UNCLASSIFIED

DP - 172

Physics

AEC Research and Development Report

IBM 650 ROUTINE I

POLYNOMIAL FIT BY LEAST SQUARES

by

R.R. Haefner
Theoretical Physics Division

September 1956

E. I. du Pont de Nemours & Co.
Explosives Department - Atomic Energy Division
Technical Division - Savannah River Laboratory

Printed in USA. Price $0.25
Available from the Office of Technical Services
U. S. Department of Commerce
Washington 25, D. C.

DP - 172

PHYSICS

IBM 650 ROUTINE I

Polynomial Fit by Least Squares

by

Richard R. Haefner
Theoretical Physics Division

September 1956

E. I. du Pont de Nemours & Co.
Explosives Department - Atomic Energy Division
Technical Division - Savannah River Laboratory

Printed for
The United States Atomic Energy Commission
Contract AT(07-2)-1

ABSTRACT

A routine was prepared for the IBM 650 computer to obtain the least squares fit of a polynomial $\sum_{i=o}^{N} a_i x^i$ to a set of 100 or fewer experimental points $f_k(x_k)$. The upper limit of N is 15.

External Distribution according to.
TID-4500 (11th Ed.)

TABLE OF CONTENTS

IBM 650 ROUTINE I
Polynomial Fit by Least Squares

INTRODUCTION

It is desirable to have a computer program that will fit a curve to experimental data. A simple expression that will fit many sets of experimental points is a polynomial in the independent variable. The equations necessary to obtain this fit by using a least squares criterion are very simple for the case of a polynomial expansion, and readily lend themselves to automatic computation.

SUMMARY

A program was prepared for the IBM 650 to obtain the coefficients of a polynomial such that

$$\sum_{k=1}^{n} \left\{ f_k(x_k) - \sum_{i=o}^{N} a_i x_k^i \right\}^2 w_k$$

is a minimum. The number n of experimental results f_k, determined as a function of an independent parameter x_k, may be as large as one hundred. The maximum order N of the polynomial is 15. The input data of the problem, f_k and x_k, are entered in the fixed decimal mode. The output parameters a_i are given in the floating decimal mode. The weight factors w_k may be (1) unity, (2) f_k^{-1}, (3) f_k^{-2}, or (4) arbitrary input parameters. The time required for solution for a given N is $[0.0016n(N^2+10N+20) + 0.002(N^3+10N^2)]$ minutes; for example with n = 20 and N = 10, the time is 10 minutes.

DISCUSSION

EQUATIONS

In order to obtain the least squares fit of a polynomial to experimental data $f_k(x_k)$, it is necessary that

$$\epsilon = \sum_{k=1}^{n} w_k \left\{ f_k(x_k) - \sum_{i=o}^{N} a_i x_k^i \right\}^2$$

be a minimum. The w_k are weight factors. This minimum condition is met by requiring that

$$\frac{\partial \epsilon}{\partial a_j} = -2 \sum_{k=1}^{n} w_k \left\{ f_k(x_k) - \sum_{i=o}^{N} a_i x_k^i \right\} x_k^j = 0$$

for all j from 0 to N.

The resulting set of N+1 simultaneous linear equations is:

$$a_o \sum_{k=1}^{n} w_k + a_1 \sum_{k=1}^{n} x_k w_k + a_2 \sum_{k=1}^{n} x_k^2 w_k + \cdots + a_N \sum_{k=1}^{n} x_k^N w_k = \sum_{k=1}^{n} w_k f_k(x_k)$$

$$a_o \sum_{k=1}^{n} x_k w_k + a_1 \sum_{k=1}^{n} x_k^2 w_k + a_2 \sum_{k=1}^{n} x_k^3 w_k + \cdots + a_N \sum_{k=1}^{n} x_k^{N+1} w_k = \sum_{k=1}^{n} w_k x_k f_k(x_k)$$

$$a_o \sum_{k=1}^{n} x_k^2 w_k + a_1 \sum_{k=1}^{n} x_k^3 w_k + a_2 \sum_{k=1}^{n} x_k^4 w_k + \cdots + a_N \sum_{k=1}^{n} x_k^{N+2} w_k = \sum_{k=1}^{n} w_k x_k^2 f_k(x_k)$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$a_o \sum_{k=1}^{n} x_k^N w_k + a_1 \sum_{k=1}^{n} x_k^{N+1} w_k + a_2 \sum_{k=1}^{n} x_k^{N+2} w_k + \cdots + a_N \sum_{k=1}^{n} x_k^{2N} w_k = \sum_{k=1}^{n} w_k x_k^N f_k(x_k)$$

This set of equations may be written in matrix notation as MA = B where M is a square matrix of order N+1, A is a column vector of order N+1, and B is a column vector of order N+1. The matrix elements of M, $M_{r,c}$, are $\sum_{k=1}^{n} w_k x_k^{r+c-2}$ where r = 1, 2, 3, ..., N+1 and c = 1, 2, 3, ..., N+1. The matrix elements of A are the coefficients of the polynomial. The elements of B, b_r, are $\sum_{k=1}^{n} w_k x_k^{r-1} f_k(x_k)$ where r = 1, 2, 3, ..., N+1.

The vector A is obtained by multiplying both sides of the above matrix equation by M^{-1}, the inverse of M; then $A = M^{-1} B$.

