Evaluation of moments of ratios of quadratic forms in normal variables and related statistics

Magnus, J.R.; Pesaran, B.

Publication date:
1990

entER
for
mic Research

Discussion paper

TILBURG UNIVERSITY

No. 9019

EVALUATION OF MOMENTS OF RATIOS OF QUADRATIC FORMS IN NORMAL VARIABLES AND RELATED STATISTICS

by Jan R. Magnus
and Bahram Pesaran

March 1990

ISSN 0924-7815

Evaluation of moments of ratios of quadratic forms in normal variables and related statistics

by

Jan R. Magnus
London School of Economics
and
CentER, Tilburg University

and

Bahram Pesaran
Economics Division, Bank of England

January 1990

Keywords: Ratios of quadratic forms; Calculation of expectations; Tests for existence

ABSTRACT

In this paper we present and describe a subroutine, called QRMOM, which calculates the exact moments of certain functions of ratios of quadratic forms in normal variables. If we let $R = x'Ax/x'Bx$, then QRMOM can calculate,

for $s \geq 1$, $E[R^s]$, $E[R^s(a'x)]$, and $E[R^s(x'Cx)]$,

where x is an nx1 vector of normally distributed variables with some mean μ and a positive definite (hence non-singular) variance-covariance matrix Ω. A, B (positive semi-definite) and C are nxn symmetric matrices and a is an nx1 vector.

The subroutine QRMOM is based on theory developed by Magnus (1986,1989), who showed that these expectations can all be expressed as single integrals and also worked out necessary and sufficient conditions for the existence of expectations in each of the above three cases. QRMOM checks for the existence of a requested s-th moment before carrying out the calculations and if the specified s is too high, it will adjust downwards (if posssible) so that existence of moments is assured.

A diskette containing the Fortran Code of QRMOM and some test programmes is available on request.

Language

Fortran 77

Description and Purpose

The subroutine QRMOM calculates the exact moments of a ratio of two quadratic forms x'Ax and x'Bx and related expectations. Letting R=x'Ax/x'Bx, QRMOM can handle the following three cases:

(1) $E\left[R^s \right]$ $s \geq 1$

(2) $E\left[R^s (a'x) \right]$ $s \geq 1$

(3) $E\left[R^s (x'Cx) \right]$ $s \geq 1$.

Here x is an nx1 vector of normally distributed variables with some mean μ and a positive definite (hence non-singular) variance-covariance matrix Ω. A, B (positive semi-definite) and C are nxn symmetric matrices and a is an nx1 vector. In case (2) when $\mu=0$ a simple symmetry argument shows that the expectation vanishes if it exists. QRMOM does not calculate the expectation in that case, but simply sets it equal to zero.

The above calculations are required in a variety of situations when econometric or time series estimators take the form of a ratio of quadratic forms in normal variables. The calculation of moments, forecast bias or mean-square forecast error relating to such estimators will involve the evaluation of expectations of the form (1), (2) or (3). Examples are provided by the estimation of an AR(1) process with or without an intercept, stationary or non-stationary. See for example Hoque, Magnus and Pesaran (1988) and Magnus and Pesaran (1988, 1989).

The subroutine QRMOM is based on theory developed by Magnus(1986,1989), who showed that these expectations can all be expressed as single integrals and also worked out necessary and sufficient conditions for the existence of expectations in each of the above three cases. QRMOM checks for the existence of a requested s-th moment before carrying out the calculations and if the specified s is too high, it will adjust s downwards (if possible) so that existence of moments is assured.

Parameter statements

The following parameters have been set in subroutines QRMOM and PARINT and function F:

NDIM=50	The dimension of various work spaces. NDIM≥n where n is the number of observations.
ISPAR=77 & ISDIM=12	The dimensions of the array ISPRTN (ISPARxISDIM) where all possible partitions for a particular s are stored. This two dimensional array is set up by subroutine PARINT.
MAXMOM=24	The maximum of s allowed.

If n>50 is to be specified, the relevant parameter statements for NDIM should be increased accordingly.

If 12<s≤24 is to be calculated, then both ISPAR and ISDIM should be increased.

Working out expectations for s>24 is possible. In that case not only ISPAR and ISDIM should be changed in the parameter statements, but also MAXMOM. Furthermore, the DATA statement in subroutine PARINT should be extended to include MAXMOM numbers. In the data statement the vector NUM has been set up to contain the number of all possible partitions of numbers up to 24. If a MAXMOM bigger than 24 is specified, the data statement for NUM should be extended accordingly. For a table containing the partitions for integers up to 100 see Hall(1986, p.38).

Common statements

Since the NAG routine D01AMF for the evaluation of the integral requires the function F(x) to have only one argument, the other arguments needed in the calculation of F(x) are passed through two labelled common areas QRREAL and QRINT.

Structure

SUBROUTINE QRMOM(ICASE, NOBS, IS1, IS2, A, B, C, ELA, IEMU, EMU, IOMEGA, OMEGA, ITEM, ISMAX, RESULT, ABSERR, IFAIL)

Formal Parameters

ICASE	integer	input:	1, 2 or 3 corresponding to (1), (2) or (3) above.
NOBS	integer	input:	No of observations n
IS1	integer	input:	order of the lowest moment required.
		output:	unchanged unless IS1 $\le$0 in which case IS1 is set equal to one.
IS2	integer	input:	order of the highest moment required.
		output:	unchanged unless IS2>M where M=MIN(ISMAX,MAXMOM,ISDIM) in which case IS2 is set equal to M.
A	real array of dimension at least NOBSx(NOBS+1)/2	input:	symmetric matrix in (1), (2) or (3). Only the lower part of A is stored as a_{11},a_{21},a_{22},a_{31},a_{32}, a_{33} etc.
B	real array of dimension at least NOBSx(NOBS+1)/2	input:	symmetric positive semidefinite matrix B in (1), (2) or (3). Only the lower part of B is stored.
C	real array of dimension at least NOBSx(NOBS+1)/2	input:	symmetric matrix C in (3). Only the lower part of C stored. No need to assign values to C in cases (1) and (2) though storage should be allocated to it.
ELA	real array of dimension at least NOBS	input:	a in (2). No need to assign values to a in cases (1) and (3) though storage should be allocated to it.

