When a change to a computer system is evaluated under workload conditions that are not controlled, it is necessary to estimate to what extent system performance has been affected by the change and to what extent by variations in the workload. This paper describes a regression-analysis method by which such an estimate was made for a particular computer system.

Evaluating system changes under uncontrolled workloads: a case study

by H. P. Friedman and G. Waldbaum

A number of approaches have been used in evaluating how computer system changes affect performance. In one approach, in which the workload is said to be controlled, a set of benchmarks, representing a range of typical workloads, is run with and without the system change. Although the evaluation is straightforward in that differences in performance simply are reported for each benchmark, the success of this approach depends on how well the benchmarks represent real workloads. When there is significant variability in the workload, one can not help but be skeptical of "typical" workloads.

Another approach is to perform the evaluation using data on the performance of the real system under a real workload. The system is run with and without the change under a workload that is said to be uncontrolled, although the experiment frequently is designed to minimize the effect of workload variations.

It was this approach that was used in evaluating a change in the APL[1] system at IBM's Thomas J. Watson Research Center. The change was intended to provide APL users with larger storage areas—but it was expected to produce some degradation in response times as well. We wanted to quantitatively evaluate the effect of the change to ensure that the amount of degradation would be acceptable.

Bard[2] has described how the effect of the variation in the workload can be overcome by continual on-line switching among different hardware/software versions of the system. This approach was not possible in our situation, however, since the system design required that a swapping area on disk be preformatted for the maximum storage area. Nor was randomization feasible, since it was not possible to switch back and forth frequently, even off-line.

Thus it became necessary to evaluate the change by trying it on the real system, where a number of factors that affect performance were not controlled. Those factors include the APL workload, the reliability of the system, and the batch workload. Moreover, any performance degradation had to be estimated within a few days so that the system could be restored to its pre-experimental version without too much difficulty if the degradation was too great.

Evaluation of a system change under an uncontrolled workload is complicated by the fact that any change in system response may be caused by a workload variation as well as by the change in the system. Furthermore, there may be feedback effects in that the user's behavior may be influenced by the system's response as well as by their knowledge of the system change.

One technique for evaluating such a change is to use regression models (i.e., equations) that relate variables describing the system's performance to variables describing the uncontrolled workload and the system modifications. In this approach, the dynamics of the feedback are modeled not directly, but indirectly through the workload variables. This technique, used previously,[3,4,5,6] is explored further in this paper.

System changes

In the APL time sharing system, which is run at the Research Center on a System/360 Model 91 under OS/MVT, users' programs and data occupy blocks of storage called workspaces. Workspaces that are not being used are stored in libraries on disk. Workspaces that *are* being used are swapped between main storage and disk according to the terminal activity of their users as well as the terminal activity of other users on the system.[7]

The modifications to the Research Center's system involved two system parameters. One parameter, WSSIZE, specifies the maximum size of each workspace in the system. Main storage and the area on disk used for swapping are formatted into fixed-size areas that are large enough to store the maximum-size workspaces. The other parameter, INCORE, specifies the number of workspaces that can be in main storage simultaneously. Thus the value of this parameter is the highest level of multiprogramming possible in the APL system. Both WSSIZE and INCORE have values that can be either defaulted (WSSIZE = 36,000 bytes, INCORE = 3) or specified within certain limits,[1] by the installation.

In 1971, a number of APL users at the Research Center requested larger workspaces. They felt that the 36,000-byte WSSIZE then currently in use was inadequate for their work. OS batch users objected, however, because a straightforward enlarging of APL workspaces would reduce the amount of main storage available for batch jobs.