After the solution is obtained, the root-mean-square error σ is determined by:

$$\sigma = \left\{ \frac{1}{n\bar{w}} \left[\sum_{k=1}^{n} w_k \left(f_k(x_k) - \sum_{i=o}^{N} a_i x_k^i \right)^2 \right] \right\}^{1/2}$$

where

$$\bar{w} = \frac{1}{n} \sum_{k=1}^{n} w_k$$

MACHINE PROCEDURE

The succession of steps in the solution of the problem is shown in the flowsheet in the Appendix. The major portions of the routine in the order of their use are as follows:

Step 1. The input parameters f_k and x_k are converted from the fixed decimal mode to the floating decimal mode.

Step 2. If $w_k = f_k^{-1}$ (Type 2) or $w_k = f_k^{-2}$ (Type 3) the w_k are computed. If the w_k are arbitrary they are introduced as input parameters (Type 4). Type 1 is the case in which $w_k = 1$. The w_k are stored in 1050 through (1049 + k).

Step 3. $M_{1,1}$ is computed and stored in 0709. $\tilde{w} = M_{1,1}/n$ is also computed and stored.

Step 4. $\sum_{k=1}^{n} w_k f_k x_k^i$, $i = 0, 1, 2, \ldots, 2N-1$ are computed and stored in 1670 through 1670 + i.

$\sum_{k=1}^{n} w_k x_k^{i+1}$, $i = 0, 1, 2, 3, \ldots, 2N-1$ are computed and stored in 1630 + i.

Step 5. The elements $M_{r,c}$ and b_r are transferred from 1630 through 1630 + 2N and 1670 through 1670 + N to the proper storages in preparation for the matrix inversion.

Step 6. The matrix M is inverted and the vector A is obtained. The vector is transferred to 1570 through 1570 + N, and is punched onto IBM cards.

Step 7. σ is computed and compared with a predetermined limit K. If σ is less than or equal to K the problem is completed. If σ is greater than K, N is increased by one, and the routine returns to Step 4 to compute only the sums required by the augmented N. Then Steps 5, 6, and 7 are repeated until $\sigma < K$.

Details of Step 4

The set of storages 1700 through 1699 + n originally contained w_k. These locations are used as temporary storages for $w_k x_k^i$ while $\sum_{k=1}^{n} w_k x_k^{i+1}$ and $\sum_{k=1}^{n} w_k f_k x_k^i$ are being computed for $i = 0, 1, 2, \ldots, 2N-1$. For a given N, $\sum_{k=1}^{n} w_k f_k x_k^i$ need only be computed for values of i from 0 to N. Since the program increases N by 1 if $\sigma > K$ at Step 7, it would be impossible to return to Step 4 and compute $\sum_{k=1}^{n} w_k x_k^{N+1}$ at that time because the temporary storages now contain $w_k x_k^{2N-1}$. Thus the sums $\sum_{k=1}^{n} w_k f_k x_k^i$ are computed for $i = N+1, N+2, \ldots, 2N-1$ although they are never used.

Details of Step 5

The matrix element $M_{r,c}$ is the same for all indices r,c for which the quantity r+c-2 is the same. This element, equal to $\sum_{k=1}^{n} w_k x_k^{r+c-2}$, is stored in 1627 + r+c, except r = c = 1. The matrix elements and vector B are transferred for inversion to locations 0709 through 0708 + (N+1)(N+2) in the sequence $M_{1,1}$, $M_{1,2}$, ... $M_{1, N+1}$, B_1, $M_{2,1}$, $M_{2,2}$, ..., $M_{2, N+1}$, B_2, $M_{3,1}$, ..., $M_{N+1, N+1}$, B_{N+1}.

The routine stores the contents of 1630 ($M_{1,2}$ and $M_{2,1}$) in locations L = 0710 and L = 0711 + N. Then, the contents of 1631 ($M_{1,3}$, $M_{2,2}$, and $M_{3,1}$) is stored in locations L = 0711, L = 0712 + N, and L = 0713 + 2N. This procedure continues until r+c-2 = N. At this point, the contents of 1629 + N ($M_{1,N+1}$, $M_{2,N}$, ..., $M_{N,2}$, $M_{N+1,1}$) is stored in 0708 + (N+1) 0708 + 2(N+1), ..., 0708 + $(N+1)^2$. The elements of the longest diagonal are now in their proper location. The parameters U and V are used in the coding sheets to explain this procedure. The number of elements on a diagonal, U, is equal to r+c-1. The number of elements not yet transferred on a diagonal U is V.

After the transfers for the longest diagonal have been made, the contents of 1630 + N, ($M_{2,N+1}$, $M_{3,N}$, ..., $M_{N,3}$, $M_{N+1,2}$), is stored in 0709 + 2(N+1), 0709 + 3(N+1), ..., 0709 + $(N+1)^2$. The elements of the diagonals below the longest diagonal are transferred in this manner until finally the contents of 1629 + 2N, ($M_{N+1,N+1}$), is stored in location L = 0709 + N(N+3). The elements of the vector B, B_r, are then transferred from 1669 + r to L = 0708 + r(N+2).

