Rectangular Full Packed Format for LAPACK Algorithms Timings on Several Computers

Fred G. Gustavson[1] and Jerzy Waśniewski[2]

[1] IBM T.J. Watson Research Center
Yorktown Heights NY 10598, USA
email: fg2@us.ibm.com
[2] Informatics & Mathematical Modeling
Technical University of Denmark
DK-2800 Lyngby, Denmark
email: jw@imm.dtu.dk

Abstract. We describe a new data format for storing triangular and symmetric matrices called RFP (Rectangular Full Packed). The standard two dimensional arrays of Fortran and C (also known as full format) that are used to store triangular and symmetric matrices waste nearly half the storage space but provide high performance via the use of level 3 BLAS. Standard packed format arrays fully utilize storage (array space) but provide low performance as there are no level 3 packed BLAS. We combine the good features of packed and full storage using RFP format to obtain high performance using L3 (level 3) BLAS as RFP is full format. Also, RFP format requires exactly the same minimal storage as packed format. Each full and/or packed symmetric/triangular routine becomes a single new RFP routine. We present LAPACK routines for Cholesky factorization, inverse and solution computation in RFP format to illustrate this new work and to describe its performance on the IBM, Itanium, NEC, and SUN platforms. Performance of RFP versus LAPACK full routines for both serial and SMP parallel processing is about the same while using half the storage. Performance is roughly one to a factor of 33 for serial and one to a factor of 100 for SMP parallel times faster than LAPACK packed routines. Existing LAPACK routines and vendor LAPACK routines were used in the serial and the SMP parallel study respectively. In both studies Vendor L3 BLAS were used.

1 Introduction

Recently many new data formats for matrices have been introduced for improving the performance of Dense Linear Algebra (DLA) algorithms. Two ACM TOMS papers [2, 1] and the survey article [5] give an excellent overview. Since then at least two new ones have emerged, [6] and the subject matter of this paper, RFP format.

2 Description of Rectangular Full Packed Format

We describe RFP (Rectangular Full Packed) format. It represents a standard packed array as a full 2D array. By using RFP format the performance of

LAPACK's [3] packed format routines becomes equal to or better than their full array counterparts. RFP format is a variant of Hybrid Full Packed (HFP) format [4]. RFP format is a rearrangement of a Standard full rectangular Array (SA) holding a symmetric / triangular matrix A into a compact full storage Rectangular Array (AR) that uses the minimal storage $NT = n(n+1)/2$. Note also that the transpose of the matrix in array AR also represents A. Therefore, level 3 BLAS can be used on AR or its transpose. In fact, with the equivalent LAPACK algorithm, using array AR or its transpose instead of array SA, gives slightly better performance. Therefore, this offers the possibility to replace all packed or full LAPACK routines with equivalent LAPACK routines that work on array AR or its transpose.

3 Cholesky Factorization using Rectangular Full Packed Format

RFP format is a standard full array of size $NT = n(n+1)/2$ that holds a symmetric / triangular matrix A of order n. It is closely related to HFP format, see [4], which represents A as the concatenation of two standard full arrays whose total size is also NT. A basic simple idea leads to both formats. Let A be an order n symmetric matrix. Break A into a block 2×2 form

$$A = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \tag{1}$$

where A_{11} and A_{22} are symmetric. Clearly, we need only store the lower triangles of A_{11} and A_{22} as well as the full matrix A_{21}. When $n = 2k$ is even, the lower triangle of A_{11} and the upper triangle of A_{22}^T can be concatenated together along their main diagonals into a $(k+1) \times k$ dense matrix. This last operation is the crux of the basic simple idea. The off-diagonal block A_{21} is $k \times k$, and so it can be appended below the $(k+1) \times k$ dense matrix. Thus, the lower triangle of A can be stored as a single $(n+1) \times k$ dense matrix AR. In effect, each block matrix A_{11}, A_{21} and A_{22} is now stored in 'full format'. This means all entries of AR can be accessed with constant row and column strides. So, the full power of LAPACK's block level 3 codes are now available for RFP format which uses the minimum amount of storage. Finally, AR^T which is $k \times (n+1)$ has the same desirable properties. In the right part of Figures 1 and 2 with $n = 7$ and $n = 6$ we have introduced colors and horizontal lines to try to visually delineate triangles T_1, T_2 representing lower, upper triangles of symmetric matrices A_{11}, A_{22}^T respectively and square or near square S_1 representing matrix A_{21}. After each $a_{i,j}$ we have added its position location in the arrays A and AR.

