Modelling Kinship with LISP

a two-sex model of kin-counts

J. Bartlema
L. Winkelbauer

November 1986
WP-86-69

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

Foreword

The Population Program at IIASA deals with various aspects of population aging phenomena in developed countries. The crucial problem related to aging is how to provide support for the increasing proportion of the elderly. The measure and way of this support depends on the kinship pattern for a particular population.

The paper develops the approach to modeling the kinship. The results of modeling show that the approach can be successfully implemented to the analysis of the family dynamics.

Anatoli Yashin
Deputy Leader
Population Program

Acknowledgements

We would like to express our indebtedness to Michael Stoto of Harvard University and Doug Wolf of IIASA, who commented upon an earlier draft of this paper. We benefitted from conversations with a number of scholars over a relatively long period of time, during which the results contained in this paper gradually took shape. Jorge Somoza of CELADE and Hein Moors of NIDI made comments which increased our understanding of certain aspects of kinship modelling. T.Pullum provided the computer program with which the results of the GKP model were calculated. The computer division of IIASA was very collaborative in helping to worm our way through the succession of bottlenecks we encountered. In this context a special word of thanks to P.Pronay and D.Legault is in it's place.

This paper was written at IIASA as part of a research project currently being undertaken by the Catholic Universtity of Tilburg, The Netherlands, on the use of Collective Provisions by the elderly. The project is supervised by prof.G.A.B.Frinking. Funds for the stay in Austria were supplied by the subfaculty of social cultural sciences of Tilburg University and by the IIASA Foundation in the Netherlands.

Jan Bartlema is a research associate at the *Subfaculty of Social Cultural Sciences of the Catholic University of Tilburg*, the Netherlands.

Lothar Winkelbauer is research assistant at the *Advanced Computer Applications* project of IIASA.

CONTENTS

1. Introduction

2. Characteristics of LISP

3. Kinship Models

4. The GKP Model: an elaboration

5. Translating Aggregate Measures into Kin Counts

Appendices

1 : An illustrative analysis of stepfamily relations
2 : A note on the sister's riddle
3 : LISP source code for assignation of fathers to children
4 : References

Modelling Kinship with LISP

a two-sex model of kin-counts

J. Bartlema and L. Winkelbauer

1. Introduction

It is frequent in family sociology and cultural anthropology to conceive of kinship structures as a socio-cultural superstructure on a biological basis

The concept of the family we shall adopt in this context is derived from R. Adams' theoretical discussion (1971) according to which there are two distinct dyadic relations which may be considered as the elementary atoms from which all human kinship structures are constructed: the mother-child and wife-husband dyads. From a demographic point of view this implies that, given prevailing levels of mortality, the processes of fertility and nuptiality are of key importance to understand and model kinship. While childbearing can be considered to be essentially a biological fact, marriage is a cultural phenomenon, albeit one with the function of regulating a biological fact. In the terms of Firth "Kinship is fundamentally a reinterpretation in social terms of the facts of procreation and regularized sex union." (1948). Schneider (1965) gives an inventarization and critique of a number of definitions of this nature. The model applied here adopts separate technical instruments to model the biological and the cultural dimension.

The objective of modelling kinship within an applied-demographic context is to produce a replica which makes best use of the available information and works out the implications of this input under an appropriate set of assumptions with a useful degree of detail. The output we want to generate is of a global nature. We would like to give a general idea of the consequences for kinship structures in society of the developments in fertility and mortality that have been taking place over the course of the century. A second topic of interest is the effect that an alteration of the nuptiality structure from monogamy to serial monogamy would have upon kinship networks.

Modelling institutions is becoming more problematic than it was in the recent past due to the fact that western societies are undergoing a process of de-institutionalization. The normative, role-defining power of our institutions with respect to for example the formation and dissolution of unions, or the entrance and exit from the workforce, the educational system and so forth is decreasing. In the process all kinds of hybrid variants of the solid institutions of the post-war era are being generated. This confronts the registration systems of industrial societies with novel conceptual ambiguities. The social scientist's object of study is becoming increasingly difficult to classify into well-defined discrete categories. It makes sense in such a situation to look for tools that can grasp soft material. A programming language that is particularly suited to manipulate symbols rather than numbers might therefore be helpful to complement existing mathematical and statistical procedures in modelling institutions.

It is also reasonable, upon experiencing a growing sense of indeterminacy to fall back upon the things we do know by biological necessity:

* people are born and therefore have fathers and mothers,

* some people enter a first reproductive union ,

* people die.

We restrict the input of the model to fertility rates by age of parent, survival-rates by sex and two-sex first-marriage matrices , all in 5-year age groups.

The model designed to transform the input we have into the output we desire consists of two distinct phases. First the numerical relations between kin of different 5-year agegroups are calculated in a two-sex stable population. Thereafter these aggregate measures are translated into a hypothetical population in which each individual is identified, with his or her network of nuclear kin. The first phase of the model uses standard biomathematical procedures, while the second applies LISP. The first phase is macroanalytic, while the second uses stochastic procedures. The result is a model with traits of macro- as well as micro-models.

The emphasis in this paper is upon methodological issues. We are mainly discussing the merits of a model, and not the implications of shrinking kinship support networks for the elderly. After a brief introduction into LISP and the field of kinship modelling the Goodman, Keyfitz, Pullum approach is summarized and an application discussed. Thereafter a simulation procedure is described. In an annex an illustrative application is presented, giving an impression of the effect of an alteration from a strictly monogamous mating system to one in which individual lifecycles may contain two successive reproductive unions.

2. Characteristics of LISP

If we define kinship modelling for the purpose at hand as " the generation of formal representations of numerical relations between kin ", then it is clear that mathematics has traditionally produced the tools to do it with. From the classical theories of branching processes, to the most recent stochastic micro-models: all are mathematical. Meaningful models cannot restrict themselves however to the construction of trees and networks of kinship but must move into the direction of representing the dynamics of family formation and dissolution in terms of cultural developments, psychological processes, and group-dynamics. What we would like to model are the forces which make the components of our model -persons- behave as they do. Some of these forces are external to these persons and have to do with the socio-economic structures within which they are embedded. Others are internal: psychological processes, and others again have to do with the interaction between the external and the internal: cultural and socio-psychological variables.

There are numerous verbal theories and empirical studies on such issues, but attempts to construct models which work out in a formal fashion what the implications of certain qualitative postulates would be on a hypothetical population, such models do not yet exist. They can be made, but require the use of instruments which can handle symbols as well as numbers. The conviction that it might be useful to think in such a direction lies behind the decision to use a computer language appropriate for this kind of task. To those of us who are not familiar with the approach and who might be willing to consider the possiblity to complement our quantitative results with formalized qualitative thought, we propose a brief digression into the structure of LISP. The reader interested in kinship modelling 'tout sec' may skip the following paragraph. Before presenting the introduction to the programming language, we hastily add that the application of LISP presented in this paper is of the most primitive nature.It is a declaration of intention.

Artificial Intelligence has been described as the art of making the computer do things that would require intelligence if performed by human beings. In 1958 **John McCarthy** created the programming language LISP (LISt Processing) to give the pioneers of the Artifical Intelligence community a tool which allows to process *symbols* (i.e. qualitative terms) in addition to *numerical calculations* (i.e. processing quantitative terms) which are the central aim of *conventional programming languages* such as FORTRAN, PASCAL, PL/I or COBOL.

Although of the programming languages still in use, only FORTRAN is older than LISP one could have said that until very recently LISP was the only AI language used by AI programmers.

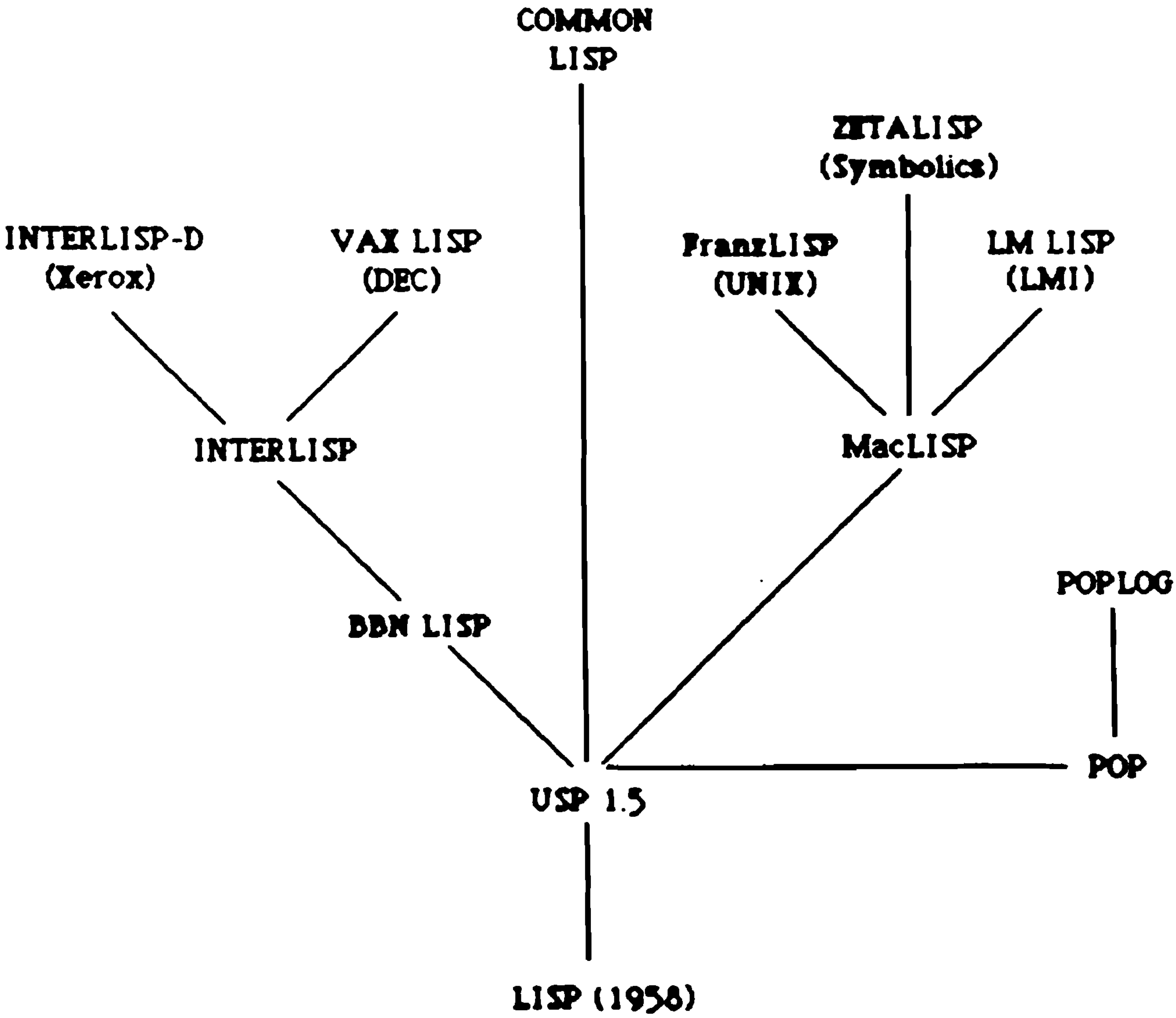


Figure 1: The family tree of LISP

McCarthy describes LISP as follows: (McCarthy in Barr and Feigenbaum, 1982b)

1. *Computing with symbolic expressions rather than numbers; that is, bit patterns in a computer's memory and registers can stand for arbitrary symbols, not just those of arithmetic.*
2. *List processing, that is, representing data as linked-list structures in the machine and as multilevel lists on paper.*
3. *Control structure based on the computation of functions to form more complex functions.*
4. *Recursion as a way to describe processes and problems.*
5. *Representation of LISP programs internally as linked lists and externally as multilevel lists, that is, in the same form as all data are represented.*
6. *The function EVAL, written in LISP itself, serves as an interpreter for LISP and as a formal definition of the language.*

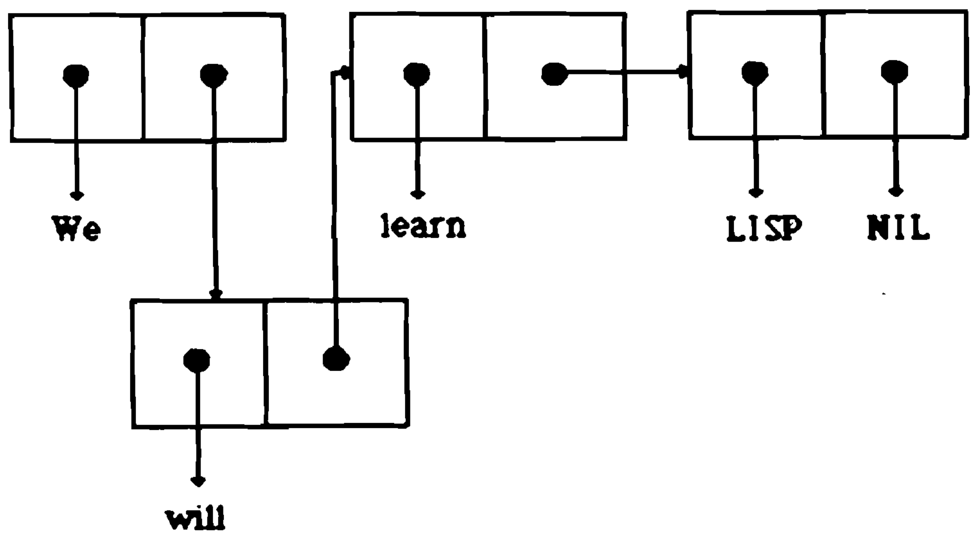


Figure 2: The basic LISP data stucture

There is no essential difference between data and programs, hence LISP programs can use other LISP programs as data. LISP is highly recursive, and data and programs are represented as nested lists. It does not always make for easy-to-read syntax, but it allows for elegant solutions to complex problems that are difficult to solve in the various conventional programming languages.

