DEUCE News No. 39, July, 1959.

1. A NOTE ON THE ZEROS OF POLYNOMIALS, W.P. Gillott, E.E. (N.R.L.)

DEUCE operators may have been puzzled to find that in using RP03/1 to solve a polynomial the programme gets into a loop and refuses to punch any answers. This is not the fault of the programme but of the polynomial. The author of the programme, J.H. Wilkinson of the National Physical Laboratory is publishing a paper (due to appear in English in the July issue of "Numerische Mathematik") which throws considerable light on this subject. Only a few of the many points covered in this paper (which is primarily concerned with eigenvalues of matrices) are mentioned here and reference should be made to the paper itself by all who are particularly interested in this subject.

It appears that the coefficients of a polynomial do not necessarily define its zeros particularly well, For instance let us consider the polynomial $(x+1)(x+2)\ldots(x+20)$ whose zeros are of course $-1, -2, \ldots -20$. We should not regard these as pathologically close. However if we commit an error of 2^{-23} in the coefficient of x^{19} (which in this case is the rounding error of single length standard floating binary) and solve this polynomial the following zeros result:

-1.00000 0000	-10.09526 6145 ± 0.64350 0904 i
-2.00000 0000	-11.79363 3881 ± 1.65232 9728 i
-3.00000 0000	-13.99235 8137 ± 2.51883 0070 i
-4.00000 0000	-16.73073 7466 ± 2.81262 4894 i
-4.99999 9928	-19.50243 9400 ± 1.94033 0347 i
-6.00000 6944	
-6.99969 7234	
-8.00726 7603	
-8.91725 0249	
-20.84690 8101	

and we see that five of the roots are non-trivially complex.

Solving the polynomial with an error of 2^{-55} (in this case the rounding error of double length standard floating binary) in x^{19} produces the following zeros

-1.00000 0000	-10.99999 9999
-2.00000 0000	-12.00000 0006
-3.00000 0000	-12.99999 9983
-4.00000 0000	-14.00000 0037
-5.00000 0000	-14.99999 9941
-6.00000 0000	-16.00000 0067
-7.00000 0000	-16.99999 9947
-8.00000 0000	-18.00000 0028
-9.00000 0000	-18.99999 9991
-10.00000 0000	-20.00000 0001

which are sufficiently accurate for our purposes.

Let us now consider what happens when we solve a polynomial by an iterative method using "nested multiplication" with ten decimal floating arithmetic with a first approximation of n = -20.00012345. In evaluating the first nest we work out $(x + 210)$ for a value $x = -20.000\ 12345$. This gives us

210 - 20.000 12345 which to ten decimal figures is 189.9998766. We would get this same figure if we were to solve a polynomial having 210.00000005 as the coefficient of x^{19} with the disastrous effects outlined above.

Thus we see that to solve a polynomial we must quite often employ much higher accuracy in the intermediate working than we require in the final result. Wilkinson uses triple length accuracy for his root finding programme (RP04/1, Triple Length Bairstow) which is soon to be publsihed. Although we could still invent polynomials sufficiently ill-conditioned to defeat this programme Wilkinson claims that it has worked for all the polynomials he has so far encountered in practice.

2. RECOVERING LOST RESULTS AFTER A PUNCH JAM, S.J.M. Denison, E.E. (N.R.L.)

It may happen that the punch jams in the middle of a programme producing a long string of related results, destroying a number of punched cards, but not upsetting the programme. Attention is drawn to the fact, which may not be universally known, that if the jam is cleared and the programme allowed to finish, the missing values can be found by a finite difference process such as this:

Several of the values on either side of those missing are differenced, and, if the number of missing results is not already known, it can usually be found by dividing the difference between the results on either side of the gap by an approximate first difference. In the example shown, the two values of f between the parallel lines have been lost.