IEMU	integer	input:	=0 if μ=0 ≠0 if μ≠0
EMU	real array of dimension at least NOBS	input:	vector μ. Values required only if IEMU≠0 though storage should be allocated to it.
IOMEGA	integer	input:	= -1 if L^{-1} is supplied where Ω = LL', L lower triangular = 1 if L is supplied where Ω = LL', L lower triangular = 2 if Ω is supplied
OMEGA	real array of dimension at least NOBSx(NOBS+1)/2	input:	either L or L^{-1} or Ω where only lower part of these are stored
ITEM	integer	output:	if ICASE=1, ITEM indicates which condition in Theorem 1 of Magnus(1989) holds ie 1≤ITEM≤3 if ICASE=2, ITEM indicates which condition in Theorem 2 of Magnus(1989) holds ie 1≤ITEM≤5 if ICASE=3, ITEM indicates which condition in Theorem 3 of Magnus(1989) holds ie 1≤ITEM≤7
ISMAX	integer	output:	the maximum of s in (1), (2) or (3) for which these expectations exist. ISMAX=100 indicates that the expectation exists for every s.
RESULT	real array of dimension at least IS2-IS1+1	output:	the required expectations stored as: RESULT(1)=IS1-th moment RESULT(2)=(IS1+1)th moment ...... RESULT(IS2-IS1+1)=IS2-th moment
ABSERR	real array of dimension at least IS2-IS1+1	output:	the absolute error in calculating each of the expectations. Stored as RESULT.

IFAIL	integer	output:	a fault indicator where: 0: no error 1: NOBS > NDIM or NOBS ≤ 1 2: ICASE out of range 3: IOMEGA out of range 4: Eigenvalues of B could not be calculated 5: B is not pos. semi-definite 6: B is the null matrix 7: if IOMEGA=2 Ω not pos. definite if IOMEGA=1 diagonal elements of L not all positive if IOMEGA=-1 diagonal elements of L^{-1} not all positive 8: if IOMEGA=2 or 1, L can't be inverted if IOMEGA=-1, L^{-1} can't be inverted 9: eigenvalues of L'BL could not be calculated 10: L'BL is not pos. semidefinite 11: L'BL is the null matrix 12: IS1>IS2 or moments in the adjusted range do not exist or ISDIM in the parameter statement is too small 13: ISPAR in the parameter statement is too small 14-19: error in calculating the integral corresponding to IFAIL=1 to 6 in the NAG library routine D01AMF

Auxiliary Algorithms

QRMOM uses two routines from the NAG library, namely F02ABF (calculation of eigenvalues and eigenvectors of a real symmetric matrix) and D01AMF (evaluation of a single integral). F02ABF is called by subroutine EVALUE and D01AMF is called by subroutine INTGRL. Users who wish not to use these two NAG library routines can substitute their own versions of subroutines EVALUE and INTGRL.

In addition to subroutines EVALUE and INTGRL, QRMOM calls various functions and subroutines:

(F1)	REAL*8 FUNCTION F(X):	used in calculating the integral
(F2)	FUNCTION INX(I,J):	picks out the appropriate element of a symmetric matrix stored in lower triangular form
(F3)	FUNCTION NFACT(N):	calculates N!
(S1)	SUBROUTINE POWER:	called by F(X)
(S2)	SUBROUTINE CALCRA:	called by F(X)
(S3)	SUBROUTINE INIT:	initializes all matrices and vectors and checks for the existence of expectations
(S4)	SUBROUTINE EXIST:	called by INIT
(S5)	SUBROUTINE PARINT:	constructs the matrix containing all partitions of an integer
(S6)	SUBROUTINE SEP:	decomposes A (pos. def.) into LL' (L lower triangular) and replaces A by L

(S7)	SUBROUTINE LOWINV:	inversion (in place) of a lower triangular matrix
(S8)	SUBROUTINE MULT1:	computes C=B'AB (A symmetric, B lower triangular)
(S9)	SUBROUTINE MULT2:	computes C=B'AB (A symmetric)
(S10)	SUBROUTINE NULL1:	checks to see if AB=0 (A symmetric)
(S11)	SUBROUTINE NULL2:	checks to see if B'AB=0 (A symmetric)

Constants

The DATA statement in QRMOM sets EPS = 1.0-11 as a small number. Any number with an absolute value below EPS will be treated as zero.

Precision

The version of the routines listed below is in double precision (Real*8). In order to change the program to single precision the following changes should be made:

(1) Change all IMPLICIT REAL*8 to IMPLICIT REAL*4

(2) Change the constants in the DATA statements to single precision versions.

(3) Change DSQRT, DABS, DEXP to SQRT, ABS, EXP in the statement functions appearing in the beginning of routines F, EXIST, SEP, LOWINV, NULL1, NULL2.

Time

In order to work out the relationship between the CPU time and n and s we did some calculations on the VAX 6330 at the London School of Economics for different combinations of s (1 to 8) and n(10,20,30,40 & 50). In Table 1 the results of regressions of LOG(CPU) as a function of n and s for different values of ICASE and IEMU are reported. These regressions enable us to estimate the absolute CPU time (in seconds) for different combinations of n, s, ICASE and IEMU. Furthermore, if we differentiate the regression equations with respect to n or s we should be able to work out the % increase in CPU time as a result of an increase in n or s. For example if we differentiate the regression equation for ICASE=1 and IEMU=0 with respect to s we have

$$\frac{\partial \log(CPU)}{\partial s} = 2.172 + .003868*n - 2*.3701*s - 2*.0005736*n*s + 3*.0229*s^2.$$

In particular, when n=40 and s=4, we obtain

$$\frac{\partial \log(CPU)}{\partial s} = .2816.$$

This implies a growth in CPU time of about 28.2%. This should be compared with an increase in actual CPU time on the VAX from 380.2 seconds to 500.4 seconds which implies an increase of 31.6%. Hence using the relationships summarized in Table 1, once the timing of a particular combination of ICASE, IEMU, n and s is known, we can work out the marginal increase in CPU time. We estimate that the error in estimating the percentage change in CPU time is less than 15%.

Table 1. Results of least-squares regressions of LOG(CPU) as a function of n and s for different values of ICASE and IEMU

	ICASE=1 IEMU=0	ICASE=1 IEMU=1	ICASE=2 IEMU=1	ICASE=3 IEMU=0	ICASE=3 IEMU=1
Constant	-4.9046	-4.6367	-4.5340	-4.5108	-3.5408
n	0.2946	0.2967	0.3191	0.3182	0.2827
s	2.1720	2.0809	1.8911	1.8457	1.6686
n^2	-0.004397	-0.004623	-0.005783	-0.004956	-0.004182
ns	0.003868	0.004229	0.0070239	-	-
s^2	-0.3701	-0.3565	-0.3135	-0.2963	-0.2578
n^3	0.0000272	0.0000299	0.0000459	0.0000319	0.0000248
n^2s	-	-	-0.0000981	0.0000532	0.0000838
ns^2	-0.0005736	-0.0005799	-	-0.000583	-0.0008435
s^3	0.0229	0.0222	0.0176	0.0185	0.0167
$\bar{R}^2$	0.9951	0.9959	0.9957	0.9964	0.9960
SE of reg.	0.1393	0.1248	0.1302	0.1171	0.1169

Accuracy

The data statement in QRMOM sets ESPABS=1.0-5 (absolute error) and EPSREL=1.00-4 (relative error) in the calculation of the integrals. These can be changed in order to achieve different levels of accuracy.