There are only a few basic LISP functions; all other LISP functions are defined in terms of these basic functions. This means that one can easily create new higher-level functions. Hence, one can create a LISP operating system and then work up to whatever higher level one wishes to go to. Because of this great flexibility, LISP has never been standardized in the way that languages such as FORTRAN and BASIC have. Instead, a core of basic functions has been used to create a wide variety of LISP dialects (see Figure 1).

LISP is unique among programming languages in storing its programs as structured data. The basic data structures in LISP are the ***atom***, any data object that cannot be further broken down, and the ***CONS node***.

Each ***atom*** has an associated property list that contains information about the atom, including its name, its value, and any other properties the programmer may desire.

A ***CONS node*** is a data structure that consists of two fields, each of which contains a pointer to another LISP data object. CONS nodes can be linked together to form data structures of any desired size or complexity (Figure 2). To change or extend a data structure in a LISP list, for example, one need only to change a pointer at a CONS node.

Elements of lists need not be adjacent in memory - it is all done with pointers. This not only means that LISP is modular, it also means that it manages storage space very efficiently and frees the programmer to create complex and flexible programs.

Conventional programming languages normally consist of sequential statements and associated subroutines. LISP consists of a group of modules, each of which specializes in performing a particular task. This makes it easy for programmers to subdivide their efforts into numerous modules, each of which can be handled independently.

For this reason LISP has been used for many AI projects in the following fields: Knowledge-based Systems (Expert Systems), Natural Language Understanding Systems, Computer Vision, Robotics, Gaming Programs, Learning Systems. It is used here to program the assignation of relatives to eachother. The dialect of LISP we used is Franz Lisp.

3. KINSHIP MODELS

Kinship modelling has been a topic of interest for demographers since the discipline emerged from the context of mathematics, and the behavioral sciences. In an implicit way notions derived from kinship structures are used when referring to the net reproduction rate, the total fertility rate and so forth. Recently, the subject of modelling kinship and household structures has received renewed attention, creating a body of literature, a common theoretical perspective and a set of methods. In short a distinct field of inquiry can said to be originating with the study of kinship and household structures as its objective.

No attempt to review the field will be given here. (See Keyfitz, 1984; Bongaarts,1982; de Vos and Palloni, 1984). For our purposes however a useful distinction between the types of models used is between those based on macro-analytic expressions and those based on micro-simulation procedures. To the first family belong the Goodman, Keyfitz ,Pullum (1974), Krishnamoorty (1979), Le Bras (1973), Madan (1986) models and so forth, while the second class contains such models as the early Hyrenius models (Hyrenius and Adolfsen, 1964; Hyrenius, Adolfsen, Holmberg 1966, Holmberg, 1968), the Universtity of North Carolina POPSIM model the Le Bras (1984) model, and the Wolf (1986) model based on stochastic simulation procedures. The analytic models have the advantage that the mechanics of constructing kin-groups are easily accessed, through mathematical expressions on an aggregate level, while with a stochastic procedure kin-structures result as the less transparent outcome of random assignation procedures on an individual basis. On the other hand the degree of detail provided by the micro-approach is superior, since all characteristics of a real population that we might be interested in can be simulated. For example, where it is frequent to have expressions for expected values and possibly distributions of kin by age under rather simple assumptions in the analytic models, more elaborate correlations between the component variables can be introduced in the stochastic models. It is possible to simulate relations for which no analytic expressions can be formulated.

It has generally been less problematic to encorporate the interaction of fertility and mortality in kinship models than that of nuptiality. This is due to a number of factors:

- *while fertility can be readily studied as a renewable event, nuptiality is usually conceptualized as a process of entrance and exit with respect to the institution of marriage leading to state-time models (eg. Wolf ,1986; Willekens, Shah, Ramachandran, 1982.) . As a result the complexity of the models is increased.*
- *Data requirements for the encorporation of nuptiality are often difficult to meet, specially if the model specification uses age-specific transition rates from never-married to married states, from married to divorced states, from marriage to widowhood and vice versa, as well as mortality rates specific for each civil-status.*
- *While the margin for ambiguity as to the definition of a live-birth is small, this can not be said of marriage. The distinction between registered marriage and consensual unions has long been recognized as one of degree and not of kind (see for example Van de Walle, 1968). Measuring the age- specific occurrence of matrimony is becoming a problem not only in countries with deficient statistics, but also in the so-called information societies of the west. The problem is not that there are no data; the problem is that the validity of the information we have on membership of institutions in our societies is becoming questionable (Bartlema and Vossen, 1984).*

Table 1. Expected Numbers of kin in three simulated theoretical populations, approximating the Netherlands 1939, 1984 and the CBS middle variant forecast for 2030.

	Age	Daughters	Grand Daughters	Grand Grand Daughters	Mothers	Sisters	Nieces	Aunts	Cousins
t =		D_x	GD_x	GGD_x	M_x	S_x	N_x	A_x	C_x
1939	x	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(11)
	60	1.68	1.24	.003	.082	1.37	2.77	.256	2.21
	65	1.66	1.84	.020	.028	1.22	2.75	.143	1.98
	70	1.64	2.33	.085	.007	1.04	2.71	.071	1.70
	75	1.62	2.63	.264	.001	.82	2.65	.031	1.40
	80	1.58	2.77	.631	.000	.59	2.57	.011	1.08
	85	1.51	2.81	1.220	.000	.38	2.44	.004	.78
	90	1.42	2.81	1.987	.000	.22	2.27	.000	.53
	95	1.28	2.78	2.811	.000	.11	2.03	.000	.33
t = 1984									
	60	1.28	1.20	.004	.237	1.18	1.64	.367	1.47
	65	1.28	1.46	.028	.103	1.11	1.63	.216	1.37
	70	1.27	1.57	.118	.032	1.01	1.62	.110	1.23
	75	1.26	1.62	.344	.006	.85	1.61	.049	1.05
	80	1.25	1.64	.730	.001	.66	1.58	.019	.84
	85	1.22	1.64	1.190	.000	.45	1.53	.006	.62
	90	1.18	1.64	1.588	.000	.26	1.46	.002	.41
	95	1.10	1.63	1.850	.000	.13	1.35	.001	.25
t = 2030									
	60	.73	.39	.001	.234	.67	.53	.198	.471
	65	.72	.47	.001	.100	.63	.52	.115	.438
	70	.72	.51	.022	.031	.57	.52	.058	.392
	75	.72	.52	.063	.007	.49	.52	.025	.332
	80	.71	.53	.133	.001	.37	.51	.010	.263
	85	.70	.53	.216	.000	.25	.49	.003	.192
	90	.67	.53	.289	.000	.15	.47	.001	.127
	95	.63	.53	.337	.000	.08	.43	.000	.076

- *While there can be no doubt that biomathematics has developed adequate tools to study the biological aspects of demographic behavior, this can not be said with such confidence with respect to the cultural component of demographic variables.*

For these reasons it can be justified to develop a model which attempts to accomplish the following:

- *Do as much as possible with analytic expressions utilizing reliable and valid information with respect to fertility, mortality and entrance into first reproductive union. The interaction between mortality and fertility determines the numbers of people in biological dyadic kin relations in a given population.*
- *Use a tried biomathematical approach for the biological basis and explore the possibilities of employing a softer instrument to model the subtler material of culture.*
- *Work out the consequences on kinship networks of an alteration of mating principles using as simple an input as possible to achieve this end. We do not know how nuptiality variables will develop, and yet would like to have an impression of what the effects would be of certain possible courses of aggregate behavior. In our application for example an alteration from a system of strict monogamy with no remarriage to one with serial monogamy will be simulated and the effects upon the density of the kin networks will be studied.*

With these goals in mind a model was developed which starts by elaborating on the Goodman, Keyfitz, Pullum expressions. The elaboration consists in making the model age-specific for both the participants in the kin-dyad under study, and in applying the general line of thinking to a two-sex stable population. The original model, it is understood, is restricted to the single sex case in which it calculates average numbers of kin with respect to ego by age. To model the consequences of altering mating principles LISP was used. The result is a model which starts with macro-analytical expressions and produces an output which is similar to that of a micro- simulation model. It gives a hypothetical population in which each individual is identified, and of whom the (nuclear) kinship networks are specified. An illustrative application to the Netherlands is given.

4. The GKP Model, an elaboration.

We depart from the analytical expressions for expected numbers of kin in a female stable population, given by Goodman, Keyfitz and Pullum (1974). The average numbers of survivng daughters of women aged a at time t is simply:

$$^{f(a)}D(.) = \int_{\alpha}^{a} l^{f}(a-x)\ {}^{f}f^{f}(x)\ dx \qquad (1)$$

here $l(x)^f$ refers to the lifetable survivorship function for the female population and ${}^f f(x)^f$ is the fertility rate at exact age x for girl children of women. The integral goes from α, the beginnning of the reproductive period until a the age of Ego at time t. The dot between brackets at the right of the D symbol for daughters indicates that we are integrating over all ages of Alter. In the fertility rate the female symbol as a right upper superscript indicates that the gender of the child is female while the left superscript informs us that the rates are by age of mother, rather than father. In these expressions all survivorship refers to the female population as indicated.

The average number of surviving mothers per woman aged a at t is

$$
{}^{f(a)}M(.) = \int_{\alpha}^{\beta} \frac{l^f(a+x)}{l^f(x)} \left\{ N_{t-a}^f(x)\ {}^f f^f(x) \right\} dx \tag{2}
$$

where x is the age of the mother at birth of Ego and $N_{t-a}{}^f(x)$ refers to the number of women of age x at time t-a. The

$$
\left\{ N_{t-a}^f(x)\ {}^f f^f(x) \right\}
$$

factor which we encounter here as well as in the expressions for siblings is a weighting distribution for birth of Ego. In this case then it represents the distribution of female births by age of mother. The integral goes from the beginning to the end of the reproductive period.

The average number of sisters of women is given as the sum of their elder and their younger sisters. These are calculated separately because a term for the survivorship of mothers has to be included in the expression for younger sisters. The expressions for elder and younger sisters are respectively:

$$
{}^{f(a)}ES(.)^f = \int_{\alpha}^{\beta} \left[\int_{\alpha}^{x} {}^f f^f(y)\, l^f(a+x-y)\, dy \right] N_{t-a}^f(x)\ {}^f f^f(x)\, dx \tag{3a}
$$

$$
{}^{f(a)}YS(.)^f = \int_{\alpha}^{\beta} \left[\int_{x}^{\beta} {}^f f^f(y)\, \frac{l^f(y)}{l^f(x)}\, l^f(a-y-x)\, dy \right] N_{t-a}^f(x)\ {}^f f^f(x)\, dx \tag{3b}
$$

The age y refers to mother's age when she had Ego's sister.

Although the Goodman, Keyfitz, Pullum article goes on to expressions for more distant kin we limit ourselves in this context to the nuclear families of origin and procreation. (see Keyfitz, 1977 for a more elaborate description of the model, and Keyfitz ,1986 for a recent application) . We note that the GKP analytical expressions are not restricted to the stable case. If we have cohort survivorhip and the weighting distributions for birth of Ego are known, expected numbers of kin can be calculated exactly in a closed population. The assumption of stability is made here for convenience, not by conceptual necessity.

NETHERLANDS STABLE APPROXIMATION 1984
BROTHERS CF MEN PATRILINEAL

W\A	65-69	70-74	75-79	80-84	85-89	90-94	95-99	100-104	105-	TOT	TOT /Nw	
10	0.000034	C	0	0	0	0	0	0	0	0.00003	C.000000	0.000000
15	0.000928	0.000027	0	0	0	0	0	0	0	0.00095	C.000002	0.000000
20	0.007532	0.000746	0.000020	0	0	0	0	0	0	0.00830	C.000015	0.000000
25	0.032051	0.006055	0.000551	0.000012	0	0	0	0	0	0.03867	C.000074	0.000000
30	0.101558	0.025714	0.004463	0.000342	0.000006	0	0	0	0	0.13208	C.000266	0.000000
35	0.309592	0.081155	0.018881	0.002760	0.000156	0.000002	0	0	0	0.41255	C.000873	0.000000
40	1.006494	0.247149	0.059532	0.011663	0.001258	0.000047	0.000000	0	0	1.32614	C.002954	0.000001
45	3.226124	0.802035	0.18C970	0.036706	0.005306	0.000381	0.000009	0.000000	0	4.25153	C.009984	0.000008
50	9.391756	2.538989	0.580014	0.110202	0.016492	0.001589	0.000070	0.0000C1	C.000000	12.63911	C.031684	0.000072
55	22.615773	7.235607	1.797440	0.345756	0.048469	0.004836	0.000287	0.C000C6	C.000000	32.04817	C.087609	0.000464
60	38.756098	16.947925	4.982479	1.042229	0.147919	0.013825	0.000849	0.000024	C.000000	61.89135	C.189679	0.001729
65	42.217528	27.833199	11.184203	2.768677	0.427301	0.040433	0.002326	0.000068	0.000001	84.47373	C.302859	0.003220
70	27.833955	28.573784	17.310257	5.857114	1.069781	0.110076	0.006412	0.C00176	C.000001	80.76156	C.359418	0.002943
75	11.184787	17.310727	16.329526	8.330050	2.079562	0.253233	0.016041	0.000445	C.000003	55.50437	C.333682	0.001390
80	2.768885	5.857420	8.330276	6.616921	2.490429	0.414511	0.031C74	0.000938	0.000007	26.51046	C.255680	0.000317
85	0.427353	1.069862	2.079671	2.490496	1.459575	0.366254	0.037528	0.001340	0.000011	7.93209	C.166339	0.000028
90	0.040442	0.110090	0.253252	0.414533	0.366264	0.143111	0.022107	0.001079	C.000010	1.35089	C.092389	0.000001
95	0.002327	0.006414	0.016043	0.031076	0.037530	0.022108	0.005318	0.000391	0.000005	0.12121	C.043916	0.000000
100	0.000068	0.000176	0.000445	0.000938	0.001341	0.001079	0.000391	0.000045	C.000001	0.00448	C.018065	0.000000
105	0.000001	0.000001	0.000003	0.000007	0.000011	0.000010	0.000005	0.0000C1	0.000000	0.00004	C.006573	0.000000
TOT	159.92329	108.64707	63.128026	28.059482	8.151398	1.371496	0.122418	0.004515	0.000040	369.40773	C.010174	0.010174
TOT/Nm	0.573364	0.483518	0.379514	0.270619	0.170938	0.093798	0.044353	0.018187	C.006597			

Table 2.

