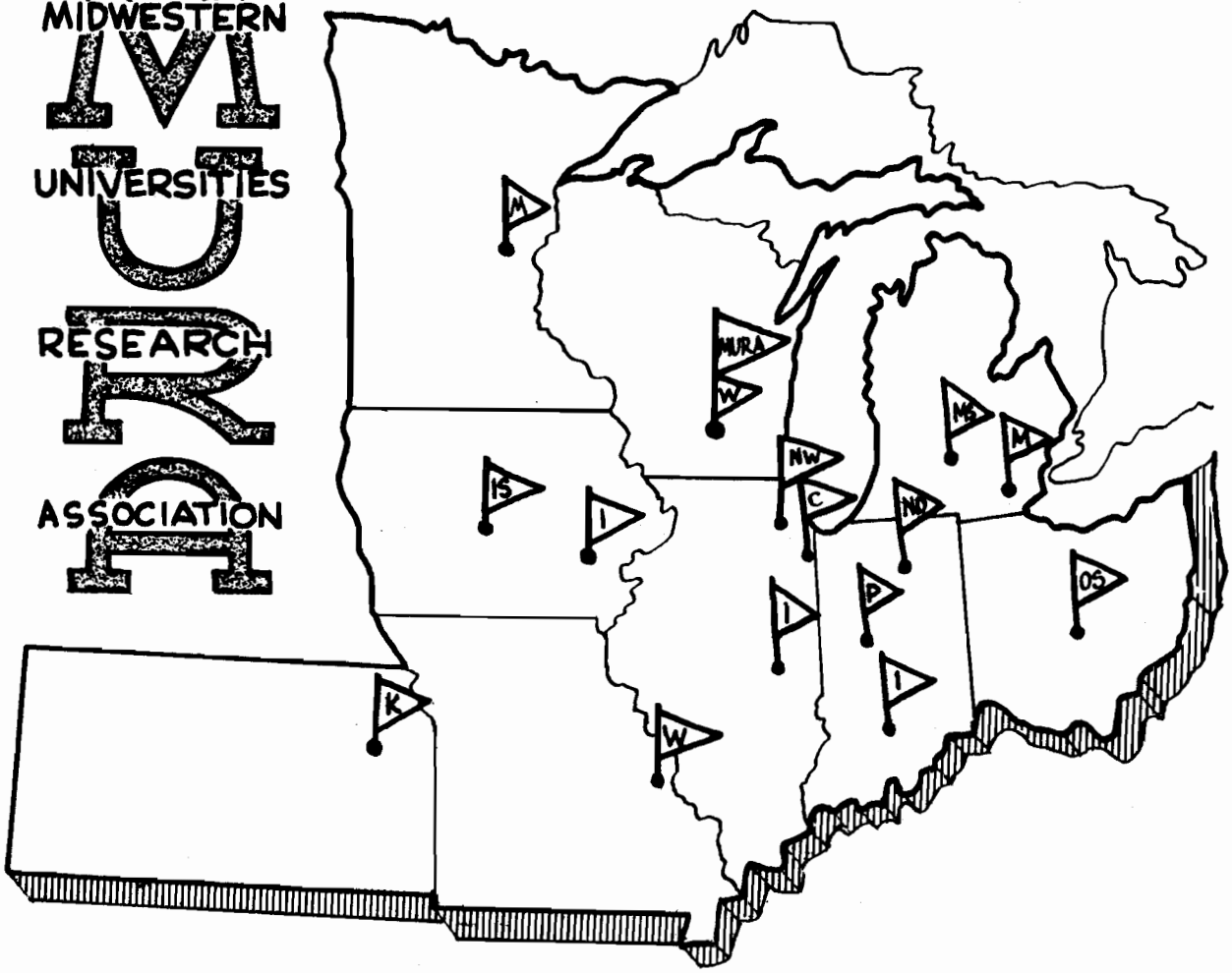
Fermilab Library
0 1160 0038659 3

302

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION



ROUND-OFF ERRORS FROM FIXED-POINT LINEAR ALGEBRAIC TRANSFORMATIONS COMPUTED BY IBM-704 PROGRAM 117

L. Jackson Laslett

REPORT NUMBER 302

MURA-302
Internal

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION*

2203 University Avenue, Madison, Wisconsin

ROUND-OFF ERRORS FROM FIXED-POINT LINEAR ALGEBRAIC TRANSFORMATIONS COMPUTED BY IBM-704 PROGRAM 117

L. Jackson Laslett**

June 11, 1957

ABSTRACT

In order to diagnose the cause of spurious linear change of phase plot dimensions observed in the use of the ALGYTEE program, a study of round-off errors in a simple fixed-point program for linear algebraic transformations has been made. A marked drift of the linear invariant is found. The amount of the drift is correlated with the length of the matrix elements expressed as binary fractions. A theory of the source of round-off bias is presented which agrees well with the observed effects.