To respect the wishes of both batch and time sharing users, it was decided to reduce the number of APL workspaces in main storage from three to two, but to enlarge the workspaces from 36,000 bytes to 48K (K = 1024), provided that the expected degradation in APL performance would be acceptable. Since two 48K-byte workspaces require less main storage than three 36,000-byte workspaces, batch storage would be increased by 9,696 bytes—that is, (3 × 36,000) − (2 × 49,152)—thereby providing a benefit for batch users. Performance with two 48K-byte workspaces was expected to be poorer than with three 36,000-byte workspaces for three reasons:

- Reducing the number of workspaces in main storage would reduce the average level of multiprogramming, thereby increasing the amount of swapping.
- More time would be required for reading and writing the larger workspaces.
- Increasing the maximum size of the workspaces would spread them out on disk and increase the time required for seeking. They would become spread out mostly because the IBM 2314 cylinder used for swapping can store only two 48K-byte workspaces, compared with four 36,000-byte workspaces,[8] and also because the average number of tracks used for storing libraries increases as more users take advantage of the larger workspaces.

Data collection

These system changes were evaluated using workload and performance data that are gathered continually by the APL system.[7] No system changes had to be made to collect the data, which user programs can access via special APL functions. Data were collected at approximately the same time (4:00 p.m.) each day.

To reduce any performance variation attributable to system reliability, data were not used for any day when the system was not up for at least five hours between 8:30 a.m. and 4:00 p.m.

The data used in the regression model were collected on 35 days during the latter part of 1971. On the first 18 days we used the pre-experimental system configuration—that is, the workspace size was 36,000 bytes and there were three workspaces in main storage. Then we converted to two 36,000-byte workspaces to find out whether performance would remain satisfactory. If not, there would be no point in converting to 48K-byte workspaces, and it would be easier to reconvert to the three-36,000-byte system. After running the two-36,000-byte configuration for nine days, we determined that the degradation in performance was small enough to warrant experimenting with a workspace size of 48K and two workspaces in main storage. This configuration was run during the last eight days of the data collection period.

In a previous analysis of these data,[9] one additional day was included. That day is not included in this analysis, because we discovered that the data for that day excluded the CPU time (estimated at more than an hour) that was consumed by a program that never finished running. This time was excluded because APL updates the CPU's record of CPU time only when it has finished servicing a conversational input.

The model variables

Each of the performance and workload variables in a regression equation can be specified at a micro or macro level. In a time sharing system like APL, a variable is said to be specified at the micro level if it is associated with a single conversational input, and it is specified at the macro level if it is associated with an aggregation of conversational inputs. For example, if a model predicts the average response time of all conversational inputs, it specifies the performance variable at the macro level. And if a model predicts the response time of each conversational input, it specifies the performance variable at the micro level. Since the data collection mechanism used in this study made it impossible to know the workload corresponding to each conversational input, model variables that characterize performance and workload are specified at the macro level.

performance variables

The performance variable that is modeled is system reaction time, the time from detection of a user's carriage return until his workspace is dispatched (i.e., receives its first time slice). In this paper, regression equations are built for three points on the CDF (cumulative distribution function) of the system reaction time:

Y_1, the 50th percentile for the system reaction time; i.e., the system reaction time (in seconds) achieved or bettered by 50 percent of the inputs
Y_2, the 90th percentile for the system reaction time (in seconds)
Y_3, the 95th percentile for the system reaction time (in seconds)

System reaction time measures the effectiveness of the scheduler in dispatching service to an input, and therefore it is a good measure for evaluating changes to the WSSIZE and INCORE parameters. For most of the conversational inputs in the Research Center's APL system the system reaction time is approximately equal to the system response time (i.e., the time from the depressing of the carriage return key until the system finishes servicing the input). Response time is a more commonly used performance measure, but it is not measured by APL. Since 85 percent of the inputs require less than 1/60th second of CPU time on the Research Center's Model 91, and this CPU time is rarely interrupted for more than a few milliseconds, the difference between system reaction time and system response time is negligible for most inputs.

workload variables

The initial selection of workload variables was limited to those that are measured routinely by the APL system and that therefore reflect the knowledge of the system's designers as to the variables that might influence performance. The following workload variables, all specified at the macro level, were selected for possible inclusion in the regression models of the system reaction time:

X_3, the number of conversational inputs per hour
X_4, the percentage of CPU time consumed by all small CPU requests (i.e., requests using two or less seconds of CPU time)
X_5, the percentage of CPU time consumed by all large CPU requests (i.e., requests using more than two seconds of CPU time)
X_6, the number of large CPU requests per hour
X_7, the number of commands per hour requiring two workspaces in main storage simultaneously
X_8, the number of logons per hour

Note that these variables can be considered daily averages, since the data were collected only once each day.

system condition variables

In this experiment, the values of the system parameter INCORE were 2 and 3, and the values of the parameter WSSIZE (in terms of tracks on a 2314 cylinder) were 5 and 7. Although the number of possible system configurations with these values is four, the 3–7 configuration (three workspaces in main storage, seven-track workspace) was not considered, since it would have made less main storage available to OS users. The three remain-

Table 1 Data used in building the model

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	Y_1	Y_2	Y_3
1	0	1350	1.5	6.1	21.5	128	21	.22	0.55	0.94
1	0	1541	1.9	18.4	15.5	134	22	.31	0.92	1.61
1	0	1214	1.3	9.7	15.9	110	20	.30	0.70	1.03
1	0	1218	1.5	21.1	28.5	104	19	.29	0.72	0.97
1	0	1423	1.9	16.7	27.7	127	18	.24	0.72	1.40
1	0	1803	1.6	11.3	18.1	130	21	.26	0.66	1.04
1	0	1918	1.5	18.4	7.4	111	18	.27	0.60	0.87
1	0	2006	1.5	15.7	13.8	173	23	.26	0.64	1.02
1	0	1567	1.0	3.3	15.3	116	20	.26	0.59	0.92
1	0	1373	1.4	6.7	13.7	136	21	.26	0.64	1.12
1	0	1092	1.2	31.7	15.1	95	20	.34	1.06	1.57
1	0	1015	0.9	28.2	15.0	102	16	.32	1.24	1.88
1	0	1670	1.2	8.4	6.3	100	22	.22	0.56	0.78
1	0	1148	1.5	21.2	20.9	96	18	.26	0.75	1.35
1	0	1416	1.4	12.1	40.9	110	19	.25	0.65	1.24
1	0	1341	1.8	8.4	18.3	109	19	.24	0.61	1.17
1	0	1279	1.4	24.5	45.2	104	15	.27	0.72	1.29
1	0	1665	1.6	10.9	17.1	135	21	.26	0.59	0.91
0	0	1378	1.5	5.9	18.5	120	18	.29	0.72	1.24
0	0	1469	1.3	18.2	22.6	107	19	.31	0.84	1.26
0	0	1365	1.4	16.9	21.0	120	24	.32	0.93	1.70
0	0	1542	1.7	18.9	20.9	137	18	.29	0.71	1.17
0	0	1057	0.7	7.0	15.2	85	18	.27	0.60	0.98
0	0	1394	1.5	19.2	7.5	108	22	.26	0.62	1.20
0	0	1245	1.3	2.2	8.8	118	18	.25	0.57	0.94
0	0	1195	1.2	6.2	13.6	120	18	.27	0.65	1.21
0	0	1748	1.7	4.2	17.0	132	22	.24	0.55	0.85
0	1	1197	1.2	7.5	14.9	131	19	.39	1.43	2.57
0	1	2029	1.4	7.0	16.7	123	19	.36	0.87	1.27
0	1	2394	1.8	10.3	19.4	150	26	.42	1.08	1.78
0	1	2018	2.1	12.1	20.0	146	22	.37	1.07	1.84
0	1	1606	2.1	9.6	17.8	126	21	.31	0.76	1.21
0	1	1103	1.3	12.5	10.5	82	17	.29	0.61	0.99
0	1	1729	1.2	3.7	9.9	96	16	.28	0.69	1.08
0	1	2117	1.9	16.6	49.3	160	17	.41	1.42	2.17

ing system conditions are described by encoding two dummy system-condition variables, X_1 and X_2, as follows:

INCORE	WSSIZE	X_1	X_2
2	5	0	0
2	7	0	1
3	5	1	0

This code has the advantage, as shown in the following section, of making it easy to test the significance of any difference between the 2–7 and 2–5 conditions, as well as any difference between the 3–5 and 2–5 conditions.