f	δf	$\delta^2 f$	$\delta^3 f$		n	
47454						
	-3213					
44241		256				
	-2957		-25			
41284		231		Mean		
	-2726		-18	-20	0	
38558		213				
	-2513		-17	(-18.6)	1	
36045		196				
	-2317		(-17)	(-17.2)	2	
33728		(179)				
	(-2138)		-20	(-15.8)	3	
(31590)		159				Check
	-1979		- 5	(-14.4)	4	Σ = -72.1
(29611)		154				
	(-1825)		-18	(-13.0)	5	
27786		(136)				
	-1689		(-12)	(-11.7)	6	
26097		124				
	-1565		- 8	(-10.3)	7	
24532		116		Mean		
	-1449		-11	- 9	8	
23083		105				
	-1344		- 8			
21739		97				
	-1247					
20492						

When the differences of some order (in this case, 3rd) are small, take the mean of an odd number of them on each side of the gap, putting each of these values opposite the central member of the group, used, and number these and the differences required to fill the gap between them from 0 to N. The differences for n = 1 to (N - 1) must now be chosen in such a way as to fill the gap in the differences of next lower order.

If we assume that $\delta^3 f = a + bn + cn^2$, then, in the example shown,

$$\left.\begin{array}{l} a = -20 \\ a + 8b + 64c = -9, \\ \text{and } \sum_{n=2}^{6} (a + bn + cn^2) = 124 - 196 = -72 \end{array}\right\} \begin{array}{l} \therefore\ b \doteqdot 1.43 \\ \&\quad c \doteqdot -0.007 \end{array}$$

These values of a, b and c give the intermediate differences shown in brackets, and the arrows show the paths back to the missing f's.

Completing the difference table (from left to right) we get the values shown in the triangle. The values of $\delta^3 f$ obtained are not quite as smmoth as they should be, but we can correct the f just found to smooth them. Let the errors in the f's be $+\epsilon_1$, & $+\epsilon_2$ respectively. Then the pattern of errors in $\delta^4 f$ will be:-

$$+\epsilon_1,\ (\epsilon_2 - 4\epsilon_1),\ (6\epsilon_1 - 4\epsilon_2),\ (6\epsilon_2 - 4\epsilon_1),\ (\epsilon_1 - 4\epsilon_2),\ +\epsilon_2$$

Differencing the $\delta^3 f$'s obtained, we get:-

0, -3, +15, -15, +6, +4

Using the two central values, which give the best estimates of ϵ_1 & ϵ_2 we find that $\epsilon_1 \doteqdot +2$, $\epsilon_2 \doteqdot -1$; therefore the true values of the missing f's are 31588 and 29612, and the middle of the difference table becomes:-

f	δf	$\delta^2 f$	$\delta^3 f$	$\delta^4 f$
			-17	
		196		-2
	-2317		-19	
33728		177		+6
	-2140		-13	
31588		164		-1
	-1976		-14	
29612		150		+1
	-1826		-13	
27786		137		0
	-1689		-13	
		124		+5
			- 8	

which confirms that these f's are in fact correct.

3. <u>PUNCHING RESULTS IN β-FIELD ONLY</u>, D.J. Ozanne, E.E. (N.R.L.)

The 32 column punch subroutines punch results in the α-field when the punch is switched to 64 column. However it is often more convenient to have these in the β-field, leaving the α-field free for punching labels etc. later. A 32 col. punch subroutine can easily be altered to do this as follows:-

Alter all A-29X instructions to { 8-24 1 X
A-29 GO

The 8-241 X instruction can be put in any convenient m.c., i.e. its timing number is immaterial.

Note that this can only be done if the routine still obeys the timing rules of the punch, namely that there must be not more than 38 m.s. between stoppers (37 m.s. for comfort), not more than 116 m.s. between cards, and not more than 20 m.s. between a 9-row and a 9-24 instruction.

4. DIVISION WITH ROUND OFF, J. Boothroyd, E.E. (N.R.L.)

The sequence of instructions

$$1\text{-}24,\quad d,\quad W,\quad T \qquad (W=T)$$

$$27\text{-}22_2 \qquad 31,\quad T_1$$

$$22_2\text{-}A$$

will give a rounded quotient in A (allowance must be made for the shift down in 21_2). No waste instruction is thus required after 1-24.

