# A Fast Minimal Storage Factorization of Symmetric and Hermitian Matrices

**Jerzy Waśniewski**

**Emeritus Senior Researcher**

**Informatics & Mathematical Modeling (IMM)**

**Technical University of Denmark (DTU)**

**and**

**Emeritus Principal Computer Scientist**

**Danish IT Centre for Research & Education (UNI•C)**

**Email: jw@imm.dtu.dk**

**EU COST Meeting, Lisbon, Portugal, October 19–20, 2009**

**In collaboration with:**

**(•) Fred Gustavson**
**IBM Watson Research Center, USA**
**(•) Jack J. Dongarra**
**U. of Tennessee, ORNL and U. of Manchester, USA & UK**
**(•) Julien Langou**
**University of Colorado Denver, USA**
**(•) Bjarne Sig Andersen**
**Danish Meteorological Institute, Denmark**
**(•) John K. Reid**
**Rutherford Appleton Laboratory, UK**
**(•) Plamen Yalamov, Minka & Alexander Karaivanov,**
**and Ivan Lirkov**
**University of Rousse, Bulgaria**

## Present Standards of DLA

- **LAPACK with Level 1, 2, and 3 BLAS (ATLAS)**
  - **Usually for single processors**
  - **Shared memory versions**
  - **Open MP**
  - **Partly for distributed memory versions**
- **ScaLAPACK with PBLAS, BLACS and MPI**
  - **Distributed memory computers**
  - **Network computing**
- **Harwell, NAG, IMSL, NetLib and others**
- **Computer vendor Libraries, i.e. ESSL and SunPerf**

## LAPACK Standard for Symmetric Matrices

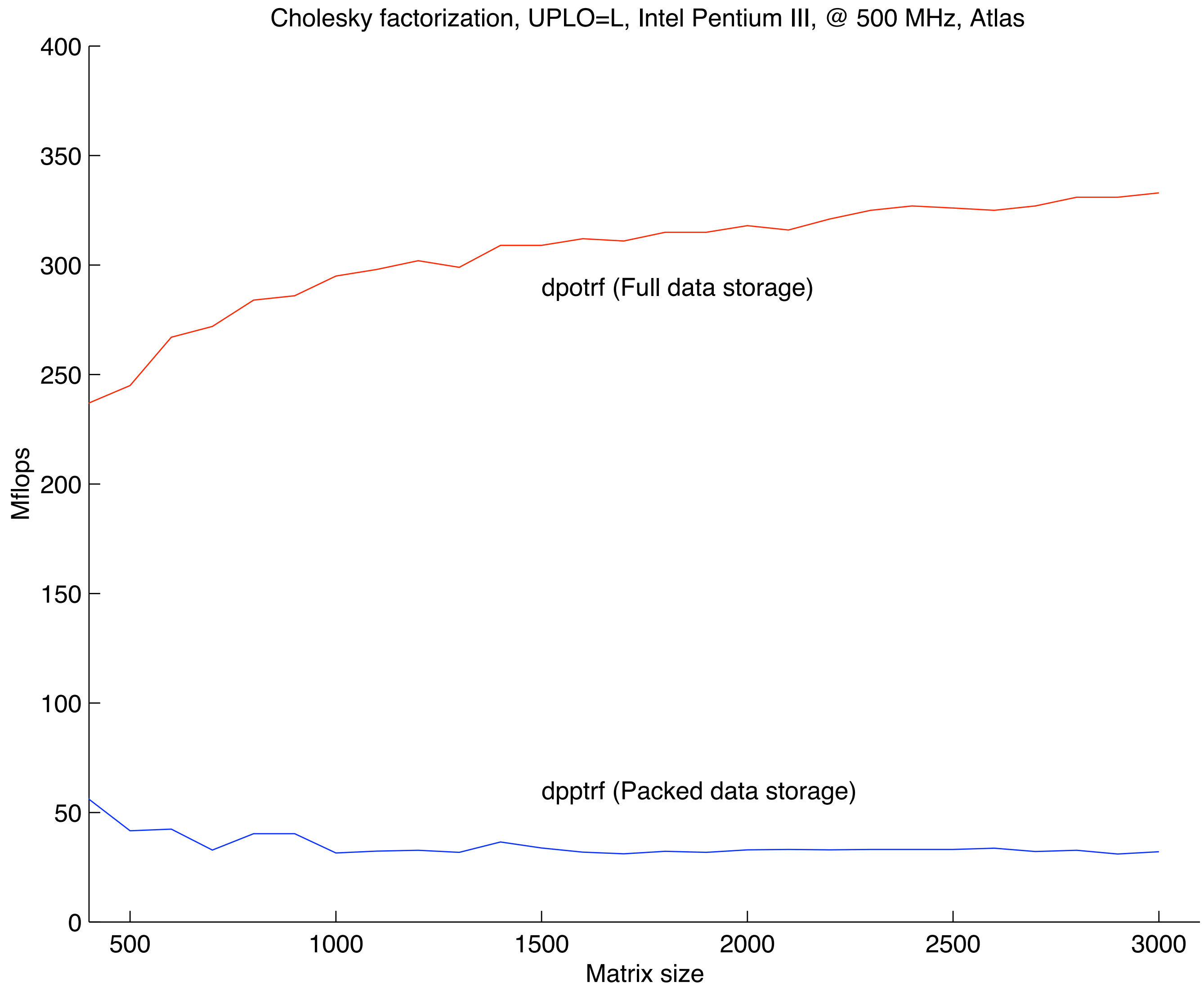
**As we know, LAPACK has two Data Formats for symmetric matrices, the full and packed. The full format works fast but requires the full matrix memory, $n^2$. The packed format requires only $n(n+1)/2$ memory but works slow, many times slower than the full format ($n$ is a size of the matrix $A$).**

**The full format uses level 3 BLAS while the packed level 2. There are no level 3 BLAS for the packed format.**

**LAPACK has two kinds of factorization algorithms for symmetric matrices.**

1. **Cholesky algorithm for the symmetric positive definite and**
2. $LDL^T$ **with Bunch-Kaufman pivoting for the symmetric indefinite; diagonal pivoting method.**