We now describe a 2 by 2 block algorithm (BA) that naturally suggests itself for use on RFP format. A has a block 2×2 form Cholesky factorization

$$LL^T = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} \tag{2}$$

where L_{11} and L_{22} are lower triangular. Equation (2) is the basis of a 2 by 2 BA on RFP. We now describe this by using existing LAPACK routines and level 3

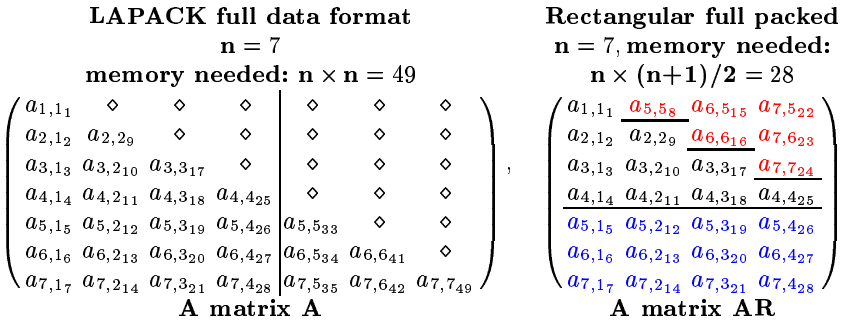

Fig. 1. Rectangular Full Packed format if n is odd

BLAS (see figure 3). The BA with block sizes k and $k+1$ where $k = \lceil n/2 \rceil$ or $k = n/2$ is: see equations (1), (2) and Figures 1, 2, 3.

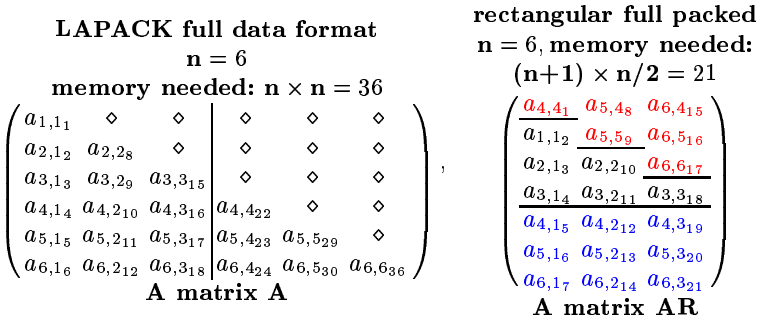

Fig. 2. Rectangular Full Packed format if n is even

This covers RFP format when uplo = 'L', and n is odd (figure 1) and even (figure 2). For uplo = 'U', for n even and odd similar layouts exist. Also, all of these layouts have associated layouts for AR^T.

We now consider performance aspects of using RFP format in the context of using LAPACK routines on triangular matrices stored in RFP format. The above BA should perform about the same as the corresponding full format LAPACK routine. This is because both the BA code and the corresponding LAPACK code are nearly the same and both data formats are full format. Therefore, the BA code should outperform the corresponding LAPACK packed code by about the same margin as does the corresponding LAPACK full code. In [4] performance results for HFP format on the IBM Power 4 Processor is given. Those results are similar to what we obtained for RFP format. Here the gain of full RFP code over packed LAPACK code is about a factor of one to 36 for serial processing and about a factor of one to 97 for SMP parallel processing.

1. factor $L_{11}L_{11}^T = A_{11}$;
   call POTRF('L',k,AR(1,1),n,info)
2. solve $L_{21}L_{11}^T = A_{21}$;
   call TRSM('R','L','T','N',k-1,k, &
   one,AR(1,1),n,AR(k+1,1),n)
3. update $A_{22}^T := A_{22}^T - L_{21}L_{21}^T$;
   call SYRK('U','N',k-1,k,-one, &
   AR(k+1,1),n,one,AR(1,2),n)
4. factor $U_{22}^T U_{22} = A_{22}^T$;
   call POTRF('U',k-1,AR(1,2),n,info)

