A PROGRAM FOR IDENTIFICATION OF LINEAR SYSTEMS

June Buell

Robert Kalaba

Enrique Ruspini*

Andrew Yakush

Department of Electrical Engineering
University of Southern California
Los Angeles, California 90007

Technical Report No. 71-46

December 1971

* Permanent Affiliation: Space Biology Laboratory, Brain Research Institute, University of California, Los Angeles.

Supported by the National Institutes of Health under Grant No. GM 16197-04, by the National Aeronautics and Space Administration under Grant NGR 05-007-195, and by the Air Force Office of Scientific Research of the Office of Aerospace Research under Contract F 44620-70-C-0017.

ABSTRACT

A program has been written for the identification of parameters in certain linear systems. These systems appear in biomedical problems, particularly in compartmental models of pharmacokinetics. The method presented here assumes that some of the state variables are regularly modified by jump conditions. This simulates administration of drugs following some prescribed drug regime.

Parameters are identified by a least-square fit of the linear differential system to a set of experimental observations. The method is especially suited when the interval of observation of the system is very long.

A PROGRAM FOR IDENTIFICATION OF LINEAR SYSTEMS

1. Computational Methods

A program for identification of linear systems is presented. By system identification is meant the evaluation of some of the internal parameters of a system so that a set of experimental observations on the state variables is optimally approximated.

The type of model considered here is suggested by studies of the time history of a drug in the human system using linear compartmental models [1]. In this particular field, system identification is an important tool not only for the study of the general pharmacokinetic behavior of a drug but also for the determination of individual differences between subjects. This may help in the prescription of a drug regime especially suited to individual characteristics.

The method used here, expecially useful when the period of observation is long, was derived and described in a previous paper [2]. The linearity of the system is used to obtain a series of recurrence relations that help reduce the amount of computation.

It is assumed that doses are administered at equal intervals and that every dose (including the initial dose) instantaneously increases the concentration of the compartment into which it is injected. Between doses, it is assumed that the system is described by a linear, homogeneous, vector-matrix differential equation [3]:

$$\dot{x} = Ax \quad . \tag{1}$$

The matrix A depends on various parameters $\alpha^1, \ldots, \alpha^m$ whose precise values are not known. It is desired to estimate those values on the basis of dynamic measurements made on the system just before each dose is administered.

It is assumed here that the time scale is chosen so that the interval Δ between dosages is

(2) $$\Delta = 1 \quad .$$

The injection of a dose at times $t = 1, \ldots, N - 1$ is simulated by the jump condition

(3) $$x(i + 0) - x(i - 0) = d_i \quad , \quad i = 1, 2, \ldots, N - 1 \quad .$$

The initial condition is

(4) $$x(0) = c \quad .$$

At times $t \neq i$ $(i = 1, 2, \ldots, N)$, the state vector x is controlled by the relation of Equation (1). The first k variables of x; $x^1, \ldots, x^k$, are assumed to have been observed at times $t = i, i = 1, \ldots, N$. Let the values of these observations be

(5) $$b^j(i) \quad , \quad j = 1, \ldots, k; \; i = 1, \ldots, N \quad ,$$

where $b^j(i)$ stands for the observation of the j-th variable at the time $t = i$.

The method requires knowledge of an approximation $(\alpha_1^0, \ldots, \alpha_m^0)$ for the desired parameters. Proceeding from this approximation and using a combination of a Newton-Raphson scheme and a series of recurrence relations, an iterative scheme is followed which eventually minimizes the quadratic form

(6) $$S = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_j \left(x^j(i) - b^j(i)\right)^2 \quad ,$$

where the γ_j are appropriate weights, provided that a sufficiently close initial approximation is given.

