A System for Large Econometric Models — Management, Estimation, and Simulation

M.C. McCracken and Carl A. Sonnen
Econometric Applications Co.

Abstract

A software system to build econometric models that have nonlinear relations requiring simultaneous solution is described. The system is presented in the context of a Canadian government research project that constructed a model of almost 1,600 equations. The system, which is designed to exploit the modular nature of the modelling process, contains programs that assist in the tasks of data management, estimation, solution, analysis, and reporting. Programs are provided to facilitate documentation of the model and error checking at the critical decision points of the model building process. Inputs of researchers generally are in the language familiar to economists and other social scientists, and little training has been required to integrate new researchers into the Canadian project. All parts of the system, which is fully documented, are operational. Some programs are operational on several systems.

KEYWORDS AND PHRASES: Information Management, Research Management, Econometric Research Process, Economic Research, Project Management, Large Model, Equation Estimation, Nonlinear Systems, Simulation, Canada

CR CATEGORIES: 3.31, 3.51, 4.42, 5.41, 8.1

INTRODUCTION

The development of econometric models has been a prominent characteristic of economic research over the last two decades. In macroeconomic research, or the study of major aggregations of economic activity, such models attempt to represent, in a consistent manner, the complex interactions of the economic process under review. Stated in equation form, such models may be described as:

$$Y_{i_t} = f(B_n, Y_{l_t}, X_{m_t}, Y_{t-j}, X_{t-k}, C_i)$$

or, the ith endogenous value in period t may be a function of some endogenous and exogenous value(s) in period t and of some endogenous and exogenous value(s) in periods earlier than t. Optionally, the equation may be subjected to constant adjustments. In the general case, the system of equations will include nonlinear specifications and will be simultaneous.

In formulating a model, the econometrician uses statistical tools to test alternative economic theories of individual relations against available data. When all equations in the model are specified and coefficients are estimated, the model is solved and its properties are analysed. This almost always leads to an iterative process in which there is a continuing interaction between solution of the model and respecification of individual equations. Finally, the model is run under alternative assumptions, the results are reported, and the model is documented. This view of a model's development suggests a modular organization for the research process.

In a static sense, modularity can be rationalized on conventional efficiency grounds, for the skills necessary for data gathering and maintenance, equation specification and estimation, system solution and analysis, and report generation can be distinguished.

A modular organization becomes imperative for large econometric models, mainly because size sharpens the specialization requirement and the corollary need for resource management. Among other, more obvious tasks, the project manager must ensure that specification of equation blocks by the different teams of economists is compatible with economic theory and does not build in explosive structures.

