

References

- Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., , and Sorensen, D. (1995). *LAPACK Users' Guide, Second Edition*. SIAM Publications, Philadelphia, PA, USA.
- Andersson, S., Bell, R., Hague, J., Holthoff, H., Mayes, P., Nakano, J., Shieh, D., and Tuccillo, J. (1998). Rs/6000 scientific and technical computing: Power3 introduction and tuning guide. RedBooks SG24-5155-00, IBM Corporation, International Technical Support Organization, 11400 Burnet Road, Austin, TX 78758-3493.
- Badia, J. M., Quintana, G., and Vidal, A. M. (1994). Efficient parallel qr decomposition on a network of transputers. In Becker, M., Litzler, L., and Tehel, M., editors, *Transputers '94. Proceedings of the International Conference*, pages 247–265, Amsterdam, Netherlands. IOS Press.
- Baker, G., Gunnels, J., Morrow, G., Riviere, B., and van de Geijn, R. (1998). Plapack: High performance through high-level abstraction. In *Proceedings of the 1998 International Conference on Parallel Processing*, pages 414–422, Lso Alamos, CA. IEEE Comput. Soc.
- Bell, G. and van Ingen, C. (1999). Dsm perspective: another point of view. *Proceedings of the IEEE*, 87(3):412–417.
- Bischof, C., Lang, B., and Sun, X. (1994). Parallel tridiagonalization through two-step band reduction. Technical Report ANL-MCS-P412-0194, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA.
- Bischof, C. and Van Loan, C. (1987). The WY representation for products of Householder matrices. *SIAM J. Sci. Comput.*, 8(1):s2–s13.
- Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D., and Whaley, R. C. (1997). *ScaLAPACK user's guide*. SIAM Publications, Philadelphia, PA, USA.
- Bodin, F. and O'Boyle, M. (1996). A compiler strategy for shared virtual memories. In Szymanski, B. K. and Sinharoy, B., editors, *Proceedings 3rd Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers*, pages 57–69, Norwell, MA. Kluwer Academic Publishers.
- Bova, S. W., Breshears, C. P., Cuicchi, C., Demirbilek, Z., and Gabb, H. (1999). Nesting openmp in an mpi application. In Olariu, S. and Wu, J., editors, *Proceedings of the ISCA 12th International Conference on Parallel and Distributed Computing Systems*, pages 566–571, Cary, NC, USA. ISCA.
- Carrig, J. J. and Meyer, G. G. L. (1997). Efficient householder qr factorization for superscalar processors. *ACM Trans. Math. Software*, 23(3):362–378.

