APR 27 1967

COO-614-52

ICR QUARTERLY REPORT

Number 12
February 1, 1967

MASTER

THE UNIVERSITY OF CHICAGO

THE INSTITUTE FOR COMPUTER RESEARCH

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

CFSTI PRICES

The Institute for Computer Research

The University of Chicago

H.C. 300; MN 65

ICR Quarterly Report

Number 12

February 1, 1967

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Prepared under the auspices of AEC Contract No. AT(11-1)-614.

Victor H. Yngve, Principal Investigator

Staff of the Institute for Computer Research

Naomi Alton

Robert Ashenhurst

Jurgen Bounin

Dolores Brennan

Kenneth Burns

Rachel Chakrin

Elaine Chow

David Eicher

Charles Fischer

Etta Franklin

William Georgas

Robert Graves

Richard Herwitt

Leonardo Herzenberg

Paul Kosinski

Hirondo Kuki

George Kreisheimer

Vincent Kruskal

John Mackoviak

Glenn Manacher

Richard Miller

William Mixon

Lily Monheit

John Munn

Nonyne Nelson

Michael Neumann

Daniel Norman

Alex Orden

Walter Petryshyn

Charles Robinson

Clemens Roothaan

Lyle Settle

John Shepherd

Cheng-kiat Teng

Serena Torres

Nancy Trombetta

Linda Glennie

Michael Williams

Victor Yngve

Table of Contents

Section I

- A. Maniac III Engineering Activity
- B. The Man-Machine Project (C. C. J. Roothaan)
- C. Some New File-Handling, Input-Output, and Monitoring Procedures for Use With the SNOBOL Language (R. B. K. Dewar & G. Manacher)

Section II

- A. On an Approach to Formal Semantics (C. L. Robinson & J. Hansley)
- B. An Experimental Method for Testing Numerical Stability in Initial Value Problems (R. H. Miller)
- C. A Class of Simple Combinatoric Patterns of Automorphisms Equivalent to Veblen-Wedderburn Planes, and a Lower Bound For the Number of Independent Complete Sets of Mutually Orthogonal Latin Squares (G. K. Manacher)

Section III

- A. Force Correlations in Finite Gravitational Systems (R. H. Miller)
- B. Remarks on the Approximation-Solvability of Nonlinear Functional Equations (W. V. Petryshyn)
- C. A New Mode of Computer-Aided Instruction (G. K. Manacher)

MANIAC III ENGINEERING ACTIVITY

1. Interface Computers

a) From discussion of 8-, 12- and 16-bit instruction sets, a 16-bit instruction word has emerged as the most generally acceptable. Some instructions will be followed by additional 16-bit addresses or parameters. Some basic features of an instruction set, notably an addressing scheme, have not yet been decided upon.

b) It is now intended to provide more flexibility in the use of the Interface Computer Internal Registers. Rather than permanently assign most registers for special functions, some of which may be used infrequently, as many of these registers as possible are to be incorporated into a "pool" from which registers may be assigned specific functions under program control.

For example: the current scheme has a "pool" of sixteen 16-bit registers, from which 3 to 12 registers may be assigned as instruction registers, 0 to 6 may be associated with data base addresses, and any not otherwise assigned may be addressed by program.

This permits many program loops to be stored within the processor, reducing memory accesses for instruction fetching and making the "Repeated Instruction" described on page I B-10 of Quarterly Report No. 11 unnecessary.

This change in concept does not entail any modification of the hardware already constructed.

c) Overdue components have now all been received. Testing of the

arithmetic unit showed that the ripple carry was slower than anticipated. Carry speed-up circuits, which increase the total cost of the computer by less than 2%, have been constructed and are now being tested.

d) Work on the simulation program and on design of the control logic is suspended pending decisions on the instruction set and addressing scheme.

e) The 8-word scratchpad memory has been completed and tested, but it has not been operated by the Interface Computer owing to the delay in design of control logic.

2. The PDP-8 Computer

a) The magnetic tape unit for the PDP-8 has been received and tested. It is now available for use.

b) Two adaptors have been built to enable the PDP-8 to be used for:

i) Automatic testing of completed circuit cards such as those used in the Interface Computer.

ii) Monitoring Maniac III activity. This was suggested by G. Estrin of the Department of Engineering, UCLA, Los Angeles in an unpublished memorandum dated 10/31/66 entitled "SNUPER COMPUTER--A Computer Instrumentation Automaton.

MANIAC III USE

October 29 - January 29

(Programmer and production use, debugging and production times are not separated in this table).

Systems Work	20.9
Engineering Computer Time	70.6
Astronomical Calculations	4.4
Language Studies (Robinson)	36.0
Ashenhurst's Class Programs	17.1
Investigations of Projective Planes (Manacher)	1.5
Production Automatic Milling Machine Tape	1.2
G. W. Kreisheimer	16.0
Library Use of 360 such that Maniac III was unavailable	100.0
Miscellaneous	<u>13.3</u>
Total	281.0 hr.

THE MAN-MACHINE PROJECT*

C. C. J. Roothaan

Progress Report No. 8

(for the period 1 October 1966 through 31 December 1966)

Switchable-Tape Interface

Messrs. Steiner and Williams completed programming for the program-switchable-tape interface between the 7094 and 7040. Job input/output transmission is now completely overlapped with 7094 job processing except for the few milliseconds required for synchronizing the two machines at job boundaries. The IBM customer engineers provided valuable assistance in the implementation and debugging of the required hardware modifications.

Remote 1401 Link

Mr. Kruskal completed programming for the link between CHAOS and the 1401 computer at the Biological Sciences Computation Center. The link allows job input/output directly between CHAOS and the 1401 peripheral equipment. Formerly, tape-to-tape transmission was used between the machines. Work is in progress on support for a 1407 typewriter which will give BSCC operators the same communication with CHAOS available to 7040 operators.

*Work on this project, supported by a grant from the International Business Machines Corporation with C. C. J. Roothaan as principal investigator, is carried on under the joint auspices of the Institute and the Computation Center.

Scheduling and Related Changes

Mr. Kubaitis is working on modifications to CHAOS necessary for the "maximum-happiness" scheduling algorithm, integrated setup and non-setup task scheduling, and certain other features such as automated processing of user requests for multiple copy output.

CHAOS Assembly Program

Miss Monheit is currently debugging the CHAOS Assembly Program. The assembler, and an associated loader, will yield the program segmentation necessary for efficient use of the CHAOS memory allocation structure.

POOR (Page Oriented Output Routine)

POOR fetches data from user locations, converts them according to his specifications, and deposits the converted information in whatever buffer(s) the user has previously assigned. A buffer holds a full logical page which is built by a series of calls. Each such call deposits a segment consisting of one or more matrices. Because of the full page buffer, it is very easy for the subroutine to generate elaborate tabular arrangements such as the left-to-right juxtaposition of different sized matrices. Between calls, the user may edit information in the buffer. Dating, page numbering, and titling are provided for. POOR automatically empties buffers (except the final buffer; the emptying of the final buffer requires that the user execute a CLOSE call sequence).

is operating. An improved version is planned which will assist the user in editing and which will facilitate the centering of segments.

360 POOR:

The subroutine will be able to simultaneously service several printers. A version has been coded and is expected to be operating by late January. Improvements are already planned.

SOME NEW FILE-HANDLING, INPUT-OUTPUT, AND MONITORING PROCEDURES FOR USE WITH THE SNOBOL LANGUAGE

R. B. K. Dewar and G. K. Manacher

1. Introduction

The SNOBOL language¹ has extensive facilities for performing operations on strings such as concatenation, insertion, scanning, lookup-by-contents, etc., and as such has become one of the most powerful and easy-to-use string-oriented languages available. It also has the capacity to accept machine-coded functions into its syntax, and this fact has led to the creation of entire satellite systems for extending its power. Routines for handling common character manipulations,² trees,³ lists,⁴ and arrays⁵ of strings have been made available, so that the programmer can incorporate them into the body of his SNOBOL program. Doing this considerably enhances the program's speed and/or power when his data structures resemble the structures these routines are designed to handle.

Among the things SNOBOL cannot do very well is managing files containing very large numbers of strings. Routines for handling these efficiently, added to SNOBOL, might make it a valuable research tool in linguistics, music, BCD data reduction, etc., as well as an occasional production tool in areas for which it is not well suited at present.

For this reason the authors have specified the preliminary design for several routines for use with SNOBOL. The basic data structure is

the array, rather than the string itself. Thus the array-handling package⁵ is assumed to be part of the SNOBOL system. Since SNOBOL machine-coded subroutines are incorporated into SNOBOL simply by loading the routines after SNOBOL, this imposes no difficulty.

The SNOBOL array in general has up to 5 dimensions. In order to be a legitimate argument in a file-handling subroutine, however, it must have only one dimension, else a fatal error will result.

We first discuss array input-output (I/O). The function

(1) AROUT (ARNAM, NEL, UNIT)

will write on the (logical) tape UNIT the first NEL elements (strings) of the array ARNAM* or else all the elements of ARNAM, if the number of elements in ARNAM is smaller than NEL. Each string comprises one logical record and no unit-record breakup occurs, as it does when using SYSPOT or SYSPIT.¹

The function

(2) ARIN(ARNAM, NEL, UNIT)

will have the effect of reading into ARNAM, sequentially from UNIT,

*i.e. the elements \$ARNAM(1), \$ARNAM(2), ..., \$ARNAM(NEL), in that order. See Ref. 5.

either NEL strings stored as logical records, or else filling ARNAM if it has fewer elements than NEL.

Both ARIN and AROUT are based on the INPUT and OUTPUT functions, described in the next section.

Consider the search function

(3) SRCHAR (ARNAM, NEL, ITEM [AR2]) ,

which is a general search routine over the array ARNAM in the range ARNAM (1) . . . ARNAM (NEL) for the item ITEM. If AR2 is missing, the routine assumes that only one occurrence of ITEM is being searched for. In that case it returns a value equal to the first index of ARNAM in which ITEM occurs. If AR2 is nonvoid, the routine assumes that it, too, is a one-dimensional array, else a fatal error occurs. In this case the routine puts into AR2, sequentially, a list of subscripts of the elements of ARNAM containing ITEM and returns as a value the number of items found. The routine stops when either NEL is exceeded or the bound on either ARNAM or AR2 is reached.

The operations on arrays are these: First,

(4) SORTAR (ARNAM, NEL [, ORDER] [, CHARNUM])

uses the previous conventions with respect to ARNAM and NEL.

It sorts the strings in ARNAM up to ARNAM(NEL) assuming the lexicographic order implied by the left-to-right ordering of the characters in the string ORDER. If ORDER is missing, the lexicographic order is assumed to be the standard one (blank, zero, one, ...). This routine uses an existing FORTRAN radix exchange sort capable at sorting 5,000 short words (≤ 6 characters) in about 8 seconds. * CHARNUM, when present and (SNOBOL-sense) numeric, specifies the number of characters in any string over which the sort is effective. Thus, if CHARNUM is 15, any string of length $N > 15$ will have its trailing $N - 15$ characters ignored in the sort.

The function

(5) MERGE(UNITLIST, UNITOUT [, ORDER])

*The connoisseur may want to know how this routine works. It operates in several steps. First, it creates a one-one mapping of BCD characters into numbers sequenced from 0 to 778. It maps the (say, N) characters in ORDER into the ordered sequence 0, 1, ..., N, then it uniquely assigns to the characters not in ORDER additional numbers higher than those in ORDER. It expresses the mapping and its inverse as two 7090 convert tables. All strings in the array are physically converted according to the mapping. Then the pointers in the array are fetched and passed to the FORTRAN routine which specifies the manner in which they may be swapped, together with the bits from the strings needed to specify whether the swap will occur. When execution of this routine has been completed, the pointers in the array have been reordered by a radix exchange sort over the strings, which are never physically moved. The inverse mapping convert table is then applied to the strings.

SNOBOL depends on scatter storage, which imposes a strict relationship between the content of the data and its location. This relationship is clearly violated by the routine, which changes the data temporarily without moving it. However, the fact that the routine does no lookup or string storage allows us to proceed as indicated without violating the scatter storage mechanism.

will merge the units given by UNITLIST, specified e. g. by "A5, B4, A6" onto the unit specified by UNITOUT according to the lexicographic order ORDER, which follows the convention above.

In order to allow the programmer to test a string to see whether it is the name of an array, the function

(6) DIM(ARNAM)

will return as a value the BCD equivalence(s) of the dimensions of ARNAM, concatenated with commas. If ARNAM is not the name of an array, it will fail. Thus, if ARNAM is a 3 x 3 x 5 array, DIM(ARNAM) will return "3, 3, 5".

2. New input-output functions.

Several general purpose routines are being written for use under the University of Chicago monitor, but a version for general distribution under FMS will also be written.

The proposed string I/O allows the input or output of strings of any length. These strings are written so that they appear as individual records to the programmer.

The calls are:

(7) INPUT(UNIT)

which returns as value the next string on UNIT

(7) OUTPUT(UNIT, STR)

which writes the string STR on UNIT.

These I/O functions will use FILE to obtain blocking buffering in the UC implementation. Since string I/O is binary and unit-record I/O in BCD, the attempt to mix these modes results in a parity error, thus providing an automatic check.

Additional functions being provided, with self-evident meanings, are:

ENDFIL (UNIT)

SKPFIL (UNIT)

(8) BAKFIL (UNIT)

UNLOAD (UNIT)

3. A new rule-count monitoring function.

The function

(9) RLIMIT (N, M)

has been programmed and is monitor-independent. In UC SNOBOL it is part of the system. This function initiates a checking routine which allows only N more rules to be executed, followed by a trace (consisting of rule and level number) of M rules. The call RLIMIT(, M) acts purely

as a trace routine.

4. Routines specific to operation under the University of Chicago monitor (CHAOS)

Certain changes and functions apply only to the version of SNOBOL working under the UC monitor. The only function in this class is

(10) CC(ARG)

which has two permissible arguments: "ON" and "OFF". This function affects carriage control and its behavior when the ARG is "OFF" is SNOBOL'S normal mode of operation. When the ARG is "ON", the carriage control is (until the execution of another CC function) under programmer's control; when he uses SYSPOT (or any other string with a print association to the monitor output tape) the first character of the transmitted string determines the carriage control. Thus overprinting, etc. becomes possible. The CC function is part of UC SNOBOL.

Other changes are as follows:

Input-Output

An interface to the SNOBOL I/O routine has been written which simulates all the SNOBOL I/O using TRIGR, FILE and OTAPE. This package allows unit-record I/O on any tape; the I/O on the monitor input and output is blocked and buffered. There is a saving in core of about 2000 cells and a considerable increase in speed compared with the standard BELSIM package.

Multiple Jobs

A driver has been written to allow batched jobs to be run. This cuts overhead on individual jobs to about three seconds.

5. Implementation.

RLIMIT and CC have been written. The remainder of the functions described are in states ranging from completely flow-charted to partially completed. The I/O linkages to CHAOS and multiple job driver are currently in operation.

6. Possible future work.

A function to print the non-void elements of an array is contemplated. This function, if implemented, will identify the index of each array element whose contents is printed. There is a possibility that the system dump will also be altered to include a similar array dump over all arrays active at the time of the dump.

BIBLIOGRAPHY

1. D. J. Farber, R. E. Griswold, and I. P. Polonsky, The SNOBOL Programming Language, BSTJ. XLV, 6, July-August 1966, 895-944.
2. G. K. Manacher, A package of subroutines for the SNOBOL language. Unpublished.⁺
3. R. E. Griswold and I. P. Polonsky, Tree functions of SNOBOL. Unpublished.⁺
4. R. E. Griswold, Linked-list functions for SNOBOL. Unpublished.⁺
5. G. K. Manacher and L. C. Varian, A dimension statement and a random number facility for the SNOBOL language. Unpublished.⁺

⁺These memos may be obtained from Mr. R. E. Griswold, Room 2B-609, Bell Telephone Laboratories, Holmdel, New Jersey.

ON AN APPROACH TO FORMAL SEMANTICS*

C.L. Robinson and J. Hansley

ABSTRACT

A method of defining formal semantics, similar to Chomsky's formal syntactic definitions, is presented. The technique is as powerful as needed, it being trivial to prove that any result obtainable by a simple Turing machine⁴ can be produced using the semantic definitions given. Further the combination of the standard syntactic definitions with the semantic rules leads to practical computational results. Examples of this practical power are taken from both the fields of theorem proving and artificial languages.

In contrast to the use of the mathematically formal definitions of syntax as given by Chomsky and others there has been a notable lack of rigor when our profession has used the word semantics. One principle, however, seems to be generally agreed upon; namely, that syntax and semantics are closely related or more precisely: syntax supplies certain semantic limits. For example, given the normal syntax of arithmetic expressions one is somehow constrained to adopt a semantic hierarchy for the operators even in the absence of the precise definitions of the operations plus, minus, times, etc. This is reflected in the ease with which infix arithmetic or boolean expressions can be transformed into their polish suffix form. It is for this reason that semantics is often

*This work was supported by the U.S. Atomic Energy Commission under contract No. AT(11-1)614.

viewed as meaning manipulation of the implied parsing tree generated by syntactic analysis.

It is our feeling that there exists at least one basic test of a definition of formal semantics. Namely such a syntactic/semantic system should allow theorems to be proved, and in particular theorems in the propositional calculus, since a decision procedure is known for this. That is, an input string (conjecture) is a sentence (theorem) in the language (propositional calculus), if and only if it is both syntactically well-formed and semantically true. This practical test avoids the pitfalls of an overly specialized definition which aids, for example, a particular compiling procedure for handling artificial languages but is impractical for dealing with other algorithms. Unfortunately many of the tree manipulative definitions of semantics fall into this latter class.