The method proceeds according to the following steps:

(i) The following initial value matrix-matrix differential systems

$$\frac{d}{dt}\phi = A\phi \quad , \quad 0 \le t \le 1 \quad , \tag{7}$$

$$\phi(0) = I \quad , \tag{8}$$

where I is the unit matrix of dimension r ($r = \dim x$);

$$\frac{d}{dt}\Psi_j = A\Psi_j + A_j\phi \quad , \quad 0 \le t \le 1 \quad , \tag{9}$$

$$\Psi_j(0) = 0 \quad , \quad j = 1, \ldots, m \quad , \tag{10}$$

where Ψ_j is an $r \times r$ matrix and

$$A_j = \partial/\partial\alpha_j A \quad , \quad j = 1, \ldots, m \quad ; \tag{11}$$

and

$$\frac{d}{dt}\Lambda_{jq} = A\Lambda_{jq} + A_q\Psi_j + A_j\Psi_q + A_{jq}\phi \quad , \quad 0 \le t \le 1 \quad , \tag{12}$$

$$\Lambda_{jq}(0) = 0 \quad , \quad j,q = 1, \ldots, m \quad ; \tag{13}$$

where Λ_{jq} are $r \times r$ matrices and

$$A_{jq} = \partial^2/\partial x_j \partial x_q A \quad , \quad j,q = 1, \ldots, m \quad ; \tag{14}$$

are integrated from $t = 0$ to $t = 1$. Notice that

$$\Lambda_{jq}(t) = \Lambda_{qj}(t) \quad , \quad 0 \le t \le 1 \quad . \tag{15}$$

The program integrates the above system of $\frac{1}{2}r^2(m+1)(m+2)$ differential equations by a combination of a Runge-Kutta and Adams-Moulton schemes. During the integration a total of $5r^2(m+1)(m+2)$ places are required to store auxiliary quantities. When the values of the matrices $\phi(1)$, $\Psi_j(1)$, $\Lambda_{jq}(1)$, $j,q = 1, \ldots, m$ are obtained, the program requires their storage which occupies $\frac{1}{2}r^2(m+1)(m+2)$ memory locations. No further auxiliary values are stored. In the previous description it is assumed that A, A_j, and A_{qj} are computed using the current approximation of α.

(ii) Denoting by $(\alpha_1^0, \ldots, \alpha_m^0)$ the current approximation of the parameters to be identified, the following quantities are computed

$$E_n = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_j (x^j(i) - b^j(i)) \frac{\partial x^j(i)}{\partial \alpha_n}, \quad n = 1, \ldots, m, \tag{16}$$

$$B_{nq} = \sum_{i=1}^{N} \sum_{j=1}^{k} \gamma_j \left[(x^j(i) - b^j(i)) \frac{\partial^2 x^j(i)}{\partial \alpha_n \partial \alpha_q} + \frac{\partial x^j(i)}{\partial \alpha_n} \cdot \frac{\partial x^j(i)}{\partial \alpha_q} \right] \quad n,q = 1, \ldots, m \tag{17}$$

requiring a total of $m(m+3)/2$ memory locations.

The values of the vectors $x(i)$,

$$x_n(i) = \partial x(i)/\partial \alpha_n, \quad n = 1, \ldots, m, \tag{18}$$

and

$$x_{nq}(i) = \partial^2 x(i)/\partial \alpha_n \partial \alpha_q, \quad n,q = 1, \ldots, m, \tag{19}$$

are computed recursively using the relations

$$x(i+1) = \phi(1)\, x(i+0) \quad , \tag{20}$$

$$x(i+0) = x(i) + d_i \quad , \quad i = 1, \ldots, N-1 \quad , \tag{21}$$

$$x(0) = c \quad , \tag{22}$$

$$x_n(i+1) = \phi(1) x_n(i) + \Psi_n(1) x(i+0) \quad , \tag{23}$$

$$x_n(0) = 0 \quad , \quad n = 1, \ldots, m;\ i = 1, \ldots, N-1 \quad , \tag{24}$$

$$x_{nq}(i+1) = \phi(1)\, x_{nq}(i) + \Psi_q(1) x_n(i) + \Psi_n(1) x_q(i) + \Lambda_{nq}(1)\, x(i+0) \tag{25}$$

$$x_{nq}(0) = 0 \quad , \quad n, q = 1, \ldots, m;\ i = 1, \ldots, N-1 \quad . \tag{26}$$

Values of the quantities produced by the recurrent process need not be sequentially stored and the quantities defined by Equations (16) and (17) may be computed as running sums. The total storage requirement for this step is then $\frac{1}{2} r (m+1)(m+2) + \frac{1}{2} m (m+3)$.