# Cholesky, a packed and full data storage



## The New Packed can be even faster than the Full

**Our experience already shows that the packed version for the symmetric matrices can be ran by level 3 BLAS.**

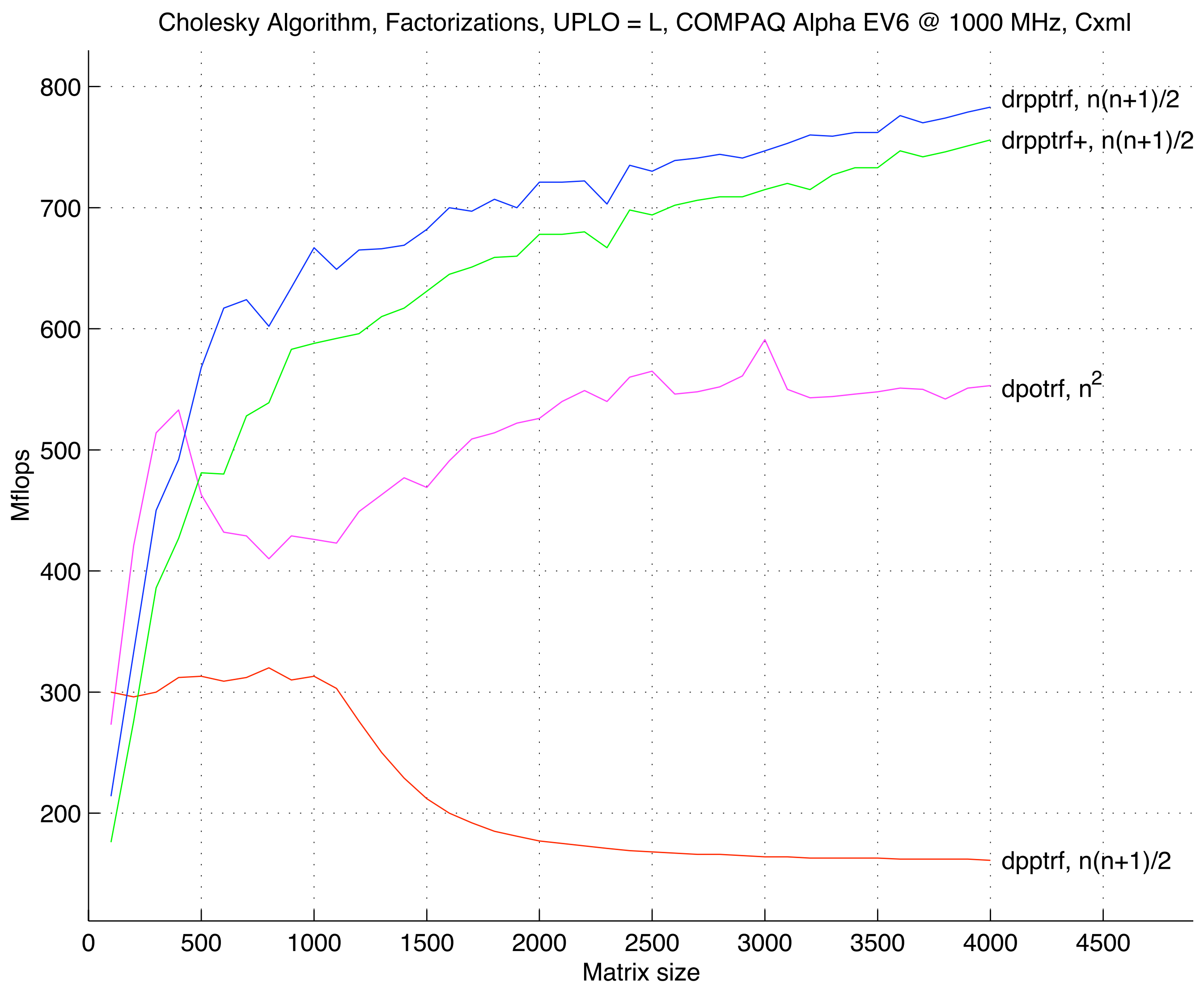
**The LAPACK Working Note Number 199 (paper already accepted for publication in TOMS of ACM) describes RFPF data format which can replace both the full and packed formats of symmetric positive definite matrices.**

**The RFPF format of symmetric positive definite matrices has already been included in LAPACK version 3.2.**

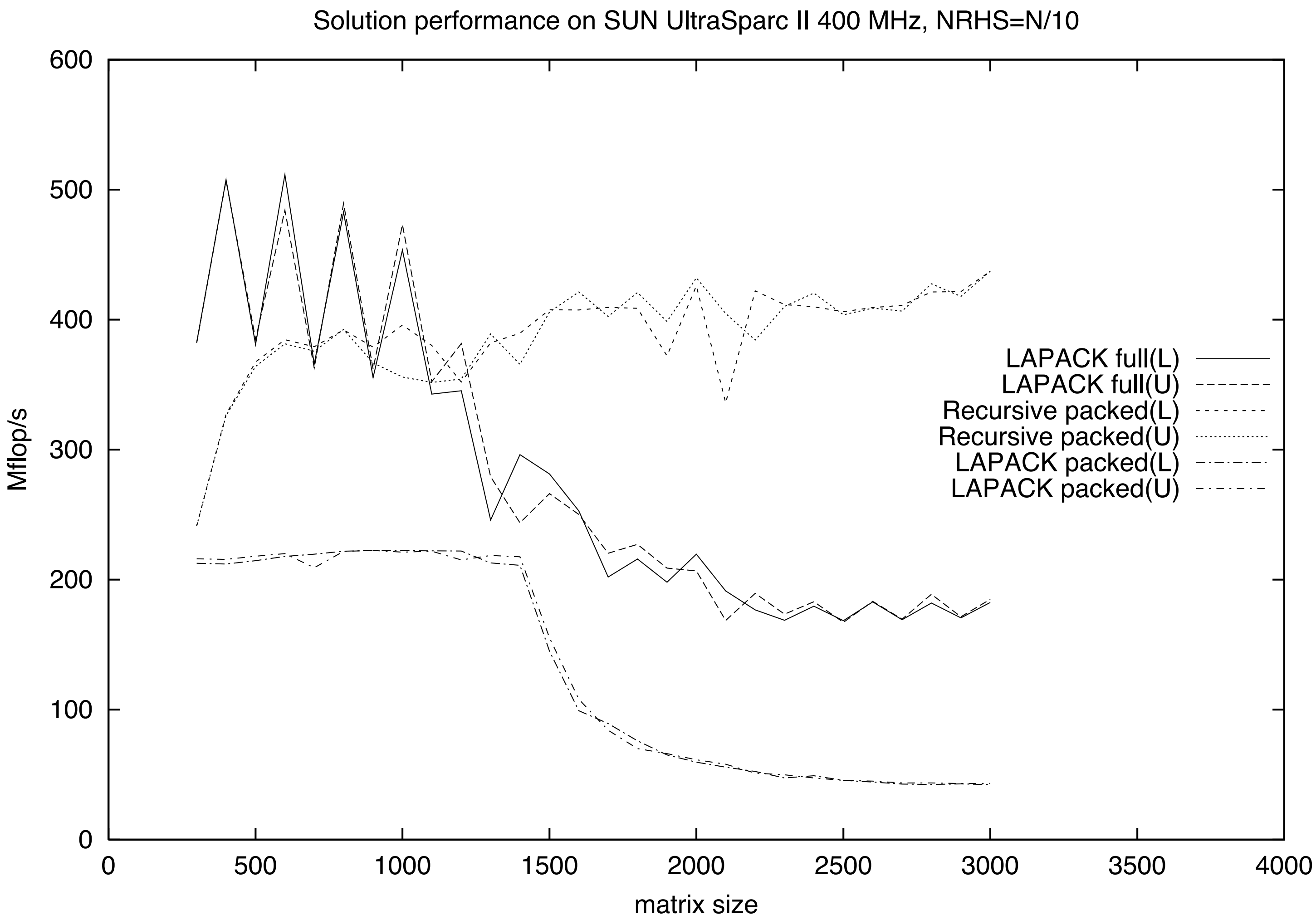
## Recursion

- F.G. Gustavson. "Recursion Leads to Automatic Variable Blocking for Dense Linear-Algebra Algorithms". IBM Journal of Research and Development, 41(6), November 1997.
- Bjarne S. Andersen, Fred G. Gustavson, and Jerzy Waśniewski. "A recursive formulation of Cholesky factorization of a matrix in packed storage". ACM Transactions on Mathematical Software, 27(2):214–244, June 2001