* Supported by Contract AEC No. AT(11-1)-384

** On leave from Iowa State College, Ames, Iowa

1. Motivation:

In production use of the ALGYTEE program it was observed that a spurious linear change of the phase-plot dimensions would develop in the course of long runs, a phenomenon which persisted in the mathematically clean case of exclusively linear transformations. To permit a study of round-off errors in a less complicated situation, a fixed-point program for linear algebraic transformations was written by D. Hutchinson and applied to an examination of the aforementioned question.

The program performs the operations

$$x_I = 2\left[(A/2)\,x_{I-1} + (B/2)\,p_{I-1}\right]$$
$$p_I = 2\left[(C/2)\,x_{I-1} + (D/2)\,p_{I-1}\right],$$

with the matrix elements (A, B, C, D) each less than 2 in absolute value and loaded OCTALLY into the computer memory. The program moreover computes, for convenience, the invariant quadratic form $K^2 \equiv (-C)x^2 + (A-1)xp + Bp^2$. The quantities x, p, and K^2 are printed on the output as called for on the agenda.

The fixed-point program has the additional feature that prior to rounding each of the products (A/2) x, etc. a Long Right Shift by N_{SH} bit positions is performed (followed by suitable Long Left Shifts after rounding), to simulate the effect of shorter register lengths.

In preliminary runs with the fixed-point program it soon became evident that a marked drift (increase) of K^2 resulted if some matrix element were a very short binary fraction (eg., .111 000 000...). This observation appeared to lend considerable credence to the contention that an element of

biased rounding is accentuated when multiplicitive factors are of a simple binary form. Basically, the rounding procedure raises the last bit of the number retained when and only when the portion to be discarded is 1/2 or more -- when a simple factor is involved, the appreciable probability for the discarded portion to be exactly 1/2 thus provides an opportunity for an appreciable bias to be present in the computations.

To depict results attributable to the aforementioned effect, a series of runs (#01503800005-48), each of 250 000 iterations, was planned at Dr. Symon's suggestion with the matrix elements A = D having the following values when expressed as binary fractions:

-.111
-.1101
-.11001
-.110001
-.1100001

-.110000001

-.11 000 000 001

-.11 000 000 000 01

-.11 000 000 000 0001

-.11 000 000 000 000 001

-.11 ;

the element C was assigned the value -1 throughout and B taken as 1-AD. It is the purpose of this memorandum to report results obtained to date from this series and to indicate for comparison the drift which would be expected by the mechanism of round-off.

To anticipate, the comparison of the predicted errors and those obtained in the course of the computations appears to support the suggested source of the computational error. It may therefore be hoped that this information will be of some value in suggesting techniques for suppressing the consistent error which grows in direct proportion to the number of iterations executed and is of substantial magnitude when simple binary fractions are employed as factors.

2. The Computational Method

Since it was requested that, for convenience, the program accommodate matrix elements as large as unity, a scaling was introduced to permit the program to work directly with A/2, B/2, C/2, & D/2. Evidently the sequence of operations which is followed is, then:

Form (A/2) x as a double-length number;

Make Long Right Shift by N_{SH} bit-positions;

Round;

Take most significant part & make Long Left Shift by N_{SH}:

Form (B/2) p, as above;

Make Long Right Shift, as above;

Round;

Make Long Left Shift, as above;

Add, to form $[(A/2)\,x]_{Rnd.} + [(B/2)\,p]_{Rnd}$;

Left Shift by one bit-position, to form new x;

and likewise to form new p.

It is noted that, even with $N_{SH} = 0$, the last bit of each of the new quantities (x & p) finally formed by this program will be 0 due to the effect of the scaling feature and that in general the number of significant bits to the right of the binary point in the result will not exceed $34-N_{SH}$.

3. The Expected Linear Drift

We consider here the expected linear drift of $K^2 \equiv (-C)\,x^2 + B\,p^2$, resulting from the average increase of $|x|$ and of $|p|$ from biased rounding of the quantities $(A/2)\,x$ and $(D/2)\,p$. It is noted in the present application that (i) the quantity $C/2$ is $(-1)/2$ and should introduce no complications, and (ii) that $(B/2)\,p$ would introduce considerably less bias if acting alone and probably should be disregarded here since the signs of x and p are completely uncorrelated.

