NOT FOR QUOTATION
WITHOUT THE PERMISSION
OF THE AUTHORS

Computation of Multi-state Models Using GAUSS, A Matrix Based Programming Language

Andrew Foster
Nathan Keyfitz

December 1986
WP-86-75

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

A great variety of software exists to help demographers to perform their calculations. What excuse, then, is there for the further set of software that this working paper presents?

The answer lies in the exceptional qualities of GAUSS, a programming language developed for micro computers in the last year or two by Lee Edlefson and his associates. It handles numbers as arrays. So do other languages, for instance APL. But GAUSS is simpler, faster, and more flexible than these competitors. It uses the standard keyboard, addresses a memory of 400k or more, and is easy to learn.

GAUSS is also very fast, taking full advantage of the coprocessor of an XT or an AT. Speed can be important in multi-state calculations, and in simulations. GAUSS, for instance, creates an 80 x 80 matrix of random elements, inverts it, and verifies the inversion to 16 decimal places, all in less than 1 minute.

Beyond these advantages is the brevity of its programs. The entire multistate life table can be written on 25 lines; its essence appears on the bottom half of the page containing Appendix I. The usual life table for mortality is a special case to which the program is applicable; it is equally applicable to single years of age up to 90. We have available diskette versions of the programs, but they are short enough that they can be keyed in not much more than the time required to write a letter asking for the diskette.

On a personal note: after programming in FORTRAN and in BASIC for most of 25 years, I have now turned to GAUSS for all of my serious work. One can learn enough to get started in it within a day; to learn to use it well and flexibly of course takes longer. I am grateful to Andrew Foster for instructions on GAUSS, and for the elegance that the programs of this working paper exhibit.

Nathan Keyfitz
Leader
Population Program

Abstract

In this paper we present a series of programs in GAUSS. Three areas of application are shown: (1) construction of a multi-state life table, (2) projection of a multi-state population, and (3) the calculation of variances and covariances for a projected population. Illustrative results are presented for marital status data by age for Canadian women in 1981. Program code and the appropriate data are provided so that the reader may reproduce our results and develop a working knowledge of multi-state analysis. The programs are readily applicable to data for other countries, and to multi-region as well as to marital status analysis.

Contents

Computation of Multi-state Models Using GAUSS, A Matrix Based Programming Language

Andrew Foster, Nathan Keyfitz***

Introduction

Despite the widespread recognition of the relevance of the multi-state model in the analysis of demographic processes, the application of multi-state methods has been limited by the size and complexity of computer programs seemingly required for such analysis. Researchers interested in gaining a practical understanding of these methods have often been faced with a need to invest a substantial amount of time to develop and modify the appropriate software. In this paper we present new and simple programs.

The programs are all written in GAUSS version 1.40, a matrix based language for the IBM or IBM-compatible personal computer. The advantage of a matrix based language is seen in the code presented here--programs are short and closely related to the analytic expressions on which they are based. These programs could be adapted to other matrix based languages such as APL or PROC MATRIX in SAS.

The programs are designed to take advantage of the PROC command, which allows one in effect to design a specialized language by loading into memory previously compiled procedures. A variety of different applications programs may then be developed which call the same general procedures.

Some generality and efficiency in the procedures has been sacrificed in order to improve readability. We have avoided tricks that produce desired results quickly but in obscure ways. In addition, we have assumed that the programs are to be applied to problems of relatively low dimension. GAUSS is at its best when processing matrices, but the maximum matrix size is at present 64K or about 8000 elements. This means, for example, that a population projection by a Leslie matrix that pre-multiplies a vector of single-year age counts can go only up to age 90. An alternative method can be constructed that does not require the full projection

*Andrew Foster, Graduate Group in Demography, University of California, 2234 Piedmont Avenue, Berkeley, CA 94720, USA

**Nathan Keyfitz, Leader, Population Program, IIASA, A-2361 Laxenburg, Austria

matrix which is rather sparse. The simplicity of the code, however, would be sacrificed.

Finally, an attempt has been made to permit even first time users of GAUSS to reproduce some of the published results—one need only enter the programs (including the data formatting program) in ASCII files with the specified names, and then run the program "msstart.can" from GAUSS. At that point any of the applications procedures can be run. See Appendix V for the full set of commands.

The multi-state life table

The program which produces a multi-state life table relies on the development in Rogers (1975), so readers who wish a more detailed discussion are referred there; in this paper we provide a short review.

The starting point for a multi-state analysis is the construction of an appropriate schedule of transition intensity matrices. If there are k states, then for each age x we would have a $k \times k$ matrix $\mu(x)$. The $ij^{th}(i <> j)$ element of $\mu(x)$ is minus the rate of transition from the j^{th} to the i^{th} state at exact age x. The ii^{th} element is the rate of movement out of state i and is thus the mortality rate in state i plus the sum of transition rates out of state i to the other $m-1$ states at that age. Note that in this formulation death is not included explicitly as one of the states. An alternative formulation would include death as an absorbing state, thus increasing the dimension of the matrices $\mu(x)$ by one.

As is the case in the simple life table, the multi-state life table does not work directly with the underlying transition intensities (supposed continuous with age) but with an approximation in finite age groups and assuming a constant transition intensity matrix M_x within age group x. The multi-state life table procedure, then, takes as input a matrix VCM which is a vertical concatenation (v.c.) of these discrete transition intensities M_x.

