NAG Toolbox for MATLAB

g13db

1 Purpose

g13db calculates the multivariate partial autocorrelation function of a multivariate time series.

2 Syntax

```
[p, v0, v, d, db, w, wb, nvp, ifail] = g13db(c0, c, nl, nk, 'ns', ns)
```

3 Description

The input is a set of lagged autocovariance matrices $C_0, C_1, C_2, \ldots, C_m$. These will generally be sample values such as are obtained from a multivariate time series using g13dm.

The main calculation is the recursive determination of the coefficients in the finite lag (forward) prediction equation

$$x_t = \Phi_{l,1}x_{t-1} + \cdots + \Phi_{l,l}x_{t-l} + e_{l,t}$$

and the associated backward prediction equation

$$x_{t-l-1} = \Psi_{l,1}x_{t-l} + \cdots + \Psi_{l,l}x_{t-1} + f_{l,t}$$

together with the covariance matrices D_l of $e_{l,t}$ and G_l of $f_{l,t}$.

The recursive cycle, by which the order of the prediction equation is extended from l to $l+1$, is to calculate

$$M_{l+1} = C_{l+1}^{\mathrm{T}} - \Phi_{l,1}C_l^{\mathrm{T}} - \cdots - \Phi_{l,l}C_1^{\mathrm{T}} \qquad (1)$$

then $\Phi_{l+1,l+1} = M_{l+1}D_l^{-1}, \qquad \Psi_{l+1,l+1} = M_{l+1}^{\mathrm{T}}G_l^{-1}$

from which

$$\Phi_{l+1,j} = \Phi_{l,j} - \Phi_{l+1,l+1}\Psi_{l,l+1-j}, \qquad j = 1,2,\ldots,l \qquad (2)$$

and

$$\Psi_{l+1,j} = \Psi_{l,j} - \Psi_{l+1,l+1}\Phi_{l,l+1-j}, \qquad j = 1,2,\ldots,l. \qquad (3)$$

Finally, $D_{l+1} = D_l - M_{l+1}\Phi_{l+1,l+1}^{\mathrm{T}}$ and $G_{l+1} = G_l - M_{l+1}^{\mathrm{T}}\Psi_{l+1,l+1}^{\mathrm{T}}$.

(Here T denotes the transpose of a matrix.)

The cycle is initialized by taking (for $l = 0$)

$$D_0 = G_0 = C_0.$$

In the step from $l = 0$ to 1, the above equations contain redundant terms and simplify. Thus (1) becomes $M_1 = C_1^{\mathrm{T}}$ and neither (2) or (3) are needed.

Quantities useful in assessing the effectiveness of the prediction equation are generalized variance ratios

$$v_l = \det D_l / \det C_0, \qquad l = 1,2,\ldots$$

and multiple squared partial autocorrelations

$$p_l^2 = 1 - v_l/v_{l-1}.$$

4 References

Akaike H 1971 Autoregressive model fitting for control *Ann. Inst. Statist. Math.* **23** 163–180

Whittle P 1963 On the fitting of multivariate autoregressions and the approximate canonical factorization of a spectral density matrix *Biometrika* **50** 129–134

5 Parameters

5.1 Compulsory Input Parameters

1: **c0(ldc0,ns) – double array**

Contains the zero lag cross-covariances between the **ns** series as returned by g13dm. (**c0** is assumed to be symmetric, upper triangle only is used.)

2: **c(ldc0,ldc0,nl) – double array**

Contains the cross-covariances at lags 1 to **nl**. **c**(i,j,k) must contain the cross-covariance, c_{ijk}, of series i and series j at lag k. Series j leads series i.

3: **nl – int32 scalar**

m, the maximum lag for which cross-covariances are supplied in **c**.

Constraint: **nl** ≥ 1.

4: **nk – int32 scalar**

the number of lags to which partial auto-correlations are to be calculated.

Constraint: $1 \leq$ **nk** $\leq$ **nl**.

5.2 Optional Input Parameters

1: **ns – int32 scalar**

Default: The dimension of the arrays **c0**, **db**. (An error is raised if these dimensions are not equal.)

k, the number of time series whose cross-covariances are supplied in **c** and **c0**.

Constraint: **ns** ≥ 1.

5.3 Input Parameters Omitted from the MATLAB Interface

ldc0, wa, iwa

5.4 Output Parameters

1: **p(nk) – double array**

The multiple squared partial autocorrelations from lags 1 to **nvp**; that is, **p**(l) contains p_l^2, for $l = 1, 2, \dots,$ **nvp**. For lags **nvp** $+ 1$ to **nk** the elements of **p** are set to zero.

2: **v0 – double scalar**

The lag zero prediction error variance (equal to the determinant of **c0**).

3: **v(nk) – double array**

The prediction error variance ratios from lags 1 to **nvp**; that is, **v**(l) contains v_l, for $l = 1, 2, \dots,$ **nvp**. For lags **nvp** $+ 1$ to **nk** the elements of **v** are set to zero.