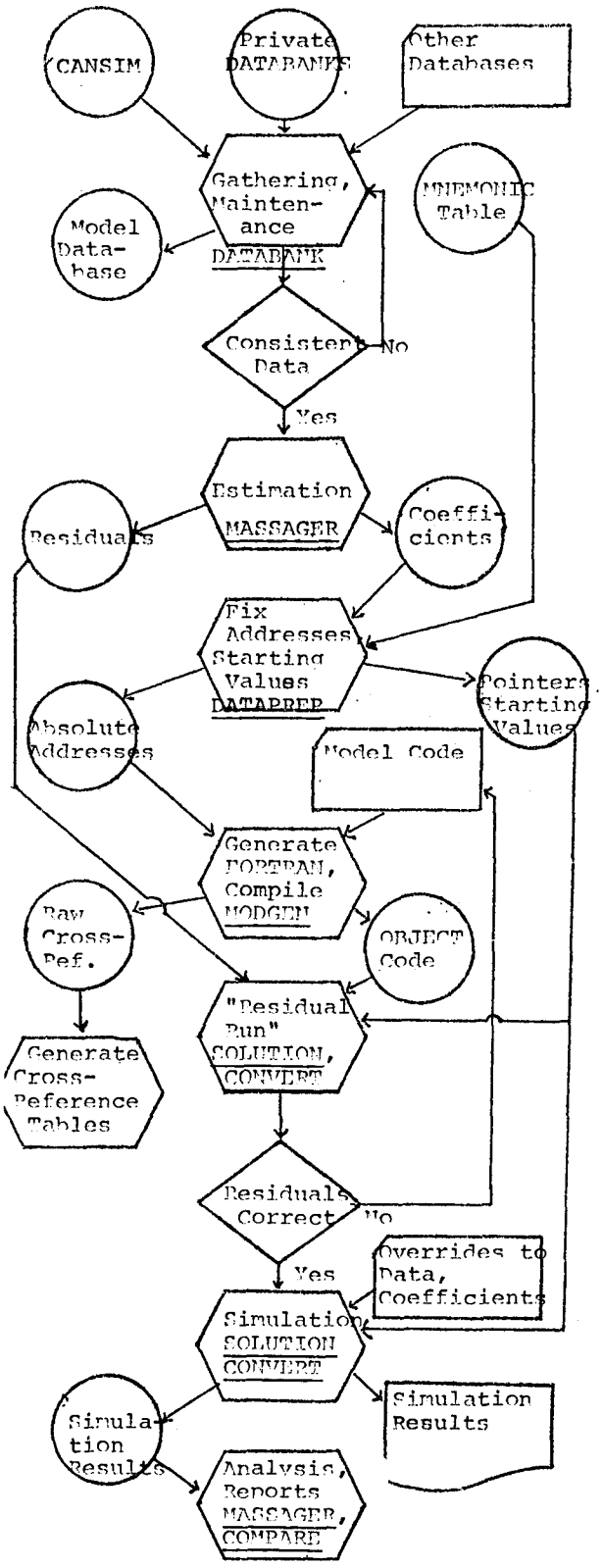
He also must identify the prospective demands on the model's services, ensuring that the equation structure permits easy use of the model for analytical studies and conditional forecasts. He budgets available resources during the model's construction, ensuring that data is available for the economists estimating equations and shifting resources to those research areas that are falling behind schedule. Mainly because he is likely to have the best global view of the model's purposes and structure, he becomes most actively involved in the mechanics of the process at that point where the blocks of equations are brought together and the model is tested for its dynamic properties. Finally, he must ensure that prospective users of the working model are informed of its capabilities and limitations.

The management and software system described below was used to develop an econometric model for the CANDIDE Model 1.0 project, an inter-agency activity of the Canadian government. A description of the model structure may be found in [1]. That model, which has been operational since January 1972, solves for almost 1,600 endogenous variables, uses more than 300 exogenous variables, and depends on more than 18,000 coefficients. For purposes of simulating the system under alternative assumptions, the model is available to all the agencies of the Canadian government that participate in the project. The model software was developed under CP-67/CMS and is operational on that system. The complete set of software also is operational for OS/360. Some programs also are operational under other systems.

The plan of this paper is to present a detailed description of the software system and the division of human activity employed to support the CANDIDE project. A schematic of the organization, software, and processing used in that project appears as Fig. 1. Because we view the research process as recursive, the discussion below is organized to cover, as phases, those activities that tend to have the most interaction. The data mangement, and equation specification and estimation activities are treated as the first phase. This is followed by a description of the "language" used to define the model in executable form. The third section describes the software that solves the system of equations and prepares the output for report generation. The final section describes such facilities as cross-referencing and optimal ordering that are particularly helpful in the development and maintenance of large models.

Figure 1

Research Process and Software



DATA MANAGEMENT; EQUATION SPECIFICATION

Overall Structure of the Project

The first meetings to organize the CANDIDE project were held in January 1971. The human resources available to the project included approximately a dozen economists, most of whom had several years of experience in specialized research subjects and were familiar with the data bases associated with those subjects. None had any knowledge of higher-level languages and only a few had even casual experience in the use of "canned" programs that generally support economic research. Two software packages -- one for data maintenance and the other for equation estimation -- that had been operational for several years were available. The Canadian federal government's major data base for variables of principal concern to economists, moreover, could produce copies of selected information in the structure used by both of the available software packages. A non-economist with a year's experience in the use of the data management software was hired to serve as the manager of the data base. Two programmers were temporarily commissioned to assist in the development of the software for model simulation.

Given these resources, two factors were emphasized in the development of the model definition and simulation software. The system is flexible in the sense that any subset of the model can be defined, estimated and simulated independently of the rest of the model and reflects an attempt to make use of special knowledge. The convention that the definition of equations should be expressible in the quasi-algebraic form with which economists are familiar is adhered to. The second focus, in the design of the system stems from the estimate that for each definition of the model, some multiple number of simulation runs will be made to test the performance of the model or to make conditional forecasts. This forces a clear emphasis on minimizing core requirements at the simulation, or solution stage. In that stage, only those values necessary for solving one period are held in core. Lagged values are "tattered", that is, space in memory is reserved only for those values of a variable for which lags are necessary.

