STRUMPACK: Scalable Preconditioning using Low-Rank Approximations and Random Sampling

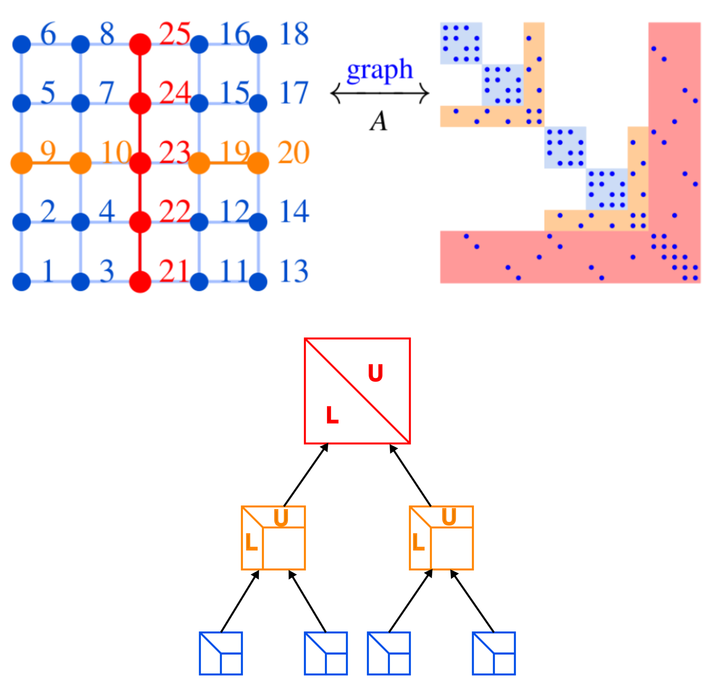
Pieter Ghysels, Xiaoye S. Li, Christopher Gorman and François-Henry Rouet
Scalable Solvers Group, Lawrence Berkeley National Laboratory, Berkeley, California, USA

Abstract

We present a parallel and fully algebraic preconditioner based on an approximate sparse factorization using rank-structured matrix compression. In sparse multifrontal LU factorization, the fill-in occurs in dense frontal matrices. These are approximated as Hierarchically Semi-Separable (HSS) rank-structured matrices using an efficient randomized sampling technique. The resulting fast solver or preconditioner has optimal or close to optimal complexity – in terms of floating point operations and memory usage – for matrices from several types of discretized partial differential equations. Our STRUMPACK approximate solver is a viable alternative to other state-of-the-art preconditioners like ILU and AMG and we demonstrate that our new method is more robust and scalable than existing ones.

Multifrontal LU factorization (sparse Gaussian elimination)

- Column permutation, row/column scaling for stability (optional)
- Fill-in-reducing ordering
  - Nested-dissection
  - Defines elimination/separator tree
  - (Par)Metis/(PT)Scotch graph partitioners
- At every node in the separator tree
  - Dense frontal matrix
  - Partial LU factorization
  - Schur complement update
  - Extend-add: parent nodes "sum" Schur complements from the children
- Linear solve phase
  - Forward and backward solve: bottom-up and top-down tree traversals



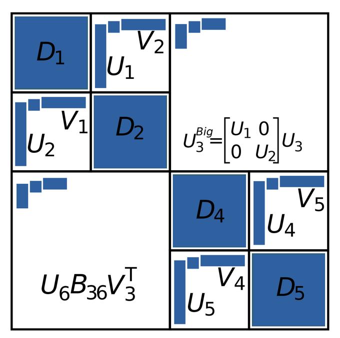
Hierarchically Semi-Separable (HSS) matrix format

- Data-sparse representation, cf. $\mathcal{H}$, $\mathcal{H}^2$, BLR, HODLR, ...
- HSS $\subseteq \mathcal{H}^2 \subseteq \mathcal{H}$
- Hierarchical partitioning
- Off-diagonal blocks are approx as low-rank

$$A_{\nu_1,\nu_2} = A(I_{\nu_1}, I_{\nu_2}) = U^{\mathrm{Big}}_{\nu_1} B_{\nu_1,\nu_2} (V^{\mathrm{Big}}_{\nu_2})^*$$