(iii) The linear algebraic system

$$E_n + \sum_{q=1}^{m} B_{nq} (\alpha_q^1 - \alpha_q^0) = 0 \quad , \quad n = 1, \ldots, m \quad , \tag{27}$$

is solved for α_q^1, $q = 1, \ldots, m$. The result is the new approximation of the parameter vector α.

Steps (i), (ii), and (iii) are repeated until

$$|\alpha_q^1 - \alpha_q^0| < \epsilon, \quad q = 1, \ldots, m, \tag{28}$$

where ϵ is user provided.

2. Flowchart

The flowchart in Figure 1 is explained by the description of the computational method in the previous section.

3. Typical Sample Run

The output of the sample run below includes the following information after the run number identification.

(a) The input constants IC(i), i = 1, ..., 13 which are user supplied. They are integer constants of the program and integer control parameters. Their function is described in the software specification section.

(b) The input constants RC(i) which are noninteger constants to be supplied by the user.

(c) The initial conditions $x(0) = c$.

(d) The initial approximations for the parameters to be identified (α_q^0, q = 1, ..., m).

(e) The drug dosages used during the observation period.

(f) The experimental observations on the system.

From this point on, the following output is printed at each step of the iterative process.

(g) The iteration cycle number.

(h) The current value of the parameters to be identified.

(i) Optional. The value of the matrix $\phi(t)$, as computed by the integration, is printed every MP steps. This option was used in the sample run with MP = 100.

(j) Optional. The values $x(i - 0)$ are printed at every observation point. This option was not used in the sample run.

(k) Optional. The values B_{nq} are printed in matrix form B and the values E_n in vector form E. This option was used in the sample run.

(l) Optional. Prints the matrix B^{-1}, the value of the determinant of B^{-1}, the vector $B^{-1}E$, and the matrix BB^{-1}. This option is used to check the reliability of the matrix inversion and was used in the sample run.

(m) The corrected value of the parameters.

When the iteration ends, a message is printed letting the user know whether convergence has occurred or the allowed number of iterations has been exceeded.

The values of $x(j - 0)$ at the observation points using the identification parameters are printed.

The particular identification problem studied in the sample run is the determination of the constants u_1, u_2 in the system.

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = -u_1 x_1(t) + u_2 x_2(t) \ , \ i \leq t < i + 1 \ , \ i = 0, 1, \ldots, 9 \ .$$

$$x_1(i + 0) - x_1(i - 0) = 0$$

$$x_2(i + 0) - x_2(i - 0) = 1 \ , \quad i = 1, \ldots, 10$$

with the initial conditions

$$x_1(0) = 0$$

$$x_2(0) = 1 \quad .$$

Observations of the variable x_1 used in the sample run were generated at times $t = 1, 2, \ldots, 10$ by exact integration of the above system using the values $u_1 = 10$ and $u_2 = 11$. No observations of the variable x_2 were made available to the program.

In this example, the solutions can be expressed in the form

$$x_1(t) = A_1 e^{-t} + A_2 e^{-10t}$$

$$x_2(t) = B_1 e^{-t} + B_2 e^{-10t}$$

where the numbers A_1, A_2, B_1, B_2 depend on the particular value $(i, i + 1)$ to which t belongs but are otherwise independent of time.

Since e^{-t} is much larger than e^{-10t} except in a neighborhood of zero, this seemingly simple differential system creates a difficult identification problem. Initial estimates for the parameters u_1 and u_2 were $u_1 = 8$, $u_2 = 8$.

Sample Run

INPUT CONSTANTS IC(I) RUN NUMBER 2

1	2
2	2
3	2
4	1
5	10
6	10
7	0
8	1
9	0
10	1
11	1
12	100
13	3

INPUT CONSTANTS RC(I)

1	0.99999979E-02
2	0.99999993E-03
3	0.10000000E-01

INITIAL CONDITIONS C(I)

1	0.0
2	0.10000000E-01

INITIAL APPROXIMATION TO UNKNOWN CONSTANTS UC(I)