- Carrig, J. J. and Meyer, G. G. L. (1999). A parameterized ordering for cache-, register- and pipeline-efficient givens qr decomposition. *Advances in Comp. Math.*, 10:97–113.
- Chamberlain, R. M. and Powell, M. J. D. (1988). Qr factorization for linear least squares problems on the hypercube. *IMA Journal of Numerical Analysis*, 8(4):401–413.
- Choi, J., Dongarra, J. J., and Walker, D. W. (1995). The design of a parallel dense linear algebra software library: Reduction to Hessenberg, tridiagonal, and bidiagonal form. *Numer. Algorithms*, 10:379–399.
- Chu, E. and George, A. (1989). Qr factorization of a dense matrix on a shared-memory multiprocessor. *Parallel Computing*, 11(1):55–71.
- Cosnard, M., Daoudi, M., Muller, J. M., and Robert, Y. (1986). On parallel and systolic givens factorizations of dense matrices. In et al., M. C., editor, *Parallel Algorithms and Architectures*, pages 245–258, Amsterdam, Netherlands, North Holland. Elsevier Science Pub. Co.
- Dagnum, L. and Menon, R. (1998). Openmp: an industry standard api for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55.
- Darte, A. and Vivien, F. (1997). Optimal fine and medium grain parallelism detection in polyhedral reduced dependence graphs. *International Journal of Parallel Programming*, 25(6):447–496.
- Demmel, J. and Kahan, W. (1990). Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Sta. Comput.*, 11(5):873–912.
- Desprez, F., Dongarra, J., Rastello, F., and Robert, Y. (1998). Determining the idle time of a tiling: new results. *J. Inform. Sci. Engrg.*, 14(1):167–190.
- Dongarra, J. J., Du Croz, J., Hammarling, S., and Duff, I. (1990). A set of level 3 basic linear algebra subprograms. *ACM Trans. Math. Software*, 16(1):1–17.
- Dongarra, J. J., Sameh, A. H., and Sorensen, D. C. (1986). Implementation of some concurrent algorithms for matrix factorization. *Parallel Computing*, 3:25–34.
- Dongarra, J. J., Sorensen, D. C., and Hammarling, S. J. (1989). Block reduction of matrices to condensed forms for eigenvalue computations. *J. Comput. Appl. Math.*, 27:215–227.
- Dunn, I. N. and Meyer, G. G. L. (2002). Qr factorization for shared memory and message passing. *Parallel Computing*, 28:1507–1530.
- Gallivan, K., Jalby, W., Meier, U., and Sameh, A. H. (1988). Impact of hierarchical memory systems on linear algebra algorithm design. *Internat. J. Supercomputer Appl.*, 2(1):12–48.
- Golub, G. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal. Ser. B2*, pages 205–224.
- Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, second edition.
- Jainandunsing, K. and Deprettere, E. F. (1989). A new class of parallel algorithms for solving systems of linear equations. *SIAM J. Sci. Stat. Comput.*, 10(5):880–912.
- Judge, A., Nixon, P., Tangney, B., Weber, S., and Gahill, V. (1999). Distributed shared memory. In Buyya, R., editor, *High performance cluster computing: Architectures and systems*, volume 1, pages 409–438. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Larriba-Pey, J. L., Valero-Garcia, M., and Navarro, J. J. (1992). Parallel qr decomposition on the suprenum multiprocessor. In Valero, M., Onate, E., Jane, M., Larriba, J. L., and Suarez, B., editors, *Parallel Computing and Transputer Applications*, pages 167–176, Barcleona, Spain. CIMNE.
- Lawson, C. L., Hanson, R., and Kincaid, D. (1979). Basic linear algebra subprograms for fortran usage. *ACM Trans. Math. Software*, 5(3):308–323.
- Lim, A. W. and Lam, M. S. (1998). Maximizing parallelism and minimizing synchronization with affine transforms. *Parallel Computing*, 24(3–4).

- Lord, R. E., Kowalik, J. S., and Kumar, S. P. (1983). Solving linear algebraic equations on an mimd computer. *J. ACM*, 30(1):103–117.
- Louka, B. and Tchente, M. (1988). Givens elimination on systolic arrays. In *Proceedings of the 1988 International Conference on Supercomputing*, pages 638–647, New York, NY. ACM.
- Lucka, M., Vajtersic, M., and Viktorinova, E. (1996). Massively parallel poisson and qr factorization solvers. *Computers and Mathematics with Applications*, 31(4-5):19–26.
- Maslennikov, O., Kaniewski, J., and Wyrzykowski, R. (1998). Fault-tolerant qr-decomposition algorithm and its parallel implementation. In Pritchard, D. and Reeve, J., editors, *Proceedings of the 4th International Euro-Par Conference*, pages 798–803, Berlin, Germany. Springer-Verlag.
- Message Passing Interface Forum (1997). Mpi: A message passing interface standard. Technical report, Univ. of Tennessee, Knoxville, TN.
- Meyer, G. G. L. and Pascale, M. (1995). A family of parallel qr factorization algorithms. *Concurrency, Practice and Experience*, 8(6):461.
- Modi, J. J. and Clarke, R. M. B. (1984). An alternate givens ordering. *Numer. Math.*, 43(1):83–90.
- OpenMP Architecture Review Board (1999). Openmp fortran application program interface. Technical report.
- Park, N., Prasanna, V. K., and Raghavendra, C. S. (1999). Efficient algorithms for block-cyclic array redistribution between processor sets. *IEEE Trans. on Parallel and Distributed Computing*, 10(12):1217–1240.
- Petit, A. P. and Dongarra, J. J. (1999). Algorithm redistribution methods for block-cyclic decompositions. *IEEE Trans. on Parallel and Distributed Systems*, 10(12):1201–1216.
- Porta, T. (1988). Using givens rotations to solve dense linear systems on the hypercube. In Kartashev, L. P. and Kartashev, S. I., editors, *Proceedings of the Third International Conference on Supercomputing*, volume 2, pages 443–447, St. Petersburg, FL, USA. International Supercomputing Institute.
- Pothen, A. and Raghavan, R. (1989). Distributed orthogonal factorizations: Givens and householder algorithms. *SIAM J. Sci. Stat. Comput.*, 10(6):1113–1114.
- Protic, J., Tomasevic, M., and Milutinovic, V. (1996). Distributed shared memory: concepts and systems. *IEEE Parallel Distrib. Technol., Syst. Appl.*, 4(2):63–71.
- Sahni, S. and Thanvantri, V. (1996). Performance metrics: Keeping the focus on runtime. *IEEE Parallel Distrib. Technol., Syst. Appl.*, 4(1):43–56.
- Sameh, A. and Kuck, D. (1978). On stable parallel linear system solvers. *J. ACM*, 25(1):81–91.
- Sarkar, V. (1997). Automatic selection of high-order transformations in the ibm xl fortran compilers. *IBM J. Res. Develop.*, 41(3):233–264.
- Schreiber, R. and Van Loan, C. (1989). A storage-efficient WY representation for products of Householder transformations. *SIAM J. Sci. Stat. Comput.*, 10(1):53–57.
- Smith, A. and Suri, S. (2000). Rectangular tiling in multidimensional arrays. *Journal of Algorithms*, 37(2):451–467.
- Sun, X.-H. and Gustafson, J. L. (1991). Toward a better parallel performance metric. *Parallel Computing*, 17:1093–1109.
- Throop, J. (1999). Openmp: shared-memory parallelism from the ashes. *Computer*, 32(5):108–109.
- Wilburn, V. C., Hak-Lim, K., and Alexander, W. E. (1996). An algorithm and architecture for the parallel solution of systems of linear equations. In *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pages 392–398, New York, NY. IEEE.
- Wilson, G. V. (1993). The history of the development parallel computing.

