The ULV/ULLV Toolbox for Matlab

Peter S. K. Hansen

Department of Mathematical Modelling
Section for Digital Signal Processing
Technical University of Denmark
Building 321, DK-2800 Lyngby, Denmark

Phone: (+45) 45253922
FAX: (+45) 45872599
Email: pskh@imm.dtu.dk
URL: http://www.imm.dtu.dk

Version 1.0
April 14, 1997

Contents

Chapter 1

Introduction

The ULV/ULLV Toolbox consists of a number of MATLAB routines for calculation of Rank-Revealing ULV/ULLV Decompositions. This includes initial calculation of the decompositions and up/downdating methods.

The algorithms are based on Givens rotations, i.e., for-loops, resulting in a slow execution of MATLAB M-files. Therefore, a C-version is also available with a MEX-file interface. Totally, the package consist of 31 M-files and 53 C-files, and the present reference documentation covers the M-version.

One of the inspirations for the toolbox was a set of MATLAB routines concerning rank-revealing ULLV updating kindly provided to us by S. Qiao, together with the papers referred in the next sections.

1.1 Rank-Revealing ULV/ULLV Decompositions

The Singular Value Decomposition (SVD) and the Quotient Singular Value Decomposition (QSVD) are robust and widely used computational tools in many applications. The SVD is the most reliable tool for detecting numerical rank-deficiency in a matrix [5, page 246] and it also yields detailed subspace information. Unfortunately, the SVD is computationally expensive and resist updating. The initial cost of computing a SVD would not be an objection, if the decomposition could be cheaply updated. However, all known updating schemes require $\mathcal{O}(n^3)$ flops for a matrix of order n even for a simple update such as adding a new row. To circumvent these drawbacks any decomposition that estimates the rank and the orthogonal spaces can be used in the place of the SVD.

The Rank-Revealing QR (RRQR) decomposition could be one possible choice [3]. RRQR algorithms can track the rank, and the computational complexity of this approach is $\mathcal{O}(n^2)$. However, it has been shown [4] that the quality of the approximation depends on the gap between the singular values, and approximations to the right singular subspaces are not directly available.

A better choice is the Rank-Revealing URV/ULV Decompositions and the related algorithms devised by Stewart [11, 12, 1, 6], which use two-sided orthogonal transformations to upper/lower triangularize the matrix. Thus, the URV/ULV decompositions are a compromise between the SVD and the RRQR decomposition with some of the virtues of both. The rank-revealing URV/ULV decompositions are effective in tracking the rank and approximations to both the left and right subspaces are explicitly available. Furthermore, the quality of the subspace approximations are not sensitive to the magnitude of the gap in the singular values, and the decompositions can be updated in $\mathcal{O}(n^2)$ flops. .

Another issue is the precision in the estimated row and column subspaces, which is not identical for the upper and lower triangular version. Therefore the ULV decomposition is normally preferred when right subspaces are desired.

$$\mathbf{A} = \mathbf{U}\,\mathbf{L}_A\mathbf{V}^T \tag{1.1}$$

For a pair of matrices the ULLV decomposition [9, 10, 8, 2] can be used, which is an approximation of the QSVD, and is also an extension of the ULV decomposition.

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_B \end{pmatrix} \begin{pmatrix} \mathbf{L}_A \\ \mathbf{I} \end{pmatrix} \mathbf{L}\,\mathbf{V}^T \tag{1.2}$$

1.2 Up- and Downdating Issues

Rank-revealing ULV/ULLV decompositions are often used in connection with up- and downdating problems arising from recursive algorithms in signal processing. Starting with the initial matrices, the decompositions are updated as the data is brought in one row at a time. A new row is processed by some or all of the following steps.

- Updating: A new row of data is incorporated into the decomposition.
- Downdating: The oldest row of data is isolated and removed in the decomposition.
- Deflation: Establish and maintain the rank-revealing nature of the decomposition.
- Refinement: The norm of the off-diagonal block in the rank-revealing triangular matrix $\mathbf{L}_A$ is reduced to improve the subspace quality.

The update method was described by Luk and Qiao [9, 10], while the downdating scheme is based on work by Bojanczyk and Lebak [2, 8]. Finally, the deflation and refinement steps are based on those defined by Stewart for the URV and ULV decompositions [11, 12]. More details about the algorithms, can be found in the reference [7].

Chapter 2

Toolbox Reference

In this chapter, the installation of the ULV/ULLV Toolbox is described, followed by an overview of the files. Finally, three sections with the manual pages are given.

2.1 Installation

The ULV/ULLV Toolbox is distributed as a compressed Unix tar-file `UllvBox.tgz`, available by anonymous ftp from Internet address `ftp.imm.dtu.dk` in directory `out/pskh`. This document is a compressed ps-file `UllvBox.ps.gz` in the same location.

To install the toolbox, download the tar-file into the directory, where you want the ULV/ULLV toolbox subdirectory to be, then type

```
tar xzvf UllvBox.tgz
```

A directory `UllvBox` will be created containing a number of subdirectories with M-files and C-files arranged as follows

```
UllvBox - MatFun
          Ulv
          Ullv
          Mex - libAlloc
                libNum
                Ulv
                Ullv
```

To generate the MEX-files, go to the `UllvBox` directory and type `make`. This will do all the job for you.