NETHERLANDS STABLE APPROXIMATION 1984
BROTHERS CF MEN MATRILINEAL

W\A	65-69	70-74	75-79	80-84	85-89	90-94	95-99	100-104	105-	TOT	TOT /Nw	
30	0.001853	C	0	C	0	0	0	0	0	0.00185	0.000004	0.000000
35	0.037750	0.001481	0	0	0	0	0	0	0	0.03923	C.000083	0.000000
40	0.336817	0.030136	0.001086	0	0	0	0	0	0	0.36804	C.000820	0.000000
45	1.936695	0.268396	0.022067	0.000670	0	0	0	0	0	2.22783	C.005232	0.000003
50	8.032926	1.524197	0.194098	0.013438	0.000301	0	0	0	0	9.76496	C.024479	0.000066
55	23.075524	6.188735	1.079033	0.115705	0.005910	0.000088	0	0	0	30.46500	C.083281	0.000641
60	42.277878	17.292455	4.261597	0.625667	0.049500	0.001686	0.000015	0	0	64.50880	C.197701	0.002874
65	46.755281	30.362412	11.411564	2.368095	0.256516	0.013531	0.000284	0.000001	0	91.16768	0.326858	0.005741
70	30.362556	31.645039	18.883246	5.976182	0.915002	0.066081	0.002146	0.000021	0.000000	87.85027	0.390965	0.005331
75	11.411698	18.883336	18.084706	9.087005	2.121837	0.216595	0.009630	0.000149	C.000000	59.81494	0.359597	0.002471
80	2.368144	5.976247	9.087048	7.328141	2.716735	0.422937	0.026578	0.000563	0.000002	27.92640	C.269336	0.000539
85	0.256525	0.915021	2.121860	2.716748	1.616458	0.399536	0.038291	0.C01147	C.000007	8.06559	C.169139	0.000045
90	0.013531	0.066083	0.216599	0.422942	0.399537	0.158493	0.024116	0.001101	0.000009	1.30241	0.089073	0.000001
95	0.000284	0.002146	0.009630	0.026578	0.038291	0.024116	0.005889	0.000427	0.000005	0.10737	C.038900	0.000000
100	0.000031	0.000021	0.000149	0.000563	0.001147	0.C01101	0.000427	0.000050	0.000001	0.00346	C.013939	0.000000
105	0.000030	0.000000	0.000000	0.000002	0.000007	0.000009	0.000005	0.000001	0.000000	0.00002	C.004060	0.000000
TOT	166.567455	113.15571	65.372683	28.681736	8.121240	1.304172	0.107381	0.003460	0.000024	383.61386	C.017712	0.017712
TOT/Nm	0.598260	0.503583	0.393009	0.276620	0.170306	0.089194	0.038905	0.013938	0.004060			

Table 3.

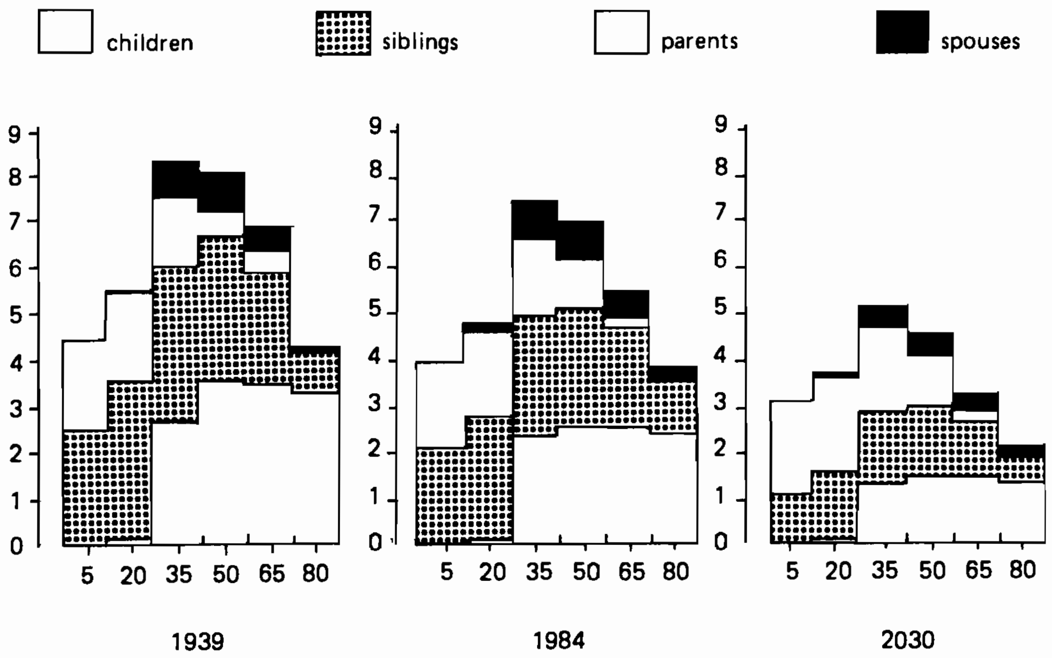


Figure 3: Average numbers of kin by age of Ego in three theoretical populations source: Bartlema, Van Nimwegen, Moors (1986)

The expressions presented above permit calculation of average numbers of female kin in a one sex stable population. They do not give distributions by age of Alter, nor do they include the male half of the population. In order to achieve these results, elaborations of the GKP expressions were worked out (Bartlema and De Jong, 1985). In a two-sex stable population for example the number of younger brothers of women can be writtten as:

$$f^{(a)}YB(w) \left[\sum_{x=12.5}^{67.5} {}_5{}^m\Psi x^f * {}_5{}^m f^m_{x+i} \frac{{}_5L^m_{x+i}}{{}_5L^m_x} \right] * {}_5L^m_x * \frac{TFR^m - 1}{TRF^m} * {}_tN_A{}^f \;; i = A - W \quad (4)$$

The step from continuous to discrete notation reflects the form in which our data are available. The survivorhip function used is therefore not l(x) but ${}_5L_x$, the fertility rates refer to five-year age groups and

$${}_5{}^m\Psi x^f$$

is used to represent the weighting distribution for female children by age of father. It is now clear why an index must be taken up for the sex of the child as well as the parent. We also keep track of the sex of the survivorship that applies. No summation over ages of Alter is carried out, since we want to make the relation age-specific for both parties in the dyad. The age of Alter is referred to with the letter W (capitals being used to distinguish these five-year agegroups from the exact ages used above).

For the rest the expression is analogous to what we saw before , except for the introduction of two additional factors:

$$N_{t\,A}^{\;f} \text{ and } \frac{TRF^m - 1}{TRF^m}.$$

The first is simply the number of persons of the female sex in the age-group A at time t. This factor is introduced in order to create absolute numbers of persons in the age-specific dyadic relation in a hypothetical stable population with a given numerical distribution of the population. We want expected values not per single woman of a given age but in a stable population with 1000 women in the over 65 age range. The second factor is an adjustment of Ego's father's fertility for the birth of Ego. The conversion from the continuous to the discrete requires such an adjustment. We will come back to the issue shortly, after a brief discussion of the results.

Besides the numbers of siblings, parents and children of Ego, we calculated numbers of first mates on basis of two-sex first marriage matrices. The expression used is derived from Hill-Trussel (1977):

$$^{f(A)}H(w) = {}_5N_A^{\;f} * \sum_{C=2.5,7.5,\ldots}^{A-a} g_{t-c}^{\;f}(a-c) * p^m(w-c \mid A-C) * \frac{{}_5L_w^m}{{}_5L_{w-c}^m}$$

where

$$g_{t-c}^{\;f}(a-c)$$

refers to the female first marriage rate at age A-C at time period t-c, and

$$p^m(w-c \mid A-C)$$

is the proportional distribution of female marriages at age A-C by age of male mate. The character of this expression is analogous to those for the other kin relations.

Both the original GKP expressions and the elaborated version were applied to the Netherlands. The objective as stated in the introduction, of this exercise is to get a rough idea of developments in the real population. The question which must be addressed is then which mortality and fertility schedules to use. It is clear that though the observed population at a given moment in time is not stable, there is a stable population which best approximates it's distribution. The issue of how the fertility and mortality schedules were found that approximate the observed populations adequately will not be dealt with here (see Bartlema and De Jong for a discussion). Suffice it to know that the empirical populations of the Netherlands in 1939 and 1984 as well as the 2030 CBS (1984) middle variant forecast were replicated fairly well with fertility and mortality schedules derived from empirical material.

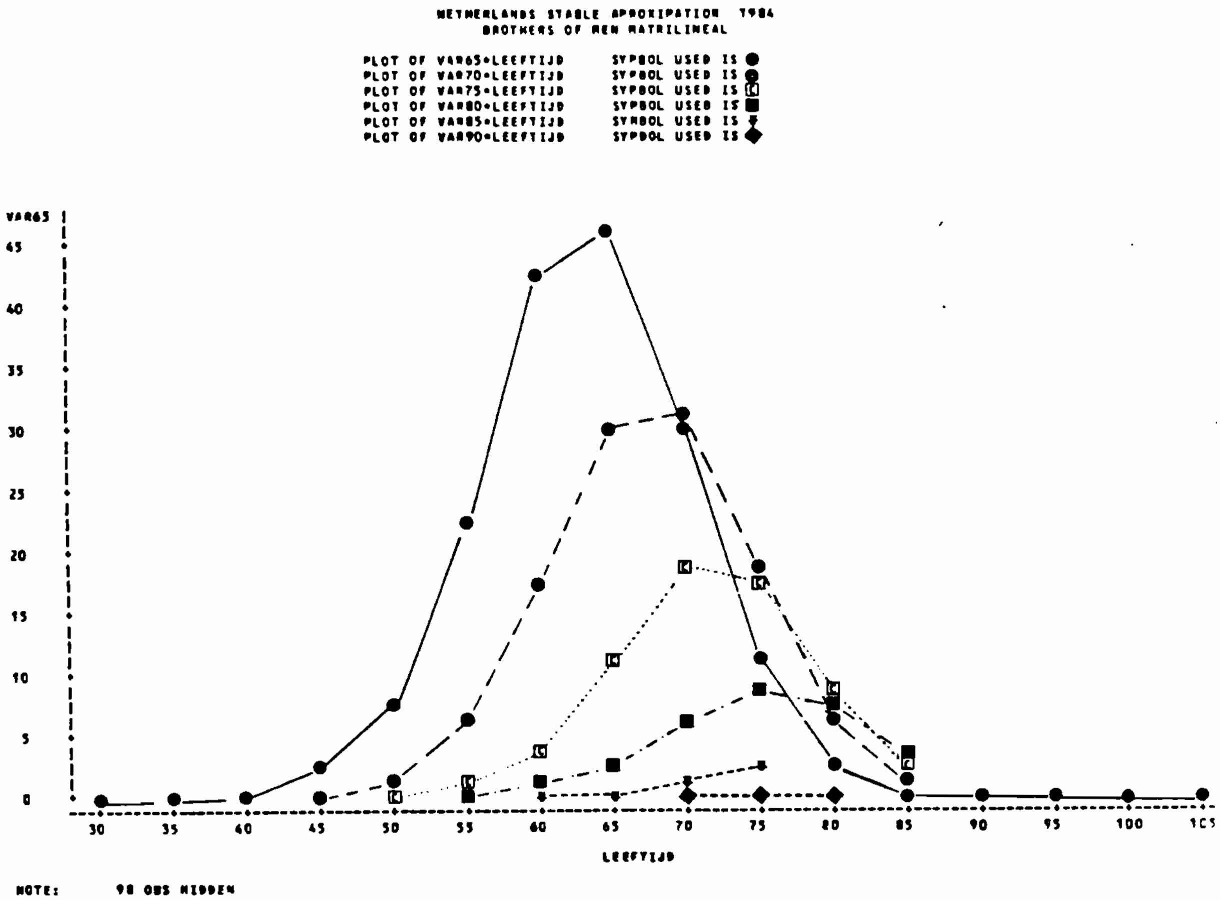


Figure 4: Absolute numbers of Matrilineal Brothers of Men in a Theoretical Population with 1000 women over the age of 65

The procedure used was to parametrize the fertility and mortality schedules in the relational Brass-type system and to find the parameter values which resulted in the best fit. If we look far enough into the past, the age structure of a population may be considered as a function of the mortality and fertility schedules that prevailed, according to the theory of weak ergodicity. If these schedules were constant the theory of strong ergodicity informs us that the resulting population will be stable. For the moment let us assume that we have succeeded in finding those constant courses which best approximate the undulations that produced the target populations. If this is so the vital rates that reproduce the population by approximation will also reproduce the prevailing kinship structures by approximation. A validation of these theoretical results with empirical evidence will be carried out.

