A contour integral-based parallel eigensolver with higher complex moments

Tetsuya Sakurai (University of Tsukuba, JST CREST)
Yasunori Futamura (University of Tsukuba)
Akira Imakura (University of Tsukuba)

Contents

- Introduction
  - Approach for parallel scalability
- Quadrature-type Eigensolver
  - SS method
  - Software package z-Pares
- Estimation of Eigenvalue Density
  - Stochastic estimation of eigenvalue count
- Numerical examples
- Conclusions

Partial Eigenvalue Problems

- We consider an eigenvalue problem

$$T(\lambda)\boldsymbol{x} = \boldsymbol{0},$$

where $T(\lambda)$ is a matrix valued function, and $\boldsymbol{x}$ is a nonzero vector of dimension n.

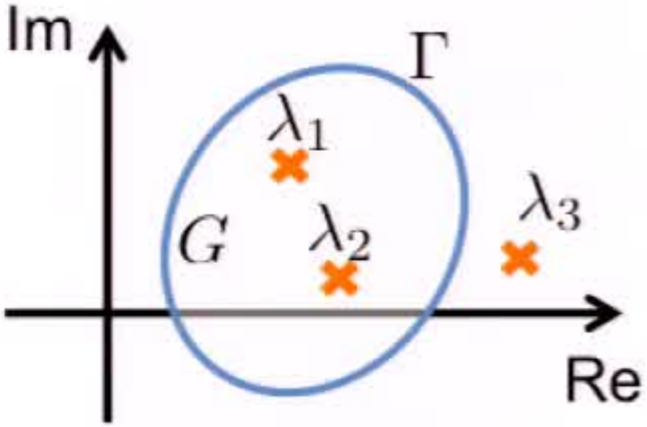
Ex. Standard EP (SEP): $T(\lambda) = A - \lambda I$

Generalized EP (GEP): $T(\lambda) = A - \lambda B$

Nonlinear EP (NEP): $T(\lambda) = A_0 + \lambda A_1 + \lambda^2 A_2$

$T(\lambda) = A_0 + \lambda A_1 + e^{\lambda} A_2$

- Find all eigenvalues located inside Γ
  - Interior eigenvalue problems
  - Large-scale sparse matrices

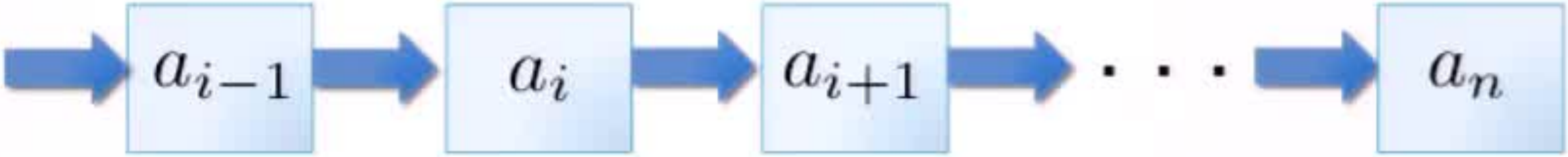


In this talk, we mainly explain GEP: $A\boldsymbol{x} = \lambda B\boldsymbol{x}$

Our Approach for Parallel Scalability

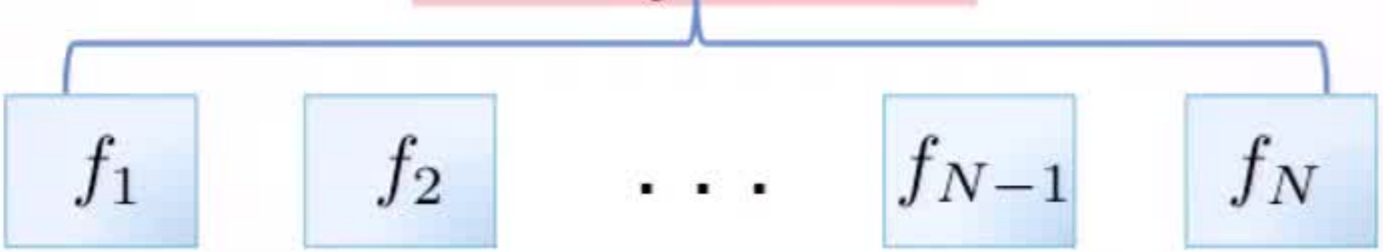
- Avoid recurrence calculations in eigenvalue computation
  - Algorithms described by recurrence relations: (ex. Krylov methods):

$$a_{i+1} = b_i\, a_i + c_i\, a_{i-1}$$



  - Algorithms without recurrence calculations: (ex. Numerical quadrature):

$$I_N = \sum_{j=1}^{N} w_j f_j$$

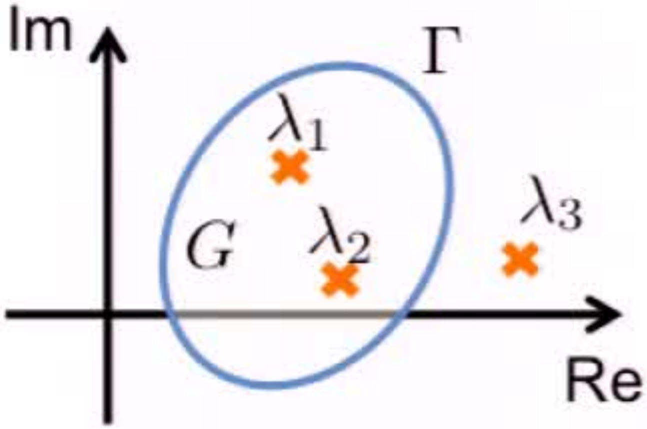


Quadrature-type Eigensolver

Spectral Projection by Contour Integral

- Contour integral for a rational function

$$\frac{1}{2\pi \mathrm{i}}\oint_\Gamma \sum_{i=1}^{n} \frac{\nu_i}{z-\lambda_i}\mathrm{d}z = \sum_{\lambda_i \in G} \nu_i$$



- Spectral decomposition of $(zB-A)^{-1}B$:

$$(zB-A)^{-1}B = \sum_{i=1}^{n} \frac{P_i}{z-\lambda_i}$$

λ_i : eigenvalue, P_i : spectral projection with respect to λ_i

(for simplicity, we consider the case that λ_i is simple)

Localization of spectral decomposition using contour integral:

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

Approximation by *N*-point 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 $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]$:

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

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

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

Extension of Subspace with Higher Order Complex Moments

- Obtain various linear combinations of projections using complex moments:

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

⬇

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

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

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

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

SS method with Rayleigh-Ritz procedure

- Algorithm of SS-RR

Appropriate path and parameters

0: Set contour path and parameters

Most time consuming part

1: Solve linear systems

$$\text{Solve } (z_j B - A)Y_j = BV \text{ for } Y_j,\ j = 1, 2, \ldots, N$$

2: Construct a subspace

$$\text{Compute } \hat{S}_k = \sum_{j=1}^{N} w_j z_j^k Y_j,\ k = 0, 1, \ldots, M-1$$

$$\text{Set } \hat{S} = [\hat{S}_0, \hat{S}_1, \ldots, \hat{S}_{M-1}]$$

3: Extract eigenpairs

$$\text{Compute low-rank approx. s.t. } \hat{S} = U\Sigma W^T \approx U_1 \Sigma_1 W_1^T$$

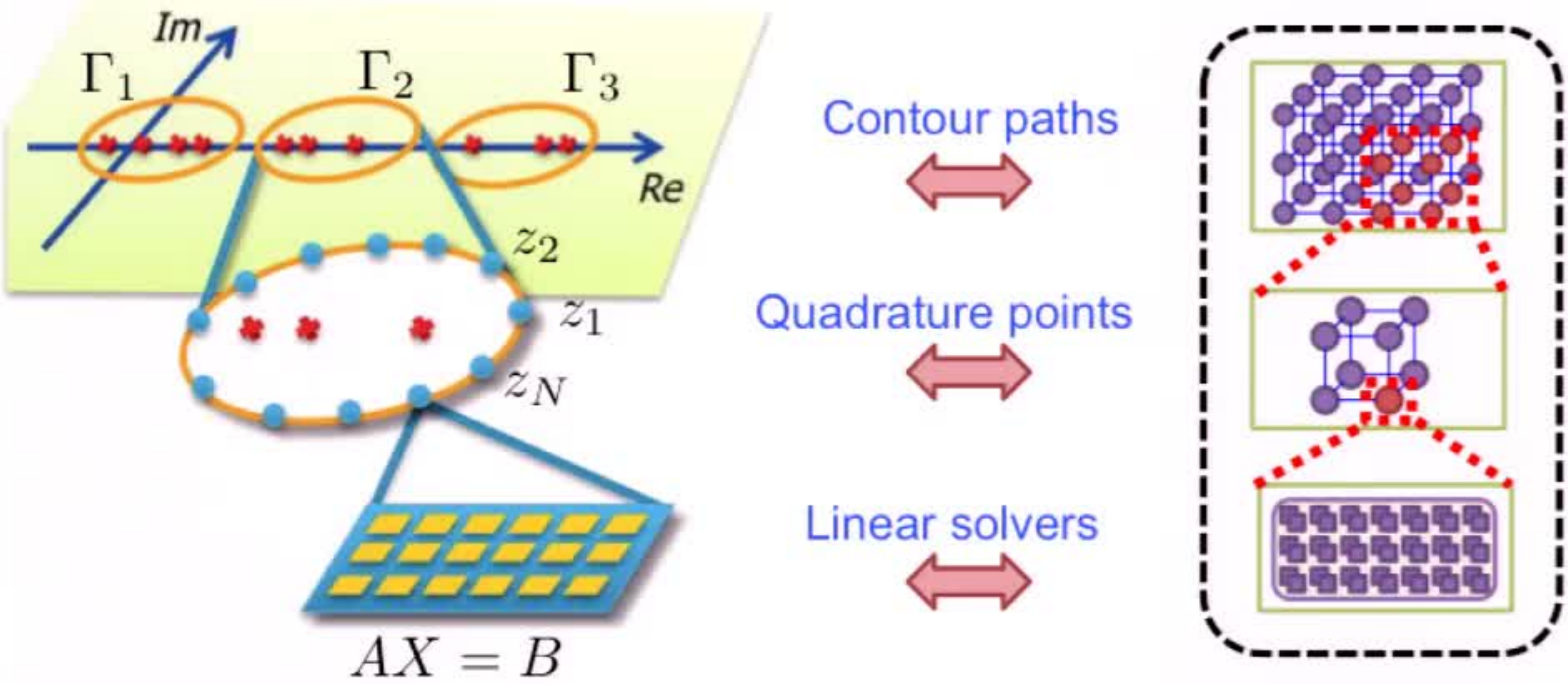
$$\text{Compute Ritz paris with } \hat{A} = U_1^H A U_1 \text{ and } \hat{B} = U_1^H B U_1$$

Hierarchical Parallel Structure

- Computing resources are assigned according to a hierarchical structure of the algorithm

Hierarchical structure of the algorithm

Hierarchical structure of a machine



Contour paths

Quadrature points

Linear solvers

Software

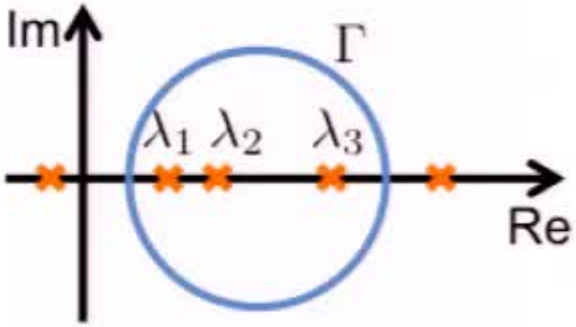
- Published software
  - z-Pares
    - Fortran95, MPI
    - For large-scale distributed parallel computing
  - CISS
    - SLEPc / PETSc
    - For evaluating efficiency of the algorithm in distributed parallel computing
  - SSEIG
    - MATLAB
    - For evaluating efficiency of the algorithm
- Available: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

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)]

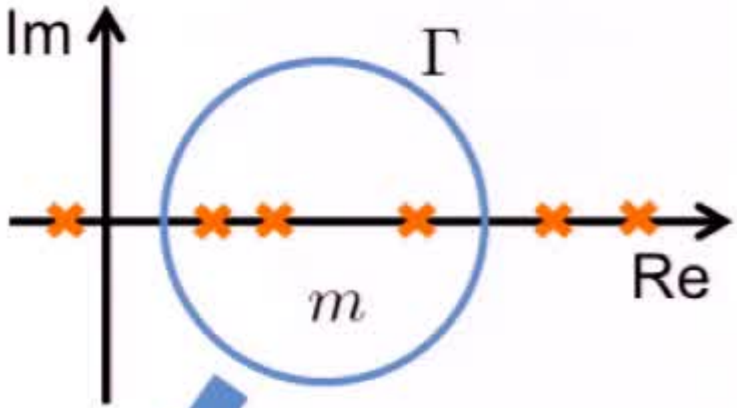
$$\boxed{m \approx \sum_{j=1}^{N} w_j \left(\frac{1}{L} \sum_{\ell=1}^{L} \boldsymbol{v}_\ell^{\mathrm{T}} (z_j B - A)^{-1} B \boldsymbol{v}_\ell \right)}$$