A further approximation is made when we construct the matrix p_x, the proportion surviving by state from the beginning of each age group, x, to the beginning of the next age group, $x+h$,

$$p_x = (I + hM_x/2)^{-1}(I - hM_x/2)$$

where h is the size of the age group, taken in the examples considered here to be 5 years. An exact expression would require a spectral decomposition of the matrix $-h\ M_x$, or some other complication. While GAUSS provides a mechanism for that decomposition in the form of a general eigenvalue procedure, as long as the transition intensities and age intervals are small the increase in computation time is difficult to justify.

The calculations used for other life table measures are straightforward generalizations of those used for constructing simple life tables. Indeed, since the GAUSS language operates on matrices the individual lines look rather like the lines that might appear in a PASCAL program, for example, that is used for the single state case. The matrices l_x, survivorship from the beginning to age x, are constructed by starting with an l_0 (if the radix age is 0) which is the identity matrix instead of the usual scalar 1 and multiplying by successive p_x matrices. The person-years in the age group ${}_hL_x$ are estimated by a linear approximation within each age group. The T_x are the sum of the ${}_hL_y$ for $y \geq x$. Finally, the state dependent life expectancy e_x whose i,j^{th} element is the expected number of years to be spent in state i for an individual currently in state j, is $T_x \times l_x^{-1}$. The matrices VCL, VCLL, VCT, and VCE are the vertical concatinations of the $l_x, {}_hL_x$, T_x, and e_x matrices, respectively.

One version of the procedure appears in Appendix I. One might readily use a cubic for integration to estimate person years and provide a more realistic closing of the life table. Whatever the methods for integration and closing, given the matrix of discrete transition intensities and a radix age the resulting multi-state life table will not be very different from ours.

Appendix I also contains an applications program for the durability of marriage in Canada. In order to facilitate the application it is helpful to create another procedure, MSMASK. MSMASK allows one to conveniently perform counterfactual experiments by setting various elements of the transition matrix to zero using a filter or "mask". Multiplication of the data matrix by MSMASK is element-by-element, not the usual row-by-column multiplication. One can give interesting substantive interpretations to the results obtained when certain masks are applied. For example, the expected time in the married state for a married individual in the absence of returns to marriage from divorce or widowhood will be the expected time spent in the **current** marriage. In these programs we consider four types of masks in addition to "Base", the mask of ones which has no effect: "NoReMa", "NoDiv", "NoWid", and "NoDea" to abstract from remarriage, divorce, widowhood and death, respectively.

Table 1 provides an illustration of this example. For two ages (20 and 65 with radix 20) we present l_x and e_x for a base run as well as after abstracting from remarriage. Consider i and $j \in \{S,M,W,D\}$ (e.g. single, married, widowed, and divorced), where j indexes columns or "originating status" and i indexes rows or "end status". If x is age in Table 1, then then $l_{ij}(x)$ may be interpreted as the probability that an individual originating in state j at age 20 will be in state i at age x; $e_{ij}(x)$ is the expected number of years an individual in state j at age x will spend in state i before dying. Thus, a woman who is single at age 20 may expect 38.7 years of married life. If we do not allow remarriage, however, she can expect only 32.2 years of married life. If mortality rates were the same in the three ever-married states, then the average length of time spent in a first marriage for a single woman of age 20 is 32.2 years.

Multi-state projection

One extension of the multi-state life table is the multi-state projection which uses a block-Leslie matrix, where the term "block" refers to the fact that each non-zero element in the usual Leslie matrix is taken to be a $k \times k$ matrix when there are k states. We constructed such a block-Leslie matrix for Canadian females (Feeney, 1970) and use the matrix in the standard fashion to provide projections based on rates and counts from 1981. Since data on fertility by age and state were not available, we imputed all fertility to the married population.

As with the multi-state life table program, the program MSLESLIE.PRC is the same as for the standard Leslie matrix except that the basic argument is the transition matrix rather than the scalar rate. The subdiagonal blocks are calculated from the multi-state life table using the expression $hL_{x+h}*inv({}_hL_x)$ and the fertility blocks are similarly expressed using a matrix version of the standard formula (see, for example, Keyfitz 1985, p. 206).

Projections were generated in five year intervals for the 50 years following 1981. The projected vector is large (72 elements) and we have taken as examples three summary measures: total population, number 65+, and number 65+ and not married (Table 2), all for Canadian females. Declining numbers are due to our using fixed 1980-82 birth rates that are considerably below replacement. No account is taken of immigration. Figure 1 shows the proportion of 65+ women who are not married when four alternative masks are imposed on the transition rates.

Variances in the multi-state model

The multi-state analysis considered up to this point, like traditional life table analysis, is based entirely on deterministic theory. While this approach may be justified on a number of grounds, there are occasions when a stochastic analysis might be considered more appropriate. In particular, we may be interested in a stochastic model in cases where a) the population size is small or b) the transition intensities are thought to be random.

It may be helpful at this point to describe briefly what is meant by a stochastic population model. Consider the matrix p_x in the first section. It specifies the probability that an individual at age x and state j will be in state i after one time period, given the underlying Markovian transition rates. When we construct the multi-state life table, however, we treat p_x as a matrix of proportions.

