). \]

By placing such a FGDAC in the feedback path of an ADC the authors also obtain an inverse-function generator.

The paper is slightly marred by the authors' attempt to defend the misconception that a multiplicity of ADCs should eliminate the need for sampling and multiplexing of analog signals in a hybrid system. Actually, as pointed out by this reviewer in another review,\(^1\) the sample-hold and multiplexing operations are functions of the characteristics of hybrid simulation and the I/O structure of the digital computer. The authors are correct, however, in implying that a multiplicity of continuous ADCs permits a natural interface to a DDA (digital differential analyzer).

Curiously, while the authors take pains to define the common term "DDA," they neglect to interpret the acronym of their own creation, PHENO.

Readers familiar with hybrid computation, AD conversion, and function generation will find the paper well worth the effort.

\(^1\) See foregoing review of "The IADIC: a hybrid computing element."

---

**B. HYBRID SIMULATION**


This is somewhat discursive but informative article on the hybrid Apollo docking simulation. The authors do not identify their objective in writing the paper until the very last sentence of the concluding section, where they state: "It is hoped that some part of the description of the problems, solutions, and experiences will be useful to others who are now involved, or who may become involved, in simulations similar to the one described." With that as an objective, much of the detailed material on problem characteristics and specifications of the docking test device could have been condensed into an introductory or parenthetical paragraph. The discussion of why a hybrid computer was selected in preference to an all-analog or all-digital system, and of the way the problem was distributed between the analog and digital portions of the hybrid system, also adds little to the paper or to previously published material on these subjects.

However, the authors have considerable working experience with hybrid systems, and have a great deal to tell us regarding problems and solutions. In large measure, therefore, they do accomplish their objective. In particular, they discuss the importance of a careful documentation procedure for recording modifications in the simulation, and of defining and implementing a setup and checkout philosophy for the overall hybrid configuration. Each of these is the type of mundane matter that is often overlooked, and yet can be critical to the success of a project. Automation of setup, checkout, and maintenance procedures, utilizing the resources of the digital computer, undoubtedly will receive increased attention as the digital portions of hybrid systems continue to grow in size, complexity, and cost. For as this happens, time lost in such activities becomes an increasingly significant economic factor. Fortunately, however, the feasibility of automating complex sequences of operations also increases as the digital computer grows. The authors review the evolution of their procedures and make some interesting comments on what they would now do if the Apollo simulation were just being started. They would develop, for example, a problem-oriented interpreter for the pot setting and static checkout calculations, the objective being to permit rapid modification of the analog program with corresponding setup program changes. This approach would also allow them to do a static check for each configuration. The interpreter would operate under the main simulation control program and would be able to communicate with the rest of the simulation by means of a symbol table; it would also be able to handle multiple analog computers. The authors also note the need for a fully automatic scaling program.

Other problems discussed include those relating to the mechanics of operating a simulation facility when the computer and simulator are housed in widely separated buildings, the matter of selecting suitable numerical integration methods, and the ever-present problem of providing adequate grounding and noise rejection between the several elements of the simulation facility.

Hopefully, future articles by these authors will reflect their experience with these proposals, and again contribute to our understanding of practical problems.

**Martin Y. Silberberg**
IBM Corporation
San Jose, Calif.


This paper deals with the problem of selecting the best approximation to an arbitrary function, using a given number of straight line segments to form the approximation. This is, of course, the basic problem of programming functions for generation by analog function generators. The author has formalized a noteworthy technique for optimal selection of breakpoints, and coupled his technique with a slope-selecting procedure based on least-squares optimization.

The subject matter is especially pertinent today with the increasing application of the card-memory type of diode function generator and its attendant need of digital autoprogramming support, as well as the formalized programming of hybrid multivarient function generation.

The author outlines a scheme for optimal slope selection which is limited to approximations having equally spaced break points. This technique, presented in 1953 by K. B. Norkin, selects an optimal set of slope increments and is based on a least-squares optimization of slope selection.
The author rightly points out that best function approximation is usually realized by unequal spacing of breakpoints, especially where the number of segments is small. He proceeds to disclose a formalized technique for selecting the best set of breakpoints for a given arbitrary function. His method provides automatic distribution of breakpoints in proportion to the reciprocal of the radius of curvature of the function \( f(X) \). The method is implemented by means of a parameter the author calls “information capacity.” For any value of the argument \( X \) he defines information capacity \( Z(X) \) as

\[
Z(X) = \int_{a}^{X} \frac{Y''}{(1 + Y'^2)^{3/2}} \, dX
\]

where

\[
Y = f(X), \quad Y' = \frac{dY}{dX}, \quad Y'' = \frac{d^2Y}{dX^2},
\]

and

\[
(radio\ of\ curvature\ of\ Y) = \frac{(1 + Y'^2)^{1/2}}{Y''}.
\]

Thus, information capacity may be thought of as a population integral, or “running total” of sorts. The author then equally divides the total information capacity \( Z(X_{AB}) \) for a range of interest \( A < X < B \) among the \( N \) increments by appropriate selection of breakpoints. This is simply done by satisfying the relationship

\[
Z(X) = Z(X_{i-1}) + \frac{Z(X_{AB})}{N}.
\]

That is to say, as the information capacity function \( Z(X) \) is scanned monotonically along \( X \), the next breakpoint \( X_i \) is selected where the information capacity function \( Z(X) \) has increased to a value equal to its value at the previously selected breakpoint plus the value of the portion of equally divided total information capacity, \( Z(X_{AB})/N \), allotted to the \( X \) bounds of each segment. Obviously, breakpoints so selected will automatically be concentrated in proportion to the severity of curvature of the arbitrary function. Unfortunately, the method tends to neglect regions of very gentle curvature, and so must be modified in such regions. This is not a serious limitation as there is no single simple relationship which will handle the total range of arbitrary function characteristics. The approach is certainly a commendable one.

My only difficulty with this paper was in attempting to relate the proposed breakpoint-selection technique to Norkin’s slope-selection technique. While the author clearly indicated that he used a modified version to optimize his slopes, my copy of the translation gave me no clue as to what his modifications actually were.

It is interesting to note that this approach parallels one which has been used by the Martin Marietta Corporation at Orlando, Fla., and presented by T. Lucas at the January 1965 Southeastern Simulation Conference. Lucas selected \( X \) values such that the maximum error within each segment span of the set of segments equaled some fixed allowable error. The allowable error was then adjusted until the set required the desired number of segments to approximate the curve. Optimizing of the slope selection was also performed on the basis of a least-squares fitting procedure.

The present segment-fitting scheme used by Martin Marietta Corporation, a simplification of Lucas’s scheme, offers greater reliability with respect to convergence to an optimum set of slopes and breakpoints. It employs a one-step fitting procedure in which best \( X \) and \( Y \) are selected simultaneously, again converging on segment number by adjustment of the allowable error, which is a linear function of slope, and performing a simple adjustment in slope to assure geometrically balanced “Fitting.” Although the latter technique was evolved somewhat empirically, the consistent convergence and superior results have recommended it strongly.

E. J. Copes
Martin Marietta Corporation
Orlando, Fla.