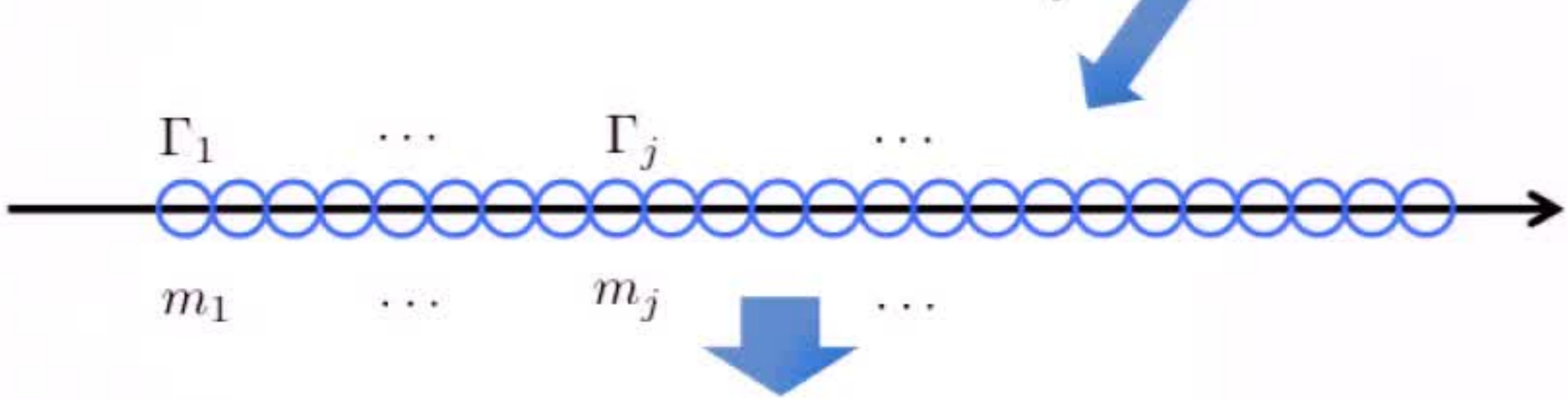
where $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L$ are L sample vectors.

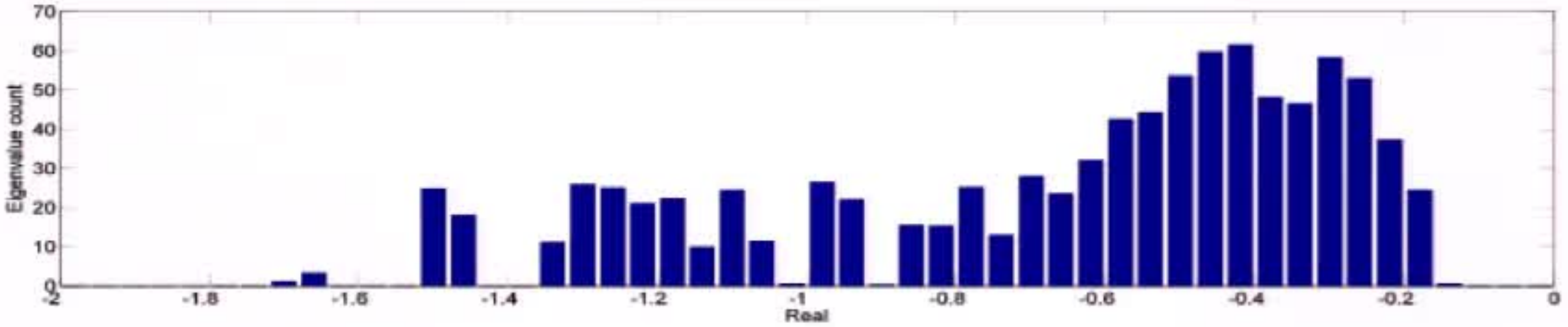
1) Futamura, H. Tadano, and T. Sakurai, JSIAM Letters 2, 127-130 (2010).
2) Y. Maeda, Y. Futamura, A. Imakura and T. Sakurai, JSIAM Letters 7, 53-56 (2015).

Estimation of Eigenvalue Density

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







Numerical Examples

Numerical Example: Density Functional Theory

- Band structure calculation with real space density functional theory
  (RSDFT by Iwata et al.)
  - An interior standard eigenvalue problem (SEP)
    - Eigenvalues around the band gap
    - Matrices with several wave numbers:

$$A(k)\boldsymbol{u} = \lambda \boldsymbol{u},\ k = 0, \Delta k, 2\Delta k, \ldots$$