This assumption, that probabilities may be treated as proportions, is usually invoked in the construction of the single state life table, and is justified by assuming that the population is sufficiently large so that random deviations from the expected values, at least on a percentage basis, may be ignored. If we use a series of transition rates to make predictions about the behavior of small populations subject to certain fixed rates, however, we should recognize that population size and its distribution will be random variables with nonzero variance and interpret the projected counts as expected values of these random variables. The idea of a stochastic population model is to provide a convenient mechanism for gauging the size of the resulting variances. A general class of stochastic population models are those which may be represented by Galton-Watson branching process. The most important aspect of this class of models is the assumption of independence among individuals.

Pollard (1966) has provided a convenient method of computing variances and covariances of a general Galton-Watson branching process. Of particular interest is the fact that an extended Leslie matrix that projects second and higher moments of the population age distribution may be constructed using the direct or Kronekar product. This result also extends to the multi-state Leslie matrix, even if the rates themselves are random (Pollard 1983). It is thus possible to construct a rather simple program, again using GAUSS, that will provide estimates of variance for a large class of multi-state projections.

The method developed by Pollard requires the construction of four matrices, T, F, B, and M, whose matrix product is the required extended projection matrix. The procedure MSVARI.PRC calculates this extended projection matrix given the same transition intensity matrix discussed above. While an initial version of the procedure actually created these four matrices and then multiplied them, it turned out to be more convenient to partition the four matrices and carry out the multiplication partition by partition. The reader should have little trouble following the code appearing in Appendix II; however, it may be helpful to briefly discuss what each matrix is designed to do.

Matrices F and M transform moments about the origin to falling factorial moments and back again. They are lower triangular, and at least for the two moment case, easy to construct. The matrix B transforms factorial moments of the set of initial states into factorial moments of a set of intermediary random variables, where each intermediate random variable corresponds to one type of transition that will be observed in a given period. The transition to death and the identity transition must both be included explicitly so that the sum of transition probabilities associated with each state is one. If P is the block diagonal matrix with the i^{th} block being the vector of transition probabilities associated with the i^{th} state, then we can write B simply as a block diagonal matrix with blocks P and $P.*.P$ (.*. is used to denote the Kronekar or direct product).

The number of individuals in each state at time $t+1$ will be a linear transformation, denoted by the matrix Q, of the intermediate random variables which in turn are derived from the number of individuals in each state at time t. If only transitions among k explicit states are possible then there will be up to k possible transitions for each state (including the identity transition). The number of intermediate random variables, and thus the number of columns of Q, will be no larger than $k*k$. In the case where all transitions are included the Q matrix will consist of k k-dimensional identity matrices concatenated horizontally. If one state (e.g. death) is included implicitly, then each pair of identity matrices will be separated by a column of zeros. In the Leslie projection model, as a result of the presence of birth processes, Q will be more complicated. In either case, the matrix T relating the first two moments of the intermediate and final random variables will be block diagonal with blocks Q and $Q.*.Q$.

Since all of the above transforms are linear and since the matrices T, M, and F are fixed by the set of possible transitions, generalizations to changing and even independent random transition rates is straightforward. In the iid random case, for example, we simply replace B with its expectation, which requires that we specify $E(P.*.P)$ and $E(P)$.

The procedure MSVARI constructs the required matrix product given the arguments P, Q, and optionally $E(P.*.P)$. Table 3 provides a simple example based on a Leslie projection matrix. The procedure MSPQT generates the requisite P and Q values from a discrete time Markov transition matrix. An applications program then uses these procedure to construct a multi-state life table with variances from the Canadian marital status data appears with the other procedures in Appendix III; a few extracted results may be found in Table 4.

In the simple case with a fixed set of age specific rates and a fixed initial population there is no need to rely on the above approach because the variance-covariance matrix at age x may be calculated directly from the l_x matrix using the multinomial distribution. The branching process is needed for a model that allows the rates to be stochastic or includes renewal by birth. We are unable to estimate variances using the (18 × 4) block-Leslie projection matrix described in the preceding section because the projection matrix would be 72+72 × 72 = 5256 × 5256, which is too large to be handled efficiently, hence the smaller matrix of our example.

The assumption that individuals are independently subject to the observed transition probabilities results in variances that are too small in application to large populations. The Pollard method initiates but by no means exhausts the study of variance in a branching process.

Figure 1
Projected proportion of 65+ Canadian women who are not married
(after abstracting from specified transitions).

