Package ‘AMA’

February 15, 2012

Type Package

Title Anderson-Moore Algorithm

Version 1.0.8

Depends rJava

Date 2010-10-13

Author Gary Anderson, Aneesh Raghunandan

Maintainer Gary Anderson <gary.anderson@frb.gov>

Description Implements Anderson-Moore algorithm for solving linear rational expectations models. For information about the algorithm and its uses, please see http://www.federalreserve.gov/pubs/oss/oss4/aimindex.html. This version works on both Unix and Windows. For questions about the algorithm and its implementation/applications, please contact gary.anderson@frb.gov. If you are having technical issues with the package, please contact aneesh.raghunandan@yale.edu.

License GPL 2.0

Copyright The ``C'' code, sparseAim.c, implementing the basic algorithm is in the public domain and may be used freely. However, the authors would appreciate acknowledgement of the source by citation of any of the following papers:
Anderson, G. and Moore, G. ``A Linear Algebraic Procedure For Solving Linear Perfect Foresight Models.'' Economics Letters, 17, 1985.
Anderson, G. ``Solving Linear Rational Expectations Models: A Horse Race.'' Computational Economics, 2008, vol. 31, issue 2, pp. 95-113
Anderson, G. ``A Reliable and Computationally Efficient Algorithm for Imposing the Saddle Point Property in Dynamic Models.'' Journal of Economic Dynamics and Control, 2010, vol 34, issue 3, pp. 472-489. \{}n

SystemRequirements Java

LazyLoad yes

R topics documented:

callAMA *Anderson-Moore Algorithm*

Description

Calls the Anderson-Moore Algorithm for the given inputs. callSparseAim is a wrapper to the C and FORTRAN functions which carry out the computation.

Usage

```
callAMA(cofh, neq, leads, lags, qmax)
```

Arguments

`cofh`	Full H matrix, passed to R columnwise. The H matrix is the matrix [H_-T ... H_0 ... H_theta]. Must have neq rows and neq*(leads+lags+1) columns.
`neq`	Number of equations in the model. Must equal number of rows in H matrix.
`leads`	Number of leads in the model. Must be a positive integer.
`lags`	Number of lags in the model. Must be a positive integer.
`qmax`	Number of elements to allocate for sparse matrix storage. Optional input. Default is neq * leads * neq * (leads + lags).

Details

Note: for details about what the H, B, Q, S, script A, script B etc. matrices are, please see [Anderson 2010]. You will need to separately load in the data file in the form of the "H" matrix that is, the matrix [H_-T ... H_0 ... H_theta]. For details on how to obtain this matrix given a model and data, see the papers referenced and use the genHmat function. This utility may be added to this R package in the future. Output will be a list of "list" objects i.e. the B, S, etc. matrices will be returned as one-dimensional list objects, embedded within a list. It is pretty straightforward to get these into matrix form (use outObject[k][[1]] to get the kth output where outObject is the result of callSparseAim). This applies to the B (structural coefficients) and Q (asymptotic linear constraints) matrices. Alternatively, you can also use the genBmat and genQmat functions to get these matrices directly, as matrix objects. You can also obtain the S (observable structure) matrix as well as the stochastic transition matrices, labelled script A and script B in [Anderson 2010].

Value

The following list object: [cofh, hrows, hcols, neq, leads, lags, nstate, qmax, ptr ,cofb, qmat] Where: The first six elements (cofh, ..., lags) are the same as the inputs nstate = leads + lags + 1 qmax is same as on input ptr is a pointer to the integer corresponding to the error code; 0 denotes successful output. See getReturnCode() to "translate". cofb = reduced form coefficients matrix qmat = asymptotic linear constraints matrix

Note

It is preferable to ensure that the matrix H is as you want it before passing it as an argument to this function (and to genBmat, genQmat, etc.) However, if you wish to simply load the matrix H from a text file with neq*(nlag+nlead+1) lines, make sure to store the entries column-wise as this is R's default method of storage.

If you are only interested in, for example, the B matrix or the S matrix, then it is not necessary to call this function. Use genBmat or genScof instead–both call this function as well. To extract a matrix from the jth element of the list returned by this function, use result[j][[1]] (where result is the name of the list).