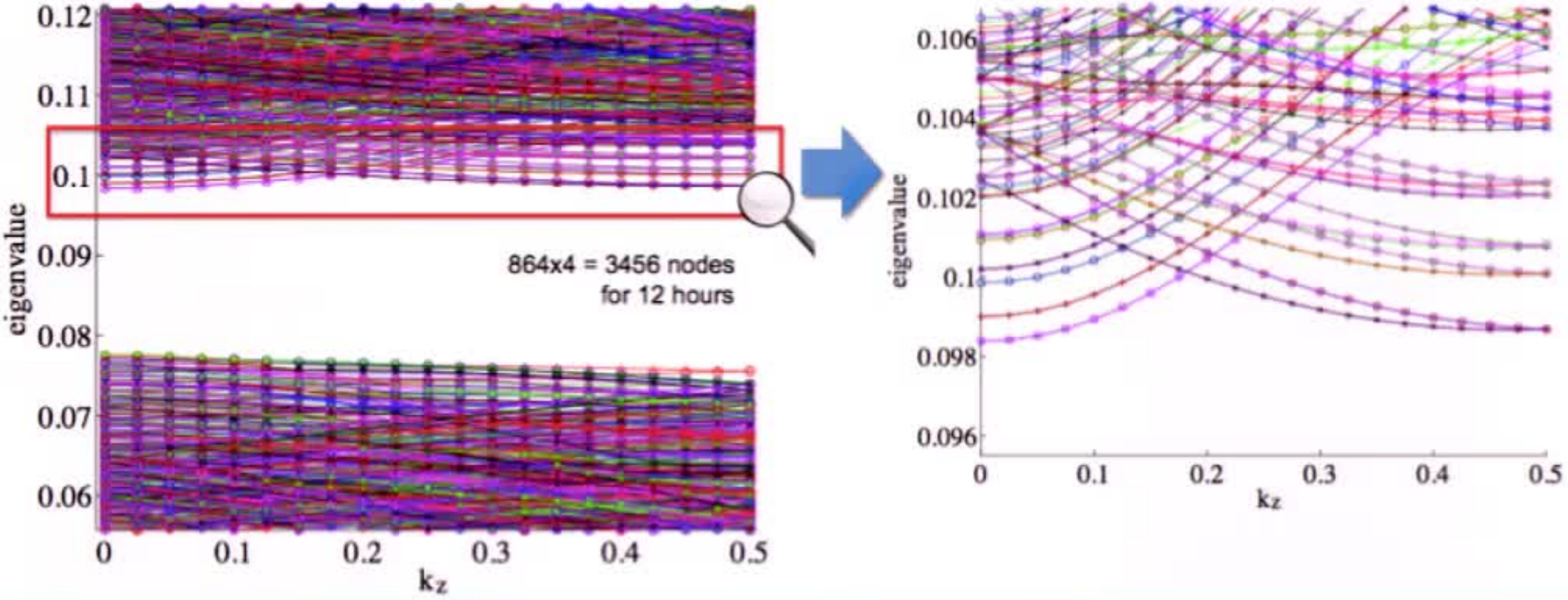
- Material: Silicon nanowire 9,924 atoms
  Matrix dim.: 8,719,488 (only mat-vec operation is given)

- Test environment: the K Computer (AICS, Japan)
  - 3,456 nodes (27,648 cores)

- Linear solver: Shifted Block CGrQ

Numerical Example : Density Functional Theory

- Band structure calculation for Silicon Nanowire 9,924 atoms
  - 21 problems are solved
  - Four intervals are set around the band gap
  - 3,456 nodes of the K Computer (about 12 hours)

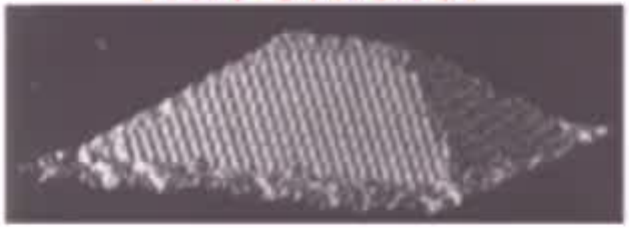


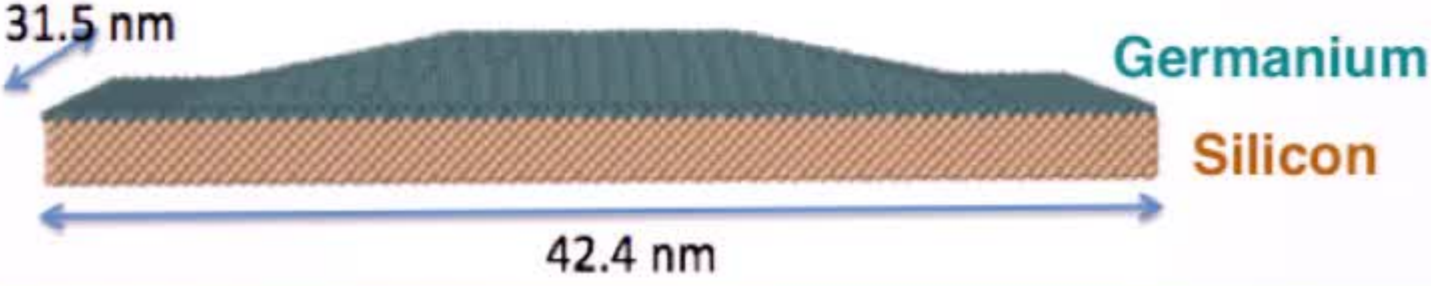
864x4 = 3456 nodes
for 12 hours

Numerical Example: Order-*N* DFT

- Order-*N* DFT code CONQUEST
  - SiGe hut cluster with 200,000 atoms by Nakata and Miyazaki
    - Matrix size : 778,292, NNZ : 13,247,248
    - GEP: 223 eigenpairs around HOMO-LUMO are computed
  - Test environment : COMA@Univ. of Tsukuba
  - Linear solver : MUMPS (sparse direct solver)