Now you can either append the relevant directories to the MATLAB path or copy the files to a known location.

2.2 Quick Reference Tables, M-files

This section contains quick reference tables to the ULV/ULLV Toolbox. All the M-files in the toolbox are listed by category in the three tables with a short description. Manual pages are given in Section 2.4–2.6, and they can also be obtained through the on-line help facility.

Rank-Revealing ULV Decomposition	
rrulvd	Initial computation of the rank-revealing ULV decomposition.
ulv_x	Up/downdating the rank-revealing ULV decomposition.
ulv_up	Updating the ULV decomposition.
ulv_dw	Downdating the ULV decomposition.
ulv_rr	Rank-revealing algorithm for the ULV up/downdating.
ulv_def	Deflation on $\mathbf{L}$ in the ULV decomposition.
ulv_ref	One iteration of refinement in the ULV decomposition.

Rank-Revealing ULLV Decomposition	
rrullvd	Initial computation of the rank-revealing ULLV decomposition.
ullv_a	Up/downdating the $\mathbf{A}$-part of the rank-revealing ULLV decomp.
ullv_b	Up/downdating the $\mathbf{B}$-part of the rank-revealing ULLV decomp.
ullv_upa	Updating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_upb	Updating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_dwa	Downdating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_dwb	Downdating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_rr	Rank-revealing algorithm for the ULLV up/downdating.
ullv_def	Deflation on $\mathbf{L}_A$ in the ULLV decomposition.
ullv_ref	One iteration of refinement in the ULLV decomposition.

Matrix Tools	
pythag	Pythagoras equation.
back_sub	Column version of Back Substitution.
forward_sub	Column version of Forward Substitution.
given_cs	Givens rank-two rotation matrix.
given_rt	Applying Givens rotations (left/right).
exp_mgsr	Modified Gram-Schmidt expansion.
exp_csneL	Corrected Semi-Normal Equations expansion in an ULV-decomp.
exp_csneLL	Corrected Semi-Normal Equations expansion in an ULLV-decomp.
ql_house	QL-factorization using Householder transformations.
ql_mgs	QL-factorization using Modified Gram-Schmidt.
svd_t22	SVD of 2×2 upper triangular matrix (real).
vmin_ccvl	Generalized LINPACK estimator of the smallest singular value.
vmin_co	Convex optimization based estimator of the smallest singular value.
vmin_lb	Look-behind based estimator of the smallest singular value.

2.3 Quick Reference Tables, C-files

The MEX-file version of the toolbox is almost identical to the M-file version, but of course some of the build-in functionality of MATLAB, such as the norm functions, the QR-decomposition etc., is implemented.

Also the C-files in the toolbox are listed by category in the following four tables with a short description, but manual pages are only available from the file headers.

Rank-Revealing ULV Decomposition	
rrulvd_mex	MEX gateway function to be called from MATLAB.
rrulvd	Initial computation of the rank-revealing ULV decomposition.
ulv_x_mex	MEX gateway function to be called from MATLAB.
ulv_x	Up/downdating the rank-revealing ULV decomposition.
ulv_up	Updating the ULV decomposition.
ulv_dw	Downdating the ULV decomposition.
ulv_rr	Rank-revealing algorithm for the ULV up/downdating.
ulv_def	Deflation on $\mathbf{L}$ in the ULV decomposition.
ulv_ref	One iteration of refinement in the ULV decomposition.

Rank-Revealing ULLV Decomposition	
rrullvd_mex	MEX gateway function to be called from MATLAB.
rrullvd	Initial computation of the rank-revealing ULLV decomposition.
ullv_a_mex	MEX gateway function to be called from MATLAB.
ullv_a	Up/downdating the $\mathbf{A}$-part of the rank-revealing ULLV decomp.
ullv_b_mex	MEX gateway function to be called from MATLAB.
ullv_b	Up/downdating the $\mathbf{B}$-part of the rank-revealing ULLV decomp.
ullv_upa	Updating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_upb	Updating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_dwa	Downdating the $\mathbf{A}$-part of the ULLV decomposition.
ullv_dwb	Downdating the $\mathbf{B}$-part of the ULLV decomposition.
ullv_rr	Rank-revealing algorithm for the ULLV up/downdating.
ullv_def	Deflation on $\mathbf{L}_A$ in the ULLV decomposition.
ullv_ref	One iteration of refinement in the ULLV decomposition.

