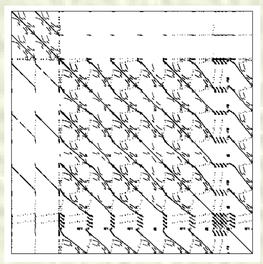


MUMPS

A MUltifrontal Massively Parallel Solver

http://graal.ens-lyon.fr/MUMPS/
http://www.enseeiht.fr/apo/MUMPS/

MAIN FEATURES

MUMPS solves large systems of linear equations of the form Ax=b by factorizing A into A=LU or LDL^T

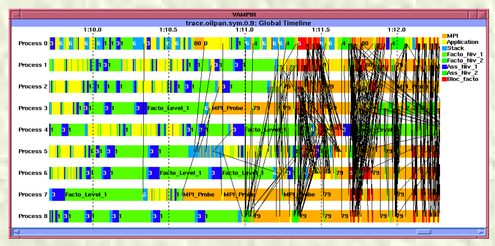
- Symmetric or unsymmetric matrices (partial pivoting)
- Parallel factorization and solution phases (uniprocessor version also available)
- Iterative refinement and backward error analysis
- Various matrix input formats
  - **assembled** format
  - **distributed** assembled format
  - sum of **elemental** matrices
- Partial factorization and Schur complement matrix
- Version for complex arithmetic
- Several orderings interfaced: AMD, AMF, PORD, METIS, SCOTCH

Recent features

- Symmetric indefinite matrices: preprocessing and 2-by-2 pivots
- Hybrid scheduling
- 2D cyclic distributed Schur complement
- Sparse multiple right-hand side
- Interfaces to MUMPS: Fortran, C, Matlab (S. Pralet, ENSEEIHT-IRIT) and Scilab (A. Fèvre, INRIA)

IMPLEMENTATION

- Distributed multifrontal solver (MPI / F90 based)
- Dynamic distributed scheduling to accomodate both numerical fill-in and multi-user environments
- Use of BLAS, ScaLAPACK

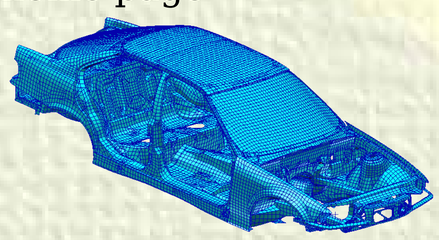


A fully asynchronous distributed solver (VAMPIR trace, 8 processors)

AVAILABILITY

- ***MUMPS*** is available free of charge
- It is used on a number of platforms (CRAY, SGI, IBM, Linux, ...) and is downloaded once a day on average (applications in chemistry, aeronautics, geophysics, ...)
- If you are interested in obtaining MUMPS for your own use, please refer to the MUMPS home page

Car body
148770 unknowns and
5396386 nonzeros
MSC.Software



Some MUMPS users: Boeing, BRGM, CEA, Dassault, EADS, EDF, MIT, NASA, SAMTECH, ...

COMPETITIVE PERFORMANCE

- Comparison with SuperLU extracted from ACM TOMS 2001 and obtained with S. Li

Factorization time in seconds of large matrices on the CRAY T3E (1 proc: not enough memory)

Matrix	Solver	Number of processors						
		1	4	8	16	32	64	128
Bbmat	MUMPS	-	32.1	10.8	12.3	10.4	9.1	7.8
	SuperLU	-	132.9	72.5	39.8	23.5	15.6	11.1
Ecl32	MUMPS	-	23.9	13.4	9.7	6.6	5.6	5.4
	SuperLu	-	48.5	26.6	15.7	9.6	7.6	5.6

- Recent performance results (ops = nb of operations)

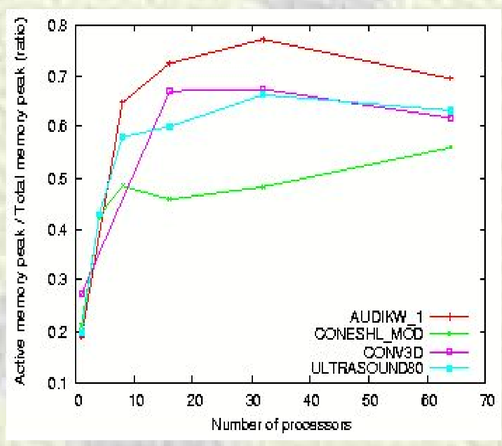
Matrix	Ops $x10^9$	Factorization time in seconds on 1 proc	Factorization time in seconds on 64 proc	Factorization time in seconds on 128 proc
AUDIKW_1	5682	3262.8	54.6	35.9
BRGM	31010	-	283.9	-
CONESHL_mod	1640	1099.0	19.6	12.1
CONV3D64	23880	-	207.5	146.5
ULTRASOUND80	3915	1542.2	37.1	29.5

