A Scalable Parallel Eigensolver for Large-scale Simulations on Post-peta Computing Environments

Tetsuya Sakurai (University of Tsukuba, JST CREST)

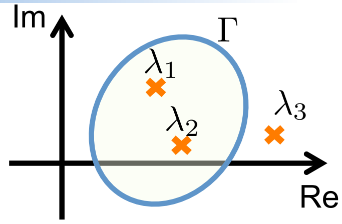
Joint work with
Yasunori Futamura
Akira Imakura
Yuto Inoue

Contents

- Parallel EIgenvalue Solver
  - Quadrature-type eigensolver
    - Sakurai-Sugiura method (SSM)
    - Stochastic estimation of eigenvalue density
    - Robust error resilient algorithm
- Deep Neural Networks
  - Computation of deep neural networks (DNNs) using nonnegative matrix factorizations (NMFs)
- Conclusions

Quadrature-type Methods

- Using Numerical Quadrature
  - GEPs (Generalized Eigenvalue Problems)
    - SS-H [S/Sugiura'2003]
      - Grid RPC(remote procedure call) (2004)
        - Coarse-grained communication, Hierarchical structure
    - SS-RR [S/Tadano'2007]
    - Block SS [Ikegami/S/Nagashima'2008]
    - FEAST [Polizzi'2009]
    - SS-Arnoldi [Imakura/Du/S'2014]



  - NEPs (Nonlinear Eigenvalue Problems)
    - SS-H [Asakura/S et al.'2009]
    - Beyn's method [Beyn'2012]
    - SS-RR [Yokota/S'2013]
    - SS-Arnoldi [Imakura/Du/S'2015]

Projection using Numerical Quadrature

- Contour integral is approximated by numerical quadrature

$$P_\Gamma = \sum_{\lambda_i \in G} P_i = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma (zB - A)^{-1} B \mathrm{d}z$$

$$\approx \sum_{j=1}^{N} w_j (z_j B - A)^{-1} B$$

z_j : quadrature point
w_j : quadrature weight

- Apply for vectors $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]$:

$$P_\Gamma V \approx \sum_{j=1}^{N} w_j \underline{(z_j B - A)^{-1} BV}$$

Systems of linear equations at shift points $z_1, \ldots, z_N$

$$(z_j B - A) Y_j = BV, \ j = 1, \ldots, N$$

Higher Order Complex Moments

- Obtain various linear combinations of projections using complex moments:

$$P_\Gamma = \sum_{\lambda_i \in G} P_i = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma (zB - A)^{-1} B \mathrm{d}z$$

$$P_\Gamma^{(k)} = \sum_{\lambda_i \in G} \lambda_i^k P_i = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma z^k (zB - A)^{-1} B \mathrm{d}z, \;\; k = 0, 1, \ldots$$

- Apply for vectors $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]$:

$$P_\Gamma^{(k)} V \approx S_k = \sum_{j=1}^{N} w_j z_j^k (z_j B - A)^{-1} BV, \;\; k = 0, 1, \ldots, M-1$$

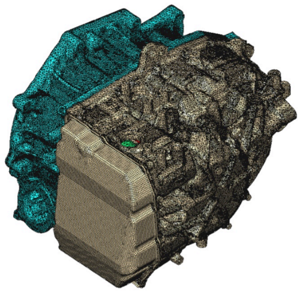
- Eigenpairs are extracted from $S = \overbrace{[S_0, S_1, \ldots, S_{M-1}]}^{L \times M}$
  - Hankel type, Rayleigh-Ritz type, etc.