1	0.80000000E-01
2	0.80000000E-01

DOSAGES

1.0	0.100000E-01
2.0	0.100000E-01
3.0	0.100000E-01
4.0	0.100000E-01
5.0	0.100000E-01
6.0	0.100000E-01
7.0	0.100000E-01
8.0	0.100000E-01
9.0	0.100000E-01
10.0	0.100000E-01

```
OBSERVATIONS

        0.40870443E-01
        0.55907674E-01
        0.61439544E-01
        0.63474536E-01
        0.64223170E-01
        0.64498603E-01
        0.64599931E-01
        0.64637244E-01
        0.64650953E-01
        0.64655960E-01

CYCLE NUMBER    1

OLD CONSTANTS
UC(  1) =          0.80000000E-01
UC(  2) =          0.80000000E-01

INTEGRATION OF PHI AT TIME = 1.000
 0.373832E-00   0.545867E-01
-0.436703E-00  -0.628696E-01

SECOND PARTIALS OF  S  MATRIX

 0.244998E-02  -0.410195E-03
-0.410195E-03   0.994315E-05

PARTIAL OF  S  VECTOR

-0.203786E-02
 0.444756E-03

INVERSE OF SECOND PARTIALS OF  S  MATRIX

-0.690977E-02  -0.285057E-04
-0.285057E-04  -0.170256E-05

DETERMINATE =  -0.143900E-06

PRODUCT OF MATRIX INVERSE AND PARTIAL VECTOR

-0.112699E-01
-0.176319E-01
```