5. SOME DIVISION OPERATIONS, C.A. Forster, E.E. Co. Luton.

The accompanying table lists the results obtaining in $21_{2,3}$ after 9 division operations in which the dividend and divisor are equal in modulus or either or both are zero. The conditions and results of each operationare given below. It will be noticed that in some cases the result is nearly correct.

The "Leading Digit" referred to is the first digit of the quotient inserted at P_2 position in 21_2 in minor cycle m+1 (when 1-24 starts in m.c. m). This digit is ultimately lost off the end of 21_2 in m.c. m+63 and few people are aware of its existence.

A = any positive number in signed convention.

LD = leading digit (used in some validity tests)

		Contents after Division		
		21_2	L.D.	21_3
1	A/A	0000 ... 001	(1)	-2A
2	A/-A	1111 ... 110	(0)	-2A
3	-A/A	0000 ... 001	(0)	-2A
4	-A/-A	11111 ... 110	(1)	-2A
5	0/A	0000 ... 00	(1)	-2A
6	0/-A	11111 ... 1111	(0)	-2A
7	A/0	$\not\equiv$ A	(1)	0
8	-A/0	$\not\equiv$ -A	(0)	0
9	0/0	11111111 ... 111	(1)	0

Results 1 to 6 could be deduced from the rules given in the Programming Manual viz:

$$2^{31} A = QB + R \text{ with } 0 \leq R < B \text{ for } B > 0$$
$$B \leq R < 0 \text{ for } B < 0$$

Results 7 and 8 may be deduced by anyone knowing how the divider works.

Result 9 is a particular case of Result 7.

6. LINEAR PROGRAMMING, D.J. Flower, E.E., (L.C.S.)

A considerable amount of work has been done at L.C.S. over the past two years on Linear Programming and allied subjects.

About a year ago the General Simplex programme (LX01M) was published and the Transportation problem using the Ford-Fulkerson method (see DEUCE News Omnibus 1 p. 29) is now ready for publication.

Below are summarised some of the further projects tackled in the period.

Dual Simplex Method.

This has been programmed and used successfully on several problems, and will shortly be published. The Dual Simplex method is particularly useful when most of the inequalities are lower-bounded as it reduces the number of artificial variables required. (Theory of Games and Linear Programming - S. Vajda).

ZC14 Programme.

A ZC14-type programme has been written to save the time used with G.I.P. on changing sections. This has been found to save up to 50% on problems of size 20 x 20, but the percentage saving is less on a large problem.

Extract Solution.

There is a programme, to be published shortly that extracts from a simplex tableau the labels, partial costs and right-hand sides and punches this 3 by (m + n - 2) matrix, sorted into label order. This is then in a convenient form for a binary decimal conversion programme such as LK09/2.

Changes in Costs or Right-hand Sides.

Programmes have been written to allow for alterations in costs or right-hand sides without having to go right back to an initial solution. This is very useful if there are several problems to be tackled each differing only slightly from one another.

Inverse Matrix Method.

The Inverse Matrix method is being programmed. This in effect performs several simplex iterations in one step and is very useful if a good guess to the final solution can be made.

Integer Solutions.

Programmes have been written for Gomory's Algorithm and several variations on this. Briefly the method is to add new restraints to those for the non-integer solution and hence gradually converge on to the optimum integer solution. The investigation is still in a very primitive stage.

("Outline of an Algorithm for Integer Solutions to Linear Programmes" Bulletin of the American Mathematical Society Vol. 64, No. 5 - R.E. Gomory).

If any other organisations are interested in this work or have done something similar themselves, the author would be pleased to hear from them.

7. DISTRIBUTION OF PROGRAMMES, R.A. Smith, E.E. (N.R.L.)

DEUCE owners and others who receive DEUCE literature will by now have seen the pamphlet "DEUCE Library Services" which details the new proposals for distribution of literature etc.