In order to check the accuracy of calculations we used QRMOM to evaluate the first 4 moments of the F distribution with 4 and 16 degrees of freedom. This was achieved by using an arbitrary (20x4) matrix R and by setting $A=R(R'R)^{-1}R'$,B=I-A and Ω=I. Setting ICASE=1 and IEMU=0 the first 4 moments of x'Ax/x'Bx were calculated. Since

$$F_{4,16} = 4(x'Ax/x'Bx),$$

we obtain

$$\mu'_s = E(F_{4,16})^s = 4^s \; E(x'Ax/x'Bx)^s .$$

The exact values for μ'_s can be calculated using the formula in Kendall and Stuart(1977, exercise 16.1 p. 423), which implies

$$(4) \qquad E(x'Ax/x'Bx)^s = \frac{\Gamma(2+s)\;\Gamma(8-s)}{\Gamma(2)\;\Gamma(8)} .$$

Table 2 allows the comparison between the exact results using (4) and the calculated values using QRMOM for s=1 to 4.

Table 2. Accuracy of moments of an F distribution calculated by QRMOM

s	Exact	Calculated
1	2/7	0.28571429
2	1/7	0.14285714
3	4/35	0.11428571
4	1/7	0.14285714

Examining Table 2 shows that in this example an accuracy of at least eight decimal points is achieved.

REFERENCES

Hall, M.(1986), Combinatorial Theory(2nd edition), John Wiley and Sons, New York.

Hoque, A., Magnus, J. R. and Pesaran, B.(1988), 'The exact multi-period mean-square forecast error for the first-order autoregressive model', Journal of Econometrics, 39, 327-346.

Kendall, M. G. and Stuart, A.(1977). The Advanced Theory of Statistics, Vol I, fourth edition, Charles Griffin & Co., London.

Magnus, J. R.(1986), 'The exact moments of a ratio of quadratic forms in normal variables', Annales d'Economie et de Statistique, 4, 95-109.

Magnus, J. R.(1989), 'On certain moments relating to ratios of quadratic forms in normal variables: Further results', Sankhya, Series B, to appear.

Magnus, J. R. and Pesaran B.(1988), 'The bias of forecasts from a first-order autoregression', submitted for publication.