Matrix Tools	
vnorm2	Vector 2-norm.
mnorm1	Matrix 1-norm.
mnormf	Matrix Frobenius-norm.
mnorminf	Matrix inf-norm.
back_sub	Column version of Back Substitution.
forward_sub	Column version of Forward Substitution.
given_cs	Givens rank-two rotation matrix.
given_lt	Applying Givens Rotations (left).
given_rt	Applying Givens Rotations (right).
house_cv	Householder column vector.
house_rv	Householder row vector.
house_lt	Householder pre-multiplication.
house_rt	Householder post-multiplication.
house_p	Product of Householder matrices (right transformations).
house_q	Product of Householder matrices (left transformations).
solveXLY	Solve triangular system with multiple right-hand sides.
exp_mgsr	Modified Gram-Schmidt expansion.
exp_csneL	Corrected Semi-Normal Equations expansion in an ULV-decomp.
exp_csneLL	Corrected Semi-Normal Equations expansion in an ULLV-decomp.
qr_house	QR-factorization using Householder transformations.
qr_mgs	QR-factorization using Modified Gram-Schmidt.
ql_house	QL-factorization using Householder transformations.
ql_mgs	QL-factorization using Modified Gram-Schmidt.
vmin_ccvl	Generalized LINPACK estimator of the smallest singular value.
vmin_co	Convex optimization based estimator of the smallest singular value.

Memory Allocation Tools	
dvector	Allocates a double vector.
dmatrix	Allocates a double matrix.
matlab2pp	Change matrix pointer format to matrix pointer-pointer format.
free_dvec	Frees memory allocated by dvector.
free_dmat	Frees memory allocated by dmatrix.
free_mat2pp	Frees memory allocated by matlab2pp.

2.4 Rank-Revealing ULV Decomposition

rrulvd

Purpose

Initial computation of the rank-revealing ULV decomposition.

Synopsis

```
[U,L,V,rr_out] = rrulvd(X,rr_in,rrflag,nref,uflag)
```

Description

The function computes a rank-revealing ULV decomposition X=U*L*V', where the n-by-n matrix L is lower triangular and will reveal the rank p of X. Thus, the norm of the (2,1) and (2,2) blocks of L are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than rr_in, where rr_in is a given numerical tolerance. The rank is returned in rr_out. If rr_in is not specified (=zero), SQRT(n)*norm(X,1)*eps is default. If rrflag=1, then deflation is done to the constant rank rr_in, and the corresponding numerical tolerance is returned in rr_out. nref number of refinement iterations are done after each deflation step. The matrix U is not calculated if uflag=0.

Algorithm

First find the QL decomposition of X=U*L, i.e., X=U*L*I. Then deflate and refine the decomposition.

See Also

ulv_def, ulv_ref, vmin_co, ulv_x

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_def

Purpose

Deflation on L in the ULV decomposition.

Synopsis

```
[U,L,V] = ulv_def(U,L,V,vmin,uflag,r,m,n)
```

Description

Deflation on L in the ULV decomposition. Given a m-by-n matrix X=U*L*V', the function deflate L(1:r,1:r). vmin is an estimate of the right singular vector of L(1:r,1:r) associated with the smallest singular value. On return norm(L(r,1:r)) is of the order sigma_(r). The matrix U is not calculated if uflag=0.

See Also

given_cs, given_rt, ulv_rr, ulv_ref, ulv_x, rrulvd

References

[1] G.W.Stewart, "An Updating Algorithm for Subspace Tracking", IEEE Trans. on SP, Vol. 40, No. 6, pp. 1535-1541, June, (1992).

[2] G.W.Stewart, "Updating a Rank-Revealing ULV Decomposition", SIAM Jour. on Matrix Analysis and Applications, Vol. 14, No. 2, pp. 494-499, April, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_dw

Purpose

Downdating the ULV decomposition.

Synopsis

```
[U,L,V,normx] = ulv_dw(X,U,L,V,uflag,kappa,m,n)
```

Description

Downdating the ULV decomposition. Given a m-by-n matrix X=U*L*V', the function computes the downdated decomposition

```
X = [   x   ]
    [U*L*V']
```

where x is the top row being removed from X, and normx is the norm of this row. The matrix U is not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed in the expansion step.

See Also

given_cs, given_rt, exp_mgsr, exp_csneL, ulv_x

References

[1] A.W.Bojanczyk, "Modifications of a Generalized ULV Decomposition", Technical Report, School of Electrical Engineering, Cornell University, Ithaca, NY, (1993).

[2] J.M.Lebak and A.W.Bojanczyk, "Modifying a Rank-Revealing ULLV Decomposition", Technical Report, Cornell Theory Center, June, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_ref

Purpose

One iteration of refinement in the ULV decomposition.

Synopsis

```
[U,L,V] = ulv_ref(U,L,V,uflag,r,m,n)
```

Description

One iteration of refinement in the ULV decomposition. Given a m-by-n matrix X=U*L*V', the function refine the last row of L(1:r,1:r). The matrix U is not calculated if uflag=0.

Algorithm

Refinement is an iterative algorithm, which move a triangular matrix L toward diagonality by flipping the matrix to upper triangular form and down again.

See Also

given_cs, given_rt, ulv_rr, ulv_def, ulv_x, rrulvd

References

[1] G.W.Stewart, "An Updating Algorithm for Subspace Tracking", IEEE Trans. on SP, Vol. 40, No. 6, pp. 1535-1541, June, (1992).