A file that we, by convention, refer to as the MNEMONIC Table integrates the first stage of the model's development. It documents responsibility among economists for data reliability and for equation estimation of each variable. As an input to the DATAPREP program, it is critical to the software in that it establishes the block (and equation) structure of the model, and the ability to convert symbolic expressions of equations to an array form. It also establishes most of the dimensions of data arrays used by later programs. The block and equation structure generally is determined from economic theory. (The specification, for example, of disaggregated consumption functions almost certainly will be "modelled" in a patterned form.) A fixed-length record file, this Table provides for every variable used in the model, its block and equation number, the length of the longest lag associated with this variable in any equation, and a short descriptive title to identify the MNEMONIC. The variable also is characterized as being exogenous or endogenous; and if the latter, it further is classified either as a behavioral or identity equation. Exogenous variables are treated as members of Block 0. This Table, which in CANDIDE was maintained with the file maintenance facilities available under CP-67/CMS by the data manager, was constructed concurrently with the gathering of data and estimation of equations. Also during this stage of the model's development, the table documented responsibility among the economists for data reliability and equation estimation of each variable.

Accumulation and Maintenance of Data

The mechanics of data maintenance were handled by the data manager, mainly to exploit the advantages of specialization but also to minimize the danger of duplicate series. The software system used is the DATABANK program [2,3], which has been in widespread use in Canada and elsewhere since early 1966. This feature is significant in that a considerable share of the data needed to formulate the model was immediately available to the economists in the DATABANK structure from the CANSIM data base, the federal government's statistical base maintained by Statistics Canada [4]. A number of other databases needed by the researchers, moreover, were available in DATABANK format.

The DATABANK program is designed to maintain data in time series form on magnetic tape. Records are variable-length and are stored in binary form. In addition to values for each time period, which may be annual, quarterly, or monthly information, each record may contain title, source, and note information in the series. The series is identified by a MNEMONIC. Optionally, security codes for purposes of retrieval and editing may be associated with each series. As a rule of thumb, approximately 10,000 series could be placed on a single 2,400 foot reel of tape. To minimize processing time dedicated to maintenance, the file is not sorted and all edits are held in core. Depending, obviously, on the area

reserved for edits, this limits the number of edits that can be performed at any one time and makes the program particularly efficient for large-core computers. For a case in which the number of edits in one pass of the file is limited by setting the number of edit requests to 2,000 new, single-precision words of information, DATABANK runs on an IBM/360 in a little more than 160K, exclusive of buffer area. The program presently is implemented on CDC and UNIVAC equipment as well.

DATABANK performs many of the operations conventionally associated with the addition, deletion, and modification of sequentially organized records. Records can be listed and indexed and can be copied in whole or in part onto other tapes. A full set of information also may be punched as card output for transfer of the data between different systems. A listing of the operations currently in the system is provided as Fig. 2.

Figure 2

DATABANK Operations

Edit Phase

- Change parameters
- Drop a series
- Change security code of a series
- Change identifying code of a series
- Replace notes
- Add word noting verification
- Replace a given number of data values
- Drop years from either end of a series
- Add new years to either end of a series
- Replace short title information
- Replace source information
- Change edit security code of a series

Add Phase

- Construct a master file with indicated series
- Add new series to a mater file

Special Operations

- Check, update, or merge master files
- Control print of audit trail in edit phase
- Copy an old master file to a new master for a specified range or for particular series
- Index the series on a master
- Print, in full, the series on a master
- Control input devices for other operations

Estimation of Equations

The economists involved in the project were responsible for the accuracy of the data in an accounting sense and, as noted, were involved primarily in the estimation of equations by block. The software used for both purposes was the MASSAGER program [3,5], a system in operation also since 1966. Considerable modifications and extensions to the program have been made since that time. MASSAGER is designed mainly as a manipulator of time series information where the operands usually are vectors but may be arrays in some instances. It was programmed mainly to perform such standard statistical procedures as linear and nonlinear multiple regressions, distributed lags, instrumental variables, three-pass least squares, and regression models with auto-correlated residuals. It also may be used to generate output tapes in user-specified formats. Finally, it is used to punch out the coefficients generated from regression operations. A listing of the operations currently in MASSAGER is provided in Fig. 3. Data inputs are allowed for variable-length records in DATABANK format and for fixed-length records, either in a predefined format or in a user-supplied format.