The results of the application of the GKP expressions to the Netherlands under such assumptions, are summarized in table 1 and represented graphically in figure 3 . The general impression is one of kinship networks which are thinning out: the average numbers of kin per person are declining. The implication for the strength of kinship support networks for the elderly is immediately obvious. Policies aimed at shifting part of the burden of caretaking for the elderly from formal organizations to primary groups must face the fact that many of the elderly of the near future will have very few biological nuclear kin to fall back upon. An impression of the type of results provided by the elaborated version of the model is given in tables 2 and 3 , and in graphs 4 and 5. The tables give expected numbers of kin of age W, W+4 for egoes aged A, A+4 in the stable population with 1000 womwn in the over 65 age category. The fact that we have separate matrices for patrilineal and matrilineal kin is useful, since the two are not synonimous, and will become lcss so if it becomes a more frequent phenomenon to have children from different mates.

We may now return to the question of the adjustment for birth of Ego which was taken up in the expressions for siblings in the discretized GKP version. The fact that the number of siblings of Ego plus one for the birth of Ego herself gives a total size of family that is larger than the total number of children per woman at the end of her reproductive span has been commented upon in the literature. A comparison of the numbers of sisters a woman of age 60 has with the values of NRR for the years in question will illustrate this point. For the years 1939, 1984 and 2030 the NRR in the stable schedules are respectively 1.67. 1.28 and .73., while the number of sisters of Ego plus 1 gives the figures of 2.37, 2.18 and 1.67. This would seem to imply that Ego's mother had a higher level of fertility than the combined fertility/mortality schedule permits.

The original GKP model is homogenous , continuous , time-independent and deterministic . Every member of the population is subjected to the same demographic regimes. These regimes apply uninterruptedly over the whole reproductive period. No account is thus taken of pregnancy and post-partum-amennorhea . The population we have before us is one with with no sterility, universal marriage and no variation around the expected value. We are imagining that all women have the same reproductive behavior, invariant over time, which we know exactly and which results in a point estimate number of kin. It is from Ego's perspective that the kinship network is overseen. Ego is viewed as a reference point in her mother's continuous invariant reproductive past, with the experience of all Egoes and all mothers being identical. Crucial for the reasoning that follows are four points with respect to the characteristics we referred to: * homogeneity implies that every member of the population is subjected to the same demographic regimes. * continuity refers to the fact that all measures are expressed as positive real numbers, not as integers. * time-independence refers to the fact that the birth of a child to a mother is independent of the birth of any other children to the mother. * determinism implies that all measures apply equally to all members of the population: there is no variance about these expected values.

The strength of such a model lies in it's abstraction: it permits a concise lucid understanding of the processes involved. If we want to construct a hypothetical population on basis of this model, however we must take account of the fact that real population dynamics are heterogenous, discontinuous, time-dependent and stochastic. Let us maintain the homogeneity in first instance, and begin with making the results discrete. In a deterministic stable population (and abstracting from the effects of mortality) the average number of sibs a person has including himself or herself) should be equal to the average number of children a woman has, as demonstrated by Preston (1976). Le Bras (1982) also refers to the question of what he calls the 'apparent paradox' of sibship in terms of this same expression. Under the assumption of homogeneity of reproduction then, and considering now Ego not as a point but as a person, we must subtract this person from her (or his) parent's reproductive experience to keep the generations in balance.

The discounting of the birth of Ego in her parent's overall fertility should be such that before Ego's birth, accumulated parental fertility is decreased, and after her birth accumulated net fertility is decreased. If we consider mortality over the reproductive span as negligible for our purposes we may neglect parental mortality. If we furthermore accept that the birth of Ego affects her parent's fertility in a fashion which is independent of the sex of Ego, it is clear that the adjustment should be carried out upon fertility of children of both sexes. The reduction in overall parental fertility should equal unity. The question remains as to where in the parent's reproductive history to introduce the adjustment.

Maternal fertility is the combined effect of natural mechanisms and fertility control. The first cluster of effects would tend to concentrate the reduction around Ego's birth. The second would push the location of the reduction to ages away from the age at which the' mother had Ego, assuming that there is a preference for a birth spacing which is not too wide. The assumption here is that these effects compensate. The resulting adjustment also has the virtue of simplicity. It appears to be fitting to keep pretensions low in this context, specially if the relation between the adjustment for paternal and maternal reproduction is concerned. Which type of natural effects we are dealing with differ completely with the sex of the parent. For men there are no immediate biological reasons for concentration of fecundity inhibiting effects around the age at which his mate had Ego. The reader interested in the issue of the expected number of siblings by age of Ego is referred to appendix 2 for a further discussion.

We shall conclude this paragraph with a brief observation on the generation of the two-sex stable population within which we calculate kinship matrices. It is well known that the intrinsic growth rate

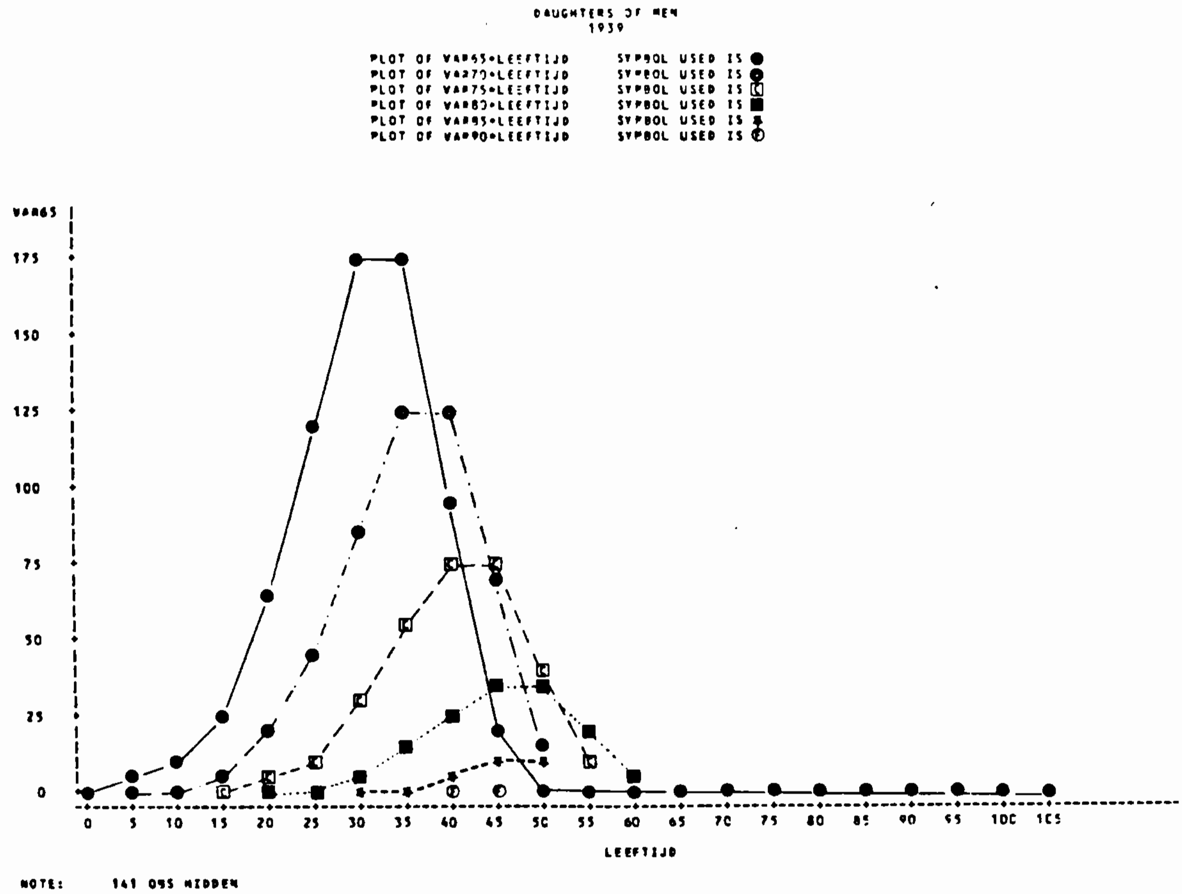


Figure 5: Absolute numbers of Daughters of men in a theoretical population with 1000 women over age 65

calculated through the combination of ${}^{f}f(x)^{f}$ with $l(x)^{f}$ will differ from that calculated with ${}^{m}f(x)^{m}$ and $l(x)^{m}$, if these fertility and mortality schedules are taken from a given calendar year for example. It is also clear that a two-sex stable population can not tolerate different growth rates for the sexes. The male and female growth rates in such a population must be equal, given a constant sex-ration at birth. In order to generate the two-sex stable population then, some stretch must be built into the components that go into the calculation of the growth rates for men and women. Where analytic solutions are hard to come by (see for example Das Gupta, 1972; Keyfitz, 1971), a pragmatic solution can often be found using iterative procedures on the computer. The approach initially chosen was to find the sex-ratio at birth which would give the same intrinsic growth rate if applied to the input fertility rates by sex. The sex-ratios with which this was accomplished lie slightly outside the range of empirical sex-ratios at birth. The numbers of persons in the male and female sectors of the population will then be affected by this minor deviation from real life populations. An alternative procedure was therefore applied, iterating not on the sex-ratio at birth, but on the male fertility level (keeping the pattern constant) , with a sex-ratio at birth of 105. The calculation procedure for Lotka's r we used was the Coale (1957) approximation with discrete data.

The procedure outlined in this chapter describes the way we calculated the numbers of persons there are in a theoretical population of a particular size, occupying different kinship relations towards each other. It would however be useful not only to have distributions of these relations by age, but also by order. How many persons are there with 3 children? How many have no direct nuclear relatives at all? In order to answer this kind of questions the matrices of kinship relations that apply on an aggregate were level were converted into a hypothetical population of individuals. These individuals possess the characteristics prescribed by the numbers calculated previously. We know how many brothers aged, say 50-54 women aged 65-69 have from their father's side. We know also how many persons there are in the corresponding age groups. The next paragraph describes how we proceeded in order to construct a hypothetical population in which the numerical relations worked out above apply, and in which we know who is who's relative.

5. Translating Aggregate measures into Kin Counts

The formal presentation of this part of the model can, by it's nature, not be in the form of mathematical expressions but will be in the form of a program, part of which is taken up as an annex. An informal description of what the program does is given here.

1. A personlist is made numbering down from the eldest person in the stable population, starting with the female sex. Thus all women in the 105-110 year age category have the lowest identification numbers, followed by the men of that age group, who in turn are followed by the women of the agegroup 100-105, and so forth. The lower boundary in our model is the 65-69 agegroup. The total number of persons in our elderly population is 911, 500 of whom are women. The total overall population of both sexes is 3363. The number of women in the over 65 age category for whom the calculations of numbers of kin were carried out in the two-sex stable population was 1000. This was reduced by half in order to sav on computing time.

2. A subroutine is run which assigns family members to each other on

basis of the numerical relations calculated as outlined in the previous paragraph. The first assignation which takes place is that of first spouses. There are no constraints here, except that each person may have only one first spouse. A random selection procedure is used to pick persons out of the agegroups dictated by the matrices calculated on the aggregate level.These are rounded to integers.The matrices refer to the approximation of the 2030 CBS forecast for the Netherlands.

3. Numbers of persons with deceased first spouse are added, on basis of the intensity of first marriage in the population. The property 'evmar' is assigned to random persons in the ages in question, such that the total number of persons with deceased and surviving first spouse equal the number of ever married persons in the population.

4. Daughters and sons are assigned to women. A random choice procedure of persons from the corresponding agegroups is used, with the following general constraints: a woman may not have more than 6 children of each sex, and not more than 2 in a single 5-year age group. This is done under three different conditions. The first scenario assumes that children are spread out as evenly as possible over the available ever married women. The second condition assumes that children are concentrated in as small a subcohort as possible within the general constraints mentioned above. The third variant applied a random assignation procedure of children to mothers. The parameter indicating these variants is the entropy H , which gives the degree to which kin are uniformly distributed over the available egoes.

5. Fathers are assigned to children. This entails a number of consistency checks with respect to the ages of spouses and children. The procedure was carried out under a number of values for the parameter S (for serial monogamy), indicating the probability in the assignation procedure, that a different man will be the father of a woman's next child. The program taken up in the annex gives the particulars of this procedure. It is conceived in such a fashion that a reader not familiar with LISP can follow the steps the program takes.