[2] G.W.Stewart, "Updating a Rank-Revealing ULV Decomposition", SIAM Jour. on Matrix Analysis and Applications, Vol. 14, No. 2, pp. 494-499, April, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_rr

Purpose

Rank-revealing algorithm for the ULV up/downdating.

Synopsis

```
[U,L,V,tau,p] = ulv_rr(U,L,V,tau,p,rrflag,uflag,nref,m,n)
```

Description

Rank-revealing algorithm for the ULV decomposition to be used after the update or downdate step. Given a m-by-n matrix X=U*L*V', the function deflate and refine the lower triangular matrix L to reveal the rank p of X. Thus, the norm of the (2,1) and (2,2) blocks of L are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than tau, where tau is a given numerical tolerance. If rrflag=1, then deflation is done to the constant rank p. nref number of refinement iterations are done after each deflation step. The matrix U is not calculated if uflag=0.

See Also

ulv_def, ulv_ref, vmin_co, ulv_x

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_up

Purpose

Updating the ULV decomposition.

Synopsis

```
[U,L,V] = ulv_up(U,L,V,x,beta,uflag,m,n)
```

Description

Updating the ULV decomposition. Given a m-by-n matrix X=U*L*V', the function computes the updated decomposition

```
[beta*X] = U*L*V'
[  x   ]
```

where x is the new row added to X, and beta is a forgetting factor in [0;1]. The matrix U is not calculated if uflag=0.

See Also

given_cs, given_rt, ulv_x

References

[1] G.W.Stewart, "An Updating Algorithm for Subspace Tracking", IEEE Trans. on SP, Vol. 40, No. 6, pp. 1535-1541, June, (1992).

[2] G.W.Stewart, "Updating a Rank-Revealing ULV Decomposition", SIAM Jour. on Matrix Analysis and Applications, Vol. 14, No. 2, pp. 494-499, April, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ulv_x

Purpose

Up/downdating the rank-revealing ULV decomposition.

Synopsis

```
[X,U,L,V,normx,rr] = ulv_x(X,U,L,V,x,beta,tau,...
                           p,nref,rrflag,kappa,uflag)
```

Description

The rank-revealing ULV decomposition. Given a m-by-n matrix X=U*L*V', the function computes the updated decomposition

```
[beta*X] = U*L*V'
[  x   ]
```

where x is the new row added to X, and beta is a forgetting factor in [0;1]. If beta is less than one, then exponential windowing is used. If beta=1, then normx is the norm of the top row being removed from the updated X after downdating (sliding window). The matrix U is not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed in the expansion step. Deflation and refinement are applied to the lower triangular matrix L to reveal the rank p of X. Thus, the norm of the (2,1) and (2,2) blocks of L are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than tau, where tau is a given numerical tolerance. The new rank is returned in rr. If rrflag=1, then deflation is done to the constant rank p, and the corresponding numerical tolerance is returned in rr. nref number of refinement iterations are done after each deflation step.

See Also

ulv_up, ulv_dw, ulv_rr, rrulvd

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

2.5 Rank-Revealing ULLV Decomposition

rrullvd

Purpose

Initial computation of the rank-revealing ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V,rr_out] = rrullvd(A,B,rr_in,rrflag,nref,uflag)
```

Description

The function computes a rank revealing ULLV decomposition A=UA*LA*L*V' and B=UB*L*V'. The n-by-n matrix L is lower triangular and the lower triangular matrix LA will reveal the rank p of A*pseudoinverse(B). Thus, the norm of the (2,1) and (2,2) blocks of LA are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than rr_in, where rr_in is a given numerical tolerance. The rank is returned in rr_out. If rr_in is not specified (=zero), SQRT(n)*norm(A,1)*eps is default. If rrflag=1, then deflation is done to the constant rank rr_in, and the corresponding numerical tolerance is returned in rr_out. nref number of refinement iterations are done after each deflation step. The matrices UA and UB are not calculated if uflag=0.

Algorithm

First find the QL decomposition of B=UB*L and solve X=A*L_inv followed by another QL decomposition of X=UA*LA. Thus, A=UA*LA*L*I and B=UB*L*I. Then deflate and refine the decomposition.

See Also

ullv_def, ullv_ref, vmin_co, ullv_a, ullv_b

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_a

Purpose

Up/downdating the A-part of the rank-revealing ULLV decomp.

Synopsis

```
[A,UA,UB,LA,L,V,norma,rr] = ullv_a(A,UA,UB,LA,L,V,a,beta,tau,...
                                   p,nref,rrflag,kappa,uflag)
```

Description

The rank-revealing ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function computes the updated decomposition

```
[beta*A] = UA*LA*L*V'  and  B = UB*L*V'
[  a   ]
```

where a is the new row added to A, and beta is a forgetting factor in [0;1]. If beta is less than one, then exponential windowing is used. If beta=1, then norma is the norm of the top row being removed from the updated A after downdating (sliding window). The matrices UA and UB are not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed in the expansion step. Deflation and refinement are applied to the lower triangular matrix LA to reveal the rank p of A*pseudoinverse(B). Thus, the norm of the (2,1) and (2,2) blocks of LA are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than tau, where tau is a given numerical tolerance. The new rank is returned in rr. If rrflag=1, then deflation is done to the constant rank p, and the corresponding numerical tolerance is returned in rr. nref number of refinement iterations are done after each deflation step.

See Also

ullv_upa, ullv_dwa, ullv_rr, ullv_b, rrullvd

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_b

Purpose

Up/downdating the B-part of the rank-revealing ULLV decomp.

Synopsis