Let S_0 and S_1 be strings; with S_0 containing terminal characters only, being originally the input string, and S_1 containing terminal and nonterminal characters. In addition let each character in S_1 be associated with pointers to the (contiguous) string of characters in S_0 which were used in the syntactic reduction process, described below, to make it up. Further, let there be a copy operation from S_0 onto S_1 which is initially applied when S_0 is assigned the value of the input string. A syntactic reduction is of the form $\alpha \Sigma \beta \longrightarrow \alpha \gamma \beta$ where α ,

Σ and β are strings of terminals and nonterminals, Σ not null, and γ is nonterminal.* The application of a syntactic

* A Chomsky type 1 grammar.

reduction takes a substring of S_1 of the form $\alpha \Sigma \beta$ and rewrites it as $\alpha \gamma \beta$ in S_1 , the pointers of γ being set to the beginning and end of the leftmost and rightmost characters respectively in the string

Σ . A semantic reduction has the form $X \rightarrow \phi$ where X and ϕ are strings of terminals and nonterminals such that if N is nonterminal in ϕ then N appears once and only once in X . An application of a semantic reduction rewrites a substring of S_0 in S_0 in accordance with the pointers in S_1 associated with the string X and the ordering given by the string ϕ . S_0 is then copied into S_1 and all the pointers of S_1 are re-established. An input string is a sentence in the language specified by the syntactic and semantic reductions if and only if there is some sequence of such reductions such that S_1 is reduced to only one character, the goal character δ . This distinguished or goal character δ has the additional attribute that it never appears on the left hand side of a reduction.

To test the applicability of these ideas the second author has revised the Gilbert's Analytic Grammar program reported upon.¹ With this routine we have indeed succeeded in proving theorems of the propositional calculus using the schemata or dialog method of Gentzen as a decision procedure. Appendix 1 shows the semantic and syntactic reductions used and gives an example of a (trivial) conjecture being shown to be a true proposition.

The most immediate application of rules of formal semantics is in compiler design. For example, the official report on Algol 60² relied

heavily on informal text to describe the effects produced by various types of FOR statements. Had such a formal rigor been available when the report had been published much of the trivial debate over the FOR statement could have been eliminated and the rule concerning the value of the control variable at the exit of a STEP UNTIL element could have been defined.

The following set of syntactic and semantic rules states precisely how FOR statements could be reduced to the more basic components or kernel of the Algol language. In doing this the notation and feeling of the Report is maintained as much as possible and the section could replace 4.6 in the Report.

4.6 FOR STATEMENTS

4.61 SYNTAX

- (a) <arithmetic expression> $\rightarrow$ <for list element>
- (b) <arithmetic expression> STEP <arithmetic expression>
UNTIL <arithmetic expression> $\rightarrow$ <for list element>
- (c) <arithmetic expression> WHILE <boolean expression>
 $\rightarrow$ <for list element>
- (d) <for list element> $\rightarrow$ <for list>
- (e) <for list>, <for list element> $\rightarrow$ <for list>
- (f)* <arithmetic expression> STEP <arithmetic expression>
UNTIL <arithmetic expression> $\rightarrow$
<arithmetic expression> STEP <arithmetic expression 2>
UNTIL <arithmetic expression>

*These syntactic reductions are required to ensure that the corresponding semantic reduction (c) is well-defined.

(g)* <arithmetic expression> STEP <arithmetic expression 2>

UNTIL <arithmetic expression> →

<arithmetic expression> STEP <arithmetic expression 2>

UNTIL <arithmetic expression 3>

4.63 SEMANTICS

(a) FOR <variable> := <for list>, <for list element> DO <statement> →

BEGIN

FOR <variable> := <for list> DO <statement>;

FOR <variable> := <for list element> DO <statement>

END

(b) FOR <variable> := <arithmetic expression> DO <statement> →

BLOCK

<variable> := <arithmetic expression>; <statement>

ENDBLOCK

(c) FOR <variable> := <arithmetic expression> STEP

<arithmetic expression 2> UNTIL <arithmetic expression 3>

DO <statement> →

BLOCK

<variable> := <arithmetic expression>;

L1: IF (<variable> - <arithmetic expression 3>)*

SIGN (<arithmetic expression 2>) > 0 THEN

GO TO L2;

<statement>;

<variable> := <variable> + <arithmetic expression 2>;


```

      GO TO L1;

L2: ENDBLOCK

(d)  FOR <variable> := <arithmetic expression> WHILE
      <boolean expression> DO <statement> →

BLOCK

L3: <variable> := <arithmetic expression>;

IF <boolean expression> THEN GO TO L4;

<statement>;

GO TO L3;

L4: ENDBLOCK

```

4.64 Notes

The declarative delimiters BLOCK and ENDBLOCK are assumed to have a similar syntax, except for commentary, to Algol's BEGIN and END and indicate the opening and closing of a block in the Algol sense. L1, L2, L3 and L4 are assumed to be system delimiters which the programmer cannot use. In all other ways they behave as normal Algol labels.

In summary let it be noted that no claim is made that this is an efficient way to build a translator for a language. Rather this specification method, as with formal syntactic specification, allows the language designer to delineate exactly what he means, in this case how a complicated structure should be reduced into its constituent parts. In particular if the semantic

meaning or "output" of the algorithm related to this specification method is considered to be the string S_0 after the last semantic reduction (or equivalently S_0 when S_1 has been reduced to the goal character, δ). then we have a technique for completely specifying how an artificial language is to be reduced to machine code.

References

1. C. Robinson, "Some Comments on Gilbert's Analytic Grammar," ICR Quarterly Report No. 9, Section III E (Institute for Computer Research, University of Chicago).
2. P. Naur (ed.), "Revised Report on the Algorithmic Language ALGOL 60," Communications of the ACM, vol. 6, (January 1963) pp. 1-17
3. H. Wang, "Toward Mechanical Mathematics," IBM Journal of Research and Development, vol. 4 (January 1960) pp. 2-22
4. M. Davis, Computability and Unsolvability, McGraw-Hill, 1958

Appendix -- An Example from the Propositional Calculus

Below are printed the syntactic and semantic reductions that define a variant of the cut-free schemata method of Gentzen, another variant of which was used by Wang³ in his classical paper on mechanical theorem proving. In this notation P, Q and R are the atomic elements with J, K, L, M, N, and O being various test characters introduced by the scan. The / separates the hypothesis from the conclusion of a conjecture while # separates sub-tableaus or sub-conjectures, all of which must be the theorems if the whole string is a theorem. C represents a correct conjecture (theorem), F is the goal non-terminal while ϕ and ? are string left and right terminators. Further < > denotes that the enclosed character is nonterminal. The scan is defined for the operations and ($\&$), or ($|$), implies ($>$), and not ($\neg$). The examples which follow are photographs of the Maniac III output where the string S_1 is printed at every stage of alteration. The starred lines represent the output of syntactic reductions while the unstarred lines give the state of S_1 immediately after a copy operation (semantic reduction).

SYNTACTIC REDUCTIONS

LEFT HAND SIDE

RIGHT HAND SIDE

1	$\emptyset$ C ?	<F>
2	P	<S>
3	Q	<S>
4	R	<S>
5	(<S> > <S>)	<S>
6	(<S> & <S>)	<S>
7	(<S> <S>)	<S>
8	($\neg$ <S>)	<S>
9	<S><S>	<S>
10	<S> (<S> <S>) / <X>	<1> (<S> <S>) / <X>
11	<1> (<S> <S>) / <X>	<1> (<2> <S>) / <X>
12	<S> (<S> <S>) / #	<1> (<S> <S>) / #
13	<1> (<S> <S>) / #	<1> (<2> <S>) / #
14	<S> (<S> <S>) / ?	<1> (<S> <S>) / ?
15	<1> (<S> <S>) / ?	<1> (<2> <S>) / ?
16	(<S> <S>) / <X>	(<1> <S>) / <X>
17	(<S> <S>) / #	(<1> <S>) / #
18	(<S> <S>) / ?	(<1> <S>) / ?
19	<S> (<S> > <S>) / <X>	<1> (<S> > <S>) / <X>
20	<1> (<S> > <S>) / <X>	<1> (<2> > <S>) / <X>
21	<S> (<S> > <S>) / #	<1> (<S> > <S>) / #
22	<1> (<S> > <S>) / #	<1> (<2> > <S>) / #
23	<S> (<S> > <S>) / ?	<1> (<S> > <S>) / ?
24	<1> (<S> > <S>) / ?	<1> (<2> > <S>) / ?
25	(<S> > <S>) / <X>	(<1> > <S>) / <X>
26	(<S> > <S>) / #	(<1> > <S>) / #
27	(<S> > <S>) / ?	(<1> > <S>) / ?
28	(<S> & <S>) /	(<1> & <S>) /
29	/ (<S> <S>)	/ (<1> <S>)
30	/ (<S> > <S>)	/ (<1> > <S>)
31	$\emptyset$ / (<S> & <S>) <X>	$\emptyset$ / (<1> & <S>) <X>
32	$\emptyset$ / (<S> & <S>) #	$\emptyset$ / (<1> & <S>) #
33	$\emptyset$ / (<S> & <S>) ?	$\emptyset$ / (<1> & <S>) ?
34	# / (<S> & <S>) <X>	# / (<1> & <S>) <X>
35	# / (<S> & <S>) #	# / (<1> & <S>) #
36	# / (<S> & <S>) ?	# / (<1> & <S>) ?
37	J #	<X> #
38	K #	<X> #
39	L #	<X> #
40	M #	<X> #
41	N #	<X> #
42	O #	<X> #
43	J ?	<X> ?
44	K ?	<X> ?
45	L ?	<X> ?
46	M ?	<X> ?
47	N ?	<X> ?
48	O ?	<X> ?
49	<S> #	<X> #
50	<S> ?	<X> ?
51	J <X>	<X>
52	K <X>	<X>
53	L <X>	<X>
54	M <X>	<X>
55	N <X>	<X>
56	O <X>	<X>
57	<S><X>	<X>

SEMANTIC REDUCTIONS

LEFT HAND SIDE

RIGHT HAND SIDE

1	K J	J K
2	L J	J L
3	L K	K L
4	K M	M K
5	L M	M L
6	L N	N L
7	N M	M N
8	O M	M O
9	O N	N O
10	N J	J N
11	O J	J O
12	O K	K O
13	P /	/ M
14	Q /	/ N
15	R /	/ O
16	/ P	/ J
17	/ Q	/ K
18	/ R	/ L
19	J <S>	<S> J
20	K <S>	<S> K
21	L <S>	<S> L
22	M <S>	<S> M
23	N <S>	<S> N
24	O <S>	<S> O
25	M J	C
26	N K	C
27	O L	C
28	J M	C
29	K N	C
30	L O	C
31	J C	C
32	K C	C
33	L C	C
34	M C	C
35	N C	C
36	O C	C
37	/ C	C
38	<S> C	C
39	C <X>	C
40	C # C	C
41	/ (<1> & <S>) <X>	/ <1><X> # / <S><X>
42	/ (<1> & <S>)	/ <1> # / <S>
43	(<1> & <S>) /	<1><S> /
44	/ (<1> <S>)	/ <1><S>
45	<1> (<2> <S>) / <X>	<1><2> / <X> # <1><S> / <X>
46	<1> (<2> <S>) /	<1><2> / # <1><S> /
47	(<1> <S>) / <X>	<1> / <X> # <S> / <X>
48	(<1> <S>) /	<1> / # <S> /
49	/ (~ <S>)	<S> /
50	(~ <S>) /	/ <S>
51	/ (<1> > <S>)	<1> / <S>
52	<1> (<2> > <S>) / <X>	<1> / <2><X> # <1><S> / <X>
53	<1> (<2> > <S>) /	<1> / <2> # <1><S> /
54	(<1> > <S>) / <X>	/ <1><X> # <S> / <X>
55	(<1> > <S>) /	/ <1> # <S> /

Ø / ((¬ P) > (P > Q)) ?

* Ø / ((¬ <S>) > (/ P > Q)) ?

* Ø / (<S> > (P > Q)) ?

U * Ø / (<S> > (<S> > Q)) ?

* Ø / (<S> > (<S> > <S>)) ?

* Ø / (<S> > <S>) ?

* Ø / (<1> > <S>) ?

Ø (¬ P) / (P > Q) ?

* Ø (¬ <S>) / (P > Q) ?

Ø / P (P > Q) ?

Ø / J (P > Q) ?

* Ø / J (<S> > Q) ?

* Ø / J (<S> > <S>) ?

* Ø / J <S> ?

Ø / (P > Q) J ?

* Ø / (<S> > Q) J ?

* Ø / (<S> > <S>) J ?

* Ø / (<1> > <S>) J ?

Ø P / Q J ?

Ø / M Q J ?

* Ø / M <S> J ?

Ø / Q M J ?

Ø / K M J ?

Ø / M K J ?

Ø / M J K ?

Ø / C K ?

Ø C K ?

* Ø C <X> ?

Ø C ?

* <F>

STRING SUCCESSFULLY REDUCED

EXAMPLE 1

$\emptyset / (((\neg P) \rightarrow P) \rightarrow P) ?$
 * $\emptyset / (((\neg \langle S \rangle) \rightarrow P) \rightarrow P) ?$
 * $\emptyset / ((\langle S \rangle \rightarrow P) \rightarrow P) ?$
 * $\emptyset / ((\langle S \rangle \rightarrow \langle S \rangle) \rightarrow P) ?$
 * $\emptyset / (\langle S \rangle \rightarrow P) ?$
 * $\emptyset / (\langle S \rangle \rightarrow \langle S \rangle) ?$
 * $\emptyset / (\langle 1 \rangle \rightarrow \langle S \rangle) ?$
 $\emptyset ((\neg P) \rightarrow P) / P ?$
 $\emptyset ((\neg P) \rightarrow P) / J ?$
 * $\emptyset ((\neg \langle S \rangle) \rightarrow P) / J ?$
 * $\emptyset (\langle S \rangle \rightarrow P) / J ?$
 * $\emptyset (\langle S \rangle \rightarrow \langle S \rangle) / J ?$
 * $\emptyset (\langle S \rangle \rightarrow \langle S \rangle) / \langle X \rangle ?$
 * $\emptyset (\langle 1 \rangle \rightarrow \langle S \rangle) / \langle X \rangle ?$
 $\emptyset / (\neg P) J \# P / J ?$
 $\emptyset / (\neg P) J \# / M J ?$
 $\emptyset / (\neg P) J \# / C ?$
 $\emptyset / (\neg P) J \# C ?$
 * $\emptyset / (\neg \langle S \rangle) J \# C ?$
 $\emptyset P / J \# C ?$
 $\emptyset / M J \# C ?$
 $\emptyset / C \# C ?$
 $\emptyset C \# C ?$
 $\emptyset C ?$
 * $\langle F \rangle$

STRING SUCCESSFULLY REDUCED

AN EXPERIMENTAL METHOD FOR TESTING NUMERICAL STABILITY IN INITIAL VALUE PROBLEMS

R. H. Miller

ABSTRACT

The numerical stability of evolutionary calculations can be tested experimentally by running two calculations in parallel, starting from initial conditions that are very similar, and monitoring the differences between the systems. An integration of an n-body system with gravitational interactions provides an example that is useful for illustrative purposes. Some features of the method are described, followed by a discussion of some considerations in its application.

1. Introduction

An initial-value problem can describe the evolution of a physical system from some specified initial state according to a set of governing equations. The equations may or may not include boundary conditions, but there is no eigenvalue character resulting from forcing some specified terminal state. When an initial-value problem is run on a computer, the results may appear plausible even if they are unreliable because of some unrecognized numerical instability. Analytic methods may not be available for studying numerical stability; the techniques of numerical analysis are usually applied to the much simpler processes out of which larger calculations are built. Some aspects of a large problem are not tractable in the usual numerical analysis study; the "errors" in inputs to a certain

numerical process inside a large calculation are not independent, for example. Initial-value problems do not share the inherent stability of some kinds of calculations: self-consistent methods, for example, are intrinsically stable if they possess unique solutions. Theoretical studies of the stability of initial value problems have been made, for example, by Lax and Richtmyer [1].

A suitably designed experimental approach can provide a method for studying the numerical stability of large calculations even when a theoretical approach is not available. An experimental method that should be applicable to a wide range of initial value problems is described in this paper (Section 2). It provided a valuable test in a gravitational n-body integration (Section 3), where it disclosed a surprisingly unstable situation. This example illustrates many of the features of the experimental method. Generalizations to other problems are discussed, along with suggestions concerning the kinds of problems for which the method might be useful.

II. Parallel Calculations

In the approach under consideration, similar systems are constructed in the computer memory and permitted to evolve together, while the differences among them are monitored. To the extent that the systems are quite similar, subtle differences can be noted. The calculation may be run as long as the comparisons are meaningful. The same copy of the program can be designed to operate on all sets of data, one after another. If variable time-steps are used, some care is required to assure that all systems use the same values of the time-variable;

this is most easily done by always using the shortest time-step.

"Similar" systems may be obtained by starting from one system, which is copied to form the other systems. When the copies are made, small perturbations can be introduced. Ideally, these perturbations should be of known character and larger than roundoff or integration errors in one time-step, but small enough that the systems are recognizable as having had a common origin. The stage of the calculation at which the copies are made can be chosen for convenience; for example, if there is some special starting process, to build up a "history" for an integration method that makes use of previous states, copies might be made after switching over to the usual (running) integration routine. The extent to which the present state is not properly attainable from the "history" is then part of the perturbation.