# Cholesky Factorization



# Cholesky, Recursive Packed Storage



## Hybrid Format Cholesky Algorithm

**B.S. Andersen, J.A. Gunnels, F. Gustavson, J.K. Reid, and J. Waśniewski. "A Fully Portable High Performance Minimal Storage Hybrid Format Cholesky Algorithm". ACM Trans. of Math. Software, 31 (2005), 201-227.**

**Mflops, Cholesky factorizations, lower case, $nb$ = 100, IBM Power4.**

| $n$ | 40 | 64 | 100 | 160 | 250 | 400 | 640 | 1000 | 1600 | 2500 | 4000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p. lapack | 747 | 951 | 1043 | 1024 | 1059 | 1101 | 1037 | 709 | 638 | 621 | 635 |
| v. p. lapack | 1750 | 2359 | 2658 | 2346 | 3107 | 3560 | 3773 | 3870 | 3969 | 3815 | 3836 |
| f. lapack | 440 | 722 | 1390 | 2119 | 2562 | 3242 | 3495 | 3797 | 3901 | 3787 | 4010 |
| v. f. lapack | 1492 | 2165 | 2486 | 3194 | 3454 | 3677 | 3832 | 3921 | 4162 | 4037 | 4327 |
| p. recursive+ | 170 | 379 | 593 | 1024 | 1586 | 2077 | 2621 | 3030 | 3434 | 3555 | 3943 |
| p. recursive | 181 | 406 | 618 | 1060 | 1652 | 2133 | 2700 | 3111 | 3523 | 3604 | 3980 |
| p. hybrid+ | 878 | 1488 | 2085 | 2721 | 3211 | 3754 | 3974 | 4112 | 4188 | 4200 | 4275 |
| p. hybrid | 1006 | 1717 | 2334 | 2977 | 3441 | 3938 | 4149 | 4279 | 4266 | 4269 | 4309 |

**Mflops, Cholesky Factorizations, lo. case, $nb$ = 200, SUN UltraSPARC III.**

| $n$ | 40 | 64 | 100 | 160 | 250 | 400 | 640 | 1000 | 1600 | 2500 | 4000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p. lapack | 183 | 247 | 296 | 312 | 321 | 325 | 328 | 331 | 315 | 200 | 169 |
| f. lapack | 299 | 436 | 637 | 842 | 933 | 1086 | 864 | 1173 | 1203 | 1169 | 1236 |
| p. recursive+ | 95 | 202 | 275 | 426 | 550 | 727 | 913 | 1010 | 1093 | 1118 | 1215 |
| p. recursive | 102 | 219 | 290 | 454 | 581 | 760 | 945 | 1043 | 1146 | 1162 | 1249 |
| p. hybrid+ | 390 | 557 | 708 | 782 | 867 | 965 | 1080 | 1180 | 1201 | 1206 | 1254 |
| p. hybrid | 529 | 778 | 959 | 973 | 1003 | 1077 | 1164 | 1267 | 1277 | 1257 | 1304 |

**Mflops, Cholesky Factorizations, lower case, $nb$ = 200, HP Itanium 2.**

| $n$ | 40 | 64 | 100 | 160 | 250 | 400 | 640 | 1000 | 1600 | 2500 | 4000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p. lapack | 218 | 328 | 463 | 644 | 825 | 952 | 1039 | 660 | 609 | 595 | 590 |
| f. lapack | 376 | 581 | 838 | 1307 | 1728 | 2189 | 2546 | 2777 | 2904 | 3028 | 3141 |
| p. recursive+ | 110 | 232 | 378 | 698 | 1079 | 1608 | 2170 | 2531 | 2758 | 2861 | 3013 |
| p. recursive | 121 | 256 | 411 | 742 | 1147 | 1675 | 2305 | 2666 | 2874 | 2942 | 3056 |
| p. hybrid+ | 657 | 1017 | 1689 | 1731 | 1279 | 1384 | 1552 | 1829 | 2089 | 2216 | 2402 |
| p. hybrid | 763 | 1139 | 1849 | 1815 | 1390 | 1504 | 1661 | 1910 | 2167 | 2274 | 2438 |

## Rectangular Full Packed Cholesky

**Fred G. Gustavson, Jerzy Waśniewski, Julien Langou and Jack J. Dongarra: LAPACK Working Note Nr 199 "Rectangular Full Packed Format for Cholesky's Algorithm: Factorization, Solution and Inversion", UT-CS-08-614, April 28, 2008. (accepted for TOMS of ACM). Already included in LAPACK version 3.2.**

## Cholesky: $A = LL^T$

$$A = \begin{pmatrix} A_{11} & \\ A_{21} & A_{22} \end{pmatrix} \quad \text{and} \quad L = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix}$$

**Do**

- **if** $n > 1$ **then**
  - $L_{11}$ := **Cholesky of** $A_{11}$
  - $L_{21}L_{11}^T = A_{21} \rightarrow$ **TRSM**
  - $\hat{A}_{22} := A_{22} - L_{21}L_{21}^T \rightarrow$ **SYRK**
  - $L_{22}$ := **Cholesky of** $\hat{A}_{22}$
- **otherwise**
  - $L := \sqrt{A_{11}}$

**End**

# Rectangular Full Packed Data Format, $n$ is odd

**n** $= 7,\qquad$ **memory needed** $=$ **n** $\times$ **n** $= 49$

$$A = \left(\begin{array}{cccc|ccc}
a_{1,1_{1}} & \diamond & \diamond & \diamond & \diamond & \diamond & \diamond \\
a_{2,1_{2}} & a_{2,2_{9}} & \diamond & \diamond & \diamond & \diamond & \diamond \\
a_{3,1_{3}} & a_{3,2_{10}} & a_{3,3_{17}} & \diamond & \diamond & \diamond & \diamond \\
a_{4,1_{4}} & a_{4,2_{11}} & a_{4,3_{18}} & a_{4,4_{25}} & \diamond & \diamond & \diamond \\ \cline{1-4}
a_{5,1_{5}} & a_{5,2_{12}} & a_{5,3_{19}} & a_{5,4_{26}} & a_{5,5_{33}} & \diamond & \diamond \\
a_{6,1_{6}} & a_{6,2_{13}} & a_{6,3_{20}} & a_{6,4_{27}} & a_{6,5_{34}} & a_{6,6_{41}} & \diamond \\
a_{7,1_{7}} & a_{7,2_{14}} & a_{7,3_{21}} & a_{7,4_{28}} & a_{7,5_{35}} & a_{7,6_{42}} & a_{7,7_{49}}
\end{array}\right)$$