Software z-Pares

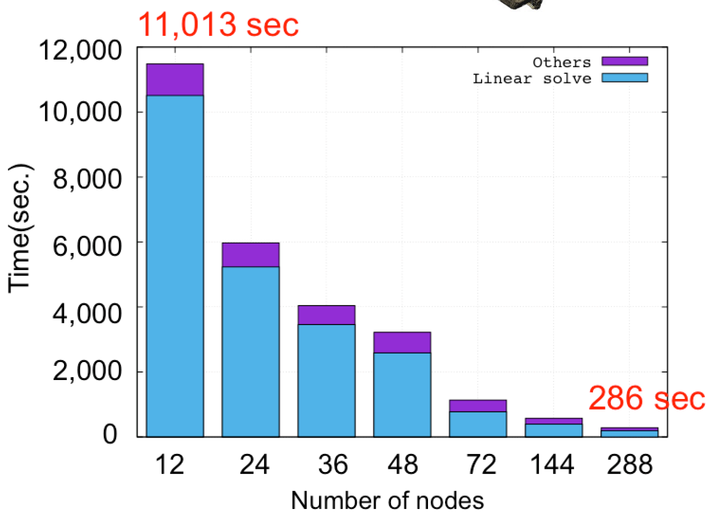
- Published software
  - z-Pares
    - Fortran95, MPI
    - For large-scale distributed parallel computing
  - CISS
    - SLEPc / PETSc
    - For evaluating efficiency of the algorithm in distributed parallel computing
  - SSEIG
    - MATLAB
    - For evaluating efficiency of the algorithm
- Available:
  - http://zpares.cs.tsukuba.ac.jp/

Numerical Example: Transmission Design

- Application: Automatic transmission design
  - Matrices are derived by FEM
    - Frequency range: 0 ~ 6,000 Hz
    - 16,747,926 DOF, 916 eigenpairs
  - Test environment: COMA @Univ. of Tsukuba



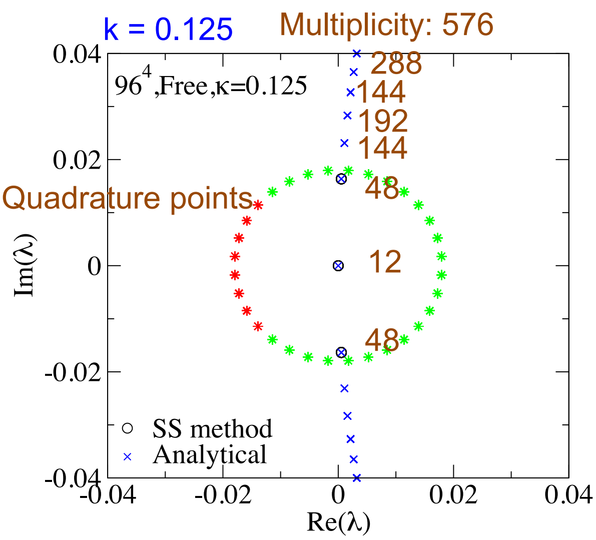
  - Solvers
    - z-Pares (SSM)
      - #contour paths: 4
      - #quadrature points: N = 16
    - Linear solver: MUMPS

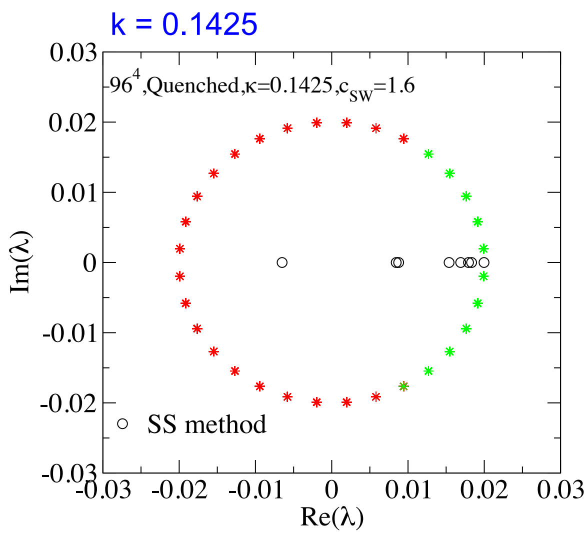


[Ide, Toda, Futamura, S '2016]

Numerical Example: Lattice QCD

- O(a)-improved Wilson-Dirac operator (Suno, Kuramashi, S, et al.)
  Matrix dim.: 1,019,215,872 (non-Hermitian)
  Test environment: 16,384 nodes of the K-Computer
  Linear solver: BiCGStab