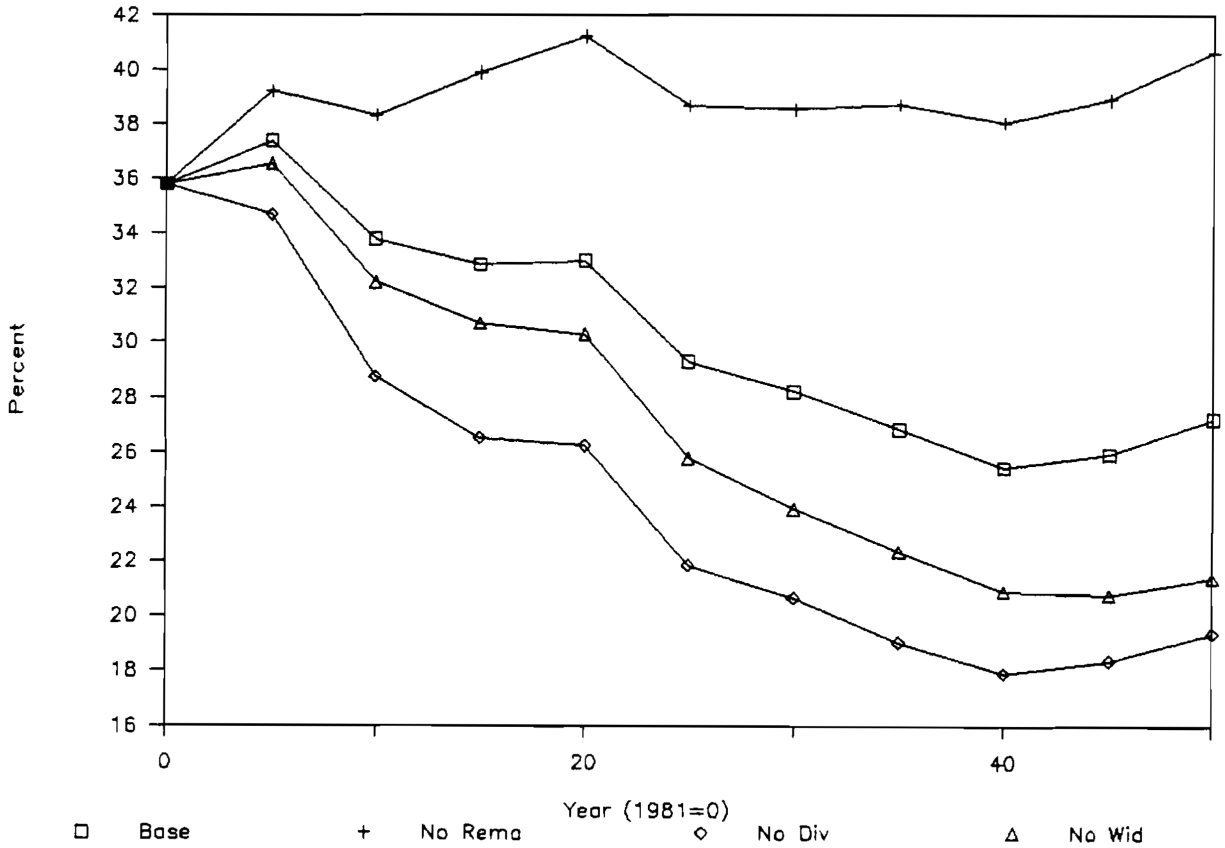

Table 1
Extract from life table by age and marital status.
Canadian Females, 1981

Table 1a: No Mask

		l_x				e_x			
	Tran.	Originating Status				Originating Status			
Age	Status	S	M	W	D	S	M	W	D
20	S	1.000	0.000	0.000	0.000	11.472	0.000	0.000	0.000
	M	0.000	1.000	0.000	0.000	38.724	48.997	34.875	45.366
	W	0.000	0.000	1.000	0.000	3.674	4.153	18.483	4.123
	D	0.000	0.000	0.000	1.000	4.440	5.560	3.902	9.046
65	S	0.092	0.000	0.000	0.000	16.633	0.000	0.000	0.000
	M	0.636	0.715	0.605	0.711	0.122	13.654	0.419	0.785
	W	0.057	0.065	0.171	0.065	0.033	3.391	16.363	0.211
	D	0.085	0.100	0.077	0.101	0.001	0.153	0.003	15.340

Table 1b: No Remarriage Mask

		l_x				e_x			
	Tran.	Originating Status				Originating Status			
Age	Status	S	M	W	D	S	M	W	D
20	S	1.000	0.000	0.000	0.000	11.472	0.000	0.000	0.000
	M	0.000	1.000	0.000	0.000	32.229	38.966	0.000	0.000
	W	0.000	0.000	1.000	0.000	3.059	3.200	55.273	0.000
	D	0.000	0.000	0.000	1.000	11.171	15.963	0.000	56.039
65	S	0.092	0.000	0.000	0.000	16.633	0.000	0.000	0.000
	M	0.486	0.499	0.000	0.000	0.122	13.620	0.000	0.000
	W	0.051	0.054	0.806	0.000	0.033	3.418	16.774	0.000
	D	0.234	0.315	0.000	0.824	0.001	0.159	0.000	16.284

Table 2
Projected Female Population (1000s)
Base Run

Year	Total	65+	65+ and Not Married
1981	12275.50	2855.10	1022.00
1986	12311.11	3035.16	1133.94
1991	12326.84	3137.76	1059.75
1996	12235.41	3162.78	1038.82
2001	12038.86	3253.52	1073.31
2006	11750.09	3016.84	883.53
2011	11355.36	3065.62	863.97
2016	10862.41	3147.13	843.68
2021	10301.30	3172.99	806.33
2026	9666.28	2807.52	727.36
2031	8933.05	2563.33	697.79

Table 3
Illustration of Stochastic Leslie Matrix

Table 3a
Intermediate transition probability matrix

Transition Type	Age group 1	2	3
1	1	0	0
2	0	0.1667	0
3	0	0.1667	0
4	0	0.3333	0
5	0	0	0.6667
6	0	0	0.3333

Table 3b
Intermediate transition counts matrix

Age Group	Transition Type 1	2	3	4	5	6
1	0	1	1	0	1	2
2	1	0	0	0	0	0
3	0	1	0	1	0	0

Table 3c
Stochastic Leslie Projection Matrix

0.00	0.33	1.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.22	0.22	0.00	0.00	0.00	0.00	0.11	0.44	0.00	0.44	1.78
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00	1.33	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.67	0.00
0.00	0.00	0.00	0.00	0.33	1.33	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.67	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00

Note: The original 3×3 Leslie matrix appears in the upper left corner of the stochastic matrix.