```
[B,UA,UB,LA,L,V,normb,rr] = ullv_b(B,UA,UB,LA,L,V,b,beta,tau,...
                                    p,nref,rrflag,kappa,uflag)
```

Description

The rank-revealing ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function computes the updated decomposition

```
A = UA*LA*L*V'  and  [beta*B] = UB*L*V'
                     [  b   ]
```

where b is the new row added to B, and beta is a forgetting factor in [0;1]. If beta is less than one, then exponential windowing is used. If beta=1, then normb is the norm of the top row being removed from the updated B after downdating (sliding window). The matrices UA and UB are not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if reorthogonalization is needed in the expansion step. Deflation and refinement are applied to the lower triangular matrix LA to reveal the rank p of A*pseudoinverse(B). Thus, the norm of the (2,1) and (2,2) blocks of LA are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than tau, where tau is a given numerical tolerance. The new rank is returned in rr. If rrflag=1, then deflation is done to the constant rank p, and the corresponding numerical tolerance is returned in rr. nref number of refinement iterations are done after each deflation step.

See Also

ullv_upb, ullv_dwb, ullv_rr, ullv_a, rrullvd

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_def

Purpose

Deflation on LA in the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V] = ullv_def(UA,UB,LA,L,V,vmin,uflag,r,mA,mB,n)
```

Description

Deflation on LA in the rank-revealing ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function deflate LA(1:r,1:r). vmin is an estimate of the right singular vector of LA(1:r,1:r) associated with the smallest singular value. On return norm(LA(r,1:r)) is of the order sigma_(r). The matrices UA and UB are not calculated if uflag=0.

See Also

given_cs, given_rt, ullv_rr, ullv_ref, ullv_a, ullv_b, rrullvd

References

[1] S.Qiao, "Computing the ULLV Decomposition", CRL Report 278, Communications Research Laboratory, McMaster Uni., Hamilton, Canada, pp. 1-13, January, (1994).

[2] G.W.Stewart, "An Updating Algorithm for Subspace Tracking", IEEE Trans. on SP, Vol. 40, No. 6, pp. 1535-1541, June, (1992).

[3] G.W.Stewart, "Updating a Rank-Revealing ULV Decomposition", SIAM Jour. on Matrix Analysis and Applications, Vol. 14, No. 2, pp. 494-499, April, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_dwa

Purpose

Downdating the A-part of the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V,norma] = ullv_dwa(A,UA,UB,LA,L,V,uflag,kappa,mA,mB,n)
```

Description

Downdating the ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function computes the downdated decomposition

```
A = [     a     ]  and  B = UB*L*V'
    [UA*LA*L*V']
```

where a is the top row being removed from A, and norma is the norm of this row. The matrices UA and UB are not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed in the expansion step.

See Also

given_cs, given_rt, exp_mgsr, exp_csneLL, ullv_a, ullv_b

References

[1] A.W.Bojanczyk, "Modifications of a Generalized ULV Decomposition", Technical Report, School of Electrical Engineering, Cornell University, Ithaca, NY, (1993).

[2] J.M.Lebak and A.W.Bojanczyk, "Modifying a Rank-Revealing ULLV Decomposition", Technical Report, Cornell Theory Center, June, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_dwb

Purpose

Downdating the B-part of the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V,normb] = ullv_dwb(B,UA,UB,LA,L,V,uflag,kappa,mA,mB,n)
```

Description

Downdating the ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function compute the downdated decomposition

```
A = UA*LA*L*V'  and  B = [   b   ]
                         [UB*L*V']
```

where b is the top row being removed from B, and normb is the norm of this row. The matrices UA and UB are not calculated if uflag=0. Thus, if uflag=1, then the modified Gram-Schmidt algorithm is used in the expansion step in the downdating algorithm, and if uflag=0, the method of corrected semi-normal equations is used. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed in the expansion step.

See Also

given_cs, given_rt, exp_mgsr, exp_csneL, ullv_a, ullv_b

References

[1] A.W.Bojanczyk, "Modifications of a Generalized ULV Decomposition", Technical Report, School of Electrical Engineering, Cornell University, Ithaca, NY, (1993).