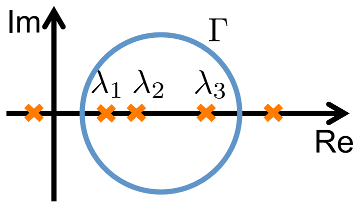


Stochastic Estimation of Eigenvalue Distribution

Eigenvalue Count in a Given Domain

- The number of eigenvalues m in Γ is given by

$$m = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma \mathrm{tr}((zB - A)^{-1}B)\mathrm{d}z$$



  - Approximate contour integral of a trace of inverse matrix by[1), 2)]

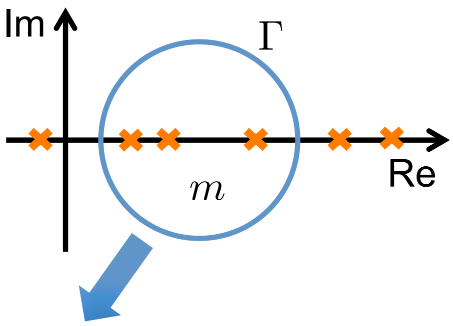
$$m \approx \sum_{j=1}^{N} w_j \left(\frac{1}{L} \sum_{\ell=1}^{L} \boldsymbol{v}_\ell^{\mathrm{T}} (z_j B - A)^{-1} B \boldsymbol{v}_\ell \right)$$

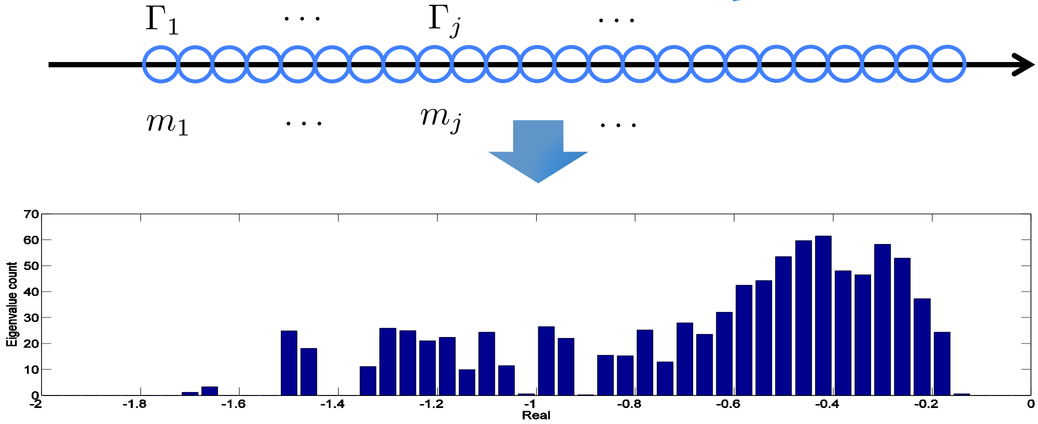
where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L$ are *L* sample vectors.

[Futamura, Tadano, S '2010], [Maeda, Futamura, Imakura, S '2016]