**LAPACK full data format**

**n** $= 7,\qquad$ **memory needed** $=$ **n** $\times$ **(n+1)/2** $= 28$

$$A^{rfp} = \left(\begin{array}{cccc}
a_{1,1_{1}} & \color{red}{a_{5,5_{8}}} & \color{red}{a_{6,5_{15}}} & \color{red}{a_{7,5_{22}}} \\ \cline{2-2}
a_{2,1_{2}} & a_{2,2_{9}} & \color{red}{a_{6,6_{16}}} & \color{red}{a_{7,6_{23}}} \\ \cline{3-3}
a_{3,1_{3}} & a_{3,2_{10}} & a_{3,3_{17}} & \color{red}{a_{7,7_{24}}} \\ \cline{4-4}
a_{4,1_{4}} & a_{4,2_{11}} & a_{4,3_{18}} & a_{4,4_{25}} \\ \hline
\color{blue}{a_{5,1_{5}}} & \color{blue}{a_{5,2_{12}}} & \color{blue}{a_{5,3_{19}}} & \color{blue}{a_{5,4_{26}}} \\
\color{blue}{a_{6,1_{6}}} & \color{blue}{a_{6,2_{13}}} & \color{blue}{a_{6,3_{20}}} & \color{blue}{a_{6,4_{27}}} \\
\color{blue}{a_{7,1_{7}}} & \color{blue}{a_{7,2_{14}}} & \color{blue}{a_{7,3_{21}}} & \color{blue}{a_{7,4_{28}}}
\end{array}\right)$$

**Rectangular full packed data format**

# Rectangular Full Packed Data Format, $n$ is even

$$\textbf{n} = 6, \qquad \textbf{memory needed} = \textbf{n} \times \textbf{n} = 36$$

$$A = \left(\begin{array}{ccc|ccc}
a_{1,1_1} & \diamond & \diamond & \diamond & \diamond & \diamond \\
a_{2,1_2} & a_{2,2_8} & \diamond & \diamond & \diamond & \diamond \\
a_{3,1_3} & a_{3,2_9} & a_{3,3_{15}} & \diamond & \diamond & \diamond \\
\hline
a_{4,1_4} & a_{4,2_{10}} & a_{4,3_{16}} & a_{4,4_{22}} & \diamond & \diamond \\
a_{5,1_5} & a_{5,2_{11}} & a_{5,3_{17}} & a_{5,4_{23}} & a_{5,5_{29}} & \diamond \\
a_{6,1_6} & a_{6,2_{12}} & a_{6,3_{18}} & a_{6,4_{24}} & a_{6,5_{30}} & a_{6,6_{36}}
\end{array}\right)$$

**LAPACK full data format**

$$\textbf{n} = 6, \qquad \textbf{memory needed} = \textbf{(n+1)} \times \textbf{n/2} = 21$$

$$A^{rfp} = \left(\begin{array}{ccc}
\color{red}{a_{4,4_1}} & \color{red}{a_{5,4_8}} & \color{red}{a_{6,4_{15}}} \\
a_{1,1_2} & \color{red}{a_{5,5_9}} & \color{red}{a_{6,5_{16}}} \\
a_{2,1_3} & a_{2,2_{10}} & \color{red}{a_{6,6_{17}}} \\
a_{3,1_4} & a_{3,2_{11}} & a_{3,3_{18}} \\
\hline
\color{blue}{a_{4,1_5}} & \color{blue}{a_{4,2_{12}}} & \color{blue}{a_{4,3_{19}}} \\
\color{blue}{a_{5,1_6}} & \color{blue}{a_{5,2_{13}}} & \color{blue}{a_{5,3_{20}}} \\
\color{blue}{a_{6,1_7}} & \color{blue}{a_{6,2_{14}}} & \color{blue}{a_{6,3_{21}}}
\end{array}\right)$$

**Rectangular full packed data format**

# Symmetric/Hermitian matrices

**n** $= 7,$ **memory needed** $=$ **n** $\times$ **(n+1)/2** $= 28$

$$A^{rfp} = \begin{pmatrix} a_{1,1_1} & a_{5,5_8} & a_{6,5_{15}} & a_{7,5_{22}} \\ a_{2,1_2} & a_{2,2_9} & a_{6,6_{16}} & a_{7,6_{23}} \\ a_{3,1_3} & a_{3,2_{10}} & a_{3,3_{17}} & a_{7,7_{24}} \\ a_{4,1_4} & a_{4,2_{11}} & a_{4,3_{18}} & a_{4,4_{25}} \\ \hline a_{5,1_5} & a_{5,2_{12}} & a_{5,3_{19}} & a_{5,4_{26}} \\ a_{6,1_6} & a_{6,2_{13}} & a_{6,3_{20}} & a_{6,4_{27}} \\ a_{7,1_7} & a_{7,2_{14}} & a_{7,3_{21}} & a_{7,4_{28}} \end{pmatrix}$$

**Rectangular full packed data format**

$$A^{rfp} = \begin{pmatrix} A_{11} & A_{22}^T \\ & A_{21} \end{pmatrix}$$

$$L^{rfp} = \begin{pmatrix} L_{11} & L_{22}^T \\ & L_{21} \end{pmatrix} ?$$

- $L_{11} :=$ **POTRF of** $(A_{11})$
- $L_{21} :=$ **TRSM of** $(L_{21} L_{11}^T = A_{21})$
- $\hat{L}_{22} :=$ **SYRK of** $(A_{22} - L_{21} L_{21}^T)$
- $L_{22} :=$ **POTRF of** $(\hat{L}_{22})$

**SUN UltraSPARC IV dual-core CPUs (1350 MHz/ 8 MB/core L2-cache)**
**Sun BLAS Lib., Real long prec., Cholesky Factorization, Mflops**

| | New | | | | lapack | | | |
|---|---|---|---|---|---|---|---|---|
| | rfp | | hpp | | po | | pp | |
| n | u | l | u | l | u | l | u | l |
| 50 | 819 | 909 | 1317 | 1321 | 910 | 800 | 431 | 617 |
| 100 | 1419 | 1533 | 1954 | 1953 | 1268 | 1389 | 591 | 806 |
| 200 | 1757 | 1807 | 1996 | 1997 | 1712 | 1925 | 708 | 377 |
| 400 | 2177 | 2200 | 2289 | 2316 | 2219 | 2314 | 793 | 257 |
| 500 | 2170 | 2206 | 2377 | 2415 | 2354 | 2350 | 812 | 251 |
| 800 | 2446 | 2409 | 2574 | 2635 | 2549 | 2416 | 800 | 242 |
| 1000 | 2512 | 2459 | 2644 | 2711 | 2617 | 2498 | 702 | 228 |
| 1600 | 2584 | 2531 | 2759 | 2849 | 2753 | 1950 | 627 | 217 |
| 2000 | 2692 | 2624 | 2804 | 2899 | 2795 | 2638 | 629 | 216 |
| 4000 | 2337 | 2753 | 2842 | 2944 | 2834 | 2700 | 509 | 97 |