Table 4

Expectations and variances of l_{65} from a condensed stochastic life table by age and marital status. Canadian Females, 1981. Radix age 20.

		1_x			
		Originating status			
Trans. Status	Moment	Single	Married	Widowed	Divorced
Single	M	0.178	0	0	0
	V	0.146	0	0	0
Married	M	0.485	0.629	0.493	0.618
	V	0.250	0.233	0.250	0.236
Widowed	M	0.075	0.099	0.242	0.096
	V	0.069	0.089	0.184	0.087
Divorced	M	0.065	0.092	0.062	0.101
	V	0.060	0.083	0.058	0.091

REFERENCES

Edlefson, L.E. and Jones, S.D. (1986) *GAUSS*. Copyright software.

Feeney, G. (1970) Stable age by region distributions. *Demography* 7:351-348.

Keyfitz, N. (1985) *Applied Mathematical Demography*. New York: Springer-Verlag.

Pollard, J.H. (1966) On the use of the direct matrix product in analyzing certain stochastic population models. *Biometrika* 53(3):397-414.

Pollard, J.H. (1973) *Mathematical Models for the Growth of Human Populations*. Cambridge, England: Cambridge University Press.

Rogers, A. (1975) *Introduction to Multiregional Mathematical Demography*. New York: John Wiley & Sons.

Schoen, R. (1975) Constructing increment-decrement life tables. *Demography* 12:313-324.

Appendix I

```
@ Program: MSLIFE.PRC @
@ This program produces a procedure which constructs a multi-state
life table from the argument "vcm", a vertical concatenation of
age specific transition rates. It is assumed that there are 5
year age groups. Transitions occuring before the radix age are
ignored.
@
proc mslife(vcm,radage);

  local k,n,m,ages,ix,id,vcp,id,vcl,vcll,vcll2,vct,vce,i,ri0,ril1;
  k=cols(vcm);                      @ k is number of states          @
  vcm=trim(vcm,k*radage/5,0);       @ Discard data below radix age   @
  n=rows(vcm);
  m=n/k;                            @ m is number of age groups      @
  ages=seqa(radage,5,m);            @ Creates vector of ages         @
  ages=vec((ages.*ones(1,k))');     @ Ages vec matched to VCM rows   @
  ix=reshape(seqa(1,1,n),m,k)';     @ ix is matrix used for indexing @
                                    @   VC matrices                  @
  id=eye(k);                        @ id is k dim identity matrix    @
  vcp=vcm; vcl=vcm; vcll=vcm;       @ Matrices set to correct size   @
  VCE=VCM; VCT=VCM;
  clear ril1;

@ Loop over ages to create px, lx, 5Lx @

  i=0;
  do until i==m; i=i+1;
    ri0=ix[.,i];
    vcp[ri0,.]=inv(id+2.5*vcm[ri0,.])*(id-2.5*vcm[ri0,.]);
    if i==1;
      vcl[ri0,.]=id;
    else;
      vcl[ri0,.]=vcp[ril1,.]*vcl[ril1,.];
      vcll[ril1,.]=(vcl[ri0,.]+vcl[ril1,.])*5/2;
      if i==m;
          vcll[ri0,.]=vcl[ri0,.]*5/2;
      endif;
    endif;
    ril1=ri0;
  endo;

@ Loop over ages to create Tx and Ex @

  vcll2=reshape(vcll,m,k*k);
  i=0;
  do until i==m; i=i+1;
    ri0=ix[.,i];
    vct[ri0,.]=reshape(sumc(trim(vcll2,i-1,0)),k,k);
    vce[ri0,.]=vct[ri0,.]*inv(vcl[ri0,.]);
  endo;

@ Return altered matrix, close the procedure, and save to disk@
  retp(ages~vcp~vcl~vcll~vct~vce);
endp;
save mslife;
```

```
@ Program: MSMASK.PRC @
@ This program produces the procedure MSMASK and a few masks of
  interest for the analysis of marital status which allow one to zero
  certain transition probabilities according to a k by k mask of
  ones and zeros; ones indicate that a particular element should
  NOT be altered. A zero along the diagonal will cause the death
  rate in that group to be set to zero. In all cases the diagonal
  elements will be adjusted so that the sums along the columns
  are equal to the death rates.
@
proc msmask(vcm,mask);
  local n,k,m,ix,nmask,ri0,i,mx,dthx,mx;
  n=rows(vcm);
  k=cols(vcm);
  m=n/k;
  ix=reshape(seqa(1,1,n),m,k)';

@ Apply mask and recalculate diagonal elements @

  i=0;
  do until i==m; i=i+1;
    ri0=ix[.,i];
    mx=vcm[ri0,.];
    dthx=sumc(mx).*diag(mask);
    nmask=mask.*(-eye(k)+1);
    mx=nmask.*mx;
    vcm[ri0,.]=mx+eye(k).*(dthx-sumc(mx));
  endo;

@ Return altered matrix, close the procedure, and save to disk@

  retp(vcm);
endp;

@ Each mask sets certain transition probabilities to
  zero by applying a 4 by 4 "mask" to the data. Here we provide
  four possible masks which may be applied to marital status data
  ordered as: single, married, widowed, divorced. The base mask
  will have no effect.
@
Base=ones(4,4);
let NoReMa[4,4]=1 1 1 1
                1 1 0 0
                1 1 1 1
                1 1 1 1 ;
let NoDiv[4,4] =1 1 1 1
                1 1 1 1
                1 1 1 1
                1 0 1 1 ;
let NoWid[4,4] =1 1 1 1
                1 1 1 1
                1 0 1 1
                1 1 1 1 ;
let NoDea[4,4] =0 1 1 1
                1 0 1 1
                1 1 0 1
                1 1 1 0 ;


save msmask,NoDea,NoWid,NoReMa,NoDiv,base;
```

```
@ Program: MARRIAGE.CAN @
@ Multi-state life tables can be created using marital status
  data from Canada. Elements of transition matrices may be
  set to zero using a masking procedure. Choose an appropriate
  data set by changing the two lines following this comment.
@
loadm cf8;
vcm=cf8;

radage=20;
k=4;
n=rows(vcm);
m=n/k;
loadp mslife,msmask;
loadm base,norema,nodiv,nowid,nodea;

@ Certain transition probabilities may be set to zero by applying
  a 4 by 4 "mask" to the data. Here we provide five possible
  masks: all rates (base) no remarriage (NoReMa), no divorce
  (NoDiv), no widowhood (NoWid), and no death (NoDea). To select
  one simply use the appropriate condition name in the call to
  MSMASK.  Combination masks may be specified simply by using
  .AND (e.g. NODEA .AND NOWID to omit death of person and of
  spouse).
@

vcm=msmask(vcm,base);

@ Call multi-state life table procedure @

lft = mslife(vcm,radage);
ages=lft[.,1];
vcp=lft[.,2:5];
vcl=lft[.,6:9];
vcll=lft[.,10:13];
vct=lft[.,14:17];
vce=lft[.,18:21];

@ Control printing @

format 10,3;
fn rd(x)=0.001*round(1000*x);

keepages=20|65;
i=0;
do until i==rows(ages); i=i+1;
  if sumc(ages[i,1].==keepages)>0;
    rd(ages[i,.]~vcl[i,.]~vce[i,.]);
    if i/4==trunc(i/4); print; endif;
  endif;
endo;
```