The type of output this model generates is given in table 4, which is a brief excerpt from the agelist of women of age 65-69. The list contains a number, which is a the name of the atom that represents the person. This atom has the properties listed thereafter: firsthusband, daughter ,son , mother, father and so forth. The variant represented here corresponds to the random assignation of children to mothers (entropy measure) with a probability of .2 that a woman has a different father for her next child than she had for the previous (serial monogamy measure). Annex 1 contains an illustrative analysis of the type of result attained by the model.

```
p00788(secondhusband p00678 son (p01833 p01987) firsthusband p00814 sex female age 65 idnr 785)
p00787(secondhusband deceased son (p02011) evmar yes sex female age 65 idnr 784)
p00786(secondhusband p01121 son (p02027 p02147 p02695) firsthusband p00902 sex female age 65 idnr 783)
p00785(evmar yes sex female age 65 idnr 782)
p00784(secondhusband p01057 son (p01997 p02416) daughter (p01899 p02138) evmar yes sex female age 65 idnr 781)
p00783(daughter (p01636 p01656) evmar yes sex female age 65 idnr 780)
p00782(secondhusband p00911 daughter (p02057 p02308) evmar yes sex female age 65 idnr 779)
p00781(firsthusband p00608 sex female age 65 idnr 778)
p00780(son (p01957) firsthusband p00851 sex female age 65 idnr 777)
p00779(secondhusband p00677 daughter (p01501 p01863 p02135) firsthusband p00646 sex female age 65 idnr 776)
p00778(divorced yes daughter (p01920) firsthusband p00623 sex female age 65 idnr 775)
p00777(secondhusband deceased son (p02383) daughter (p02501) firsthusband p00405 sex female age 65 idnr 774)
p00776(secondhusband p01289 son (p02152 p02573) daughter (p01889 p01895 p02320) firsthusband p00644 sex female age 65 idnr
p00775(secondhusband p00676 son (p01778 p01956) evmar yes sex female age 65 idnr 772)
p00774(son (p01996) evmar yes sex female age 65 idnr 771)
p00773(evmar yes sex female age 65 idnr 770)
p00772(daughter (p02093 p02079 p02330) firsthusband p00857 sex female age 65 idnr 769)
p00771(son (p02376) evmar yes sex female age 65 idnr 768)
p00770(secondhusband deceased son (p02167 p02381) evmar yes sex female age 65 idnr 767)
p00769(daughter (p02103) evmar yes sex female age 65 idnr 766)
p00768(son (p01982 p02241) mother p00076 daughter (p01717 p01868 p02059) evmar yes sex female age 65 idnr 765)
p00767(son (p01965 p02169) evmar yes sex female age 65 idnr 764)
p00766(father p00119 divorced yes mother p00031 daughter (p01875) firsthusband p01101 sex female age 65 idnr 763)
```

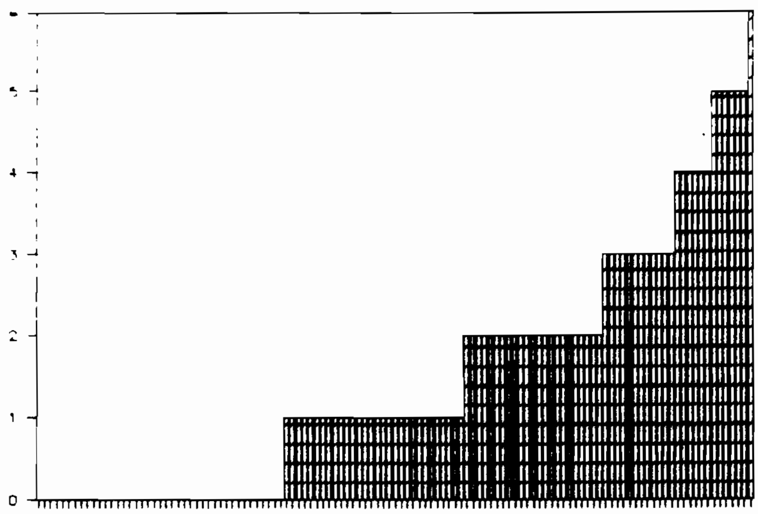
Table 4: Excerpt of personlist of simulation run Netherlands 2030, variant with random assignation of children to mothers, serial monogamy indicator .2

The procedure described permits the identification of each person in our hypothetical population with all his or her attributes. If for example a woman has a second husband, we may trace the person in question and look up his age, and his firstwife . We may also see whether he has children that he does not share with our reference person. That is to say that step-sons, half-brothers and such can be easily found and their numbers analyzed. The model has not yet reached it's final stage. A first version of the assignation of offspring to their respective parents has been completed. The assignation of siblings to eachother is still to be done. No analysis of results has yet been performed, except for the quick look at the distributions of step-family relations that you will find in annex 1.

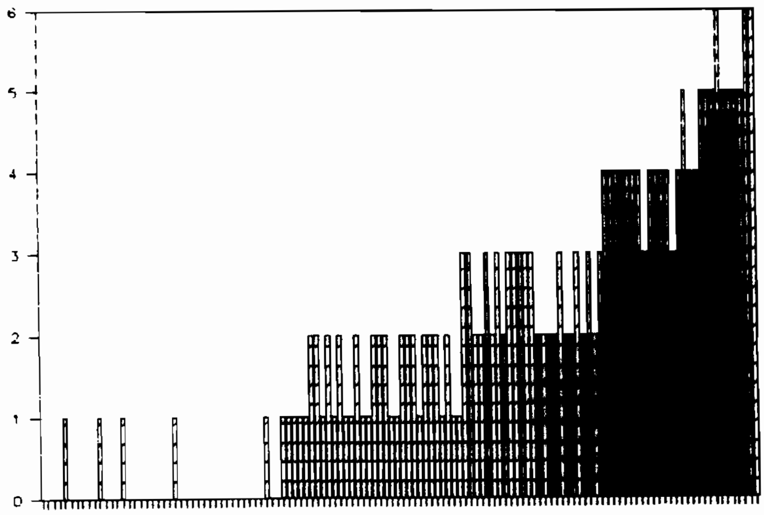
A point which has not been clarified is the entropy concept which we have used occasionally. Coleman (1965) and Willekens, Por , Raquillet (1979), have proposed the use of this classic measure in similar contexts. It is clear that a measure of the degree to which reproduction is concentrated in a subcohort (or equivalently, the degree to which it is "spread out" over the population) is of key importance for a model which wishes to simulate kin-counts. As Blum (1984) has demonstrated with a micro-simulation model using French data, it makes considerable difference for the distributions of kin if a subcohort of women would not participate, because for example they give priority to other goals in life, not considered to be compatible with childrearing . As pointed out, the model under consideration here does posses the possibility of manipulating and measuring this degree of concentration of childbearing. The measure to be used is the entropy H. Three entropy variants have been simulated, a maximum, a minimum, and a random assignation variant. The calculations of resulting entropies have not yet been performed, but they are non-problematical.

We have encorporated the male sex, and devised procedures to simulate step-family ties, but can not yet be satisfied with the simplistic set of assumptions used in the runs with our prototype model. The goal of constructing a satisfactory model of kinship formation lies far before us yet . We do feel however, that a combined macro-micro model such as the one described here might be of some use in reaching that goal.

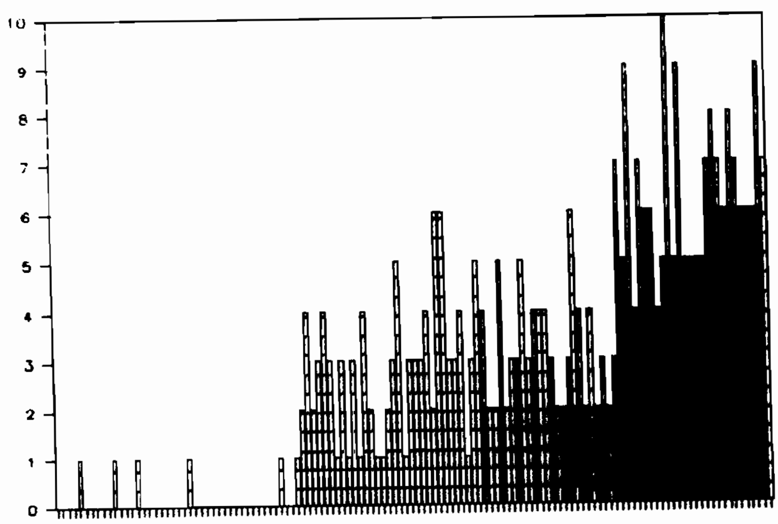
numbers of surviving children



numbers of children plus surviving first husbands



numbers of children, stepchildren and surviving spouses



Kin of women aged 65-69 in a theoretical population approximating the 2030 forecast, middle variant, CBS, 1984.

ANNEX I

Illustrative analysis of stepfamily relations

The graphs contained in this annex provide an illustration, for women aged 65 to 69, in our simulated population of the year 2030 of the degree to which an increase in the proportions of persons who enter into more than one reproductive union, under the conditions given in the LISP code attached, causes an increase in the size of kinship networks. The example refers to numbers of surviving (step-) children and spouses. In our simulated populations ties refer to numbers of kin intermediated through surviving direct relatives of ego. For example, the numbers of stepsons would include those from surviving husbands, but exclude those from deceased husbands. The assignation procedure used, assigned a woman's children to a surviving man whose first wife is still alive if and only if the male's first wife's youngest child was older than the second wife's eldest. This avoided having to find a second father for the children that the male's first wife had after he wed his second wife. In all other cases the child was assigned to a deceased man, the numbers of whom are known. It is clear that the less children a man's first wife has, the more likely it is that he will be considered as a suitable candidate for a second reproductive union in our model. It is not unrealistic to assume a negative relationship between the propensity to remarry and the number of children one has with one's first wife. In any case, if we attribute to all women with deceased husbands a number of step-children equal to what was attained on average by the women with surviving second husband, we are making a conservative estimate of the effect of a switch from strict monogamy to serial monogamy. This was done for the 2030 approximation, with the random assignation of children to mothers variant, S=40. The results are given in the graphs contained in this annex.

The x-axis of these graphs line up all women in the 65-69 age group, according to the number of children they had. These numbers are on the y-axis. If there were no remarriage some of these women would have surviving first spouses. These are added in the second graph. If the formation of second reproductive unions occurs however, we may also add the surviving second spouses to the woman's kinship network, plus all stepchildren. These are defined here as the children a woman's spouse has from another woman than herself. There is an increase in the numbers of relatives a woman has under such assumptions. The question to which degree the increase in numbers of kin through such cultural mechanisms might offset the decrease in numbers through biological mechanisms referred to in the text (figure 1), is a complex one and will not be addressed here.

The objective of this document is not to present results, but to indicate that the procedure outlined in the text is suitable for the purpose of generating the type of result that we are interested in. A more exhaustive discussion of the effect of altering the various parameters of the model, and possibly the incorporation of an alternative algorithm for the assignation of survivng second husbands to women will follow.

ANNEX II

A note on the Sister's Riddle

The issue of the expected number of siblings in a stable or stationary population appears in the demographic literature every now and again in the form of some apparent paradox. The riddle is then solved, only to reappear again later in a different guise. The matter has been referred to directly or indirectly by such distinguished demographers as Goodman, Keyfitz, Pullum (1974), Preston (1976), Goldman (1975), Hill, Trussel (1977), by Wachter (1980), Henry (1976), Le Bras (1982) and undoubtedly by many other authors, of whose contributions we are not aware. Our ambition will therefore not be to resolve the issue here. We do hope however to develop an internally consistent system of ideas, one where certain conclusions follow logically and necessarily from a number of postulates. To be more precise we shall develop two such systematic accounts, and compare them.

The GKP expressions for expected numbers of kin can be looked at in two different ways, as explained by Keyfitz (1977). The counting method is the first of these distinct perspectives. "A large population can be seen as developing according to given rules, and in effect we can make counts of the number of individuals having kin relations of interest." It is clear that this adequately describes the task we have set ourselves in the simulation model under consideration. As the original GKP article does, we loooked at the expressions from this angle. The result of this exercise is the first systematic account you will find in the following paragraphs.

The second way to look at the GKP expressions is probabilistic. "We can start by thinking of an individual and work out probabilities and expected values for his various kin.". This latter approach to the matter was followed in most of the derivations given in Keyfitz (1977). It was also followed by Krishnamoorty (1979), and by Goldman (1978). The second systematic account we give of the GKP expression for siblings follows this tradition.

As far as the pertinent mathematics are concerned, it has been performed by others. We may limit ourselves to a recollection of relevant results and discuss their implications. Preston (1976) and Keyfitz (1977) derive the following expression (abstracting from mortality):

AVSIB=AVPAR + VARPAR/AVPAR - 1 ....(1)

where AVSIB refers to the average number of siblings Ego has, not including Ego in the average, AVPAR is the average number of children ever born per woman and VARPAR is the variance of this distribution of terminal parities. Both the measures refer to terminal distributions, that is to numbers of offspring to women who have completed childbearing, and to numbers of siblings at ages beyond the age at which one's mother has completed childbearing. This is a classic result, which requires no further comment.

We shall now elaborate our first account. As a starting point for the model we are constructing, we adopt the simplest set of assumptions possible. These are:

- *Homogeneity: all persons are subjected to the same age-specific demographic regimes: instantaneous mortality and fertility rates by age.*
- *Independence: the occurrence of birth at a particular age is independent of the occurrence of birth at a different age.*
- *Continuity: there are continuous streams of increment and decrement leading to positive real-valued relations between points in a continuous space.*
- *Determinism: there is no variance around expected values.*

Although it might be argued that the GKP model applies to a broader set of circumstances than those embodied in these restrictive assumptions, there is no doubt whether the expressions are valid in such a context. Ego is thus considered as a reference point, from whose point of view the kinship space, so to speak, is overseen. The treatment of maternal reproduction in the GKP estimation procedure for sisters amounts to adding up " a continuous stream of fractions of child", as Wachter (1980) pointedly formulated it. The stream in question is uninterrupted. There is no provision for the birth of Ego. It is clear then, that what the expression gives is an estimate of sibship size, that is the average number of sisters (in a one-sex female population). If we abstract from mortality, the average sibship size is evidently equal to the average number of children ever born in a deterministic stable population. At the risk of boring the reader we shall write this out in terms of the expressions given above. Let SIB=AVSIB + 1 refer to the average sibship size, then

SIB= AVPAR + VARPAR/AVPAR ....(2) and since VARPAR=0, SIB=AVPAR. The output of the GKP program, confirms this result. The average numbers of sisters ever born are practically identical to the average numbers of daughters ever born, at ages where there are no further increments to numbers of persons in these dyadic relations.

Under the assumptions chosen, this relation is necessary: it always applies. There are no assumptions with respect to distributions of women by terminal parity, except that all women have the same number of children ever born. This sentence forms the prelude to our second account of the GKP expressions for expected numbers of sisters. The assumptions we adopt are somewhat broader than those chosen above. We adopt a parity-specific version of the homogeneity assumption: to all woman of all parities the same fertility rates apply. This makes it possible to look at the reproductive process as a series of failure time distributions, which fits into the probabilistic framework. Note the difference between this asumption and that used above. We are now referring to parity specific fertility being identical, which implies that a range of distributions of births by order will result. The assumption that there is no variance around the expected values can thus not be made. The other assumptions underlying this account are synonimous to those given above. We have the following assumptions:

- *Parity-Specific Homogeneity: To all women for each order of birth the same instantaneous fertility rates apply, and to all persons the same instantaneous mortality rates apply.*
- *Independence (same as above).*
- *Continuity (same as above).*

Under these conditions we must consider the outcome of the GKP expressions for sisters as AVSIB, and not of SIB, as in our first account. Since AVSIB is a function of the mean as well as the variance of the distribution of women by numbers of children ever born, it can only apply under certain conditions with respect to these distributions. If the condition of independence is combined with that of parity-specific homogeneity, and we neglect mortality for the sake of the argument, Krisnamoorty (1979) has demostrated that the probabilty of having exactly r children follows the Poisson distribution. Since in a Poisson distribution the variance and the mean are equal, VARPAR/AVPAR becomes 1. The assumptions of the model, thus formulated imply a particular distribution, under which,

AVSIB=AVPAR + 1 - 1 ....(3)

Once again we have a relation in which the average numbers of siblings at ages where one's mother has stopped having children is equal to the average number of children one has, but now it reads AVPAR=AVSIB. The numerical results confirm the perspective.

