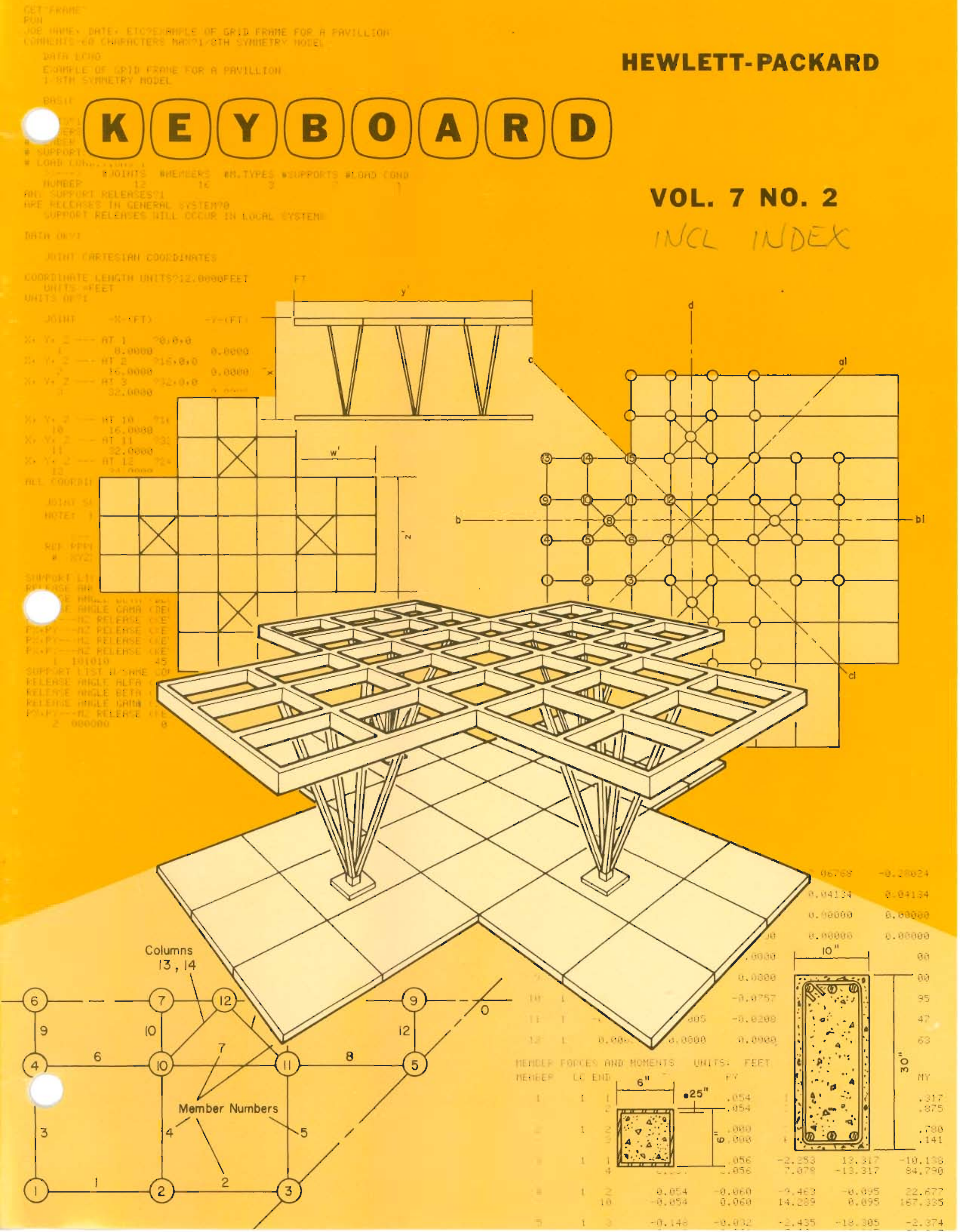
HEWLETT-PACKARD

# KEYBOARD

VOL. 7 NO. 2

INCL INDEX


GET FRAME
RUN
JOB NAME, DATE, ETC?EXAMPLE OF GRID FRAME FOR A PAVILLION
COMMENTS-60 CHARACTERS MAX?1/8TH SYMMETRY MODEL
DATA ECHO
EXAMPLE OF GRID FRAME FOR A PAVILLION
1/8TH SYMMETRY MODEL
# SUPPORT
# LOAD COND
#JOINTS #MEMBERS #M.TYPES #SUPPORTS #LOAD COND
NUMBER 12 16 3 2 1
ANY SUPPORT RELEASES?1
ARE RELEASES IN GENERAL SYSTEM?0
SUPPORT RELEASES WILL OCCUR IN LOCAL SYSTEM
DATA OK?1
JOINT CARTESIAN COORDINATES
COORDINATE LENGTH UNITS?12.0000FEET
UNITS =FEET
UNITS OK?1
FT
y'
w'
z'
x'
d
c
al
b
bl
cl
Columns
13, 14
Member Numbers
MEMBER FORCES AND MOMENTS UNITS: FEET
10"
30"
6"
6"
.25"

## TABLE OF CONTENTS

APPLICATIONS INFORMATION
FOR HEWLETT-PACKARD CALCULATORS
PUBLISHED AT P.O. BOX 301,
LOVELAND, COLORADO 80538

Editor: Nancy Sorensen Artist/Illustrator:V H.V. Andersen

# Forum

Some of our readers have suggested the need for a column in which HP calculator users could contact each other — a "clearing house" column. The opportunity to be of greater service to our readers is always attractive, so here it is. "Forum" is a communications link between you and other *KEYBOARD* readers. If you are interested in specialized programs not generally available, wish contact with others having similar job interests, or if you have a "how to" question that our other readers may be able to help you with, please address your inquiry to "Forum" in care of *KEYBOARD*. We'll publish your question and how you can be contacted in the next issue.

We hope you find this addition a useful, contributing feature.

The Editor

(We are looking for)...programs written or available for use on the 9810A pertaining to the oil and gas industry.

Particular applications of interest would be general reservoir engineering, valuations, pressure drawdown, oil and gas royalty distribution, decline curve analysis, and so forth...

We have written a few rather elementary programs for our own use that might possibly be of use to others engaged in the same line of work.

John H. Wilson II
Wilson Exploration Company
1212 West El Paso
Fort Worth, Texas 76102

## OVERVIEW

Education is faced with many problems, as is any institution whether it's business, agriculture, government, or whatever. One of these problems is time. Michael Wartell's article on how Metropolitan State College in Denver, Colorado, uses the 9830A to give and grade exams and to survey the students deals with saving time, for both student and instructor.

In the "Crossroads" feature, John Nairn discusses methods of solutions for the remaining four problems in "The Art of Science — Part 3."

In this issue, we begin a new feature, "Forum," which we hope will be interesting and useful. It's designed to strengthen communications between *KEYBOARD* readers — the primary *raison d'etre* for our magazine.

Structural analysis is a discipline requiring the juggling of many pieces of information, usually performed by a computer. In "Structural Analysis of 3-Dimensional Frames," Rick Olson discusses a new structural software package that can eliminate resorting to time-share or batch processing service bureaus and the attendant high costs.

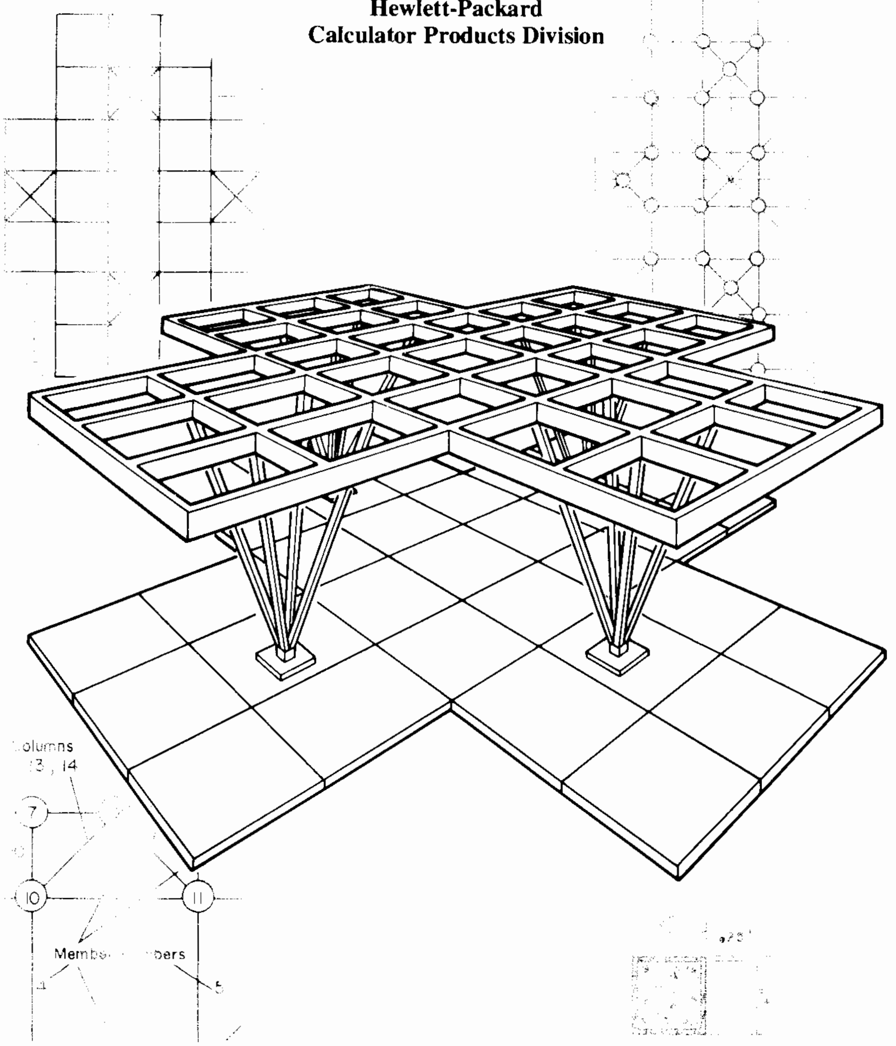
Sumitomo Special Metals Co., Inc., in Japan is one of the world's largest manufacturers of magnetic materials. The company uses a 9820A Calculator to design permanent magnets. Time and money are two of several factors that have been improved upon.

The article on Sumitomo Special Metals Co., Inc., was written by former *KEYBOARD* editor, Al Sperry, in his new capacity of Applications Coordinator. The good our readers have gained is the direct result of his efforts as editor. Al's policy has been to inform, intrigue, and involve the reader, and we will continue that policy.

Nancy Sorensen

# Structural Analysis of 3-Dimensional Frames

by Rick Olson
Hewlett-Packard
Calculator Products Division



ATI(

Structural engineers have long been searching for a truly economical, timely, and readily usable method of performing linear analysis of complex, 3-dimensional structural frames. There have been numerous attempts to develop a computerized procedure that is highly flexible and responsive to data changes and yet is standardized enough to be practical.