[2] J.M.Lebak and A.W.Bojanczyk, "Modifying a Rank-Revealing ULLV Decomposition", Technical Report, Cornell Theory Center, June, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_ref

Purpose

One iteration of refinement in the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V] = ullv_ref(UA,UB,LA,L,V,uflag,r,mA,mB,n)
```

Description

One iteration of refinement in the ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function refine the last row of LA(1:r,1:r). The matrices UA and UB are not calculated if uflag=0.

Algorithm

Refinement is an iterative algorithm, which move a triangular matrix L toward diagonality by flipping the matrix to upper triangular form and down again.

See Also

given_cs, given_rt, ullv_rr, ullv_def, ullv_a, ullv_b, rrullvd

References

[1] S.Qiao, "Computing the ULLV Decomposition", CRL Report 278, Communications Research Laboratory, McMaster Uni., Hamilton, Canada, pp. 1-13, January, (1994).

[2] G.W.Stewart, "An Updating Algorithm for Subspace Tracking", IEEE Trans. on SP, Vol. 40, No. 6, pp. 1535-1541, June, (1992).

[3] G.W.Stewart, "Updating a Rank-Revealing ULV Decomposition", SIAM Jour. on Matrix Analysis and Applications, Vol. 14, No. 2, pp. 494-499, April, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_rr

Purpose

Rank-revealing algorithm for the ULLV up/downdating.

Synopsis

```
[UA,UB,LA,L,V,tau,p] = ullv_rr(UA,UB,LA,L,V,tau,p,rrflag,...
                               uflag,nref,mA,mB,n)
```

Description

Rank-revealing algorithm for the ULLV decomposition to be used after the update or downdate step. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function deflate and refine the lower triangular matrix LA to reveal the rank p of A*pseudoinverse(B). Thus, the norm of the (2,1) and (2,2) blocks of LA are of the order sigma_(p+1). If rrflag=0, then deflation is done to sigma_(p) is greater than tau, where tau is a given numerical tolerance. If rrflag=1, then deflation is done to the constant rank p. nref number of refinement iterations are done after each deflation step. The matrices UA and UB are not calculated if uflag=0.

See Also

ullv_def, ullv_ref, vmin_co, ullv_a, ullv_b

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_upa

Purpose

Updating the A-part of the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V] = ullv_upa(UA,UB,LA,L,V,a,beta,uflag,mA,mB,n)
```

Description

Updating the ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function computes the updated decomposition

```
[beta*A] = UA*LA*L*V'  and  B = UB*L*V'
[  a   ]
```

where a is the new row added to A, and beta is a forgetting factor in [0;1]. The matrices UA and UB are not calculated if uflag=0.

See Also

given_cs, given_rt, ullv_upb, ullv_a, ullv_b

References

[1] F.T.Luk and S.Qiao, "A New Matrix Decomposition for Signal Processing", In "Linear Algebar for Large Scale and Real-Time Applications", pp. 241-247, Kluwer Academic Publishers, (1993).

[2] S.Qiao, "Computing the ULLV Decomposition", CRL Report 278, Communications Research Laboratory, McMaster Uni., Hamilton, Canada, pp. 1-13, January, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ullv_upb

Purpose

Updating the B-part of the ULLV decomposition.

Synopsis

```
[UA,UB,LA,L,V] = ullv_upb(UA,UB,LA,L,V,b,beta,uflag,mA,mB,n)
```

Description

Updating the ULLV decomposition. Given a mA-by-n matrix A=UA*LA*L*V' and mB-by-n matrix B=UB*L*V', the function computes the updated decomposition

```
A = UA*LA*L*V'  and  [beta*B] = UB*L*V'
                     [  b   ]
```

where b is the new row added to B, and beta is a forgetting factor in [0;1]. The matrices UA and UB are not calculated if uflag=0.

See Also

given_cs, given_rt, ullv_upa, ullv_a, ullv_b

References

[1] F.T.Luk and S.Qiao, "A New Matrix Decomposition for Signal Processing", In "Linear Algebar for Large Scale and Real-Time Applications", pp. 241-247, Kluwer Academic Publishers, (1993).

[2] S.Qiao, "Computing the ULLV Decomposition", CRL Report 278, Communications Research Laboratory, McMaster Uni., Hamilton, Canada, pp. 1-13, January, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

2.6 Matrix Tools

back_sub

Purpose

Column version of Back Substitution.

Synopsis

```
z = back_sub(R,z,n)
```

Description

The algorithm overwrites the right-hand side z with the solution x to R*x=z, where R is an upper triangular n-by-n matrix and z is a n-vector.

See Also

forward_sub

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 89, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

exp_csneL

Purpose

Corrected Semi-Normal Equations expansion in an ULV-decomp.

Synopsis

```
[u1,q] = exp_csneL(X,L,V,kappa,m,n)
```

Description

The method of Corrected Semi-Normal Equations obtain the first row u1 of the m-by-n matrix U and the expanted column q, which is orthogonal to the columns of U, by solving the following semi-normal equations in the least-squares sense

```
(X*V)'*(X*V)*z = L'*L*z = (X*V)'*e1
```

where

```
X = U*L*V'
```

The parameter kappa (greater than one) is used to decide if correction of the solution is needed. A typical value for kappa is sqrt(2).

Algorithm

The algorithm is based on triangular solves, and relies on the fact that one re-orthogonalization is always enough. If L is near rank deficient, then the rank information is used in the triangular solves.