It thus appears that there are (at least) two different consistent settings in which the GKP expressions are valid. The first interpretation came natural to us, in view of the objective of the model, which was to construct a theoretical kin-count

Let us imagine that an analyst would like to try to settle this issue: "Does the GKP expression for sisters give AVSIB or SIB ?", through the execution of a micro-simulation exercise. According to this reasoning, any micro-simulation exercise using the assumptions underlying account 2, would reproduce the numbers generated by the GKP program, which might lead to the conclusion that the 'correct' interpretation is that resulting from the probabilistic viewpoint. If on the other hand more complex assumptions would be used in the simulation, resulting in distributions of numbers of children ever born by mothers over 50, in which the variance and mean are not equal, the results would diverge from those the GKP program produced. It is therefore not surprising that Wachter's (1980) results did not replicate the relations Goldman (1978) postulated. Wachter attributes this fact to sampling variability. It would be interesting, in the light of the viewpoint given here, to have information on VARPAR. We also recall that the sibling-survival methods originally worked out in terms of regression coefficients in the Hill-Trussel computer-runs, did not find their way into standard demographic techniques for mortality estimation. Their orphanhood and widowhood results of course did. The reason for the inconsistent behavior of sibling survival in mortality estimation might very well be that an important independent variable was left out: VARPAR. Preston (1976) presents empirical material for the United States which suggests that there are no a reasons to believe that the relation VARPAR/AVPAR should generally be close to unity. The Le Bras 1982 simulation results confirm this broad conclusion. We shall leave the issue of relationships between macro and micro models for what they are, for the moment.

The conclusion of this note is that there is no unique model for kinship, and that there is indeed no unique concept of a given model. From different postulates follow different conclusions, leading to different systematic accounts. Within these systems there should be no inconsistencies: all relations should be necessary, and follow undeniably from the basic assumptions made. Both concepts of the GKP model appear to be valid according to these criteria. After discussing the hypotheses and their implications, a postcript should follow on the interpretation of the model's outcome in relation to reality. We have said some things about modelling but very little as yet about what it is we are supposed to be modelling. This is unsatisfactory. What we really need at this moment are kin-surveys. It will be much easier to discuss the validity of our models, if we have a better idea of their relation to the real thing. These surveys will be held no doubt, and they will inform us of the near future. A body of empirical material for the recent past which might be subjected to a secondary analysis in this light are for example survey results which were used to make indirect estimates of mortality and fertility. I am confident that we will soon be in a better position to relate models to each other and to measurements of kinship in our societies. The present paper attempts to combine the strong points of macro and micro models in a straightforward fashion. It is to be seen whether these results stimulate further developments in that direction.

ANNEX III

Lisp Source Code for Assignation of Fathers to Children

"DEFINITION ENVIRONMENT; FILE WHOSMYDADDY
This file defines the functions needed for the assignation of fathers to children on basis of the children previously assigned to mothers. The parameters used are the personlist with offspring assigned to women plus an indicator of serial monogamy, S. This is simply the probability that a woman's next child will be conceived by another man than the man who conceived the previous child. The functions in this file can be run in LISP by doing (givetheladiestheirmates personlist 100) for example, which applies to the strict monogamy case. It is assumed that the personlist with this is done contains persons with previous properties assigned.

The functions presented here are not ordered in the sequence in which they are called. It is essential upon reading the program to know what is called where. In order to make this transparent the mother function is referred to as A and the functions she calls as A1, A1 etc, while functions called elsewhere would be marked as for ex. A11, A12, etc."

"A33

The following function assigns a newhusband to a wife and vice-versa."

```
(defun givewomanman (whoremarried newhusband )
  (putprop whoremarried newhusband 'secondhusband)
  (putprop newhusband whoremarried 'secondwife))
```

"A32
This function assigns a father to a child and vice-versa."

```
(defun givechildafather (sonordaughter whosthefather whosthechild)
 (putprop whosthechild whosthefather 'father)
 (putprop whosthefather (cons whosthechild (get whosthefather sonordaughter))
                                             sonordaughter)); since he can
                                             ;have )1 son or daughter we
                                             ;cons the kid on to the list.
```

"A391
This function assigns a mother to a child and vice-versa."

```
(defun givechildamother (sonordaughter whosthemother whosthechild)
 (putprop whosthechild whosthemother 'mother)
 (putprop whosthemother (cons whosthechild (get whosthemother sonordaughter))
                                             sonordaughter)); since she can
                                             ;have )1 son or daughter we
                                             ;cons the kid on to the list.
```

"A35
The decision whether a woman's next husband is to be a living man or a deceased man is taken by drawing a random number with chances proportional to the number of women with and without surviving firstmates.It returns t if the man is alive and nil otherwise. This function uses variables bound in the function bindmenofagewifenow, which is called in the loop that encloses the function whodidiremarry.

At higher ages where most women are widows the choice will ususlly be among deceased men."

```
(defun nexthusbandalive (whoremarried)
  (if (( (random (+ (length (eval menofagewifenow))   ;menofagewifenow evaluates
                                                      ;to for example menofagewife65
                                                      ;which we eval to get the list.
              (eval deadmenofagewifenow)))
           (length (eval menofagewifenow))) t ))
```

"A34
This function is called 4 times in this file. Three times it is used to alter a matrix definitively (see it's use in whodidiremarry, givethemanrestofherkids) and once it is used to alter a matrix for the course of a given function (see cleanup).

The function fixpapasmatrix deduces the child assigned to a man from the matrix of patrilineal sons or daughters. Once the father is identified it is known from which cell in the matrix in question to subtract the child. All entries still in the matrix after the children by matrilineal descent are assigned, are assumed to have a mother who has died. These children are assigned to their fathers with kinmatrix. The function returns the altered matrix. Does not alter anything as a side effect.

If this function is called by whodidiremarry in the process of subtracting a child from a samehusband's matrix, then it may return a list rather than a matrix. This list is used as a flag to have the calling function subtract the child from her mother's list. No further action is taken."

```
(defun fixpapasmatrix (kidsage papasage psd)
                                                   ;last arg must be for ex papasdatos or
  (prog (pasd)
  (setq row (- (quotient kidsage 5 ) 3 ))          ;papassons.
  (setq col (- (quotient papasage 5 ) 13 ))        ;which are names
  (cond ((< col 0)(setq pasd psd)(return psd)))
  (setq newentry ( - ( psd row col) 1 ))           ;of data matrices.
  (store (psd row col ) newentry )                 ;recall that arrays dimensions
                                                   ;flag is to return to whodidiremarry if
                                                   ; samehusband unacceptable
  (cond                                            ;start at 0,0 and not at 1,1
    ((< newentry 0)                                ;check whether newentry <0
      (cond                                        ;if so see
       ((= flag '(fixing for samehusband in whodidiremarry))  ;if this is the case; if so

        (store (psd row col) (1+ newentry))             ;restore previous entry,
        (setq pasd '(samehusband cannot have this child))    ;prepare this message
        (return pasd)))                                 ;and end cond with a return, sending mesg.
                                                        ;in other cases signal error
      (patom "msg from fixpapasmatrix   ")(patom " treating ")(print sonordaughter)(terpr)
      (patom "kidsage papasage ")(print kidsage)(patom "  ")(print papasage)(terpr)
      (patom "  there's a negative number of kids in row   ") ;and correct it ifnecessary
      (print row )(patom "    col  ") (print col )(terpr)     ;which may happen if call is from
      (patom "  starting from 0,0 as usual. ")                ;givethemanrestofherkids
      (patom "  That number is  " )(print newentry)(terpr)
      (store (psd row col) (1+ newentry)) (passthechildon thischild)
      (setq pasd psd)(return pasd)  ))                        ; ends one-clause-cond
 (setq pasd psd)
 (return pasd )
                         ))                            ;prog, defun
```

"A31
The next function gives t if a randomdraw from 1-100 is < S . The value of S should be an integer from 1 to 100 , giving the procent chance that a woman's next child will have another father than the previous."

```
(defun drawwithpropS (S)
  (setq randomnumber (random  100))
  (cond
    ((< randomnumber  S) t)
    ( t nil )))
```

"A1
We bind some variables we need using other variables we made in the dynamic file with the function given hereafter. The dynamic file makes variables such as menofagewife65, menofagewife70 etc. These are set equal to menofagewifenow for use as a variable later."

```
(defun bindmenofagewifenow (agewife)
 (setq menofagewifenow
  (implode  (append (explode 'menofagewife) (explode agewife))))
 (setq deadmenofagewifenow
  (implode (append (explode 'deadmenofagewife)(explode agewife)))))
```

"A362
This function finds out whether a man's firstwife's youngest is older than
his potential secondwife's eldest child. If so the secondwife is acceptable.
The function returns nil or thismansok. The function is called in gettherightguy
which looks for an appropriate surviving husband. Under txThe condition embodied in the
function given here we do not have to look for another husband for the man's
firstwife. That is, she may have a secondhusband assigned to her in the course
of the program, but is not forced to, by virtue of the fact that we took her
firsthusband away from her and left her with some younger children without a husband.
Other solutions would have been possible here.
"

```
(defun fiwisyoungestolder (firstwoman secondwife)
 (patom "checking whether firstwife's youngest is older than secondwife's eldest")(terpr)
 (prog (thismansok)
  (cond                                  ;if second wife has no kids there's noproblem.
   ( (null sortedkids) (setq thismansok '(thismansok)) ; so make it something non-nil
                       (return thismansok))) ; sortedkids was bound in the calling
  (setq sewiseldest (last sortedkids))       ; function givetheladiestheirmates.
  (setq fiwisdaughters (get firstwoman 'daughter) fiwissons (get firstwoman 'son))
  (cond                                  ;find 1stwifes youngest with a cond in a cond.
   (( and (null fiwisdaughters) (null fiwissons))(setq thismanok '(thismansok))
                                             (return thismansok))  ;which is non-nil
                                         ;if 1swife childless no problem, otherwise find youngest
   (t (setq youngestdato (last fiwisdaughters) youngestson (last fiwissons));do this
      (cond ((null youngestdato)(setq fiwisyoungest youngestson))          ;and see
            ((null youngestson)(setq fiwisyoungest youngestdato))          ;whos youngest
            ((< (get youngestdato 'age)(get youngestson 'age))(setq fiwisyoungest youngestdato))
            (t (setq fiwisyoungest youngestson))))   ;ends inside cond
                                           )     ;ends outside cond
  (cond                                  ;if fiwisyoungest age >= sewiseldest we accept
   ((not (< (get fiwisyoungest 'age)(get sewiseldest 'age)))(setq thismansok 'thismansok)
                                                 (return thismansok))
   (t (setq thismansok nil)(return thismansok)))     ;ends cond
                                   ))            ;prog, defun
```

"NOT CALLED
The next function takes an age of woman and does three things:

it lists their firsthusbands, counts the number of women with a deceased
firsthusband and those who never married. It returns the list of surviving
men and as a byproduct binds the variables deadmenofagwife and
nevermarriedofage. This funtion calls persons-agesex-list, which is
defined in the file helpfun. It is called in the dynamic file longpapaskids."

```
(defun menofagewife (agewife)
  (setq menofagewife nil)
  (setq deadmenofagewife 0)
  (setq nevermarriedofage 0)
  (setq ladiesthatage (persons-agesex-list agewife 'female personlist))
  (do
   ((i 1 (1+ i)))
    ((= i (1+ (length ladiesthatage))) menofagewife )
    (setq thislady (nthelem i ladiesthatage))
    (cond
     ((member 'firsthusband (plist thislady) )
       (setq menofagewife (cons (get  thislady  'firsthusband)
                               menofagewife)))
     ((member 'evmar (plist thislady))
       (setq deadmenofagewife (1+ deadmenofagewife)))   ;the deceased husbands of surviving ladies
     (t(setq nevermarriedofage (1+ nevermarriedofage)))
                                              ))) ;cond,do&defun
```

```
"A
The following function forms the enclosure within which the kernel is run.
It's a do-loop over all women, and for each woman over all children. It calls
only sortbyage, bindmenofagewifenow and whodidiremarry"

(defun givetheladiestheirmates (personlist S)
 (setq flag nil)
 (do ((ageyouwant 65 (+ 5 ageyouwant)))             ;for all ages over 65
    ((= ageyouwant 110)(terpr)(patom "WE ARE THROUGH GIVETHELADIESTHEIRMATES"))
    (setq theladiesweworkwith (persons-agesex-list ageyouwant 'female personlist))
    (bindmenofagewifenow ageyouwant)
    (setq agesecondwife ageyouwant)                 ;used in whodidiremarry
    (do (( jjj 1 (1+ jjj)))                         ;and for each woman in the agegr.
       ((= jjj (1+ (length theladiesweworkwith))) (terpr);who are potential secondwives, ie who remarry (or not).
          (patom "we gave newhusbands to ladies aged  ") ;there's a message
          (print ageyouwant))                       ;at the end of each agegr
       (setq ageofmarmale 110)
       (setq whoremarried (nthelem jjj theladiesweworkwith))
       (setq herdatos (get whoremarried 'daughter))
       (setq hersons  (get whoremarried 'son))
       (setq herkids (append hersons herdatos))
       (setq sortedkids (sortbyage herkids))
       (setq howmanykidsdoeshehave (length sortedkids))
       (setq whichoneofthekids 1)
       prog                                         ;the inner loop
loep                                                ;over all of her children
          (setq whichchild (nthelem whichoneofthekids sortedkids))
          (if (= whichchild nil) (go nextwomaninline))
          (cond
            ((= (get whichchild 'sex) 'male)(setq sonordaughter 'son ))
            ((= (get whichchild 'sex) 'female)(setq sonordaughter 'daughter))
            (t (error"msg from givetheladiestheirmates:whichchild must be male or female")))
          (setq restofherkids (nthcdr whichoneofthekids sortedkids))
          (setq keyvalue (whodidiremarry whoremarried S whichchild sonordaughter restofherkids))
          (cond
           ((= keyvalue 'newhusband)(go newman))
           ((= keyvalue 'deadhusband)(go deadman))
           ((= keyvalue 'samehusband)(go sameman))
           (t )  (patom keyvalue )
                (error "whodidiremarry is returning something strange ")) ;ends cond
sameman
           (patom "she keeps her husband")(terpr)
           (setq whichoneofthekids (1+ whichoneofthekids))
           (if () whichoneofthekids howmanykidsdoeshehave)
                (go nextwomaninline))

           (go loep)
deadman
           (patom "she gets a man who has deceased ")(terpr)
           (setq whichoneofthekids (1+ whichoneofthekids))
           (if () whichoneofthekids howmanykidsdoeshehave)
                (go nextwomaninline))
           (go loep)
newman
           (patom "she gets a secondhusband")(terpr)
nextwomaninline
           (patom "            we move on to the next lady ")(terpr)(terpr)
                                             )    ;which ends the jjj do loop
                                             )    ;which ends the 1st do loop
                                             )    ;goes with defun
```

```
"A39
This fuction takes a child out of a woman's property list, if in whodidiremarry
the decision is made that the child cannot be attributed to the father it  is
supposed to have. Then it passes the child on to a mother the same age who
fits the constraints of the model. If we reach the end of the list of women
without finding an appropriate mother, the function breaks, permitting the
introduction of an ad hoc solution."

(defun passthechildon (thechild)
 (remprop thechild 'mother)                          ;this is no longer the mother, so
 (cond                                               ;take thechild away from her
  ((= sonordaughter 'son)(setq hersons(remove thechild hersons))
                        (putprop whoremarried hersons 'son))
  ((= sonordaughter 'daughter)(setq herdatos (remove thechild herdatos))
                             (putprop whoremarried herdatos 'daughter))) ;cond
 (prog ()                                            ;and give it to somebody else

 (setq newindex jjj)
nextladymaybe
  (setq newindex (1+ newindex))                      ;next in line
  (setq referenceperson  (nthelem newindex theladiesweworkwith))
  (cond                                              ;if we run out of ladies
     ((= referenceperson nil )
       (debug "msg from passthechildon: we're at the end of our list of ladies: you're in debug mode") (return)))
                                                     ;stops program execution, permits intervention.
                                                     ;make assignments. A simple way to do this is by finding
                                                     ;an evmar same-age lady with (= 1 child and then using
                                                     ;(givechildamother sonordaughter theevmarlady thechild)
                                                     ;and do OK to resume processing.
                                                     ;This is a rare event. Normally the program
                                                     ;finds a candidate and checks whether
                                                     ; she can be the new mother
  (setq refplist (plist referenceperson))            ;who must have been married once
  (if (and (not (member 'evmar refplist))(not (member 'firsthusband refplist)))(go nextladymaybe))
  (setq referencedaughters (get referenceperson 'daughter)) ;and fit the constraints wrt children
  (setq referencesons (get referenceperson 'son))
  (setq referencekids (append referencesons referencedaughters))
  (cond                                                 ;cond 1
    ((= (length referencekids) 0)                       ;
     (givechildamother sonordaughter referenceperson thechild) ;
     (return)))                                         ;no kids no prob
  (setq referenceage (get thechild 'age))               ;with kids you want
  (setq thenumberofkidsthatage (do (( numero 1 (1+ numero)) ;to know how many which age
                                    (total 0))          ;var init step
                                 ((= numero (1+ (length referencekids))) total)
                                 (cond                  ;cond 2 is do-body
                                  ((= (get (nthelem numero referencekids) 'age) referenceage)
                                  (setq total (1+ total)))
                                                    )))  ;ends cond 2, do, setq
  (cond                                                ;cond 3
   ((and    (< (length referencesons) 6)
            (< (length referencedaughters) 6)
            (< thenumberofkidsthatage 2))              ;if all that's true
                (givechildamother sonordaughter referenceperson thechild)
                (return))                              ;
   (t (go nextladymaybe))                              ;otherwise
                                    )))                ;ends cond 3,prog,defun



"A37
Take somebody out of a list."
(defun taketheguyout (whototakeout ofwhichlist)
 (setq ofwichlist (remove whototakeout ofwhichlist)))
```

"A361
This function is called in gettherightguy.
We remove men from the list menofagewifenow, who have ages in which no children are found, thus avoiding that the matrices papassons and papasdatos acquire negative numbers in the process. This would occur if we subtracted the child from a zero cell entry in those matrices. A list is returned which is equal to the initial list with the non-eligible males removed.

If a woman has in the restofherkids say 3 sons in a given agegr say 65, and if there is 1 child in the respective cell for father's age, we must avoid the following from happening. For each child we check whether the papassons matrix can accomodate him and every time we find a positive number. If we subtract all three children however, the entry would become -2. This is avoided by working with parametermatrices. These are copies of the datamatrices used only within the environment of the function. Every time we call the function for a new woman the parametermatrices are set to the value that papassons or datos currently has.
This is accomplished by copymat.
No definitive alterations of the matrices of children by age of father are carried out here.

Recapitulating:
We locate an entry in papassons or papasdatos. If there's a 0 there, all men of the corresponding age are temporarily removed from the pool. If there's not a 0 there, the entry is temporarily reduced by 1.
"

```
(defun cleanup (menofagewifenow parameterkids)      ;parameterkids are bound to restofherkids in the call
 (setq parameterboys (copymat papassons) parametergirls (copymat papasdatos))
 (prog (menforthischild)                            ;for all her younger kids
 (setq kid 0 menforthischild  menofagewifenow)      ;initialize. Note menofagewifenow evaluated in call
 (setq  parameterkids (cons whichchild parameterkids)) ;we consider younger kids from whichchild down
outsideround
  (setq fathersage 60)                              ;initialize
  (setq kid (1+ kid))                               ;increase index
  (if (> kid (length parameterkids )) (return  menforthischild)) ;findout if you're finished
  (setq kiddy (nthelem kid parameterkids))          ;if not , take  a kid
  (setq kidsage (get kiddy  'age))                  ;and find it's age
  (setq kidsrow (- (quotient kidsage 5 ) 3 ))       ;set corresponding row index
  (setq fatherscol (minus 1))                       ;initialize col index,  then
  (cond                                             ;get the right matrix to check
   ((= (get kiddy 'sex) 'male ) (setq parametermat parameterboys))
   ((= (get kiddy 'sex) 'female )(setq parametermat parametergirls))
   (t (error "msg from cleanup :kiddy's sex must be male or female")))
middleround                                         ;check parametermat
   (setq  fatherscol (1+ fatherscol))               ;for zero entries
   (if( = fatherscol (last (arraydims parametermat))) ;if you're through the row of mens ages
      (go outsideround))                             ;ends the if statement
   (setq fathersage (+ fathersage 5))                ;otherwise update father's age
   (cond                                             ;and see whether
     (( = (parametermat kidsrow fatherscol) 0 )      ;there's a zero; if so
     (innerround )                                   ;call an inner function with no arguments
                                              )      ;ends 1st clause of cond. So if there's no 0
     (t (setq parametermat                           ;remove the child for the next check to be valid
        (fixpapasmatrix kidsage fathersage parametermat)))
                                              )     ;ends cond
(go middleround)
(go outsideround)                                   ;after you've done that take the next child
                                              ))    ;prog & defun
```

"A3612
This function works as an innerloop in the previous one. It goes through menforthischild and takes out the men with the age that has no sons or daughters."

```
(defun innerround ()
 (do ((thismarmale 1 (1+ thismarmale)))       ;and go through menforthischild
     ((> thismarmale  (length                 ;except if you are at  the end
                         menforthischild)))

                                              ;then go to next col. Otherwise see if there are men of that age
     (cond
      ((= (get (nthelem thismarmale  menforthischild) 'age ) ;men of that age have no kids that age
              fathersage)                     ;so if that's the age
       (setq menforthischild (taketheguyout    ;then take
                   (nthelem thismarmale  menforthischild) ;thisfellow
                   menforthischild))          ; out of this list
                           ))                 ;end of one clause cond
                              )               ;do
                              )               ;defun
```

```
"A38
If you find a man who fulfills the requirements, the rest of his secondwife's
 children are assigned to him, ie all her younger children."

(defun givethemanrestofherkids (thisman restofherkids)
  (do ((k 1 (1+ k )))
      ((= k (1+ (length restofherkids)))
   (patom "we gave a man restofherkids")(terpr))
  (setq thischild (nthelem k restofherkids)) ;the body of the loop starts here.
   (cond
     ((= (get thischild 'sex) 'male)(setq sonordaughter 'son pasonsordaters papassons));bind ORIGINAL
     ((= (get thischild 'sex ) 'female)(setq sonordaughter 'daughter pasonsordaters papasdatos));data matrices
     (t (error "msg from givethemanrestofherkids: thischild's sex must be male or female")))
   (givechildafather sonordaughter thisman thischild)
   (if () (get thisman 'age) 60)
   (setq pasonsordaters                                           ;and change these
   (fixpapasmatrix (get thischild 'age) daddysage pasonsordaters))) ;originals
                               ))   ;do, defun
```

"A3
The next function is applied to each child of each women in our personlist. It returns samehusband, newhusband or deadhusband and as a side-effect carries out the assignations secondhusband, secondwife, daughters to men and sons to men. It may als return thisisnothischild, which will be explained later. If it returns deadhusband this may mean that she remarried a man who has died, or that the child is asigned to the same deceased man that she was previously tied to. The assignation-changes referring to the first union are taken care of in the function gettherightguy. If a newhusband is chosen he gets all the woman's subsequent children, because we limit the number of consecutive reproductive unions to 2 in this simulation.

If the function returns thisisnothischild the cannot have samehusband as a father, while the woman must have the next child assigned to the samehusband, according to a decision taken in the program. In this case the child is assigned to another woman. This way all husband-wife and child-parent dyadic relations are kept in the numerical proportions calculated previously.

First we decide whether a woman is to have the same man assigned to her as the father of her children that was the father of her previous children, and act accordingly with respect to the assignation of the son, daughter, father, secondhusband properties. The decision is taken on basis of a parameter S, which gives the probability that a woman's next child will have another father.

A man can not have children overlapping in age from two different women in serial monogamy. The kind of man you would like to find as a father is one who's firstwife's youngest child is older than his second wife's oldest. The function gettherightguy looks for such a man.

This function is the kernel of the do-loop used to give the women secondhusbands and the children their fathers. It therefore calls a rather large number of modular functions. These are:

drawwithpropS	:to decide whether a woman remarries or not.
nexthusbandalive	:to decide whether a woman's 2ndhusb is to be chosen :from among the living or the dead.
passthechildon	:to assign the child to another woman if inconsistent :with a firsthusband
gettherightguy	:to find a surviving man with thr right characteristics.
givechildafather	:assign children to secondhusband and vice-versa.
givewomanman	:assign secondhusband to women and vice versa.
fixpapasmatrix	:to subtract an assigned child from the matrix of :children to be assigned.
taketheguyout	:to remove an unsuitable candidate out of a list.
givethemanrestofherkids	:once 2ndwife found 2ndhusb gets her successive kids.