Magnus, J. R. and Pesaran B.(1989), 'The exact multi-period mean-square forecast error for the first-order autoregressive model with an intercept', Journal of Econometrics, 42, 157-179.

```
      SUBROUTINE QRMOM(ICASE,NOBS,IS1,IS2,A,B,C,ELA,IEMU,EMU,
     +IOMEGA,OMEGA,ITEM,ISMAX,RESULT,ABSERR,IFAIL)
C
C       ALGORITHM AS??? APPL. STATIST. (1990)
C       WORKS OUT IS1-TH TO IS2-TH MOMENTS RELATING TO
C       RATIOS OF QUADRATIC FORMS IN NORMAL VARIABLES
C
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(NDIM=50,ISDIM=12,ISPAR=77,MAXMOM=24)
      PARAMETER(NSYM=NDIM*(NDIM+1)/2)
      DIMENSION A(*),B(*),C(*),ELA(*),EMU(*),OMEGA(*),
     +RESULT(*),ABSERR(*)
      DIMENSION WMAT1(NDIM,NDIM),WMAT2(NDIM,NDIM),WORK(NSYM)
      DIMENSION WGRL(2000),IWGRL(252)
      COMMON/QRINT/ICODE,IMU,N,IS,ISROW,ISPRTN(ISPAR,ISDIM)
      COMMON/QRREAL/RLANDA(NDIM),SMU(NDIM),SLA(NDIM),
     +AA(NSYM),CC(NSYM)
      EXTERNAL F
      DATA ZERO,ONE,EPS,INF/0.0D0,1.0D0,1.D-11,100/
      DATA EPSABS,EPSREL/1.D-05,1.D-04/
      IFAIL=0
      ICODE=ICASE
      IMU=IEMU
      IOM=IOMEGA
      N=NOBS
C
C       CHECK THE EXISTENCE OF EXPECTATIONS AND
C       INITIALIZE ALL MATRICES AND VECTORS
C
      CALL INIT(ICODE,N,NDIM,A,B,C,ELA,IMU,EMU,
     +IOM,OMEGA,AA,RLANDA,CC,SLA,SMU,WMAT1,WMAT2,WORK,
     +ITEM,ISMAX,ZERO,ONE,EPS,INF,IFAIL)
      IF(IFAIL.NE.0) RETURN
```

```
      IFAIL=12
      IF(IS1.GT.IS2) RETURN
      IS1=MAX(IS1,1)
      IS2=MIN(IS2,ISMAX,MAXMOM,ISDIM)
      IF(IS1.GT.IS2) RETURN
C
C       WORK OUT IS1-TH TO IS2-TH MOMENTS
C
      DO 1000 IS=IS1,IS2
C
C       WORK OUT ALL PARTITIONS OF IS
C
      CALL PARINT(IS,ISPRTN,ISPAR,ISDIM,ISROW,IFAIL)
      IFAIL=IFAIL+11
      IF(IFAIL.NE.11) RETURN
C
C       CALCULATE THE IS-TH MOMENT AS AN INTEGRAL
C
      CALL INTGRL(F,ZERO,EPSABS,EPSREL,RES,ABSE,
     +WGRL,IWGRL,IFAIL)
      IFAIL=IFAIL+13
      IF(IFAIL.NE.13) RETURN
      FACT=NFACT(IS-1)
C
C       STORE RESULTS
C
      RESULT(IS-IS1+1)=RES/FACT
      ABSERR(IS-IS1+1)=ABSE/FACT
1000  CONTINUE
      IFAIL=0
      RETURN
      END
```

```
      REAL*8 FUNCTION F(X)
C
C       EVALUATION OF F AS A FUNCTION OF X
C       USED IN THE CALCULATION OF THE INTEGRAL
C
C       NOTE: ALL OTHER ARGUMENTS OF F ARE IN
C             COMMONS QRINT AND QRREAL
C
      IMPLICIT REAL*8(A-H,O-Z)
      PARAMETER(NDIM=50,ISDIM=12,ISPAR=77)
      PARAMETER(NSYM=NDIM*(NDIM+1)/2)
      PARAMETER(ISSYM=ISDIM*(ISDIM+1)/2)
      COMMON/QRINT/ICODE,IMU,N,IS,ISROW,ISPRTN(ISPAR,ISDIM)
      COMMON/QRREAL/RLANDA(NDIM),SMU(NDIM),SLA(NDIM),
     +AA(NSYM),CC(NSYM)
      DIMENSION VLA(NDIM),VMU(NDIM),R(NSYM),GAMMA(NSYM),
     +WORK1(NSYM),WORK2(NSYM),PP(ISSYM),POW(NDIM,ISDIM),
     +TR1(ISDIM),TR2(ISDIM),OUT(3),NK(ISDIM)
      DATA ZERO,HALF,ONE,TWO,FOUR/0.0D0,0.5D0,1.0D0,2.0D0,4.0D0/
      ZSQRT(H)=DSQRT(H)
      ZEXP(H)=DEXP(H)
      F1=X**(IS-1)
      F2=ONE
      TRACE=ZERO
      DO 10 I=1,N
      SPI=ONE+TWO*X*RLANDA(I)
      SPI=ONE/ZSQRT(SPI)
      F2=F2*SPI
      IF(IMU.NE.0) VMU(I)=SMU(I)*SPI
      IF(ICODE.EQ.2) VLA(I)=SLA(I)*SPI
      DO 10 J=1,I
      SPJ=ONE+TWO*X*RLANDA(J)
      SPJ=ONE/ZSQRT(SPJ)
      SP=SPI*SPJ
```

```
      IJ=INX(I,J)
      R(IJ)=AA(IJ)*SP
      IF(ICODE.EQ.3) THEN
        GAMMA(IJ)=CC(IJ)*SP
        IF(I.EQ.J) TRACE=TRACE+GAMMA(IJ)
      END IF
10    CONTINUE
      F3=ONE
      IF(IMU.NE.0) THEN
        F3=ZERO
        DO 20 J=1,N
        SP=SMU(J)*SMU(J)-VMU(J)*VMU(J)
20      F3=F3+SP
        F3=ZEXP(-HALF*F3)
      END IF
      CALL POWER(ICODE,IMU,R,GAMMA,VMU,TR1,TR2,
     +POW,WORK1,WORK2,N,NDIM,IS,ZERO,ONE,TWO)
      IF(IMU.NE.0) THEN
        IF(ICODE.NE.1) THEN
          SUM=ZERO
          DO 40 I=1,N
          IF(ICODE.EQ.2) THEN
            SUM=SUM+VLA(I)*VMU(I)
          ELSE IF(ICODE.EQ.3) THEN
            TT=TWO
            DO 30 J=1,I
            IF(I.EQ.J) TT=ONE
30          SUM=SUM+TT*VMU(I)*GAMMA(INX(I,J))*VMU(J)
          END IF
40        CONTINUE
          TRACE=TRACE+SUM
        END IF
        DO 70 L=1,IS
        RL=L
```

```
      SUM1=ZERO
      SUM2=ZERO
      DO 60  I=1,N
      RP=POW(I,L)
      SUM1=SUM1+VMU(I)*RP
      IF(ICODE.EQ.2) THEN
        SUM2=SUM2+VLA(I)*RP
      ELSE IF(ICODE.EQ.3) THEN
        DO 50 J=1,N
50      SUM2=SUM2+RP*GAMMA(INX(I,J))*VMU(J)
      END IF
60    CONTINUE
      TR1(L)=TR1(L)+RL*SUM1
      IF(ICODE.EQ.2) THEN
        TR2(L)=SUM2
      ELSE IF(ICODE.EQ.3) THEN
        TR2(L)=TR2(L)+TWO*SUM2
      END IF
70    CONTINUE
      IF(ICODE.EQ.3) CALL MULT2(GAMMA,POW,PP,N,IS,NDIM,ZERO)
    END IF
    F4=ZERO
    DO 90 I=1,ISROW
    IB=1
    DO 80 L=1,IS
    K=ISPRTN(I,L)
    NK(L)=K
    IF(K.NE.0)IB=IB*NFACT(K)*((2*L)**K)
80  CONTINUE
    IB=(NFACT(IS)*(2**IS))/IB
    RB=IB
    LCODE=ICODE
    IF(ICODE.EQ.2.AND.IMU.EQ.0) LCODE=0
    IF(ICODE.EQ.3.AND.IMU.EQ.0) LCODE=2
```

```
      IF(LCODE.NE.0) CALL CALCRA(LCODE,IS,NK,TR1,TR2,PP,OUT,ZERO,
     +ONE,TWO)
      IF(LCODE.EQ.0) THEN
        RA=ZERO
      ELSE IF(LCODE.EQ.1) THEN
        RA=OUT(1)
      ELSE IF(LCODE.EQ.2) THEN
        RA=OUT(1)*TRACE+TWO*OUT(2)
      ELSE IF(LCODE.EQ.3) THEN
        RA=OUT(1)*TRACE+TWO*OUT(2)+FOUR*OUT(3)
      END IF
      F4=F4+RA*RB
90    CONTINUE
      F=F1*F2*F3*F4
      RETURN
      END

      SUBROUTINE POWER(ICODE,IMU,R,GAMMA,VMU,TR1,TR2,POW,
     +WORK1,WORK2,N,NDIM,IS,ZERO,ONE,TWO)
C
C       CALLED BY FUNCTION F(X)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(*),GAMMA(*),WORK1(*),WORK2(*)
      DIMENSION VMU(*),TR1(*),TR2(*)
      DIMENSION POW(NDIM,*)
      NN=N*(N+1)/2
      DO 90 L=1,IS
      IF(L.EQ.1) THEN
        DO 10 I=1,NN
10      WORK1(I)=R(I)
      ELSE
        DO 30 I=1,N
        DO 30 J=1,I
```

```
      IJ=INX(I,J)
      SUM=ZERO
      DO 20 K=1,N
      IK=INX(I,K)
      KJ=INX(K,J)
20    SUM=SUM+R(IK)*WORK1(KJ)
30    WORK2(IJ)=SUM
      DO 40 I=1,NN
40    WORK1(I)=WORK2(I)
    END IF
    SUM=ZERO
    DO 50 I=1,N
    RI=WORK1(INX(I,I))
50  SUM=SUM+RI
    TR1(L)=SUM
    IF(IMU.NE.0) THEN
      DO 70 I=1,N
      SUM=ZERO