PRODUCT OF MATRIX AND INVERSE

```
 0.100000E-01     0.190735E-05
-0.707805E-07     0.999999E-00
```

CYCLE NUMBER 1

NEW CONSTANTS
UC(1) = 0.91269932E-01
UC(2) = 0.97631855E-01

- -

CYCLE NUMBER 3

OLD CONSTANTS
UC(1) = 0.97914543E-01
UC(2) = 0.10681258E-02

INTEGRATION OF PHI AT TIME = 1.000

```
 0.405721E-00   0.419558E-01
-0.410815E-00  -0.424259E-01
```

SECOND PARTIALS OF S MATRIX

```
 0.859218E-03  -0.189624E-03
-0.189624E-03   0.445226E-04
```

PARTIAL OF S VECTOR

```
-0.111303E-03
 0.235987E-04
```

INVERSE OF SECOND PARTIALS OF S MATRIX

```
0.193811E-05    0.825453E-05
0.825454E-05    0.374026E-06
```

DETERMINATE = 0.229721E-08

PRODUCT OF MATRIX INVERSE AND PARTIAL VECTOR

-0.209212E-00
-0.361004E-00

PRODUCT OF MATRIX AND INVERSE

0.999982E-00 -0.610352E-04
0.0 0.100001E-01

CYCLE NUMBER 3

NEW CONSTANTS
UC(1) = 0.10000666E-02
UC(2) = 0.11042262E-02

- -

CYCLE NUMBER 6

OLD CONSTANTS
UC(1) = 0.99998312E-01
UC(2) = 0.10999827E-02

INTEGRATION OF PHI AT TIME = 1.000
 0.408749E-00 0.408706E-01
-0.408704E-00 -0.408252E-01

SECOND PARTIALS OF S MATRIX

 0.766933E-03 -0.174278E-03
-0.174278E-03 0.451110E-04

PARTIAL OF S VECTOR

-0.184601E-07
 0.397415E-08

INVERSE OF SECOND PARTIALS OF S MATRIX

0.106786E-05 0.412546E-05
0.412546E-05 0.181547E-06

DETERMINATE = 0.422444E-08

```
PRODUCT OF MATRIX INVERSE AND PARTIAL VECTOR

-0.331756E-04
-0.400703E-04

PRODUCT OF MATRIX AND INVERSE

0.999994E-00     0.0
0.953674E-06     0.999998E-00

CYCLE NUMBER    6

NEW CONSTANTS
UC(  1)=          0.99998636E-01
UC(  2) =         0.10999867E-02
CONVERGENCE OF UNKNOWN CONSTANTS

INTEGRATION OF SYSTEM USING IDENTIFIED PARAMETERS

CYCLE NUMBER    6

NEW CONSTANTS
UC(  1) =         0.99998636E-01
UC(  2) =         0.10999867E-02

X( 1, 1-0) =      0.40870443E-01
X( 2, 1-0) =     -0.40825028E-01

X( 1,2-0) =       0.55907670E-01
X( 2, 2-0) =     -0.55862244E-01

X( 1, 3-0) =      0.61439551E-01
X( 2, 3-0) =     -0.61394121E-01

X( 1, 4-0) =      0.63474596E-01
X( 2, 4-0) =     -0.63429177E-01

X( 1, 5-0) =      0.64223230E-01
X( 2, 5-0) =     -0.64177811E-01

X( 1, 6-0) =      0.64498663E-01
X( 2, 6-0) =     -0.64453185E-01

X( 1, 7-0) =      0.64599991E-01
X( 2, 7-0) =     -0.64554513E-01
```

```
X( 1,  8-0) =     0.64637244E-01
X( 2,  8-0) =    -0.64591825E-01

X( 1,  9-0) =     0.64650953E-01
X( 2,  9-0) =    -0.64605534E-01

X( 1, 10-0) =     0.64655960E-01
X( 2, 10-0) =    -0.64610541E-01
```

4. Hardware and Software Specifications

(a) HARDWARE SPECIFICATIONS

The program was tested on an IBM 360/60 system. The sample run presented above uses the values

$$r = 2 , \tag{29}$$

$$m = 2 , \tag{30}$$

$$N = 10 , \tag{31}$$

and the step used in the integration was $h = 0.01$. The total processor time was 28 seconds. Several similar examples were used to test the program.

(b) SOFTWARE SPECIFICATIONS

The program is written in FORTRAN IV (G). The following table lists those parameters that are to be supplied by the user. The parameters are to be punched in cards with one entry per card consisting of an integer i (the variable number)* and the value of the variable. The table includes the program names of the variable, the symbol used for

---

* The variable number has the purpose of helping the user organize his data deck by consecutive enumeration of the individual cards on the first two columns. It does not serve any other purpose.

the variable in the theory (if any), the format in which the card is to be punched, and the value of the variable. All the values should be mandatorily supplied. Also given in the table is the value of the variable in the sample run.

TABLE 1

Data Input Requirements

a) Integer Values

Program Names	Symbol	Meaning	Format	Sample Run
IC(1)	-	run number for use of user records	I2, E21.8	2
IC(2), NE	r	number of linear ordinary differential equations	I2, E21.8	2
IC(3), NUC	m	number of parameters to be identified	I2, E21.8	2
IC(4), NCS	k	number of variables observed	I2, E21.8	1
IC(5), NC	N	number of observation times	I2, E21.8	10
IC(6), NCYCLE	--	maximum number of iterations to be computed	I2, E21.8	10
IC(7), IFLAG	--	= 1, observations are to be generated by exact integration of x with given values =0, this process is omitted	I2, E21.8	0
IC(8), IFLAG1	--	=1, then the matrix $\phi(t)$ is printed every MP-th step =0, no such printout	I2, E21.8	1
IC(9), IFLAG2	--	=1, x(i - 0) at every observation point is printed =0, no such output	I2, E21.8	0

Program Names	Symbol	Meaning	Format	Sample Run
IC(10), IFLAG3	--	=1, matrix B and vector E are printed =0, no such printout	I2, E21.8	1
IC(11), IFLAG4	-- -0	=1, B^{-1}, det B^{-1}, $B^{-1}E$ and BB^{-1} are printed =0, no such printout	I2, E21.8	1
IC(12), MP		See IC(8)		100
b) Real Variables				
RC(1), H	h	integration step for integration of fundamental matrices	I2, E21.8	0.01
RC(2), EPS	ϵ	criterion for convergence	I2, E21.8	0.001
RC(3), GAMA(1)	γ_1	weighting factor for observations of first component of x	I2, E21.8	1.0
. . . . .		. . . . .	. . . . .	. . .
RC(2+NOS), GAMA(NOS)	γ_k	see above	I2, E21.8	1.0
C(1)	c_1	initial condition for variable 1	I2, E21.8	8.0
. . . . .		. . . . .	. . . . .	. . .
C(NE)	c_r	initial condition for variable r	I2, E21.8	8.0
UCO(1)	α_1	initial guess for parameter α_1	I2, E21.8	8.0
. . . . .		. . . . .	. . . . .	. . .
UCO(NUC)	α_m	initial guess for parameter α_m	I2, E21.8	8.0

The option given by IC(7) allows the program to be tested by using values produced by exact integration of the given system as experimental data. This is a good feature to check the reliability and convergence characteristics of the program for the particular user's system.