"

```
(defun whodidiremarry (whoremarried S whichchild sonordaughter restofherkids)
 (prog (thisistheman)
  (setq sonordaughtersage (get whichchild 'age))
  (setq samehusbandas (get whoremarried 'firsthusband ))
  (setq herprops (plist whoremarried))
   (cond                                              ;ORIGINAL datamatrices are bound
    ((= sonordaughter 'son)(setq pasonsordaters papassons)) ;var pasonsordaters
    ((= sonordaughter 'daughter )(setq pasonsordaters papasdatos)) ;is used in fixpapasmatrix call.
    (t (error "msg from whodidiremarry:  sonordaughter must be either son or daughter  ")))


  (cond
   ((drawwithpropS S)(go newhusband)) ;decides whether she stays with her husband
   (t (go samehusband)))              ;or gets a new one.

samehusband
   (cond ((= samehusbandas nil)(setq thisistheman 'deadhusband) ;if the woman isevmar, ie firsthusb has died,
                            (return thisistheman)))         ;there is no survivor to assign kids or wife to
   (setq flag '(fixing for samehusband in whodidiremarry))
   (if ()(get samehusbandas 'age) 60)                       ;if the man's in the elderly age range
   (setq matrixormessage
        (fixpapasmatrix sonordaughtersage (get samehusbandas 'age) pasonsordaters))) ;change ORIGINAL datamatrix
   (cond                                                    ;if fixpapasetc. returns this either the husband has to be changed
     ((= matrixormessage '(samehusband cannot have this child))  ;or the child. We chose for the latter.
     (passthechildon whichchild)(setq flag nil) (setq thisistheman 'thisisnothischild)
      (return thisistheman))                                ;if we get the matrix back
      (t (setq pasonsordaters matrixormessage))      )       ;ends cond
   (givechildafather sonordaughter samehusbandas whichchild)    ;no givewomanman, they're already tied.
   (setq thisistheman 'samehusband) (return thisistheman)

newhusband
   (if (nexthusbandalive whoremarried) (go pickaliveman)
                                       (go getadeadman))
pickaliveman
   (setq thisistheman (gettherightguy  (eval menofagewifenow) whoremarried agesecondwife ))
   (if (= thisistheman '(deadman))(go getadeadman))
   (set menofagewifenow
    (taketheguyout thisistheman (eval menofagewifenow))) ;so he cannot be chosen as a
                                                    ;2ndhusband again in the same agegr.
   (givechildafather sonordaughter  thisistheman whichchild)
   (givewomanman whoremarried thisistheman)     ;assign 2nd husb & wife props
   (setq pasonsordaters                         ;change ORIGINAL datamatrices definitively
     (fixpapasmatrix sonordaughtersage (get thisistheman 'age) pasonsordaters))
   (setq daddysage (get thisistheman 'age))

   (givethemanrestofherkids thisistheman restofherkids) ;where restofherkids is bound in the calling function
   (setq thisistheman 'newhusband)
   (return thisistheman)

getadeadman
   (setq thisistheman 'deadhusband)
   (putprop whoremarried 'deceased 'secondhusband) ;no fixpapasmatrix which is for living kids of living men.
   (set  deadmenofagewifenow (1- (eval deadmenofagewifenow))) ;eg 1-  deadmenofage65. Note use of set not setq
   (cond ((( deadmenofagewife 0) (patom "msg from whodidiremarry: neg nr deadmenofagewife for    ")
                            (print deadmenofagewifenow)(terpr))) ;ends cond
   (return thisistheman)


                          ))         ;prog,defun


"A3611
This function makes a new matrix with the same contents as the input
matrix; that is the output matrix is equal, but not eq to the matrix
it gets as a parameter. It is called by cleanup."

(defun copymat (matrix)
 (setq dim (arraydims matrix))
 (setq newmat (*array 'newmat (car dim) (cadr dim) (caddr dim))) ;note: *array evaluates arguments, array doesn't
 (fillarray newmat (listarray matrix))
 newmat)
```

"A363
Here we look for an ever married man, first in the same agegroup as whichage.
If we can't find such a man in the agegroup in question we go one agegroup lower
until we find a man who qualifies. If we run out of ever married men, which can happpen
only in high S cases (100 for example), then we go back to gettherightguy with
the message that we must take a dead man. If there are no
ever married men in a given agegroup, then we are informed of the fact, and
we take a man in a younger agegroup. We only take a deadman if we reach the 65
age group without finding an available evmarmale."

```
(defun findanevmarmale (wheretolook whichsex)
 (prog (anevmarmale )
outsideloop
  (setq ii 1)
  (setq oneofthesepersons wheretolook )
insideloop
  (cond
    ((= ii (1+ (length oneofthesepersons))) (patom "no evmarmale in this agegroup  ")
                                        (terpr)
                                        (go youngermen)))    ;ends cond
  (setq thisfellow (nthelem ii oneofthesepersons))
  (setq thisfellowsprops (plist thisfellow))
  (cond
   ((and (member 'evmar thisfellowsprops)
        (not (member 'secondwife thisfellowsprops)))
    (setq anevmarmale thisfellow)
    (return anevmarmale))
   (t (setq ii (1+ ii)))
                         )    ;ends cond
  (go insideloop)
youngermen                                   ;this should occur only at high ages
 (setq whichage ( -  ageofmarmale 5))      ;if all available evmar men already
 (setq wheretolook (persons-agesex-list whichage whichsex personlist))
 (go outsideloop)                           ;have a 2nd wife.
                        ))       ; prog, defun
```

"A36
This function gettherightguy is called in didiremarry above. It
makes sure the man chosen as a second husband does not have children from
his firstwife that are younger than the one that is going to be assigned
to him as child of his secondwife.

The function returns a man with the characteristics required. It is stochastic
in the sense that a random age is chosen during the procedure.

A two step procedure is followed:

1. A list of the firsthusbands of the women in the reference woman's agegroup
is made. For example, if we are treating a particular woman of age 65, a list
is made of all men firstmarried to women that age. From this list we will try
to select a secondhusband. Such a selection would entail giving this man all
this woman's younger children. We must therefore remove from our list
all men of ages where the total number of children of the age in question is
smaller than the number of children of that age the woman has. For example if
a woman has a child of age 65 and there are no children of age 65 for men of
age 75, then all men of age 75 must be taken out of our pool of selection.
This operation is performed by the function cleanup. If there are no men at all
from whom we can choose, then the newhusband must be a deadman.

2. Depart from the clean list. The first man out of the list of firsthusbands is chosen.
A check is performed to find out whether he fits the criterium that his firstwife's
youngest child is older than his potential secondwife's eldest. If we find such
a man he is chosen. If not the next man is processed. If we reach the end of the
list without finding a man that qualifies, we assume his firstwife is dead. The
secondwife is then chosen from men with the property evmar. They are ever-married,
but have no living firstspouse. See the function findanevmarmale . Another function
called from here is fiwisyoungestoldest which performs the age-check referred to.
The variables pasonsordaters and sortedkids are assumed to be bound."

```
(defun gettherightguy (menofagewifenow secondwife agesecondwife)
 (prog (wefoundafather)
 (patom "we are cleaning up for so many kids:  ")(print (1+ (length restofherkids)))(terpr)
 (setq menforthischild                          ;specific for each child under consideration
 (cleanup menofagewifenow restofherkids))        ;restofherkids bound  in givetheladiestheirmates
 (cond                                          ;if no men available go back
  ((null menforthischild) (setq wefoundafather '(deadman))(return wefoundafather)))
 (setq j 0 )
  (setq testvalue (1+ (length menforthischild)))
loop                                            ;start at the beginning of the list
 (setq j (1+ j))                                ;and go through it one by one.
 (if (= j testvalue) (go takeaguywithadeadfiwi) ) ;if you reach the end do this
 (setq wetakethisman (nthelem j menforthischild ));otherwise look for a man who fits description
  (cond                                         ;a man with a 2nd wife already
   ((member 'secondwife (plist wetakethisman))    ;is reported
      (patom "msg from gettherightguy: this man's been married before  ")
      (patom "something's wrong with the loop, since we threw such men out; CHECK)
      (error "we stopped the search for a suitable man"))) ;and the process jammed.

  (setq firstwife (get wetakethisman 'firstwife )); this always returns non-nil
  (setq ageoffirstwife (get firstwife 'age ))     ;since he is one of menofagewife.

  (if (fiwisyoungestolder firstwife secondwife )(go takethisman ) )
  (go loop)
takethisman                          ;someone else gets her 1st husb so she's divorced.
  (putprop firstwife 'yes 'divorced )  ;all separated women are then the divorced
                                     ;women plus those with a secondhusband. Here
                                     ;separated means that a reproductive union is terminated.
  (setq wefoundafather wetakethisman )(return wefoundafather)
takeaguywithadeadfiwi
  (patom "taking a man who's first spouse has deceased.")(terpr)
  (setq compare ageofmarmale)                                  ;stochastic is the following:
tooyoung
  (setq ageofmarmale (get (pick-a-person menforthischild) 'age));a random age from the distr of ages of
                                                               ;eligible firsthusbands of these ladies.
  (if (< ageofmarmale 65)(go tooyoung))                          ;remarriage of dead men over 65 only
  (cond
  (( not (= ageofmarmale compare))   ;if the age's the same as that of the previous list avoid doing it again
  (terpr)(patom "looking in agegroup ")(print ageofmarmale)
  (patom " for an evmar male as husband of a woman aged ")
  (print agesecondwife)(terpr)
  (setq wheretolook (persons-agesex-list ageofmarmale 'male personlist));men with that age
                                          )) ;ends the one-clause cond
  (setq wefoundafather (findanevmarmale  wheretolook 'male))           ;an evmar male with that age
  (return wefoundafather)))                         ; ends prog, defun
```

```
"A2
This little recursive function sorts a woman's children by age, from young
to old. It calls findtheyoungest, which in turn calls last."

(defun sortbyage (herchildren)
  (cond
    ((null herchildren) nil )
    (t ( cons (findtheyoungest herchildren)
            (sortbyage (remove (findtheyoungest herchildren) herchildren))))
                               ))     ;cond, defun
"A21"
(defun findtheyoungest (herchildren)
  (do (( elt 1 (1+ elt))
       ( youngest (last herchildren)))             ;var, init, step.
     (( = elt (1+ (length herchildren))) youngest)   ;endtest: stop when you're through the list.
     (if (< (get (nthelem elt herchildren) 'age)
            (get youngest 'age))
         (setq youngest (nthelem elt herchildren)))
                                  ));do & defun.


"A211
This is the last function in this file"
(defun last (laat)
  (car (reverse laat)))
```

ANNEX IV

References

Adams,R ,(1971);"The Nature of the Family", in 'Kinship',ed. J.Goody, Penguin, London.

Barr, A, Feigenbaum,E, (1982);"The Handbook of Artificial Intelligence", Pitman, London

Bartlema, J, Vossen, A; (1984), "Modelling Institutions in a de-institutionalizing society", Workshop on Modelling Household Formation, NIDI, 6 dec. 1984

Bartlema, J, de Jong, P; (1985),"Shrinking Kinship Support Networks in the Netherlands, the numerical results of a simulation.", Catholic University of Tilburg, The Netherlands

Bartlema, J,Van Nimwegen, N, Moors, H; (1986), "Kleinere Families, verdwijnende verwanten en de zorg voor bejaarden.",DEMOS, NIDI, The Hague

Blum, A. ;(1984),"Fecondite, Solidarite Intergenerationelle, Isolement", Association Internationelle de Demographie Francaise, Geneve, Sept 1984

Bongaarts, J ;(1982) ,"The formal Demography of Families and Households, An Overview, The Population Council, New York.

Bongaarts, J ; (1981),"Simulation of the Family Life Cycle",IUSSP, International Population Conference, Manilla.

CBS, Cruissen, H,(1984),"Forecast of the Netherlands, 1984-2030",Staatsuitgeverij, The Hague

Coale,A, (1957);"A new method for calculating Lotka's r, the intrinsic rate of increase in a stable population", Population Studies, 11

Coleman, J ;(1964),"Introduction to Mathematical Sociology", The Free Press, London.

Das Gupta, (1972),"On two-sex models leading to stable populations",Theoretical Population Biology, 3, 1972

Firth,R,(1948), "We the Tikopia", cited in Schneider, 1965

Goodman, Keyfitz, N, Pullum, T.,(1974),"Family Formation and the Frequency of various Kinship Relationships.",Theoretical Population Biology, 5, 1974.

Henry, L ;(1976),"Population, Analysis and Models", Edward Arnold, 1976

Hill, K,Trussel, J; (1977),"Further Developments in Indirect Estimation.", Population Studies, Vol 31., No 2.

Hyrenius,H, Adolfsen, I; (1964)," A fertility simulation model ", Almquist & Wicksell, Stockholm

Hyrenius H., Adolfsen I,, Holmberg,I; (1968)," Demographic Models: second report, DM2 ", Elanders Boktryder, Aktiebolag, Goteborg

Holmberg,I (1968),"Demographic Models, DM4", Almquist & Wicksell, Stockholm

Keyfitz, N. ;(1977), "Applied Mathematical Demography", Springer, New York

Keyfitz, N. ;(1984), "The Demography of the Family", Workshop on Modelling Household Formation, NIDI, Voorburg, The Netherlands

Keyfitz, N. ;(1971), "The Mathematics of Sex and Marriage", Proceedings of the 6th Berkely Symposium on Mathematical Statistics and Prabability", June 1971, University of California Press

Keyfitz, N; (1986), "Canadian Kinship Patterns based on 1971 and 1981 Data", IIASA, Laxenburg, Austria

Krishnamoorty, S, (1979), "Family Formation and the Life Cycle", Demography, Vol 16, 1979

Le Bras,H. ,(1973), "Parents, Grands Parents, Bisaieux, Population, Jan., 1973

Le Bras, H, (1982) ; "Evolution des liens de Famille au cours de l'Existence", VII7eme Colloque Nationale de Demographie, Strasbourg, May 1982

Madan,A.; (1986), "A formalization of Ryder's nuclear family Theory", University of Western Ontario, Canada

Preston, S ; (1976), "Family Sizes of Children and Family Sizes of Women", Demography, vol 13, 1976

Schneider, D, (1965), "Kinship and Biology", in 'Aspects of the Analysis of Family Structures', M.Levy cs, Priceton University Press

Van de Walle, E ,(1968); "Marriage in African Censuses and Surveys", in W.Brass cs, 'The Demography of Tropical Africa', Princeton University Press

de Vos,S, Palloni,A,(1984),"Formal Methods for Analyzing Kinship and Household Organization", ASA, meeting, August 29, San Antonio, Texas

Wachter, K, (1980); "The sister's Riddle and the importance of Variance when guessing demographic rates from kin counts", Demography, Feb, 1987

Willekens,F., Shah,J, Ramachandran,P, (1982);" Multi State Analysis of Marital Status Life Tables: Theory and Application.", Population Studies, 36

Weigkricht, E, Winkelbauer, L, (1986), "Knowledge Based Systems - Overview and selected Examples", WP-86-xx, International Institute for Applied Systems Analysis, Laxenburg, Austria (forthcoming)

Willekens, F, Por, A, Raquillet, R, (1979), "Entropy, Multiproportional and Quadratic Techniques for Inferring detailed Migration Patterns from aggregate data.", IIASA, working paper.

Winston, P.H, Horn, B.K.P, (1981), "LISP", Addison Wesley, Reading, MA

Wolf, D, (1986),"Simulation Methods for Analyzing Continuous-Time Event History Models", IIASA, Laxenburg, Austria