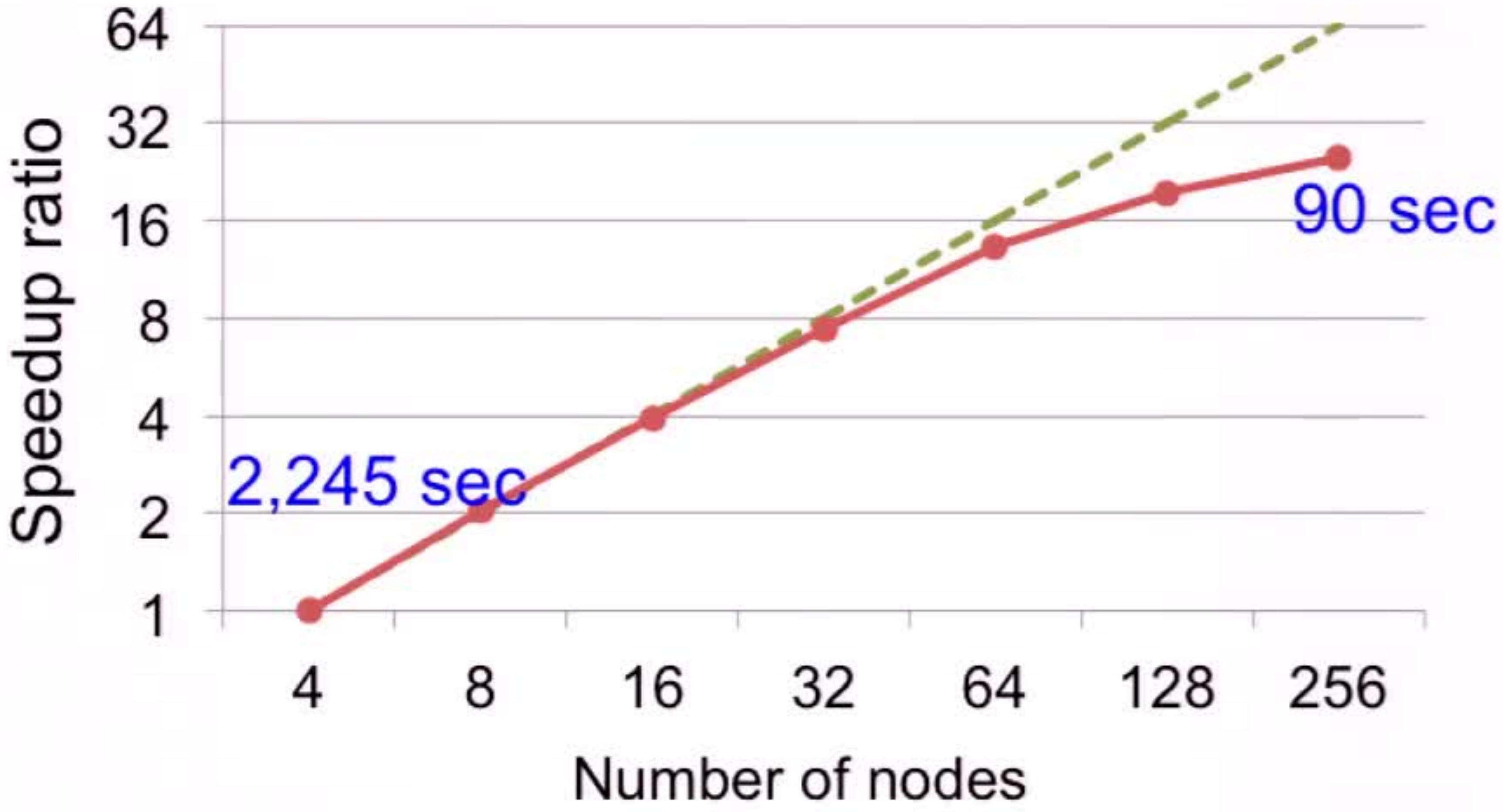
hut cluster





Numerical Example: Order-*N* DFT

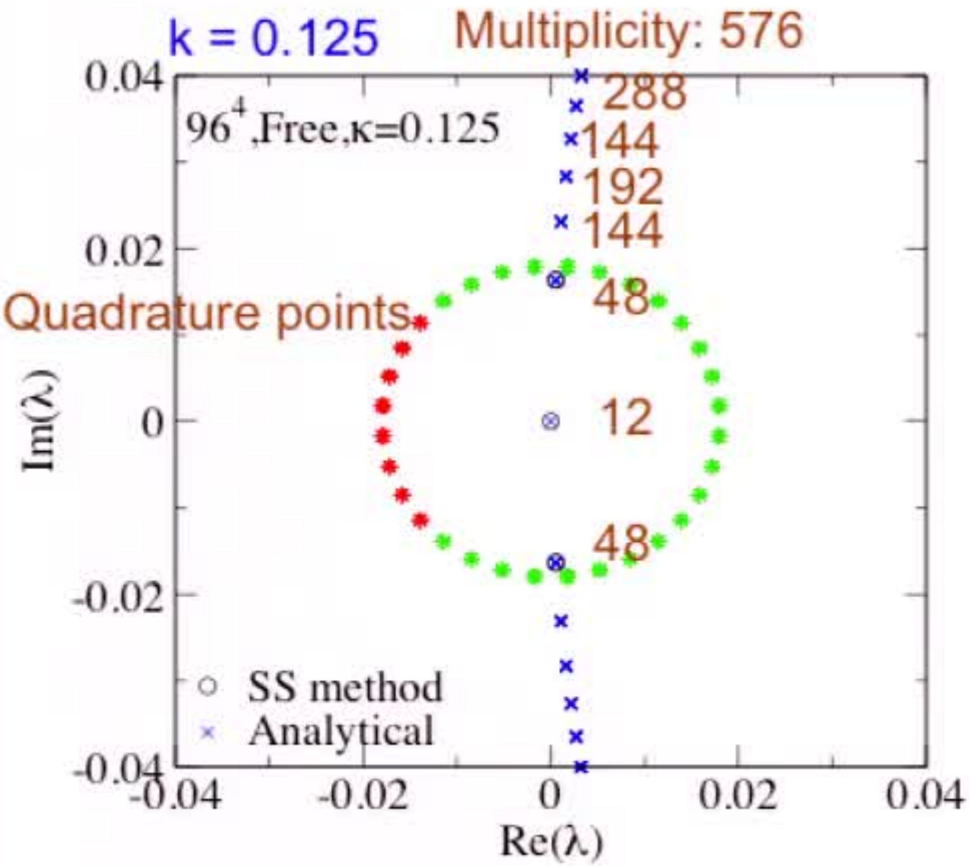
Application to order-*N* DFT code CONQUEST