**SUN UltraSPARC IV dual-core CPUs (1350 MHz/ 8 MB/core L2-cache)**

**Sun BLAS Lib., Real long prec., Cholesky Factorization, Mflops**

| n | rfp u | rfp l | hpp u | hpp l | rpp u | rpp l | pp u | pp l |
|---|---|---|---|---|---|---|---|---|
| 50 | 830 | 921 | 1319 | 1321 | 256 | 285 | 432 | 619 |
| 100 | 1416 | 1531 | 1950 | 1952 | 535 | 601 | 591 | 806 |
| 200 | 1760 | 1815 | 2005 | 2006 | 968 | 1060 | 708 | 376 |
| 400 | 2182 | 2213 | 2294 | 2321 | 1527 | 1632 | 793 | 257 |
| 500 | 2168 | 2205 | 2386 | 2422 | 1827 | 1845 | 812 | 251 |
| 800 | 2451 | 2408 | 2579 | 2640 | 2076 | 2142 | 784 | 237 |
| 1000 | 2513 | 2472 | 2644 | 2717 | 2361 | 2313 | 702 | 227 |
| 1600 | 2589 | 2538 | 2765 | 2853 | 2489 | 2497 | 627 | 216 |
| 2000 | 2654 | 2633 | 2810 | 2903 | 2454 | 2417 | 617 | 216 |
| 4000 | 2448 | 2762 | 2845 | 2942 | 2847 | 2814 | 492 | 94 |

**SUN UltraSPARC IV dual-core CPUs (1350 MHz/ 8 MB/core L2-cache)**
**Sun BLAS Lib., Complex long prec., Cholesky Factorization, Mflops**

| | rfp | | | | lapack | | | |
|---|---|---|---|---|---|---|---|---|
| | no trans | | trans | | po | | pp | |
| n | u | l | u | l | u | l | u | l |
| 50 | 1450 | 1550 | 1642 | 1436 | 1313 | 1310 | 893 | 1347 |
| 100 | 2046 | 2001 | 2072 | 1858 | 1564 | 1944 | 1292 | 1368 |
| 200 | 2343 | 2285 | 2346 | 2220 | 2123 | 2389 | 1657 | 543 |
| 400 | 2688 | 2607 | 2524 | 2602 | 2570 | 2653 | 2006 | 482 |
| 500 | 2794 | 2708 | 2636 | 2729 | 2707 | 2750 | 2068 | 478 |
| 800 | 2912 | 2794 | 2750 | 2862 | 2895 | 2246 | 1430 | 443 |
| 1000 | 2951 | 2834 | 2824 | 2880 | 2925 | 2852 | 1319 | 435 |
| 1600 | 2994 | 2900 | 2495 | 2902 | 3070 | 1499 | 1256 | 430 |
| 2000 | 3039 | 2969 | 2965 | 3033 | 3091 | 2989 | 1206 | 423 |
| 4000 | 3036 | 2942 | 2980 | 3068 | 3092 | 2144 | 596 | 146 |

**IBM Power4 1300 MHz, caches: L1 128KB, L2 1.5MB, L3 32MB**

| n | n proc | Mflops rfptrf | Times rfptrf | Times in rfptrf potrf | trsm | syrk | potrf | lapack potrf | lapack 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 | 9114 | 0.04 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 | 0.16 |
| 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 |

**SUN UltraSPARC-IV, 1300 MHz, caches: L1 64KB, L2 8MB**

| n | n proc | Mflops | Times | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | rfptrf | | in rfptrf | | | | lapack | |
| | | | | 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 |

# Symmetric/Hermitian Indefinite Matrices

**Fred Gustavson gave a talk at PARA 2000 conference in Bergen (Norway) on “A Fast Minimal Storage Symmetric Indefinite Solver”.**

**The method started to be developed at UNI•C (Danish IT Centre for Research and Education) with collaboration of the IBM T.J. Watson Research Center.**

**However the research hasn’t been finished because UNI•C has stopped doing any numerical analysis research development.**

## Reference

**F. Gustavson, A. Karaivanov, M. Marinova, J. Waśniewski, and P. Yalamov. "A Fast Minimal Storage Symmetric Indefinite Solver". In Para'2000 Conference Proceedings, Bergen, Norway, 2000.**

**We decided to continue the research on “A Fast Minimal Storage Factorization of a Symmetric Indefinite Matrix” after the success with the symmetric positive definite matrices.**

**We’ll show here that the factorization of the linear symmetric indefinite matrix in packed version can be obtained by using level 3 BLAS. Comparison numbers between the new Data Format and LAPACK proper subroutines on the modern computers will be presented.**

# Symmetric Indefinite Matrix

$$\overset{A}{\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{pmatrix}} \quad \text{or} \quad \overset{A}{\begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}}$$

**A diagonal pivoting method is used to factor $A$ as**

$$A = P^T\, U\, D\, U^T\, P, \quad \text{if} \;\; \textbf{UPLO} = \text{'U'} \quad \textbf{or}$$

$$A = P^T\, L\, D\, L^T\, P, \quad \text{if} \;\; \textbf{UPLO} = \text{'L'}$$

- **$P$ is a permutation matrix**
- **$U$ and $L$ are unit upper and lower triangular matrices, respectively**
- **$D$ is a symmetric block diagonal with $1 \times 1$ and $2 \times 2$ diagonal blocks.**

# Bunch-Kaufman Pivoting Strategy

$\alpha = (1+\sqrt{17})/8$; $\lambda = |a_{r1}| = \max\{|a_{21}|, \ldots, |a_{n1}|\}$

**if** $\lambda > 0$

  **if** $|a_{11}| \geq \alpha\lambda$ **then** $s = 1;\quad P_1 = I$

  **else**

   $\sigma = |a_{pr}| = \max\{|a_{1r}, \ldots |a_{r-1,r}, |a_{r+1,r}|, \ldots, |a_{nr}|\}$

   **if** $\sigma|a_{11}| \geq \alpha\lambda^2$ **then** $s = 1,\ P_1 = I$

   **else if** $|a_{rr}| \geq \alpha\sigma$

    $s = 1$ **and choose** $P_1$ **so** $(P_1^T A P_1)_{11} = a_{rr}$

   **else**

    $s = 2$ **and choose** $P_1$ **so** $(P_1^T A P_1)_{21} = a_{r1}$

   **end**

  **end**

**end**

## Perturbation Approach

- **Perturbation approach with iterative refinement.**
- **Perturbation approach with the Sherman-Morrison-Woodbury formula.**

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1},$$

  **where $A$ is $n$-by-$n$, $U$ and $V$ are $n$-by-$k$ matrices ($k$ must be small).**

- **Mixed approach.**

## Perturbation Approach

### Recursive Perturbation-Based Algorithm

**We add a small number $\delta > 0$ to each divisor $|a| < \delta$:**

$$a = a + Sgn(a)\delta$$

**where**

$$Sgn(a) = \begin{cases} sign(a) & \textbf{if } a \neq 0 \\ 1 & \textbf{if } a = 0 \end{cases}$$