We denote the last non-zero bit of $A = D$ by 2^{-n}, so that the last non-zero bit of $A/2 = D/2$ is $2^{-(n+1)}$. In the products $(A/2)\,x$ and $(D/2)\,p$ non-zero bits will therefore extend no further than through $2^{-(35+n-N_{SH})}$, since (as a result of the scaling) x and p have no more than $34-N_{SH}$ non-zero bits to the right of the binary point. Thus in the second register after a Long Right Shift by N_{SH} there are only n digits which might be either zero or one. The probability of the particular configuration 100... which leads to biased rounding is thus 2^{-n}. The error which this biased rounding introduces into $(A/2)x$ and $(D/2)\,p$ <u>when it occurs</u> is 2^{-36} before the restoring Long Left Shift, $2^{-(36-N_{SH})}$ after the Long Left Shift, and $2^{-(35-N_{SH})}$ in Ax and Dp themselves. Introducing the probability factor 2^{-n}, the <u>average</u> bias (spurious increase of magnitude) in Ax and in Dp is thus found to be

$\Delta \equiv 2^{-(35-N_{SH}+n)}$ and contributes on the average an error of this amount per iteration to x and to p.

The "square invariant", K^2, receives, then, spurious increases from each of these consistent errors, increasing per iteration by an average amount

$$2\left[(-C)\langle|x|\rangle + B\langle|p|\rangle\right]\cdot\Delta$$

$$= 2\left[(-C)+\sqrt{B(-C)}\right]\cdot\frac{2}{\pi}\cdot|x|_{max}\cdot\Delta$$

We tabulate below the expected and observed average rate of increase of K^2 for runs 5-8, 9-12, & 13-16, $|x|_{max}$ being .1 and $N_{SH} = 5$ in all cases:

Runs	A = D, binary	n	$(-C)+\sqrt{B(-C)} = 1+\sqrt{B}_{dec}$	Incr. of K^2 per iter'n. Theor.	Obs.
5-8	-.111	3	1.48412	22.0×10^{-12}	$19.8_4 \times 10^{-12}$
9-12	-.1101	4	1.58296	11.7	9.5_6
13-16	-.11001	5	1.62422	6.0	4.8_2

4. The Cumulative Random Error

The variable plus or minus error, measured with respect to the mean (consistent) error discussed previously, in each of the Right-shifted and rounded quantities (A/2) x, (B/2) p, & (D/2) p will have a R. M. S. value

$$\delta = 2^{-35} \frac{\sqrt{\dfrac{1^2+3^2+5^2+\ldots\ldots+(2^n-1)^2}{2^{n-1}}}}{2^{n+1}}$$

$$= 2^{-35}\sqrt{\frac{1-2^{-(2n-1)}+2^{-2n}}{12}} \cong \frac{2^{-35}}{\sqrt{12}} = \frac{2^{-36}}{\sqrt{3}} . \quad *$$

The corresponding error for the Long Left Shifted and rescaled quantities is

$$\cong \frac{2^{-(35-N_{SH})}}{\sqrt{3}} .$$

The contributions to the error in K_I^2 from the individual errors in Ax_{I-1}, Bp_{I-1}, and Dp_{I-1} are all added in quadrature to obtain a RMS error in K^2 of amount

$$\sqrt{\sum \left\{ [2(-c)x_I]^2 + [2(-c)x_I]^2 + [2Bp_I]^2 \right\}}\ \delta$$

$$= 2\sqrt{3I\langle x^2\rangle}\,\delta = 2\sqrt{(3/2)I}\,|x|_{max} \cong \sqrt{2}\,.\,2^{-(35-N_{SH})}\sqrt{I}\,|x|_{max} ,$$

the quantity I denoting the number of iterations executed. With $x_{max} = .1$ and $N_{SH} = 5$, this suggests

$$\text{RMS Error in } K^2 = 1.3 \times 10^{-10}\sqrt{I} ;$$

a rough preliminary estimate of this same quantity, obtained from the computational results so far available, suggests an average absolute error

- - - - - - - - - - - - -

* This is $1/\sqrt{3}$ times the corresponding consistent error arising from the configuration 100. . . when it occurs. The factor $1/\sqrt{3}$ can be readily interpreted as $\sqrt{\langle x^2\rangle|_0^{1/2}} : \frac{1}{2} = 2\sqrt{2\int_0^{1/2} x^2\,dx} = 1/\sqrt{3}$

$\approx 0.9 \times 10^{-10} \sqrt{I}$, and a later estimate from run No. 41 suggests $1.3 \times 10^{-10} \sqrt{I}$ in agreement with the theoretical estimate.