This method is conceptually similar to "interval arithmetic" [2]. It differs by extending over the entire calculation and by not requiring well-behaved or monotonic input/output relationships.

Other means for checking initial-value calculations can easily be devised, but the parallel calculation approach seems better. Time-reversible systems might be run forward for a while, then reversed and run back to the starting point where the final state can be compared with the initial state. Time-reversal is not as effective as parallel calculation because of errors incurred in reversing the system.

The parallel calculation usually requires less modification to a running

program for test purposes, and may be used with systems that are not time-reversible.

III. Example

This method was devised to check an n-body integration in stellar dynamics that failed to time reverse [3]. In this case the state of a system is given by the location of its representative point in the $6n$ -dimensional phase space. Two systems are described by two points in the same space. If the systems are similar, the points are very near each other; a convenient measure of the departure of one system from the other is the separation of the phase points:

$$\Delta^2 = \sum_{i=1}^n \left\{ (x_i^{(2)} - x_i^{(1)})^2 + T^2 (u_i^{(2)} - u_i^{(1)})^2 \right\}, \quad (1)$$

where T is a dimensional factor introduced to make all coordinates and velocities equivalent, the x 's are configuration space coordinates, the u 's are velocities, the superscripts refer to corresponding particles in the two systems, and the sum extends over all particles. (In gravitational problems with nonvarying masses, it is not necessary to distinguish between velocities and momenta.)

The problem, apart from the parallel calculation feature, was formulated according to the recipes of von Hoerner [4]. In this formulation, the integration is carried out for n particles of equal mass in Cartesian coordinates. Variable time steps based on the closest pair of particles are used to retain the accuracy of integration for close

encounters while permitting coarser steps to be used when all the particles are well separated. The same time step is used for all particles. A second-order predictor-corrector integration method is used in which an attempt is made to retain reasonable accuracy without requiring re-evaluation of the forces more than once per integration step. The first ten integrals of the motion were used as controls, and were constant to within the same limits as von Hoerner reported [4].

A better integration procedure has since been devised [5]; the usefulness of the experimental method reported here is independent of the methods of handling the main problem.

The system was started from some set of initial conditions that was externally supplied. After enough first-order integration steps to build up the history required for the predictor-corrector method, the second copy of the system was made as part of the changeover to the normal second-order integration procedure. In copying, a perturbation was introduced as a small change in one velocity component of one particle.

Plots of $\ln \Delta$ against time are shown in Figure 1 for two different initial conditions. The track shows a steady increase of $\ln \Delta$ with time, but has a series of spikes superposed. The linear trend of the minima between the spikes illustrates an exponential increase of the separation of phase points with time. The general trend may be followed from $\ln \Delta \approx -20$ to $\ln \Delta \approx 0$; the latter value corresponds roughly to the mean particle separation or to the r. m. s. particle velocity. Beyond

this the system will blow up. This long track illustrates the sensitivity of parallel calculations. The spikes are of impressive amplitude--about a factor of 1000 or more over the prevailing level. The spikes are associated with close collisions; the spike results because the two systems enter the close collision at slightly different phases. The recovery afterward is especially dramatic.

Several experiments were tried with this system. The computation was stopped during a close encounter (on top of a spike) to determine which particles were participating. The calculation could then be repeatedly restarted from the same initial conditions, but with the perturbation placed on one of the participants in the close encounter or on a particle that did not participate. In this way it was found that information propagates through the system very rapidly.

Another experiment was to modify the step size in the integration procedure. The nature of the tracks was unchanged.

Parallel calculations with the perturbation set to zero keep $\Delta = 0$ for all time. This is characteristic of computer calculations. The size of the perturbation was varied; in general it is desirable to keep the perturbation as small as possible. If it is too small, however, Δ fluctuates about some small value for a substantial time before the reported trends set in.

This calculation raised many questions concerning the nature of the physical system that is supposedly represented. The principal point at issue is the sense in which the calculation represents a physical system. The calculation takes on a Monte Carlo character with the random element

generated by roundoff and integration errors. With an exponential growth, the error terms will ultimately dominate, and the current state of the system cannot be causally related to the initial state in the manner that the physical system would be related to its initial state. Subsidiary questions include the rate of information transfer through the system, the factors that govern the growth of Δ , and so on. A result like this stimulates further work. In the gravitational case, it was possible to reproduce some of these effects analytically [6], although it is unlikely that the effect would have been discovered without the computer experiments. Fuller discussions of the physical implications of these results appear elsewhere [3, 6].

IV. Generalizations

Some guides to the application of the parallel calculation method to other problems emerged in the treatment of this example.

a) Measure of departure

The quantity Δ that was used in the example is very sensitive to departures of one system from another. It may be argued that it is too sensitive and that the quantities sought are functionals of the motion, which may be much less sensitive to calculational errors than Δ . In the example of Section 3, the conventional "first integrals of the motion" are well conserved; it is a consequence of the equations of motion that errors in these integrals are of higher order than the errors in coordinates. Other functionals may have intermediate behavior.

A safe and convenient procedure appears to be to subject the quantities of interest--the "results" of the calculation--to the test afforded by parallel calculations. However, the most sensitive indicator is useful to warn of a possibly dangerous situation. Other kinds of problems may not admit of as obvious a sensitive indicator as Δ of the example.

The inherent stability of the usual first integrals means that a calculation may contain unstable features even if it appears safe or plausible when they are used as indicators.

b) Magnitude and character of initial perturbation

Ideally the initial perturbation should be large enough that a response will develop immediately; it should also be substantially larger than the errors of integration in one step or the errors of roundoff, but otherwise it should be as small as possible. Several trials may be required to find a good perturbation. Various perturbations should be tried stepping in different directions in the function space of the problem. Complicated perturbations that might be difficult to interpret should be avoided.

c) Interpretation

As in the example, it can be difficult to determine whether an unstable result indicates a purely numerical effect or a property of the physical system that is being represented. Usually more experimentation is required to distinguish; parallel calculations provide additional measures

of the behavior of the system under these experiments. A very sensitive indicator of departure is helpful. The checks with different integration step sizes, mentioned in Section 3, are an example of this. Different kinds of experiments, which should alter the interplay between numerical and physical effects, can be tried.

Perhaps the most difficult point to answer, in the case of as strong an instability as that of Section 3, is whether any result can have meaning. The best answer to this is to display some quantities, such as the first integrals of the motion, that behave as expected. Then it might be reasonable to infer that some things can be reliably calculated, even if some others cannot.

d) Application to other calculations

The method described here should be of some use with almost any calculation of evolutionary character. Stellar evolution calculations are an obvious example [7]; in effect, parallel calculations have almost been done in the past, because the initial conditions are quite similar when exploring the evolutionary history of stars of nearly the same mass. Comparisons using a sensitive measure have not been explicitly carried out, however.

This work was supported by the Atomic Energy Commission under Contract AT(11-1)-614.

REFERENCES

1. P. D. Lax and R. D. Richtmyer, Comm. Pure Appl. Math. 9, 267 (1956).
2. R. E. Moore, Interval Analysis, Prentice-Hall (1966).
3. R. H. Miller, Astrophys. Journ. 140, 250 (1964).
4. S. von Hoerner, Zeits.f. Astrophys. 50, 184 (1960).
5. S. J. Aarseth, Monthly Notices of the Roy. Astron. Soc., 126, 133 (1963).
6. R. H. Miller, Astrophys. Journ. 146, 831 (1966).
7. E. Hofmeister, R. Kippenhahn, and A. Weigert, Meth. Comp. Phys. 7 (to be published).

FIGURE CAPTION

Figure 1. Plot of $\ln \Delta$ vs. time for an 8-body system. Two different starting conditions are shown. The repeated equidistant spikes on the solid curve are caused by a long-lived binary pair that is destroyed by an encounter of one of its members with a third star at the last spike of that curve.

Fig.. 1.

A CLASS OF SIMPLE COMBINATORIC PATTERNS OF AUTOMORPHISMS EQUIVALENT TO VEBLEN-WEDDERBURN PLANES, AND A LOWER BOUND FOR THE NUMBER OF INDEPENDENT COMPLETE SETS OF MUTUALLY ORTHOGONAL LATIN SQUARES

G. K. Manacher

Abstract

A class of simple "combinatoric algebras" is devised by piecing together fragments of information resulting from computer experiments on Latin squares. These algebras have the same addition group as a field with p^n elements, but a multiplication rule of the form

$$(A) \quad x^i \circ x^t = x^i * x^t \cdot p^{m \cdot k(i)}$$

where x is a generator of the field, "*" represents multiplication in the field, m is a proper divisor of n and $k(i)$ may take on any integral values subject to the constraint $k(i) = k(i')$ when $i \equiv i' \pmod{p^m - 1}$. Of particular interest in (A) is the fact that the new multiplication rule does not depend on addition, as it does for most other known systems.

A method is developed for enumerating the classes of algebras having sets of Latin squares not transformed into one another under common row and column permutations and/or individual symbol permutations. When "multiplications" analogous to (A) are applied to suborthomorphic squares¹⁹ rather than Cayley tables for addition groups, a triple of mutually orthogonal Latin squares of order 22 is discovered, the lowest order known for such a triple $\equiv 2 \pmod{4}$.

1. Introduction

We first recall some definitions.

Definition 1. A set A of Latin squares is said to be independent of a set B if and only if it is impossible to find any iteration of the following transformations that will map A into B (and therefore B into A):

- (1) Row permutations applied identically to all the squares in A .
- (2) Column permutations applied identically to all the squares in A .
- (3) Symbol permutations applied to any square in A .

Definition 2. An incidence matrix for a projective plane is a binary matrix whose rows (columns) represent the lines of the plane and whose columns (rows) represent the points. The size of the matrix must be $(s^2 + s + 1)^2$, for squares of size $s \times s$. A "1" in the i^{th} row (column) and the j^{th} column (row) indicates the presence of point j on line i . A "0" indicates the absence of point j on line i .

Definition 3. Two planes are isomorphic if it is possible to transform the incidence matrix of one into the other by means of a permutation of its columns followed by a permutation of its rows. A plane whose incidence matrix is isomorphic to its transpose is called self-dual.

It is known that the lowest order planes not isomorphic to the Desarguesian plane are planes of order 9. Three such planes are known: A self-dual plane discovered by Veblen and Wedderburn in 1907³ and two non-self-dual planes not characterizable by a Veblen-Wedderburn system, known as Hughes planes.¹⁴

Since our entire discussion is based on a combinatoric generalization of the Veblen-Wedderburn plane of order 9, it seems appropriate to review the properties of this plane. Our object will be to relate the multiplication law for this plane to a complete set of automorphisms, and to isolate certain simple regularities in these automorphisms that suggest combinatoric generalization to higher orders.

The algebra used by Veblen and Wedderburn³ in their original article is Dickson's.¹³ This is a system of order p^{2m} , with p an odd prime and m an integer, with elements of the form (a, b) where a and b are elements of the field of order p^m . The rules of combination are:

$$(1-1) \quad (a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2),$$

$$(1-2) \quad (a_1, b_1) \circ (a_2, b_2) = (a_1 * a_2 + \epsilon * \mathcal{D} * b_1 * b_2, b_1 * a_2 + \epsilon * a_1 * b_2)$$

where $\mathcal{D}$ is a fixed non-square in the field of order p^m , and $\epsilon = \pm 1$ according as $a_1^2 - \mathcal{D} * b_1^2$ is a square or not a square. In (1-2, "*" represents the product of two elements defined by the multiplication law of the field, and " $\circ$ " is the multiplication law of the VW (Veblen-Wedderburn) system.

If the algebra is thought of as consisting of elements with two bases 1 and i , so that its elements are of the form $a + bi$ with a and b residues (mod 3), we obtain a system consisting of 9 complex numbers

0, 1, 2, $1+i$, $2+i$, $1+2i$, $2+2i$, i , $2i$, whose multiplication law is essentially given by (1-3); for any element e it is the case that $1 \circ e = e \circ 1 = e$ and $0 \circ e = e \circ 0 = 0$.

As it stands, Table 1 is not particularly revealing, since there is no particular sense to the ordering of the elements. It is an attractive idea, therefore, to employ the field of order 9 as a device serving to order the elements, even though the multiplication expressed by Table 1 is not multiplication in the field. The device used is as follows. The addition group for (a, b) is obviously $\{3\} \times \{3\}$, where $\{c\}$ represents the cyclic group of order c and $\times$ represents a direct product of groups. The field of order 9 has elements $a+bx$, where a and b are residues (mod 3) and its addition group is also $\{3\} \times \{3\}$. In the Bose representation [Ref. 19, formula (1-5a)] the elements are ordered: $\alpha_0 = 0$, $\alpha_j = x^j \pmod{f(x)}$ ($j \neq 0$) where $f(x)$ is a minimal polynomial and the α_j are all distinct. In the field, $x^j = x \cdot x \cdot \dots \cdot x$ (j terms); in the VW system no such generator exists and the elements cannot all be represented as some power of a single element. Nevertheless, the "i" of the VW system and the "x" of the field can be formally identified together for purposes of ordering the elements of the VW system. When this is done, one obtains a multiplication table for the VW system (Table 2) in which only the subscripts j are shown. The subscripts correspond to: $\alpha_0 = 0$; $\alpha_1 = i$; $\alpha_2 = 1 + 2i$; $\alpha_3 = 2 + 2i$; $\alpha_4 = 2$; $\alpha_5 = 2i$; $\alpha_6 = 2 + i$; $\alpha_7 = 1 + i$; $\alpha_8 = 1$.

	2	i	2i	1+i	1+2i	2+i	2+2i
2	1	2i	i	2+2i	2+i	1+2i	1+i
i	2i	2	1	2+i	1+i	2+2i	1+2i
2i	i	1	2	1+2i	2+2i	1+i	2+i
1+i	2+2i	1+2i	2+i	2	2i	i	1
1+2i	2+i	2+2i	1+i	i	2	1	2i
2+i	1+2i	1+i	2+2i	2i	1	2	i
2+2i	1+i	2+i	1+2i	1	i	2i	2

Table 1. VW Multiplication Table. From Ref. 3

The above table shows that in general $a \circ (b+c) = a \circ b + a \circ c$, but $(b+c) \circ a \neq b \circ a + c \circ a$.

	1	2	3	4	5	6	7	8
1	4	7	2	5	8	3	6	1
2	3	4	5	6	7	8	1	2
3	6	1	4	7	2	5	8	3
4	5	6	7	8	1	2	3	4
5	8	3	6	1	4	7	2	5
6	7	8	1	2	3	4	5	6
7	2	5	8	3	6	1	4	7
8	1	2	3	4	5	6	7	8

Table 2. Multiplication Table for VW System with "Field Ordering".

The elements shown in Table 2 combine in such a striking way that it is immediately possible to give a simpler construction than Dickson's to coordinatize the VW plane of order 9. We define

$$(1-3) \quad \alpha_i \circ \alpha_t = \alpha_i * \alpha_t \text{ [for } i \equiv 0 \pmod{2}]$$

$$= \alpha_i * \alpha_t^3 \text{ [for } i \equiv 1 \pmod{2}]$$

Relation 1-3 provides the first glimpse of the kind of construction we shall deal with in the remainder of this section.

Table 2 allows us to specify the Latin squares making up the VW plane with considered ease. Let $L_{1,1}$ represent the basic addition table in Bose-normal form, i. e. with the elements ordered according to $\alpha_0 = 0, \alpha_i = x^i$. Let the column positions of $L_{1,1}$ be identified by the element topmost in the column, i. e. column 0, the leftmost column, column 1, the next-leftmost, etc. Then the Latin square $L_{x,r}$ is defined as a square having the same column 0 as $L_{1,1}$ and having as its column y , column $x + (y-1) \cdot r \pmod{s-1}$ of $L_{1,1}$.^{*} Thus, $L_{2,3}$ has columns 0, 2, 5, 8, 3, 6, 1, 4, 7, respectively, of $L_{1,1}$ as its columns 0 through 8. This definition clearly restricts r to being relatively prime to $s-1$, else $L_{x,r}$ will not be Latin. We will call x the start column and r the submodulus. The complete set of mutually orthogonal Latin squares characterizing the VW of order 9 plane is then

^{*}The symbol $\cdot$ represents ordinary integer multiplication. From here on $a \cdot b$ will be assumed to mean $a \cdot b \pmod{s-1}$, where the reduction mod $s-1$ is similar to the reduction mod s except that $x \pmod{s-1}$ is written as y rather than 0 when $x \pmod{s-1} \equiv 0$.

$$(1-4) \quad L_{1,3}; L_{2,1}; L_{3,3}; L_{4,1}; L_{5,3}; L_{6,1}; L_{7,3}; L_{8,1}$$

Since only column interchange is permitted, it is clear that in any complete set, every value of x must be represented exactly once. Letting the values of x ascend monotonically from 1 to $s-1$, and omitting the symbol L , we can rewrite (1-4) as

$$(1-5) \quad 3, 1, 3, 1, 3, 1, 3, 1.$$

We shall call (1-5) the pattern for the plane.

Finally, we observe that the pattern (1-5) contains a smallest repeating element, namely, 3, 1. We call this the fundamental pattern Ψ , whose length is called the period of the pattern.

Given a fundamental pattern of a plane of order s , it becomes possible to represent the plane very compactly. For instance, for the VW plane of order 9, we can describe the whole plane by writing

$$(1-6) \quad \Psi(9) = 3, 1$$

It turns out, as we shall prove presently, that the enumeration of the pairs (x, r) and (x', r') for which $L_{x, r} \not\perp L_{x', r'}^*$ is effectively accomplished by listing the values of (x, r) for which $L_{1,1} \not\perp L_{x, r}$.