As with the DATABANK program, MASSAGER was designed to take advantage of high-speed memory and accordingly, the amount of data that can be manipulated in one run is restricted by the size of the working array that conventionally is set at 10,000 locations. For some, this is restrictive, but it is possible to re-use columns in the data matrix for different calculations. With the working array set at 10,000 locations, MASSAGER runs in about 180K on an IBM/360, exclusive of buffer areas. The program presently is implemented also on CDC and UNIVAC equipment.

The use of these two programs to build, maintain, and verify the data base, and to specify and estimate equations took the team of economists a little more than nine months, at which time a "first pass" set of equations had been defined. Estimation was performed using the technique of Ordinary Least Squares. For purposes of estimation, MASSAGER control decks were set up to perform three basic tasks for each behavioral equation: (a) make an estimate of the equation, (b) punch the coefficient values, and (c) punch DATABANK control cards that were used to create a DATABANK tape of the calculated values for all stochastic equations. Control decks were maintained as card files and almost all estimation was performed with remote batch facilities.

Figure 3

MASSAGER Operations

Unary Operations

log(e)X
log(10)X
sin X
cos X
X**w
e**X
X(t)-X(t-k)
X(t-k)
1/X
Cumulator (backwards and/or forwards)
c*X
SQRT(X)
Index (deflate a series)
Collapse over a specified range
c+X
Scaling
Move from one storage column to another
Delete data values
Rank values
Three-group values
Percent change
Weighted moving sum
Truncation
Row and column shift
Absolute value
Base 10 antilog
Max(X(t), Z(t-1))
Sign of X

Series Generation

Random number generator (0,1)
Dummy (1,0,...)
Time trend
Constant term

Multiple Operations

X+Y
X-Y
X*Y
X/Y
Weighted sum of variables
Summation of contiguous variables

Input and Output

Output by variable
Output by observation
Core save
Core restore
Input of individual elements
Generalized input
Punch out regression coefficients

Regression Analysis

Multiple regression
Three-pass least squares
Nonlinear regression
Instrumental variables regression
Residual analysis
Generation of distributed lags -- Almon

Figure 3 (cont.)

Fisher instrumental variables
Autocorrelation -- Hildreth-Lu and Merlove

Special Operations

Multiple plot
Simple plot
Report generation
Growth rates

Matrix Operations

Matrix Operations

MODEL DEFINITION, AND SPECIFICATION ANALYSIS

As blocks of equations became available, they were integrated into the model. This task also was performed by a team of economists. Among other problems, the "language" employed to define the model for simulation differs from that used to estimate equations. Recourse to a single team to make the transfer is not necessary; but for CANDIDE, it allowed us to exploit the benefits of specialization in that this small group quickly became familiar with the model "language" and with the steps necessary to ensure that the transfer was properly accomplished. This was deemed to have been accomplished when the simulation of a block, using actual values on the right-hand side of each equation, yielded the same results as were produced during estimation. The team that performed this task was composed almost entirely of economists not involved in the original estimation work. Perhaps the greatest value produced by this separate, single-team approach lay in the fact that the transfer served to check, at this early stage in the model's development, on the clarity of the documentation provided by the estimating teams. Two programs -- DATAPREP and MODGEN -- support this integration stage.

Functions of DATAPREP

The DATAPREP program performs three tasks. Given a MNEMONIC table and a file of the coefficients, it sets the dimensions of the arrays used in the simulation program and constructs sets of pointers that permit the translation of symbolic variable names into absolute addresses within the arrays used in the simulation program. The program also prepares an output file that provides the values of the exogenous variables that are needed by the simulation program. For simulation over all or part of the sample period, and to start any forecast, that output file also will include values for the endogenous variables. Data

generally are provided to this program by the DATABANK-structured master tape, but may also be entered from fixed-length record files. This phase of the system is necessary, because data on the master tape are organized sequentially by economic time series, and a transposition of the data is necessary since the simulation program expects its data by time period. The DATAPREP program also has features designed to support the previous stage of a model's development. It may be used to check the MNEMONIC table for missing and duplicate equations; to provide a listing of the MNEMONIC table; to check for duplicate sets of coefficients; to provide a listing, by block and equation, of the coefficients; to provide a listing of the variables that are in the MNEMONIC table but not on the master tape; and to provide a compact listing of the values of the endogenous and/or exogenous variables.