Author(s)

Gary Anderson and Aneesh Raghunandan

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`genBmat`, `genScof`, `genQmat`, `genHmat`, `getStochTrans`, `getFactorMatrices`

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
output <- callAMA(hmat, neq, leads, lags)
```

---

example7 *example7.mod – Text input for example*

Description

Contains model definition

Usage

```
example7.mod
```

Format

Uses modelez format

Examples

```
hmat<-genHmat(system.file("extdata/example7.mod",package="AMA"),system.file("extdata/example7params.prm",
bmat<-genBmat(hmat,4,1,1)
```

`example7params` *example7params.txt – Text input for example*

Description

Contains parameter assignments for the example7 matrix assignments.

Format

Rows of assignment statements for model parameters.

Examples

```
hmat<-genHmat(system.file("extdata/example7.mod",package="AMA"),system.file("extdata/example7params.prm",
bmat<-genBmat(hmat,4,1,1)
```

`genBmat` *Reduced-Form Coefficients Matrix*

Description

Computes the reduced-form coefficients matrix ("B") output by the Anderson-Moore Algorithm. Arguments are identical to those passed to callSparseAim.

Usage

```
genBmat(cofh, neq, leads, lags, qmax)
```

Arguments

`cofh`	Full H matrix, passed to R columnwise. The H matrix is the matrix [H_-T ... H_0 ... H_theta]. Must have neq rows and neq*(leads+lags+1) columns.
`neq`	Number of equations in the model. Must equal number of rows in H matrix.
`leads`	Number of leads in the model. Must be a positive integer.
`lags`	Number of lags in the model. Must be a positive integer.
`qmax`	Number of elements to allocate for sparse matrix storage. Optional input. Default is neq * leads * neq * (leads + lags).

Value

cofb ("B" matrix)

Note

Alternatively, can be called from the object that callAMA returns; i.e. if res is the output of callAMA, then use "bmat = res[10][[1]]".

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`

Examples

```
hmat <- read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
bmat <- genBmat(hmat, 4, 1, 1)


hmat<-genHmat(system.file("extdata/example7.mod",package="AMA"),system.file("extdata/example7params.prm",
bmat<-genBmat(hmat,4,1,1)
```

`genHmat` *Structural Coefficients Matrix Scripts*

Description

Creates the structural coefficients matrix H, given the model file and parameter values as inputs.

Usage

```
genHmat(modelFileNameFull, params, wantParamVec = FALSE)
```

Arguments

modelFileNameFull
: String object containing the name of the model file. Model file must be in MODELEZ syntax.

params
: Either: -A string object containing the name of the parameters file (One parameter per line, with each line in the format "param = <value>") OR: -A vector containing the parameters. Vector must have names associated with each entry (i.e. names(params) should be correct and complete)

wantParamVec
: If TRUE, then the output of the function is a list object containing as its first element the H matrix and as its second element the vector of parameters. Default is FALSE. You should probably set this to TRUE if you're reading in from a parameter file, and leave it as FALSE if you already have a parameter vector.

Details

Model input file must be in MODELEZ syntax.

Value

Structural coefficients matrix H.

References

For the details of MODELEZ syntax, see for example http://www.federalreserve.gov/pubs/oss/oss4/papers/habitMatlab/habitMatlab.html

For details about the algorithm:

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

callAMA

Examples

```
hmat<-genHmat(system.file("extdata/example7.mod",package="AMA"),system.file("extdata/example7params.prm",
bmat<-genBmat(hmat,4,1,1)
```

`genHmatrixScripts` *Structural Coefficients Matrix Scripts*

Description

Creates an R file in the present working directory which, when run, creates the matrix H.

Usage

```
genHmatrixScripts(modelFileNameFull)
```

Arguments

`modelFileNameFull`
: String object containing the name of the model file. Model file must be in MODELEZ syntax.

Details

Model input file must be in MODELEZ syntax.

Value

An R script, in the current directory, called "<model name>_sparseAimMatrices.r", which when run generates the matrix H.

References

For the details of MODELEZ syntax, see for example `http://www.federalreserve.gov/pubs/oss/oss4/papers/habitMatlab/habitMatlab.html`

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`

Examples

```
genHmatrixScripts(system.file("extdata/example7.mod",package="AMA"))
```

`genQmat` *Asymptotic Linear Constraints*

Description

Computes the asymptotic linear constraints matrix ("Q") used by the Anderson-Moore Algorithm. Arguments are identical to those passed to callSparseAim.

Usage

```
genQmat(cofh, neq, leads, lags, qmax)
```

Arguments

`cofh`	Full H matrix, passed to R columnwise. The H matrix is the matrix [H_-T ... H_0 ... H_theta]. Must have neq rows and neq*(leads+lags+1) columns.
`neq`	Number of equations in the model. Must equal number of rows in H matrix.
`leads`	Number of leads in the model. Must be a positive integer.
`lags`	Number of lags in the model. Must be a positive integer.
`qmax`	Number of elements to allocate for sparse matrix storage. Optional input. Default is neq * leads * neq * (leads + lags).

Value

qmat (asymptotic linear constraints matrix)

Note

Alternatively, can be called from the object that callAMA returns; i.e. if res is the output of callAMA, then use "qmat = res[11][[1]]".

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
Qmatrix <- genQmat(hmat, neq, leads, lags)
```

---

genScof *Observable Structure Matrix*

Description

Computes the observable structure matrix ("S") using outputs from the Anderson-Moore Algorithm.