* $\not\perp$ means "is orthogonal to".

We shall find that such an enumeration yields complete sets of mutually orthogonal Latin squares for many values of $s = p^n$, and that there are always non-Desarguesian planes among these sets except in the case that $p^n - 1$ is prime.

Such an enumeration has been accomplished for many small values of s using the Maniac III computer. The results show striking regularities. The case of $s = 3^4 = 81$ is listed below. This case is exhibited because it is one of the lowest orders to show the complexity in the structure of such tables that seems to be characteristic when s is sufficiently large.

The rows of Table 3 will be called incomplete patterns. The values of r characterizing each incomplete pattern will be called the submodulus of that incomplete pattern.

For $s \leq 121$, tables such as Table 3 were generated. The entries in them are consistent with the following theorem:

Theorem 1. Given $s = p^n$, $s \leq 121$, the values of x for which $L_{1,1} \not\subset L_{x,r}$ (r is relatively prime to $s-1$) form an incomplete pattern with period $q(r) = q/m$, with m some integer and

$$(1-7) \quad q = (r-1, s-1). *$$

It is conjectured that (1-7) is valid for all $s = p^n$.

* (x, y) indicates the g. c. d. of x and y .

r	$x \rightarrow$
1	2, 3, 4, 5, 6... (all values of x except 1)...
3	2, 4, 6, 8... ($x \equiv 0 \pmod{2}$)
7	-
9	2, 3, 4, 5, 6, 7, 8, 10, 11... (all values except $x \equiv 1 \pmod{8}$)
11	-
13	-
17	9, 25, 41, 57, 73 ($x \equiv 9 \pmod{16}$)
23	-
27	2, 4, 6, 8... ($x \equiv 0 \pmod{2}$)
29	-
31	-
33	9, 25, 41, 51, 73 ($x \equiv 9 \pmod{16}$)
37	-
39	-
41	5, 6, 8, 11, 12, 13... ($x \equiv 5, 6, 8, 11, 12, 13, 16, 18, 20, 21, 22, 24, 26, 29, 30, 31, 34, 36, 37 \pmod{40}$)
43	-
47	-
49	5, 13, 21, ... ($x \equiv 5 \pmod{8}$)
51	6, 16, 26, ... ($x \equiv 6 \pmod{10}$)

(no higher values of r yield a square $L_{x,r} = L_{1,1}$).

Table 3. Table of values of x for each r relatively prime to $s-1 = 80$ for which $L_{x,r} \neq L_{1,1}$.

In Table 3, all the values of r for which there are incomplete patterns have period $(r-1, s-1)$ except $r = 49$, for which the period is $(r-1, s-1)/2$.

Consider now the set of column transformations $T_{x,r}$ defined by $L_{x,r} = T_{x,r} L_{1,1}$. It is helpful to consider the transformational properties of $T_{x,r}$. It is in general true that under the operation $\circ$ neither associativity nor commutativity holds; therefore the various $T_{x,r}$ do not form a group. ++

Clearly, however, there always exists some pair (y, z) for which $T_{x',r'} T_{x,r} L_{1,1} = T_{y,z} L_{1,1}$. One may show straightforwardly that $T_{x',r'} T_{x,r} = T_{y,r' \cdot r}$. The value of y may be computed easily. The successive columns of $T_{x,r} L_{1,1}$ are respectively $0, x, x+r, x+2r, \dots$. Therefore column number l of $T_{x',r'} T_{x,r} L_{1,1}$ is just the x^{th} element of the above sequence, that is,

$$(1-8) \quad y = x + (x'-1) \cdot r^*$$

where y is of course reduced (mod' $s-1$).

The motivation for this digression is our desire to establish a method for converting the incomplete patterns of e. g. Table 3 into projective planes, when such a conversion is possible. The method consists simply of inserting $L_{1,x}$ into the incomplete pattern for each value of x that is not present in the incomplete pattern. For instance, such an insertion would result in the following fundamental patterns in

Table 3:

++ (1-3) is a group, but this is fortuitous. For larger s , groups are in general not obtained.

*where $\cdot$ appears in an expression, $+$ in the same expression will denote ordinary addition (mod' $s-1$).

$$(1-9) \quad r = 3: 1, 3$$

$$r = 9: 1, 9, 9, 9, 9, 9, 9, 9$$

$$r = 17: 1, 1, 1, 1, 1, 1, 1, 1, 17, 1, 1, 1, 1, 1, 1, 1$$

and so forth. When an incomplete pattern is filled in in this way, we shall call the result a primitive pattern, and string such as those in (1-9) will be called primitive fundamentals.

We will now prove that the primitive patterns resulting from the filling of Table 3 are projective planes when $r = p^m$, and in no other case.

Each letter of a primitive fundamental will be called an element.

As we already have seen, a primitive pattern results from a repeated concatenation of primitive fundamentals. Each repetition of the fundamental includes a repetition of each element; each repetition of an element will be called an image of that element. An element together with its images will be called an image set. The image set of an element E will be denoted $|E|$.

We note first that for $x \neq x'$, $L_{x,1} \not\subset L_{x',1}$. Operating on both $L_{x,1}$ and $L_{x',1}$ with the column transposition operator $T_{1,r}$, we obtain

$$(1-10) \quad L_{x,r} \not\subset L_{x',r}$$

Next, we note that if $T_{x',r} T_{x,r} L_{1,1} = T_{y,r} r' L_{1,1}$, then by (1-8) we have

$$(1-11) \quad y = x' + x - 1 \pmod{r-1}.$$

Now by (1-7), we know that the period $\ell(r)$ of the primitive fundamental of submodulus r is a divisor of $r-1$. From (1-11) it then follows that

$$(1-11a) \quad T_{x', r'} T_{x, r}^{L_{1,1}} \in |T_{x'+x-1, r \cdot r'}^{L_{1,1}}|^{\ell(r)}$$

where we have introduced for the sake of clarity a superscript on the image set to indicate the length of the fundamental pattern with respect to which the image set is defined.

Consider now any $L' \in |T_{x, r}^{L_{1,1}}|^{\ell(r)}$. Obviously

$$(1-11b) \quad L' = T_{x+k \cdot \ell(r), r}$$

for some integer k . From (1-11a),

$$(1-11c) \quad T_{x', r'}^{L'} \in |T_{x'+x+k \cdot \ell(r)-1, r \cdot r'}^{L_{1,1}}|^{\ell(r)}$$

But $T_{x'+x+k \cdot \ell(r)-1, r \cdot r'}^{L_{1,1}}$ is an image of $T_{x'+x-1, r \cdot r'}^{L_{1,1}}$.

Therefore

$$(1-11d) \quad T_{x', r'}^{L'} \in |T_{x'+x-1, r \cdot r'}^{L_{1,1}}|^{\ell(r)}.$$

Consider now all $L' \in |T_{x,r} L_{1,1}|^{l(r)}$. Operating on each L' with $T_{x',r'}$ we obviously obtain distinct elements of $|T_{x'+x-1,r \cdot r'} L_{1,1}|^{l(r)}$. Therefore there is a one-one-mapping of the form

$$(1-11e) \quad T_{x',r'} |T_{x,r} L_{1,1}|^{l(r)} = |T_{x'+x-1,r \cdot r'} L_{1,1}|^{l(r)}.$$

In this sense the mappings of image sets form a group.

We are in a position to determine which primitive patterns actually correspond to projective planes. Let us first determine whether $L_{x,1}$ is orthogonal to $L_{y,r}$, where $L_{x,1} \in |L_{1,1}|^{l(r)}$ and $L_{y,r} \in |L_{1,r}|^{l(r)}$. We have $T_{x,1} L_{1,1} = L_{x,1}$ and $T_{x,1} |L_{y,r}|^{l(r)} = |L_{x+y-1,r}|^{l(r)}$. Therefore $L_{x,1}$ is orthogonal to $L_{y,r}$ if and only if $L_{1,1}$ is orthogonal to some $L' \in |L_{y-x+1,r}|^{l(r)}$, which can be read off of tables like Table 3 directly. When this is done, it is found that the filling procedure never yields a plane for $r \neq p^m$ ($m < n$), at least for $s \leq 121$. We conjecture that this is always true.

Let us then turn our attention to the case $r = p^m$. Tables such as Table 3 have been assembled for $s \leq 3^6 = 729$. Every case of $r = p^m$ has been examined. The results may be expressed as

Theorem 2. Given $s = p^n$, $n \geq 1$, $s \leq 729$, $r = p^m$, $0 < m < n$, there exists a projective plane with a primitive fundamental pattern of the form

$$(1-12) \quad \Pi(r, s) = 1, r, r, \dots, r, r$$

where the number of elements in Π , e. g. the period, is precisely

$$(1-13) \quad \ell(r) = (r-1, s-1) = (p^m - 1, p^n - 1).$$

In the case $r = p^m$ (hereafter r will always denote p^m) the missing Latin squares that are "filled" are all images of $L_{1,1}$. This fact, together with an application of the reasoning immediately preceding, establishes that the primitive fundamentals of (1-12) are all projective planes.

As an illustration, for $s = 16$, we have $s-1 = 15$; for $r = 2^2 = 4$, we obtain $(r-1, s-1) = 3$, and thereby a primitive fundamental pattern $\Pi(4, 16) = 1, 4, 4$, which is clearly non-Desarguesian.

The primitive fundamentals (1-12) may be generalized to form new sets of mutually orthogonal Latin squares that may in some cases represent new planes. To begin with, we may prove

Theorem 3. In any primitive fundamental pattern, $\Pi(r, s)$, any number of "r"s may be replaced by "1"s; the result will still represent a projective plane.

Proof. To establish the orthogonality of $|L_{x,1}|^{\ell(r)}$ and $|L_{y,r}|^{\ell(r)}$, $(L_{y,r} \neq |L_{x,r}|^{\ell(r)})$, operate on both with $T_{-x+2,r}$ obtaining respectively $|L_{1,1}|^{\ell(r)}$ and $|L_{y-x+1,r}|^{\ell(r)}$, the orthogonality of which is clear from Theorem 2. Q. E. D.

Theorem 3 raises the interesting question of just how much substitution is possible within a primitive fundamental. Are there symbols other than "1" that can be substituted into (1-12) such that a plane is still

represented? Some simple results are first needed:

Theorem 4. A fundamental pattern of submodulus $r = p^m$ has a length that is a divisor of the length of a fundamental pattern of submodulus $r = p^{km}$, where $km < n$.

Proof: $p^m - 1$ divides $p^{km} - 1$, therefore $(p^m - 1, p^n - 1)$ divides $(p^{km} - 1, p^n - 1)$. Q. E. D.

Theorem 5. If a primitive fundamental pattern $\pi(p^m, p^n)$ is concatenated with itself $(p^{km} - 1, p^n - 1) / (p^m - 1, p^n - 1)$ times, the result is a fundamental pattern of length $(p^{km} - 1, p^n - 1)$ that represents a plane.

Proof: follows directly from Theorems 3 and 4.

From this point on, "patterns" will always be assumed to mean "patterns representing planes".

We shall adopt the notation $\Psi(p^m, p^n)$ to represent a fundamental pattern (in general, not primitive) for projective planes of order p^n , with length $(p^m - 1, p^n - 1)$.

Theorem 6. If p is prime, then $(p^n - 1, p^m - 1) = p^{(n,m)} - 1$.

Proof: Clearly $p^{(n,m)} - 1$ divides both $p^n - 1$ and $p^m - 1$. Hence $p^{(n,m)} - 1$ divides $d = (p^n - 1, p^m - 1)$. Let us then show that d divides $p^{(n,m)} - 1$. It must be the case that d divides $p^{kn} - 1$ and $p^m - 1$, with k and l integers, since $p^n - 1$ divides $p^{kn} - 1$ and $p^m - 1$ divides $p^{lm} - 1$. Therefore d divides $(p^{kn} - 1) - (p^{lm} - 1) = p^{kn} - p^{lm} = p^{lm}(p^{kn-lm} - 1)$. If d is 1, then d divides $p^{(n,m)} - 1$. If d is not 1, then d does not divide p^{lm} , and therefore d divides $p^{kn-lm} - 1$. For suitable choices of k, l , $(n, m) = kn - lm$. Hence d divides $p^{(n,m)} - 1$. Q. E. D. (The above proof was

contributed by S. Lin and R. L. Graham.)

Theorem 7. Let $v \geq n$. Then $(p^v - 1, p^n - 1) = (p^q - 1, p^n - 1)$ where $q \equiv v \pmod{n}$ and $q < n$.

Proof: By Theorem 6, $(p^v - 1, p^n - 1) = p^{(v, n)} - 1 = p^{(kn+q, n)} - 1$ where k and q are integers. But $(kn+q, n) = (q, n)$, which proves the theorem.

Theorem 8. Theorem 4 can be strengthened as follows: A fundamental pattern for $s = p^n$ and $r = p^m$ has a length that is a divisor of the length of the fundamental pattern for $r = p^{km}$. *

Proof. If $km \leq n$, Theorem 8 is true by Theorem 4. If $km > n$, let $q = km \pmod{n}$. Then clearly q is a multiple of m . But by Theorem 7, $(p^{km} - 1, p^n - 1) = (p^q - 1, p^n - 1)$ which is by Theorem 4 again a multiple of $p^m - 1$. Q. E. D.

From Theorem 8, we see that the values of $(p^m - 1, p^n - 1)$ are arranged in patterns which sharply reflect the cycle structure of n . In particular, it now becomes clear that Theorem 3 may be fully generalized as follows:

Theorem 9. A fundamental pattern $\Psi(p^m, p^n)$, with $m|n$ (i. e. m divides n) and $m < n$ can be composed of any sequence, whatsoever of elements chosen from the set $Q = \{p^m, p^{2m}, \dots, p^n\}$; the pattern resulting from it will represent a plane.

Theorem 9 is the key theorem in this paper. It asserts that any algebra based on a field of order p^n whose multiplication rule is given by

*In order to make sense, r is always reduced $\pmod{p^n - 1}$.

$$(1-14) \quad \alpha_i \circ \alpha_t = \alpha_i * \alpha_t^{k(i) \cdot m},$$

$$m|n, m < n, k \cdot (i) = k(i') \text{ when } i \equiv i' \pmod{p^m - 1},$$

yields a plane for any permissible choice whatsoever of positive integers $k(i)$. (For a given algebra, m is fixed.)

This theorem will now be proved on the basis of our computer results, which extend up to order 729. However, a straightforward proof of (1-14) has been discovered by R. L. Graham,* which establishes the validity of these formulations for any order. The proof will be exhibited in a subsequent paper.

Proof. Let $q' \in Q$ and $q'' \in Q$, and let $x \neq x'$ and $m|n$. Consider two elements $\sigma \in |L_{x', q'}|^{l(p^m)}$ and $\tau \in |L_{x'', q''}|^{l(p^m)}$. We need only to show that for arbitrary $q', q'', x', x'', \sigma \neq \tau$.

By Theorem 8, $l(p^m) | l(q')$ and $l(p^m) | l(q'')$. Therefore we can write

$$(1-14a) \quad |L_{x', q'}|^{l(p^m)} = |L_{x', q'}|^{l(q')} \cup |L_{x'+l(p^m), q'}|^{l(q')} \cup \dots$$

and a similar equation for $|L_{x'', q''}|^{l(p^m)}$ as a union of sets of submodulus $l(q'')$. From (1-11e) we find

*Private communication

$$(1-14b) \quad T_{-x'+2, n/q'} |L_{x, q'}| l_{(p^m)} = |L_{1, 1}| l_{(q')} \cup |L_{1+l_{(p^m)}, 1}| l_{(q')} \cup \dots$$

$$= |L_{1, 1}| l_{(p^m)}$$

and by similar reasoning,

$$T_{-x'+2, n/q'} |L_{x'', q''}| l_{(p^m)} = |L_{x''-x'+1, n \cdot q''/q'}| l_{(p^m)}.$$

Let $q = n \cdot q''/q'$; evidently $q \in Q$. Let $y = x''-x'+1$; clearly $y \neq 1$. We thus need only show that

$$(1-14c) \quad |L_{1, 1}| l_{(p^m)} \not\subseteq |L_{y, q}| l_{(p^m)}.$$

But by Theorem 3, the primitive fundamental pattern for $r = q$ may have "1"s inserted anywhere in the place of q . Since the length of this fundamental pattern for $r = q$ is, by Theorem 8, a multiple of the length of the fundamental pattern for $r = p^m$, make these "1" substitutions so that they are spaced precisely $l_{(p^m)} = p^m - 1$ apart, including the leading "1" present in the primitive fundamental. We obtain in this way a new fundamental pattern of length (p^m) , of submodulus q , representing a plane. But this plane is the one for which (1-14c) holds, thus validating the theorem.

We are now ready to discuss the enumeration of distinct sets

of orthogonal Latin squares under arbitrary transformation $T_{x',q'}$. (We postpone until Section 3 the discussion of independence under the general transformation specified at the beginning of Section 1). From (1-14) we may visualize the effect of $T_{x',q'}$ on any symbol in $\Psi(p^m, p^n)$ in the following way: Consider a "cyclic alphabet" containing the n/m symbols $p^m, p^{2m}, \dots, 1$, in that order, and a "cyclic code word" (p^m, p^n) of length (p^m-1, p^n-1) ($=p^m-1$ if m divides n). The effect of $T_{x',q'}$ for $q' = p^{km}$ can be looked at as (1) replacement of the symbol by another symbol separated k places "from it" in the alphabet, and (2) cyclic "transportation" of the symbol by $x'-1$ places within the code word. Applying $T_{x',q'}$ to every symbol in $\Psi(p^m, p^n)$ we are led to the following theorem:

Theorem 10. The number of distinct sets of mutually orthogonal Latin squares in a fundamental pattern $\Psi(p^m, p^n)$ with m dividing n , under arbitrary column transformations $T_{x',q'}$ with $q' \in \{p^m, p^{2m}, \dots, 1\}$, is equal to the number of classes of code words of length p^m-1 with letters chosen within a "cyclic alphabet" of n/m letters that are non-equivalent under any iteration of (1) cyclic permutation of the code word, or (2) replacement of each letter by another letter, the "distance" traversed within the "cyclic alphabet" being the same for each replacement.