4: **d(ldc0,ldc0,nk) – double array**

The prediction error variance matrices at lags 1 to **nvp**.

Element (i,j,k) of **d** contains the prediction error covariance of series i and series j at lag k, for $k = 1,2,\ldots,$ **nvp**. Series j leads series i; that is, the (i,j)th element of D_k. For lags **nvp** $+ 1$ to **nk** the elements of **d** are set to zero.

5: **db(ldc0,ns) – double array**

The backward prediction error variance matrix at lag **nvp**.

db(i,j) contains the backward prediction error covariance of series i and series j; that is, the (i,j)th element of the G_k, where $k =$ **nvp**.

6: **w(ldc0,ldc0,nk) – double array**

The prediction coefficient matrices at lags 1 to **nvp**.

w(i,j,l) contains the jth prediction coefficient of series i at lag l; that is, the (i,j)th element of Φ_{kl}, where $k =$ **nvp**, for $l = 1,2,\ldots,$ **nvp**. For lags **nvp** $+ 1$ to **nk** the elements of **w** are set to zero.

7: **wb(ldc0,ldc0,nk) – double array**

The backward prediction coefficient matrices at lags 1 to **nvp**.

wb(i,j,l) contains the jth backward prediction coefficient of series i at lag l; that is, the (i,j)th element of Ψ_{kl}, where $k =$ **nvp**, for $l = 1,2,\ldots,$ **nvp**. For lags **nvp** $+ 1$ to **nk** the elements of **wb** are set to zero.

8: **nvp – int32 scalar**

The maximum lag, L, for which calculation of **p**, **v**, **d**, **db**, **w** and **wb** was successful. If the function completes successfully **nvp** will equal **nk**.

9: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **ldc0** < 1,
or **ns** < 1,
or **ns** $>$ **ldc0**,
or **nl** < 1,
or **nk** < 1,
or **nk** $>$ **nl**,
or **iwa** $< (2 \times$ **ns** $+ 1) \times$ **ns**.

ifail $= 2$

c0 is not positive-definite.

v0, **v**, **p**, **d**, **db**, **w**, **wb** and **nvp** are set to zero.

ifail $= 3$

At lag $k =$ **nvp** $+ 1 \leq$ **nk**, D_k was found not to be positive-definite. Up to lag **nvp**, **v0**, **v**, **p**, **d**, **w** and **wb** contain the values calculated so far and from lag **nvp** $+ 1$ to lag **nk** the matrices contain zero. **db** contains the backward prediction coefficients for lag **nvp**.

7 Accuracy

The conditioning of the problem depends on the prediction error variance ratios. Very small values of these may indicate loss of accuracy in the computations.

8 Further Comments

The time taken by g13db is roughly proportional to $\mathbf{nk}^2 \times \mathbf{ns}^3$.

If sample autocorrelation matrices are used as input, then the output will be relevant to the original series scaled by their standard deviations. If these autocorrelation matrices are produced by g13dm, you must replace the diagonal elements of C_0 (otherwise used to hold the series variances) by 1.