INPUT DATA

The input data $f_k(x_k)$ are entered into storages 1900 through 1899 + k in the form .yyyyyyyyyy x 10^F. The exponent F must be the same for all f_k. The exponent, a two-digit number, is entered into storage 1599 in the form 0000FF0000. The number of points to be used, n_f, is read into storage 1597 in the form 00 0zzz 0000. $n_f \leq 100$.

The values of the independent parameters x_k are entered into storages 1800 through 1799 + k in the form .yyyyyyyyyy x 10^X. The exponent X must be the same for all x_k. This exponent, also a two-digit number, is entered into storage 1598 in the form 0000xx0000.

The order of the polynomial N is entered into 1596 as N+1 in the form 00 00(N+1) 0000.

The parameter K is entered into 1594 in the floating decimal mode.

The problem type, 1, 2, 3, or 4, is entered into 1595 in the form 00 0000 000t. If Type 4 is used, the w_k must be input parameters. They are entered into storages 1700 through 1699 + k in the floating decimal mode.

All input data are read into the machine on load cards interpreted by the L-1 sub-routine of IBM[1].

SUB-ROUTINES

The floating decimal arithmetic is performed by a sub-routine of G. R. Trimble and E. C. Kubie[2]. The matrix inversion sub-routine was devised by R. W. de Sio of IBM. This latter routine has not been published, so the program is given in the Appendix. The essential features of this routine are as follows:

1. The routine must be used with the Trimble sub-routine for floating decimal arithmetic.
2. For the routine of this report, the number (N+1) 0000 (N+2) must be entered into 0708. For example, for N = 10, the number entered into 0708 would be 11 0000 0012.
3. The matrix M and the vector B are entered into 0709 through 0708 + (N+1)(N+2) in the order previously given.
4. The entry to the sub-routine is 0500.
5. The exit from the sub-routine is the instruction address of 0628.

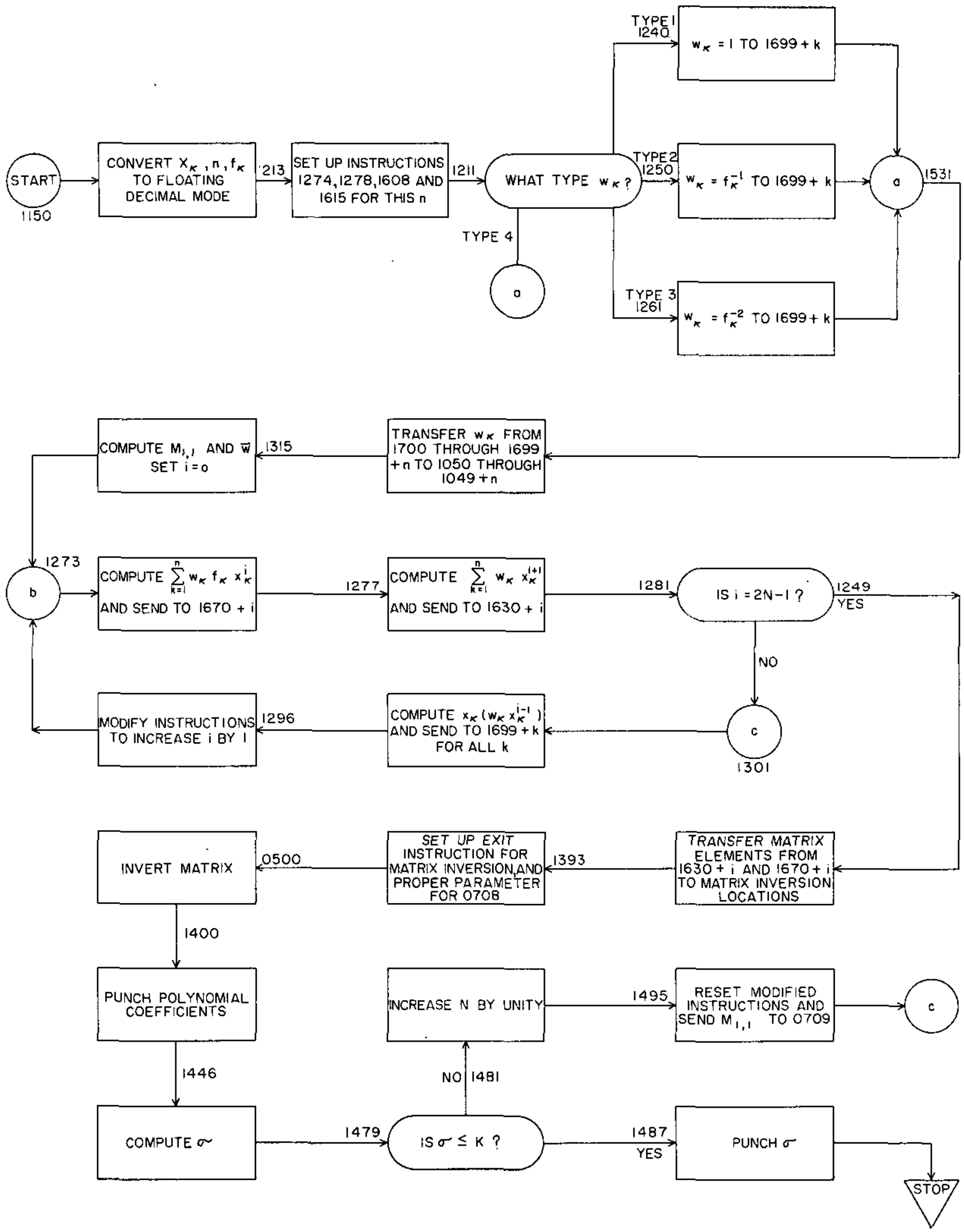
The input information is entered by the L-1 sub-routine,[1] which was relocated from 1977-1992 to 0677-0692. This relocation was necessary to make available the 1950 band. Storage 0688 contains 70 0694 1150. This instruction sends control to 1150 when a nonload card is encountered. The nonload card is placed immediately after the load cards containing instructions and input data.