Estimation of Eigenvalue Density

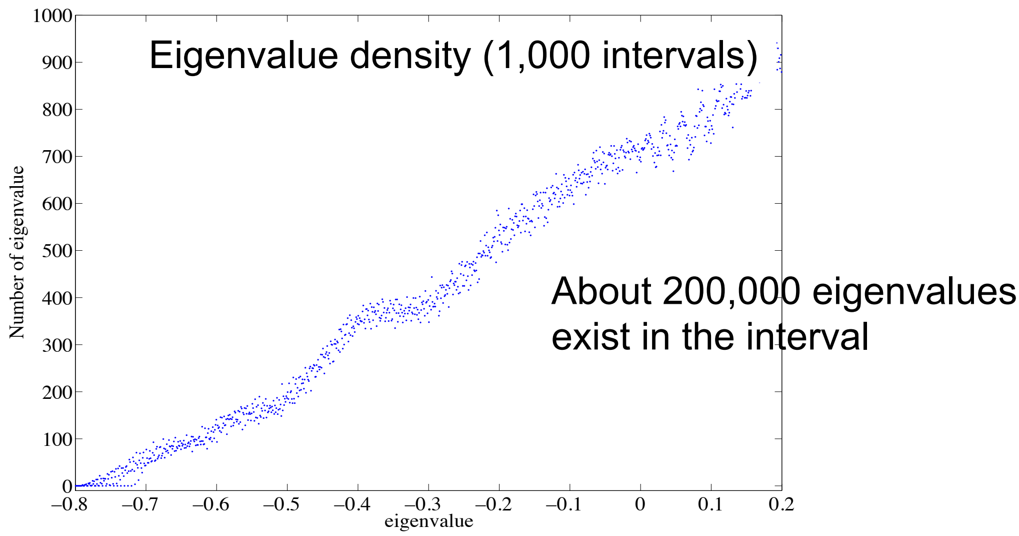
- Estimate the eigenvalue density
  - Put sub-intervals on a target interval





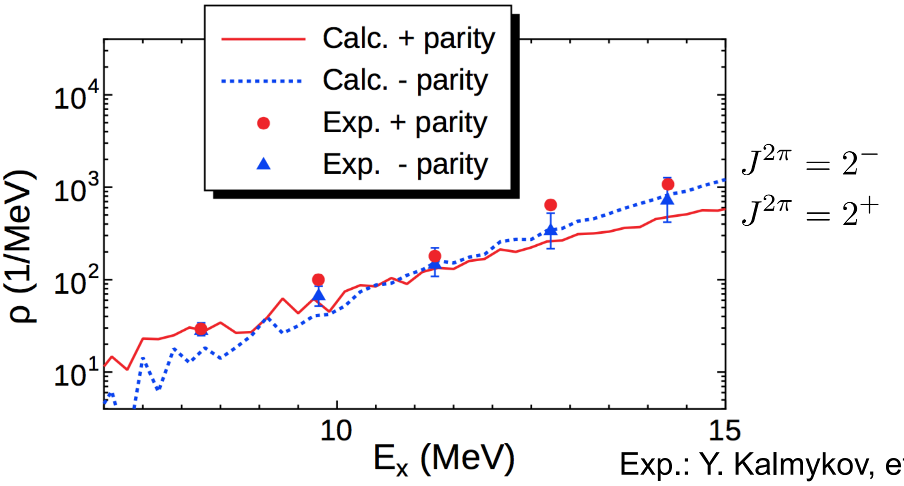
Numerical Example: Density Functional Theory

- RSDFT SiNW 107,292 atoms (RSDFT by Iwata et al.) Eigenvalue density at initial status
  - Matrix dimension: 64,700,000
  - 10,800 nodes of the K-Computer, 11,890 sec



Numerical Example: Shell Model Nuclear Level Density

- Shell Model Code: Kshell (Shimizu, et al.)
  - Spin-dependent level density of ^{58}Ni
    - Matrix dim.: 15 billion
    - 2,304 nodes of the K-Computer, 24 hours