For n odd, and $k = \lceil n/2 \rceil$

1. factor $L_{11}L_{11}^T = A_{11}$;
   call POTRF('L',k,AR(2,1),n+1,info)
2. solve $L_{21}L_{11}^T = A_{21}$;
   call TRSM('R','L','T','N',k,k,one, &
   AR(2,1),n+1,AR(k+2,1),n+1)
3. update $A_{22}^T := A_{22}^T - L_{21}L_{21}^T$;
   call SYRK('U','N',k,k,-one, &
   AR(k+2,1),n+1,one,AR(1,1),n+1)
4. factor $U_{22}^T U_{22} = A_{22}^T$;
   call POTRF('U',k,AR(1,1),n+1,info)

For n even, and $k = n/2$

Fig. 3. The Cholesky Factorization Algorithm using RFP format

4 A Performance Study using RFP Format

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRF		PPTRF	
	U	L	U	L	U	L	U	L
50	456	520	516	482	460	464	291	294
100	753	813	829	768	612	827	399	369
200	946	979	997	955	933	1150	455	370
400	1208	1231	1158	1183	1081	1244	483	339
500	1173	1227	1138	1186	1121	1340	511	343
800	1316	1318	1189	1269	1256	1310	522	324
1000	1275	1318	1281	1303	1313	1406	530	288
1600	1350	1387	1358	1312	1405	1234	502	223
2000	1264	1367	1403	1323	1360	1491	394	163
4000	1287	1450	1537	1263	1392	1565	300	153

Table 1. Performance in Mflops of Cholesky Factorization on SUN UltraSPARC-III computer

There are 11 tables giving performance results of LAPACK and RFP routines. The LAPACK routines POTRF, PPTRF, POTRI, PPTRI, POTRS and PPTRS are compared with the RFPTRF, RFPTRI and RFPTRS for Cholesky factorization, inverse and solution respectively. In all cases real long precision arithmetic (double precision) is used. Results were obtained on several different computers using everywhere the vendor level 3 and level 2 BLAS.

Due to space limitations, we cannot present all of our timing results. We noticed a few anomalies in the performance runs for POTRS on SUN in Table 3, the PP runs on NEC (Tables 7, 8 and 9) and the PP and PO runs on SUN SMP parallel, Table 11. We have re-run these cases and have obtained the same results. At this time we do *not* have a rational explanation for these anomalies. Finally, our timings do *not* include the cost of sorting any LAPACK data formats to RFP data formats and vice versa.

The tables from 1 to 9 show the performance comparison in Mflops of factorization, inversion and solution on SUN UltraSPARC-III (clock rate: 1200 MHz; L1 cache: 64 kB 4-way data, 32 kB 4-way instruction, and 2 kB Write,

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRI		PPTRI	
	U	L	U	L	U	L	U	L
50	379	379	380	381	330	328	318	338
100	698	696	699	700	698	707	412	446
200	1012	989	1008	997	1052	1030	467	558
400	1290	1223	1229	1263	1229	1212	452	606
500	1290	1276	1238	1330	1285	1276	448	595
800	1446	1445	1330	1356	1343	1325	408	566
1000	1428	1337	1442	1436	1378	1318	404	531
1600	1418	1372	1369	1396	1142	1317	262	450
2000	1387	1333	1370	1536	1400	1366	242	394
4000	1460	1408	1389	1453	1421	1395	201	288

Table 2. Performance in Mflops of Cholesky Inversion on SUN UltraSPARC-III computer

r h s	n	RFP				LAPACK			
		NO TRANS		TRANS		POTRS		PPTRS	
		U	L	U	L	U	L	U	L
100	50	1132	1123	1153	1135	1103	1069	353	353
100	100	1193	1163	1237	1211	1262	1262	478	478
100	200	1477	1478	1500	1490	1280	1235	557	554
100	400	1494	1505	1514	1534	1150	1149	582	582
100	500	1466	1443	1436	1445	1217	1229	560	569
100	800	1503	1505	1535	1469	1151	1096	528	526
100	1000	1553	1524	1499	1576	1089	1125	513	513
160	1600	1595	1564	1603	1577	1155	1121	421	403
200	2000	1600	1636	1610	1615	1105	1087	347	338
400	4000	1666	1668	1696	1665	1080	1084	292	290