R. R. Haefner
R. R. Haefner
Theoretical Physics Division

BIBLIOGRAPHY

1. Trimble, G. R., Jr. and Kubie, E. C. "IBM Type 650 Loading Routines." IBM Technical Newsletter No. 8, p. 47, September 1954.

2. Trimble, G. R., Jr. and Kubie, E. C. "An Interpretative Floating Decimal System for the IBM Type 650." IBM Technical Newsletter No. 8, p. 17, September 1954.



FLOWSHEET FOR POLYNOMIAL ROUTINE

PROBLEM: ROUTINE PAGE 1

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1150	60	1157	1182	
1182	10	1598	1154	
1154	21	1176	1184	
1184	69	1189	1192	Set up constants and exit
1192	24	1159	1185	
1185	69	1190	1193	instruction to convert x_k to
1193	24	1166	1215	
1215	69	1218	1221	floating decimal number
1221	24	1153	1186	
1186	69	1191	1194	
1194	24	1158	1159	
Conversion Sub-Routine				
1159	(60	(1800)	1161)	Set up
1161	36	0000	1162	
1162	47	1167	1167	
1167	21	1172	1175	
1175	46	1163	1164	
1163	61	8002	1173	
1173	11	1176	1171	
1171	15	1172	1170	
1170	31	0002	1166	
1166	(20	(1800)	1165)	Set up
1164	61	8002	1174	
1174	10	1176	1171	
1165	60	1159	1169	
1169	11	1158	1181	
1181	11	1597	1153	
1153	44	1177	(1310)	Variable exit instruction
1177	60	1159	1178	
1178	10	1183	1187	
1187	21	1159	1179	
1179	60	1166	1180	
1180	10	1183	1188	
1188	21	1166	1159	
1310	65	1597	1311	
1311	35	0005	1312	
1312	44	1490	1491	Convert n
1490	30	0001	1492	
1491	10	1313	1493	to floating decimal mode
1492	10	1314	1493	
1493	31	0002	1494	
1494	20	1439	1195	

PROBLEM: ROUTINE PAGE 2

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1195	60	1157	1196	
1196	10	1599	1155	
1155	21	1176	1197	
1197	69	1151	1156	Set up constants and exit
1156	24	1159	1198	
1198	69	1152	1205	instruction to convert f_k to
1205	24	1166	1199	
1199	69	1202	1206	floating decimal number
1206	24	1153	1207	
1207	69	1210	1194	
1194	24	1158	1159	Enter conversion sub-routine
1213	61	1597	1214	
1214	21	1274	1228	00 00 n 0000 - to 1274
1228	24	1278	1309	and to 1278
1309	60	1597	1212	
1212	30	0004	1224	
1224	10	1597	1219	
1219	10	1618	1223	
1223	21	1608	1220	07 00n 00n to 1608
1220	60	1597	1226	
1226	10	1602	1229	
1229	21	1615	1211	05 00n 0000 to 1615
1211	60	1595	1204	What type weight factor?
1204	10	1208	8003	
8003	00	0000	1229+t	
1230	69	1619	1160	Type 1 Start fl. dec. comp. at 1240
1231	69	1620	1160	Type 2 Start fl. dec. comp. at 1250
1232	69	1621	1160	Type 3 Start fl. dec. comp. at 1261
1233	69	1612	1160	Type 4 Start fl. dec. comp. at 1531
1160	24	0029	0026	
1240	05	1600	1601 -	
1241	00	(1700)	0000 -	
1242	01	1241	1597 -	Put $w_k = 1$ in floating decimal
1243	02	1603	0000 -	
1244	13	0057	1246 -	into 1700 through 1699 + n
1245	12	1531	0000 -	
1246	06	1241	1183 -	
1247	00	1241	0000 -	
1248	12	1240	0000 -	

PROBLEM: ROUTINE PAGE 3

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1250	09	1600	(1900) -	
1251	00	(1700)	0000 -	
1252	01	1251	1597 -	Compute $w_k = f_k^{-1}$ and send
1253	02	1603	0000 -	
1254	13	0057	1256 -	
1255	12	1531	0000 -	to 1700 through 1699 $\neq$ n
1256	06	1251	1183 -	
1257	00	1251	0000 -	
1258	06	1250	1604 -	
1259	00	1250	0000 -	
1260	12	1250	0000 -	
1261	08	(1900)	(1900) -	
1262	09	1600	0057 -	
1263	00	(1700)	0000 -	Compute $w_k = f_k^{-2}$ and send
1264	01	1263	1597 -	
1265	02	1603	0000 -	
1266	13	0057	1268 -	to 1700 through 1699 $\neq$ n
1267	12	1531	0000 -	
1268	06	1263	1183 -	
1269	00	1263	0000 -	
1270	06	1261	1605 -	
1271	00	1261	0000 -	
1272	12	1261	0000 -	
(a) 1531	69	(1700)	1533	
1533	24	(1050)	1534	
1534	60	1531	1535	
1535	11	1539	1536	Transfer w_k from 1700 through 1699 $\neq$ n
1536	11	1597	1537	
1537	44	1540	1545	
1540	60	1531	1541	to 1050 through 1049 $\neq$ n
1541	10	1183	1542	
1542	21	1531	1543	
1543	60	1533	1544	
1544	10	1183	1538	
1538	21	1533	1531	
1545	69	1601	1546	
1546	24	1222	0026	Zero to 1222
1532	00	1315	0000 -	