Conceptually, these programs are quite similar. They are designed to determine deformations, member end forces and moments, and reactions resulting for a particular load condition. The programs vary in the number of members and joints that can be analyzed, the number of loads that can be analyzed simultaneously, and other options of importance from a structural analysis point of view.

Therefore, choosing from the available methods becomes a matter of preference. The engineer must decide which method most nearly optimizes factors, such as:

- Sufficient size to handle the problem at hand, including
  Number of joints,
  Number of members,
  Number of supports,
  Number of member property types,
  Number of load conditions that can be analyzed simultaneously.
- Results within a reasonable time after the engineer begins specification of the problem, for the least cost.
- Easy review of the data for errors prior to expensive processing.
- A simple means of modifying input data, either before or after the analysis has been performed.
- Streamlined data entry that minimizes possibilities for error and the effort required by the engineer to describe his problem to the machine.
- Reducing to a minimum the number of people involved in the specification of the problem — or, in other words, limiting the number of people standing between him, the problem poser, and the machine, the problem solver.
- A minimum number of computer hassles, including JCL (Job Control Language) cards, core allocation decisions, run time limits, and other computer overhead factors.

Computer-based techniques have always tended to favor ease of solution by the computer over ease of description of the problem by the engineer. A major reason is that most computer programs are written by mathematicians and computer scientists, who have invested in extensive educations to become familiar with the language and internal workings of a computer. But most are not equally familiar with the problems a structural engineer needs to solve.

The structural engineer rarely has the computer knowledge to communicate directly with the machine, or has the direct access to do so. He needs "interpreters" to talk to the computer for him. The interpreters (and the capability of the machine) influence the balance between the computer's needs and those of the engineer. So the computer talks the interpreter's language, not the structural engineer's.

Problems encountered are data takeoff from drawings or sketches, transferring it to computer entry formatting sheets, sending the data off to be keypunched (with the ever present errors — mistaking a 2 for a Z, zero for a capital letter O, the forgotten comma), verifying data takeoff and keypunching, and so on.

## A SOLUTION

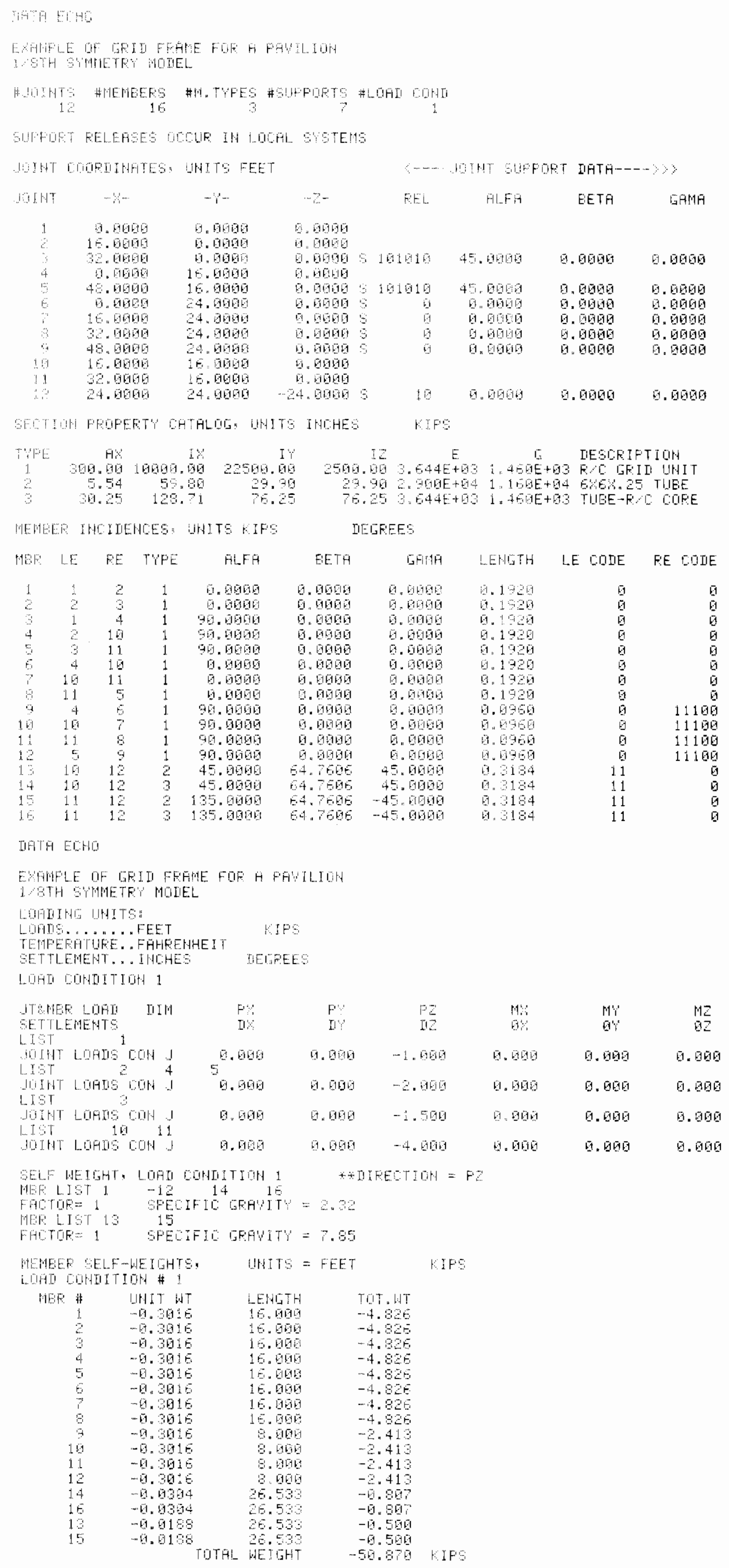
Structural engineers asked for a versatile space frame analysis package that meets the criteria established above. To execute this complex project, Hewlett-Packard turned to the consulting engineer who wrote the first structural package for the company, Mr. Louis O. Bass. Mr. Bass has many years of structural engineering experience using the currently available 3-D analysis packages on major computing systems across the country. The package he wrote for HP incorporates nearly every advantageous feature and eliminates or reduces virtually every difficulty commonly met in using 3-D packages.

The program optimizes the relationship between ease of specification and economical solution. It has been used since May of 1974 by a major consulting engineering firm in Texas on numerous complex structures. Contributed comments and opinions have helped a great deal in shaping the final product.

Some features and benefits are:

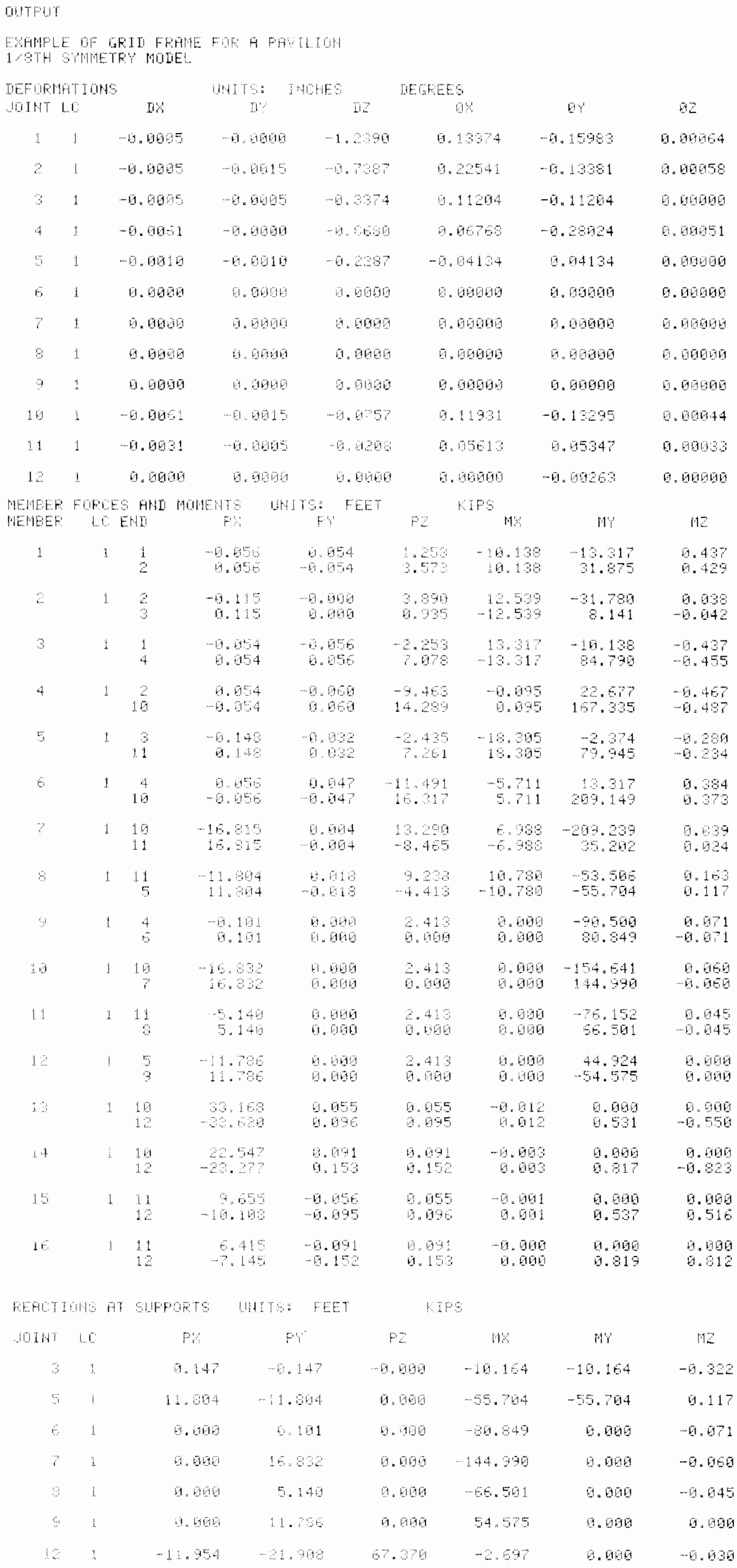
- Systems with 300 to 400 joints are easily solved.
- Simultaneous solution for the effects of from 1 to 12 load conditions.
- Analysis of from 1 to 12 combinations (with any factor) of the specified load conditions.
- Loadings may consist of joint loads; concentrated, uniform, linear, or temperature-change member loads, which may be specified in either the general system or the local member system.
- Loads induced by support motion in any or all of 6 degrees of freedom can be treated alone or in conjunction with other loads.
- Supports may be fixed or released in any manner, either with respect to the general system or in any user-specified local axis system.
- There is no limitation on the permissible bandwidth.
- Any stable set of "releases" on member ends may be specified.
- The entire solution is simulated before actual solution to ensure there is room and to provide time estimate.

