jh170011-DAJ

Takahiro Katagiri (Nagoya University)

High-performance Randomized Matrix Computations for Big Data Analytics and Applications

● Background

- Developing random sketching algorithms with high-performance implementations on supercomputers to compute **singular value decomposition (SVD) and linear system (LS)** solutions of very large-scale matrices.
- Few numerical solvers, especially **randomized algorithms**, are designed to tackle very large-scale matrix computations on the latest supercomputers. .
- We intend to develop efficient sketching schemes to compute approximate SVD and LS solutions of large-scale matrices. The main idea is to sketch the matrices by randomized algorithms **to reduce the computational dimensions and then suitably integrate the sketches** to improve the accuracy and to lower the computational costs.
- We intend to implement the proposed algorithms on supercomputers. One essential component of this project is to develop effective **automatic software auto-tuning (AT)** technologies, so that the package can fully take advantage of the computational capabilities of the target supercomputers that include CPU homogeneous and CPU-GPU heterogeneous parallel computers.

● Members

- **Takahiro Katagiri** (Nagoya U., Japan) : AT (ppOpen-AT), parallel eigenvalue algorithms, and supercomputer implementations.
- **Weichung Wang** (National Taiwan U., Taiwan): Numerical linear algebra, parallel computing, and AT (surrogate-assisted turning). and big data applications.
- **Su-Yun Huang** (Institute of Statistical Science, Academia Sinica, Taiwan) : Mathematical statistics and machine learning (random sketching algorithm).
- **Kengo Nakajima** (U. Tokyo, Japan) : Parallel algorithms in numerical iterative method (hybrid MPI/OpenMP execution).
- **Osni Marques** (LBNL, USA) : Eigenproblem and its implementation (LAPACK, SVD algorithms).
- **Feng-Nan Hwang** (National Central U., Taiwan) : Eigenproblem and its parallelization (SLEPc, SVD algorithms)
- **Toshio Endo** (TITECH, Japan) : System software (optimizations for hierarchical memory and adaptation of its AT)
- (Collaborator) **Akihiro Ida** (U. Tokyo, Japan): Providing matrices by H-Matrix

● iSVD Algorithm

Rank-k SVD

$$\boldsymbol{A} \approx \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$$

$\boldsymbol{U}_k$ is an $m \times k$ orthonormal matrix that $k < m$, $\boldsymbol{\Sigma}_k$ is a $k \times k$ diagonal matrix, and $\boldsymbol{V}_k$ is an $n \times k$ orthonormal matrix. The columns of $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ are the leading left singular vectors and right singular vectors of $\boldsymbol{A}$, respectively. The diagonal entries of $\boldsymbol{\Sigma}_k$ are the k largest singular values of $\boldsymbol{A}$.

Algorithm 2 Integrated SVD with multiple sketches (iSVD).

Require: Input $\boldsymbol{A}$ (real $m \times n$ matrix), k (desired rank of approximate SVD), p (oversampling parameter), $\ell = k + p$ (dimension of the sketched column space), q (power of projection), N (number of random sketches)

Ensure: Approximate rank-k SVD of $\boldsymbol{A} \approx \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{\Sigma}}_k \widehat{\boldsymbol{V}}_k^\top$

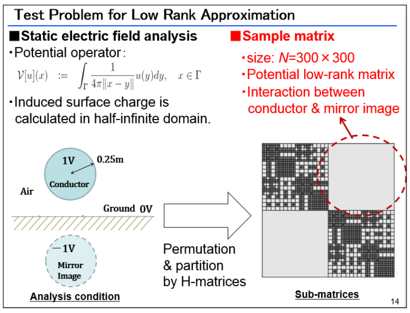
Generate $n \times \ell$ random matrices $\boldsymbol{\Omega}_{[i]}$ for $i = 1, \ldots, N$
Assign $\boldsymbol{Y}_{[i]} \leftarrow (\boldsymbol{A}\boldsymbol{A}^\top)^q \boldsymbol{A}\boldsymbol{\Omega}_{[i]}$ for $i = 1, ..., N$ with $\boldsymbol{\Omega}_{[i]} = \boldsymbol{\Omega}_{gp}$ or $\boldsymbol{\Omega}_{cs}$ (in parallel)
Compute $\boldsymbol{Q}_{[i]}$ whose columns are orthonormal basis of $\boldsymbol{Y}_{[i]}$ (in parallel)
Integrate $\overline{\boldsymbol{Q}} \leftarrow \{\boldsymbol{Q}_{[i]}\}_{i=1}^N$ (by Algorithm 3 or Algorithm 4)
Compute SVD of $\overline{\boldsymbol{Q}}^\top \boldsymbol{A} = \widehat{\boldsymbol{W}}_\ell \widehat{\boldsymbol{\Sigma}}_\ell \widehat{\boldsymbol{V}}_\ell^\top$
Assign $\widehat{\boldsymbol{U}}_\ell \leftarrow \overline{\boldsymbol{Q}} \widehat{\boldsymbol{W}}_\ell$
Extract the largest k singular-pairs from $\widehat{\boldsymbol{U}}_\ell$, $\widehat{\boldsymbol{\Sigma}}_\ell$, $\widehat{\boldsymbol{V}}_\ell$ to obtain $\widehat{\boldsymbol{U}}_k$, $\widehat{\boldsymbol{\Sigma}}_k$, $\widehat{\boldsymbol{V}}_k$