Under the scheme all DEUCE Owners will receive two copies of all future programme reports (and one copy of the punched cards). This will apply from and including DEUCE programme number 486. Subroutines will be distributed on the basis of 4 copies per machine from and including number 309.

8. AUXILLIARY D.L.'S ON MARK IIA MACHINES, R.A. Smith, E.E. (N.R.L.).

Some Mark IIA DEUCES are having 7 auxiliary D.L.'s fitted. These will be referred to as D.L.'s 1A, 2A, 3A, ... 7A.

9. G.I.P. 7 AND 64 COLUMN BRICKS, R.A. Smith, E.E., (N.R.L.)

All standard bricks have now been converted to 64 column and G.I.P. 7 (ZC01T/5, No. 475) which reads them has been tested.

Copies of G.I.P.7 and 64 column bricks together with some notes on the differences between G.I.P. 7 and G.I.P. 5 can be had on application to N.R.L., Blackheath Lane.

A full report on G.I.P. 7 will be published at a later date.

10. BINARY INPUT AND OUTPUT TO ALPHACODE. W.P. Gillott, E.N. Hawkins, E.E. (N.R.L.)

The methods of input and output so far provided have the disadvantage that if a large amount of information is to be handled they are slow. To overcome this a function has now been written in the 64 column version of alphacode (ZC16T/2 and ZC17T/2) for input and output of numbers in binary form from X stores but not from N stores or T stores.

We will first consider output. Suppose we have a string of twenty numbers in X20-X39 which we wish to punch in binary. To do this we put the number 1 in X18, and 20 in X19. The instruction

Punch BIN. RESULTS X18

will then punch these twenty numbers.

If instead of punching a string of numbers (a row vector) we wish to punch a block of m rows of n numbers each (an m x n matrix) we would write m in Xc, n in X (c+1) where the block of numbers is in X (c+2) onwards and use the same instruction to punch the numbers. The output on cards will be as follows:

First card,	Y row	m
	X row	n
	0 row	blank.
	1 row	P_{17} (row sum shift of one)
	2 row	P_1
	3 row	batch number times P_{17}

Second card onwards - elements of the matrix in standard floating binary matrix punched with the mantissa in DEUCE field α, the exponent in DEUCE field β. The elements are punched by rows of the matrix each row starting on a separate card.

The input is similar. Thus if we have an m x n matrix in standard floating binary on cards punched by a previous alphacode or DEUCE programme and wish to read it we write:

Xa = BIN DATA.

This instruction will cause m to be read into Xa, n into X (a+1) and the m.n elements into X(a+2) onwards.

A block floated matrix can also be read provided that it is in standard DEUCE form which is as follows

Card 1
- Y row m
- X row n
- 0 row number of binary places.
- 1 row P_{17} (row sum shift of one)
- 2 row blank.

Card 2
Elements of the matrix punched successively in the order Y_α Y_β X_α X_β... 9_α 9_β each row being started on a separate card.

These facilities will be included in revised packs to be circulated shortly. (They will be dated 31.7.59).

11. SUBROUTINE AND PROGRAMME CATEGORIES.

The following categories have been added to the library.

Subroutine Category C - commercial routines.
Programme Category XV - analysis of variance.

12. DEUCE News 37.

Some of the copies of DEUCE News 37 were issued without sheets 6 and 7. Replacement copies can be obtained on application to Mr. E.J. Ellis, C.I.S., E.E. Co. Ltd., Kidsgrove, Stoke-on-Trent, Staffs.

13. DEUCE USERS COLLOQUIUM ON PARTIAL DIFFERENTIAL EQUATIONS.

This will be held on Tuesday, 6th October. If anyone wishes to attend, who is not in an establishment with its own DEUCE would they please inform Dr. V.E. Price, London Computing Service, Marconi House.

18. AMENDMENT TO DEUCE NEWS.

DEUCE News 30, Sheet 4. To paragraph 6.5 add:

"After rewind the rewound tape is no longer selected. (After all other functions a tape remains selected)".