- Full rank diagonal blocks

$$D_\tau = A(I_\tau, I_\tau)$$



- Column and row bases are nested

$$U^{\mathrm{Big}}_\tau = \begin{bmatrix} U_{\nu_1} & 0 \\ 0 & U_{\nu_2} \end{bmatrix} U_\tau$$

Randomized HSS Construction for a Frontal Matrix

- R random matrix with $d = r + p$ columns
  - r is the estimated maximum rank and p is an oversampling parameter
- $S^r = AR$: sample rows of A, columns of S^r span the column space of A
- $S^c = A^*R$: sample columns of A, columns of S^c span the row space of A
- Sample off-diagonal blocks $S^{r,(\ell)} = \left(A - D^{(\ell)}\right) R = S^r - D^{(\ell)} R$
- Apply rank-revealing QR factorization (column pivoted QR) to $S^{r,(\ell)}$ and $S^{c,(\ell)}$ on each level of the HSS hierarchy

Benefits of randomized algorithm

- Extend-add operation is simplified. No explicit large dense matrices (save memory).
- $\mathcal{O}(r^2 N)$ when fast ($\mathcal{O}(N)$) matrix-vector product is available

STRUMPACK – STRUctured Matrix PACKage

Solver & Preconditioner for Sparse Linear Systems

- General algebraic sparse linear solver/preconditioner
- Rank-structured sub-matrices: Hierarchically Semi-Separable (HSS)
- Aimed at matrices from PDE discretizations
- Efficient hybrid MPI+OpenMP implementation
- Real, complex, single & double precision and 64-bit indexing (C++ templates)
- BSD licensed

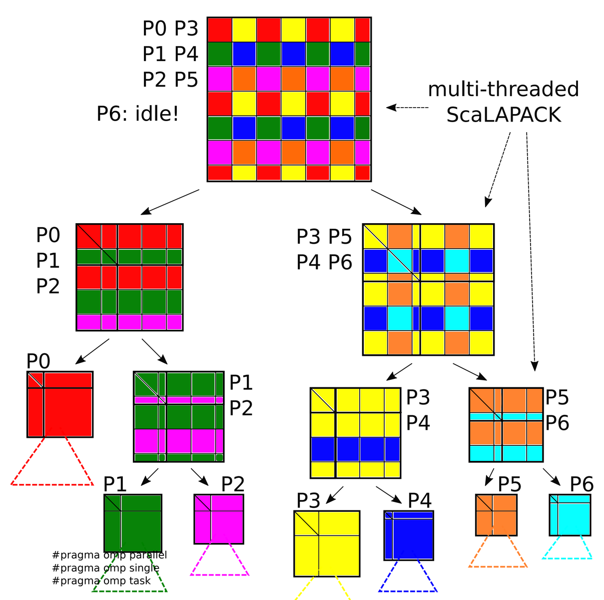
Dense Rank-Structured Matrices

- Fast HSS matrix construction using Randomized sampling/projection and Rank-revealing QR factorization
- ULV factorization and solve
- HSS-matrix $\times$ vector product
- Applicable to Toeplitz, Cauchy, BEM, QuantumChem, machine learning, ...
- Scalable distributed memory MPI code

http://portal.nersc.gov/project/sparse/strumpack/

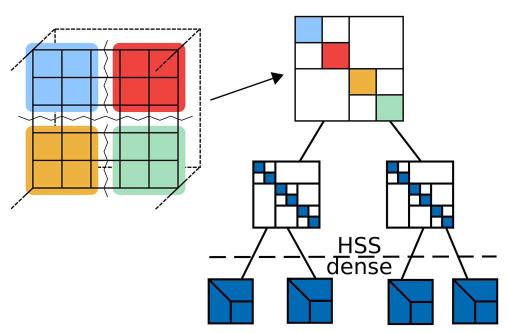
Data layout and partitioning

- ScaLAPACK 2D block cyclic layout
- Multi-threaded ScaLAPACK
- Allow idle procs to reduce comm
  - Prefer 2×3 BLACS grid over 1×7
- Subtree mapped to MPI procs proportional to estimated work
- Each MPI proc has local subtree, using OpenMP task parallelism
- System BLAS is called
  - *outside* parallel: MT
  - *within* OpenMP parallel: seq
- Also proportional mapping based on HSS tree → ranks not known a-priori