Proof: Already established.

Theorem 11. For a code word of length ℓ , and a cyclic alphabet of length q , the number of distinct classes of code words $\mathcal{L}(\ell, q)$ detailed in Theorem 10 is given by

$$(1-15) \quad \nu(l, q) = \frac{1}{l_q} \sum_{t=0}^{q-1} \sum_{k=0}^{q-1} q^{(k, l)},$$

where in the second sum (over k), only those values of k are allowed for which

$$(1-15a) \quad lt/(l, k) \equiv 0 \pmod{q}$$

Proof: We use Frobenius' classic theorem, according to which the number of equivalence classes under a group G is given by

$$(1-16) \quad \nu = \frac{1}{g} \sum_{\alpha \in G} I(\alpha),$$

where $I(\alpha)$ is the total number of elements of G left invariant under α and g is the order of G .¹⁵ (We note also that (1-14) indicates that $T_{x, q}$ can be thought of as a group relation).

The number of elements in the group, g , is clearly equal to the product of the number of letters in the codeword l and the number of letters in the alphabet, q . Now consider a transformation in which there is no "alphabet rotation," corresponding to $t = 0$ in (1-15).

Consider a cyclic code-word permutation by k places, $k = 0, 1, \dots, l-1$. Code-words will be mapped into themselves if and only if each sequential segment of size (k, l) is identical. But there are $q^{(k, l)}$ such segments possible. In the case that there is an alphabet rotation $t = 1, 2, \dots, q-1$,

nearly the same situation as above arises, except that each segment must be the result of applying an "alphabet rotation" by t "letters" to the segment separated from it by k places cyclically in the code-word. This is possible if and only if t times the number of sectors $l/(l, k)$ is a multiple of q . We are led directly to (1-15). Q. E. D.

The problem remaining is to show how to combine the values of $\nu(l, q)$ for $l = p^m - 1$, and m the various divisors of n . It is not quite right to add the values of ν for each value of m , as there is some redundancy. Consider two values of m , called m' and m'' , both less than n and dividing n , and suppose m' is less than m'' and divides m'' . Then (1) the length of the fundamental pattern for m'' is a multiple of the length of the fundamental pattern for m' , and (2) the letters in the fundamental pattern for m'' are all present in the fundamental pattern for m' (though of course not vice versa). Consequently, there is a class of fundamental patterns for m'' already enumerated in the fundamental patterns for m' for all m' dividing m'' . The number of classes of these is just the number of classes of code-words that can be formed with the letters of the fundamental pattern for m'' (of which there are n/m'') embedded into a code-word whose length is $l(p^{m''})$. These considerations lead at once to a recursive equation for the total number of independent sets of mutually orthogonal Latin squares $N(p^n)$ under any column transformation $T_{x, q}$:

$$(1-17) \quad N(p^n) = 1 + \sum_{\substack{m \leq n \\ m|n}} \mu(p, m, n/m)$$

where

$$(1-17a) \quad \mu(p, m, x) = \nu(p^{m-1}, x) - 1 - \sum_{\substack{u \leq m \\ u|m}} \mu(p, u, x)$$

and (λ, q) is given by (1-15). The 1 in (1-17) is required because the sum (1-17) goes over all m dividing n , in principle; but $\mu(p, n, m=n, 1)$ is always 1 (the previous paragraph concerning redundancy being inapplicable in this case). Similarly, the 1 in (1-17a) is subtracted because the enumeration $\nu(p^{m-1}, x)$ always counts the Desarguesian plane $\mu(p, n, m=n, x)$ once, and there is just one of these.

It now remains to show that all the planes embraced by (1-17) are Veblen-Wedderburn planes. The conditions on addition are a priori satisfied; multiplication evidently forms a loop. To show that left-distributivity holds, we note that for our type of construction, multiplication can be represented by

$$(1-18) \quad \alpha_i \circ \alpha_t = \alpha_i * \alpha_{t \cdot p^{m(i)}} = \alpha_i * (\alpha_t)^{p^{m(i)}}$$

where, as usual, $s = p^n$ and $m(i)$ is some integer, a function of i , that is $\leq n$.

p^n	$N(p^n)$	p^n	$N(p^n)$	p^n	$N(p^n)$	p^n	$N(p^n)$
2^2	1	2^9	105	5^3	8	17^2	2,068
2^3	1	3^2	2	5^4	344,732	19^2	7,316
2^4	2	3^3	2	7^2	8	23^2	95,420
2^5	1	3^4	21	7^3	46	29^2	4,794,088
2^6	14	3^5	3	11^2	56		
2^7	1	3^6	1,291,150	13^2	180		
2^8	1,100	5^2	4				

Table 4. Some values of $N(p^n)$ for small values of p^n .

In table 4 $N(p^n)$ is tabulated for small values of p^n . *

(There are, of course, restrictions on $m(i)$ established by our previous discussion but it is not necessary to dwell on them here.) Now evidently

$$\begin{aligned}
 (1-19) \quad \alpha_i * (\alpha_a + \alpha_b) &= \alpha_i * (\alpha_a + \alpha_b)^{p^{m(i)}} \\
 &= \alpha_i * \sum_j \binom{p^{m(i)}}{j} \alpha_a^j \alpha_b^{p^{m(i)}-j}
 \end{aligned}$$

In (1-19) the binomial coefficient "multiplies" the α 's by means of "ordinary" multiplication, i. e., $3 \cdot \alpha^i = \alpha^i + \alpha^i + \alpha^i$, etc. The possibility

*The Maniac III computer was used for compiling Table 4.

of a binomial representation of course depends on the commutativity of addition.

It is not difficult to establish the following lemma:

Lemma 11.1 $\binom{p}{k} \equiv 0 \pmod{p}$ if $k \neq 0$ and $k \neq p$, else $\binom{p}{k} \equiv 1 \pmod{p}$. The proof is straightforward and will not be given.*

Evidently, if $u \equiv 0 \pmod{p}$ then $u \cdot \alpha_i = 0$. From Lemma 11.1 it then follows that

$$\begin{aligned} (1-20) \quad \alpha_i \circ (\alpha_a + \alpha_b) &= \alpha_i * \alpha_a^{p^{m(i)}} + \alpha_i * \alpha_b^{p^{m(i)}} \\ &= \alpha_i \circ \alpha_a + \alpha_i \circ \alpha_b \quad \text{Q. E. D.} \end{aligned}$$

2. Orthogonal sets of Latin squares generated by cyclic column transpositions of suborthomorphic squares.

It appears natural to try to combine the methods of Ref. 19 and the present paper. Let a square $\ell_{1,1}$ be generated from a suborthomorphism, and squares $\ell_{x,r}$ be generated from $\ell_{1,1}$ analogous to generating $L_{x,r}$ from $L_{1,1}$, viz:

$$(2-1) \quad \ell_{x,r} = T_{x,r} \ell_{1,1}$$

*A more general formulation is available in a forthcoming paper of Graham and Mrs. McWilliams¹⁶ to wit: $\binom{n}{k} \equiv 0 \pmod{p}$ (p a prime) if at least one digit of k is greater than the corresponding digit of n , when both n and k are expressed as numbers in the base p .

To study any large orders (>17) it is necessary to restrict our attention to some subset of the set of all suborthomorphisms in order to prevent an explosion in computing time. In fact, only suborthomorphisms having commutative and "suborthomorphic" symmetry were generated. This made it possible to study orders up to $s = 36$. The technique was to generate a table similar to Table 3, showing values of x for which $\ell_{x,r} \notin \ell_{1,1}$ for each value of r relatively prime to $s-1$. Interestingly enough, these incomplete patterns, as we have called them, display a property in common with those generated by operating on $L_{1,1}$; namely they are periodic with period

$$(2-2) \quad \dots (r-1, s-1)/m \quad \dots$$

as before.

The big difference is, of course, that not every value of x save 1 must be present when $r = 1$, as it must be for fields. Our strategy of "filling"--e.g. inserting $\ell_{x,1}$ in the incomplete pattern for the missing values of x --will not work here. In general, these incomplete patterns correspond simply to a few mutually orthogonal Latin squares. The tables for s a prime power seem to be richer in "content" even when $\ell_{1,1}$ is a suborthomorphism not equivalent to a field. For fields of order $s = p$ (prime), partial patterns with $(s-1)/2 - 2$ elements were found for many orders when $r = (s+1)/2$.

For instance, the orthogonality table for $s = 37$, with $L_{1,1}$ generated

from a field, is

$r = 1$	$2, 3, 4, 5, \dots (\not\equiv 1 \pmod{36})$
$r = 13$	$3, 15, 27, \dots (\equiv 3 \pmod{12})$
$r = 19$	$2, 4, 5, 7, 13, 15, 16, 18, 20, 22, 23, 25, 31, 34, 36$ $(\equiv 2, 4, 5, 7, 13, 15, 16, 18 \pmod{18})$
$r = 25$	$11, 23, 35 (\equiv 11 \pmod{12})$

Table 5.

Entries for which the partial pattern is void are omitted.

The entry for $r = 19$ above, when "filled" according to our previous prescription, is very nearly a projective plane, in that most of the squares are pair-wise orthogonal. This would seem to indicate that the non-existence of non-Desarguesian planes of prime order may be a very close question and should by no means be assumed.

We show here the richest orthogonality table obtained for a suborthomorphism of order not a power of a prime, for order $s = 22$. Six similar suborthomorphisms were obtained. A typical one is

(2-3) 1, 0, 10, 6, 20, 11, 3, 18, 15, 21, 2, 5, 16, 14, 13, 8, 12, 19, 7, 17, 4, 9.

For (2-3) the orthogonality table is

$r = 1$	$2, 9, 14, 21 (\equiv 2, 9, 14, 21 \pmod{21})$
$r = 13$	$2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21 (\equiv 2, 3 \pmod{3})$

Table 6.

Operating on $\lambda_{1,1}$ and $\lambda_{11,13}$ with $T_{2,1}$ we obtain respectively $\lambda_{2,1}$ and $\lambda_{3,13}$. Thus, $\lambda_{1,1}$, $\lambda_{2,1}$ and $\lambda_{3,13}$ form a triple of mutually orthogonal Latin squares. This triple is the lowest order triple for $s \equiv 2 \pmod{4}$ that has been obtained, to the best knowledge of the author.

For other orders, pairs were commonly obtained.

3. Independence under arbitrary column, row, and symbol permutation.

We have established that certain equivalence classes of sets of mutually orthogonal Latin squares may be enumerated that are independent under column transformations of the form $T_{x,q}$. The question then arises as to whether a set of mutually orthogonal Latin squares comprising an element of one of these equivalence classes might not be transformable to another element of a different equivalence class when the more general transformations listed at the beginning of Section 1 are taken into account.

In order for this to happen, such a general transformation Γ must of course have the property that $\Gamma L_{x,q} = L_{x',q'}$. The transformation of a fundamental pattern under Γ may be more general than the combination of code-word and alphabet "rotations" that we have shown to be equivalent to a column permutation $T_{x,q}$ and must be carefully examined.

Since any fundamental pattern can be transformed by a suitable $T_{y,r}$ to a form in which one of the component Latin squares is $L_{1,1}$, e.g. to a form in which its first symbol is 1, we shall start out with the assumption that $L_{1,1}$ is present. Clearly the most general transformation that takes $L_{1,1}$ into $L_{x,q}$ is

$$(3-1) \quad \Gamma = T_{x,q} U$$

where U is defined to be a combination of row, column, and symbol transformation satisfying

$$(3-2) \quad UL_{1,1} = L_{1,1}.$$

Now U can be broken into three parts: Ω , a row mapping; Ψ , a column mapping; and U_s , a symbol mapping. It is convenient to define

$$(3-3) \quad U_{rc} = \Omega \Psi = \Psi \Omega$$

so that

$$(3-4) \quad U = U_{rc} U_s = U_s U_{rc}.$$

The transformation corresponding to (3-1) that must be applied to each of the Latin squares in the fundamental pattern is

$$(3-5) \quad \Gamma' = T_{x,q} U'$$

where $U' = U_{rc} \sigma$, and σ is a symbol transformation that restores the square's column "0" to standard form. Since the other Latin squares

in the fundamental pattern are of the form $L_{x', q'}$ ($x' \neq 1$), we must ask ourselves what Latin square L' is obtained from

$$(3-6) \quad L' = U' L_{x', q'}$$

In particular, if a U' can be found such that $L' = U' L_{x', q'}$ and $L_{x'', q''}$ is not in the image set $\{L_{x', q'}\}$, it follows at once that the effect of Γ is broader than that of any column permutation $T_{x, q}$, and that the number of equivalence classes of fundamental patterns under Γ is smaller than under columns permutations alone. If, on the other hand, no such Γ can be found, then the above statement will necessarily be false.

If $L' = \sigma L''$, we say that $L' = L''$ to within a symbol transformation.

To begin our discussion, we note that $L_{1, 1}$ (written simply L from now on) satisfies

$$(3-7) \quad UL_{ij} = \alpha_i + \alpha_j$$

by definition. Therefore

$$(3-8) \quad UL_{ij} = U_s(\Omega \alpha_i + \Psi \alpha_j).$$

By hypothesis, therefore,

$$(3-9) \quad \alpha_i + \alpha_j = U_s(\Omega \alpha_i + \Psi \alpha_j).$$

Letting $\alpha_j = 0$ in (3-9) we obtain

$$(3-10) \quad \alpha_i = U_s(\Omega \alpha_i + \Psi \alpha_0).$$

Similarly, putting $\alpha_i = 0$ in (3-9) we obtain

$$(3-10a) \quad \alpha_i = U_s(\Psi \alpha_i + \Omega \alpha_0)$$

where we have deliberately substituted the dummy argument i for j .

Comparing (3-10) and (3-10a) and cancelling U_s we obtain

$$(3-11) \quad \Omega \alpha_i + \Psi \alpha_0 = \Psi \alpha_i + \Omega \alpha_0, \text{ or}$$

$$(3-12) \quad \Psi \alpha_i = \Omega \alpha_i + \Psi \alpha_0 - \Omega \alpha_0.$$

From (3-10) we may also deduce*

$$(3-13) \quad U_s = \Omega^{-1} - \Psi \alpha_0$$

so that (3-9) can be simplified to

*Transformations are applied to arguments from the right to the left in functional fashion; e. g. $\Omega = D + \beta$ means $\Omega : \alpha \rightarrow D(\alpha + \beta)$.

$$\begin{aligned} \alpha_i + \alpha_j &= (\Omega^{-1} - \Psi\alpha_0)(\Omega\alpha_i + \Omega\alpha_j + \Psi\alpha_0 - \Omega\alpha_0) \\ (3-14) \quad &= \Omega^{-1}(\Omega\alpha_i + \Omega\alpha_j - \Omega\alpha_0), \text{ or finally} \end{aligned}$$

$$\Omega(\alpha_i + \alpha_j) = \Omega\alpha_i + \Omega\alpha_j - \Omega\alpha_0.$$

We now have an equation for Ω alone. An exactly similar equation holds for Ψ .

It is quite easy to show that the most general solution for (3-14) and its companion equation for Ψ is

$$\begin{aligned} (3-15) \quad \Omega &= D + \beta, \\ \Psi &= D + \beta', \end{aligned}$$

where D is any mapping that is left distributive over addition, i. e. any automorphism over addition, and β, β' are arbitrary.

Relation (3-15) can be thought of in the following way. A row or column mapping that simply adds a fixed quantity to each element is, in effect, adding a fixed quantity to the whole square. Therefore the most general transformation U is one which permutes rows and columns with the same automorphism, then adds some fixed constant to the whole square.

Suppose we let the permutations corresponding to (3-15) operate on L' , one of the $L_{x,q}$. Putting

$$(3-16) \quad L'_{ij} = \alpha_i + \alpha_q \circ \alpha_j,$$

and applying U_{rc} , we obtain

$$(3-17) \quad U_{rc} L'_{ij} = \alpha_i + \alpha_q \circ (\Psi \alpha_j).$$

The second term of (3-17) must be considered carefully to understand why it is not written $\Psi(\alpha_q \circ \alpha_j)$. Ψ is a mapping defined on the elements of the group whose Cayley table is L . The operation U_{rc} includes a column permutation U_c which is of course the same as the permutation carried out when the mapping is performed on L . Suppose, for instance, that Ψ involved $\alpha_0 \rightarrow \alpha_6$, $\alpha_1 \rightarrow \alpha_6$, $\alpha_2 \rightarrow \alpha_4$. Then occupying columns 0, 1, 2 of ΨL would be columns 0, 6, and 4, respectively, of L . Now the columns of L' are respectively $\alpha_q \circ \alpha_0$, $\alpha_q \circ \alpha_1, \dots, \alpha_q \circ \alpha_{s-1}$. After application of the column permutation U_c , the columns of L' occupying columns 0, 1, and 2 of $U_c L'$ are $\alpha_q \circ \alpha_0$, $\alpha_q \circ \alpha_6$, and $\alpha_q \circ \alpha_4$. From this illustration, one can see that (3-17) is valid.

