Documentation

RRQR Factorization

Linux and Windows MEX-Files for MATLAB

March 29, 2007

1 Contents of the distribution file

The distribution file contains the following files:

- rrqrGate.dll: the Windows-MEX-File;
- rrqrGate.mexglx: the Linux-MEX-file;
- rrqr.m: a Matlab-Script that calls the gateway;
- matlabBLAS: a BASH-Script to change the used BLAS-implementation MATLAB uses (only Linux);
- Readme.pdf: this documentation.

2 Usage

2.1 Usage of the gateway

The rrqr.m gateway accepts input and supplies output as described in the following.

- $[Q,R,p,r]$ = rrqr(A), where A is m-by-n, produces an m-by-n upper triangular matrix R and an m-by-m unitary matrix Q so that

  $$A \cdot P = Q \cdot \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

  p is a permutation vector and r is the rank of A.

- $[Q,R,p,r]$ = rrqr(A,'s') produces the "economy size" decomposition. If m$\leq$n, R is m-by-n and Q is m-by-m, otherwise R is n-by-n and Q is m-by-n.

- $[Q,R,p,r]$ = rrqr(A,tol). $\frac{1}{\text{tol}}$ specifies an upper bound on the condition number of R_{11}. If tol == 0 or tol is unset, tol = "machine precision" is chosen as default. tol must be ≥ 0. The tol parameter can be combined with the 's' parameter.

- $[B,R,p,r]$ = rrqr(A,C) returns a matrix B so that $B = C \cdot Q$. The tol parameter is accepted as well.

- $[R,p,r]$ = rrqr(A) is identical to the upper cases but does not compute Q. The tol and 's' parameters are accepted as well.

2.2 Usage of the BASH-Script matlabBLAS

To change the BLAS-library Mathlab uses, set the appropriate path in the script an run it from a BASH-shell. It will set two variables and call Matlab. So make sure, the matlab-command is in your path.

3 Measurements and comparisons of computing times

3.1 Measurements on 2800+ Athlon-MP CPUs

The results shown in the following sections are obtained from 100 runs with random matrices of different size and rank on a SMP workstation with two 2800+ Athlon-MP processors using Gentoo Linux and Matlab 7 R14 SP1. (eqs) means the "economy size" decomposition of the matrices.

3.1.1 Full rank 1000x1000 random matrix

Table 3.1.1 shows results which are obtained from 100 runs with random 1000x1000 matrices of full rank.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	52.3389	51.3627	56.0433	0.6652	0.4425
svd (eqs)	52.2612	50.5984	55.9522	1.0024	1.0048
qr	2.2350	2.1963	2.4476	0.0362	0.0013
qr (eqs)	2.2276	2.1901	2.4856	0.0417	0.0017
rrqr	21.0639	20.6593	23.5019	0.3780	0.1429
rrqr (eqs)	21.0294	20.6306	22.4741	0.3331	0.1109
	goto-BLAS				
svd	53.5756	52.1159	54.9210	0.5922	0.3507
svd (eqs)	53.4010	52.3449	54.9472	0.5426	0.2944
qr	1.5069	1.4873	1.5347	0.0117	$1.3734 \cdot 10^{-4}$
qr (eqs)	1.5087	1.4853	1.5418	0.0101	$1.0221 \cdot 10^{-4}$
rrqr	16.2951	15.6238	17.6923	0.3225	0.1040
rrqr (eqs)	16.0000	15.6168	16.3580	0.1431	0.0205

Table 1: Results using full rank 1000x1000 matrices on AMD Athlon.

3.1.2 1000x1000 random matrix of rank 250

The same test as above is shown in table 2, but now with random 1000x1000 matrices of rank 250.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	37.6314	36.0534	41.2308	1.2026	1.4462
svd (eqs)	37.2076	35.9431	39.7057	1.0490	1.1004
qr	2.2607	2.1855	2.4985	0.0477	0.0023
qr (eqs)	2.2521	2.1740	2.4655	0.0478	0.0023
rrqr	20.1333	18.0342	23.0979	1.0784	1.1629
rrqr (eqs)	20.0007	17.8640	22.6585	1.0106	1.0214
	goto-BLAS				
svd	39.2555	38.2654	40.8706	0.5632	0.3172
svd (eqs)	39.0235	38.0157	40.1970	0.4932	0.2433
qr	1.5186	1.4930	1.5346	0.0099	$9.7738 \cdot 10^{-5}$
qr (eqs)	1.5092	1.4885	1.5227	0.0080	$6.3262 \cdot 10^{-5}$
rrqr	13.4885	13.1555	14.5537	0.2694	0.0726
rrqr (eqs)	13.4839	13.1175	14.8138	0.2665	0.0710

Table 2: Results using 1000x1000 matrices of rank 250 on AMD Athlon.