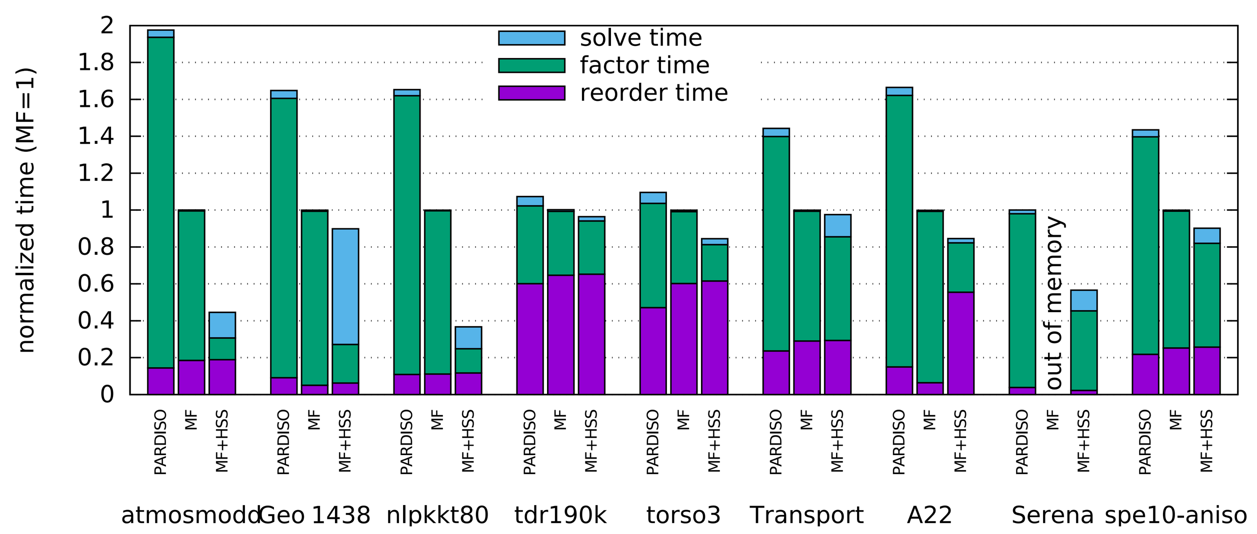


Multifrontal HSS-enabled sparse solver/preconditioner

- HSS approx. for largest fronts only
- HSS partitioning from recursive bisection of separator
  - Uses METIS partitioner
  - Goal is to reduce HSS ranks
- HSS construction via random sampling: BLAS3 usage
- Partial ULV-like HSS matrix factorization