CURRENT RESEARCH: ACTIVE RESEARCH FEEDS MUMPS

On-going research on Out-of-Core solvers

(Ph.D. E. Agullo, ENS Lyon and Ph.D. M. Slavova, CERFACS)

- Use disk storage to solve very large problems
- Parallel out-of-core factorization
- Preprocessing to minimize volume of I/O
- Scheduling for out-of-core solution



Ratio of active and total memory peak on different numbers of processors for several large problems

Hybrid Scheduling

- Both memory and workload information are used to obtain a better behaviour in terms of estimated memory, memory used and factorization time in the context of parallel factorization algorithms
- Estimated memory much closer to effective memory used

		MUMPS standard		MUMPS hybrid	
Matrix		Estim	real	Estim	real
AUDIKW_1	Max	118.7	50.7	73.9	41.9
	Avg	76.2	31.4	49.5	32.1
BRGM	Max	406.6	-	257.6	175.1
	Avg	185.0	-	158.9	123.5
CONESHL_mod	Max	59.6	33.1	33.8	22.5
	Avg	25.2	16.8	21.6	16.2
CONV3D64	Max	93.7	88.4	86.9	81.0
	Avg	68.7	60.5	60.9	60.2
ULTRASOUND80	Max	43.0	38.9	29.3	27.2
	Avg	26.2	22.4	23.5	21.8

Estimated and effective memory (millions of reals) for the factorization on 64 processors

Max: maximum amount of memory
Avg: average memory per processor

Related project: Grid TLSE

- Expertise website for sparse linear algebra
- On a user's specific problem, compare execution time, accuracy, memory usage, ... of various solvers
- Find best parameter values and reordering heuristics on a given problem

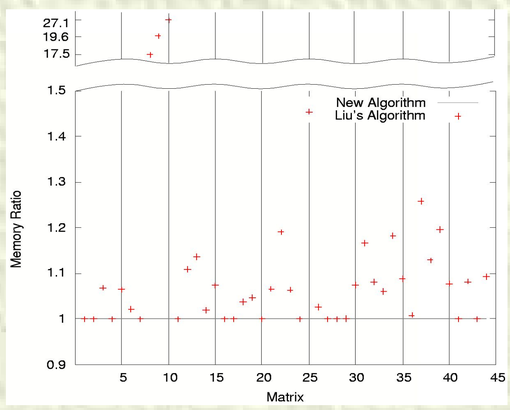
http://www.enseeiht.fr/lima/tlse/

Memory minimizing schedules

- Multifrontal methods can use a large amount of temporary data
- By decoupling task allocation and task processing, we can reduce the amount of temporary data: new optimal schedule proposed in this context (Guermouche, L'Excellent, ACM TOMS)
- Memory gains:

Active memory ratio (new algorithm vs Liu's ordering)

Remark: Gains relative to Liu's algorithm are equal to 27.1, 17.5 and 19.6 for matrices 8, 9, and 10 (gupta matrices), respectively



Preprocessing for symmetric matrices (S. Pralet, ENSEEIHT-IRIT)

- Preprocessing: new scaling available, symmetric weighted matching and automatic tuning of the preprocessing strategies
- Pivoting strategy (2-by-2 pivot and static pivoting)
- Improvement:
  - factorization time
  - robustness in particular on KKT systems arising from optimization
  - memory estimation

Factorization time on a Linux PC (Pentium4 2.80 GHz)

Matrix	n	nnz	Old	New
CONT-300	180095	539396	-	4.2
BOYD2	466316	890091	-	2.6
STOKES128	49666	295938	1.5	0.8

The latest versions of MUMPS are public domain, based on public domain software developed during the Esprit IV European project PARASOL (1996-1999). Since 1999, the developments are supported by CERFACS, ENSEEIHT-IRIT, and INRIA Rhône-Alpes.

Current development team: Patrick Amestoy, Aurelia Fèvre, Abdou Guermouche, Jean-Yves L'Excellent, Stéphane Pralet. Main contributors: Patrick Amestoy, Iain Duff, Abdou Guermouche, Jacko Koster, Jean-Yves L'Excellent, and Stéphane Pralet.

Other contributors: Emmanuel Agullo, Caroline Bousquet, Christophe Daniel, Vincent Espirat, Chiara Puglisi, Grégoire Richard, Mila Slavova, Miroslav Tuma, and Christophe Vömel.