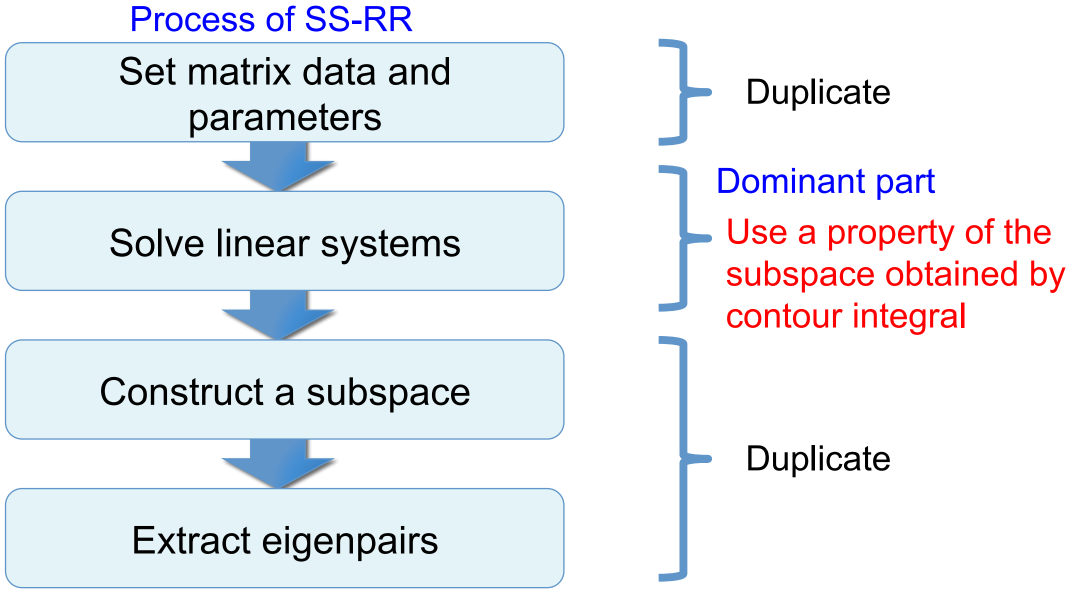
Exp.: Y. Kalmykov, et al. (2007)

[Shimizu, Utsuno, Futamura, S '2016]

Fault Tolerance

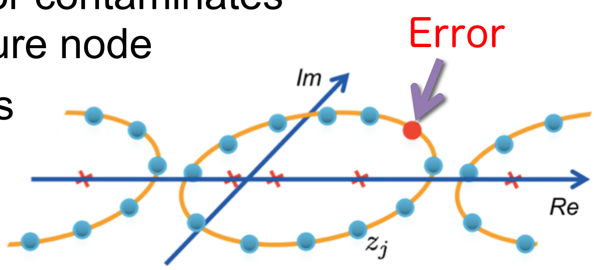
Robust Error Resilient Algorithm

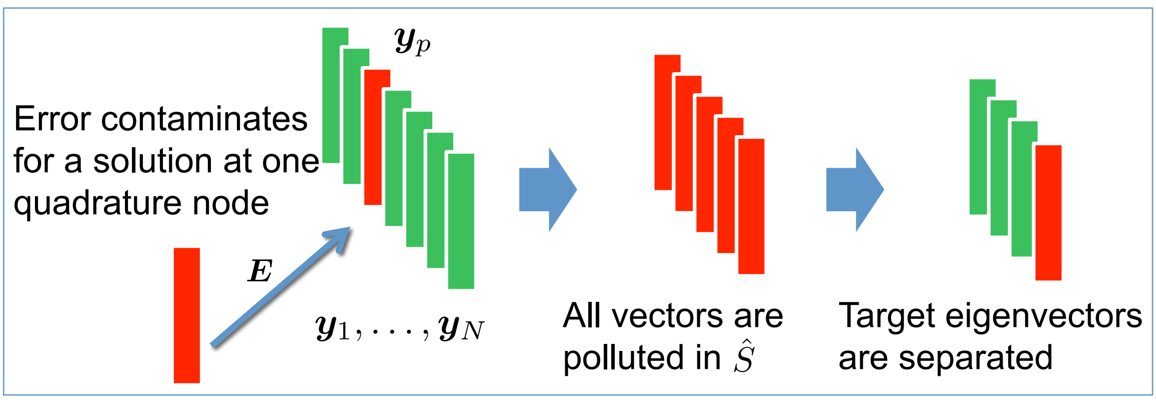
- Algorithm level fault tolerance
  - Dominant part for computational cost and memory requirement is solving systems of linear equations



Error Contamination and Separation

- Consider a case that silent error contaminates in the calculation at a quadrature node
  - Numerical error contaminates
    - Calculation is performed without detecting error





Tetsuya Sakurai @CCS.Tsukuba May 13, 2016

Accuracy analysis of SS-RR with error

- We consider the case that an error E contaminates in the solution $Y_{j'}$ on the j'-th quadrature node.