**Input**



```
DATA ECHO

EXAMPLE OF GRID FRAME FOR A PAVILION
1/8TH SYMMETRY MODEL

#JOINTS  #MEMBERS  #M.TYPES #SUPPORTS #LOAD COND
    12        16         3         7          1

SUPPORT RELEASES OCCUR IN LOCAL SYSTEMS

JOINT COORDINATES, UNITS FEET              <----JOINT SUPPORT DATA---->>>

JOINT      -X-         -Y-         -Z-          REL       ALFA        BETA        GAMA

   1     0.0000      0.0000      0.0000
   2    16.0000      0.0000      0.0000
   3    32.0000      0.0000      0.0000 S 101010    45.0000      0.0000      0.0000
   4     0.0000     16.0000      0.0000
   5    48.0000     16.0000      0.0000 S 101010    45.0000      0.0000      0.0000
   6     0.0000     24.0000      0.0000 S      0     0.0000      0.0000      0.0000
   7    16.0000     24.0000      0.0000 S      0     0.0000      0.0000      0.0000
   8    32.0000     24.0000      0.0000 S      0     0.0000      0.0000      0.0000
   9    48.0000     24.0000      0.0000 S      0     0.0000      0.0000      0.0000
  10    16.0000     16.0000      0.0000
  11    32.0000     16.0000      0.0000
  12    24.0000     24.0000    -24.0000 S     10     0.0000      0.0000      0.0000

SECTION PROPERTY CATALOG, UNITS INCHES       KIPS

TYPE      AX        IX        IY          IZ         E          G     DESCRIPTION
 1     300.00 10000.00  22500.00     2500.00 3.644E+03 1.460E+03 R/C GRID UNIT
 2       5.54    59.80     29.90       29.90 2.900E+04 1.160E+04 6X6X.25 TUBE
 3      30.25   128.71     76.25       76.25 3.644E+03 1.460E+03 TUBE-R/C CORE

MEMBER INCIDENCES, UNITS KIPS          DEGREES

MBR  LE   RE  TYPE     ALFA       BETA       GAMA      LENGTH   LE CODE   RE CODE

 1    1    2    1     0.0000     0.0000     0.0000     0.1920         0         0
 2    2    3    1     0.0000     0.0000     0.0000     0.1920         0         0
 3    1    4    1    90.0000     0.0000     0.0000     0.1920         0         0
 4    2   10    1    90.0000     0.0000     0.0000     0.1920         0         0
 5    3   11    1    90.0000     0.0000     0.0000     0.1920         0         0
 6    4   10    1     0.0000     0.0000     0.0000     0.1920         0         0
 7   10   11    1     0.0000     0.0000     0.0000     0.1920         0         0
 8   11    5    1     0.0000     0.0000     0.0000     0.1920         0         0
 9    4    6    1    90.0000     0.0000     0.0000     0.0960         0     11100
10   10    7    1    90.0000     0.0000     0.0000     0.0960         0     11100
11   11    8    1    90.0000     0.0000     0.0000     0.0960         0     11100
12    5    9    1    90.0000     0.0000     0.0000     0.0960         0     11100
13   10   12    2    45.0000    64.7606    45.0000     0.3184        11         0
14   10   12    3    45.0000    64.7606    45.0000     0.3184        11         0
15   11   12    2   135.0000    64.7606   -45.0000     0.3184        11         0
16   11   12    3   135.0000    64.7606   -45.0000     0.3184        11         0

DATA ECHO

EXAMPLE OF GRID FRAME FOR A PAVILION
1/8TH SYMMETRY MODEL

LOADING UNITS:
LOADS........FEET          KIPS
TEMPERATURE..FAHRENHEIT
SETTLEMENT...INCHES      DEGREES

LOAD CONDITION 1

JT&MBR LOAD   DIM        PX         PY         PZ         MX         MY         MZ
SETTLEMENTS              DX         DY         DZ         0X         0Y         0Z
LIST       1
JOINT LOADS CON J     0.000      0.000     -1.000      0.000      0.000      0.000
LIST       2    4    5
JOINT LOADS CON J     0.000      0.000     -2.000      0.000      0.000      0.000
LIST       3
JOINT LOADS CON J     0.000      0.000     -1.500      0.000      0.000      0.000
LIST      10   11
JOINT LOADS CON J     0.000      0.000     -4.000      0.000      0.000      0.000

SELF WEIGHT, LOAD CONDITION 1       **DIRECTION = PZ
MBR LIST 1    -12    14    16
FACTOR= 1     SPECIFIC GRAVITY = 2.32
MBR LIST 13    15
FACTOR= 1     SPECIFIC GRAVITY = 7.85

MEMBER SELF-WEIGHTS,     UNITS = FEET          KIPS
LOAD CONDITION # 1
  MBR #      UNIT WT       LENGTH       TOT.WT
     1      -0.3016       16.000       -4.826
     2      -0.3016       16.000       -4.826
     3      -0.3016       16.000       -4.826
     4      -0.3016       16.000       -4.826
     5      -0.3016       16.000       -4.826
     6      -0.3016       16.000       -4.826
     7      -0.3016       16.000       -4.826
     8      -0.3016       16.000       -4.826
     9      -0.3016        8.000       -2.413
    10      -0.3016        8.000       -2.413
    11      -0.3016        8.000       -2.413
    12      -0.3016        8.000       -2.413
    14      -0.0304       26.533       -0.807
    16      -0.0304       26.533       -0.807
    13      -0.0188       26.533       -0.500
    15      -0.0188       26.533       -0.500
                  TOTAL WEIGHT      -50.870  KIPS
```

## Output



```
OUTPUT

EXAMPLE OF GRID FRAME FOR A PAVILION
1/8TH SYMMETRY MODEL

DEFORMATIONS            UNITS:  INCHES       DEGREES
JOINT LC        DX         DY          DZ          0X          0Y          0Z

    1   1    -0.0005    -0.0000    -1.2390     0.13374    -0.15983     0.00064

    2   1    -0.0005    -0.0015    -0.7387     0.22541    -0.13381     0.00058

    3   1    -0.0005    -0.0005    -0.3374     0.11204    -0.11204     0.00000

    4   1    -0.0051    -0.0000    -0.0680     0.06768    -0.28024     0.00051

    5   1    -0.0010    -0.0010    -0.2287    -0.04134     0.04134     0.00000

    6   1     0.0000     0.0000     0.0000     0.00000     0.00000     0.00000

    7   1     0.0000     0.0000     0.0000     0.00000     0.00000     0.00000

    8   1     0.0000     0.0000     0.0000     0.00000     0.00000     0.00000

    9   1     0.0000     0.0000     0.0000     0.00000     0.00000     0.00000

   10   1    -0.0061    -0.0015    -0.0757     0.11931    -0.13295     0.00044

   11   1    -0.0031    -0.0005    -0.0208     0.05613     0.05347     0.00033

   12   1     0.0000     0.0000     0.0000     0.00000    -0.00263     0.00000
MEMBER FORCES AND MOMENTS   UNITS:  FEET        KIPS
MEMBER    LC END       PX         PY         PZ        MX        MY         MZ

   1       1   1     -0.056      0.054      1.253   -10.138   -13.317      0.437
               2      0.056     -0.054      3.573    10.138    31.875      0.429

   2       1   2     -0.115     -0.000      3.890    12.539   -31.780      0.038
               3      0.115      0.000      0.935   -12.539     8.141     -0.042

   3       1   1     -0.054     -0.056     -2.253    13.317   -10.138     -0.437
               4      0.054      0.056      7.078   -13.317    84.790     -0.455

   4       1   2      0.054     -0.060     -9.463    -0.095    22.677     -0.467
              10     -0.054      0.060     14.289     0.095   167.335     -0.487

   5       1   3     -0.148     -0.032     -2.435   -18.305    -2.374     -0.280
              11      0.148      0.032      7.261    18.305    79.945     -0.234

   6       1   4      0.056      0.047    -11.491    -5.711    13.317      0.384
              10     -0.056     -0.047     16.317     5.711   209.149      0.373

   7       1  10    -16.815      0.004     13.290     6.988  -209.239      0.039
              11     16.815     -0.004     -8.465    -6.988    35.202      0.024

   8       1  11    -11.804      0.018      9.238    10.780   -53.506      0.163
               5     11.804     -0.018     -4.413   -10.780   -55.704      0.117

   9       1   4     -0.101      0.000      2.413     0.000   -90.500      0.071
               6      0.101      0.000      0.000     0.000    80.849     -0.071

  10       1  10    -16.832      0.000      2.413     0.000  -154.641      0.060
               7     16.832      0.000      0.000     0.000   144.990     -0.060

  11       1  11     -5.140      0.000      2.413     0.000   -76.152      0.045
               8      5.140      0.000      0.000     0.000    56.501     -0.045

  12       1   5    -11.786      0.000      2.413     0.000    44.924      0.000
               9     11.786      0.000      0.000     0.000   -54.575      0.000

  13       1  10     33.168      0.055      0.055    -0.012     0.000      0.000
              12    -33.620      0.096      0.095     0.012     0.531     -0.550

  14       1  10     22.547      0.091      0.091    -0.003     0.000      0.000
              12    -23.277      0.153      0.152     0.003     0.817     -0.823

  15       1  11      9.655     -0.056      0.055    -0.001     0.000      0.000
              12    -10.108     -0.095      0.096     0.001     0.537      0.516

  16       1  11      6.415     -0.091      0.091    -0.000     0.000      0.000
              12     -7.145     -0.152      0.153     0.000     0.819      0.812


 REACTIONS AT SUPPORTS   UNITS:  FEET        KIPS

 JOINT  LC          PX         PY          PZ         MX         MY         MZ

     3   1       0.147     -0.147     -0.000    -10.164    -10.164     -0.322

     5   1      11.804    -11.804      0.000    -55.704    -55.704      0.117

     6   1       0.000      6.101      0.000    -80.849      0.000     -0.071

     7   1       0.000     16.832      0.000   -144.990      0.000     -0.060

     8   1       0.000      5.140      0.000    -66.501      0.000     -0.045

     9   1       0.000     11.786      0.000     54.575      0.000      0.000

    12   1     -11.954    -21.908     67.370     -2.697      0.000     -0.030
```

- An extremely flexible routine that makes data changes easy, either before or after the solution has been run.
- The program is designed to permit restarting at key points in case of power failure, error, or the need to use the machine for a higher priority project.
- Extreme cost effectiveness compared with batch or time-share.
- Much less input required.

### [illegible]ed

The Frame Analysis Program Pac is designed to operate on a system including the following equipment: HP 9830A Calculator, Option 276; HP 9866A Thermal Page Printer or HP 9881A Line Printer Subsystem; HP 11270B Matrix ROM; HP 11274B String Variables ROM; and HP 9880B Mass Memory Subsystem. The HP 9862A Plotter with HP 11271B Plotter Control ROM is optional.

A short example of the types of problems the Frame Analysis Program Pac is designed to solve is shown at the left. Space limitations do not allow us to present the problem in its entirety, but we can show a summary of the input and output.

The problem is to analyze the grid depicted in the drawing at the beginning of this article for live and dead gravity loadings applied simultaneously. We will take full advantage of frame and load symmetry, requiring only that we analyze a 1/8th sector, repeated 8 times around.