      DO 60 J=1,N
      WW=WORK1(INX(I,J))
60    SUM=SUM+WW*VMU(J)
70    POW(I,L)=SUM
    END IF
    IF(ICODE.EQ.3) THEN
      SUM=ZERO
      DO 80 I=1,N
      TT=TWO
      DO 80 J=1,I
      IJ=INX(I,J)
      WW=WORK1(IJ)
      IF(I.EQ.J) TT=ONE
80    SUM=SUM+TT*WW*GAMMA(IJ)
      TR2(L)=SUM
    END IF
90  CONTINUE
    RETURN
    END
```

```
      SUBROUTINE CALCRA(LCODE,IS,NK,TR1,TR2,PP,OUT,ZERO,ONE,TWO)
C
C        CALLED BY FUNCTION F(X)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NK(*),TR1(*),TR2(*),PP(*),OUT(*)
      PR=ONE
      DO 10 J=1,IS
      NJ=NK(J)
10    IF(NJ.NE.0) PR=PR*(TR1(J)**NJ)
      OUT(1)=PR
      IF(LCODE.EQ.1) RETURN
      SUM=ZERO
      SUM1=ZERO
      DO 60 J=1,IS
      RJ=J
      NJ=NK(J)
      RNJ=NJ
      PR=ONE
      DO 20 L=1,IS
      NL=NK(L)
      IF(L.EQ.J) NL=NL-1
20    IF(NL.GT.0)PR=PR*(TR1(L)**NL)
      IF(NJ.NE.0) SUM=SUM+RJ*RNJ*PR*TR2(J)
      IF(LCODE.EQ.2) GO TO 60
      PR=ONE
      DO 30 L=1,IS
      NL=NK(L)
      IF(L.EQ.J) NL=NL-2
30    IF(NL.GT.0) PR=PR*(TR1(L)**NL)
      IF(NJ.GT.1) SUM1=SUM1+RJ*RJ*RNJ*(RNJ-ONE)*PR*PP(INX(J,J))
      IF(J.GT.1) THEN
        DO 50 I=1,J-1
```

```
      RI=I
      NI=NK(I)
      RNI=NI
      PR=ONE
      DO 40 L=1,IS
      NL=NK(L)
      IF(L.EQ.I.OR.L.EQ.J) NL=NL-1
40    IF(NL.GT.0) PR=PR*(TR1(L)**NL)
50    IF(NI.GT.0.AND.NJ.GT.0)
     +  SUM1=SUM1+TWO*RI*RJ*RNI*RNJ*PR*PP(INX(I,J))
      END IF
60    CONTINUE
      OUT(2)=SUM
      OUT(3)=SUM1
      RETURN
      END

      SUBROUTINE INIT(ICODE,N,NDIM,A,B,C,ELA,IMU,EMU,
     +IOM,OMEGA,AA,RLANDA,CC,SLA,SMU,WMAT,PMAT,WORK,
     +ITEM,IMAX,ZERO,ONE,EPS,INF,IFAIL)
C
C       INITIALIZES ALL MATRICES AND VECTORS AND CHECKS
C       FOR THE EXISTENCE OF EXPECTATIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),B(*),C(*),OMEGA(*),AA(*),CC(*),WORK(*)
      DIMENSION ELA(*),EMU(*),SLA(*),SMU(*),RLANDA(*)
      DIMENSION WMAT(NDIM,*),PMAT(NDIM,*)
      IFAIL=1
      IF(N.GT.NDIM.OR.N.LE.1) RETURN
      IFAIL=2
      IF(ICODE.LT.1.OR.ICODE.GT.3) RETURN
      IFAIL=3
      IF(IOM.NE.-1.AND.IOM.NE.1.AND.IOM.NE.2) RETURN
```

```
      CALL EVALUE(B,NDIM,N,RLANDA,PMAT,WMAT,CC,EPS,IRANK,ZERO,IFAIL)
      IFAIL=IFAIL+3
      IF(IFAIL.NE.3) RETURN
      IFAIL=6
      IF(IRANK.EQ.0) RETURN
      CALL EXIST(ICODE,N,NDIM,A,C,ELA,PMAT,IRANK,
     +ITEM,IMAX,ZERO,EPS,INF)
      DO 10 I=1,N
      DO 10 J=1,I
      IJ=INX(I,J)
10    CC(IJ)=OMEGA(IJ)
      IF(IOM.EQ.1.OR.IOM.EQ.-1) THEN
        IFAIL=7
        DO 20 I=1,N
        R=CC(INX(I,I))
        IF(R.LT.EPS) RETURN
20      CONTINUE
      END IF
      IFAIL=0
      IF(IOM.EQ.2) CALL SEP(CC,N,ONE,EPS,IFAIL)
      IF(IOM.EQ.-1) CALL LOWINV(CC,N,ZERO,ONE,EPS,IFAIL)
      IF(IFAIL.NE.0) THEN
        IF(IOM.EQ.2) IFAIL=7
        IF(IOM.EQ.-1) IFAIL=8
        RETURN
      END IF
      CALL MULT1(B,CC,AA,N,ZERO)
      CALL EVALUE(AA,NDIM,N,RLANDA,PMAT,WMAT,WORK,EPS,IRANK,ZERO,IFAIL)
      IFAIL=IFAIL+8
      IF(IFAIL.NE.8)RETURN
      IFAIL=11
      IF(IRANK.EQ.0) RETURN
      DO 40 I=1,N
      DO 40 J=1,N
```

```
      SUM=ZERO
      DO 30 K=1,I
30    SUM=SUM+CC(INX(I,K))*PMAT(K,J)
40    WMAT(I,J)=SUM
      CALL MULT2(A,WMAT,AA,N,N,NDIM,ZERO)
      IF(IMU.NE.0) THEN
        IF(IOM.GT.0) THEN
          CALL LOWINV(CC,N,ZERO,ONE,EPS,IFAIL)
          IFAIL=IFAIL+7
          IF(IFAIL.NE.7) RETURN
        END IF
        DO 60 I=1,N
        SUM=ZERO
        DO 50 J=1,N
        DO 50 K=1,J
        IF(IOM.GT.0) SUM=SUM+PMAT(J,I)*CC(INX(J,K))*EMU(K)
        IF(IOM.EQ.-1) SUM=SUM+PMAT(J,I)*OMEGA(INX(J,K))*EMU(K)
50      CONTINUE
60      SMU(I)=SUM
      END IF
      IF(ICODE.EQ.2) THEN
        DO 80 I=1,N
        SUM=ZERO
        DO 70 J=1,N
70      SUM=SUM+WMAT(J,I)*ELA(J)
80      SLA(I)=SUM
      ELSE IF(ICODE.EQ.3) THEN
        CALL MULT2(C,WMAT,CC,N,N,NDIM,ZERO)
      END IF
      IFAIL=0
      RETURN
      END
```

```
      SUBROUTINE EXIST(ICODE,N,NDIM,A,C,ELA,Q,IRANK,
     +ITEM,IMAX,ZERO,EPS,INF)
C
C        CHECKS THE EXISTENCE OF THE EXPECTATION TO BE CALCULATED
C        USING THEOREMS 1-3 OF MAGNUS(1989)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),C(*),ELA(*),Q(NDIM,*)
      ZABS(H)=DABS(H)
      ITEM=1
      IMAX=INF
      IF(IRANK.EQ.N) RETURN
      IZERO=N-IRANK
C ... CHECK TO SEE IF A*Q IS ZERO
      CALL NULL1(A,Q,N,IZERO,NDIM,ZERO,EPS,IX1)
      IF(IX1.EQ.0) RETURN
C ... CHECK TO SEE IF Q'*A*Q IS ZERO
      CALL NULL2(A,Q,N,IZERO,NDIM,ZERO,EPS,IX1)
      IF(ICODE.EQ.1) THEN
        IF(IX1.EQ.0) THEN
          IMAX=IRANK-1
          ITEM=2
        ELSE IF(IX1.EQ.1) THEN
          IMAX=(IRANK-1)/2
          ITEM=3
        END IF
      ELSE IF(ICODE.EQ.2) THEN
C ...   CHECK TO SEE IF Q'*ELA IS ZERO
        IX2=1
        DO 20 K=1,IZERO
        SUM=ZERO
        DO 10 J=1,N
10      SUM=SUM+Q(J,K)*ELA(J)
        IF(ZABS(SUM).GT.EPS) GO TO 30
```

```
20        CONTINUE
          IX2=0
30        CONTINUE
          IF(IX1.EQ.0.AND.IX2.EQ.0) THEN
            IMAX=IRANK
            ITEM=2
          ELSE IF(IX1.EQ.0.AND.IX2.EQ.1) THEN
            IMAX=IRANK-1
            ITEM=3
          ELSE IF(IX1.EQ.1.AND.IX2.EQ.0) THEN
            IMAX=IRANK/2
            ITEM=4
          ELSE
            IMAX=(IRANK-1)/2
            ITEM=5
          END IF
        ELSE IF(ICODE.EQ.3) THEN
C ...     CHECK TO SEE IF C*Q IS ZERO
          CALL NULL1(C,Q,N,IZERO,NDIM,ZERO,EPS,IX2)
C ...     CHECK TO SEE IF Q'*C*Q IS ZERO
          CALL NULL2(C,Q,N,IZERO,NDIM,ZERO,EPS,IX3)
          IF(IX1.EQ.0) THEN
            IF(IX2.EQ.0) THEN
              IMAX=IRANK+1
              ITEM=2
            ELSE IF(IX2.EQ.1.AND.IX3.EQ.0) THEN
              IMAX=IRANK
              ITEM=3
            ELSE IF(IX3.EQ.1) THEN
              IMAX=IRANK-1
              ITEM=4
            END IF
          ELSE IF(IX1.EQ.1) THEN
            IF(IX2.EQ.0) THEN
```

```
          IMAX=(IRANK+1)/2
          ITEM=5
        ELSE IF(IX2.EQ.1.AND.IX3.EQ.0) THEN
          IMAX=IRANK/2
          ITEM=6
        ELSE IF(IX3.EQ.1) THEN
          IMAX=(IRANK-1)/2