This program presently is implemented only under OS/360 and CP-67/CMS. The size of the program is mainly a function of the number of variables in the model and of the number of periods that are to be simulated. For the almost 2,000-variable CANDIDE project, and allowing for up to 24 periods of data preparation, the program runs in about 280K under OS/360. Obviously, the size of the program could be reduced drastically by using direct-access techniques to make periodic dumps of the data matrix that is to be transposed, but our estimate is that the increased use of system resources and prolonged execution time would sharply increase run costs.

Functions of MODGEN

As in most simulation systems that process econometric models, the definition of a model in this system uses a "language" that is quasi-algebraic in form. Since declarations, or the characteristics of each variable are defined in the MNEMONIC table, the description of equations in each block is particularly straightforward in appearance and provides an immediate means of documenting the model. (To be complete, such documentation requires a listing of the coefficient values, which may be obtained from the DATAPREP program, and a listing of regression statistics, which may be obtained from the MASSAGER program.) The program --MODGEN-- that processes this code translates it into FORTRAN statements describing all variables in terms of one of five vectors: E (endogenous variables in period t), X (exogenous variables in period t), EL (endogenous variables lagged), XL (exogenous variables lagged), and S (the calculated variables of period t). It is not necessary to code the values of the regression coefficients into each equation. They are coded as B's and are represented in the FORTRAN code as elements in a vector (B). In simulation, their values are provided by the file produced by the DATAPREP program. This scheme permits the re-estimation of coefficient values with no change to the model description. Finally, constant adjustments to an equation are expressed as elements of a vector (C). All equations must be normalized.

Equations submitted to the system may contain legal mnemonics, arithmetic operators, the assignment sign (=), parentheses, arithmetic function names, MODGEN operators, and coefficients, which may be expressed as B's or real numbers. Mnemonics that are global to the model are expressed as a 2-6 character alphanumeric string whose first character is alphabetic and that is defined in the MNEMONIC table. The arithmetic operators are +, -, *, /, and **; the FORTRAN symbols for addition, subtraction, multiplication, division, and exponentiation, respectively. Arithmetic function names may be either FORTRAN library function names or user-defined FORTRAN subprograms available in the simulation module. Arithmetic operators and parentheses follow the syntax and precedence defined for them in the FORTRAN language.

The simulation program stores all coefficients sequentially, ordered by block and equation number. Within the FORTRAN subroutine generated by MODGEN, the address of a coefficient in the B array is referenced, starting with B(1) for the first coefficient in the first behavioral equation in the block as declared in the MNEMONIC table. Given the pointers provided by DATAPREP, MODGEN determines the index of the first B for each equation. This index is set as the initial value of a counter that is incremented at each occurrence of an unsubscripted B within the equation. Assuming that the equation processed has been identified in the MNEMONIC table as the fourth, and that equations one through three required 5 coefficients, an input code of:

```
ABC = B+B*TIME
```

would produce (regardless of the block) FORTRAN code of:

```
S(4) = B(6)+B(7)*X(1)
```

Coefficients may also be coded in subscripted form, which may be used to handle multiple references to the same coefficients or to accommodate a different ordering in the original storage of the coefficients.

Through MODGEN operators, lags, distributed lags, first differences, relative change, percentage change, and summation can be introduced as primitive operators. Lags and distributed lags are signalled by the use of the delimiters < and >, or [and], that bound signed or unsigned (if 0)

integers describing the length of the lag. For simple lags, GNP[-1], which represents the endogenous variable GNP(t-1), would be translated to an address in the EL array. The notation is available for lagged expressions. If expr is an arithmetic expression (of potentially lagged) variables and n denotes the length of the lag, the expr[-n] denotes a lag of n periods on each mnemonic variable in the expression. For example:

```
B*(EN1-EN2)[-1]
```

is expanded to:

```
B*(EN1[-1]-EN2[-1])
```

before it is translated; and

```
B*(EN1-EN2[-1])[-1]
```

is expanded to:

```
B*(EN1[-1]-EN2[-2])
```

before it is translated. Coefficients are unaffected.

The distributed lag of an expression expr from period t-n back through t-m is represented by:

```
expr[-n,-m].
```

If the coefficient occurs within the expression, then

```
(B*EN1/EN2)[0,-2]
```

is expanded to:

```
(B(index)*EN1/EN2)
+(B(index+1)*EN1/EN2)[-1]
+(B(index+2)*EN1/EN2)[-2]
```

which then is expanded by the simple lag operator before it is translated.