## Reference

- **F. Gustavson, A. Karaivanov, J. Waśniewski, and P. Yalamov. "A Recursive Formulation of Algorithms for Symmetric Indefinite Linear Systems". UNI•C report, 15 pages, Number UNIC-99-03, 1999.**

**Symmetric/Hermitian Indefinite Matrices**

# Block Packed Algorithm

# Symmetric Indefinite Matrix

$$\mathbf{n} = 7, \quad \textbf{memory needed} = \mathbf{n} \times (\mathbf{n}+1)/2 = 28$$

$$\begin{pmatrix}
a_{1,1_1} & & & & & & \\
a_{2,1_2} & a_{2,2_8} & & & & & \\
a_{3,1_3} & a_{3,2_9} & a_{3,3_{14}} & & & & \\
a_{4,1_4} & a_{4,2_{10}} & a_{4,3_{15}} & a_{4,4_{19}} & & & \\
a_{5,1_5} & a_{5,2_{11}} & a_{5,3_{16}} & a_{5,4_{20}} & a_{5,5_{23}} & & \\
a_{6,1_6} & a_{6,2_{12}} & a_{6,3_{17}} & a_{6,4_{21}} & a_{6,5_{24}} & a_{6,6_{26}} & \\
a_{7,1_7} & a_{7,2_{13}} & a_{7,3_{18}} & a_{7,4_{22}} & a_{7,5_{25}} & a_{7,6_{27}} & a_{7,7_{28}}
\end{pmatrix}$$

**The mapping of 7× 7 real symmetric, complex symmetric or complex Hermitian matrix for the LAPACK algorithm using the packed storage. Lower triangular case.**

## Symmetric Indefinite Matrix

**n** $= 7$, **lda** $= 9$, **nb** $= 2$, **memory needed** $= 51$

$$
\left(\begin{array}{cc|cc|cc|c}
a_{1,1_1} & \otimes & \odot & & & & \\
a_{2,1_2} & a_{2,2_{11}} & \odot & & & & \\
a_{3,1_3} & a_{3,2_{12}} & a_{3,3_{21}} & \otimes & \odot & & \\
a_{4,1_4} & a_{4,2_{13}} & a_{4,3_{22}} & a_{4,4_{29}} & \odot & & \\
a_{5,1_5} & a_{5,2_{14}} & a_{5,3_{23}} & a_{5,4_{30}} & a_{5,5_{37}} & \otimes & \odot \\
a_{6,1_6} & a_{6,2_{15}} & a_{6,3_{24}} & a_{6,4_{31}} & a_{6,5_{38}} & a_{6,6_{43}} & \odot \\
a_{7,1_7} & a_{7,2_{16}} & a_{7,3_{25}} & a_{7,4_{32}} & a_{7,5_{39}} & a_{7,6_{44}} & a_{7,7_{49}} \\
\oslash & \oslash & \oslash & \oslash & \oslash & \oslash & \oslash \\
\oslash & \oslash & \oslash & \oslash & \oslash & \oslash & \oslash
\end{array}\right)
$$

**The mapping of 7$\times$ 7 real symmetric, complex symmetric or complex Hermitian matrix for the algorithm using the block packed overlapped storage. Lower triangular case.**

## Symmetric Indefinite Matrix

**n** $= 7$, **lda** $= 9$, **nb** $= 3$, **memory needed** $= 54$

$$
\left(\begin{array}{ccc|ccc|c}
a_{1,1_{1}} & \otimes & \otimes & \odot & & & \\
a_{2,1_{2}} & a_{2,2_{11}} & \otimes & \odot & & & \\
a_{3,1_{3}} & a_{3,2_{12}} & a_{3,3_{21}} & \odot & & & \\
a_{4,1_{4}} & a_{4,2_{13}} & a_{4,3_{22}} & a_{4,4_{31}} & \otimes & \otimes & \odot \\
a_{5,1_{5}} & a_{5,2_{14}} & a_{5,3_{23}} & a_{5,4_{32}} & a_{5,5_{38}} & \otimes & \odot \\
a_{6,1_{6}} & a_{6,2_{15}} & a_{6,3_{24}} & a_{6,4_{33}} & a_{6,5_{39}} & a_{6,6_{45}} & \odot \\
a_{7,1_{7}} & a_{7,2_{16}} & a_{7,3_{25}} & a_{7,4_{34}} & a_{7,5_{40}} & a_{7,6_{46}} & a_{7,7_{52}} \\
\oslash & \oslash & \oslash & \oslash & \oslash & \oslash & \oslash \\
\oslash & \oslash & \oslash & \oslash & \oslash & \oslash & \oslash_{54}
\end{array}\right)
$$

**The mapping of 7$\times$ 7 real symmetric, complex symmetric or complex Hermitian matrix for the algorithm using the block packed overlapped storage. Lower triangular case.**

## Symmetric Indefinite Matrix

```
SUBROUTINE SSPTRF( UPLO, N, AP, &
      IPIV, INFO)
!
 CHARACTER UPLO
 INTEGER INFO, N
!
 INTEGER IPIV( * )
 REAL AP( * )
```

## Symmetric Indefinite Matrix

**uplo = 'l'**, **n** $= 7$, **lda** $= 9$, **nb** $= 2$, **length of AP** $= 28$

$$\mathbf{AP} = \left(\begin{array}{cc|c|c}
a_{1,1_1} & \otimes & \odot & \diamond \\
a_{2,1_2} & a_{2,2_{11}} & \odot & \\
a_{3,1_3} & a_{3,2_{12}} & a_{3,3_{21}} & \\
a_{4,1_4} & a_{4,2_{13}} & a_{4,3_{22}} & \\
a_{5,1_5} & a_{5,2_{14}} & a_{5,3_{23}} & \\
a_{6,1_6} & a_{6,2_{15}} & a_{6,3_{24}} & \\
a_{7,1_7} & a_{7,2_{16}} & a_{7,3_{25}} & \\
\oslash & \oslash & \oslash & \\
\oslash & \oslash & \oslash &
\end{array}\right)$$

**length of BUFF** $= 31$

$$\mathbf{BUFF} = \left(\begin{array}{cc|cc|c}
a_{3,3_1} & \otimes & \odot & & \\
a_{4,3_2} & a_{4,4_9} & \odot & & \\
a_{5,3_3} & a_{5,4_{10}} & a_{5,5_{17}} & \otimes & \odot \\
a_{6,3_4} & a_{6,4_{11}} & a_{6,5_{18}} & a_{6,6_{23}} & \odot \\
a_{7,3_5} & a_{7,4_{12}} & a_{7,5_{19}} & a_{7,6_{24}} & a_{7,7_{29}} \\
\oslash & \oslash & \oslash & \oslash & \oslash \\
\oslash & \oslash & \oslash & \oslash & \oslash
\end{array}\right)$$