Numerical Results – Shared Memory OpenMP Task Parallelism

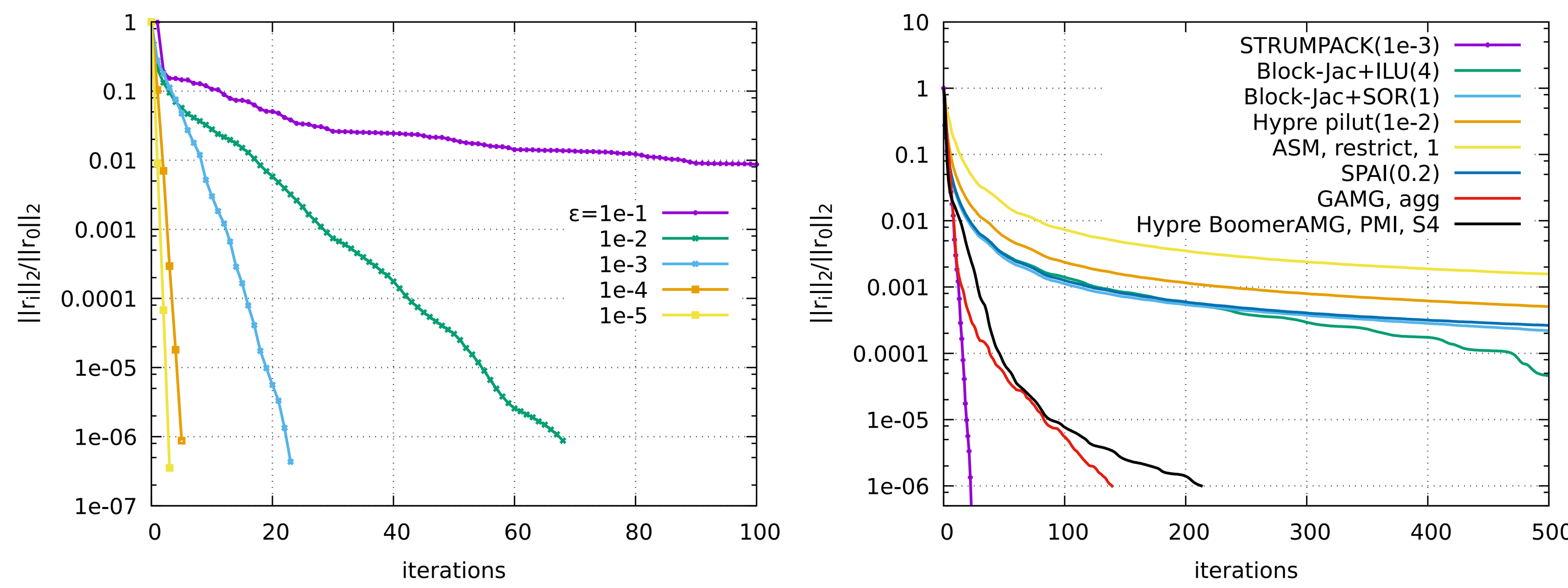


- Vs Intel MKL PARDISO
- 12-core Intel Ivy Bridge
- UF set, SciDAC
- Metis nested - dissection

Numerical Results – Distributed Memory Code

ML_Geer.mtx matrix from UF Sparse matrix collection

- Non-symmetric matrix, Petrov-Galerkin discretization of porous medium
- $N = 1,504,002$, nnz $= 110,686,677$
- 96 MPI processes on NERSC's Edison (Cray XC30)



- STRUMPACK(ϵ) preconditioner depends on tolerance ϵ used in rank-revealing factorization (low-rank approximation accuracy)
- Comparison with preconditioners available from PETSc:

Block-Jacobi+ILU($k = \{0, 1, 2, 3, 4\}$), 1 block per MPI. **Block-Jacobi+SOR**($\omega = \{.6, .8, 1., 1.2, 1.4\}$). **Jacobi**: diagonal scaling. **PILUT**($\epsilon = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$), Hypre. **ASM**: type = $\{basic, restrict, interpolate, none\}$, overlap = $\{1, 2, 3\}$. **SPAI**, $\epsilon = \{.2, .4, .6, .8\}$. **GAMG**: PETSc's AMG, type = $\{smoothedaggregation, classical, hybridgeometric\}$, bs = #dofs. **BoomerAMG** Hypre's AMG. Default (2D) and coarsening = $\{HMIS, PMIS\}$, interpolation = $\{ext{+}i, ext{+}i\text{-}cc\}$ and $\{4, 5\}$ elems/row.

TIMINGS WILL BE UPDATED IN FINAL POSTER

PETSc Interface

As of PETSc 3.8, an interface to STRUMPACK-sparse is available.

- Configure PETSc with STRUMPACK support:
  `./configure --download-strumpack ....`
- Select STRUMPACK direct solver (`lu`) or preconditioner (`ilu`):

```
mpirun -n 9999 ./science_app -pc_type (i)lu
  -pc_factor_mat_solver_package strumpack
```

Try it!

Summary

- Developed an efficient sparse direct solver (MPI+OpenMP)
- HSS preconditioner has (close to) linear or scaling for PDE problems
- Preconditioner appears to be robust; on problems where AMG has slow convergence

References

- Xia. *Randomized sparse direct solvers.* SIAM SIMAX, 34.1 (2013): 197-227.
- Martinsson. *A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix.* SIAM SIMAX, 32.4 (2011): 1251-1274.
- Chandrasekaran et al. *On the numerical rank of the off-diagonal blocks of Schur complements of discretized elliptic PDEs.* SIAM SIMAX, 31.5 (2010): 2261-2290.
- Rouet, et al. *A distributed-memory package for dense Hierarchically Semi-Separable matrix computations using randomization.* TOMS 42.4 (2016)
- Ghysels, et al. *An efficient multi-core implementation of a novel HSS-structured multifrontal solver using randomized sampling.* SIAM SISC, Special issue CSE 2015.