The difference operator, which is denoted by the symbol D, is performed only for first differences. In general, if expr is an expression, D(expr) is expanded to (expr-expr[-1]) before translation occurs. Higher order differences may be coded by using nested D operators; e.g., D(D(expr)). The relative change operator, which is denoted by the symbol Q, and appears as Q(expr) is expanded to ((expr-(expr)[-1])/(expr)[-1]). The percentage change operator, which is denoted by the symbol P, and appears as P(expr) is equivalent to Q(expr)*100. Finally, the sum operator (S) permits summation of an indexable expression. The expression:

$$\sum_{i=m}^{n} expr(i)$$

can be coded as:

```
S(i= m,n:expr(i))
```

Specifically, the expression:

$$EN1 = \sum_{i=1}^{9} (.75)^{i} EN2_{i-1}$$

would be coded as:

```
EN1=S(I=1,9:(.75**I)*EN2[I-1])
```

Logical operations can also be placed into a block of code for purposes of introducing "rules" into the model, or for generating matrix operations. Two card types are provided. The F or FORTRAN card is processed with no mnemonic substitution or operator expansion. This card type is used to introduce local variables into the program, and to insert GOTO, CONTINUE, and other FORTRAN control statements. The S or SUBSTITUTION card allows the introduction of FORTRAN statements referring to mnemonics in the model. Thus, decisions based on the value of a variable can be introduced. By enclosing a mnemonic in apostrophes, the index of that variable can be generated, and in conjunction with F cards, is used in generating DO loops whose index is to range over a contiguous set of variables. Finally, C or COMMENT cards can be used to provide for titling of equations. A representative set of the code used in the CANDIDE project is presented as Fig. 4.

Figure 4

Sample of MODGEN Code

```
B 21 BALANCE OF PAYMENTS BLOCK
C    DIRECT INVEST. IN CANADA BY U.S.
E 1  DIPUS=B+B*CINET+(B*RINDB)[-1,-3]
2          +(B*TR)[-1,-3]
C    DIRECT INVEST. IN CANADA BY ROW
E 2  DIROW=B+(B*RINDB)[0,-3]
2          +(B*RLUK)[-1,-3]
C    DIRECT INVEST. BY CANADA IN U.S.
E 3  DIRUS=B+B*QSALE+(B*RINDB)[0,-3]
2          +(B*TR)[0,-3]
C    DIRECT INVEST. BY CANADA IN ROW
E 4  DIABRO=B+B*CINET+(B*RLUK)[-1,-3]
2           +B*(DIABRO[-1]
3           -(B(1)+B(2)*CINET[-1]
4           +S(I=2,4:B(I+1)*RLUK[-I])))
C    TRADE IN OUTSTANDING SECURITIES +
C         NEW ISSUES OF CDN. SECURITIES
C         SOLD IN U.S., BETWEEN CANADA
C         AND U.S.
E 5  TBSUS=B+B*D(GNPC)+B*D(GNPUS)
2          +(B*RINDB)[0,-3]
B 23 RECURSIVE FUNCTIONS
C    LONG TERM U.K. INTEREST RATE
S    IF(TIME.GT.69.)GOTO 100
C    EXOGENIZE THRU 1969
E 11 RLUK=RLUK
F    GOTO 101
F100 CONTINUE
E 11 RLUK=1.2*TR
F101 CONTINUE
C    U.K. TREASURY BILL RATE
S    IF(TIME.GT.69.)GOTO 25
C    EXOGENIZE THRU 1969
E 12 RGTBUK=RGTBUK
F    GOTO 26
F25  CONTINUE
E 12 RGTBUK=1.3*RGTBUS
F26  CONTINUE
C    U.S. INDUSTRIAL PRODUCTION,1961=100.
E 1  USIP=B+B*IN04+B*IN11+B*IN22+B*IN30
C    OECD INDUSTRIAL PRODUCTION
E 2  OECDIP=(66.66*USIP+33.34*OIP)/100.0
B 25 GROSS OUTPUTS
F    REAL*8 Y
F    M=1
F    DO 11 I=1,51
F    Y=0.0
F    DO 10 J=1,84
S    Y=Y+B(M)*F('GRANCR'+J-1)
F10  M=M+1
F11  S(I)=Y
```

For the CANDIDE project, MODGEN was implemented to allow for the processing of a 2,000 variable model. Under OS/360, the program runs in a region size of almost 140K, exclusive of buffer area. Blocks were processed individually; and the resultant object code was stored as separate files. This allowed us to introduce variants of "rules" for simulation at the block level with each object deck stored as a separate file. It also eased the file management problem as equations in several of the blocks were respecified.