Subroutines Supplied by the Program

1.	INT1 and INT2	Adams-Moulton integration routines for the integration of the fundamental matrices $\phi(t)$, $\Psi_j(t)$, $\Lambda_{jq}(t)$.
2.	DAUX	Contains the differential equations for the fundamental matrices. Called by INT1 and INT2.
3.	MATINV	Matrix inversion of B .
4.	MULT **	Storage of matrix B before inversion. Multiplication of B by B^{-1} and printout. This routine is called if IFLAG4 = 1.
5.	OUTCON	Output of old and new estimates of the unknown parameters each cycle.
6.	OUT1 **	Called if IFLAG1 = 1 to output ϕ every MP-th step.
7.	OUT2(1B) **	Called if IFLAG2 = 1 to output the value x(1B-0) at time 1B .
8.	OUT3 **	Called if IFLAG3 = 1 to output B matrix.
9.	OUT4 **	Called if IFLAG = 1 to output B^{-1} , Determ B^{-1} , and $B^{-1}E$.

** Fortran printing format is 8E15.6. If more than eight differential equations are used, subsequent values are printed in following lines. The format may be changed by user.

Subroutines to be Supplied by User

1.	SYSMAT	Input of the system matrix A . Array A(I, J) should contain the (I, J)-th element of the system matrix A . The equations for the compartments being observed must be the first equations of the system.

2. PDA — Input of partial derivatives of A with respect to the unknown parameters. Array AP(K,i,J) should contain the (I,J)-th element of $\partial A/\partial \alpha_k$. Array APP(M,I,J) should contain the (I,J)-th element of $\partial A/\partial \alpha_k \partial \alpha_\ell$ where $M = k(k-1)/2 + \ell$ for $k \geq \ell$.

3. INDOSE — Input and printout (if desired) of dosages. DI(I,J) should contain dosage to be administered to I-th compartment at time J .

4. CHANGE — Called if IFLAG = 0 . Input and printout (if desired) of observations BI(I,J) should contain the observation of the i-th compartment at time J .

Dimensioning

To be supplied by user to fit his particular demand and memory constraints.

The following arrays are dimensioned in the main program.

DIMENSION T(N*10), PD(NUC),BB(NUC,NUC),IPIVOT(NUC),INDEX (NUC,2),PIVOT(NUC) where N = (1+NUC)*(2+NUC)/2*NE**2.

The following COMMON statement with arrays dimensioned as indicated should be the first statement of the main program and the second of all subroutines.

```
COMMON DETERM,EPS,HH,ICYCLE,IFLAG,IFLAG1,IFLAG2,IFLAG3,
       IFLAG4,JMAX,NN,N2,NCON,NCYCLE,NE,NO,NOS,NUC,
       A(NE,NE),AP(NUC,NE,NE),APP(M,NE,NE),B(NUC,NUC),
       BI(NOS,NO),C(NE),DI(NE,NO),E(NUC,1),GAMA(NOS),
       IC(12),RC(2+NOS),UCO(NUC),X(NE),XP(NE,NUC),XPP(NE,M),
       Y(NE),YP(NE,NUC),YPP(NE,M)
```

where M = NUC*(NUC+1)/2.

5. Mode of Availability

Persons interested in obtaining a copy of this program should contact

June Buell
Dept. of Electrical Engineering
University of Southern California
Los Angeles, California 90007

6. Indexing Information

System identification, linear systems, drug regimes, pharmacokinetics.

REFERENCES

1. Kruger-Thiemer, E., "Formal Theory of Drug Dosage Regimes," Part 1, Journal of Theoretical Biology, Vol. 13, No. 2, 1966.

2. Buell, J., R. Kalaba, and E. Ruspini, "Identification of Linear Systems Using Long Periods of Observation," Journal of Optimization Theory and Applications, Vol. 5, No. 3, March 1970, pp. 170-177.

