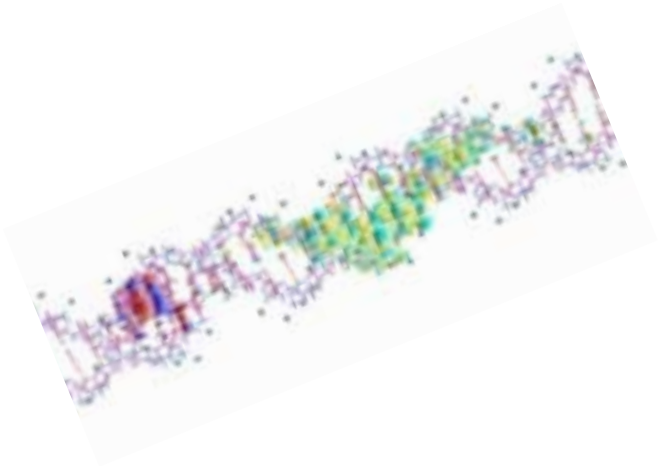
A parallel eigensolver using contour integration for generalized eigenvalue problems in molecular simulation

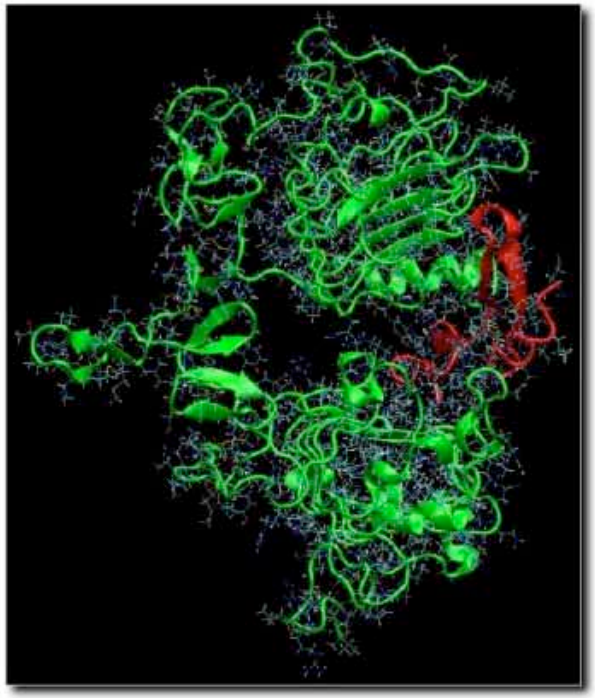
Tetsuya Sakurai (University of Tsukuba)
Hiroto Tadano (Kyoto University)
Umpei Nagashima (AIST)

Contents



Molecular Orbital Computation

Design of Anticancer Drugs



EGFR
(Epidermal Growth Factor Receptor)