Combining (3-17) with (3-15) we get

$$(3-18) \quad U_{rc} L'_{ij} = (\beta_i + D\alpha_i) + \alpha_q \circ (\beta'_j + D\alpha_j).$$

Since by (3-10) we know that the operator $\circ$ is left-distributive, we can

write

$$(3-19) \quad U_{rc} L'_{ij} = D\alpha_i + \alpha_q \circ (D\alpha_j) + \alpha''$$

where $\alpha'' = \beta + \alpha_q \circ \beta'$ is some element of the addition group.

We have yet to specify the symbol transformation to be applied to (3-19). Our intention is, of course, that $U_{rc} L'_{ij}$ should be within a symbol transformation of $L_{x,q}$ for some x, q . For the sake of generality, however, consider some arbitrary one-one mapping P and define

$$(3-20) \quad L'' = \alpha_i + P\alpha_j$$

to be the result of using the mapping to permute the columns of L . Suppose it is the case that (3-19) is within a symbol transformation τ of L'' . Then

$$(3-21) \quad \tau(D\alpha_i + \alpha_q \circ (D\alpha_j) + \alpha'') = \alpha_i + P\alpha_j.$$

Let $\alpha_i = 0$ in (3-21). Then

$$(3-22) \quad P\alpha_j = \tau(D\alpha_0 + \alpha_q \circ (D\alpha_j) + \alpha'').$$

(3-22) may be solved for τ by first putting $\tau = \tau' - D\alpha_0 - \alpha''$.

We obtain

$$(3-23) \quad \tau = PD^{-1}(\alpha_q \circ)^{-1} - D\alpha_0 - \alpha''$$

where $(\alpha_q \circ)^{-1}$ is the inverse of the mapping $\alpha_q \circ$. Putting (3-23) into (3-21) we get

$$(3-24) \quad \alpha_i + P\alpha_j = PD^{-1}(\alpha_q \circ)^{-1}(D\alpha_i + \alpha_q \circ (D\alpha_j) - D\alpha_0).$$

Now $D\alpha_0 = 0$ since D is an automorphism, and $(\alpha_q \circ)^{-1}$ is left-distributive. One can write, therefore

$$(3-25) \quad \alpha_i + P\alpha_j = PD^{-1}(\alpha_q \circ)^{-1}D\alpha_i + P\alpha_j$$

Subtracting and comparing, one obtains

$$(3-26) \quad 1 = PD^{-1}(\alpha_q \circ)^{-1}D \text{ or finally, the key equation}$$

$$(3-27) \quad P = D^{-1}(\alpha_q \circ)D$$

Now this entire train of argument is reversible, so we have the following theorem:

Theorem 12. Consider a row, column, symbol transformation U satisfying $UL = L$. The most general form of U is given by $(UL)_{ij} = D(\alpha_i + \alpha_j) + \alpha'$, where D is an arbitrary automorphism of the addition

group and α' is an element of the addition group. Consider now two automorphisms P and Q , from which two Latin squares $L_{i,j}^P = \alpha_i + P\alpha_j$ and $L_{i,j}^Q = \alpha_i + Q\alpha_j$ are defined. Then UL^P will be within a symbol transformation of L^Q if and only if $Q = D^{-1}PD$.

4. Concluding Remarks

With the inclusion of one further step, now under study, we shall hopefully be able to specify a lower bound to the number of independent sets of mutually orthogonal Latin squares for all orders. It is an attractive idea that many of these, perhaps all may represent distinct projective planes. Such a possibility is worth investigating. The combinatoric scheme wherein these are enumerated is especially simple. All the planes included are Veblen-Wedderburn planes, but not all known VW planes are included in the formulation, the division ring planes of Knuth's¹⁷ of order 32, being among the most prominent. In fact, it appears likely that the algebras we have discussed include no division ring planes at all, save the Desarguesian plane.

Work is now in progress to determine the number of independent sets of mutually orthogonal Latin squares constructed by our method. The idea is to rely on Theorem 12, and to search for automorphisms D that transform patterns of Latin squares $\Psi(m,n)$ into similar patterns $\Psi'(m,n)$ via Theorem 12. The results are encouraging.

Acknowledgements. I wish to acknowledge the advice and encouragement given me by A. A. Albert, R. L. Graham, S. Lin, P. Bateman,

L. E. MacMahon, R. L. Ashenurst, and E. T. Parker. Among those whose programming assistance is gratefully acknowledged are Messrs. C. L. Robinson, W. L. Mixon, and Mrs. Minja Choe, who also made numerous contributions to the arguments developed in this paper, pointed out several errors, and made many useful suggestions for its improvement.

REFERENCES

1. D. M. Johnson, A. L. Dulmage, and N. S. Mendlesohn, Orthomorphisms of Groups and Orthogonal Latin Squares, Can. J. Math., 13, (1961), pp. 356-372.
2. R. C. Bose, On the Application of the Properties of Galois Fields to the Problem of Construction of Hyper-Greco-Latin Squares, Sankhya, 4 (1938), 323-338.
3. O. Veblen and J. H. Maclagan-Wedderburn, Non-Desarguesian and Non-Pascalian Geometries, Trans. Am. Math. Soc., 26, 379-388.(1907)
4. D. R. Hughes, Review of Some Results in Collineation Groups, University of Chicago, October, 1959, code AFOSR TN S9-1124.
5. R. H. Bruck and H. J. Ryser, The Nonexistence of Certain Finite Planes, Can. J. Math., 1 (1949), pp. 88-93.
6. L. Euler, Recherches sur un espèce des quarrés magiques, Verh. Zeeuwsch. Genoot. Weten Vliss, 9 (1782), pp. 85-239.
7. G. Tarry, Le problème des 36 officiers, Assn. Franc. Av. Sci., 29 (1900), pp. 170-203.
8. E. T. Parker, Orthogonal Latin Squares, Proc. Nat. Acad. Sci., 45 (1959), pp. 859-862.
9. R. C. Bose, S. S. Shrikhande, and E. T. Parker, Further Results on the Construction of Mutually Orthogonal Latin Squares and the Falsity of Euler's Conjecture, Can. J. Math., 12, (1960), pp. 189-203.

10. E. T. Parker, Computer Investigations of Orthogonal Latin Squares of Order Ten, Proc. of Symposia on Applied Math., 15, (1963), pp. 73-81.
11. R. C. Bose, S. S. Shrikhande, R. K. Bhattacharya, On the Construction of Group Divisible Incomplete Block Designs, Ann. Math. Stat., 24, (1953), 167-195.
12. H. B. Mann, The Construction of Orthogonal Latin Squares, Ann. Math. Stat., 13, (1942), 418-423.
13. L. E. Dickson, On Finite Algebras, Gottinger Nachrichten, 1905, pp. 358-394.
14. D. R. Hughes, A Class of Non-Desarguesian Projective Planes, Can. J. Math., 9, (1951), 378-388.
15. Michael A. Harrison, Introduction to Switching and Automata Theory, McGraw-Hill, New York, 1965.
16. R. L. Graham and J. McWilliams (to be published).
17. D. L. Knuth, Finite Semifields and Projective Planes, J. Algebra, 2, (1965), pp. 182-217. See also R. J. Walker, Determination of Division Algebras with 32 Elements, Proc. of symposia in Appl. Math., 15, (1962), 83-85.
18. A. A. Albert, Finite Division Algebras and Finite Planes, Proc. Symp. Appl. Math., 10, Amer. Math. Soc., Providence, R. I., 1960.
19. G. K. Manacher, Some New Methods for Studying Sets of Orthogonal Latin Squares, ICR Quarterly Report, U. of Chicago, October, 1966.

ERRATA

P. C-10, line 17, put $L_{x,1}$ in place of $L_{1,x}$.

P. C-22, line 6, put $\mu(p, m=n, 1)$ in place of $\mu(p, n, m=n, 1)$.

P. C-22, line 9, put $\mu(p, m=n, x)$ in place of $\mu(p, n, m=n, x)$.

FORCE CORRELATIONS IN FINITE GRAVITATIONAL SYSTEMS

R. H. Miller

1. Introduction

In stellar dynamics a stellar system is pictured as a swarm of stars, each moving under the gravitational attraction of all the other stars in the system. With a system consisting of many stars (clusters and galaxies may have from 1000 to 10^{12} stars), it is customary to separate the forces into those derivable from a smooth potential with a superposed fluctuating force. The smooth potential represents the effect of all the remaining stars in the system, while the fluctuating part is attributed to nearest-neighbor two-body interactions. Relaxation effects are ascribed to the fluctuating part, which should somehow appear in the diffusion coefficients of a Fokker-Planck equation. An attempt to calculate the fluctuating part leads to logarithmically divergent integrals, from which Chandrasekhar (1960) showed that the divergence due to close encounters can be removed by considering the details of the hyperbolic orbits during the encounter. The divergence at large separations remains and represents a fundamental obstacle to the formulation. Attempts to include three-body encounters would require solutions to the celebrated "three body problem."

Some years ago, Chandrasekhar (1943) attempted to treat the fluctuating part as a stochastic process, regarding the stars as travelling on straight line orbits in an infinite homogeneous medium of point stars.

The distribution of forces acting on a star is a Holtsmark distribution. However, the rate of change of forces is also important; Chandrasekhar attempted to work this out in a series of papers with von Neumann (1942, 1943, 1944a, b), but no complete solution was obtained. Several approximations were worked out: the first and second moments of the distribution, the correlation of forces at separated points, and the correlation of forces acting on a given star at different times. Unfortunately, the correlations die out very slowly--inversely in the time while an exponential decrease would be expected for Markovian behavior. Prigogine (1966 a, b) has recently remarked on the non-Markovian character of these systems.

A finite system might look quite different. Finite systems are significantly more difficult to handle analytically. Bertiau and Roberts (1958) calculated the distribution of forces at the center of a spherical cluster; they found that the Holtsmark distribution is unaltered for large forces (even to keeping the divergence), but that the tail at small forces is modified, as might have been expected. The finite system is amenable to machine calculation. A study of the force correlations at separated points in a finite static system is reported here. The static system avoids the need to know details of orbits or to have a means of treating escapes, but cannot provide any information concerning temporal development of the force correlations. Even so, some kind of a mass distribution must be assumed. The problem was so formulated that models could be synthesized later, but no need developed.

2. The Calculation

The correlations were determined by means of Monte Carlo calculations. The strategy was to build up an ensemble of representative systems in such a way that they could be assembled into various kinds of larger systems. Because force components are simply additive, the force components at certain points were tallied and stored (on a disk file) for each member of the ensemble.

Ensembles were built up for each of three kinds of systems. Each ensemble consisted of 100 systems of the appropriate kind. Each system, in turn, had 1000 stars at "random" locations to approximate a uniform density within the appropriate boundaries. The three kinds of systems were (1) interior to the unit sphere; (2) in the spherical shell between radii 1 and 2; and (3) in the spherical shell between radii 2 and 3. It was hoped that some analytic approximation might be used beyond $r = 3$ (Camm, 1963). The different densities that result from placing the same number of stars in different volumes are of no concern. For each ensemble member, the force components that would act on a fictitious unit test mass were calculated for a set of points symmetrically disposed along a diameter of the unit sphere. Different configurations may be stepwise approximated by taking the appropriate number of systems of each kind, summing the forces, and building up ensembles of such configurations in a combinatorial manner. If the same basic system is used more than once in this process, the forces will not be statistically independent.

Finally, the force correlations may be evaluated as suitably normalized ensemble averages of products of the force components. It is convenient to evaluate the force correlations for pairs of points that are symmetrically disposed about the center, in order to make use of the simplifications that result from symmetry considerations. Although translational invariance no longer holds for this (inhomogeneous) system, the extra symmetry resulting from the choice of field points and the subtraction of the expected average force due to stars inside leaves the same terms as are left with velocity correlation tensors in homogeneous isotropic incompressible turbulence (Batchelor, 1953). The only nonzero components of the force correlation tensor are diagonal with the longitudinal term outstanding; these are expressible in terms of two scalar functions, but there is a relation between them. Thus, if $F_i^{(k)}(a)$ represent the i^{th} component ($i = 1, 2, 3$) of the force evaluated for the k^{th} member of the selected ensemble at a point whose distance from the center is a , while $F_j^{(k)}(-a)$ represents the j^{th} component of the force at the diametrically opposed point for the same member of the ensemble, the tensor $C_{ij}^{(k)}(a) = F_i^{(k)}(a) F_j^{(k)}(-a)$ represents the required product of forces. Then, for an ensemble containing N members, the expectation and covariance of $C_{ij}(a)$ are:

$$C_{ij}(a) = \frac{1}{N} \sum_{k=1}^N C_{ij}^{(k)}(a) \quad (1)$$

and

$$\text{Cov}(C_{ij}(a)C_{i'j'}(a)) = \frac{1}{N} \sum_{k=1}^N C_{ij}^{(k)}(a)C_{i'j'}^{(k)}(a) - C_{ij}(a)C_{i'j'}(a) \quad (2)$$

where $C_{ij}(a)$ should be 0 for $i \neq j$. Because the Monte Carlo process is equivalent to the use of sample distributions, sample covariances between different ensemble members will not be zero as has been assumed in writing equations 1 and 2. If any one system of the basic three kinds were used more than once in constructing model systems, explicit allowance would have to be made for the resulting nonzero covariance between different ensemble members. No basic system was used more than once in any of the work reported here.

The full sample distribution might be displayed, rather than its moments. The tensors introduced above are moments of the actual distributions, and lose information about the details of the distributions as moments always do. They were used in preference to the full sample distributions because of the bulk of information that would be involved, and because theoretical comparison should be easier for the moments.

3. Results

The force correlations were first worked out separately for each of the three kinds of basic system, in order to develop some feeling for the indicated trends. These were of such a character that development of more complex models did not seem warranted.

The results obtained as ensemble averages over 100 systems of each of the basic three kinds are displayed in Figures 1, 2, and 3. The off-diagonal terms all vanished satisfactorily. The mean square force components at the center, for comparison purposes, are:

$$\overline{F_x^2(0)} = C_{xx}(0) = 80,000 \quad (\text{Set 1: } 0 \leq r < 1; \text{ figure 1}),$$

$$\overline{F_x^2(0)} = 70 \quad (\text{Set 2, } 2 \leq r < 3; \text{ figure 2}),$$

and

$$\overline{F_x^2(0)} = 10.5 \quad (\text{Set 3, } 2 \leq r < 3; \text{ figure 3}).$$

Widely different scales are used for the ordinates in the three figures. It is evident from these figures that the innermost sphere dominates the force correlations, and that while long range terms are present they are very much smaller than the short range terms. The bulk of the correlation effect has a range on the order of the mean near-neighbor distance. In view of this striking dominance of the nearest stars, it did not seem worthwhile to construct models with different envelopes. The dominant terms are contributed by a few stars near the test point.

REFERENCES

- Batchelor, G. K., 1953: The Theory of Homogeneous Turbulence (Cambridge: at the University Press).
- Camm, G. L. 1963. M. N. 126, 283-298.
- Chandrasekhar, S. 1943. Rev. Mod. Phys. 15, 1-89.
- _____ 1944a. Ap. J. 99, 25-46.
- _____ 1944b. Ap. J. 99, 47-53.
- _____ 1960. Principles of Stellar Dynamics (Chicago: University of Chicago Press).
- Chandrasekhar, S., and von Neumann, J. 1942. Ap. J. 95, 489-531.
- _____ 1943. Ap. J. 97, 1-27.
- Prigogine, I. 1966a. Nature 209, 602-3.
- _____ and Severne 1966b. Physica 32, 1376-96.
- Roberts, P. H. and Bertiau, F. 1958. Ap. J. 128, 130-138.

FIGURE CAPTIONS

- Figure 1. Correlations between force components evaluated at points symmetrically disposed about the center of a spherical system containing 1000 stars within the unit sphere. The ordinate is the longitudinal correlation (the correlation of x-components of forces where the separation is in the x-direction). The other components behave similarly. The abscissa represents the distance of each of the sampling points from the center of the system. The shaded region shows the dispersion of the $x x$ component of the correlation tensor.
- Figure 2. Correlations between force components evaluated at points symmetrically disposed about the center of a spherical shell containing 1000 stars between radii 1 and 2. Only the longitudinal component is shown. The error bars represent the dispersion of the $x x$ component of the correlation. The smooth pattern disappears at points near the region containing stars.
- Figure 3. Same as Figure 2, but with 1000 stars in a spherical shell with $2 \leq r < 3$. All the correlations shown are ensemble averages.

Fig. 1

Mean Near Neighbor Distance

α
Fig. 2

Fig. 3

REMARKS ON THE APPROXIMATION-SOLVABILITY OF NONLINEAR FUNCTIONAL EQUATIONS

W. V. Petryshyn

Introduction. Let X be a separable reflexive Banach space, X_n a subspace of X and P_n a linear projection of X onto X_n . Let " $\rightarrow$ " and " $\rightharpoonup$ " denote respectively strong and weak convergence in X . A sequence $\{X_n\}$ of finite dimensional subspaces of X is called projectionally complete in X if $P_n x \rightarrow x$ for each x in X (and hence $\|P_n\| \leq C$ for each n and some constant $C > 0$). Let A be a (possibly) nonlinear mapping of X into X and $\{A_n\}$ a sequence of nonlinear mappings defined by $A_n = P_n A P_n$.