9 Example

```
c0 = [0.0109, -0.0077917, 0.0013004, 0.0012654;
     -0.0077917, 0.05704, 0.002418, 0.014409;
     0.0013004, 0.002418, 0.04396, -0.021421;
     0.0012654, 0.014409, -0.021421, 0.072289];
c = zeros(4, 4, 5);
c(1, 1, 1) = 0.0045889;
c(1, 1, 2) = 0.0018652;
c(1, 1, 3) = -8.054999999999999e-05;
c(1, 1, 4) = 0.00076079;
c(1, 1, 5) = -0.0006436500000000001;
c(1, 2, 1) = 0.0004651;
c(1, 2, 2) = -0.0064389;
c(1, 2, 3) = -0.0037759;
c(1, 2, 4) = -0.0010134;
c(1, 2, 5) = -0.004455600000000001;
c(1, 3, 1) = -0.00013275;
c(1, 3, 2) = 0.0088307;
c(1, 3, 3) = 0.007546300000000001;
c(1, 3, 4) = 0.01187;
c(1, 3, 5) = 0.0051334;
c(1, 4, 1) = 0.007753100000000001;
c(1, 4, 2) = -0.0024808;
c(1, 4, 3) = -0.004227600000000001;
c(1, 4, 4) = -0.0041651;
c(1, 4, 5) = 0.00071587;
c(2, 1, 1) = -0.0024419;
c(2, 1, 2) = -0.011865;
c(2, 1, 3) = 0.0041447;
c(2, 1, 4) = 0.0036014;
c(2, 1, 5) = 0.006361700000000001;
c(2, 2, 1) = -0.011667;
c(2, 2, 2) = 0.0072367;
c(2, 2, 3) = -0.0037987;
c(2, 2, 4) = -0.0036375;
c(2, 2, 5) = 0.00015217;
c(2, 3, 1) = -0.021956;
c(2, 3, 2) = -0.019802;
c(2, 3, 3) = 0.0019332;
c(2, 3, 4) = -0.025571;
c(2, 3, 5) = 0.002727;
c(2, 4, 1) = -0.0045803;
c(2, 4, 2) = 0.005906900000000001;
c(2, 4, 3) = -0.017564;
c(2, 4, 4) = 0.0050218;
c(2, 4, 5) = -0.0022261;
c(3, 1, 1) = 0.001108;
c(3, 1, 2) = -0.0080307;
c(3, 1, 3) = -0.010582;
c(3, 1, 4) = -0.013924;
c(3, 1, 5) = -0.008585500000000001;
```

```
c(3, 2, 1) = -0.0080479;
c(3, 2, 2) = 0.014306;
c(3, 2, 3) = 0.006773299999999999;
c(3, 2, 4) = 0.011718;
c(3, 2, 5) = 0.0014468;
c(3, 3, 1) = 0.013621;
c(3, 3, 2) = 0.014546;
c(3, 3, 3) = 0.0069832;
c(3, 3, 4) = -0.0059088;
c(3, 3, 5) = -0.0028698;
c(3, 4, 1) = -0.0085868;
c(3, 4, 2) = 0.01351;
c(3, 4, 3) = 0.0061747;
c(3, 4, 4) = 0.0059297;
c(3, 4, 5) = 0.0044384;
c(4, 1, 1) = -0.0005061400000000001;
c(4, 1, 2) = -0.0021791;
c(4, 1, 3) = 0.0041352;
c(4, 1, 4) = 0.010739;
c(4, 1, 5) = 0.0068339;
c(4, 2, 1) = 0.014045;
c(4, 2, 2) = -0.029528;
c(4, 2, 3) = -0.016013;
c(4, 2, 4) = -0.014571;
c(4, 2, 5) = -0.002179;
c(4, 3, 1) = -0.0010087;
c(4, 3, 2) = -0.015887;
c(4, 3, 3) = 0.017043;
c(4, 3, 4) = 0.013816;
c(4, 3, 5) = 0.013759;
c(4, 4, 1) = 0.012269;
c(4, 4, 2) = 0.00088308;
c(4, 4, 3) = -0.013412;
c(4, 4, 4) = -0.012588;
c(4, 4, 5) = 0.00028217;
nl = int32(5);
nk = int32(3);
[p, v0, v, d, db, w, wb, nvp, ifail] = g13db(c0, c, nl, nk)
```

```
p =
    0.6450
    0.9267
    0.8430
v0 =
   1.3670e-06
v =
    0.3550
    0.0260
    0.0041
d =
(:,:,1) =
    0.0081   -0.0051    0.0016   -0.0003
   -0.0051    0.0409    0.0076    0.0184
    0.0016    0.0076    0.0383   -0.0189
   -0.0003    0.0184   -0.0189    0.0676
(:,:,2) =
    0.0035   -0.0009   -0.0007   -0.0011
   -0.0009    0.0195    0.0053    0.0057
   -0.0007    0.0053    0.0190   -0.0107
   -0.0011    0.0057   -0.0107    0.0406
(:,:,3) =
    0.0030   -0.0009   -0.0005    0.0007
   -0.0009    0.0182    0.0087    0.0025
   -0.0005    0.0087    0.0093   -0.0022
    0.0007    0.0025   -0.0022    0.0225
db =
    0.0033   -0.0039   -0.0011    0.0059
   -0.0039    0.0189    0.0035   -0.0033
   -0.0011    0.0035    0.0100   -0.0105
    0.0059   -0.0033   -0.0105    0.0334
```

```
w =
(:,:,1) =
    0.8186    0.2340   -0.1710    0.0926
    0.0674   -0.4872   -0.1406    0.0429
    0.1504    0.1192   -0.3672   -0.4209
   -0.7097    0.0300    0.5978    0.3461
(:,:,2) =
   -0.3405   -0.1337    0.4061   -0.0218
   -1.2757   -0.1359   -0.6578   -0.1127
   -0.4544    0.1938    0.6342    0.3392
   -0.4324   -0.5485   -0.6290    0.1667
(:,:,3) =
    0.1644    0.1386    0.0129    0.0346
    0.3929    0.0741   -0.0880   -0.1536
   -1.2924   -0.2449    0.3023    0.3944
    0.8977   -0.3904    0.2515   -0.2830
wb =
(:,:,1) =
    0.4154    0.0615    0.1532    0.0508
    0.1237   -0.2647   -0.2272    0.4850
   -0.8693   -0.4737    0.3792    0.1381
    1.3078   -0.0918   -1.4540   -0.2197
(:,:,2) =
   -0.0674   -0.1226   -0.1367   -0.0973
   -1.2480    0.0309    0.5171   -0.2892
    0.9804   -0.2019    0.1631   -0.1087
   -1.6839   -0.7459    0.5290    0.4158
(:,:,3) =
    0.0379    0.1049   -0.2164    0.0801
    0.7539    0.2260   -0.2566   -0.4745
   -0.0034    0.0564   -0.0882    0.1272
    0.5502   -0.4123    0.7165   -0.1457
nvp =
           3
ifail =
           0
```