          ITEM=7
        END IF
      END IF
    END IF
    RETURN
    END

    SUBROUTINE PARINT(M,MPRTN,MRDIM,MDIM,MR,IFAIL)
C
C       CONSTRUCTS THE MR X M MATRIX "MPRTN" CONTAINING
C       ALL MR PARTITIONS OF THE INTEGER M
C
    PARAMETER(MAXMOM=24)
    DIMENSION MPRTN(MRDIM,MDIM)
    DIMENSION NUM(MAXMOM),IWORK(MAXMOM)
    DATA NUM/1,2,3,5,7,11,15,22,30,42,56,77,101,135,176,231,
   +297,385,490,627,792,1002,1255,1575/
    IFAIL=1
    IF(M.LT.1.OR.M.GT.MAXMOM.OR.M.GT.MDIM) RETURN
    IFAIL=2
    IF(NUM(M).GT.MRDIM) RETURN
    IFAIL=0
    N1=0
    N2=1
    N3=0
    MR=1
    M1=1
```

```
      L=0
      MPRTN(1,1)=1
      IF(M.EQ.1) RETURN
      DO 10 J=2,M
10    MPRTN(1,J)=0
      DO 99 K=2,M
      IF(N2.NE.0) THEN
        DO 30 I=1,N2
        MPRTN(MR+I,1)=0
        DO 20 J=2,M
20      MPRTN(MR+I,J)=MPRTN(I+N1,J)
30      MPRTN(MR+I,2)=MPRTN(MR+I,2)+1
      END IF
      IF(N3.NE.0) THEN
        L=0
        DO 80 I=N1+N2+1,MR
        DO 80 J=2,K-1
        IF(MPRTN(I,J).EQ.0) GO TO 80
        DO 40 JJ=1,M
40      IWORK(JJ)=MPRTN(I,JJ)
        IWORK(J)=IWORK(J)-1
        IWORK(J+1)=IWORK(J+1)+1
        IF(N2.NE.0.OR.L.NE.0) THEN
          DO 60 II=MR+1,MR+N2+L
          DO 50 JJ=1,M-1
50        IF(MPRTN(II,JJ).NE.IWORK(JJ)) GO TO 60
          IF(MPRTN(II,M).EQ.IWORK(M)) GO TO 80
60        CONTINUE
        END IF
        L=L+1
        DO 70 JJ=1,M
70      MPRTN(MR+N2+L,JJ)=IWORK(JJ)
80      CONTINUE
      END IF
```

```
      DO 90 II=1,MR
90    MPRTN(II,1)=MPRTN(II,1)+1
      N1=M1
      N3=N2+L
      N2=MR-N1
      M1=MR
      MR=MR+N3
99    CONTINUE
      RETURN
      END

      INTEGER FUNCTION INX(I,J)
C
C       PICKS OUT THE APPROPRIATE ELEMENT OF A SYMMETRIC
C       MATRIX STORED IN LOWER TRIANGULAR FORM
C
      IF(I.GE.J) THEN
        INX=I*(I-1)/2+J
      ELSE
        INX=J*(J-1)/2+I
      END IF
      RETURN
      END

      INTEGER FUNCTION NFACT(N)
C
C       CALCULATES N FACTORIAL
C
      NFACT=1
      IF(N.LE.1)RETURN
      DO 10 I=2,N
10    NFACT=NFACT*I
      RETURN
      END
```

```
      SUBROUTINE SEP(A,N,ONE,EPS,IFAIL)
C
C       DECOMPOSES A (POSITIVE DEFINITE) INTO LL'
C       (L LOWER TRIANGULAR) AND REPLACES A BY L
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*)
      ZSQRT(H)=DSQRT(H)
      IFAIL=1
      DO 20 I=1,N
      DO 20 J=I,N
      IJ=INX(J,I)
      R=A(IJ)
      IF(I.NE.1) THEN
        DO 10 K=1,I-1
10      R=R-A(INX(I,K))*A(INX(J,K))
      END IF
      IF(I.EQ.J) THEN
        IF(R.LT.EPS) RETURN
        D=ONE/ZSQRT(R)
      END IF
20    A(IJ)=R*D
      IFAIL=0
      RETURN
      END

      SUBROUTINE LOWINV(A,N,ZERO,ONE,EPS,IFAIL)
C
C       INVERSION (IN PLACE) OF N X N LOWER TRIANGULAR MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*)
      ZABS(H)=DABS(H)
```

```
      IFAIL=1
      DO 20 I=1,N
      R=A(INX(I,I))
      IF(ZABS(R).LT.EPS) RETURN
      D=ONE/R
      DO 20 J=1,I
      IJ=INX(I,J)
      IF(I.NE.J) THEN
        SUM=ZERO
        DO 10 K=J,I-1
10      SUM=SUM+A(INX(I,K))*A(INX(K,J))
        A(IJ)=-SUM*D
      ELSE
        A(IJ)=D
      END IF
20    CONTINUE
      IFAIL=0
      RETURN
      END

      SUBROUTINE MULT1(A,B,C,N,ZERO)
C
C       FINDS C = B'AB (A=A', B LOWER TRIANGULAR)
C       ALL MATRICES N X N
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),B(*),C(*)
      DO 20 I=1,N
      DO 20 J=1,I
      IJ=INX(I,J)
      SUM=ZERO
      DO 10 K=I,N
      KI=INX(K,I)
      DO 10 L=J,N
```

```
      LJ=INX(L,J)
      KL=INX(K,L)
10    SUM=SUM+B(KI)*A(KL)*B(LJ)
20    C(IJ)=SUM
      RETURN
      END

      SUBROUTINE MULT2(A,B,C,N,M,NDIM,ZERO)
C
C       FINDS C = B'AB (A=A')
C       A IS N X N, B IS N X M, C IS M X M
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),B(NDIM,*),C(*)
      DO 20 I=1,M
      DO 20 J=1,I
      IJ=INX(I,J)
      SUM=ZERO
      DO 10 L=1,N
      DO 10 K=1,N
      KL=INX(K,L)
10    SUM=SUM+B(K,I)*A(KL)*B(L,J)
20    C(IJ)=SUM
      RETURN
      END

      SUBROUTINE NULL1(A,B,N,M,NDIM,ZERO,EPS,IEX)
C
C       CHECKS TO SEE IF AB = 0 (A=A')
C       A IS N X N, B IS N X M
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),B(NDIM,*)
      ZABS(H)=DABS(H)
```

```
      IEX=1
      DO 20 K=1,M
      DO 20 I=1,N
      SUM=ZERO
      DO 10 J=1,N
10    SUM=SUM+A(INX(I,J))*B(J,K)
      IF(ZABS(SUM).GT.EPS) RETURN
20    CONTINUE
      IEX=0
      RETURN
      END

      SUBROUTINE NULL2(A,B,N,M,NDIM,ZERO,EPS,IEX)
C
C       CHECKS TO SEE IF B'AB = 0 (A=A')
C       A IS N X N, B IS N X M
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),B(NDIM,*)
      ZABS(H)=DABS(H)
      IEX=1
      DO 20 I=1,M
      DO 20 J=1,I
      SUM=ZERO
      DO 10 K=1,N
      DO 10 L=1,N
10    SUM=SUM+B(K,I)*A(INX(K,L))*B(L,J)
      IF(ZABS(SUM).GT.EPS) RETURN
20    CONTINUE
      IEX=0
      RETURN
      END
```

```
      SUBROUTINE EVALUE(A,NDIM,N,RLANDA,P,WMAT,WORK,EPS,
     +IRANK,ZERO,IFAIL)
C
C       FINDS THE RANK, THE EIGENVALUES AND THE
C       EIGENVECTORS OF A POSITIVE SEMIDEFINITE
C       N X N MATRIX A (IN IRANK, RLANDA AND P)
C       NB: USES ROUTINE FROM NAG LIBRARY
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(*),RLANDA(*),WORK(*)
      DIMENSION P(NDIM,*),WMAT(NDIM,*)
      IFAIL=0
      DO 10 I=1,N
      DO 10 J=1,N
10    WMAT(I,J)=A(INX(I,J))
      CALL F02ABF(WMAT,NDIM,N,RLANDA,P,NDIM,WORK,IFAIL)
      IF(IFAIL.NE.0) THEN
        IFAIL=1
        RETURN
      END IF
      IFAIL=2
      IF(RLANDA(1).LT.-EPS) RETURN
      IFAIL=0
      IRANK=N
      DO 20 I=1,N
      IF(RLANDA(I).GT.EPS) RETURN
      RLANDA(I)=ZERO
20    IRANK=IRANK-1
      RETURN
      END
```

```
      SUBROUTINE INTGRL(F,ZERO,EPSABS,EPSREL,RES,ABSERR,W,IW,IFAIL)
C
C        FINDS VALUE OF INTEGRAL FROM ZERO TO INFINITY OF F(X)
C        NB: USES ROUTINE FROM NAG LIBRARY
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION W(2000),IW(252)
      EXTERNAL F
      IONE=1
      IFAIL=0
      CALL D01AMF(F,ZERO,IONE,EPSABS,EPSREL,RES,ABSERR,W,2000,
     +IW,252,IFAIL)
      RETURN
      END
```