**Block packed overlapped storage. Lower triangular case.**

# Symmetric Indefinite Matrix

$$\mathbf{n} = 7, \quad \mathbf{lda} = 9, \quad \mathbf{nb} = 3, \ \textbf{length of BUFF} = 36$$

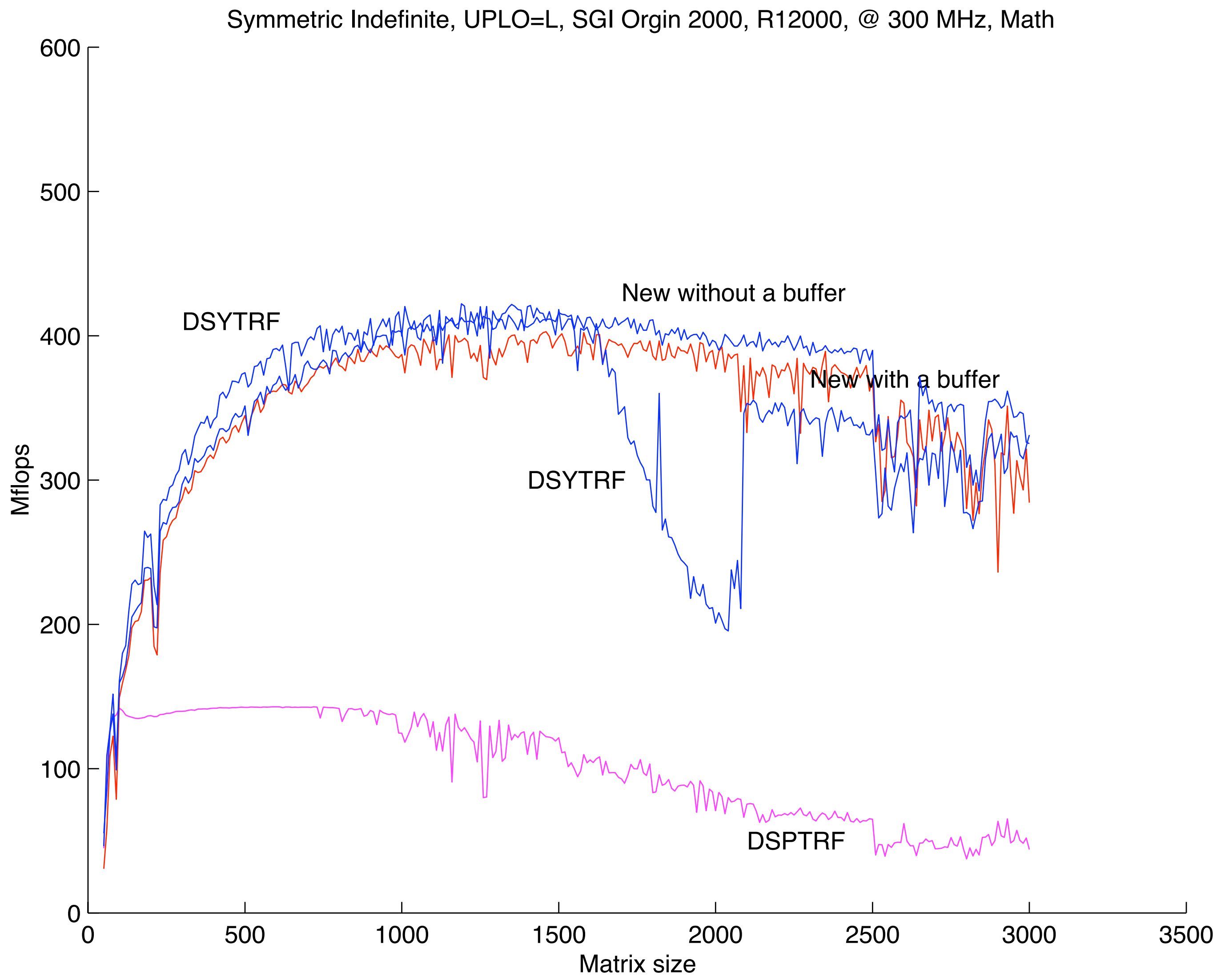
$$\mathbf{BUFF} = \left(\begin{array}{ccc|c}
a_{1,1_1} & \otimes & \otimes & \odot \\
a_{2,1_2} & a_{2,2_{11}} & \otimes & \odot \\
a_{3,1_3} & a_{3,2_{12}} & a_{3,3_{21}} & \odot \\
a_{4,1_4} & a_{4,2_{13}} & a_{4,3_{22}} & a_{4,4_{31}} \\
a_{5,1_5} & a_{5,2_{14}} & a_{5,3_{23}} & a_{5,4_{32}} \\
a_{6,1_6} & a_{6,2_{15}} & a_{6,3_{24}} & a_{6,4_{33}} \\
a_{7,1_7} & a_{7,2_{16}} & a_{7,3_{25}} & a_{7,4_{34}} \\
\oslash & \oslash & \oslash & \oslash \\
\oslash & \oslash & \oslash & \oslash
\end{array}\right)$$