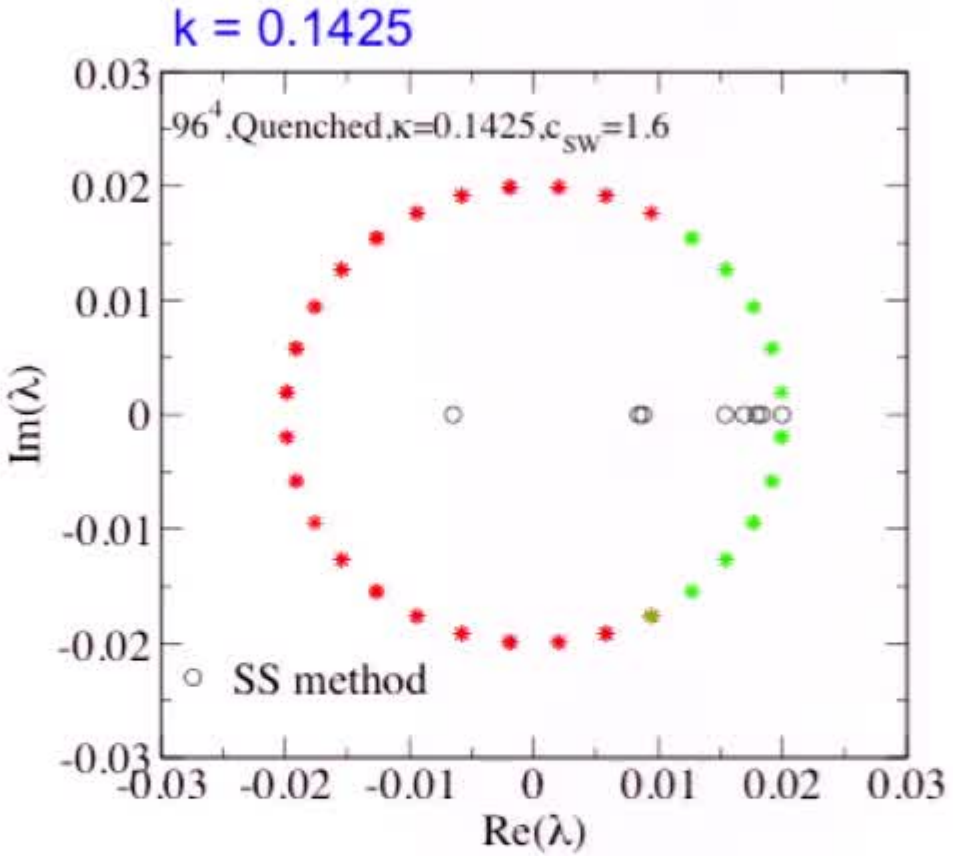
Numerical Examples: Lattice QCD

- 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

k = 0.125



k = 0.1425



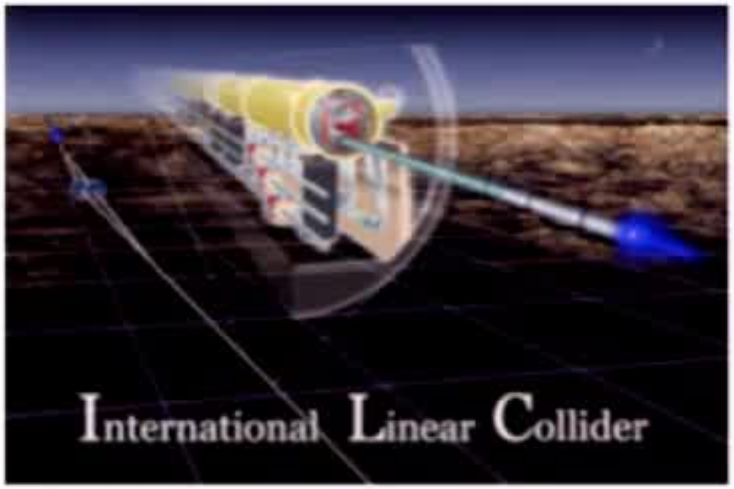
Numerical Example: Nonlinear Eigenvalue Problem

- Test problem:
  Simulation of the international linear collider

$$T(\lambda)\boldsymbol{x} = \mathbf{0}$$

$$T(\lambda) = K - \lambda^2 M + \mathrm{i}\sum_{j=1}^{t} \sqrt{\lambda^2 - \sigma_j^2} W_j,$$

where $t = 1$, $\sigma_1 = 0$.



http://www.linearcollider.org/

- Test environment:
  Cray-XT4 at NERSC @Berkeley
- Linear solver: SuperLU_DIST

Numerical Example: Nonlinear Eigenvalue Problem

- Strong scalability for nonlinear (NEP) case
  - Matrix size: 2,738,556

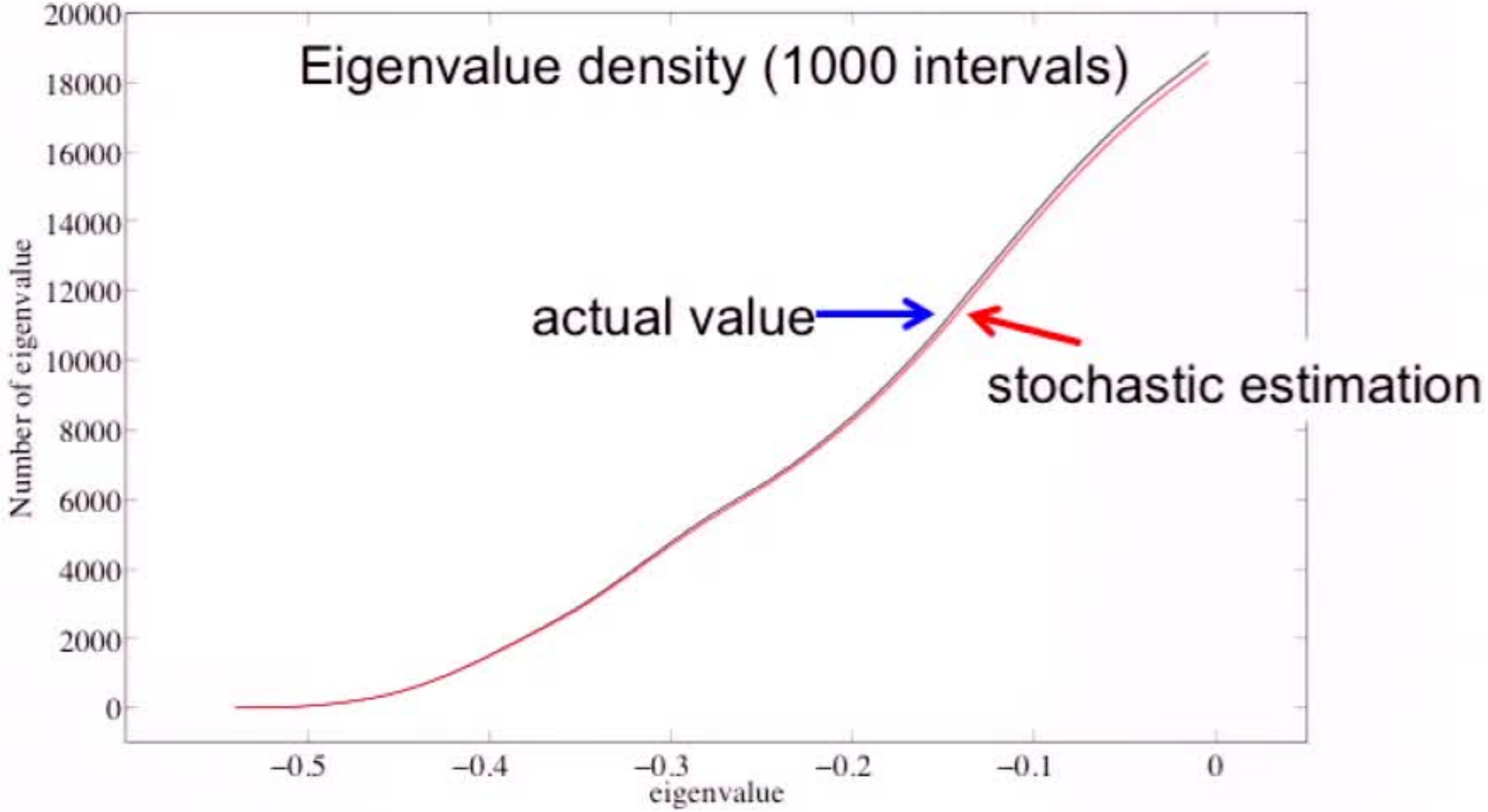
- Two contour paths are located.
  - The number of quadrature points is $N = 32$.
  - 64 linear systems are solved in total.