See Also

exp_csneLL, exp_mgsr

References

[1] A. Bjorck, H. Park and L. Elden, "Accurate Downdating of Least Squares Solutions", Department of Mathematics, Linkoping University, Sweden.

[2] J.W. Daniel, W.B. Gragg, L. Kaufman and G.W. Stewart, "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization", Mathematics of Computation, vol.30, no. 136, pp. 772-795, October, (1976).

[3] H. Park and L. Elden, "Downdating the Rank-Revealing URV Decomposition", Department of Mathematics, Linkoping University, Sweden, Technical Report, December, (1992).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

exp_csneLL

Purpose

Corrected Semi-Normal Equations expansion in an ULLV-decomp.

Synopsis

```
[ux1,q] = exp_csneLL(X,Lx,L,V,kappa,m,n)
```

Description

The method of Corrected Semi-Normal Equations obtain the first row ux1 of the m-by-n matrix Ux and the expanted column q, which is orthogonal to the columns of Ux, by solving the following semi-normal equations in the least-squares sense

```
(X*V)'*(X*V)*z = (Lx*L)'*(Lx*L)*z = (X*V)'*e1
```

where

```
X = Ux*Lx*L*V'
```

The parameter kappa (greater than one) is used to decide if correction of the solution is needed. A typical value for kappa is sqrt(2).

Algorithm

The algorithm is based on triangular solves, and relies on the fact that one re-orthogonalization is always enough. If Lx is near rank deficient, then the rank information is used in the triangular solves.

See Also

exp_csneL, exp_mgsr

References

[1] A. Bjorck, H. Park and L. Elden, "Accurate Downdating of Least Squares Solutions", Department of Mathematics, Linkoping University, Sweden.

[2] J.W. Daniel, W.B. Gragg, L. Kaufman and G.W. Stewart, "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization", Mathematics of Computation, vol.30, no. 136, pp. 772-795, October, (1976).

[3] J.M.Lebak and A.W.Bojanczyk, "Modifying a Rank-Revealing ULLV Decomposition", Technical Report, Cornell Theory Center, June, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

exp_mgsr

Purpose

Modified Gram-Schmidt expansion.

Synopsis

```
q = exp_mgsr(U,kappa,m,n)
```

Description

The method of Modified Gram-Schmidt with Re-orthogonalization expand the m-by-n matrix U with a new column q, which is orthogonal to the other columns of U. The parameter kappa (greater than one) is used to decide if re-orthogonalization is needed. A typical value for kappa is sqrt(2).

Algorithm

The algorithm relies on the fact that one re-orthogonalization is always enough.

See Also

exp_csneL, exp_csneLL

References

[1] J.W. Daniel, W.B. Gragg, L. Kaufman and G.W. Stewart, "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization", Mathematics of Computation, vol.30, no. 136, pp. 772-795, October, (1976).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

forward_sub

Purpose

Column version of Forward Substitution.

Synopsis

```
z = forward_sub(L,z,n)
```

Description

The algorithm overwrites the right-hand side z with the solution x to L*x=z, where L is a lower triangular n-by-n matrix and z is a n-vector.

See Also

back_sub

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 88, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

given_cs

Purpose

Givens rank-two rotation matrix.

Synopsis

```
[c,s,r] = given_cs(a,b)
```

Description

Given scalars a and b, the function computes r, c=cos(theta) and s=sin(theta) so

```
[ c  s].[a] = [r]  <=>  [a b].[c -s] = [r 0]
[-s  c] [b]   [0]             [s  c]
```

See Also

given_rt

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 202, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

given_rt

Purpose

Applying Givens rotations (left/right).

Synopsis

```
[u1,u2] = given_rt(v1,v2,c,s)
```

Description

Given vectors v1, v2 and scalars c=cos(theta), s=sin(theta) then the following algorithm computes u1 and u2 as:

```
left rotation - row vectors:        [ c  s].[v1]
                                    [-s  c] [v2]

right rotation - column vectors:    [v1 v2].[c -s]
                                            [s  c]
```

See Also

given_cs

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 203, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

pythag

Purpose
Pythagoras equation.

Synopsis
```
x = pythag(y,z)
```

Description
Returns sqrt(y*y + z*z) but is careful to scale to avoid overflow.

Revision
Christian H. Bishof, Argonne National Laboratory, 31.03.89.
Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ql_house

Purpose

QL-factorization using Householder transformations.

Synopsis

```
[Q,L] = ql_house(A)
```

Description

QL-factorization based on the QR-factorization using Householder transformations. Given a m-by-n matrix A, the following function computes the factorization A=QL, where the m-by-n matrix Q has orthonormal columns and the n-by-n matrix L is lower triangular. If there is only one output argument, the matrix L is returned.

See Also

ql_mgs

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 211, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

ql_mgs

Purpose

QL-factorization using Modified Gram-Schmidt.

Synopsis

```
[Q,L] = ql_mgs(A)
```

Description

QL-factorization using Modified Gram-Schmidt. Given a m-by-n matrix A with full rank, the following function computes the factorization A=QL, where the m-by-n matrix Q has orthonormal columns and the n-by-n matrix L is lower triangular.