Usage

```
genScof(hmat, bmat, neq, leads, lags)
```

Arguments

`hmat`	Full H matrix, passed to R columnwise. The H matrix is the matrix [H_-T ... H_0 ... H_theta].
`bmat`	The B (reduced-form coefficients) matrix.
`neq`	Number of equations in the model.
`leads`	Number of leads in the model. Must be a positive integer.
`lags`	Number of lags in the model. Must be a positive integer.

Value

cofS (observable structure matrix)

Note

Alternatively, can be called from the object that callAMA returns; i.e. if res is the output of callAMA, then use "bmat = res[10][[1]]".

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

callAMA, genBmat

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
qmax = 150000
bmat <- genBmat(hmat, neq, leads, lags)
Scoef <- genScof(hmat, bmat, neq, leads, lags)
```

---

getFactorMatrices *Shock Factor Matrices for Inhomogeneous Solutions*

Description

Computes the exogenous shock scaling (phi) and transfer (F) matrices used in the inhomogeneous case (i.e.

Usage

```
getFactorMatrices(hmat, bmat, neq, leads, lags)
```

Arguments

hmat	Full H matrix, passed to R columnwise. The H matrix is the matrix [H_-T ... H_0 ... H_theta]. Must have neq rows and neq*(leads+lags+1) columns.
bmat	Full B matrix, as generated either by callSparseAim or genBmat.
neq	Number of equations in the model. Must equal number of rows in H matrix.
leads	Number of leads in the model. Must be a positive integer.
lags	Number of lags in the model. Must be a positive integer.

Details

[to be filled in]

Value

list object containing phi as first element and F as second element.

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`, `genBmat`

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
bmat <- genBmat(hmat, neq, leads, lags)
hmat = matrix(as.matrix(hmat), neq)
factorMats <- getFactorMatrices(hmat, bmat, neq, leads, lags)
phi = factorMats[1][[1]]
F = factorMats[2][[1]]
```

---

getReturnCode *Return Code*

Description

Computes the asymptotic linear constraints matrix ("Q") used by the Anderson-Moore Algorithm. Arguments are identical to those passed to callSparseAim.

Usage

```
getReturnCode(sparseAimObject)
```

Arguments

`sparseAimObject`
: Output object from callSparseAim.

Details

[to be filled in]

Value

Return code (corresponds to 0 if call to SparseAIM is successful; prints error message otherwise)

Note

Integer corresponding to error code can be extracted directly from the object that callSparseAim returns; i.e. if res is the output of callSparseAim, then use "returnCode = res[9][[1]]".

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
output <- callAMA(hmat, neq, leads, lags)
errorMsg = getReturnCode(output)
```

---

getStochTrans *Stochastic Transition Matrices*

Description

Computes the stochastic transition matrices ("script A" and "script B") used by the Anderson-Moore Algorithm. Matrices H and S must already be matrix objects of proper dimension.

Usage

```
getStochTrans(hmat, scof)
```

Arguments

hmat	Full H matrix object. The H matrix is the matrix [H_-T ... H_0 ... H_theta]. (i.e. T lags and theta leads)
scof	Matrix S ("observable structure") .

Details

A and B are the solution to [x_t-T+1 ... x_t]' = A[x_t-T ...xt-1]' + B[e_t + psi*(E[z_t|I_t] - E[z_t|I_t-1)]

Value

A, B (returned as a "list" object; the first element is A and the second is B.)

References

Gary S. Anderson. "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models." Journal of Economic Dynamics & Control, 2010.

Gary Anderson and George Moore. "An Efficient Procedure for Solving Linear Perfect Foresight Models." Unpublished manuscript, Board of Governors of the Federal Reserve System. 1983.

See Also

`callAMA`, `genScof`, `genBmat`

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
hmatrix = matrix((as.matrix(hmat)), neq)
leads = 1
lags = 1
bmat <- genBmat(hmat, neq, leads, lags)
Scoef <- genScof(hmat, bmat, neq, leads, lags)
stochMatrices <- getStochTrans(hmatrix, Scoef)
scriptA = stochMatrices[1][[1]]
scriptB = stochMatrices[2][[1]]
```

`Lcofbob2.cof` *Lcofbob2.cof Example Input data*

Description

Input matrix for example

Usage

`Lcofbob2.cof`

Format

Matrix elements columnwise.

Examples

```
hmat = read.table(system.file("extdata/Lcofbob2.cof",package="AMA"))
hrows = 4
hcols = 12
neq = 4
leads = 1
lags = 1
output <- callAMA(hmat, neq, leads, lags)
```