Table 1 depicts the values of the system-condition, workload, and performance variables for each of the 35 days.

The evaluation

analysis ignoring workload

First, we determine, without regard to the workload, whether there are any differences in performance among the three system conditions. For each condition, Figure 1 depicts the sample CDF of the system reaction time by averaging the data over all days when the system was operated under that condition. The averages of system reaction time (in seconds) for the 50th, 90th, and 95th percentiles are:

Condition	*50th*	*90th*	*95th*
2-5	0.28	0.69	1.17
2-7	0.36	0.99	1.61
3-5	0.27	0.72	1.17

Clearly, the system reaction time is longer on days when the 2-7 condition is run than when the 2-5 or 3-5 conditions are run, and there is little difference in reaction time between days when the 2-5 condition is run and when the 3-5 condition is run. In the following sections, these conclusions are substantiated statistically and then reexamined to determine whether the difference in performance between the days when the 2-7 condition is run and the other days is due to the 2-7 condition itself or to a heavier workload under that condition.

a least squares fitted model ignoring workload

To the 35 days of data in Table 1, we fit, by least squares, the following regression equation for the system reaction time, containing only the dummy system-condition variables:

$$YF = b_0X_0 + b_1X_1 + b_2X_2$$

In this equation, X_0 is 1, X_1 is 1 for condition 3-5 and 0 otherwise, X_2 is 1 for condition 2-7 and 0 otherwise, and YF is the fitted value of Y, which is a 35×1 column vector of the observed system reaction times for the 35 days.

The least-squares estimates of b_0, b_1, and b_2, the unknown constants in the regression equations, are found by solving the normal equations, which are written below as a single matrix equation:

$$X'XB = X'Y$$

where

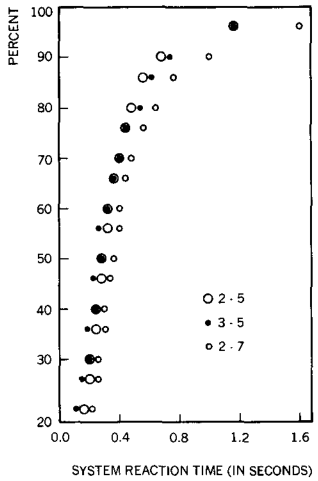
$$B = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix},$$

X is a 35×3 matrix whose columns are X_0, X_1, and X_2 respectively, and X' is the transpose of X. For the data in Table 1, the matrix equation above then becomes:

$$\begin{pmatrix} 35 & 18 & 8 \\ 18 & 18 & 0 \\ 8 & 0 & 8 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{35} Y_i \\ \sum_{i=1}^{18} Y_i \\ \sum_{i=28}^{35} Y_i \end{pmatrix}$$

Thus we see that $b_0 + b_1$ is the average of the Ys for condition 3-5, and $b_0 + b_2$ is the average of the Ys for condition 2-7. And a little algebra shows that b_0 is the average of the Ys for condition 2-5. Hence b_1 is an estimate of the difference between the 2-5 and 3-5 conditions, and b_2 is an estimate of the difference between the 2-5 and 2-7 conditions. The difference between the 3-5 and 2-7 conditions is $b_1 - b_2$.