$$\textbf{length of AP} = 28$$

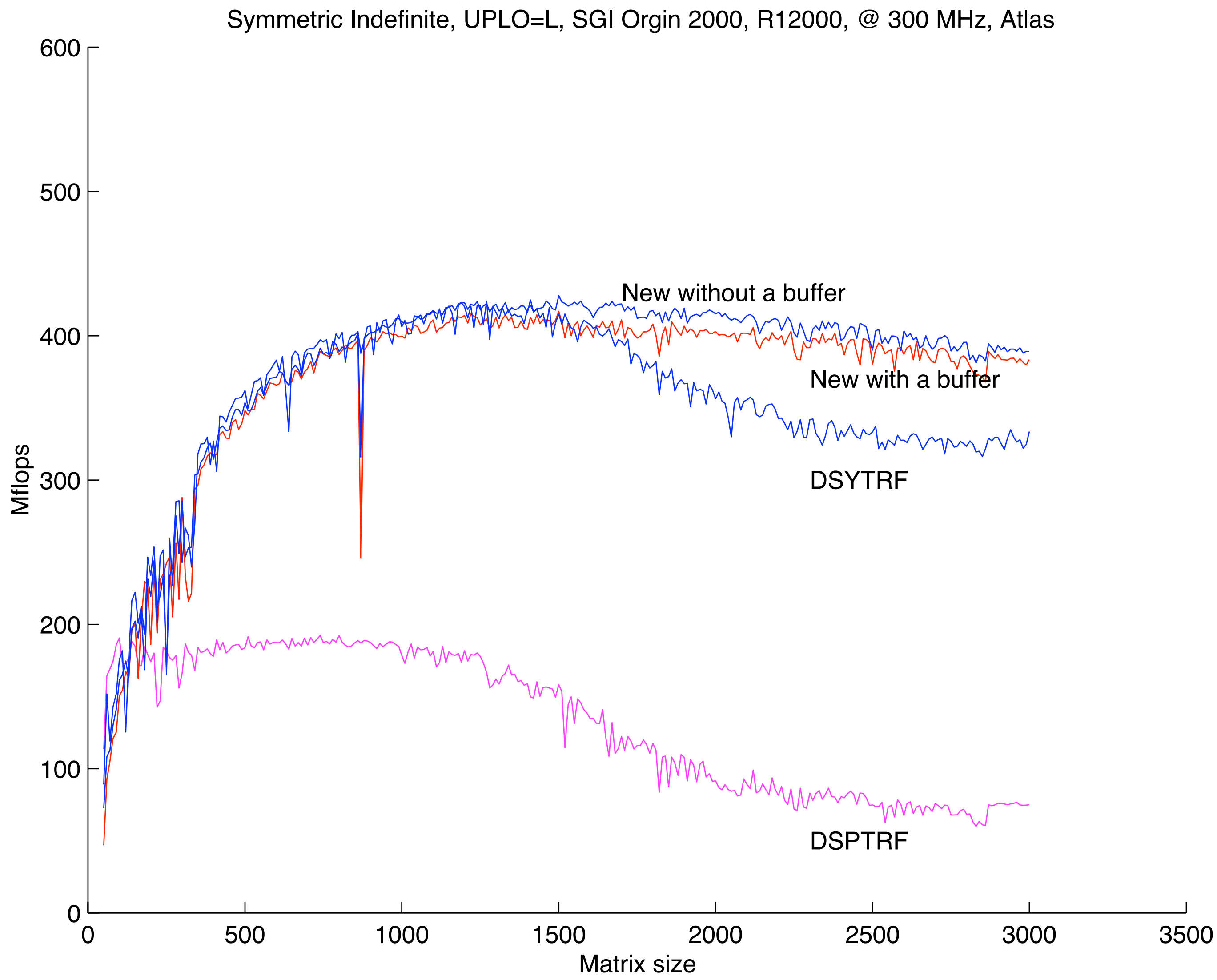
$$\mathbf{AP} = \left(\begin{array}{ccc|cc}
a_{4,4_1} & \otimes & \otimes & \odot & \diamond \\
a_{5,4_2} & a_{5,5_8} & \otimes & \odot & \diamond \\
a_{6,4_3} & a_{6,5_9} & a_{6,6_{15}} & \odot & \diamond \\
a_{7,4_4} & a_{7,5_{10}} & a_{7,6_{16}} & a_{7,7_{22}} & \diamond \\
\oslash & \oslash & \oslash & \oslash & \\
\oslash & \oslash & \oslash & \oslash &
\end{array}\right)$$

**Block packed overlapped storage. Lower triangular case.**

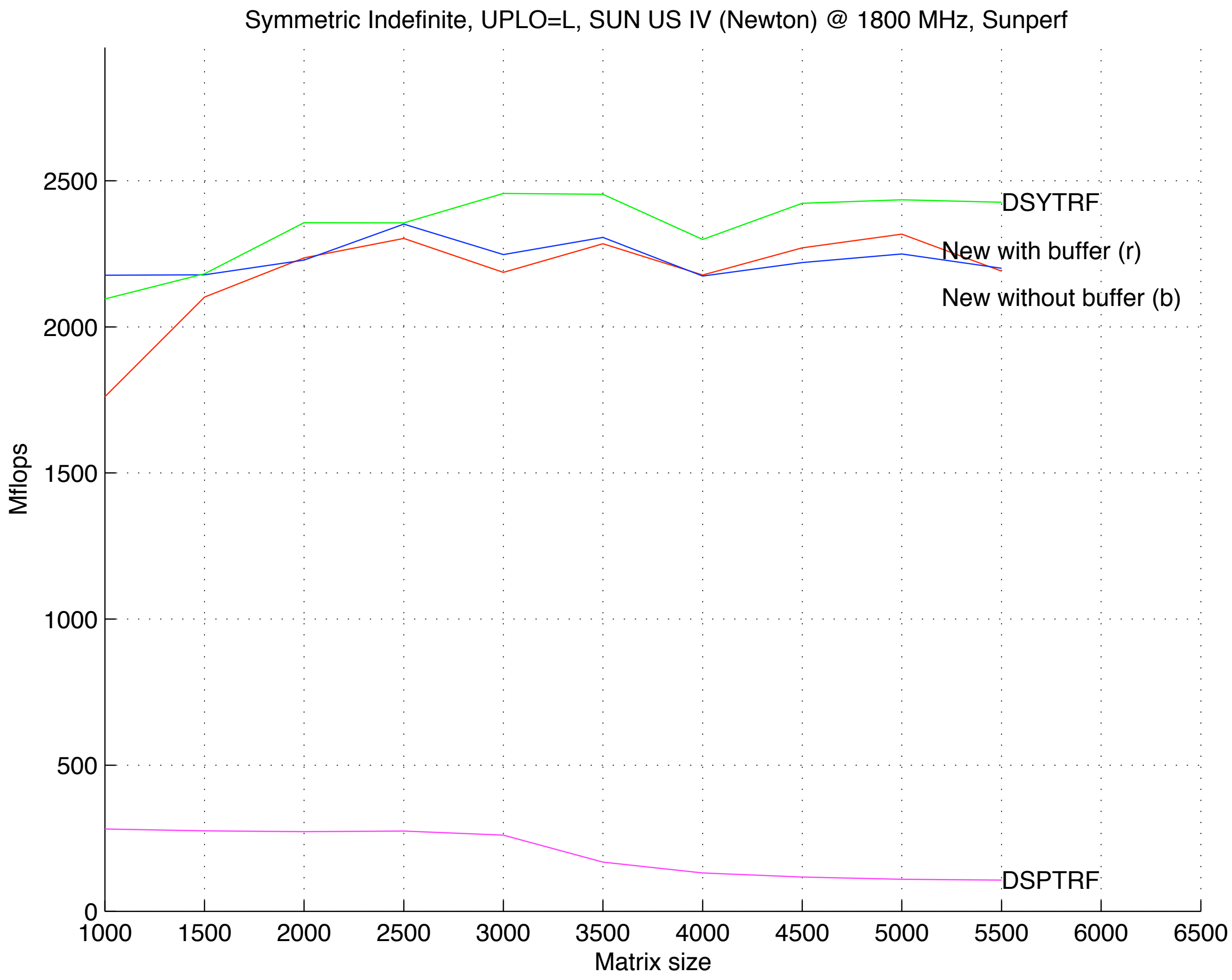
# $LDL^T$, Four Factorization Algorithms, SGI BLAS



# $LDL^T$, Four Factorization Algorithms, Atlas BLAS



# $LDL^T$, Four Factorization Algorithms, Sunperf BLAS



# A Fast Minimal Storage Factorization of Symmetric and Hermitian Matrices

**Jerzy Waśniewski**

**Emeritus Senior Researcher**

**Informatics & Mathematical Modeling (IMM)**

**Technical University of Denmark (DTU)**

**and**

**Emeritus Principal Computer Scientist**

**Danish IT Centre for Research & Education (UNI•C)**

**Email: jw@imm.dtu.dk**

**EU COST Meeting, Lisbon, Portugal, October 19–20, 2009**