$$Y_{j'} = (z_{j'}B - A)^{-1}BV + E$$

- Since

$$\hat{S}'_k = \hat{S}_k + w_{j'} z_{j'}^k E$$

we have

$$\hat{S}' = [\hat{S}'_0, \hat{S}'_1, \ldots, \hat{S}'_{M-1}] = \hat{S} + \hat{E}$$

where

$$\hat{E} = [w_{j'}E, w_{j'}z_{j'}E, \ldots, w_{j'}z_{j'}^{M-1}E].$$

Accuracy analysis of SS-RR with error

P : Orthogonal Projection on $\mathrm{span}(\hat{S}')$

$A_{\mathcal{P}}, B_{\mathcal{P}} : A_{\mathcal{P}} = PAP,\ B_{\mathcal{P}} = PBP$

$(\lambda_i, \boldsymbol{x}_i)$: Eigen pairs s.t. $|f(\lambda_i)| \geq |f(\lambda_{i+1})|$

$f(\lambda)$: Filter function for an eigensubspace defined by numerical quadrature

Theorem

$$\|(A_{\mathcal{P}} - \lambda_i B_{\mathcal{P}})\boldsymbol{x}_i\|_2 \leq c_i \left| \frac{f(\lambda_{LM-L+1})}{f(\lambda_i)} \right|$$

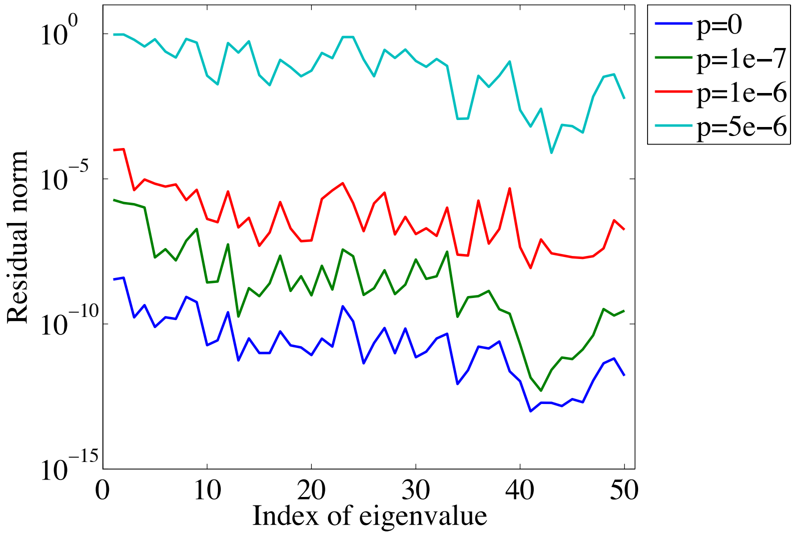
were c_i is a constant that depends on the input matrix V and the coefficient matrices A and B.

[Imakura, Du, S '2016]

Preliminary results

Generalized eigenproblem: BCSST12, Size: 1473
#quadrature points: N = 32

Bit-flip errors are injected to matrix data before LU factorization.



Probability	# of total bit flips
p = 0	0
p = 1e-7	19
p = 1e-6	148
p = 5e-6	699

Working with Teranishi (Sandia NL) and Yamazaki (U. Tennessee)

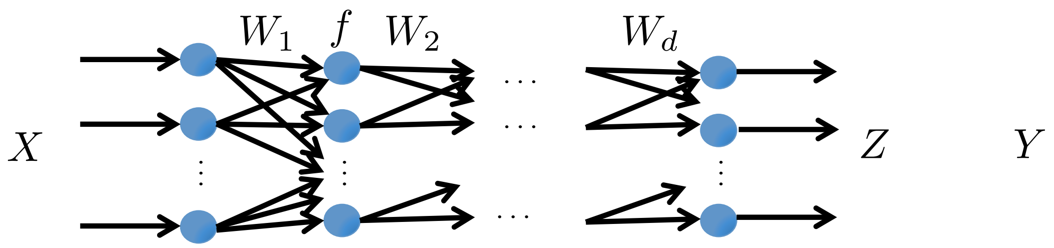
Computation of deep neural networks using nonnegative matrix factorizations