SIMULATION, ANALYSIS OF RESULTS, AND REPORT GENERATION

In the CANDIDE project, the simulation software was used first by the small team of economists who defined the model in terms of the MODGEN "language" to ensure that the specification of equations in the model code was equivalent to that used in the estimation process. This, therefore, represented part of the second research stage, or that concerned with the definition of the model. This "residual check" is achieved by using actual values for all variables on the right-hand side of each equation in the solution of the model. If the coding of equations in the estimation process and that defining equations in the model are the same, if all identities are properly coded, and if no data changes have been made to the master tape between the estimation phase and the simulation stage, then the residuals of all identities in the model should equal zero and those of all stochastic equations should equal the residuals generated by the estimation process. In fact, residual checks on individual blocks for the CANDIDE model were performed as they became available and little calendar time was lost in this stage of the model's development. Initially this comparison of estimation and residual results was done by eye; more recently, additional software has been used to compare the results of the two runs and report only those series that diverge from each other by more than some user-specified degree of precision.

After blocks, or sets of blocks, complete the "residual check" they then are simulated to review the dynamic properties of those subsets. At this point, the direct intervention of the project manager is critical. In this instance, the user-specified blocks are solved simultaneously, all others are treated as exogenous inputs to the system. This phase enabled us to examine particular sectors of the economy with respect to their past performance and to their theoretical reasonableness. On the basis of this, several equations were respecified. The process that had been followed for solving subsets of the model -- single-period and multi-period analysis -- was adhered to for analysing the full system. The ability to alter convergence criteria at the equation level proved important because several identities that had been introduced into the model for reporting purposes, that were recursive, and that had small calculated values prevented solution of the system. Exception reports generally enabled us to quickly identify these equations. For equations that require re-estimation, ammendments to the model have proved to be reasonably simple. The model code for that block is reprocessed, and values for the coefficients are overridden at solution time. If new variables are required by the model, they are introduced as symbolic variables that have been defined for this purpose at the beginning of the model's construction. It is not necessary to reprocess all of the model's definition each time a change in specification occurs. Since the speed of convergence is affected by the order of the equations, the user can control, at the block level, the order for the solution of blocks at each iteration. The solution of a block, or blocks, can be repeated within one iteration. In the CANDIDE model, for example, several major aggregates that are frequently used on the right-hand side of the wage mechanism are solved prior to that block; they in turn, depend on wages and are solved again to pick up the information gained; finally they serve as important inputs to the price formation blocks. This reduced the number of iterations required for convergence in each period by about 25 per cent, although some additional time was used in each iteration.

Functions of SOLUTION

A basic set of controlling software --SOLUTION-- together with the executable code describing the model that is generated by MODGEN constitutes the simulation software. The basic features of the software include:

1. The ability to make residual, single-period, and multi-period runs.
2. Any subset (controlled at the block level) of the model can be processed.
3. Exception reporting is provided to aid in analysing the convergence properties of the model.
4. Convergence criteria and damping factors can be introduced at the equation level.
5. New assumptions (changes to exogenous variables) can be introduced without affecting the master file.
6. Changes to the values of coefficients can be introduced.

7. A small number of control cards (as few as two) is required to cause execution. And

8. All the values required to simulate one period only are held in core thereby minimizing core requirements and making the program size independent of the number of periods simulated.

The solution of systems of nonlinear, simultaneous equations requires substantial judgement and understanding of the properties of the actual system of equations to be solved. Many sophisticated algorithms have been developed to cope with multiple solutions, local options, and saddlepoints. Fortunately, most econometric models are not difficult to solve. We use the Gauss-Seidel procedure because of its ease in programming, relatively low core requirements, and modest computer costs. As with other iterative procedures, the error build-up from the limited word size of the machine is restricted to the error accumulation in one iteration. With this method, the ordering of equations and the choice of convergence and damping criteria are important. Features are incorporated in the program so that the solution can be tailored for a particular econometric model. No doubt, as new types of nonlinearities are introduced into econometric models, other solution algorithms will be necessary. Users also are encouraged to attempt solutions with differing starting values to assure that the system converges to the same solution set. To date, we have not encountered models with multiple solutions in the same "neighborhood", although a discussion of this problem appears in [6].