PROBLEM: ROUTINE PAGE 4

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1315	01	1050	1051` -	
1316	05	(1052)	0057 -	
1317	00	1222	0000 -	
1318	01	1316	1615 -	
1319	02	1616	0000 -	Compute $M_{1,1} = \sum_{k=1}^{n} w_k$
1320	13	0057	1322 -	
1321	12	1326	0000 -	
1322	06	1316	1617 -	
1323	00	1316	0000 -	
1324	01	1601	1222 -	
1325	12	1316	0000 -	
1326	05	1601	1222 -	
1327	00	0709	0000 -	$M_{1,1}$ to 0709
1328	12	1556	0000 -	
1556	09	1222	1439 -	-
1557	00	1438	0000 -	w to 1438
1558	12	1273	0000 -	
(b) 1273	18	1700	1900 -	
1274	00	00 n	0000 -	Set up
1275	05	0057	1601 -	
1276	00	(1670)	0000 -	$\sum_k w_k f_k x_k^i$ to 1670 + i
1277	18	1700	1800 -	
1278	00	00 n	0000 -	Set up
1279	05	0057	1601 -	
1280	00	(1630)	0000 -	$\sum_k w_k x_k^{i+1}$ to 1630 + i
1281	01	1606	1596 -	
1282	12	1306	0000 -	
1306	01	0057	1596 -	
1307	02	1276	0000 -	
1308	12	1283	0000 -	
1283	13	0057	1301 -	Is i = 2n-1? No to 1301
1284	12	1249	0000 -	
(c) 1301	06	1601	1610 -	
1302	00	1285	0000 -	
1303	06	1601	1611 -	
1304	00	1286	0000 -	
1305	12	1285	0000 -	
1285	07	(1800)	(1700) -	$(w_k x_k^{i+1}) x_k$ to 1699 + k
1286	00	(1700)	0000 -	
1287	01	1285	1607 -	
1288	02	1608	0000 -	
1289	13	0057	1291 -	
1290	12	1296	0000 -	
1291	06	1285	1609 -	
1292	00	1285	0000 -	
1293	06	1286	1183 -	
1294	00	1286	0000 -	
1295	12	1285	0000 -	

PROBLEM: ROUTINE PAGE 5

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1296	06	1276	1183 -	
1297	00	1276	0000 -	
1298	06	1280	1183 -	
1299	00	1280	0000 -	
1300	12	1273	0000 -	
Transfer Mij to Inversion Storages				
(d) 1249	60	1596	1168	
1168	10	1183	1237	
1237	21	1222	1234	00 (N+2) 0000 to 1222
1234	19	1596	1235	
1235	30	0004	1236	
1236	15	1622	1238	24 [0707 + (N+1) (N+2)] 1342 in LA
1238	20	1362	1239	Send to 1362
1239	60	1596	1329	
1329	11	1183	1330	
1330	19	1222	1331	
1331	30	0004	1332	
1332	15	1623	1333	24 [0709 + N (N+2)] 1342 in LA
1333	20	1363	1334	Send to 1363
1334	60	1596	1335	
1335	10	1624	1336	24 [0710 + N] 1387 in UA
1336	21	1386	1484	Send to 1386
1484	69	1488	1485	
1485	24	1343	1340	U = 2 to 1343
(f) 1340	69	1343	1346	
1346	24	1349	1341	V = U to 1349
(g) 1341	69	(1630)	1344	
1344	24	(0710)	1342	L = 0710 for transfer of $M_{1,2}$
1342	60	1349	1353	
1353	11	1183	1348	
1348	44	1350	1357	Is V = 1? Yes to 1357
1350	21	1349	1356	V becomes V-1
1356	60	1344	1351	
1351	10	1596	1345	
1345	21	1344	1341	L becomes L+N+1. Control to (g)
1357	60	1344	1359	
1359	11	1362	1367	
1367	44	1347	1385	Has $M_{N+1,N+1}$ been transfered? Always no.
1347	10	8001	1360	
1360	11	1363	1368	
1368	44	1372	1373	Has $M_{N+1,1}$ been transfered? Yes to 1373
1372	61	1343	1361	
1361	10	1183	1364	
1364	19	1596	1365	
1365	30	0004	1375	
1375	15	1183	1366	