Appendix II

```
@ Program: MSLESLIE.PRC @
@ Produces procedure that returns Leslie matrix for the multi-
state projection of female population with fertility rates cvasfr
and transition rates cvm. Age structures of rates are those for
Canada: population data has 16 5-year age          transition
data has 17 5-year age groups; Fertility is age specific
fertility 10-49 in single year age groups. It is assumed that
these rates apply to married people, and that there is no
childbearing outside of marriage.
@
loadp mslife;

proc msleslie(cvasfr,cvm);
  local n,m,k,x,vcll,ix,asfr5,vmf,les,i,ri0,ril1;
  n=72; m=18; k=4;
  x=mslife(cvm,0);
  vcll=x[.,10:13];
  ix=reshape(seqa(1,1,n),m,k)';
  asfr5=zeros(m*5,1);
  asfr5[10:49,1]=cvasfr;
  asfr5=sumc(reshape(asfr5,m,5)');
  vmf=zeros(m,k*k);
  vmf[.,2]=asfr5;
  vmf=reshape(vmf,m*k,k);
  les=zeros(n,n); i=0;
  do until i==m; i=i+1;
    ri0=ix[.,i];
    if i>1;

@ Mortality projections using ratio of 5Lx for consecutive years @

      les[ri0,ril1]=vcll[ri0,.]*inv(vcll[ril1,.]);

@ Fertility projection using product of infant survival (5L0/5),
average 5 year fertility for member of xth age group (5Fx+
5Fx+5*5Px)/2, and fraction of children female (1/2) @

      les[1:4,ril1]=vcll[1:4,.]*
               (vmf[ril1,.]+vmf[ri0,.]*les[ri0,ril1])/20;

    endif;
    ril1=ri0;
  endo;
  retp(les);
endp;
```

```
@ Program: MSPROJ.CAN @
@ This program uses the MSLESLIE procedure to project the female
  population by age and marital status in Canada for 50 years in
  5 year steps. Data is aggregated in a variety of ways and numbers
  are plotted. For notes on masking see marriage.can and msmask.prc.
@
loadm ccf8,cf8,casfr;
#include "msleslie.prc";
loadp msmask;
loadm base,norema,nowid,nodiv,nodea;

cvm=msmask(cf8,base);

popmat=zeros(72,11);
popv=vec(ccf8');
les=msleslie(casfr,cvm);
i=0;
do until i==11; i=i+1;
  popmat[.,i]=popv;
  popv=les*popv;
endo;
format 8,2;
@ Create summary measures of population projection,
    p_t : total female population by year
    p_o : number of females older than 65
    p_o_nm : number of females older than 65 and not married
@
years=seqa(0,5,11);
ages=vec(seqa(0,5,18).*ones(1,4));
states=reshape(seqa(1,1,4),72,1);
old=ages.>=65;
notmar=states./=2;
all=ones(72,1);
p_o_nm=sumc(packr(miss(old .and notmar,0)~popmat));
p_o=sumc(packr(miss(old,0)~popmat));
p_t=sumc(packr(miss(all,0)~popmat));
print years~trim(p_t~p_o~p_o_nm,1,0);
```