3.1.3 1000x500 random matrix of rank 100

Table 3 shows the results of 100 runs with 1000x500 matrices of rank 100.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	8.1901	7.6518	8.9248	0.3341	0.1116
svd (eqs)	8.1835	7.6409	8.8966	0.3285	0.1079
qr	0.5196	0.5031	0.5358	0.0068	$4.6710 \cdot 10^{-5}$
qr (eqs)	0.5197	0.5025	0.6055	0.0111	$1.2352 \cdot 10^{-4}$
rrqr	2.9114	2.6033	3.2606	0.1674	0.0280
rrqr (eqs)	4.2021	3.8361	4.6211	0.1605	0.0258
	goto-BLAS				
svd	8.3732	7.9203	9.3079	0.2844	0.0809
svd (eqs)	8.3474	7.9157	8.8350	0.2705	0.0732
qr	0.4390	0.4122	0.4503	0.0112	$1.2435 \cdot 10^{-4}$
qr (eqs)	0.4388	0.4088	0.4512	0.0124	$1.5312 \cdot 10^{-4}$
rrqr	2.6849	2.4712	2.9011	0.1048	0.0110
rrqr (eqs)	3.8557	3.5945	5.0681	0.2039	0.0416

Table 3: Results using 1000x500 matrices of rank 100 on AMD Athlon.

3.1.4 500x1000 random matrix of rank 100

Table 4 shows the results of 100 runs with 500x1000 matrices of rank 100.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	9.1192	8.4224	9.8544	0.3245	0.1053
svd (eqs)	7.3229	6.7067	8.0295	0.3121	0.0974
qr	2.6061	2.5440	2.7472	0.0339	0.0011
qr (eqs)	0.8578	0.8175	0.9045	0.0202	$4.0927 \cdot 10^{-4}$
rrqr	14.7318	14.2380	15.2473	0.2320	0.0538
rrqr (eqs)	14.7913	14.2985	15.4434	0.2451	0.0601
	goto-BLAS				
svd	8.1805	7.6580	9.1493	0.2833	0.0803
svd (eqs)	6.8472	6.4359	7.6134	0.2295	0.0527
qr	1.9238	1.8704	2.0573	0.0352	0.0012
qr (eqs)	0.6387	0.5673	0.6924	0.0297	$8.8177 \cdot 10^{-4}$
rrqr	10.3042	9.7091	10.9949	0.3355	0.1125
rrqr (eqs)	10.4260	9.7216	12.5812	0.5057	0.2557

Table 4: Results using 500x1000 matrices of rank 100 on AMD Athlon.

3.2 Measurements on an Intel Pentium M CPU

The results shown in the following sections are obtained from 100 runs with random matrices of different size and rank on a laptop-computer with an Intel Pentium M processor at 2GHz, using Gentoo Linux and Matlab 7 R14 SP1. (eqs) means the ”economy size” decomposition of the matrices.

3.2.1 Full rank 1000x1000 random matrix

Table 5 shows results which are obtained from 100 runs with random 1000x1000 matrices of full rank.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	27.5705	27.2702	29.2503	0.2057	0.0423
svd (eqs)	27.5100	27.0587	28.1303	0.1682	0.0283
qr	2.1239	2.1110	2.1427	0.0081	$6.5853 \cdot 10^{-5}$
qr (eqs)	2.1226	2.1104	2.1483	0.0080	$6.3333 \cdot 10^{-5}$
rrqr	7.1780	7.1272	7.2395	0.0221	$4.8649 \cdot 10^{-4}$
rrqr0	7.1885	7.1347	7.2347	0.0229	$5.2611 \cdot 10^{-4}$
	goto-BLAS				
svd	28.0039	27.5700	29.2671	0.2393	0.0573
svd (eqs)	28.0287	27.6067	31.3470	0.5083	0.2584
qr	2.0030	1.9706	2.0683	0.0153	$2.3341 \cdot 10^{-4}$
qr (eqs)	2.0010	1.9671	2.0649	0.0163	$2.6670 \cdot 10^{-4}$
rrqr	9.6715	9.4396	12.9509	0.4772	0.2278
rrqr (eqs)	9.5751	9.4473	10.5777	0.1954	0.0382

Table 5: Results using full rank 1000x1000 matrices on Intel Pentium M.