# THE Crossroads

## THE ART OF SCIENCE — PART 3

> *"It is a truth very certain that when it is not in our power to determine what is true, we ought to follow what is most probable."*
>
> Descartes

In the last "Crossroads" article, we saw how the calculator could be used to solve three of the seven problems presented in an earlier *KEYBOARD* (Vol. 6, No. 4). Those three problems dealt with what we might call "equation solving." Once the problem had been reduced to the form of an equation, the calculator could be programmed to search for the desired solutions. The remaining four problems deal with the subject of probability, and I will attempt to put them to rest in this article.

It has been said that probability is nothing but common sense reduced to numbers. It has also been said, however, that there is no other branch of mathematics in which an accomplished mathematician is more likely to go astray than in problems dealing with probability. It is not at all uncommon to come up with two ways of approaching a probability question, each giving a different answer; and both seeming to be perfectly correct. Mathematicians A and B claim different answers to the question, and it will not do for each to simply repeat his arguments loudly and slowly. One of them must show what is wrong with the other's reasoning, which is not often an easy thing to do. To find the error in the wrong approach, it is quite helpful to know which is the correct answer, and here is where the calculator can be a very helpful tool.

All the problems answered below can (and for completeness, should) be solved analytically. But, since our purpose is to show the calculator as a tool in finding the correct numeric answers to such questions, I shall present the calculator approach to the problems. The reader interested in seeing the analytic (and sometimes very clever) methods of solution can consult the references at the end of this article.

Since the probability of an event is merely the ratio of the number of ways that event can occur to the total number of possibilities, one method of solving a probability question is by simple enumeration. The problem of the six men at the party is an example of solution by enumeration. This job is ideally suited to a calculator. If there are N men at the party, there are N! (N factorial) ways in which they can each leave with one coat. If for example, there are three men at the party, the possible distributions of coats are ABC, ACB, CBA, BAC, BCA, and CAB. Of the 3! = 6 arrangements, at least one person got his own coat in the first four. Thus, for three men, the probability is 2/3 that someone gets his own coat. In order to solve the problem for N men on the calculator, we will have the program enumerate all the arrangements of N things (in this case, just the numbers 1 through N) and count how many of the arrangements have at least one number in its proper place. The result of such a program is shown in Table 1.

**Table 1**

| N | WAYS | N! | PROBABILITY |
|---|---|---|---|
| 2 | 1 | 2 | 1/2 |
| 3 | 4 | 6 | 2/3 |
| 4 | 15 | 24 | 5/8 |
| 5 | 76 | 120 | 19/30 |
| 6 | 455 | 720 | 91/144 |

This table shows the number of ways that someone gets his own coat, the total number of arrangements (N!), and the probability (ways/N!) for N = 2, 3, 4, 5, and 6. Thus, the answer to the problem given is 91/144, or about 63.2%. The analytic approach gives the general solution as

$$P(N) = 1 - 1/2 + 1/6 - 1/24 + \ldots \pm 1/N!\,. \qquad (1)$$

The reader may verify that taking N terms of this series gives the probability for N men. You may recognize equation (1) as the series expansion for 1/e (e = the base of natural logarithms = 2.71828...). One's first reaction might be that, if there were a very large number of men at the party, it would be almost certain that someone would get his own coat. But as we take more and more terms, the series approaches closer to the value 1/e, giving a limiting probability of about 36.8%.

The next problem involves finding the probability that three lines, whose lengths are chosen at random in the range zero to one, will form a triangle. Three lines will form a triangle if no one of them is longer than the sum of the other two; that is, if $a \leq b+c$, $b \leq a+c$, and $c \leq b+a$. Thus, three lines of length 0.2, 0.2, and 0.6 cannot form a triangle, since $0.6 > 0.2 + 0.2$. This problem can be solved analytically by calculus methods, or by noting that only those points that fall inside of a tetrahedron inscribed in a unit cube satisfy the conditions and finding the volume of that tetrahedron.

To solve the problem on a calculator, we cannot enumerate the solutions, as we did in the last problem, since the number of possibilities is infinite (or, to be more precise, since we are dealing with a machine that has a finite set of numbers it can represent, let us say, a bunch). Nevertheless, we can make use of a technique called Monte Carlo evaluation, named after the business locale of the same name. According to this method, if we generate a large number of sets of three lines of random lengths and keep a count of the number of sets that form a triangle (i.e., satisfy the above conditions) and the total number of trials, then the ratio of these two numbers approaches the probability we are seeking as the number of trials gets larger. When we write the program and let it run for several thousand cases, we find that the probability approaches 1/2. If the probability doesn't approach 1/2 for this particular problem, our random number generator is not picking numbers with a uniform distribution over the range (0,1). Problems such as this can often serve as test cases for random number generators. I shall have more to say about random number generators in a later article.

In this last problem, our program found an answer that approached 1/2 as the number of trials increased. We suspect that the answer is some simple fraction and guess 1/2 based on the Monte Carlo results (although we cannot be sure until we solve the problem analytically). Difficulties can arise from this method, however, since the results approach, but never actually reach, the true solution.

In the problem of the dice, the Monte Carlo results are given in Table 2. These results show that the game is most likely to end on the fourth throw. However, all we can really be sure of from these results is that the probability is a bell-shaped curve, with a maximum somewhere around the fourth throw. We have no guarantee that the probability for the third throw might not catch or even pass that for the fourth throw if we make more trials. The results are too close! If we try to solve the problem analytically, the reasoning might go something like this: The game can't end on the first throw, so P(1)=0. The game will end on the second throw if I match the number showing from the first throw, so P(2)=1/6.

(continued on page 9)

HEWLETT-PACKARD

# KEYBOARD

# cumulative index

## volumes 4, 5 and 6

## AUTHOR INDEX

NOTE:

| | | |
|---|---|---|
| A | = | Article |
| P | = | Program |
| PT | = | Programming Tip |
| TC | = | Teacher's Corner |

ASHCROFT, D.F.
Extending 9820A Definable Functions, PT, Vol. 6, No. 1, p 23

BEAUJEAN, J.A.
Clearing Model 10 Data Registers, PT, Vol. 5, No. 3, p 21

BLAZEK, C.R.
Valuation with MRA, A, Vol. 6, No. 5, p 4

BORGOGNO, J.P.
Extending Model 10 Lagrangian Interpolation, PT, Vol. 4, No. 4, p 30
Lagrangian Interpolation, P, Vol. 4, No. 2, p 36

BUTLER, R. and TURNER, F.
Demonstration Stops (9100A/B), PT, Vol. 4, No. 2, p 36

BUTTERWORTH, W. J.
Model 10 Data Printout, PT, Vol. 4, No. 2, p 37
$x^y$ (9810A), PT, Vol. 6, No. 1, p 23

CAMPANINI, R.
Eliminating RND Predictability (9830A), PT, Vol. 6, No. 6, p 13

CARDOT, C.
Computer-Aided Intuition in Walsh Functions Research, A, Vol. 5, No. 4, p 14
9100 Step-Saving Routines, PT, Vol. 4 No. 3, p 19

CONLEY, R.W.
Three Dimensional Transformation Subroutines for the Model 10, P, Vol. 5, No. 1, p 13

DAWDY, P.A.
One-Line Averaging — 9820A, PT, Vol. 5, No. 4, p 20

DEFFEYES, R.J.
9810A Takes Over Swim Meet Officiating, A, Vol. 6, No. 1, p 18

DEMME, R.P.
Rounding to Nearest Fractions (9830A), PT, Vol. 6, No. 5, p 14

DOBMEIER, J.T.
Ballistic Trajectory Program, TC, Vol. 4, No. 2, p 31

FEIGLIN, D.
Analysis of Gamma Camera Imaging Data Using a Programmable Calculator, A, Vol. 6, No. 6, p 4

FLIPPIN, J.B.
The Nation's First Coin-Operated Computer, A, Vol. 4, No. 1, p 8

FORD, E.T. and HEACOCK, E.L.
Lunar Module Landing Simulator, P, Vol. 4, No. 2, p 21

FRIEDMAN, B.A.
The Model 30 in Business Applications, A, Vol. 6, No. 1, p 14

FUCHS, R.B. and HOLCROFT, J.W.
The HP 9810A at San Francisco General Hospital, A, Vol. 6, No. 3, p 4

GANGI, A.F.
Arctan Between ± 180 Degrees — 9820A, PT, Vol. 5, No. 4, p 22
Model 20 Integer X Without Math ROM, PT, Vol. 5, No. 1, p 22
Transferring 9820A Program Lines, PT, Vol. 5, No. 3, p 22

GITCHELL, K.R.
Critical Shaft Speed and Natural Frequency of Complex Beams, P, Vol. 4, No. I, p 19

GLADWIN, A.S.
Extending Definable Function Key to Any Number of Functions, PT, Vol. 5, No. 4, p 20

GLASSER, L.
Economical "If Y = 0" Test — Model 10, PT, Vol. 5, No. 3, p 22

GOODE, C.D.
Sequential "If" Conditions (9810A), PT, Vol. 4, No. 2, p 36

HARDENBERGH, H.W.
Chebyshev Low — Pass Filter Design, Unequal Terminations, Vol. 4, No. 3, p 6

HAUSRATH, A.S.
Model 10 Non-Zero Data Printout, PT, Vol. 5, No. 2, p 21

HOLFORD, R.M.
Double Unary Minus (9820A/9821A), PT, Vol. 6, No. 3, p 12

HONCŮ, J.
Rectangular to Polar Coordinates (9820A/9821A), PT, Vol. 6, No. 6, p 12

KUNCAR, J.
Lettering Syntax (9820A/9821A), PT, Vol. 6, No. 6, p 12

LANGDON, J.G.
Law of Cosines (9100A/B), PT, Vol. 6, No. 5, p 14

LANIGAN, J.D. and KREAMER, K.
Assignment of Enlisted Marines, A, Vol. 5, No. 4, p 12

LEE, C.
Stock Market Analysis System, Vol. 5, No. 3, p 3

LINDAUR, J.
Pen Drop Control for 9810A, PT, Vol. 4, No. 4, p 29

MADACHY, J.S.
Integer X Without an Int X Key, PT, Vol. 4, No. 2, p 36

MAEDER, D.G.
More on Available Memory (9830A), PT, Vol. 6, No. 6, p 12

MARSH, J.S. and Smith, R.C.
Computer Holography with the 9100B, A, Vol. 6, No. 5, p 6

MARTIN, F.
String Comparisons (9830A), PT, Vol. 6, No. 5, p 14

McCAMMON, B.P. and RAINEY, D.A.
Fire Danger Index Generation, P, Vol. 5, No. 1, p 17

McCULLOUGH, C.T.
Flag 1 Switch During Program Execution (9820A), Vol. 6, No. 1, p 23

McKAGEN, O.H.
Terminating Data Entry — 9810A, PT, Vol. 5, No. 4, p 20

METCALF, M.
Velocity — Radius Relationship of Orbiting Bodies, A, Vol. 4, No. 3, p 2