Figure 1 System reaction time empirical CDFs for the three system conditions



To test the statistical significance of the observed differences, we have to make further assumptions about the unexplained variations. In particular, we have to assume that the unexplained variations are distributed normally, with zero mean and variance equal to a constant that can be estimated from the residuals (i.e., observed value minus fitted value). Under these assumptions, we can estimate the standard error of a coefficient and compute a t-statistic that is equal to the ratio of the coefficient to its standard error.

Ignoring workload, the t-statistics corresponding to the coefficient b_2 for the 50th, 90th, and 95th percentiles of reaction time are 4.20, 3.00, and 2.50. Large values of $|t|$ (>2 in this case) indicate that the coefficient is significant. Therefore there is strong evidence that there was a real difference between the reaction times of the 2-7 and 2-5 system conditions, thereby substantiating the conclusion drawn in the previous section. Likewise, the conclusion that there was no statistical difference between the 2-7 and 3-5 conditions is substantiated since the corresponding t-statistics for b_1 are -0.62, 0.35, and 0.004.

analysis considering workload

Although we have shown that the system reaction time corresponding to the 2-7 data is slower than that corresponding to the 2-5 and 3-5 data, the system conditions do not account for all of the difference. Therefore we must expand the analysis to find out whether the effects we observed can be accounted for by the workload. Further, we would like to see whether the variations in reaction time are explained fully by the workload variables together with the system parameters.

The regression approach to evaluating a change in a computer system requires that an equation be found that adequately relates workload and system conditions to performance. If, for the different system conditions described by such an equation, the response surfaces are parallel to one another, then the effect of the system conditions is independent of the workload. In that case it will be possible to make performance statements that are independent of the workload. However, if the response surfaces are not parallel, the effect of the system conditions is not independent of the workload, and it will be necessary to make the performance statements contingent upon some statement about the workload. Thus, it is desirable to explore the parallel regression model first.

We assume that the response surfaces for the different system conditions are parallel. In other words, we assume that the difference in performance under any two system conditions and any workload is independent of the workload (i.e., the difference is a constant). Although it would not be expected that such a parallel regression model would be valid for a computer system, it is worth exploring because of its advantages over a nonparallel regression model (i.e., performance statements independent of workload, fewer constants to be determined). Even if the model is not entirely correct, we might still be able to draw some useful conclusions from it.

For each response measure, then, we fit a simple model[10,11] of the following form:

$$YF = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_8X_8$$

Because of the way the system conditions are encoded, this model is equivalent to the following set of equations:

For the 2-5 data: $YF = b_0 + b_3X_3 + \cdots + b_8X_8$

For the 3-5 data: $YF = b_0 + b_1 + b_3X_3 + \cdots + b_8X_8$

For the 2-7 data: $YF = b_0 + b_2 + b_3X_3 + \cdots + b_8X_8$

Thus, b_0 again is an estimate of the effect of the 2-5 condition, b_1 is an estimate of the difference between the 2-5 and 3-5 conditions, and b_2 is an estimate of the difference between the 2-5 and 2-7 conditions. (The difference between the 3-5 and 2-7 conditions is estimated by $b_2 - b_1$.) Note that the other coefficients correspond exclusively to workload variables and that they are the same for the different conditions. (This is the parallel regression assumption.) Thus, this model enables us to separate the variation in performance due to workload from the variation due to system conditions. That is, it enables us to make performance statements independently of the workload.

If this model is correct, it is possible, for example, to determine whether the 2–7 condition differs statistically from the 2–5 condition by examining the t-statistic associated with the coefficient b_2. Note that this procedure takes into account the effect of the workload variables in the equation, and if the variation can be accounted for by workload alone, the t-statistic corresponding to b_2 would be insignificant.