- Wilson, R. P., French, R. S., Wilson, C. S., Amarasinghe, S. P., Anderson, J. M., Tjiang, S. W. K., Shih-Wei, L., Chau-Wen, T., Hall, M. W., Lam, M. S., and Hennessy, J. L. (1994). Suif: an infrastructure for research on parallelizing and optimizing compilers. *SIGPLAN Notices*, 29(12):31–37.
- Wolf, M. E. and Lam, M. S. (1991). A loop transformation theory and algorithm to maximize parallelism. *IEEE Trans. on Parallel and Distributed Systems*, 2(4):452–471.
- Wolfe, M. (1996a). Parallelizing compilers. *ACM Computing Surveys*, 28(1):261–262.
- Wolfe, M. J. (1996b). *High performance compilers for parallel computing*. Addison-Wesley Publishing Company, Redwood City, CA, USA.
- Wright, K. (1991). Parallel algorithms for qr decomposition on a shared memory multiprocessor. *Parallel Computing*, 17(6-7):779–790.

Index

- Algorithm, 2
 - Asynchronous parameterized parallel fast Givens, 57
 - Compact WY, 38
 - Compact WY QR factorization, 3
 - Example algorithm, 3
 - Fast Givens QR factorization, 3
 - Hybrid parameterized parallel fast Givens, 59
 - Load balancing, 49, 50, 79
 - Matrix bidiagonalization, 3
 - Modified Gram-Schmidt, 18
 - Parallel matrix bidiagonalization, 91, 92
 - Parameterized parallel compact WY, 82
 - Parameterized parallel fast Givens, 42, 50
 - Standard Compact WY QR factorization, 40
 - Standard fast Givens QR factorization, 33, 35
 - Standard Givens QR factorization, 32
 - Standard Householder bidiagonal factorization, 40
 - Standard Householder QR factorization, 37
 - Synchronous parameterized parallel fast Givens, 52
 - Task partitioning, 79
- AP, 56
- ARC, 81
- Arithmetic complexity, 11, 49
- ASC, 82
- ASYNC, 57, 60, 61
- Asynchronous message passing procedure, 56
- Asynchronous parameterized parallel fast Givens algorithm, 57
- Asynchronous receive/send communication procedure, 81
- Asynchronous send communication procedure, 82
- Basic primitive, 15, 16
- Block QR factorization, 38
- Cache, 7, 14
- Compact WY, 38
- Compact WY algorithm, 38
- Compact WY QR factorization algorithm, 3
- Concurrency set, 24, 47, 48
 - Sink, 17
 - Source, 17
- CWY, 38
- Distributed memory model, 9, 10
- Distributed shared memory, 7
- Efficiency, 12
- Execution time, 12
- Fast Givens QR factorization algorithm, 3
- Fat tree, 8
- Givens rotation, 89
- Givens-based QR factorization, 3, 31, 59
- HMP, 58
- Householder bidiagonal factorization, 40
- Householder-based QR factorization, 3, 36, 59
- HP, 8, 14, 42, 60–62, 66–69, 84, 85, 89
 - MLIB, 61
- HPFG, 59
- Hybrid message passing procedure, 58
- Hybrid parameterized parallel fast Givens algorithm, 59
- Hybrid shared memory/message passing, 10, 41, 42, 57, 58, 60, 65, 67
- Hypercube, 8
- IBM, 8–10, 14, 42, 60, 67, 68, 70
- Interconnection network, 15