Appendix III

```
@ Program: MSVARI.PRC @
@ This program creates a procedure MSVARI which returns the
projection matrix for the first two moments of a multi-type
branching process. Three arguments are used.
    P is the block diagonal combination of the column vectors
      p(i) of probabilities of possible transition sets from the ith
      state.
    Q is the matrix of counts for each transition state. The ith
      row has the number of counts that should be added to the
      ith state for each transition type.
    PKP is the expectation of the own direct product of p. Set to
      zero unless transition rates are themselves random so that
      PKP/=P.*.P  @

proc msvari(P,Q,PKP);
  local sts,trn,i,T11,B11,T22,B22,F21,M21,A11,A12,A22,A21,A;
  sts=rows(q); trn=cols(q);
  T11=Q; T22=Q.*.Q;
  B11=P; B22=P.*.P;
  F21=zeros(sts*sts,sts);
  i=0;
  do until i==sts; i=i+1;
    F21[(i-1)*sts+i,i]=-1;
  endo;
  M21=zeros(trn*trn,trn);
  i=0; do until i==trn; i=i+1; m21[(i-1)*trn+i,i]=1; endo;
  A11=T11*B11;
  A21=T22*M21*B11+T22*B22*F21;
  A22=T22*B22;
  A12=zeros(sts,sts*sts);
  A=A11~A12|A21~A22;
  retp(A);
endp;
save msvari;
```

```
@ Program: MSPQT.PRC @
@ This program creates a procedure MSPQT which takes as an
argument a Markov transition intensity matrix and produces the P
and Q (transposed) matrices that are needed by the MSVARI
procedure.
@
proc mspqt(mu);
  local k,id,p,p2,bigp,bigqt,i,ik,vp,pak;
  k=rows(mu);
  id=eye(k);
  p=inv(id+2.5*mu)*(id-2.5*mu);
  p2=p|(-sumc(p)'+1);
  bigp=zeros(k*(k+1),k);
  bigqt=bigp;
  i=0;
  do until i==k; i=i+1;
    ik=seqa((k+1)*(i-1)+1,1,k+1);
    bigp[ik,i]=p2[.,i];
    bigqt[ik,.]=eye(k)|zeros(1,k);
  endo;
  vp=miss(vec(p2),0);
  pak=packr(vp~bigp~bigqt);
  pak=trim(pak',1,0)';
  retp(pak);
endp;
save mspqt;
```

```
@Program: LESLIE.EXP @
@ This program illustrates the Pollard method for calculating
  variances in a simple 3x3 Leslie matrix application. The original
  Leslie matrix may be reconstructed by the product q*p.
@
loadp msvari;

let p[6,3]= 1     0          0
            0     0.1667     0
            0     0.1667     0
            0     0.3333     0
            0     0        0.6667
            0     0        0.3333;

let q[3,6]= 0   1   1   0   1  2
            1   0   0   0   0  0
            0   1   0   1   0  0 ;
format 5,2;
fn rd(x)=0.01*round(100*x);
rd(msvari(p,q,0));
```

```
@ Program: MARVAR.CAN @
@ Multi-state life tables with variances are created using
  Canadian marital status data. Elements of transition matrices
  may be set to zero using a masking procedure. For notes
  on data and masking see marriage.can
@
loadm cf8;
vcm=cf8;

radage=20;
k=4;
n=rows(vcm);
m=n/k;
#include "mslifeva.prc";
loadp msmask;
loadm base,norema,nowid,nodiv,nodea;

vcm=msmask(vcm,base);

@ Call multi-state life table procedure @

lft = mslifeva(vcm,radage);
ages=lft[.,1];
states=lft[.,2 3];
vcl=lft[.,4:7];

@ Control printing @

format 10,3;
fn rd(x)=0.001*round(1000*x);

keepages=20|65;
i=0;
do until i==rows(ages); i=i+1;
  if sumc(ages[i,1].==keepages)>0;
    rd(ages[i,.]~states[i,.]~vcl[i,.]);
    if i/20==trunc(i/20); print; endif;
  endif;
endo;
```