Table 2 Results of fitting the parallel regression model

	Y_1	Y_2	Y_1
b_0	0.128	0.134	0.117
b_1	−0.022	−0.030	−0.083
b_2	0.086	0.400	0.640
$t(b_1)$	1.9	0.5	0.7
$t(b_2)$	5.6	4.9	4.3
R^2	0.78	0.71	0.66

Table 2 depicts the results of fitting the data to this regression model. (R^2, the multiple correlation coefficient squared, describes the proportion of variation accounted for by the fitted model, and $t(b_i)$ is the t-statistic for the significance of coefficient b_i.) If the model is correct, there is a real difference between the 2–7 and 3–5 conditions (estimated to be 0.086 seconds for the 50th percentile, 0.4 seconds for the 90th percentile, and 0.64 seconds for the 95th percentile) but there is no significant difference between the 2–5 and 3–5 conditions.

adequacy of the model

In the usual formulation of the statistical model for linear least-squares fitting of equations to data, it is assumed that the observed values $Y_1, Y_2, \cdots, Y_N$ of a dependent variable Y are generated by an equation of the form:

$$Y_i = b_0 + b_1 X_{i1} + \cdots + b_p X_{ip} + e_i$$

where $b_0, b_1, \cdots, b_2$ are constants to be estimated, and the e_i are independent outcomes from a normal distribution with mean zero and unknown variance σ^2. The Xs are assumed to be measured without error. The bs are estimated by the criterion of least squares. That is, they are chosen in order to minimize $\Sigma_{i=1}^{N} (Y_i - YF_i)^2$, where $YF_i = b_0 + b_1 X_{i1} + \cdots + b_p X_{ip}$.

The minimum value of $\Sigma_{i=1}^{N} (Y_i - YF_i)^2$ is called the residual sum of squares and is denoted by RSS. If the model is correct, then $RSS/(N - (p + 1))$ is an unbiased estimate of the unknown variance σ^2 of the error term. It is clear, then, that the estimate of error variance depends on the accuracy of the functional form of the equation.

To check on how well the functional form fits the data, it would be convenient to have an estimate of the error variance that does not depend on the fitted functional form. The classical requirement for this estimate—a good estimate of random error from randomized replication—could not be met in our case because we did not control the workload variables. Instead, we used the "near replicate" concept, which is introduced on page 123 of reference 11. That is, we looked for pairs of observations taken far apart in time, but under "nearly the same" X conditions. The variation in the dependent variable for these pairs of observations was used to define *another* estimate of error variance. This estimate was compared with the RSS and then was used as a basis for checking lack of fit. This procedure was implemented in the computer program[12] that was used for performing the analysis.

Another check on the adequacy of the model is to make various plots of the residuals (i.e., $Y_i - YF_i$) to check for indications of outlying observations, systematic departures from randomness, nonconstancy of variance, and nonnormality. A good exposition of the types of plots that are possible is given in Chapter 3 of reference 10 as well as in reference 11. Many of these plots are implemented in reference 12.

With the above remarks as background, we now discuss the adequacy of the models fitted for this problem. First, we found that 78 percent of the variation in reaction time ($R^2 = 0.78$) could be explained by the model in which Y is the 50th percentile of reaction time. The square root of the variance estimated from the unexplained variation was 0.027 second. This quantity is usually called the standard error of the estimate. Various plots of the residuals gave no cause for suspecting strong departures from the assumptions. In addition, an estimate of error variation from near-replicates proved consistent with the error derived from the residual sum of squares, indicating that any attempt to fit more complicated functional forms with the chosen variables would only be "overfitting."

The same patterns were present for the 90th and 95th percentile equations. The standard errors of estimates were larger for these equations, but the corresponding estimates from the near-replicates were also larger. Thus, any attempt to fit more complicated forms with these data would also be overfitting.