MILACEK, S.
Arctg in the 0° to 360° Range (9830A), PT, Vol. 6, No. 6, p 13

MITCHELL, K.
Airplane Stability Augmentation System Design on the HP 9830A, A, Vol. 6, No. 4, p 8

NEAL, R.L., Jr.
Saving a Register When Cumulating Unordered Data, PT, Vol. 4, No. 3, p 20

NOUEL, M.A.
Correction of Entries, P, Vol. 4, No. 4, p 23

PALSMA, D.M.H.
9830A in Nuclear Medicine, A, Vol. 6, No. 6, p 10

PARRISH, A.S.
Improved Tape Identification (9865A), PT, Vol. 6, No. 1, p 24

REININGER, E.J.
The Model 10 in Physiology, A, Vol. 5, No. 2, p 14

RIPLEY, D.A.
9830A Interrupt System, PT, Vol. 6, No. 4, p 13

ROSS, B.B.
Multiple Branching (9100A/B), PT, Vol. 6, No. 5, p 14

SACCO, A.
Frame with Several Floors and Spans, P, Vol. 4, No. 2, p 17

SASSMAN, R.W.
Numerical Integration Using Gaussian Quadrature, P, Vol. 4, No. 3, p 17

SCHACHER, D.L.
Basic Integer to Algebraic (9820A/9821A), PT, Vol. 6, No. 4, p 12
"Do" Loops (9820A/9821A), PT, Vol. 6, No. 4, p 12
Faster Integer Powers (9820A), PT, Vol. 6, No. 1, p 23
Rect/Polar and Polar/Rect Subroutines — 9820A, PT, Vol. 5, No. 4, p 21
"Table" Identification (9820A/9821A), PT, Vol. 6, No. 4, p 13

SHAPIRO, J.N.
Least Squares Solution of M Equations in N Unknowns, P, Vol. 4, No. 2, p 11

SHAPIRO, J.N. and GANGI, A.F.
Model 20 — Multiple Execution of Single Line, PT, Vol. 4, No. 3, p 19

SHAPIRO, J.N. and JONES, D.
Harmonic Analysis, P, Vol. 4, No. 4, p 15

SHAPIRO, J.N. and WOODWARD, R.J.
Routine for Equal Dash Length, PT, Vol. 6, No. 2, p 28

SULLIVAN, D.
9800 Program Verification, PT, Vol. 5, No. 2, p 20

TAKAKUBO, K.
Addition and Subtraction of Octal Integers, P, Vol. 4, No. 1, p 25

TANNER, K.
The Number $\pi$ Determined by Monte Carlo Methods, TC, Vol. 5, No. 4, p 17

TROMMER, R.
Foolproof Data Entry Line (9820A), PT, Vol. 5, No. 2, p 20
Divisibility Test (9820A), PT, Vol. 5, No. 2, p 20

van der ZEE, T.P.
String Variables (9830A), PT, Vol. 6, No. 6, p 13

VRÁNA, K.
HP Calculators Used by the Blind, A, Vol. 6, No. 6, p 1

WALSINGHAM, R.J.
The Model 20 as an Aid to Wreck Location, A, Vol. 5, No. 3, p 16

WEEKS, S.W.
Change 9820A Settings During Program Execution, PT, Vol. 5, No. 4, p 20

WIANT, H.R., Jr.
Summing a Series of Numbers and Determining the Proportion Each Number Is of the Total, PT, Vol. 5, No. 2, p 20

WILKINSON, K.S.
S.F. Key Programs Used in Program (9830A), PT, Vol. 6, No. 3, p 13

WILLIAMS, A.D.
Multiple 2-Point Gaussian Integration, P, Vol. 4, No. 4, p 28

WOOD, R.E.
Ge(Li) Gamma Spectrum Program for Environmental Sample Analysis, P, Vol. 5, No. 3, p 19

WOODING, N.H., Jr.
9100A/B Printout Identification, PT, Vol. 5, No. 4, p 19

WRAY, A.A.
A Chess-Playing Program for the 9810A, A, Vol. 6, No. 4, p 4

Arctan Between ± 180 Degrees — 9820A, Vol. 5, No. 4, p 22

Arctg in the 0° to 360° Range (9830A), Vol. 6, No. 6, p 13

Available 9830A Memory, Vol. 5, No. 4, p 21

Basic Integer to Algebraic (9820A/9821A), Vol. 6, No. 4, p 12

Cascading 9100B Programs with Pretested Stored Data, Vol. 4, No. 2, p 35

Change 9820A Settings During Program Execution, Vol. 5, No. 4, p 20

Clearing Model 10 Data Registers, Vol. 5, No. 3, p 21

Data Entry Termination on 9100A/B and Model 10, Vol. 4, No. 2, p 35

Demonstration Stops (9100A/B), Vol. 4, No. 2, p 26

Divisibility Test (9820A), Vol. 5, No. 2, p 20

"Do" Loops (9820A/9821A), Vol. 6, No. 3, p 12

Double Unary Minus (9820A/9821A), Vol. 6, No. 3, p 12

Economical "If Y = 0" Test — Model 10, Vol. 5, No. 3, p 22

Eliminating RND Predictability (9830A), Vol. 6, No. 6, p 12

Extending Definable Function Key to Any Number of Functions, Vol. 5, No. 4, p 20

Extending 9820A Definable Functions, Vol. 6, No. 1, p 23

Extending Model 10 Lagrangian Interpolation, Vol. 4, No. 4, p 30

Fast Circle Plot (9820A/9821A), Vol. 6, No. 3, p 12

Faster Integer Powers (9820A), Vol. 6, No. 1, p 23

Flag 1 Switch During Program Execution (9820A), Vol. 6, No. 1, p 23

Foolproof Data Entry Line (9820A), Vol. 5, No. 2, p 20

High Speed File Identification, Vol. 6, No. 4, p 12

Improved Tape Identification (9865A), Vol. 6, No. 1, p 24

Integer X Without an Int X Key, Vol. 4, No. 2, p 36

Law of Cosines (9100A/B), Vol. 6, No. 5, p 14

Lettering Syntax (9820A/9821A), Vol. 6, No. 6, p 12

Magnetic Card Versatility, Vol. 4, No. 1, p 29

Model 10 Data Printout, Vol. 4, No. 2, p 37

Model 10 Non-Zero Data Printout, Vol. 5, No. 2, p 21

Model 20 Data Storage, Vol. 4, No. 3, p 19

Model 20 Integer X Without Math ROM, Vol. 5, No. 1, p 22

Model 20 Multiple Execution of Single Line, Vol. 4, No. 3, p 19

Model 20 Stored Data Printout, Vol. 4, No. 1, p 29

Monetary Formatting (9830A), Vol. 6, No. 4, p 13

More on Available Memory (9830A), Vol. 6, No. 6, p 12

Multiple Branching (9810A), Vol. 6, No. 5, p 14

"No-Operation" Editing Aid (9820A), Vol. 4, No. 2, p 36

Null String Entry on 9830A, Vol. 6, No. 3, p 12

One-Line Averaging — 9820A, Vol. 5, No. 4, p 20

Pen Drop Control for 9810A, Vol. 4, No. 4, p 29

Rect/Polar and Polar/Rect Subroutines — 9820A, Vol. 5, No. 4, p 21

Rectangular to Polar Coordinates (9820A/9821A), Vol. 6, No. 3, p 13

Rectangular to Polar Coordinates (9820A/9821A), Vol. 6, No. 6, p 12

Reducing Forward Search Time in Cassette Applications, Vol. 5, No. 1, p 22

Rounding to Nearest Fractions (9810A), Vol. 6, No. 5, p 14

Routine for Equal Dash Length, Vol. 6, No. 2, p 28

Saving a Register When Cumulating Unordered Data in 9100A/B, Vol. 4, No. 3, p 20

Sequential "If" Conditions (9810A), Vol. 4, No. 2, p 36

S.F. Key Programs Used in Program (9830A), Vol. 6, No. 3, p 13

Simple Counter for 9100A/B or 9810A, Vol. 4, No. 3, p 19

Special Label Sequence for 9810A, Vol. 4, No. 2, p 35

Speeding 9830A Execution Time, Vol. 6, No. 3, p 13

String Comparisons (9830A), Vol. 6, No. 5, p 14

String Variables (9830A), Vol. 6, No. 6, p 13

Summing a Series of Numbers and Determining the Proportion Each Number Is of the Total, Vol. 4, No. 2, p 20

"Table" Identification (9820A/9821A), Vol. 6, No. 4, p 13

Terminating Data Entry 9810A, Vol. 5, No. 4, p 20

Transferring 9820A Program Lines, Vol. 5, No. 3, p 22

$x^y$ (9810A), Vol. 6, No. 1, p 23

9100A/B Printout Identification, Vol. 5, No. 4, p 19

9100 Step-Saving Routines, Vol. 4, No. 3, p 19

9800 Program Verification, Vol. 4, No. 2, p 20

9820A Entry Space Saving, Vol. 5, No. 4, p 22

9830A Interrupt System, Vol. 6, No. 4, p 13

Ballistic Trajectory Program, Vol. 4, No. 2, p 31

Complex Number Arithmetic, Vol. 4, No. 4, p 25

The Number π Determined by Monte Carlo Methods, Vol. 5, No. 4, p 17

Addition & Subtraction of Octal Integers, Vol. 4, No. 1, p 25

Chebyshev Low-Pass Filter Design, Unequal Terminations, Vol. 4, No. 3, p 6

Correction of Entries, Vol. 4, No. 4, p 23

Critical Shaft Speed and Natural Frequency of Complex Beams, Vol. 4, No. 1, p 19

Fire-Danger Index Generation, Vol. 5, No. 1, p 17

Fourier Series Coefficients for Unequally Spaced Data Points, Vol. 4, No. I, p 13

Frame with Several Floors and Spans, Vol. 4, No. 2, p 17

Ge(Li) Gamma Spectrum Program for Environmental Sample Analysis, Vol. 5, No. 3, p 19

General Function Plot, Vol. 5, No. 2, p 11

Harmonic Analysis, Vol. 4, No. 4, p 15

Lagrangian Interpolation, Vol. 4, No. 4, p 21

Least Squares Solution of M Equations in N Unknowns, Vol. 4, No. 2, p 11

Lunar Module Landing Simulator, Vol. 4, No. 2, p 21

Multiple 2-Point Gaussian Integration, Vol. 4, No. 4, p 28

Numerical Integration Using Gaussian Quadrature, Vol. 4, No. 3, p 17

The Game of Life, Vol. 4, No. 3, p 13

Three Dimensional Transformation Subroutines for the Model 10, Vol. 5, No. 1, p 11

A Chess-Playing Program for the 9810A, Vol. 6, No. 4, p 4

Accounts Payable with the 9880B Mass Memory System, Vol. 6, No. 1, p 6

Accounts Receivable Billing with the 9880B Mass Memory System, Vol. 6, No. 1, p 8