Appendix IV

```
@Program: MAKEDATA.CAN@
@Makes the data needed for multi-state applications programs and
 saves to disk in the form of GAUSS matrices. Data are for
 Canadian females, 1981. Elements of cf8 are transition
 intensities by age (0-89 in 5 year groups) and marital status
 (Single, Married, Widowed, Divorced).   Elements of ccf8 are the
 1981 population by age and marital status for the same age and
 marital status groups.Elements of casfr are single year age
 specific fertility rates from ages 10-49 for all women.
@
let cf8[72,4]=
      0.0021     0          0          0
      0          0          0          0
      0          0          0          0
      0          0          0          0

      0.0002     0          0          0
      0          0          0          0
      0          0          0          0
      0          0          0          0

      0.0002     0          0          0
      0          0          0          0
      0          0          0          0
      0          0          0          0

      0.0275     0          0          0
     -0.0270     0.0038    -0.0083    -0.1450
      0         -0.0001     0.0100     0
      0         -0.0036     0          0.1450

      0.1439     0          0          0
     -0.1432     0.0139    -0.0881    -0.2880
      0         -0.0001     0.0919     0
      0         -0.0136     0          0.2891

      0.1354     0          0          0
     -0.1343     0.0205    -0.1026    -0.2433
      0         -0.0002     0.1051     0
      0         -0.0199     0          0.2446

      0.0700     0          0          0
     -0.0685     0.0179    -0.0643    -0.1563
      0         -0.0004     0.0662     0
      0         -0.0170     0          0.1576

      0.0360     0          0          0
     -0.0337     0.0157    -0.0426    -0.0982
      0         -0.0007     0.0451     0
      0         -0.0142     0          0.1004

      0.0218     0          0          0
     -0.0188     0.0136    -0.0306    -0.0715
      0         -0.0011     0.0337     0
      0         -0.0110     0          0.0744

      0.0156     0          0          0
```

```
 -0.0115       0.0122      -0.0210      -0.0533
  0           -0.0018       0.0250       0
  0           -0.0081       0            0.0574

  0.0134       0            0            0
 -0.0074       0.0121      -0.0144      -0.0385
  0           -0.0029       0.0201       0
  0           -0.0055       0            0.0444

  0.0135       0            0            0
 -0.0048       0.0140      -0.0100      -0.0265
  0           -0.0047       0.0182       0
  0           -0.0036       0            0.0347

  0.0150       0            0            0
 -0.0035       0.0190      -0.0076      -0.0172
  0           -0.0080       0.0191       0
  0           -0.0023       0            0.0290

  0.0182       0            0            0
 -0.0014       0.0280      -0.0050      -0.0096
  0           -0.0126       0.0220       0
  0           -0.0016       0            0.0278

  0.0258       0            0            0
 -0.0008       0.0464      -0.0028      -0.0046
  0           -0.0229       0.0282       0
  0           -0.0009       0            0.0329

  0.0417       0            0            0
 -0.0004       0.0804      -0.0013      -0.0033
  0           -0.0422       0.0418       0
  0           -0.0005       0            0.0454

  0.0685       0            0            0
 -0.0003       0.1498      -0.0004      -0.0020
  0           -0.0841       0.0687       0
  0           -0.0003       0            0.0900

  0.1443       0            0            0
  0            0.2813      -0.0001       0
  0           -0.1580       0.1455       0
  0           -0.0003       0            0.2000 ;

let casfr[40,1]=
    0
    0
    0
    0.0002
    0.0012
    0.0045
    0.0122
    0.0234
    0.0360
    0.0494
    0.0637
    0.0805
    0.0970
    0.1110
    0.1216
    0.1310
```

```
      0.1312
      0.1275
      0.1194
      0.1099
      0.0930
      0.0805
      0.0661
      0.0522
      0.0408
      0.0315
      0.0237
      0.0169
      0.0120
      0.0087
      0.0063
      0.0041
      0.0026
      0.0014
      0.0008
      0.0004
      0.0003
      0.0001
      0.0001
      0 ;

let ccf8[18,4]=

      868.9        0          0          0
      864.9        0          0          0
      936.1        0          0          0
     1057.5       74.4        0.6        0.4
      597.8      561.3        1.4        9.1
      218.3      839.6        3.2       32
      106.4      857.3        5.9       47.6
       59        693.9        9.2       45.9
       40.5      569.7       14.6       38.5
       35.9      525.6       25.5       33.8
       37.5      507.7       47.2       29.5
       38.5      469.6       79.5       24
       36.8      355.1      109         16
       38.6      261.5      143         10.7
       34        157        155.3        5.9
       25.9       78.6      145.1        2.7
       16.6       30.6      113.6        1
       13.4       11.1      105.3        0.4 ;

save ccf8,cf8,casfr;
```

Appendix V

The results for the base run of each program may be reproduced in the following way. First, enter each of the programs in ASCII format including the data creation program and the following short program which may be used to create the precompiled procedures:

```
@ Program: MSSTART.CAN @
@ This program will create the required data matrices and
  precompiled procedures and save them to disk.
@
#include "makedata.can";;
#include "mslife.prc";;
#include "msmask.prc";;
#include "msvari.prc";;
#include "mspqt.prc";;
```

Each program should be in a separate file, and the file name should be taken from the first line of the program. Then, from DOS type:

```
GAUSS <RTN>
RUN MSSTART.CAN <F4> <F2>
```

You may then run each of the four applications programs by typing:

```
RUN MARRIAGE.CAN <F4> <F2>
RUN MSPROJ.CAN <F4> <F2>
RUN LESLIE.EXP <F4> <F2>
RUN MARVAR.CAN <F4> <F2>
```