PROBLEM: ROUTINE PAGE 6

	LOCATION	OP.	DATA	INSTRUCTION	REMARKS
	1366	15	1344	1369	
	1369	20	1344	1370	L becomes L ≠ 1 ≠(N≠1)(1-U)
	1370	60	1341	1371	
	1371	10	1183	1374	
	1374	21	1341	1376	
	1376	60	1343	1377	
	1377	10	1183	1378	
	1378	21	1343	1340	U becomes U≠1. Control to (i)
	After $M_{N≠1,1}$ is transferred:				
	1373	69	1380	1379	
	1379	24	1367	1381	
	1381	69	1626	1337	
	1337	24	1374	1338	
	1338	69	1627	1339	
	1339	24	1377	1382	
	1382	69	1628	1383	
	1383	24	1378	1376	
(j)	1376	60	1343	1377	
	1377	11	1183	1378	
	1378	21	1343	1372	U becomes U-1
	1372	61	1343	1361	
	1361	10	1183	1364	
	1364	19	1596	1365	
	1365	30	0004	1375	
	1375	15	1183	1366	
	1366	15	1344	1369	
	1369	20	1344	1370	L becomes L ≠ 1 ≠(N≠1)(1-U)
	1370	60	1341	1371	
	1371	10	1183	1374	
	1374	21	1341	1340	
	1340	69	1343	1346	
	1346	24	1349	1341	V = U to 1349
(k)	1341	69	(1630≠N)	1344	
	1344	24	(0711≠2N)	1342	L = 0711 ≠ 2N for $M_{2,N≠1}$
	1342	60	1349	1353	
	1353	11	1183	1348	
	1348	44	1350	1357	Is V = 1? Yes to 1357
	1350	21	1349	1356	
	1356	60	1344	1351	
	1351	10	1596	1345	
	1345	21	1344	1341	L becomes L ≠ N≠1. Control to (k)
	1357	60	1344	1359	
	1359	11	1362	1367	
	1367	44	1376	1385	Has $M_{N≠1, N≠1}$ been transferred? Yes to 1385. No to (j)

PROBLEM: ROUTINE PAGE 7

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
(m) 1385	69	(1670)	1386	
1386	24	(0710+N)	1387	
1387	60	1385	1384	
1384	11	1629	1388	Transfer b_r from 1670 through 1670 + N
1388	11	1596	1389	
1389	44	1392	1393	to proper matrix inversion storages.
1392	60	1386	1391	
1391	10	1596	1390	
1390	10	1183	1394	
1394	21	1386	1395	
1395	60	1385	1396	
1396	10	1183	1397	
1397	21	1385	1385	
(e) 1393	69	1625	1354	
1354	24	0029	1355	65 0601 0439 to 0029
1355	69	1352	1358	
1358	24	0628	1209	Exit instruction for Mat. Inv. to 0628
1209	60	1596	1201	
1201	10	1183	1200	
1200	30	0004	1216	
1216	21	1222	1225	
1225	60	1596	1203	
1203	35	0004	1217	
1217	10	1222	1227	
1227	21	0708	0500	(N+1) 0000 (N+2) to 0708
0500	Matrix Inversion Routine			
1400	69	(0709)	1398	
1398	24	(1570)	1399	
1399	24	1428	1401	
1401	71	1427	1402	Punch polynomial coefficients,
1402	60	1398	1403	
1403	11	1406	1411	one per card
1411	11	1596	1404	
1404	44	1407	1420	
1407	60	1398	1405	
1405	10	1183	1409	
1409	21	1398	1410	
1410	60	1400	1412	
1412	10	1183	1413	
1413	10	1596	1414	
1414	21	1400	1415	
1415	60	1427	1416	
1416	10	1183	1408	
1408	21	1427	1400	

PROBLEM: ROUTINE PAGE 8

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1420	69	1430	1433	
1433	24	0709	1421	Zero to 0709
1421	60	1424	1431	
1431	10	1596	1422	
1422	21	1425	1432	05 (1570 + N) 0057 to 1425
1432	60	1435	1440	
1440	10	1597	1423	
1423	21	1419	1434	03 00n 0000 to 1419
1434	69	1437	1436	
1436	24	0029	0026	65 1445 0439 to 0029
1446	05	1601	1601-	
1447	00	1222	0000-	
1448	06	1601	1425-	
1449	00	1451	0000-	
1450	08	(1800)	1222-	
1451	05	1570+N	0057-	
1452	00	1222	0000-	Compute:
1453	01	1451	1417-	
1454	13	0057	1456-	
1455	12	1459	0000-	$\sigma = \left[\frac{1}{nw}\sum_k w_k\left(f_k - \sum_{i=0}^{N} a_i x_k^i\right)^2\right]^{\frac{1}{2}}$
1456	05	1451	1617-	
1457	00	1451	0000-	
1458	12	1450	0000-	
1459	03	(1900)	1222-	
1460	08	0057	0057-	
1461	08	(1050)	0057-	
1462	05	0057	0709-	
1463	00	0709	0000-	
1464	01	1459	1419-	
1465	02	1418	0000-	
1466	13	0057	1468-	
1467	12	1475	0000-	
1468	06	1450	1429-	
1469	00	1450	0000-	
1470	06	1459	1426-	
1471	00	1459	0000-	
1472	06	1461	1429-	
1473	00	1461	0000-	
1474	12	1446	0000-	
1475	10	0709	1438-	
1476	10	0057	1439-	
1477	16	0057	0709-	σ to 0709
1478	03	0709	1594-	σ to 0057
1479	11	0057	1487-	
1480	02	1594	0000-	