3.2.2 1000x1000 random matrix of rank 250

The same test as above is shown in table 6, but now with random 1000x1000 matrices of rank 250.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	18.7391	17.8493	19.5197	0.4786	0.2291
svd (eqs)	18.7266	17.8389	19.5524	0.4623	0.2137
qr	2.1282	2.1160	2.1551	0.0085	$7.2096 \cdot 10^{-5}$
qr (eqs)	2.1290	2.1155	2.1506	0.0087	$7.5192 \cdot 10^{-5}$
rrqr	6.4038	6.2432	6.5368	0.0738	0.0054
rrqr (eqs)	6.5144	6.3438	6.6524	0.0777	0.0060
	goto-BLAS				
svd	19.8298	19.3869	20.5647	0.2789	0.0778
svd (eqs)	19.8091	19.3039	20.5152	0.2880	0.0830
qr	2.0048	1.9939	2.0213	0.0087	$7.5785 \cdot 10^{-5}$
qr (eqs)	2.0058	1.9931	2.0218	0.0094	$8.8588 \cdot 10^{-5}$
rrqr	8.2805	8.0924	8.4196	0.0765	0.0059
rrqr (eqs)	8.4128	8.2118	8.5632	0.0815	0.0066

Table 6: Results using 1000x1000 matrices of rank 250 on Intel Pentium M.

3.2.3 1000x500 random matrix of rank 100

Table 7 shows the results of 100 runs with 1000x500 matrices of rank 100.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	4.0693	3.7419	4.4355	0.1823	0.0332
svd (eqs)	4.0764	3.7548	4.4431	0.1801	0.0324
qr	0.5007	0.4924	0.5278	0.0079	$6.2069 \cdot 10^{-5}$
qr (eqs)	0.4990	0.4922	0.5138	0.0073	$5.2962 \cdot 10^{-5}$
rrqr	1.1163	1.0112	1.1961	0.0425	0.0018
rrqr (eqs)	1.6277	1.5165	1.7264	0.0545	0.0030
	goto-BLAS				
svd	4.1331	3.8588	4.5410	0.1840	0.0338
svd (eqs)	4.1448	3.8764	4.7940	0.2129	0.0453
qr	0.4729	0.4620	0.5833	0.0176	$3.1121 \cdot 10^{-5}$
qr (eqs)	0.4684	0.4610	0.4904	0.0074	$5.5403 \cdot 10^{-5}$
rrqr	1.6338	1.5454	1.7447	0.0548	0.0030
rrqr (eqs)	2.3476	2.2401	2.4557	0.0608	0.0037

Table 7: Results using 1000x500 matrices of rank 100 on Intel Pentium M.

3.2.4 500x1000 random matrix of rank 100

Table 8 shows the results of 100 runs with 500x1000 matrices of rank 100.

Method	Mean value [s]	Min. [s]	Max. [s]	Std deviation [s]	Variance [s^2]
	Standard MATLAB BLAS implementation				
svd	3.9760	3.6192	4.4062	0.1961	0.0385
svd (eqs)	3.2262	2.8924	3.6846	0.1824	0.0333
qr	1.3992	1.3767	1.4255	0.0114	$1.3059 \cdot 10^{-4}$
qr (eqs)	0.6567	0.6461	0.6810	0.0095	$9.1124 \cdot 10^{-5}$
rrqr	4.2966	4.1147	4.4790	0.1026	0.0105
rrqr (eqs)	4.3042	4.1322	4.4822	0.0993	0.0099
	goto-BLAS				
svd	4.0596	3.8376	4.3230	0.1256	0.0158
svd (eqs)	3.3115	3.0986	3.6301	0.1348	0.0182
qr	1.3840	1.3636	1.4101	0.0110	$1.2111 \cdot 10^{-4}$
qr (eqs)	0.6318	0.6215	0.6428	0.0077	$5.9548 \cdot 10^{-5}$
rrqr	5.6876	5.5074	5.8456	0.0997	0.0099
rrqr (eqs)	5.7421	5.5852	5.9013	0.0875	0.0077

Table 8: Results using 500x1000 matrices of rank 100 on Intel Pentium M.