Computation of Deep Neural Networks (DNNs) using Nonnegative Matrix Factorizations (NMFs)

- We consider a matrix representation of the form

$$Z = W_d\, f(W_{d-1}\, f(W_{d-2} \cdots f(W_2\, f(W_1 X)) \cdots))$$

for a model of deep neural networks



where the activation function $f(\cdot)$ is set as the rectified linear units (ReLU).

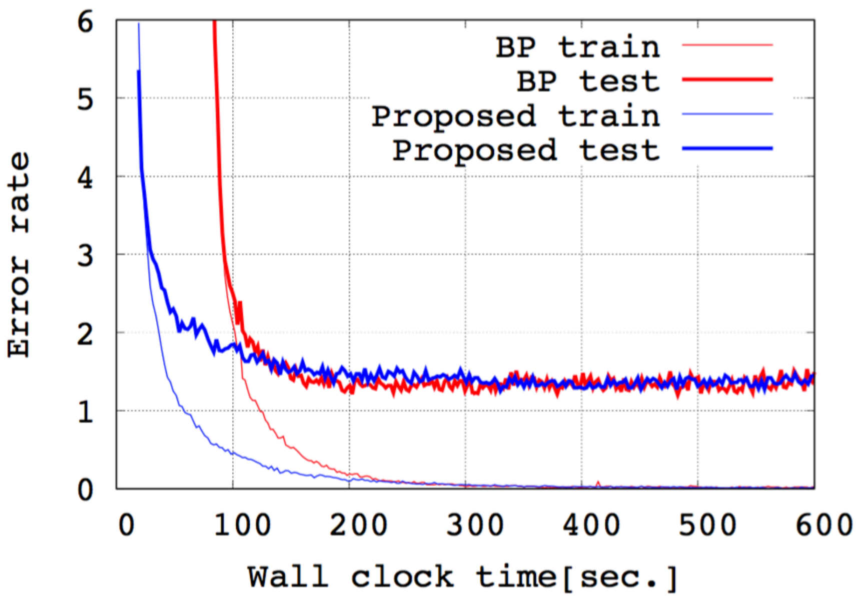
- Computation of fully-connected DNNs without backpropagation
  - Two types of semi-NMFs → weight matrices of DNNs.
  - a NMF → a stacked autoencoder

Numerical Example: MNIST

- Proposed method was implemented in MATLAB and backpropagation(BP) was implemented using TensorFlow.