PROBLEM: ROUTINE PAGE 9

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1481	60	1596	1565	
1565	10	1183	1560	Increase N by unity
1560	21	1596	0026	
1482	12	1495	0000 -	
1495	61	1486	1530	
1530	21	1316	1497	
1497	69	1489	1498	
1498	24	1341	1499	
1499	69	1443	1500	
1500	24	1344	1501	Reset all variable instructions
1501	69	1445	1502	
1502	24	1374	1503	
1503	69	1506	1509	
1509	24	1377	1504	
1504	69	1507	1505	
1505	24	1374	1508	
1508	69	1511	1514	
1514	24	1367	1510	
1510	69	1513	1516	
1516	24	1385	1512	
1512	69	1515	1518	
1518	24	1398	1517	
1517	69	1520	1523	
1523	24	1400	1519	
1519	69	1522	1525	
1525	24	1427	1521	
1521	61	1524	1527	
1527	21	1450	1441	
1441	61	1444	1526	
1526	21	1459	1442	
1442	61	1528	1529	
1529	21	1461	0026	
1496	12	1047	0000 -	
1047	07	1438	1439 -	
1048	00	0709	0000 -	$M_{1,1}$ to 0709
1049	12	1301	0000 -	Return to (c)
1487	69	1561	1564	
1564	24	1427	1562	
1562	69	0709	1563	Punch σ
1563	24	1428	1566	
1566	71	1427	7000	

PROBLEM: CONSTANTS

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1151	60	1900	1161	
1152	20	1900	1165	
1157	00	0000	0049	
1183	00	0001	0000	
1189	60	1800	1161	
1190	20	1800	1165	
1191	60	1799	1161	
1202	44	1177	1213	
1208	00	0000	1229	
1210	60	1899	1161	
1218	44	1177	1310	
1313	00	0000	0050	
1314	00	0000	0051	
1352	45	0601	1400	
1380	44	1376	1385	
1406	24	1569	1399	
1417	05	1570	0057	
1418	03	1899	1222	
1419	03	00 n	0000	This constant is set up
1424	05	1569	0057	
1426	03	0001	0000	
1429	08	0001	0000	
1430	20	0000	0000	
1435	03	0001	0000	
1437	65	1445	0439	
1443	24	0710	1342	
1444	03	1900	1222	
1445	21	1341	1376	
1486	05	1052	0057	
1488	00	0002	0000	
1489	69	1630	1344	
1506	10	1183	1378	
1507	21	1343	1340	
1511	44	1347	1385	
1513	69	1670	1386	
1515	24	1570	1399	
1520	69	0709	1398	
1522	00	1570	0001	
1524	08	1800	1222	
1528	08	1050	0057	
1539	69	1699	1533	
1561	00	6666	0000	
1600	50	1000	0000	
1601	00	0000	0000	
1602	05	0000	0000	
1603	00	1699	0000	
1604	09	0000	0001	
1605	08	0001	0001	
1606	00	1667	0000	
1607	07	1799	1699	

PROBLEM: CONSTANTS

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
1608	07	00n	00 n	This constant is set up
1609	07	0001	0001	
1610	07	1800	1700	
1611	00	1700	0000	
1612	65	1530	0439	
1615	05	00 n	0000	This constant is set up
1616	05	1049	0057	
1617	05	0001	0000	
1618	07	0000	0000	
1619	65	1239	0439	
1620	65	1249	0439	
1621	65	1260	0439	
1622	24	0707	1342	
1623	24	0709	1342	
1624	24	0709	1387	
1625	65	0601	0439	
1626	21	1341	1340	
1627	11	1183	1378	
1628	21	1343	1372	
1629	69	1669	1386	

PROBLEM: de Sio Matrix Inversion

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
0500	60	0708	0501	
0501	30	0004	0502	
0502	21	0708	0503	
0503	35	0008	0504	
0504	21	0699	0505	
0505	19	0708	0506	
0506	30	0004	0507	
0507	15	0707	0508	
0508	20	0703	0509	
0509	15	0699	0510	
0510	16	0706	0511	
0511	69	0607	0512	
0512	22	0607	0513	
0513	69	0616	0514	
0514	22	0616	0515	
0515	69	0708	0516	
0516	24	0702	0517	
0517	65	0699	0518	
0518	16	0706	0519	
0519	20	0705	0520	
0520	65	0708	0521	
0521	16	0706	0522	
0522	20	0704	0523	
0523	65	0703	0524	
0524	69	0603	0525	
0525	22	0603	0526	
0526	65	0707	0527	
0527	15	0706	0528	
0528	69	0602	0529	
0529	22	0602	0026	
0530	11	8001	0531	
0531	15	0602	0532	
0532	10	0603	0533	
0533	20	0602	0534	
0534	21	0603	0535	
0535	65	0705	0536	
0536	16	0706	0537	
0537	20	0705	0026	
0538	69	0612	0539	
0539	22	0612	0540	
0540	15	0699	0541	
0541	69	0609	0542	
0542	22	0609	0543	
0543	15	0706	0544	
0544	69	0611	0545	
0545	22	0611	0546	
0546	65	0703	0547	
0547	30	0004	0548	
0548	69	0609	0549	
0549	23	0609	0550	