This leaves us with a model that has a significant amount of reaction-time variation that can be explained neither by the system conditions nor by the measured workload variables. Surely, part of the unexplained variation is due to the fact that the variables in the model are specified at the macro level (i.e., they are daily averages). However, it is possible that there are other unmeasured variables (e.g., characteristics of the batch workload) that might account for the unexplained variation in reaction time.

a more complete model

A check on the parallel regression assumption can be made by fitting separate equations for each system condition. This is equivalent to fitting a more complete model that allows the coefficients of the workload variables to be different for each system condition. The completed fitted model would look like this:

$$
\begin{aligned}
YF = {} & b_0X_0 + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_8X_8 \\
& + b_9X_1X_3 + b_{10}X_2X_3 + \cdots + b_{19}X_1X_8 + b_{20}X_2X_8
\end{aligned}
$$

where, as before, X_0 is always 1, X_1 is 1 for condition 3–5 and is 0 otherwise, X_2 is 1 for condition 2–7 and is 0 otherwise, and $X_3, \cdots, X_8$ are the workload variables. However, this was not done because the model contained too many parameters to estimate reliably from only 35 data points.

Conclusions

The results indicate that the 2–7 system condition gives a statistically poorer system reaction time than the other system conditions. However, the amount of performance degradation was not deemed serious by management.

The results also indicate that there was no statistical difference between the 3–5 and 2–5 conditions.

These conclusions are somewhat limited since the variables were specified at the macro level (i.e., averages over a day). There may still be periods within a day when the degradation in performance is more severe than that estimated by the model.

The conclusions are also conditional on the workloads observed. If the workload became "heavier" and took on values outside the ranges observed, the system differences might become much larger than those estimated. Conceivably, this can happen as users accomodate to larger workspaces.

CITED REFERENCES AND FOOTNOTE

1. *APL/360-OS and APL/360-DOS General Information Manual,* order number GH20-0850, IBM Corporation, Data Processing Division, White Plains, New York.
2. Y. Bard, "Experimental evaluation of system performance," *IBM Systems Journal* **12,** 3, 302–314 (1973).
3. G. Waldbaum, *Large Capacity Storage and Time Sharing System/360,* IBM Research Report RC-2698, IBM Thomas J. Watson Research Center, Yorktown Heights, New York (1969).
4. R. A. Watson, *Computer Performance Analysis: Applications of Accounting Data,* RAND Report R-573-NASA/PR, RAND Corp., Santa Monica, California (1971).
5. Y. Bard, "Performance criteria and measurement for a time-sharing system," *IBM Systems Journal* **10,** 3, 193–216 (1971).
6. H. F. Silverman and P. C. Yue, *The Response Time Characterization of an Information Retrieval System,* IBM Research Report RC-3796, IBM Thomas J. Watson Research Center, Yorktown Heights, New York (1972).
7. L. Breed and R. Lathwell, "The implementation of APL/360," *Proceedings, ACM Symposium on Interactive Systems for Experimental Applied Mathematics* (1967).

8. A 2314 cylinder contains 20 tracks. When formatted for APL, each track can contain a maximum of 7,248 bytes of data.[1] Workspaces that are 36,000 and 48K bytes long require storage areas of five and seven tracks, respectively (although those storage areas are capable of storing workspaces as large as 36,240 and 50,736 bytes). Since APL does not permit a swapped

workspace smaller than 21 tracks to be stored on two 2314 cylinders, a 2314 cylinder can store four five-track workspaces or two seven-track workspaces.

9. G. Waldbaum, "Evaluating computing system changes by means of regression models," *Proceedings, 1st Annual SIGME Symposium on Measurement and Evaluation,* 127–135 (1973).
10. R. Draper and H. Smith, *Applied Regression Analysis,* John Wiley and Sons, Inc., New York, New York (1966).
11. C. Daniel and F. S. Wood, with J. W. Gorman, *Fitting Equations to Data, Computer Analysis of Multifactor Data for Scientists and Engineers,* Wiley-Interscience, New York, New York (1971).
12. F. S. Wood, *The Linear and Nonlinear Least-squares Curve Fitting Programs,* program documentation, numbers 360D-13.6.008 and 360D-13.6.007, SHARE Program Library Agency, Triangle Universities Computation Center, Research Triangle Park, North Carolina.

Reprint Order No. G321-5020