#cores	256	512	1024	2048
time(sec.)	2513	1273	661	334
speedup	-	1.97	1.93	1.92

[Yamazaki et al. '2013]

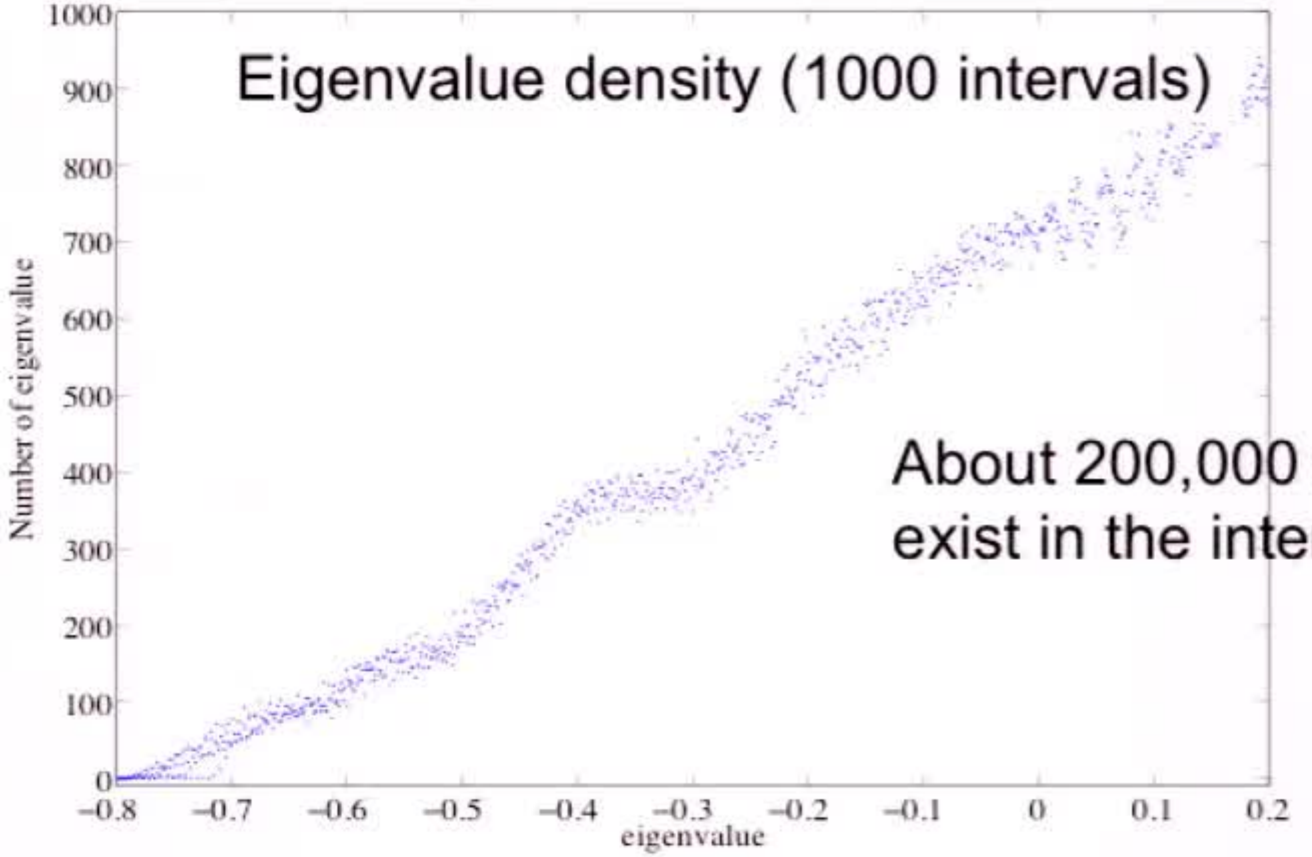
Numerical Example: Eigenvalue Density

- RSDFT SiNW 9,924 atoms
  Accumulated total from left



Numerical Example: Eigenvalue Density

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

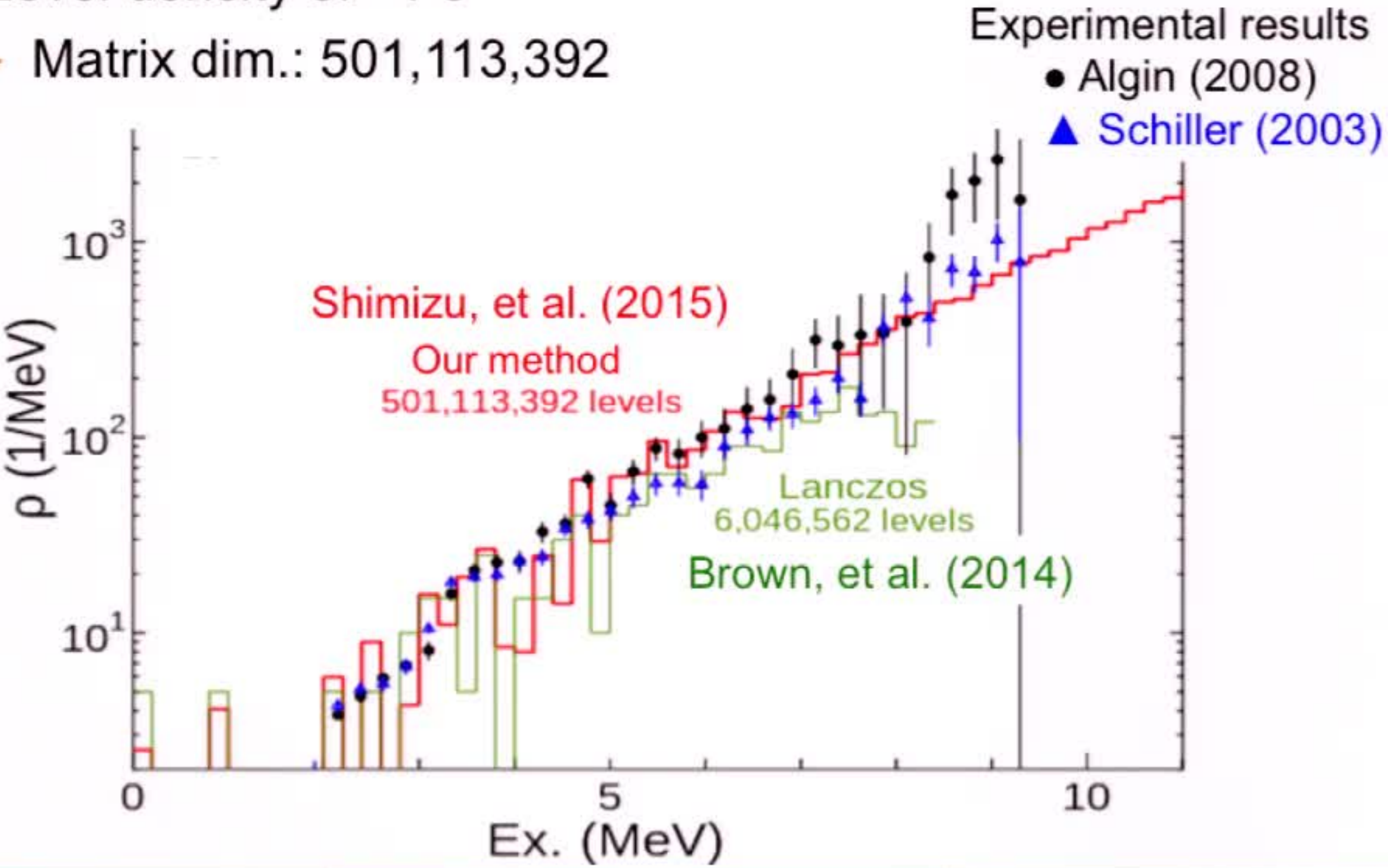