PROBLEM: de Sio Matrix Inversion

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
0550	65	0699	0551	
0551	16	0706	0552	
0552	20	0705	0026	
0553	11	8001	0554	
0554	15	0612	0555	
0555	10	0611	0556	
0556	20	0612	0557	
0557	21	0611	0558	
0558	66	0706	0559	
0559	30	0004	0560	
0560	15	0609	0561	
0561	20	0609	0562	
0562	65	0705	0563	
0563	16	0706	0564	
0564	20	0705	0026	
0565	69	0571	0566	
0566	22	0571	0567	
0567	65	0609	0568	
0568	30	0004	0569	
0569	69	0616	0570	
0570	23	0616	0026	
0571	20	0000	0572	
0572	65	0704	0573	
0573	16	0706	0574	
0574	20	0704	0575	
0575	45	0576	0026	
0576	66	0706	0577	
0577	11	8001	0578	
0578	15	0611	0579	
0579	10	0612	0580	
0580	20	0611	0581	
0581	21	0612	0582	
0582	66	0699	0583	
0583	15	0609	0584	
0584	22	0609	0546	
0585	20	0705	0586	
0586	65	0703	0587	
0587	69	0592	0588	
0588	22	0592	0589	
0589	16	0699	0590	
0590	69	0593	0591	
0591	22	0593	0592	
0592	69	0000	0593	
0593	24	0000	0594	
0594	65	0705	0595	
0595	16	0706	0596	
0596	45	0597	0626	
0597	20	0705	0598	
0598	65	0706	0599	
0599	10	8001	0622	

PROBLEM: de Sio Matrix Inversion

LOCATION	OP.	DATA	INSTRUCTION	REMARKS
0600	00	0000	0000	
0601	20	0702	0517	
0602	09	0000	0709 -	
0603	00	0000	0000 -	
0604	66	0706	0530	
0605	13	0705	0602 -	
0606	09	0701	0709 -	
0607	00	0000	0000 -	
0608	65	0707	0538	
0609	07	0000	0000	
0610	00	0700	0000 -	
0611	06	0000	0700 -	
0612	00	0000	0000 -	
0613	66	0706	0553	
0614	13	0705	0609 -	
0615	65	0612	0565	
0616	07	0000	0000	
0617	00	0700	0000 -	
0618	66	0700	0571	
0619	13	0704	0609 -	
0620	65	0699	0585	
0621	12	0602	0000 -	
0622	15	0592	0623	
0623	10	0593	0624	
0624	20	0592	0625	
0625	21	0593	0592	
0626	65	0702	0627	
0627	16	0706	0628	
0628	45	0601	1400	
0699	00	0000	0000	Temporary storage
0700	00	0000	0000	Temporary storage
0701	50	1000	0000	
0702	00	0000	0000	Temporary storage
0703	00	0000	0000	Temporary storage
0704	00	0000	0000	Temporary storage
0705	00	0000	0000	Temporary storage
0706	00	0001	0000	
0707	00	0709	0000	
0708	N/1	0000	N/2	Code word and temporary storage

OSR 25-5
(REV. 10-55)

DU PONT
ESTABLISHED 1802
E. I. DU PONT DE NEMOURS & COMPANY
INCORPORATED
SAVANNAH RIVER PLANT
AIKEN, SOUTH CAROLINA
(TEL. & TEL. ADDRESS, AUGUSTA, GA.)

EXPLOSIVES DEPARTMENT
ATOMIC ENERGY DIVISION

May 21, 1957

To: Recipients of DP-172, "IBM 650 Routine I - Polynomial Fit by Least Squares", by R. R. Haefner

CORRECTION NOTICE

The following errors have been noted in the IBM 650 program of DP-172, "Polynomial Fit by Least Squares":

The instruction in location 1532, page 13, should be 12 1315 0000-
1505, page 19, should be 24 1378 1508+
1435, page 20, should be 03 0000 0000+
0609, page 24, should be 07 0000 0000-
0616, page 24, should be 07 0000 0000-

The printed format from output cards can be improved by loading 00 1570 0001+ into location 1427 in the initial read-in.

The program in the Appendix of DP-172 was given in the logical sequence of computer interpretation. Since the subroutine that transfers the matrix elements to the inversion locations is used differently for elements below the largest diagonal than for elements above that diagonal, both sequences of the subroutine are given in the Appendix. Thus, if one prepared one-field load cards from the program given in DP-172, those cards which apply to the second use of the subroutine would be loaded last and would be executed for the first use of the subroutine, giving erroneous results. To prevent this error, one should insure that the following orders are read-in:

Location	Instruction
1344	24 0710 1342+
1367	44 1347 1385+
1374	21 1341 1376+
1377	10 1183 1378+
1378	21 1343 1340+

The routine described in DP-172 permits the punching of the root-mean-square error, σ, only when $\sigma < K$. The routine was recently revised to punch σ for each order of the polynomial. The following orders are necessary for this revision:

Location	Instruction
1479	11 0057 1567-
1523	24 1400 1521+
1525	24 1427 0026+
1529	21 1461 1487+
1566	71 1427 1519+
1567	69 1568 1569+
1568	71 1427 7000+
1569	24 1566 1487+

TECHNICAL DIVISION
SAVANNAH RIVER LABORATORY

R. R. Haefner
Theoretical Physics Division

RRH:eef