- LAPACK, 89, 90, 95
 - DBDSQR, 95
 - DGEQRF, 61, 64, 83, 84
 - DGESVD, 95, 96, 98
- Latency, 11
- LB, 79
- Leading task, 78
- Least-recently-used replacement policy, 15
- LH, 94
- Linear array, 15
- Load balancing algorithm, 49, 50, 79
- Load imbalance, 24
- Matrix bidiagonalization algorithm, 3
- Memory hierarchy, 10, 13, 14
 - Cache, 7, 14
 - Register file, 14
 - Value-based reuse, 19
- Memory hierarchy parameterization procedure, 23, 44
- Memory hierarchy reuse, 13, 21, 48
 - Temporal, 22
 - Value-based, 19
- Mesh, 8
- Message Passing Interface (MPI), 10
- Modified Gram-Schmidt algorithm, 18
- MPI, 10
- Network interface controller, 14, 15
- OpenMP, 10
- Ordering constraints, 17
- Parallel algorithm synthesis procedure, 3
- Parallel matrix bidiagonalization algorithm, 91, 92
- Parameterized parallel compact WY algorithm, 82
- Parameterized parallel fast Givens algorithm, 42, 50
- PCWY, 82–85, 87
- PFG, 42, 52, 60–62
- PMB, 90, 91, 93–98
- Procedure, 2
 - Asynchronous message passing, 56
 - Asynchronous receive/send communication, 81
 - Asynchronous send communication, 82
 - Example procedure, 3
 - Hybrid message passing, 58
 - Memory hierarchy parameterization, 23, 44
 - Sink concurrency set definition, 17
 - Source concurrency set definition, 17
 - Superscalar parameterization, 19, 20, 22, 42
 - Synchronous message passing, 51
- Programming model
 - Distributed memory, 9, 10
 - Hybrid shared memory/message passing, 10, 41, 42, 57, 58, 60, 65, 67
 - Shared memory, 9, 10
- Register file, 14
- Row range, 51
- SBH, 94
- ScaLAPACK PDGEQRF, 41, 61, 64, 83, 84
- Scalar unit, 14, 15
- SCWY, 40
- SFG, 33, 35
- SG, 32
- SGI, 8–10, 14, 42, 60–62, 67–69, 84, 85, 87, 90, 96
 - SGIMATH, 61
- SH, 37, 40
- Shared memory model, 9, 10, 67
- SHB, 40, 90, 94
- Sink concurrency set, 17
- Sink concurrency set definition procedure, 17
- Source concurrency set, 17
- Source concurrency set definition procedure, 17
- SP, 51
- Speedup, 12
- Standard Compact WY QR factorization algorithm, 40
- Standard fast Givens QR factorization algorithm, 33, 35
- Standard Givens QR factorization, 31
- Standard Givens QR factorization algorithm, 32
- Standard Householder bidiagonal factorization algorithm, 40
- Standard Householder QR factorization algorithm, 37
- Subordinate task, 78
- Superscalar parameterization procedure, 19, 20, 22, 42
- Superscalar processor, 7, 14–16
- SYNC, 52, 60, 61
- Synchronization index, 47
- Synchronous message passing procedure, 51
- Synchronous parameterized parallel fast Givens algorithm, 52
- Task, 48
 - Leading, 78
 - Subordinate, 78
- Task definition, 47
- Task index, 47
- Task partitioning algorithm, 79
- Temporal reuse, 22
- Throughput, 11
- Torus, 8
- TP, 79
- Value-based data dependency, 17
- Value-based reuse, 19