Eigenvalue density (1000 intervals)

About 200,000 eigenvalues exist in the interval

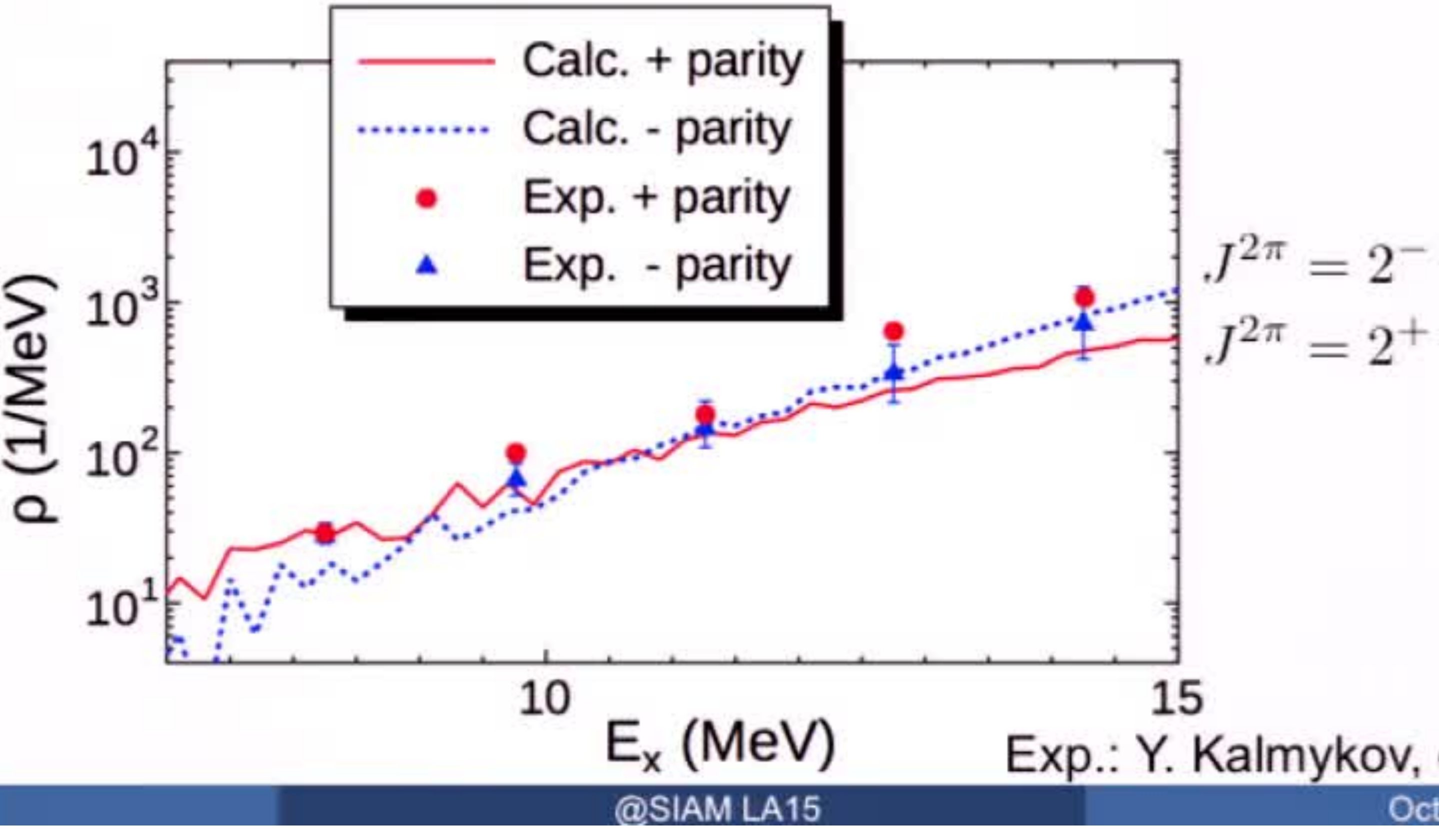
Numerical Example: Shell Model Nuclear Level Density

- Shell Model Code: Kshell (Shimizu, et al.)
  - Level density of ^{56}Fe
    - Matrix dim.: 501,113,392



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)