Table 3. Performance in Mflops of Cholesky Solution on SUN UltraSPARC-III computer

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRF		PPTRF	
	U	L	U	L	U	L	U	L
50	781	771	784	771	1107	739	495	533
100	1843	1788	1848	1812	1874	1725	879	825
200	3178	2869	2963	3064	2967	2871	1323	1100
400	3931	3709	3756	3823	3870	3740	1121	1236
500	4008	3808	3883	3914	4043	3911	1032	1257
800	4198	4097	4145	4126	3900	4009	612	1127
1000	4115	4038	4015	3649	3769	3983	305	697
1600	3851	3652	3967	3971	3640	3987	147	437
2000	3899	3716	3660	3660	3865	3835	108	358
4000	3966	3791	3927	4011	3869	4052	119	398

Table 4. Performance in Mflops of Cholesky Factorization on ia64 Itanium computer

2 kB Prefetch; L2 cache: 8 MB; TLB: 1040 entries), ia64 Itanium (CPU: Intel Itanium2: 1.3 GHz, cache: 3 MB on-chip L3 cache), and NEC SX-6 computer (8 CPU's, per CPU peak : 8 Gflops, per node peak : 64 Gflops, vector register length: 256). The tables 10 and 11 show the SMP parallelism of these subroutines on the IBM Power4 (clock rate: 1300 MHz; two CPUs per chip; L1 cache: 128 KB (64 KB per CPU) instruction, 64 KB 2-way (32 KB per CPU) data; L2 cache: 1.5 MB 8-way shared between the two CPUs; L3 cache: 32 MB 8-way shared (off-chip); TLB: 1024 entries) and SUN UltraSPARC-IV (the same hardware parameters as SUN UltraSPARC-III except the clock rate: 1350 MHz) computers respectively. They compare SMP times of RFPTRF, POTRF and PPTRF. The tables 10 and 11 also show the times of the four operations (POTRF, TRSM, SYRK and again POTRF) inside the new algorithm RFPTRF.

5 Summary and Conclusions

This paper describes RFP format as a standard minimal full format for representing both symmetric and triangular matrices. Hence, these matrix layouts are a replacement for both the standard formats of DLA, namely full and packed storage. These new layouts possess three good features: they are supported by level 3 BLAS and LAPACK full format routines, and they require minimal storage.

6 Acknowledgments

The results in this paper were obtained on five computers, an IBM, two SUN, Itanium and NEC computers. The IBM machine belongs to the Center for Scientific Computing at Aarhus, the SUN machines to the Danish Technical University, and the Itanium and NEC machines to the Danish Meteorological Institute. We would like to thank Bernd Dammann for consulting on the SUN systems; Niels Carl W. Hansen for consulting on the IBM system; and Bjarne Stig Andersen for obtaining the results on the Itanium and NEC computers.

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRI		PPTRI	
	u	l	u	l	u	l	u	l
50	633	659	648	640	777	870	508	460
100	1252	1323	1300	1272	1573	1760	815	810
200	2305	2442	2431	2314	2357	2639	1118	1211
400	3084	3199	3188	3094	3152	3445	1234	1363
500	3204	3316	3329	3218	3400	3611	1239	1382
800	3617	3741	3720	3640	3468	3786	1182	1268
1000	3611	3716	3637	3590	3456	3790	767	946
1600	3721	3802	3795	3714	3589	3713	500	609
2000	3784	3812	3745	3704	3636	3798	473	596
4000	3822	3762	3956	3851	3760	3750	467	614

Table 5. Performance in Mflops of Cholesky Inversion on ia64 Itanium computer

rhs	n	RFP				LAPACK			
		NO TRANS		TRANS		POTRS		PPTRS	
		u	l	u	l	u	l	u	l
100	50	2409	2412	2414	2422	3044	3018	725	714
100	100	3305	3301	3303	3303	3889	3855	1126	1109
100	200	4149	4154	4127	4146	4143	4127	1526	1512
100	400	4398	4403	4416	4444	4469	4451	1097	1088
100	500	4313	4155	4374	4394	4203	4093	1054	1045
100	800	3979	3919	4040	4051	3969	4011	692	720
100	1000	3716	3608	3498	3477	3630	3645	376	372
160	1600	3892	3874	4020	3994	4001	4011	188	182
200	2000	4052	4073	4040	4020	4231	4203	119	119
400	4000	4245	4225	4275	4287	4330	4320	115	144