MNIST
Dataset of 28x28 gray-scale handwritten digits

60,000 training images
10,000 test images

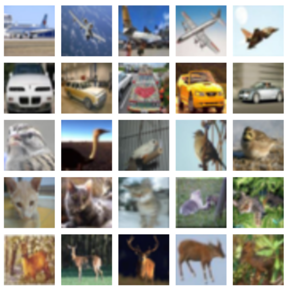


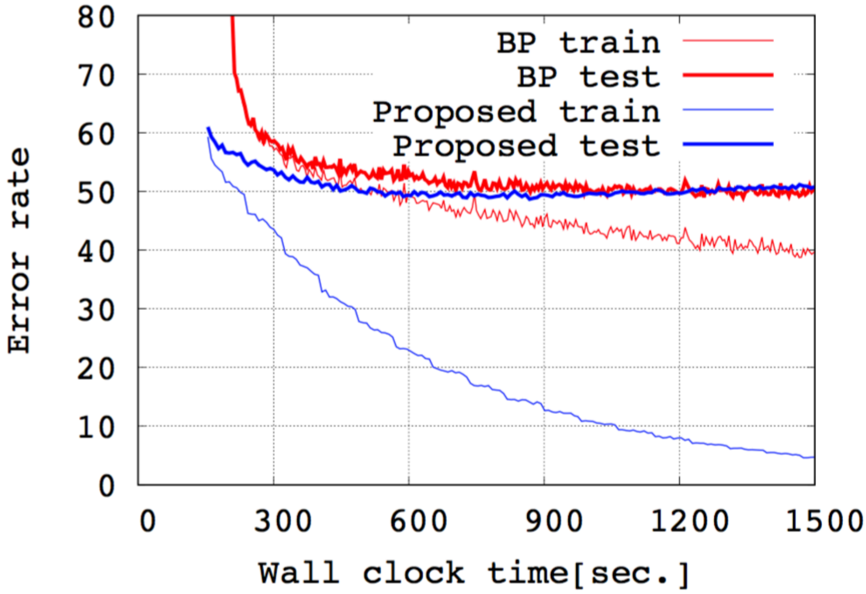
[1000-500] hidden units

Numerical Example: CIFAR10

CIFAR-10
Dataset of 32x32 color images in 10 classes

50000 training images
10000 test images





[1000-500] hidden units

Conclusions

- Scalable Algorithm: SS method
  - Quadrature-type eigensolver
    - Highly scalable method
- Stochastic estimation of eigenvalue density
  - Estimation of eigenvalue density
- Fault tolerance
  - Robust error resilient algorithm
- Computation of DNNs using NMFs
  - Alternative approach to DNNs without using BP

References

[1] T. Sakurai, H. Sugiura, A projection method for generalized eigenvalue problems using numerical integration, J. Comput. Appl. Math. 159, 119-128, 2003.

[2] T. Sakurai, H. Tadano, CIRR: a Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems, Hokkaido Math. J. 36, 745-757, 2007.

[3] I. Ikegami, T. Sakurai, U. Nagashima, A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method,CS-TR-08-13, Tsukuba, 2008.

[4] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, K. Kimura, A numerical method for nonlinear eigenvalue problems using contour integrals, JSIAM Lett. 1, 52-55, 2009.

[5] Y. Futamura, H. Tadano, and T. Sakurai, Parallel stochastic estimation method of eigenvalue distribution, JSIAM Letters 2, 127-130, 2010.

[6] Y. Futamura, T. Sakurai, S. Furuya and J. Iwata, Efficient algorithm for linear systems arising in solutions of eigenproblems and its application to electronic-structure calculations, LNCS 7851, 226-235, 2012.

[7] I. Yamazaki, T. Ikegami, H. Tadano, T. Sakurai, Performance comparison of parallel eigensolvers based on a contour integral method and a Lanczos method, Parallel Computing 39, 280–290, 2013.

[8] T. Yano, Y. Futamura, T. Sakurai, Multi-GPU scalable implementation of a contour-integral-based eigensolver for real symmetric dense generalized eigenvalue problems, Proc. 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2013), 121-127, 2013.

[9] A. Imakura, L. Du, T. Sakurai, A block Arnoldi-type contour integral spectral projection method for solving generalized eigenvalue problems, Appl. Math. Lett. 32, 22-27, 2014.

[10] Y. Maeda, Y. Futamura, A. Imakura and T. Sakurai, Filter analysis for the stochastic estimation of eigenvalue counts, JSIAM Letters 7, 53-56, 2015.

[11] A. Imakura, L. Du and T. Sakurai, Error bounds of Rayleigh-Ritz type contour integral-based eigensolver for solving generalized eigenvalue problems, Numerical Algorithms 71 103-120, 2016.

[12] T. Ide, K. Toda, Y. Futamura, T. Sakurai, Highly parallel computation of eigenvalue analysis in vibration for automatic transmission using Sakurai-Sugiura method and K-Computer, SAE Technical Paper, 2016-01-1378, 2016.

[13] N. Shimizu, Y. Utsuno, Y. Futamura, T. Sakurai, Stochastic estimation of nuclear level density in the nuclear shell model: An application to parity-dependent level density in ^{58}Ni, Phys. Lett. B 753, 13-17, 2016.

[14] T. Sakurai, A. Imakura, Y. Inoue, Y. Futamura, Alternating optimization method based on nonnegative matrix factorizations for deep neural networks (submitted).