Airplane Stability Augmentation System Design on the HP 9830A, Vol. 6, No. 4, p 8

Analysis of Gamma Camera Imaging Data Using a Programmable Calculator, Vol. 6, No. 6, p 4

Application Contest Winners, Vol. 5, No. 3, p 18

Assignment of Enlisted Marines, Vol. 5, No. 4, p 12

Calculator Art Contest, Vol. 4, No. 3, p 9

Cassette Memory Ups Series 9800 Capability, Vol. 4, No. 2, p 14

Computer-Aided Intuition in Walsh Functions Research, Vol. 5, No. 4, p 10

Computer Holography with the 9100B, Vol. 6, No. 5, p 6

Contest Awards, Vol. 6, No. 1, p 25

Czechoslovakia Users' Club, Vol. 6, No. 6, p 14

Digitizer Unplots Data, Vol. 4, No. 2, p 5

Fast Information Input for the 9800 System, Vol. 4, No. 1, p 11

Hewlett-Packard Obsoletes Business Tables, Vol. 5, No. 4, p 6

Hewlett-Packard Series 9800 Model 21, Vol. 6, No. 2, p 6

HP Calculators Used by the Blind, Vol. 6, No. 6, p 1

HP-45 and HP-46 Advanced Scientific Calculators, Vol. 5, No. 3, p 12

HP-80 — The Time and Money Machine, Vol. 5, No. 1, p 2

International Macro Economics and the HP 9830A, Vol. 6, No. 3, p 1

Introducing a Full-Page Printer for Your Calculator, Vol. 4, No. 4, p 8

Introducing the BASIC-Language Calculator, Vol. 4, No. 4, p 3

Introducing the Mass Memory, Vol. 6, No. 1, p 2

Inventory Control with the 9880B Mass Memory System, Vol. 6, No. 1, p 11

Investment Analysis on the 9830A, Vol. 4, No. 4, p 10

Mass Memory Accounting Systems, Vol. 6, No. 1, p 5

Meet the HP Series 9800 Model 20, Vol. 4, No. 1, p 2

Model 20 Calculator Users' Club, Vol. 6, No. 1, p 22

Model 30 Contributes to Solid State Physics, Vol. 6, No. 1, p 17

Model 30 Program Pac's, Vol. 4, No. 4, p 18

Multiple Regression in Real Estate, Vol. 5, No. 3, p 9

New HP 9870A Card Reader, Vol. 6, No. 5, p 1

Office Furniture for Calculator Systems, Vol. 6, No. 4, p 6

Practical Files System Programs, Vol. 6, No. 2, p 12

Precision Encoder Plate Measurement, Vol. 5, No. 2, p 9

Project Cost Accounting, Vol. 6, No. 2, p 2

Recent Contributed Programs, Vol. 6, No. 3, p 7

Tape Operating Systems Make Data Storage Easy, Vol. 5, No. 1, p 8

The Crossroads — Boolean Algebra, Vol. 6, No. 2, p 15

The Crossroads — Calculations on the Calendar, Vol. 6, No. 3, p 8

The Crossroads — The Art of Science, Vol. 6, No. 4, p 7

The Crossroads — The Eigen-Solution of Linear Systems, Vol. 6, No. 5, p 11

The HP 9810A at San Francisco General Hospital, Vol. 6, No. 3, p 4

The Model 10 in Physiology, Vol. 5, No. 2, p 14

The Model 20 as an Aid to Wreck Location, Vol. 5, No. 3, p 16

The Model 30 in Business Applications, Vol. 6, No. 1, p 14

The Model 30 in Modern Printing, Vol. 5, No. 4, p 3

The Nation's First Coin-Operated Computer, Vol. 4, No. 1, p 8

Software Information, Vol. 6, No. 2, p 19

Southern California Surveying, Vol. 4, No. 2, p 8

Statistical Analysis at Your Desk, Vol. 5, No. 2, p 3

Stock Market Analysis System, Vol. 5, No. 3, p 3

Surveillance of Oil-Fired Power Stations, Vol. 6, No. 6, p 8

Valuation with MRA, Vol. 6, No. 5, p 4

Velocity — Radius Relationship of Orbiting Bodies, Vol. 4, No. 3, p 2

Whistler Duct Research and the HP Calculator, Vol. 4, No. 2, p 2

1974 Calculator System Application Contest, Vol. 6, No. 2, p 30

1974 Calculator System Application Contest, Vol. 6, No. 3, p 11

1974 Calculator System Application Contest: Outside U.S.A., Vol. 6, No. 5, p 8

1974 Calculator System Application Contest — Two U.S.A Winners, Vol. 6, No. 4, p 2

9810A Takes Over Swim Meet Officiating, Vol. 6, No. 1, p 18

9820A Calculator Users' Club Goes to Scandinavia, Vol. 6, No. 6, p 14

9820A Payroll Pac, Vol. 5, No. 4, p 10

9830A in Nuclear Medicine, Vol. 6, No. 6, p 10

So far, so good! Let's say that the first two dice came up a 1 and a 2. Since there are two numbers out of six that will match one of the numbers already rolled, the probability of ending on the third roll is 1/3, right? Wrong! Look at the Monte Carlo results. They may not be exact, but they aren't that far off. So, even though our test results are not accurate enough to give exact answers, they serve as a red-light indicator of fuzzy thinking! Let's rethink our analytic evaluation of P(3).

**Table 2**

| THROW | MONTE CARLO | ACTUAL |
|---|---|---|
| 1 | 0.00000 | 0.00000 = 0 |
| 2 | 0.16614 | 0.16667 = 1/6 |
| 3 | 0.27827 | 0.27778 = 5/18 |
| 4 | 0.28063 | 0.27778 = 5/18 |
| 5 | 0.18518 | 0.18519 = 5/27 |
| 6 | 0.07332 | 0.07716 = 25/324 |
| 7 | 0.01645 | 0.01543 = 5/324 |

We said that the probability of ending on the third roll was 1/3 since two numbers are showing; but this assumed that we even got to the third throw. We forgot to consider that the game might have ended on the second throw! What we really want is the probability that the game ends on the third throw AND that it did not end before the third throw. This time, P(3) = (1/3) * (1 − P(2)) = (1/3) (5/6) = 5/18 = 0.27778. Notice that if P(2) is the probability that the game ended on the second move, 1-P(2) is the probability that it did not end on the second move. In general, P(N) = (N−1)/6 * S(N−1), where S(N−1) is 1 minus the sum of all the probabilities through N−1. The last column in Table 2 gives these analytical results, which verify the Monte Carlo calculations (or are verified by the Monte Carlo results if you like your chickens before your eggs!). In any case, the moral of the story is that it is always a good idea to cross-check analytic results with numeric evaluations.

One final point remains to be made concerning machine calculations of probabilities, or of any analytic results for that matter. The last problem asks for the probability that, if I break a stick into three pieces, the three pieces will form a triangle. The problem is similar to Number 5 (the second one discussed in this article) except that instead of each piece being in the range (0,1), the sum of all three pieces is 1 (whatever the length of the stick, we will take that as our unit of length). There is a theorem in geometry that says, if I take any point inside of an equilateral triangle and draw the perpendiculars to each of the three sides, the sum of these perpendiculars is equal to the altitude of the triangle.

In Figure 1 we have drawn an equilateral triangle with unit altitude. Since the sum of the lines a, b, and c are equal to the altitude (one unit), every point in the large triangle corresponds to a way of breaking our unit stick. However, only points in the central small triangle correspond to points that can form a triangle. To see this, notice that in any of the three outer small triangles, one of the three pieces is longer than 1/2 and thus greater than the sum of the other two pieces. Now, since the area of the center small triangle is 1/4 of the large triangle which represents all possible ways of breaking the stick, the probability we are looking for is 1/4.

**Figure 1**

Let's verify this by a Monte Carlo calculation. To break our stick the first time, we generate a random number, R1, between 0 and 1, and this is the length of the first piece. We then take the remaining piece, with length 1-R1, generate a second random number, R2, and let R2(1-R1) be the length of the second piece. What is left over is the length of the third piece. All that remains is to apply the same test as in Problem 5 to see if the three pieces form a triangle. We do this for several thousand trials, count the number of triangles found and the total number of trials, and lo-and-behold we come up with a probability of 0.19315! What? It's supposed to be 1/4. Or is it? More fuzzy thinking? Not exactly. But this does illustrate a very dangerous pitfall in translating problems to programmed solutions.

The whole difficulty lies cleverly disguised in the phrase, "break a stick into three pieces." One way to do this is to randomly break the stick once, choose one of the pieces at random, and randomly break the chosen piece. This is what our program simulated. Another equally valid method is to pick two random points on the stick as "break-points" in order to obtain the three pieces. One assumes that the two methods are equivalent, since for every partition of the stick into three pieces by the first method, there is a corresponding set of break points that will give the same result under the second method. And indeed this reasoning would be correct if we were dealing with a finite number of partitions. But one must be more cautious in comparing infinite sets.

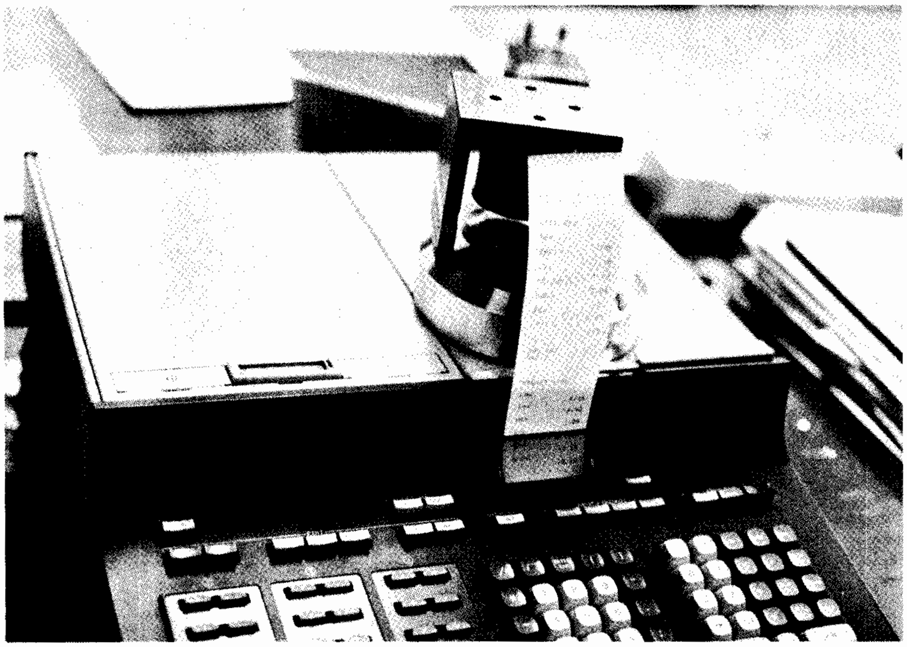
Our geometric reasoning of the probability being 1/4 is correct for the second method of breaking the stick, since R1 and R2 are each chosen over the range (0,1) independently of each other. In the first method, however, the second break point is chosen over a stick shorter than the original unit length, and the probability of forming a triangle is reduced slightly from 1/4 to 0.19315. The actual value is ln(2) - 1/2, and its derivation is a bit more complicated than that employed in finding the probability of 1/4 for the other method. The probability we are seeking depends on which method of breaking the stick we are trying to simulate in our Monte Carlo calculations. What we must be careful of is that the problem being simulated is indeed the problem we are trying to solve and not an entirely different problem disguised as an apparent twin. I would not even care to guess how many hours of computer time have been wasted by researchers who come up with unexpected results, only to find that they have sent their faithful electronic companion off solving the wrong problem. Only the programmer can decide if he has given the computer a proper representation of his problem; the computer could not care less.