Discussion Paper Series, CentER, Tilburg University, The Netherlands:

(For previous papers please consult previous discussion papers.)

No.	Author(s)	Title
8901	Th. Ten Raa and P. Kop Jansen	The Choice of Model in the Construction of Input-Output Coefficients Matrices
8902	Th. Nijman and F. Palm	Generalized Least Squares Estimation of Linear Models Containing Rational Future Expectations
8903	A. van Soest, I. Woittiez, A. Kapteyn	Labour Supply, Income Taxes and Hours Restrictions in The Netherlands
8904	F. van der Ploeg	Capital Accumulation, Inflation and Long-Run Conflict in International Objectives
8905	Th. van de Klundert and A. van Schaik	Unemployment Persistence and Loss of Productive Capacity: A Keynesian Approach
8906	A.J. Markink and F. van der Ploeg	Dynamic Policy Simulation of Linear Models with Rational Expectations of Future Events: A Computer Package
8907	J. Osiewalski	Posterior Densities for Nonlinear Regression with Equicorrelated Errors
8908	M.F.J. Steel	A Bayesian Analysis of Simultaneous Equation Models by Combining Recursive Analytical and Numerical Approaches
8909	F. van der Ploeg	Two Essays on Political Economy (i) The Political Economy of Overvaluation (ii) Election Outcomes and the Stockmarket
8910	R. Gradus and A. de Zeeuw	Corporate Tax Rate Policy and Public and Private Employment
8911	A.P. Barten	Allais Characterisation of Preference Structures and the Structure of Demand
8912	K. Kamiya and A.J.J. Talman	Simplicial Algorithm to Find Zero Points of a Function with Special Structure on a Simplotope
8913	G. van der Laan and A.J.J. Talman	Price Rigidities and Rationing
8914	J. Osiewalski and M.F.J. Steel	A Bayesian Analysis of Exogeneity in Models Pooling Time-Series and Cross-Section Data
8915	R.P. Gilles, P.H. Ruys and J. Shou	On the Existence of Networks in Relational Models