In [8] the author considered the problem of giving a constructive proof of the existence and the uniqueness of solutions of the equation

$$(1) \quad Ax = f, \quad f \in X,$$

as either the strong or the weak limits of solutions $x_n \in X_n$ of equations

$$(2) \quad A_n x = P_n f.$$

Definition I. Eq. (1) is said to be strongly (weakly) projectionally-solvable if there exists an integer $N > 0$ such that for each $n \geq N$ and each f in X Eq. (2) has a unique solution $x_n \in X_n$ such that $x_n \rightarrow x$ ($x_n \rightharpoonup x$) and x is the unique solution of Eq. (1).

One of the main results obtained in [8] was the following theorem.

Theorem I. Suppose that there exists an integer $N > 0$ and a continuously strictly increasing function $\alpha(r)$ of $R^+ = \{r | r \geq 0\}$ into R^+ with $\alpha(0) = 0$ and $\lim_{r \rightarrow \infty} \alpha(r) = \infty$ such that A_n is continuous in X_n for each $n \geq N$, $A_n x \rightarrow Ax$ for each x in X and

$$(3) \quad \|A_n x - A_n y\| \geq \alpha(\|x - y\|) \text{ for each } n \geq N \text{ and all } x \text{ and } y \text{ in } X_n.$$

Then Eq. (1) is strongly projectionally-solvable provided A satisfies the following condition (c): If $\{X_m\}$ is any projectionally complete sequence in X and $\{x_m\}$ is an arbitrary sequence in X with $x_m \in X_m$ such that $x_m \rightarrow x$ and $A_m x_m \rightarrow g$, then $Ax = g$.

Theorem I was then used to establish the strong projectional-solvability of various classes of linear and nonlinear equations involving strongly monotone, complex monotone, P-compact and other classes of operators.

In his forthcoming paper [3] Browder notes that for certain classes of nonlinear operators (e. g. J-monotone operators) a direct verification of condition (c), seems sometimes to require additional hypothesis, which thus might in general restrict the class of equations to which Theorem I could be applicable. He then shows that if instead of condition (c) we assume that Eq. (1) possesses a solution, then somewhat weaker assumptions ensure that Eq. (1) is strongly projectionally-solvable. In

his subsequent paper [4] Browder exploits further the relation (*) between solvability and projectional-solvability of a given equation and derives rather general results on the relation between solvability and approximation-solvability, where the approximation operators A_n need no longer be of projectional type.

The purpose of this note is to investigate the relation between our condition (c) and the solvability condition used by Browder in [3, 4]. We show that even for the general (not necessarily projectional) approximations considered in [4], Eq. (1) is uniquely approximation-solvable (and, in particular, strongly projectionally-solvable) if and only if the corresponding condition (C) (to be stated below) holds. Thus we show that at least when X is reflexive the assumption of the corresponding condition (C) is equivalent to the assumption of solvability. Finally we show that a rather general class of operators which often appears in applications satisfies condition (C).

Finally let us comment on the views expressed in [4] concerning the role of numerical functional analysis. Needless to say, the conclusions of abstract functional analysis can (and actually do) have direct relevance to the problems of numerical functional analysis, and, in fact, are being continually exploited. It is, however, true that until recently (mostly due to the development of new computers) few pure mathematicians in the

(*) Let us remark that this relation between solvability of linear or non-linear equations and the convergence of various approximation methods (projectional variational, iterative, finite difference, etc.) has also been used by a number of authors (for example [5, 9, 10, 1] to mention just a few) as well as by the present author [6, 7]; we are certain that few if any would question the practical usefulness of the knowledge that a given equation has a solution.

United States and Western Europe have devoted their full-hearted attention to the constructive and computational aspects in the solution of functional equations and to the systematic study of the numerical methods which are used in applied mathematics (and which, unfortunately, very often have a rather ad hoc nature). We believe that in addition to the development of new methods, part of the role of numerical functional analysis (or whatever name one may give to it) is to provide an applied mathematician with a rigorous, unified and (insofar as possible) self-contained theory of various numerical procedures (in all their aspects) which he can (or actually does) use. It is essential that, if possible, every effort be made to develop the theory in such a way that at least a sophisticated applied mathematician can hope to understand it.

Remark 1. Since the purpose of this note is to make a few relevant observations on the results presented in [3, 4] we do not include here references to various authors who have made considerable contributions to the theory of approximation methods. For this we refer the reader to the papers [4, 8].

The author is indebted to F. E. Browder for having made a draft of [3, 4] available to him.

1. Approximation-solvability. Let X and Y be two separable reflexive Banach spaces and let $\{X_n\} \subset X$ and $\{Y_n\} \subset Y$ be two sequences of finite dimensional subspaces. Let A be a (possibly) nonlinear mapping of X into Y , P_n a mapping of X into X_n and Q_n a mapping of Y into Y_n . Let $\{A_n\}$ be a sequence of mappings of X_n

into Y_n defined by $A_n = Q_n A|_{X_n}$, i.e. A_n is the restriction of $Q_n A$ to X_n .

We consider the problem of constructing solutions of the equation

$$(4) \quad Ax = f, \quad f \in Y,$$

as strong limits of solutions $x_n \in X_n$ of approximate equations

$$(5) \quad A_n(x_n) = Q_n(f).$$

To make our problem more precise we introduce the following two definitions (see [4, 8]).

Definition 1. A quadruple of sequences $(\{X_n\}, \{Y_n\}, \{P_n\}, \{Q_n\})$ is called an approximation scheme for the Eq. (4) if for each n the dimension of X_n equals the dimension of Y_n , the mappings P_n and Q_n are continuous and, for each x in X and y in Y , $P_n x \rightarrow x$ and $Q_n y \rightarrow y$ as $n \rightarrow \infty$.

Definition 2. Eq. (4) is said to be uniquely approximation-solvable with respect to the given approximation scheme if there exists an integer $N > 0$ such that, for each $n \geq N$ and each f in Y , Eq. (5) has a unique solution x_n in X_n such that $x_n \rightarrow x$ in X and x is the unique solution of Eq. (4).

The main result of our note is the following theorem.

Theorem 1. Suppose that there exists an integer $N > 0$ and a continuous strictly increasing function $\alpha(r)$ of $R^+ = \{r | r \geq 0\}$ to R^+

with $\phi(0) = 0$ and $\lim_{r \rightarrow \infty} \phi(r) = \infty$ such that A_n is continuous in X_n for each $n \geq N$, $Q_n A P_n x \rightarrow Ax$ for each x in X as $n \rightarrow \infty$, and

$$(6) \quad \|A_n x - A_n y\| \geq \phi(\|x - y\|) \text{ for each } n \geq N \text{ and all } x \text{ and } y \text{ in } X_n.$$

Then Eq. (4) is uniquely approximation-solvable if and only if the mapping A satisfies the following condition (C): Let $(\{X_m\}, \{Y_m\}, \{P_m\}, \{Q_m\})$ be an arbitrary approximation scheme for Eq. (4). If x_m is any sequence in X with $x_m \in X_m$ such that $x_m \rightarrow x$ and $A_m x_m \rightarrow g$, then $Ax = g$.

Proof. (Uniqueness) Suppose that $u \neq v$ and $Au = Av$. Because, for each u and v in X , $P_n u$ and $P_n v$ lie in X_n , (6) implies that for each $n \geq N$

$$\|Q_n A P_n u - Q_n A P_n v\| \geq \phi(\|P_n u - P_n v\|).$$

Since, by hypothesis, $P_n u \rightarrow u$, $P_n v \rightarrow v$ and $Q_n A P_n x \rightarrow Ax$ the passage to the limit in the above inequality yield the inequality

$$0 = \|Au - Av\| \geq \phi(\|u - v\|),$$

which contradicts the properties of $\phi(r)$. Hence $u = v$ and A is one-to-one.

(Existence) First note that, by (6), for each $n \geq N$ the mapping A_n of X_n into Y_n is one-to-one and continuous and, therefore, by Brouwer

theorem on invariance of domain, A_n is an open mapping and its range $R(A_n)$ is an open set in Y_n . Furthermore, it follows easily (for details see [8, 4]) that $R(A_n)$ is also a closed set in Y_n and, therefore, $R(A_n) = Y_n$. Hence for each $n \geq N$ and each given f in Y there exists a unique element x_n in X_n such that $A_n x_n = Q_n(f)$. For the sequence x_n thus determined, (6) and the fact that $\|Q_n A(0)\| \leq K_0$ for some constant K_0 imply that

$$\|A_n x_n\| \geq \|A_n x_n - A_n(0)\| - \|A_n(0)\| \geq \alpha(\|x_n\|) - K_0$$

whence, in view of (5) and the fact that $\|Q_n(f)\| \leq K_1$ for some constant K_1 ,

$$\alpha(\|x_n\|) \leq \|Q_n f\| + K_0 \leq K_1 + K_0.$$

Let $\beta(r)$ denote the continuous strictly increasing function for R^+ into R^+ with $\beta(0) = 0$ which is the inverse of $\alpha(r)$. Then for each $n \geq N$,

$\|x_n\| \leq \beta(K_1 + K_0)$. Since X is reflexive, there exists a subsequence $\{x_{n_i}\}$ of $\{x_n\}$ which converges weakly to some element x in X .

Furthermore, in view of (5) and the properties of Q_n , $A_{n_i} x_{n_i} = Q_{n_i}(f) \rightarrow f$ as $n_i \rightarrow \infty$; therefore, condition (C) implies that $Ax = f$. Since, as was shown above, x is the unique solution of Eq. (4), we conclude a posteriori that the selection of the subsequence was not necessary.

Hence, $x_n \rightarrow x$.

Moreover, $x_n \rightarrow x$. Indeed, if we apply (6) to the elements x_n and $P_n x$ in X_n for $n \geq N$ then, since x is the solution of Eq. (5), our conditions imply that

$$\begin{aligned} \alpha(\|x_n - P_n x\|) &\leq \|A_n x_n - A_n P_n x\| = \|Q_n(f) - A_n P_n x\| = \|Q_n Ax - Q_n AP_n x\| \\ &\leq \|Q_n Ax - Ax\| + \|Ax - Q_n AP_n x\| \rightarrow 0 \text{ as } n \rightarrow \infty. \end{aligned}$$

Hence the properties of the function $\beta(r)$ imply that

$$\|x_n - P_n x\| \leq \beta(\|Q_n Ax - Ax\| + \|Ax - Q_n AP_n x\|) \rightarrow 0 \text{ as } n \rightarrow \infty.$$

Consequently, $\|x_n - x\| \leq \|x_n - P_n x\| + \|P_n x - x\| \rightarrow 0$ as $n \rightarrow \infty$. Thus x is also the strong limit of x_n and hence Eq. (4) is uniquely approximation-solvable.

Converse. Let $(\{X_m\}, \{Y_m\}, \{P_m\}, \{Q_m\})$ be an arbitrary approximation scheme for Eq. (4) and let $\{x_m\}$ be any sequence in X with x_m in X_m such that $x_m \rightarrow x$ and

$$(7) \quad A_m x_m = g_m \rightarrow g \text{ as } m \rightarrow \infty.$$

To prove that $Ax = g$ note first that, by our conditions on Q_m , as $m \rightarrow \infty$

$$(8) \quad \|g_m - Q_m(g)\| \leq \|g_m - g\| + \|g - Q_m(g)\| \rightarrow 0.$$

Since Eq. (4) is uniquely approximation-solvable, for every $m \geq N$

and every g in Y there exists a unique $y_m \in X_m$ such that

$$(9) \quad A_m y_m = Q_m(g),$$

$y_m \rightarrow y$ and y is the unique solution of the equation $Ay = g$. Subtracting

(9) from (7) we obtain the equation

$$(10) \quad A_m y_m - A_m x_m = Q_m(g) - g_m$$

whence, in virtue of (6), we derive the inequality for all $m \geq N$

$$\alpha(\|y_m - x_m\|) \leq \|A_m y_m - A_m x_m\| = \|Q_m(g) - g_m\|.$$

This, (8) and the properties of $\beta(r)$ imply that

$$(11) \quad \|y_m - x_m\| \leq \beta(\|Q_m(g) - g_m\|) \rightarrow 0 \quad \text{as } n, m \rightarrow \infty.$$

Since $y_m \rightarrow y$, (11) shows that $x_m \rightarrow y$ and $y = x$ because, by hypothesis,

$x_m \rightarrow x$. Thus, $Ax = f$, i.e., condition (C) holds.

Remark 2. Let us note that Definition 1 is slightly different from the definition of an approximation scheme used in [4], where it was stipulated that, for each y in Y , $Q(y) \rightarrow y$ and uniformly on compact subsets of Y . On the other hand, in Theorem 1 we assumed that $Q_n A P_n x \rightarrow Ax$.

for each x in X while Browder assumes the continuity of A and proves the latter relation by using the strengthened properties of Q_n .

Remark 3. Our weaker assumption that the inequality (6) be valid only for all sufficiently large n instead of all $n \geq 1$ has the advantage that, as was shown in [8], Theorem 1 includes all known results on the projection or Galerkin method for linear equations involving bounded operators.

Theorem 2. Suppose that there exists an integer $N > 0$ and a continuous strictly increasing function $\varphi(r)$ of R^+ into R^+ with $\varphi(0) = 0$ and $\lim_{r \rightarrow \infty} \varphi(r) = \infty$ such that A_n is continuous in X_n for each $n \geq N$, $Q_n A P_n x \rightarrow Ax$ for each x in X as $n \rightarrow \infty$, and the inequality (6) holds. Then, under the above conditions, the following equivalent assertions are valid: $(A_1) \iff (A_2) \iff (A_3)$, where

(A_1) A satisfies condition (C)

(A_2) Eq. (4) is uniquely approximation-solvable

(A_3) For any given f in Y , Eq. (4) has a solution x in X .

Proof. In view of Theorem 1 and Definition 2, to prove Theorem 2 it suffices to show that $(A_3) \implies (A_1)$. Let $(\{X_m\}, \{Y_m\}, \{P_m\}, \{Q_m\})$ be an approximation scheme for Eq. (4) and let $\{x_m\}$ be any sequence in X with x_m in X_m such that $x_m \rightarrow x$ and $A_m x_m \rightarrow g$. To prove that $Ax = g$ first note that, by assumption, equation $Ax = g$ has a solution y in X and our conditions on A and (6) imply that y is unique. Furthermore, since $A_m x_m \rightarrow g = Ay$ as $m \rightarrow \infty$, the inequality (6) and our conditions on A imply that for all $m \geq N$ we have

$$\varphi(\|x_m - P_m y\|) \leq \|A_m x_m - A_m P_m y\| \leq \|A_m x_m - g\| + \|Ay - Q_m A P_m y\| \rightarrow 0, \text{ as } m \rightarrow \infty.$$

Hence $\|x_m - P_m y\| \rightarrow 0$ and, consequently, $x_m \rightarrow y$ with $y = x$ since $x_m \xrightarrow{m} x$, as $m \rightarrow \infty$. Thus $Ax = g$ and the proof of Theorem 2 is complete.

Remark 4. Let us observe that if $X_n = Y_n$ for each n and if $P_n = Q_n$ is a linear projection mapping, then Theorem 1 in this paper is a strengthening of Theorem I in [8] in the sense that under the conditions of Theorem I Eq. (1) is strongly projectionally-solvable if and only if the operator A satisfies condition (C).

2. Remarks on condition (C). It should be emphasized that though, as was observed in [3, 4], the direct verification of condition (C) for certain classes of nonlinear operators may be not easy or may require an additional hypothesis nevertheless, as was shown in [8], its validity for linear bounded operators is obvious while its verification for P -compact operators and for bounded and continuous strongly monotone and complex monotone operators in Hilbert space H is trivial. Furthermore, its validity for continuous monotone operators in H follows from Browder's Lemma 3 in [2], and for continuous or demicontinuous and J -monotone mappings A of X into X (where X has property $(\pi)_1$, X^* is strictly convex and there exists a weakly continuous duality mapping J of X into X^*) its validity follows from author's Lemma 3 in [8]. It is also not hard to verify (see [8]) condition (C) for operators $A = T + S$ in H , where T is a continuous and either monotone or complex monotone and where S is strongly continuous.

Finally, it can be shown that if a reflexive Banach space has property $(\pi)_1$ (i. e., there exists a sequence $\{X_n\}$ of finite dimensional subspaces of X and a sequence of linear projections $\{P_n\}$ defined on X such that $P_n X = X_n$, $X_n \subset X_{n+1}$, $\bigcup_n X_n$ is dense in X and $\|P_n\| = 1$) and the mapping A of X into X has the form $A = I - S - T$, where T is completely continuous and S is strictly contractive, then A satisfies condition (C). The validity of the above assertion will be proved elsewhere without the assumption that X is reflexive.

REFERENCES

1. J. P. Aubin, Nonlinear stability-approximation of nonlinear operational equations (to appear).
2. F. E. Browder, Nonlinear elliptic boundary value problems, Bull. Amer. Math. Soc. 69 (1963), 862-874.
3. _____, Nonlinear accretive operators in Banach Spaces (to appear).
4. _____, Approximation-solvability of nonlinear functional equations in normal linear spaces (to appear).
5. D. D. Lax and R. D. Richtmyer, Survey of the stability of linear finite difference equations, Comm. Pure and Appl. Math. 9 (1956), 267-293.
6. W. V. Petryshyn, On two variants of a method for the solution of linear equations with unbounded operators and their applications, J. Math. and Phys. 44 (1945), 297-312.
7. _____, On the extension and the solution of nonlinear operator equations, Illinois J. Math. 19 (1966), 255-274.
8. _____, Projection methods in nonlinear numerical functional analysis (to appear).
9. N. I. Polsky, On the convergence of certain approximate methods in analysis, Ukrainian Math. J. 7 (1955), 56-70.
10. P. E. Sobolevsky, On equations with operators forming an acute angle, Dokl. Akad. Nauk SSSR 116 (1957), 255-271.