See Also

ql_house

References

[1] G.H.Golub and C.F.Van Loan, Matrix Computations, John Hopkins University Press, 2.ed., pp. 219, (1989).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

svd_t22

Purpose

SVD of 2x2 upper triangular matrix (real).

Synopsis

```
[smax,smin,cl,sl,cr,sr] = svd_t22(f,g,h)
```

Description

Compute SVD of 2x2 upper triangular matrix (real):

```
T = [f g] = [cl -sl].[smax  0  ].[cr -sr]' = U*S*V'
    [0 h]   [sl  cl] [ 0   smin] [sr  cr]
```

smax is larger singular value and smin is smaller singular value.

References

[1] Z.Bai and J.Demmel, "Computing the Generalized Singular Value Decomposition", SIAM J.Sci.Comput. 14, pp. 1464-1486, (1993).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

vmin_ccvl

Purpose

Generalized LINPACK estimator of the smallest singular value.

Synopsis

```
[smin,vmin] = vmin_ccvl(R,n)
```

Description

Estimator of the smallest singular value smin and its corresponding right singular vector vmin of an upper n-by-n triangular matrix R.

Algorithm

The function is based on the generalized LINPACK codition estimator.

See Also

vmin_co, vmin_lb

References

[1] A. K. Cline, A. R. Conn & C. F. Van Loan, "Generalizing the LINPACK condition estimator", Lecture Notes in Mathematics, Vol. 909, pp. 73-83, Springer Verlag, (1982).

Revision

Christian H. Bishof, Argonne National Laboratory,06.06.89
Modified by Per Christian Hansen, UNI-C, 08.01.91
Modified by C.-T. Pan, NIU, 08.08.92
Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

vmin_co

Purpose
Convex opt. based estimator of the smallest singular value.

Synopsis
```
[smin,vmin] = vmin_co(R,n)
```

Description
Estimator of the smallest singular value smin and its corresponding right singular vector vmin of an upper n-by-n triangular matrix R.

Algorithm
The estimator is based on a convex optimization approach.

See Also
vmin_ccvl, vmin_lb

References
[1] N.Higham, "A Survey of Condition Number Estimation for Triangular Matrices", SIAM Rewiew 29, pp. 575-596, (1987).

Revision
Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

vmin_lb

Purpose

Look-behind based estimator of the smallest singular value.

Synopsis

```
[smin,vmin] = vmin_lb(R,n)
```

Description

Estimator of the smallest singular value smin and its corresponding right singular vector vmin of an upper n-by-n triangular matrix R.

Algorithm

The estimator is based on the look-behind algorithm.

See Also

vmin_ccvl, vmin_co

References

[1] S.Qiao, "Computing the ULLV Decomposition", CRL Report 278, pp. 1-13, January, (1994).

Revision

Peter S. K. Hansen, IMM, DTU, Denmark, April 10, 1997.

Bibliography

[1] G. Adams, M. F. Griffin, and G. W. Stewart. Direction-of-Arrival Estimation Using the Rank-Revealing URV Decomposition. Computer Science Technical Report Series CS-TR-2640, University of Maryland, College Park, MD 20742, March 1991.

[2] A. W. Bojanczyk. Modifications of a Generalized ULV Decomposition. School of Electrical Engineering, Cornell University, Ithaca, NY 14853-3801, 1993.

[3] T. F. Chan. Rank Revealing QR Factorizations. *Linear Algebra and Its Applications*, 88:67–82, 1987.

[4] Ricardo D. Fierro and James R. Bunch. Bounding the Subspaces from Rank Revealing Two-Sided Orthogonal Decompositions. *SIAM Journal on Matrix Analysis and Applications*, 16(3):743–759, July 1995.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 2 edition, 1989.

[6] M. F. Griffin, E. C. Boman, and G. W. Stewart. Minimum-Norm Updating with the Rank-Revealing URV Decomposition. In *ICASSP–92*, volume 5, pages 293–296. IEEE, 1992.

[7] P. S. K. Hansen. *Noise Reduction of Speech Signals using Signal Subspace Methods*. PhD thesis, Technical University of Denmark, 1997. Preprint.

[8] J. M. Lebak and A. W. Bojanczyk. Modifying a Rank-Revealing ULLV Decomposition. Technical report, Cornell Theory Center, June 1994.

[9] F. T. Luk and S. Qiao. A New Matrix Decomposition for Signal Processing. In M. S. Moonen et al., editor, *Linear Algebra for Large Scale and Real-Time Applications*, pages 241–247. Kluwer Academic Publishers, 1993.

[10] Sanzheng Qiao. Computing the ULLV Decomposition. CRL Technical Report 278, Communications Research Laboratory, McMaster University, Hamilton, Ontario L8S 4K1, Canada, January 1994.

[11] G. W. Stewart. An Updating Algorithm for Subspace Tracking. *IEEE Trans. on Signal Processing*, 40(6):1535–1541, June 1992.

[12] G. W. Stewart. Updating a Rank-Revealing ULV Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 14(2):494–499, April 1993.

Index