● Research Plan

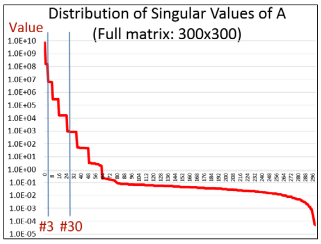
- Year 1 (FY2016): Algorithm development and testing environments deployment. (A prototyping)
- Year 2 (FY2017): Large-scale implementation and software integrations.
- Year 3: Auto-tuning of large-scale codes and tests of applications.

[By Ting-Li Chen, Su-Yun Huang, Hung Chen, David Chang, Chen-Yao Lin, and Weichung Wang]

● Test Problem



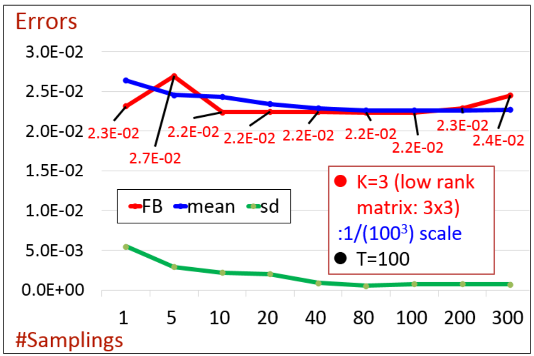
(By Prof. Ida at U. Tokyo)

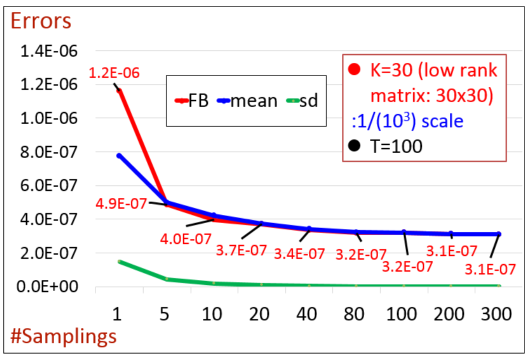


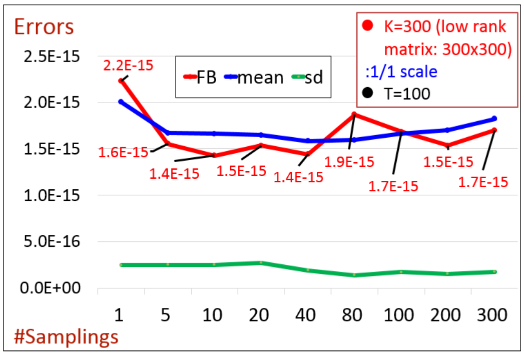
- Frobenius Norm (FB)
  $\|A - U\Lambda V^T\|_F / \|A\|_F$
- Mean
  $\sum_{i=1}^T (\|A - U\Lambda V^T\|_F / \|A\|_F)\_i \,/\, T$
- Sqrt (sd)
  $\sqrt{(\sum_{i=1}^T (\|A - U\Lambda V^T\|_F / \|A\|_F - mean)\_i\,)^2}$

Where *T* is the number of test trials of iSVD.

●Results (Accuracy of Low Rank Approximation by iSVD)







●⇒If you accept accuracy of O(2E-02), computation complexity is reduced to $O(1/100^3) = O(1/1{,}000{,}000)$