A NEW MODE OF COMPUTER-AIDED INSTRUCTION

G. K. Manacher

ABSTRACT

A new type of program applicable to computer-aided instruction is discussed. Its use permits speedy and accurate checking and diagnosis of mathematical relations supplied by students as answers to problems in physics. The method proposed, is, to a limited degree, capable of evaluating the physical content of the relation.

In the literature describing the languages and systems available for writing teaching programs, i. e. programs that provide for the man-machine interaction in computer-aided instruction,* two main lines of development have been apparent.¹ At one pole is the linguistic mode system. Here, the student interacts "verbally" with the computer, in the sense that messages are sent to the computer by him by any one of a variety of techniques (pointing to words on a CRT with light pen, typing in words at a console, etc.), and are then interpreted "verbally" by the machine. By this we mean that the steps the computer takes assume that the words received need not be further decoded: the words themselves make up the target language. Within this mode, there are many things the computer can do, differing considerably in complexity. For

*Abbreviated CAI. The term computer-assisted learning (CAL) is also sometimes employed.

instance, it can ask that the message transmitted by the student correspond exactly to a precoded message. Or it can ask for the presence, unordered, ordered, or partly ordered, of a set or subset of special key words. The computer's response may or may not depend on variables computed as functions of the student's prior responses.

The most obvious use of the linguistic mode is in non-mathematical subjects. It has already been used, with considerable success, in the teaching of languages, notably German. It is probably adaptable to teaching substantial portions of more mathematical subjects such as physics or chemistry. When used for the latter, however, it has certain limitations that must be respected. It cannot accept answers in mathematical form, except in the most trivial cases. Consequently it is-- at best--employed in a discursive way to conduct a dialogue about the quality of things. It can, of course, demand answers in the form of numbers, or can accept responses to multiple choice questions. In the area in numerical manipulation it is not suitable.

At the other pole is the computational mode system. This is much like a "conventional" time-sharing system, but typically includes pre-coded routines of special interest to a student pursuing a particular course. In Ref. 1, which provides a short but thorough survey of the field as well as several illuminating glimpses into the near future, several examples are provided. In one, the student supplies the constants of integration (i. e. the initial conditions in the problem) to a program that assumes a force law for a force field acting on a point mass.

The program computes the motion of the particle and in some fashion exhibits it to the student, either through a table or a visual display.

The student can perceive the connection between the various parameters of the orbit (period, point of nearest approach to the origin) and the inputs (initial position and momentum, as a rule). Several variants of the program may be made available to the student, one for each of a variety of force laws. In another example provided (op. cit.), the student supplies a prepared program with data giving the specification of a system of lenses. The computer ascertains where the images are and computes certain figures of merit for the system.

In general, the student works in a "mixed" environment, in part preparing functions of his own to carry out certain procedures (hopefully learning thereby something about the subject, rather than just learning how to program the machine) and in part depending on author-prepared functions to perform more complex tasks or to simulate some features of the environment.

The computational mode system can keep track of a student's part performance, and can check sequences of numerical values computed by the student against sequences computed by a corresponding author-prepared function. It can participate in a dialogue, indicating to the student the ways in which the student's sequence sequence of values may fail to match the correct one. For instance, if the student were required to find $f(x)$, he could communicate his function by

communicating a sequence of values corresponding to $f(1)$, $f(2)$,

On the basis of a comparison of these values of $f(x)$ with the correct function, the machine can provide hints to the student as to where he may have erred.

In both of these modes there is one thing the student is not allowed to do: to present to the computer a relation he has constructed, with the intention that the relation be judged on the basis of its physical content by the author-prepared function, which will (hopefully) ascertain whether the relation is right, and, if not, offer an intelligent suggestion as to what is wrong.

It is my contention that this omission is a serious defect. In every discipline, the objective sought is the development of one's inductive capacities. It is true that in the early phases of learning, emphasis is placed on the fundamentals, or "building blocks" of the discipline, and the learning at this stage is for the most part noninductive and rote. If development is to proceed beyond this stage, however, the student must acquire familiarity with inductive techniques. In mathematics, the exercise of this formative skill takes the principal form of the construction of a theorem or algorithm, or to a lesser extent, equation or relation. In the humanities, it takes the form of composition in the natural language or some well-defined subset of it. In physics it takes the form of the physical relation.

I think it is fair to say that a student who has not had practice in the composition of such forms has not really participated--has not

really had practice--in the learning of inductive skills. The trouble lies, then, in the danger that the student will build up a superficial question-answering skill that masquerades as inductive ability. What is needed in physics, at least, are programs that will evaluate relations in conformance with their physical content.

It would seem at first glance that such programs are neither feasible nor well-defined. The most obvious way to check a relation would seem to be to embed it in a linguistic-mode system. A relation would be regarded as an algebraic identity, and an attempt would be made to reduce it to some "simplest form" wherein its validity could be checked. But this technique is certain to founder because of the enormous number of alternative algebraic forms available for the same expression, i. e.,

$$\sin \theta = \sqrt{1 - \cos^2 \theta} = \sqrt{1 - (\sqrt{1 - \sin^2 \theta})^2} \dots, \quad a/b = 1/(b/a) = (b/a)^{-1} = ab^{-1} = 1/a^{-1}b = 1/ba^{-1}, \dots \text{ etc.}$$
 In addition, if the relation is discovered to be wrong, not much can be inferred about the error committed.

It would seem that numerical techniques to test relations would face two roadblocks of equally frustrating magnitude, namely, the fact that only a discrete sampling of the relation is available, and the apparent difficulty of conducting a dialogue with the student based on so limited a sampling:

It is my feeling, nevertheless, that this difficulty can be overcome; that such a discrete analysis of a student-supplied relation can be analyzed according to principles that will, in many cases, permit a mechanical decision to transmit to the student a message concerning his (incorrect)

relation that will appear to rival the acuity of a good teacher in indicating to the student the nature of the error committed, in such a fashion as to

allow the student to see where and how improvement is to be obtained.

I believe that the principles in question can, with some effort, be applied

to a substantial number of problems in physics that require answers in

the form of relations. I have found, however, that devising the logical

steps is a problem-by-problem affair, requiring something of the order

of one hour per problem of the program writer's time.

First observe that there is merit in taking the student's relation, given in the form

$$(1) \quad \lambda \leftarrow \text{expression} \quad \text{where } \lambda \text{ is any simple variable,}$$

and instantiating it in the following way: Suppose the student has supplied his answer in terms of variables appearing in the problem statement

only. This is no real restriction, as the only hidden variables are likely

to be simple physical constants. Within the analyzing program, we assign

sequences of numbers to each of the variables in the expression of (1).

Each sequence is different, but all the sequences are of the same length.

Suppose the (common) sequence length is denoted L . We compute L

values of λ , each one consisting of a computation of the expression.

Let the values of λ thus derived be called λ_i , $i = 1, 2, \dots, L$. Then

if the expression is a function of variables $v_1, v_2, \dots, v_n$, say

$E(v_1, v_2, \dots, v_n)$, then for each v_j ($j = 1, 2, \dots, n$) there is assigned a

sequence v_{ji} as explained above. Then define E_i by

$$(2) \quad E_i = E(v_{1i}, v_{2i}, \dots, v_{ni}).$$

Each of the E_i is a value rather than an expression. The vector

$\vec{E} = (E_1, E_2, \dots, E_n)$ will be called the instantiation of E .

The reason for choosing the sequences $v_{ji}, v_{j'i'}$ different for any $j \neq j'$ can be seen from the following example. Suppose the student were given the task of specifying the moment of inertia of a body having two dimensional quantities r and a , and suppose the correct formula were given by

$$(3) \quad I = 1/2(mr^2) + 1/2(ma^2).$$

Suppose that the student had supplied either the relation

$$(4) \quad I = ma^2$$

or the relation

$$(4') \quad I = mr^2.$$

It is clear that the instantiations for the student's

result and the correct result would be the same in the case that the

instantiation sequences for a and r were the same. It is easy to see from this example why they should be different.

The basic strategy in checking the student's relation, then, is to choose the instantiation sequences v_{ji} so that (1) $v_{ji} \neq v_{j'i}$ for $j \neq j'$, (2) the instantiation sequences are long enough and contain numbers chosen from a sufficiently large range as to insure that the instantiations E, E' , belonging respectively to the student's relation and to the correct relation, will agree or disagree accordingly as the relations themselves agree or disagree, except in the rare case of coincidental agreement.

The correct relation is of course part of the checking program, and as such is not accessible to the student. The main burden of a checking program is not the verification that the student's relation is right. After all, there are no hints to be given, remedial strategies to be worked out, etc. in the case of a right answer. The problem is to respond intelligently to a wrong answer, to determine how serious it is, and so forth.

Various properties can be checked within the strategy described. For instance, suppose that the correct answers were separable in v_1 , in the sense that

$$(5) \quad E^1(v_1, v_2 \dots v_n) = R(v_2, v_3 \dots v_n) + R'(v_2, v_3 \dots v_n) * R''(\bar{v}_1),$$

where R, R', R'' are arbitrary functions, and suppose further that the student who gets (5) wrong can nevertheless be expected to provide a relation that, like (5), is separable in v_1 . An appropriate strategy might be the following: Let the instantiation vectors v_j be made of constant components for $j \neq 1$, that is, $v_{ji} = v_{ji'}$ for any $i, i', j \neq 1$. In this case we speak of the v_j as restricted; in the general case we refer to the v_j as unrestricted. Compare the values of $\vec{E}^1$ and $\vec{E}^c$, for the instantiation of the correct answer. Consider any component E_m^1 of $\vec{E}^1$; if it differs from E_m^c by some amount, adjust all the components of $\vec{E}^1$ by adding a constant to all of them so that E_m^1 and E_m^c agree. If $\vec{E}^1$ now equals $\vec{E}^c$, the student can be told that some term not involving v_1 is wrong; that in all other respects the relation is right. If $\vec{E}^1 \neq \vec{E}^c$, examine two pairs of components, say (E_m^1, E_m^c) and (E_ℓ^1, E_ℓ^c) , ($\ell \neq m$). Find a constant factor b and a constant term a such that

$$(6) \quad E_m^c = A + B * E_m^1 \quad \text{and}$$

$$E_\ell^c = a + b * E_\ell^1,$$

and compute $\vec{E}^{1'} = a + b * \vec{E}^1$. If now $\vec{E}^{1'} = \vec{E}^c$, the student may be informed that his functional dependence on v_1 is correct, but it is multiplied by an incorrect factor and (if $a \neq 0$) there is also an incorrect term. If $\vec{E}^{1'} \neq \vec{E}^c$, the student can be asked whether his answer is in the form (5). If he says no, he can be told to put it in that form; if he says yes, he can

be told that his functional dependence on v_1 is wrong (in the sense that R'' is wrong in (5)).

Such techniques as (6) are mathematical in nature and as such shed little light on the underlying principles of the problem. Moreover, they demand that the correct answer be in a special form and are insufficiently general though occasionally applicable. More comprehensive techniques are clearly needed.

The central problem is the logical structure of the analyzing program. If the answer is correct, we are done. If not, we perform some subsidiary test; if that succeeds in identifying the nature of the error we are again done; if that fails, we must make another test, etc. In general, the logical structure of such tests forms a partial order. Frequently, these partially-ordered structures are just simple linear lists.

Let us now consider a strategy as embodied in such a partial order. Clearly, it must meet several criteria in order to be effective:

- (1) It must aim at mistakes of an elementary nature that are easy to detect, easy to point out to the student, and likely to be made. The principles communicated to the student in clarifying the nature of the error should be simpler, if possible, than the problem itself.
- (2) The number of specific tests for mistakes embedded in the strategy should be small, since processor time and memory allocation are likely to be at a premium in most implementations of CAL. Yet the strategy should have the property that, in the event that no single test isolates

the trouble in a relation known to be wrong, some account of the probable nature of the difficulty can still be communicated.

What sorts of principles might underlie such strategies? A principle already suggested and found wanting was the separability of certain variables. Two much more powerful principles that can be applied are those of symmetry and boundary values. To show how these can be applied, we have worked out as an illustration a strategy for analyzing a student's relation, given as a solution to the following problem: Two masses m_1 and m_2 are placed on a frictionless double inclined plane and connected by a massless string passing over a massless pulley, as shown in Figure 1.

Figure 1.

The problem is to determine the acceleration a as a function of m_1 , m_2 , θ_1 , θ_2 , and the acceleration g due to Earth's gravity.

The correct relation is given by the formula.

$$(7) \quad a = g \frac{m_1 \sin \theta_1 - m_2 \sin \theta_2}{m_1 + m_2}$$

as can be verified by considering the tension in the string, using Lagrange's equations of motion, etc. This relation is not separable with respect to m_1, m_2, θ_1 or θ_2 , so our first method is of course inapplicable, except with respect to g .

The two observations that are the cornerstones of the strategy we shall construct are (1) that in the case $m_2 = 0$, $a = g \sin \theta_1$, corresponding to motion on a simple inclined plane, and correspondingly $m_1 = 0$ implies $a = -g \sin \theta_2$; and (2) the answer is anti-symmetric when $m_1 \longleftrightarrow m_2$ and $\theta_1 \longleftrightarrow \theta_2$.

The verification of each of the principles mentioned above is easy within the framework of our methods of instantiation. To check (1), we instantiate the variable m_2 in the student's relation for a with a vector consisting of all zeroes; i. e., we restrict it to zero. The other vectors are unrestricted. An instantiation a of the student's formula for a -- whatever it is -- is compared with an instantiation of the formula $a \leftarrow g \sin \theta_1$ stored in the author-prepared program. If the two instantiations agree in every component, it is virtually certain that regardless of whether student's relation for a is right or wrong, he has recognized that in the case $m_2 = 0$, a must reduce to $g \sin \theta_1$.

To test the anti-symmetry of (2) above, first conduct an unrestricted instantiation of the student's relation. Then exchange the instantiation vectors of m_1 and m_2 and of θ_1 and θ_2 and instantiate his relation again. If every component of one is the negative of the corresponding component of the other, the anti-symmetry of the answer is

assured regardless of whether the answer is right. These two examples should suffice to demonstrate how instantiation can be used to verify various properties of a student's answer.

It is possible to combine checks of this form into a strategy structure satisfying the criteria we have listed. It is shown in Figure 2.

Figure 2.

A bit of explanation may help to understand Fig. 2. An answer of the form

$$(8) \quad a = g \frac{m_1 \sin \theta_1 - m_2 \sin \theta_2}{m_1 + k m_2},$$

where k is a numerical constant, will pass tests 2 and 3 but not 4.

An answer of the form

$$(9) \quad a = g \frac{m_1^k \sin \theta_1 - m_2^k \sin \theta_2}{m_1^k + m_2^k} \quad \text{or} \quad g \frac{m_1^k \sin \theta_1 - m_2^k \sin \theta_2}{(m_1 + m_2)^k}$$

would fail all the tests and get through to the comment in box 5.

A finer "grain" to the problem can be had by replacing box 5 by an additional test that ascertains that when $m_1 \sin \theta_1 = m_2 \sin \theta_2$ the respective forces on the string are equal and opposite in magnitude, wherein $a = 0$. This is done by using unrestricted instantiations for any three of m_1, m_2, θ_1 and θ_2 and fixing the fourth to obey this relation. Upon instantiating a , one should obtain all zeroes. Failure here again can be communicated to the student. In this case, a term of the form

$$(10) \quad a' = g * h(m_1, m_2) * i(m_1 \sin \theta_1, m_2 \sin \theta_2)$$

where $h(0, x) = h(x, 0) = 0$, $h(x, y) = h(y, x)$, and $i(x, y) = -i(y, x)$ may be added to the correct formula (8) and the resulting form $a = a' + a''$ will pass

all the tests. An additional test might then be added to let $m_1 \rightarrow \infty$ while m_2 remains finite and vice-versa. This would still allow answers of the form (1) to get through, when

$$(11) \quad a^k = g m_1 m_2 \frac{m_1 \sin \theta_1 + m_2 \sin \theta_2}{(m_1 + m_2)^k}, \quad k \geq 3.$$

Observe, however, that while no finite series of tests such as these is airtight, it is still true that a well-chosen subset of them, in tandem, is likely to catch and identify student errors in the overwhelming majority of cases. The "right" number is a matter of the tradeoff between the programmer's time and the space reserved on auxiliary storage, etc. on the one hand, and the comprehensiveness desired on the other. I am inclined to think that 3-5 tests will usually be found to be about the right number.

To recapitulate: The basic principle is simply that boundary cases and symmetry properties combine together in two advantageous ways: (1) pedagogically, because these principles are usually correspondent with notions familiar to the student, (2) logically, because they can be linked together into a "sieve" that focusses rapidly on the student's blind spot. Routines written in this style can still be "fooled" into declaring a wrong answer right, but the construction of a wrong answer capable of doing this is likely to require more skill than the construction of the right answer itself.

Finally, it may be noted that considerable creativity is required in recognizing what symmetries, boundary cases, etc. are useful for tests and how they should be ordered. Other determinations, such as maxima and minima, etc. may also be useful.

It is a pleasure to thank Mr. John L. Lawrence of Science Research Associates, Inc., of Chicago, Ill. for his active interest in this work. Mr. William Mixon had some interesting observations, and Prof. Robert M. Krauss contributed an illuminating discussion of the psychological topics touched on in this paper.

REFERENCES

1. The computer in physics education, report on the uses of the computer in undergraduate physics instruction, held Nov. 4-6, 1965 at the University of California at Irvine.
Published by the Commission on College Physics.