H. E. Dudeney, *Amusements in Mathematics* (Dover, 1958)

Martin Gardner, *Mathematical Puzzles and Diversions* (Simon & Schuster, 1961)

Maurice Kraitchik, *Mathematical Recreations* (Dover, 1953)

# MAGNETIC CIRCUIT DESIGN
# [illegible] A



**The magnetic circuit on top of this HP 9820A Calculator was designed using a program written at Sumitomo Special Metals Co., Ltd.**

The total worldwide production of magnets is estimated at over 25,000 tons a year. Of this total, about one-half is manufactured by Japan, which may be called the ''kingdom of magnets.'' Sumitomo Special Metals Co., Ltd., a subsidiary of Sumitomo Metal Industries, Ltd., with a plant located at Yamazaki, in the Mishima district of Osaka Prefecture, Japan, produces about 35% of Japan's magnetic materials, so it ranks among the world's top magnetic manufacturers.

The Magnet Division at the Yamazaki plant manufactures magnets for loudspeakers, motor generators, communication devices, electric meters, electronic cooking ranges, and other apparatus utilizing magnets. These magnets are made of appropriate grades of KS and NKS alloys (Fe, Ni, Co, Al, and Ti) for best balance of economy, size, and desired characteristics. The earliest type of KS alloy contained about 35% cobalt, making it relatively expensive, but NKS alloys containing less cobalt have been developed for increased economy and give satisfactory performance in practical applications.

The Ferrite Division manufactures both soft ferrite, which is widely used as magnetic heads in computers, and hard ferrite. Hard ferrite is being applied increasingly in electronic cooking ranges. This division also produces piezoelectric elements used in ceramic filters, and some machine structural parts.

Although most of the output of these Sumitomo divisions is ordered by Japanese customers, an expanding volume of orders is being received from European and American firms.

## FOR DESIGN OF
## MAGNETIC CIRCUITS

All magnetic circuits are principally solved using calculations derived from Maxwell's fundamental equations of the magnetic field. It is, however, impossible to solve these 3-dimensional vector equations by hand, or by using a small calculator. Recently, several methods of solution by a large computer were reported for special applications or special permanent magnet materials. But in actual applications of permanent magnets, their geometry, dimensions, and material characteristics are too complex to give reasonable boundary conditions to these equations.

Calculation methods by large computers require enormous amounts of time, technicians, and money for programming, compared to the cost of each piece of permanent magnet and its assembly. Computer usage would cause an inordinate increase in the cost of the end product, especially when the total production quantity is small. For this reason, many designs and calculations of permanent magnets for practical applications are still carried out by hand, and designers are limited to specialists who have enough experience in determining the magnet material, the yoke material, and their configurations. Consequently, users of these designs have to be satisfied with only one answer from the specialists or authorities.

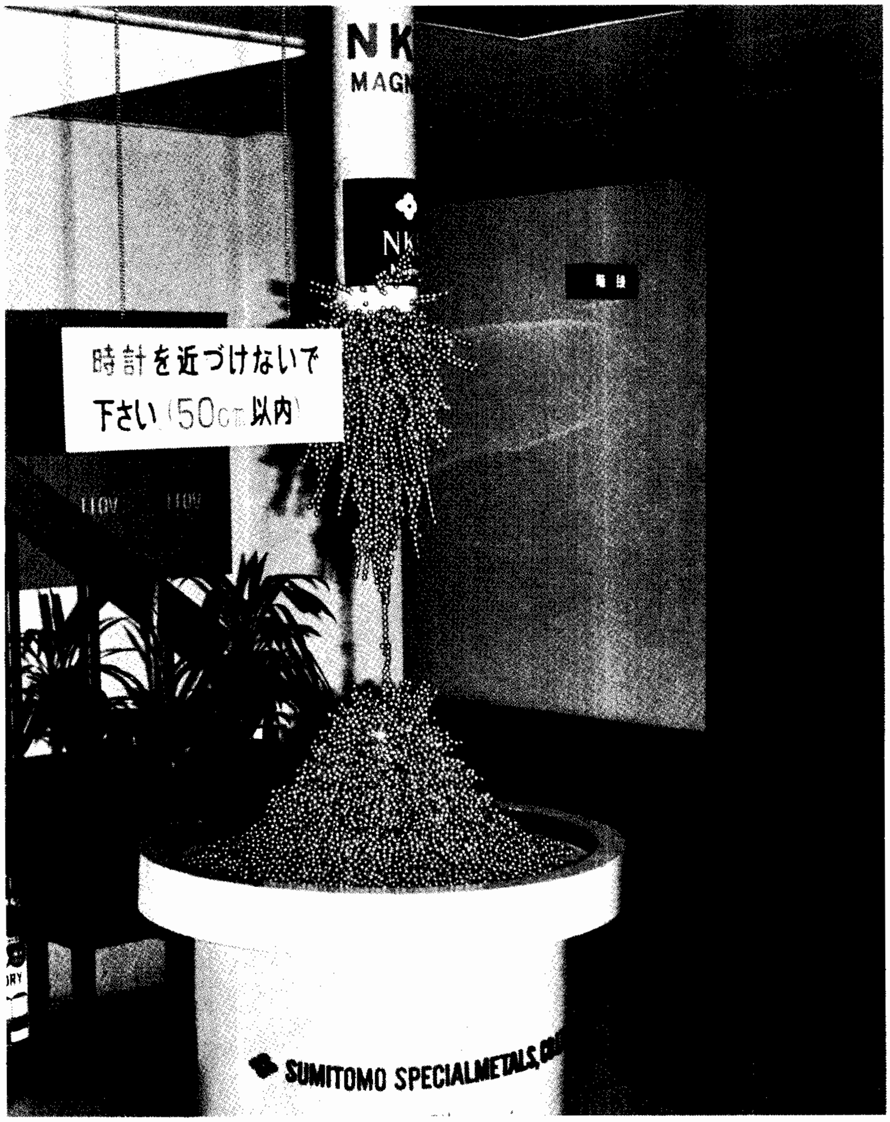
Fortunately, considerable data, experience, and know-how about permanent magnets and their applications have been accumulated by Sumitomo Special Metals Co., Ltd., from their producing magnetic materials for more than half a century. It was decided in 1973 that the calculations method by authority should be converted to an electronic calculator. The Hewlett-Packard 9820A Calculator was selected as the most profitable type for this work. Calculation formulas for the characteristics of permanent magnets and yoke materials, leakage factors, and reluctance factors of magnetic circuits were programmed for this machine. This makes it possible to design optimum magnetic circuits conforming to given specifications for several main applications of permanent magnets. One program allows calculations for several different materials. The 9820 with 429 registers allows storage of 20 characteristic magnetic curves. Any curve, which is stored as a function equation, can be called to allow recalculation of a design for a different material without a program change.

Two forms of programs are usually used for design at Sumitomo. One is the "U-program," or user program, which incorporates the user's specifications for material and dimensions to design the most effective magnetic circuit. The other is the "M-program," or manufacturer's program. This is used to check or improve existing magnetic circuit designs.

## MERITS OF CALCU[illegible]

Several benefits have been realized by Sumitomo Special Metals Co., Ltd., consequent to programming the 9820 for magnetic circuit design, including the following:

- Design time is shortened, so the users of magnets receive prompt, suitable answers and recommendations based on their requirements. A typical design, which took 8 hours by hand calculations, is now completed in about 2 hours.
- Most designs can be calculated by a person not having years of personal design experience.
- Errors occuring in hand calculations are completely eliminated.
- Not only are optimum conditions of the magnet obtained, but also the yoke design and the stability of magnetic characteristics, which could not be calculated by hand, are estimated accurately.
- Users can receive several answers centered on the optimum conditions, instead of just one answer from the authority. This enables them to compare and select the most suitable design from some viewpoint, such as cost, weight, and dimensions.



**The sign on this Sumitomo permanent magnet, which was displayed at the New York World's Fair, 1964, warns against bringing watches closer than 50 centimeters from the magnet.**

# C )UF [illegible] NATIO [illegible] RVEY
# :VA [illegible] S USING [illegible] 330A

Michael Wartell, P[illegible]
Craig Bishop, BS;
John Garbarino, BS

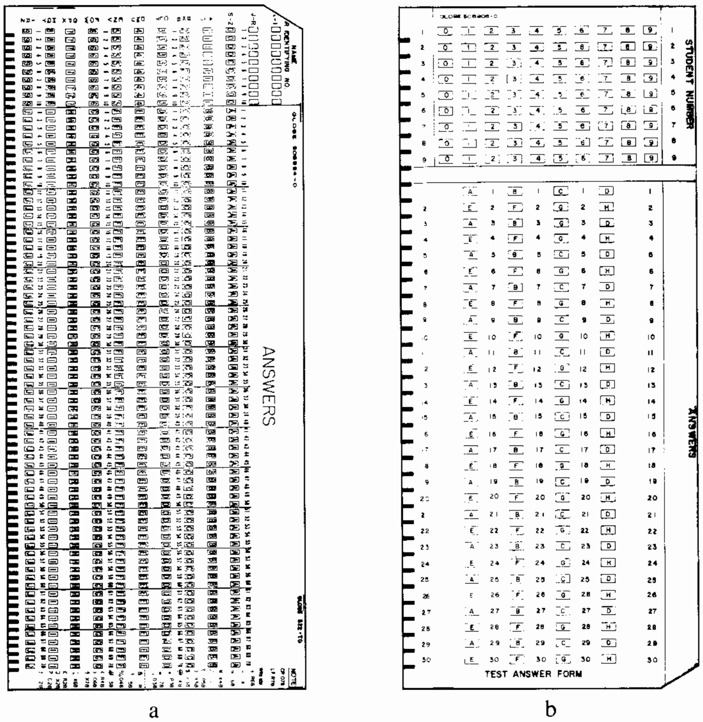


**Figure 1. Two card formats designed and used by many departments at Metropolitan State College. Figure 1a is an 80-column card, Figure 1b shows the 40-column format.**

At Metropolitan State College in Denver, Colorado, we have been making extensive use of examination and survey evaluation programs written for our Hewlett-Packard 9830A Calculator (8k memory) with 9869A Card Reader and 9862A Plotter.