Table 6. Performance in Mflops of Cholesky Solution on ia64 Itanium computer

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRF		PPTRF	
	u	l	u	l	u	l	u	l
50	206	200	225	225	365	353	57	238
100	721	728	789	788	1055	989	120	591
200	2028	2025	2005	2015	1380	1639	246	1250
400	3868	3915	3078	3073	1763	3311	479	1975
500	4483	4470	4636	4636	4103	4241	585	2149
800	5154	5168	4331	4261	3253	4469	870	2399
1000	5666	5654	5725	5703	5144	5689	1035	2474
1600	6224	6145	5644	5272	5375	5895	1441	2572
2000	6762	6788	6642	6610	6088	6732	1654	2598
4000	7321	7325	7236	7125	6994	7311	2339	2641

Table 7. Performance in Mflops of Cholesky Factorization on SX-6 NEC computer with Vector Option

n	RFP				LAPACK			
	NO TRANS		TRANS		POTRI		PPTRI	
	u	l	u	l	u	l	u	l
50	152	152	150	152	148	145	91	61
100	430	432	428	432	313	310	194	126
200	950	956	940	941	636	627	404	249
400	1850	1852	1804	1806	1734	1624	722	470
500	2227	2228	2174	2181	2180	2029	856	572
800	3775	3775	3668	3686	3405	3052	1186	842
1000	4346	4346	4254	4263	4273	3638	1342	985
1600	5313	5294	5137	5308	5438	4511	1690	1361
2000	6006	6006	5930	5931	5997	4832	1854	1536
4000	6953	6953	6836	6888	7041	4814	1921	2122

Table 8. Performance in Mflops of Cholesky Inversion on SX-6 NEC computer with Vector Option

r h s	n	RFP				LAPACK			
		NO TRANS		TRANS		POTRS		PPTRS	
		U	L	U	L	U	L	U	L
100	50	873	870	889	886	1933	1941	88	88
100	100	2173	2171	2200	2189	3216	3236	181	179
100	200	4236	4230	4253	4245	4165	4166	352	347
100	400	5431	5431	5410	5408	5302	5303	648	644
100	500	5563	5562	5568	5567	5629	5632	783	779
100	800	6407	6407	6240	6240	5569	5593	1132	1128
100	1000	6578	6578	6559	6558	6554	6566	1325	1320
160	1600	6781	6805	6430	6430	6799	6809	1732	1727
200	2000	7568	7569	7519	7519	7406	7407	1920	1914
400	4000	7858	7858	7761	7761	7626	7627	2414	2410

Table 9. Performance in Mflops of Cholesky Solution on SX-6 NEC computer with Vector Option

n	n proc	Mflp RFPTRF		Times in RFPTRF				Times LAPACK	
				POTRF	TRSM	SYRK	POTRF	POTRF	PPTRF
1000	1	2695	0.12	0.02	0.05	0.04	0.02	0.12	0.94
	5	7570	0.04	0.01	0.02	0.01	0.01	0.03	0.32
	10	10699	0.03	0.01	0.01	0.01	0.00	0.02	0.16
	15	18354	0.02	0.00	0.01	0.00	0.00	0.01	0.11
2000	1	2618	1.02	0.13	0.38	0.38	0.13	0.97	8.74
	5	10127	0.26	0.04	0.10	0.09	0.04	0.24	3.42
	10	17579	0.15	0.02	0.06	0.05	0.03	0.12	1.65
	15	23798	0.11	0.02	0.04	0.04	0.01	0.13	1.11
3000	1	2577	3.49	0.45	1.33	1.28	0.44	3.40	30.42
	5	11369	0.79	0.11	0.28	0.30	0.11	0.71	11.76
	10	19706	0.46	0.06	0.19	0.16	0.05	0.38	6.16
	15	29280	0.31	0.05	0.12	0.10	0.04	0.26	4.28
4000	1	2664	8.01	1.01	2.90	3.09	1.01	7.55	75.72
	5	11221	1.90	0.26	0.68	0.72	0.24	1.65	25.73
	10	21275	1.00	0.13	0.39	0.36	0.12	0.86	13.95
	15	31024	0.69	0.09	0.28	0.24	0.08	0.59	10.46
5000	1	2551	16.34	2.04	6.16	6.10	2.04	15.79	154.74
	5	11372	3.66	0.45	1.37	1.44	0.40	3.27	47.76
	10	22326	1.87	0.25	0.78	0.62	0.22	1.73	28.13
	15	32265	1.29	0.17	0.53	0.45	0.14	1.16	20.95