No.	Author(s)	Title
8916	A. Kapteyn, P. Kooreman and A. van Soest	Quantity Rationing and Concavity in a Flexible Household Labor Supply Model
8917	F. Canova	Seasonalities in Foreign Exchange Markets
8918	F. van der Ploeg	Monetary Disinflation, Fiscal Expansion and the Current Account in an Interdependent World
8919	W. Bossert and F. Stehling	On the Uniqueness of Cardinally Interpreted Utility Functions
8920	F. van der Ploeg	Monetary Interdependence under Alternative Exchange-Rate Regimes
8921	D. Canning	Bottlenecks and Persistent Unemployment: Why Do Booms End?
8922	C. Fershtman and A. Fishman	Price Cycles and Booms: Dynamic Search Equilibrium
8923	M.B. Canzoneri and C.A. Rogers	Is the European Community an Optimal Currency Area? Optimal Tax Smoothing versus the Cost of Multiple Currencies
8924	F. Groot, C. Withagen and A. de Zeeuw	Theory of Natural Exhaustible Resources: The Cartel-Versus-Fringe Model Reconsidered
8925	O.P. Attanasio and G. Weber	Consumption, Productivity Growth and the Interest Rate
8926	N. Rankin	Monetary and Fiscal Policy in a 'Hartian' Model of Imperfect Competition
8927	Th. van de Klundert	Reducing External Debt in a World with Imperfect Asset and Imperfect Commodity Substitution
8928	C. Dang	The D_1-Triangulation of R^n for Simplicial Algorithms for Computing Solutions of Nonlinear Equations
8929	M.F.J. Steel and J.F. Richard	Bayesian Multivariate Exogeneity Analysis: An Application to a UK Money Demand Equation
8930	F. van der Ploeg	Fiscal Aspects of Monetary Integration in Europe
8931	H.A. Keuzenkamp	The Prehistory of Rational Expectations

No.	Author(s)	Title
8932	E. van Damme, R. Selten and E. Winter	Alternating Bid Bargaining with a Smallest Money Unit
8933	H. Carlsson and E. van Damme	Global Payoff Uncertainty and Risk Dominance
8934	H. Huizinga	National Tax Policies towards Product-Innovating Multinational Enterprises
8935	C. Dang and D. Talman	A New Triangulation of the Unit Simplex for Computing Economic Equilibria
8936	Th. Nijman and M. Verbeek	The Nonresponse Bias in the Analysis of the Determinants of Total Annual Expenditures of Households Based on Panel Data
8937	A.P. Barten	The Estimation of Mixed Demand Systems
8938	G. Marini	Monetary Shocks and the Nominal Interest Rate
8939	W. Güth and E. van Damme	Equilibrium Selection in the Spence Signaling Game
8940	G. Marini and P. Scaramozzino	Monopolistic Competition, Expected Inflation and Contract Length
8941	J.K. Dagsvik	The Generalized Extreme Value Random Utility Model for Continuous Choice
8942	M.F.J. Steel	Weak Exogenity in Misspecified Sequential Models
8943	A. Roell	Dual Capacity Trading and the Quality of the Market
8944	C. Hsiao	Identification and Estimation of Dichotomous Latent Variables Models Using Panel Data
8945	R.P. Gilles	Equilibrium in a Pure Exchange Economy with an Arbitrary Communication Structure
8946	W.B. MacLeod and J.M. Malcomson	Efficient Specific Investments, Incomplete Contracts, and the Role of Market Alternatives
8947	A. van Soest and A. Kapteyn	The Impact of Minimum Wage Regulations on Employment and the Wage Rate Distribution
8948	P. Kooreman and B. Melenberg	Maximum Score Estimation in the Ordered Response Model

No.	Author(s)	Title
8949	C. Dang	The D_3-Triangulation for Simplicial Deformation Algorithms for Computing Solutions of Nonlinear Equations
8950	M. Cripps	Dealer Behaviour and Price Volatility in Asset Markets
8951	T. Wansbeek and A. Kapteyn	Simple Estimators for Dynamic Panel Data Models with Errors in Variables
8952	Y. Dai, G. van der Laan, D. Talman and Y. Yamamoto	A Simplicial Algorithm for the Nonlinear Stationary Point Problem on an Unbounded Polyhedron
8953	F. van der Ploeg	Risk Aversion, Intertemporal Substitution and Consumption: The CARA-LQ Problem
8954	A. Kapteyn, S. van de Geer, H. van de Stadt and T. Wansbeek	Interdependent Preferences: An Econometric Analysis
8955	L. Zou	Ownership Structure and Efficiency: An Incentive Mechanism Approach
8956	P.Kooreman and A. Kapteyn	On the Empirical Implementation of Some Game Theoretic Models of Household Labor Supply
8957	E. van Damme	Signaling and Forward Induction in a Market Entry Context
9001	A. van Soest, P. Kooreman and A. Kapteyn	Coherency and Regularity of Demand Systems with Equality and Inequality Constraints
9002	J.R. Magnus and B. Pesaran	Forecasting, Misspecification and Unit Roots: The Case of AR(1) Versus ARMA(1,1)
9003	J. Driffill and C. Schultz	Wage Setting and Stabilization Policy in a Game with Renegotiation
9004	M. McAleer, M.H. Pesaran and A. Bera	Alternative Approaches to Testing Non-Nested Models with Autocorrelated Disturbances: An Application to Models of U.S. Unemployment
9005	Th. ten Raa and M.F.J. Steel	A Stochastic Analysis of an Input-Output Model: Comment
9006	M. McAleer and C.R. McKenzie	Keynesian and New Classical Models of Unemployment Revisited

No.	Author(s)	Title
9007	J. Osiewalski and M.F.J. Steel	Semi-Conjugate Prior Densities in Multivariate t Regression Models
9008	G.W. Imbens	Duration Models with Time-Varying Coefficients
9009	G.W. Imbens	An Efficient Method of Moments Estimator for Discrete Choice Models with Choice-Based Sampling
9010	P. Deschamps	Expectations and Intertemporal Separability in an Empirical Model of Consumption and Investment under Uncertainty
9011	W. Güth and E. van Damme	Gorby Games - A Game Theoretic Analysis of Disarmament Campaigns and the Defense Efficiency-Hypothesis
9012	A. Horsley and A. Wrobel	The Existence of an Equilibrium Density for Marginal Cost Prices, and the Solution to the Shifting-Peak Problem
9013	A. Horsley and A. Wrobel	The Closedness of the Free-Disposal Hull of a Production Set
9014	A. Horsley and A. Wrobel	The Continuity of the Equilibrium Price Density: The Case of Symmetric Joint Costs, and a Solution to the Shifting-Pattern Problem
9015	A. van den Elzen, G. van der Laan and D. Talman	An Adjustment Process for an Exchange Economy with Linear Production Technologies
9016	P. Deschamps	On Fractional Demand Systems and Budget Share Positivity
9017	B.J. Christensen and N.M. Kiefer	The Exact Likelihood Function for an Empirical Job Search Model
9018	M. Verbeek and Th. Nijman	Testing for Selectivity Bias in Panel Data Models
9019	J.R. Magnus and B. Pesaran	Evaluation of Moments of Ratios of Quadratic Forms in Normal Variables and Related Statistics

P.O. BOX 90153. 5000 LE TILBURG. THE NETHERLANDS