In purchasing the system, we had intended that it be used to provide an automated exam grading procedure for any department in the college wishing to use it. Not only does this system relieve the faculty member of a time-consuming responsibility, but it also provides the student with accurate, immediate feedback on his proficiency level.

Since a suitable card format had not been developed for our application, we became card designers. Two formats designed and used by us are shown in Figure 1. For some tests, large numbers of questions are necessary. Attempting to retain as many question spaces as possible, we chose the 80-column format, reserving the first 10 columns for either the student's social security number or name. The last 70 columns are answer columns. This is shown in Figure 1a. Since the columns are relatively close together, enhancing the possibility that a student might make a parallax-type error when filling out the card, vertical lines have been drawn after every 5 answer columns in order to minimize this possibility. Boxes are used for answers so that they are easier to fill in. Marked boxes increase the efficiency of grading and lessen stray-mark errors. For situations where fewer questions are adequate, the card shown in Figure 1b is used. This, of course, is a 40-column format.

The types of exams we are accustomed to handling fall into three general categories.

### [illegible]eneral Multiple-Ch[illegible] [illegible] per column)

An instructor's card containing the correct answers is read into a string. Each student card is then read into two strings; one for the identifying name or number and an answer string. This reading can be accomplished with a looped input statement following a card demand statement. The answer string is then compared term by term with the instructor answer string. Output consists of student identifying code (name or social security number), student answer string, results of the comparison (x's below wrong answers, dashes below correct answers), and final grade. Strings rather than subscripted variables are used because such mistakes as blank columns or double-marked columns will not cause the program to abort. An item analysis and percent of students answering a question correctly are also calculated.

### [illegible] Type 1 E[illegible]ams on the [illegible]

Grading of these exams is similar to grading Type 1 exams, but output is slightly different. In this case, the first 10 columns are used for identification, the next 10 contain survey data (some of which is shown on the output), and the last 60 columns are test material.

### [illegible] Mark Exam[illegible] (more than [illegible]umn)

The Type 3 exam contains multiple marked columns. Reading this data from cards is a more difficult problem and must be handled using the card reader "image" mode. In the image mode, each column is read as a binary number, depending on the number of boxes marked. The binary code is translated into a string diagrammed in the following way:

■ 2
□ 3
■ 4 → 52 → 101100
■ 5 (binary)
□ 6
□ 7

Instructor and student strings are compared in a similar manner to the Type 1 and Type 2 exams.

For each type of exam, a histogram of total grades is plotted on a 9862 Plotter. A typical histogram is shown in Figure 2.

We have been using this grading system for exams from the Departments of Chemistry, Technology, Psychology, Nursing, Economics, and English. We hope to expand the service to other departments. Also, we are now doing a great deal of survey evaluation work including collation and statistics.

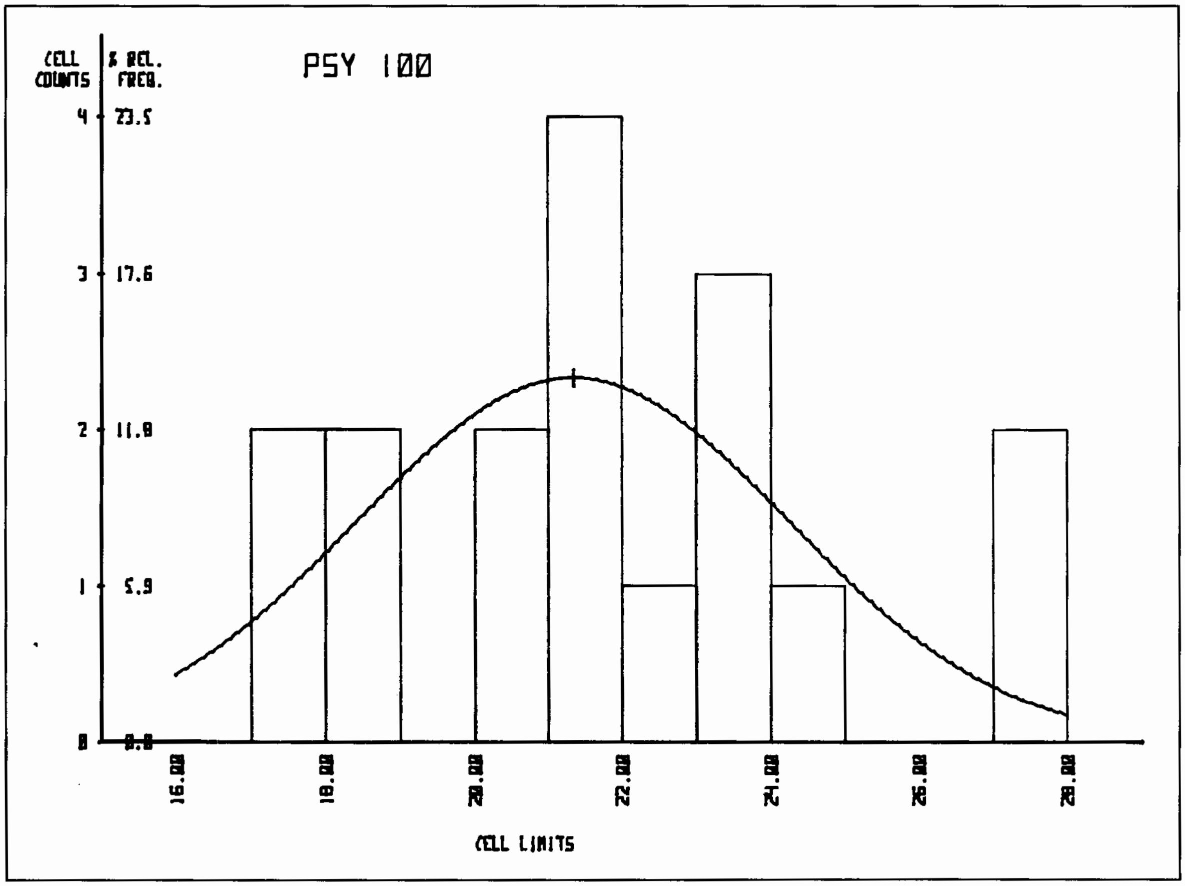


Figure 2. Histogram of the total grades in an examination.

## CURRICULUM VITAE

Michael Wartell received his BS in 1967 from the University of New Mexico, his MS in 1968, and PhD in 1971 from Yale University. His present position is Assistant Professor, Department of Chemistry, Metropolitan State College in Denver, Colorado. Dr. Wartell has had several articles published in a number of professional journals and is author of an introductory chemistry textbook.

Craig V. Bishop and John R. Garbarino assisted Dr. Wartell in writing the examination and survey evaluation programs. They both received BS degrees in Chemistry from Metropolitan State College in 1971.

# programming tips

## AVOIDING VAL FUNCTION ERRORS (9830A)

William J. Zehner of Seascope Electronics, Inc., in Lynn Haven, Florida, submits this useful programming tip.

When writing programs that perform several distinct but related functions, it is sometimes useful to arrange for the operator to branch to various routines from a command/data input statement by using specifically designated alpha commands. Using this technique, the input variable must be a string name, and if, after looking through a set of defined alpha commands, the calculator finds no recognizable match, it should assume that numeric data is present. At that point we can utilize the VAL function to extract the numeric from the input string.

The difficulty with this procedure is that if the operator misspells or accidentally uses an undefined string, the tests for defined commands will be failed, an attempt will be made to take the VAL of a nonnumeric argument, and an Error 76 will result. The accompanying program illustrates one nice way to get around this problem. Beginning at line 110, the first character of the input string A$ is compared with each of the digits 0 through 9 contained in the check string C$. If any of the 10 digits is found in A$(1,1), the program branches to exercise the VAL function. Otherwise, an Invalid Entry message is flashed, the input rejected, and the program returns to the input statement to give the operator another try. A String Variables ROM is necessary for this program.

Example:

```
10 REM STRING/VALUE WILLIAM J ZEHNER 7/31/75
20 REM TO INSURE THAT A STRING HAS A LEADING
30 REM NUMERIC VALUE BEFORE ATTEMPTING TO
35 REM EXECUTE THE "VAL" FUNCTION.
40 DIM C$[10],A$[80]
50 C$="0123456789"
60 DISP "ENTER COMMAND OR DATA";
70 INPUT A$
80 IF A$="STOP" THEN 3000
90 IF A$="ANALYSIS" THEN 2000
100 IF A$="*" THEN 500
110 REM TO CHECK FOR NUMERIC DATA
120 FOR P=1 TO 10
130 IF C$[P,P]=A$[1,1] THEN 180
140 NEXT P
150 DISP "INVALID ENTRY.  RETRY"
160 WAIT 3000
170 GOTO 60
180 V=VAL(A$)
190 DISP "OK"
200 WAIT 1000
210 GOTO 60
500 REM TO CALCULATE V.
510 V=20*LGT(PI↑2/9)
520 GOTO 190
2000 REM ANALYSIS WOULD BEGIN HERE.
3000 END
```

## STORING ALPHA ON A DATA TAPE (9830A)

John E. Barber of Cook Coggin Engineers, Inc., of Tupelo, Mississippi, shares this programming tip.

This routine is used to store alpha on a data tape without using an AP ROM. There are many ways to use this routine, but the example shown below uses an external cassette and stores the alpha in the first row of the array. With this method, your data tape can be marked in equal size files so it can be used to store more than one set of data. If all storage is alpha, the precision should be changed to save storage space.

Example:

```
10 DIM T[5,40],A$[80],B$[40],C$[4]
20 A$=" ! #$%&'()*+,-./0123456789:;<=>?ABCDEF
   GHIJKLMNOPQRSTUVWXYZ    ↑"
30 DISP "DATA FILE #"
40 INPUT D
50 LOAD  DATA #5,D,T
60 DISP "NAME";
70 INPUT B$
80 B=LEN(B$)
90 FOR I=1 TO B
100 C$=B$[I,1]
110 C=POS(A$,C$)
120 T[1,I]=C+31
130 NEXT I
140 FORMAT B
150 FOR I=1 TO 40
160 WRITE (15,140)T[1,I];
170 IF  NOT T[1,I] THEN 190
180 NEXT I
190 PRINT
200 STORE  DATA #5,D,T
210 GOTO 30
220 END
```

## ALIGNING PRINTED HEADINGS (9830A)

Our thanks to Jack L. Gehrs, Tico Office Equipment and Supplies, River Forest, Illinois, for submitting this programming tip.

When I wish to align a printed heading with a succeeding formatted output, I fill the WRITE line with numbers that equal each format statement first. I then go back and fill in the necessary heading.

Example:

```
10 FORMAT F8.0
20 WRITE (15,10)"8765432187654321876543218765432187654321"
20 WRITE (15,10)" CODE #  CODE #  CODE #  CODE #  CODE # "
   (second writing)
30 FOR I=1 TO 5
40 WRITE (15,10)I;
50 NEXT I
```

I have found that by filling in the words first and then going back and removing the numbers remaining with the space bar to provide spaces is the easiest procedure.

Printed in U.S.A
July 1,1975