Table 10. Performance Times and MFLOPS of Cholesky Factorization on an IBM Power 4 computer using SMP parallelism on 1, 5, 10 and 15 processors. Here vendor codes for Level 2 and 3 BLAS and POTRF are used, ESSL library version 3.3. PPTRF is LAPACK code. UPLO = 'L'.

n	n	Mflp	Times					
	pr	RFPTRF	in RFPTRF				LAPACK	
	oc		POTRF	TRSM	SYRK	POTRF	POTRF	PPTRF
1000	1	1587 0.21	0.03	0.09	0.07	0.03	0.19	1.06
	5	4762 0.07	0.02	0.02	0.02	0.02	0.07	1.13
	10	5557 0.06	0.01	0.01	0.02	0.02	0.06	1.12
	15	5557 0.06	0.02	0.01	0.01	0.02	0.06	1.11
2000	1	1668 1.58	0.22	0.63	0.52	0.22	1.45	11.20
	5	6667 0.40	0.07	0.13	0.13	0.07	0.38	11.95
	10	8602 0.31	0.06	0.07	0.11	0.07	0.25	11.24
	15	9524 0.28	0.06	0.06	0.08	0.08	0.23	11.66
3000	1	1819 4.95	0.62	1.98	1.72	0.63	4.86	45.48
	5	6872 1.31	0.20	0.42	0.48	0.20	1.38	55.77
	10	12162 0.74	0.14	0.22	0.21	0.16	0.76	46.99
	15	12676 0.71	0.14	0.16	0.30	0.16	0.61	45.71
4000	1	1823 11.70	1.52	4.62	4.01	1.55	11.86	112.52
	5	7960 2.68	0.40	0.94	0.92	0.42	2.74	112.77
	10	14035 1.52	0.26	0.47	0.49	0.30	1.61	112.53
	15	17067 1.25	0.24	0.37	0.35	0.29	1.29	111.67
5000	1	1843 22.61	2.92	8.76	8.00	2.93	23.60	218.94
	5	8139 5.12	0.77	1.81	1.80	0.74	5.45	221.58
	10	14318 2.91	0.50	0.97	0.93	0.51	3.11	214.54
	15	17960 2.32	0.45	0.72	0.68	0.47	2.40	225.08

Table 11. Performance in Times and Mflops of Cholesky Factorization on SUN UltraSPARC-IV computer with a different number of Processors, testing the SMP Parallelism. PPTRF does not show any SMP parallelism. UPLO = 'L'.

References

1. B.S. Andersen, J. Gunnels, F.G. Gustavson, J.K. Reid, and J. Waśniewski. A fully portable high performance minimal storage hybrid format Cholesky algorithm. *TOMS*, Vol. 31, pp. 201-227, 2005.
2. B. S. Andersen, F.G. Gustavson, J. Waśniewski. A recursive formulation of Cholesky factorization of a matrix in packed storage. *TOMS*, Vol. 27, No. 2, June. 2001, pp. 214-244.
3. E. Anderson, et. al.. LAPACK Users' Guide Release 3.0, SIAM, Philadelphia, 1999, (http://www.netlib.org/lapack/).
4. J. A. Gunnels, F. G. Gustavson. A New Array Format for Symmetric and Triangular Matrices. PARA 2004, J. J. Dongarra, K. Madsen, J. Waśniewski, eds., Lecture Notes in Computer Science 3732. Springer-Verlag, pp. 247-255, 2004.
5. E. Elmroth, F. G. Gustavson, B. Kagstrom, and I. Jonsson. Recursive Blocked Algorithms and Hybrid Data Structures for Dense Matrix Library Software. *SIAM Review*, Vol. 46, No. 1, Mar. 2004, pp. 3-45.
6. J. R. Herrero. A Framework for Efficient Execution of Matrix Computations, *PhD thesis, Universitat Politècnica de Catalunya*, May 2006