The MODGEN program produces U.S.A. Standard FORTRAN code and the basic simulation program is written to the same specifications. Presently the system has been used only under CP-67/CMS and OS/360. For the CANDIDE project, the entire model requires approximately 350K of core, exclusive of buffer areas. Using overlay techniques, we have managed to reduce the core requirements to less than 150K, but only at the cost of greatly increased execution time.

Report Generation

The output of the simulation program optionally is directed both to the printer and to a sequential file. Output is ordered by period. To facilitate analysis of the model, the latter file can be transposed into time series and is reconstructed as a DATABANK-structured file. This enabled us, in the CANDIDE project, to quickly disseminate the results of simulations to a large number of economic specialists. For purposes of a full report, the DATABANK-structured file was processed by the DATAPREP program. Individual analysts were able to focus on specific variables by using MASSAGER to perform residual and multiplier analyses, plots, and reports for management. This procedure prevented any bottlenecks in analysis because the economists were familiar with the manipulative system from the estimation phase.

CROSS-REFERENCES AND OTHER ESSENTIALS

Data revisions, particularly for more recent periods, are a dismal fact of life in economic research. And the change of one data point in the sample period requires the re-estimation of all equations that use that data point on the right-hand side of the equation. A cross-reference map of all variables, therefore, has been incorporated into the MODGEN program as an optional by-product. The map also is used to "clean up" the MNEMONIC table, i.e., to reduce the number of lag periods available for any variable.

The records produced by MODGEN for the cross-reference map also are used by a small program that implements Steward's algorithm for optimally ordering equations [7]. In the CANDIDE project, this was implemented to order blocks, and equations within blocks. Re-ordering equations within blocks presents no important problem, because this requires only repositioning the equations within the block. To reorder all of the variables in the model, however, would mean a complete reprocessing of the model beginning with the use of DATAPREP and reconstruction of the MNEMONIC file. We have not done so because of the still volatile nature of the model's specification. For smaller models implementation of the algorithm would be a minor task.

SUMMARY AND CONCLUSIONS

It can be argued that the following modifications and/or extensions to the software should be made. Presently, the user is required to learn two "languages"; one for the estimation phase and another for model definition. To define both of them in the quasi-algebraic form used as input to MODGEN would require the use of direct access techniques for maintenance of the data base and the development of a system to manage an equation bank. Because the former would impose constraints on implementation and the latter would impose a far more complicated system of control on the researchers as well as require frequent updating of tables, we are not fully convinced of their merit. The system is designed for large memory machines, although the amount of memory

required is adjustable (about 10 changes to the source code).

The software has the following merits. It is fully operational for a very large econometric model. The system assists in the tasks of data management, estimation, solution, and analysis. There is substantial error-checking between all parts of the process and various documentation aids produced. Generally, little training of economists is required to enable them to develop a model from the beginning. All programs are fully documented. The CANDIDE model was constructed and is maintained exclusively by economic researchers. Approximately 10 hours in the classroom were required to train the team to use the full set of simulation software. Another 4-8 hours of classroom instruction has been required to teach the MASSAGER program to those not previously familiar with the package. Given our view that the econometric research process usually is staged, the system is designed as a set of distinct modules. Although the software is used on OS/360 and under CP-67/CMS, not all programs as implemented are machine-dependent. In particular, the estimation phase, simulation of a developed model, and analysis of the model are essentially implementation-independent.

REFERENCES

1. Economic Council of Canada, An Overview of CANDIDE Model 1.0, (forthcoming), Ottawa

2. Bank of Canada, User's Guide: DATABANK System, 1970, Ottawa

3. McCracken, M. C., "A Computer System for Econometric Research", Social Science Information, Vol. VI, No. 5, pp. 151-158

4. Statistics Canada, Canadian Socio-Economic Information Management System, 1971, Ottawa

5. McCracken, M.C., MASSAGER User's Manual, 1971, Ottawa

6. Friedman, B.M., "Econometric Simulation Difficulties: An Illustration", Review of Economics and Statistics, Vol. LIII, No. 4, Nov. 1971, pp. 381-384

7. Steward, D., "Partitioning and Tearing Systems of Equations", SIAM Journal, Series B, Vol. 2, No. 2, 1965, pp. 345-365