3. Bellman, R., Introduction to Matrix Analysis, McGraw-Hill Book Company, New York, 1960.

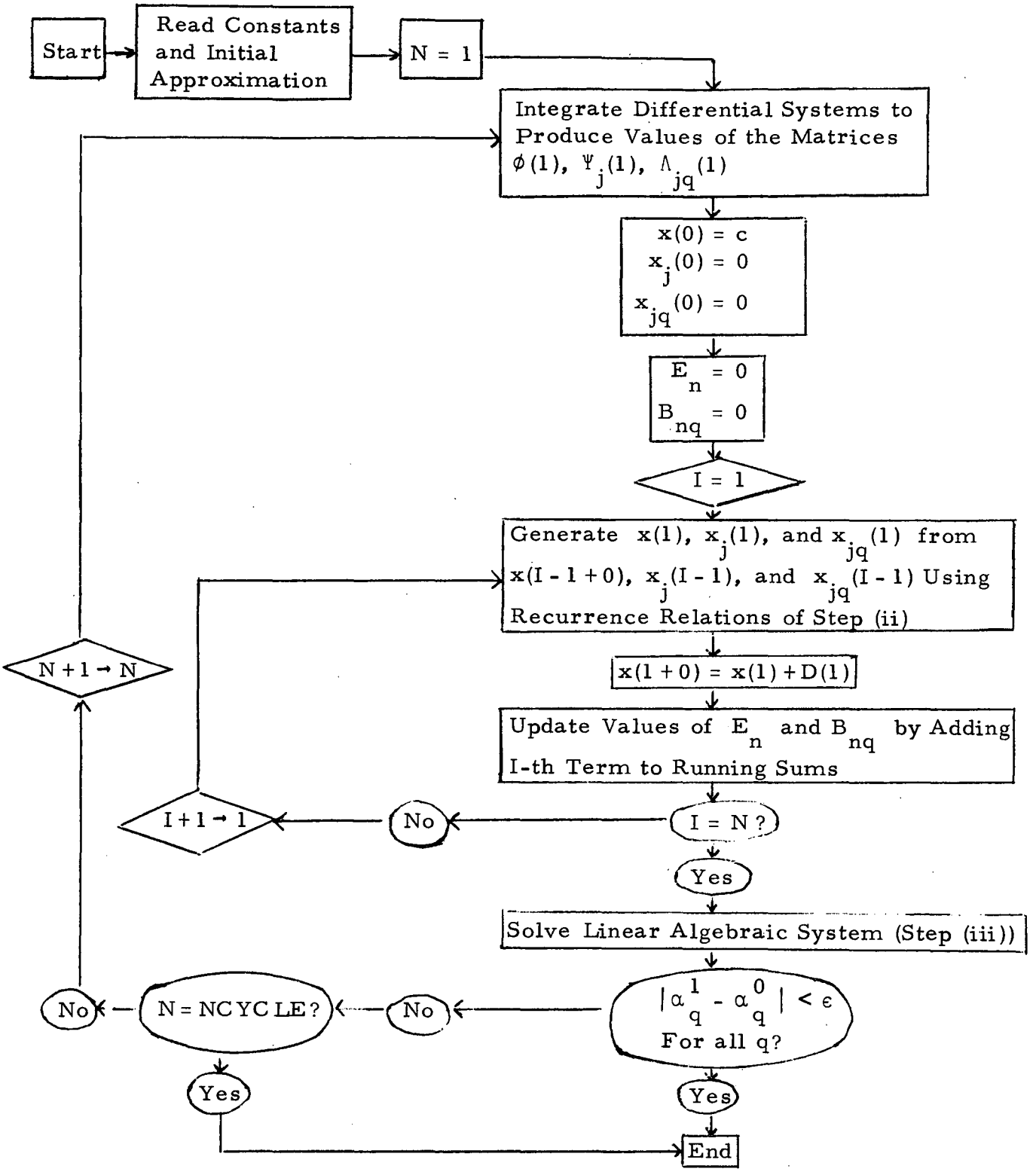

Figure 1

Flowchart of the Method