Schrödinger Equation

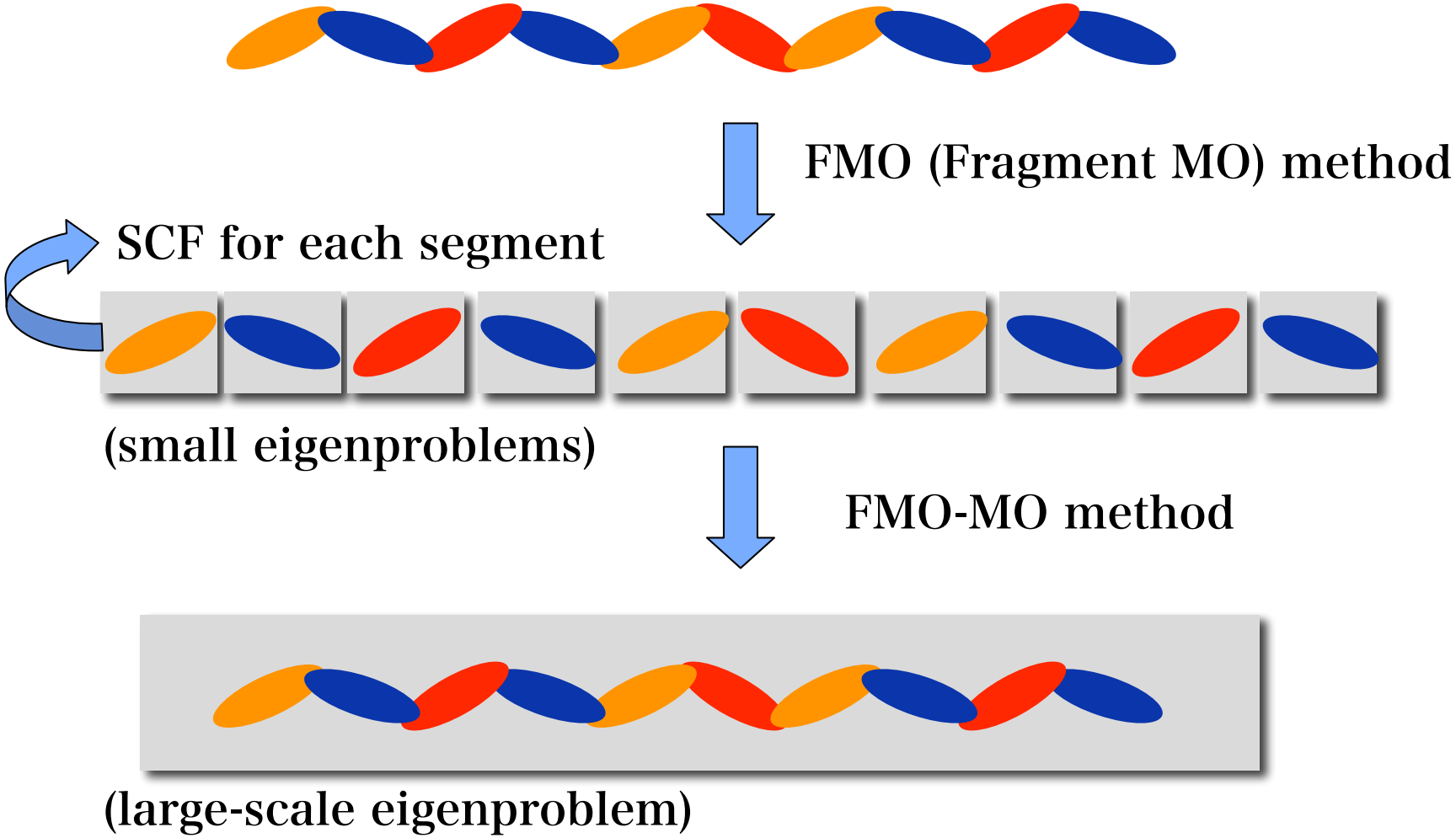
$$H\Psi = E\Psi$$

↓ Hartree-Fock approximation

Generalized Eigenvalue Problems

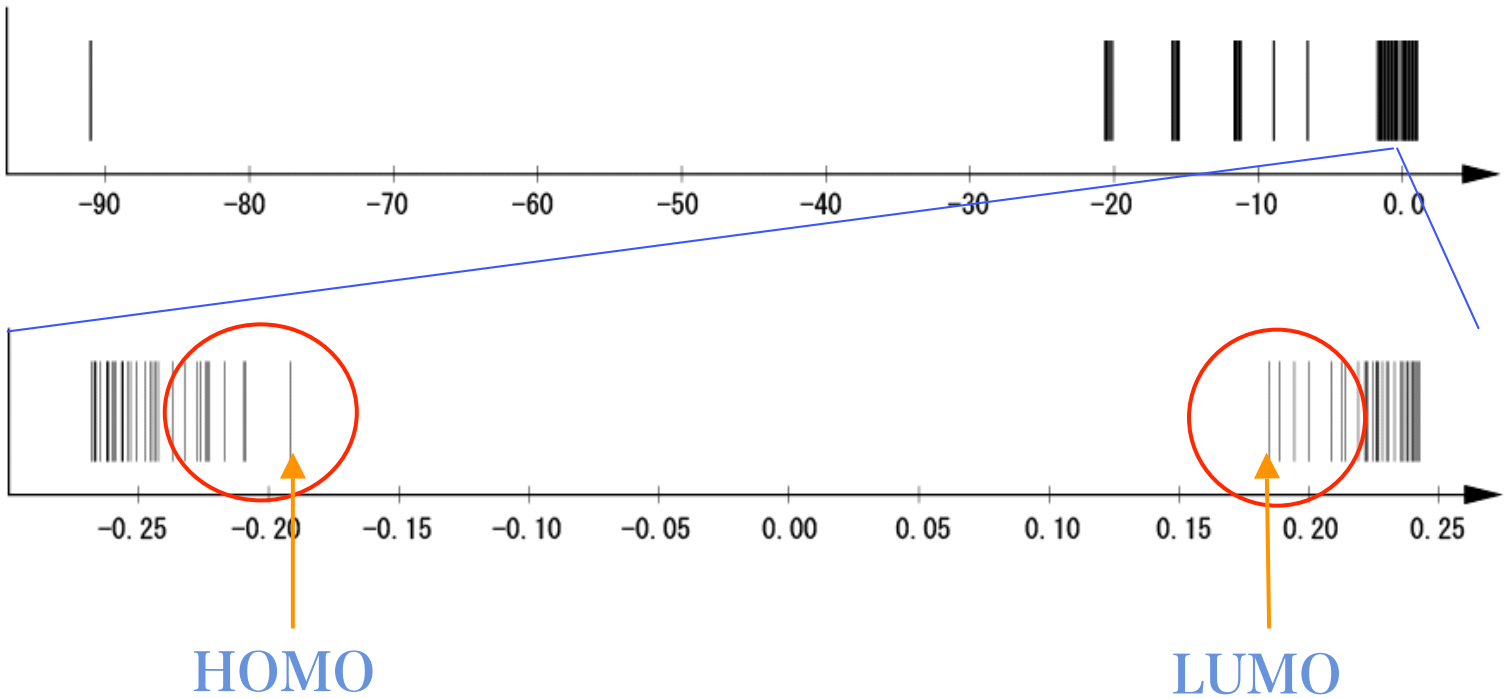
Matrix Generation

A large molecule is separated to small segments.



Required Orbitals

Energy state:



Eigenvectors related to chemical activities:

→ Interior eigenvalue problems

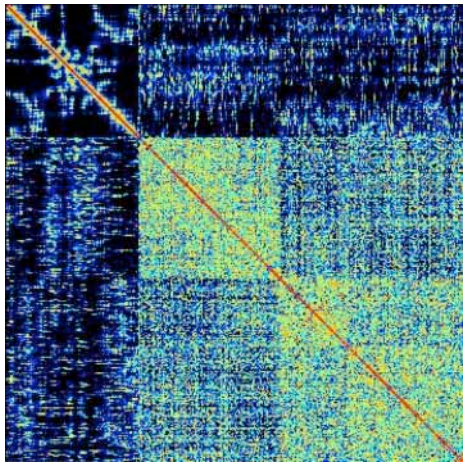
Matrix Properties

- The size of matrix: 2K ~ 200K
  The number of nonzero elements: 1M ~ 400M

  - relatively large number of nonzero elements

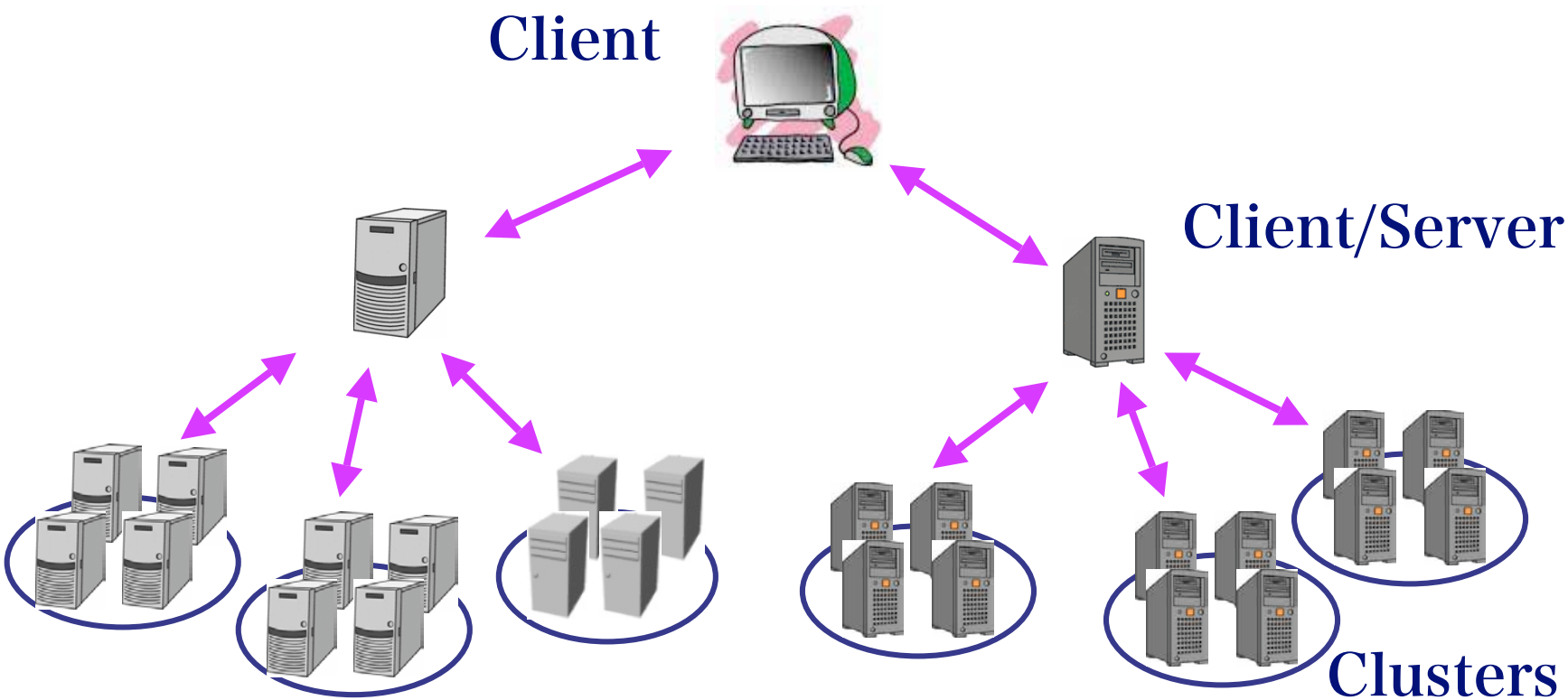
  - unstructured sparsity pattern

Fock matrix of
Lysozyme + H2O



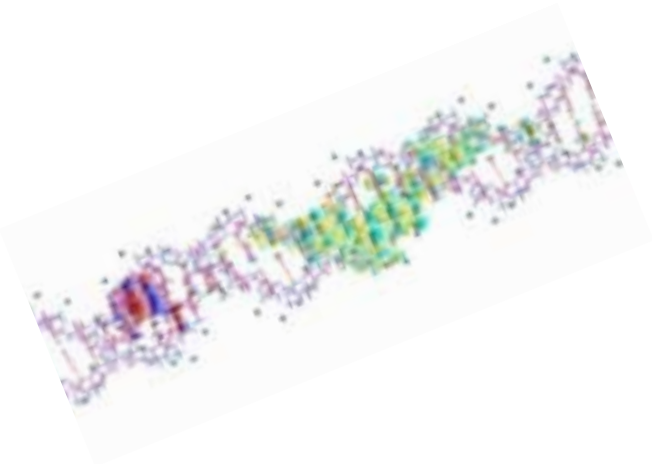
Computing Environment

FMO-MO method is suitable for GRID computing.



Highly parallelized eigensolver is required.

Contents



Generalized Eigenvalue Problem

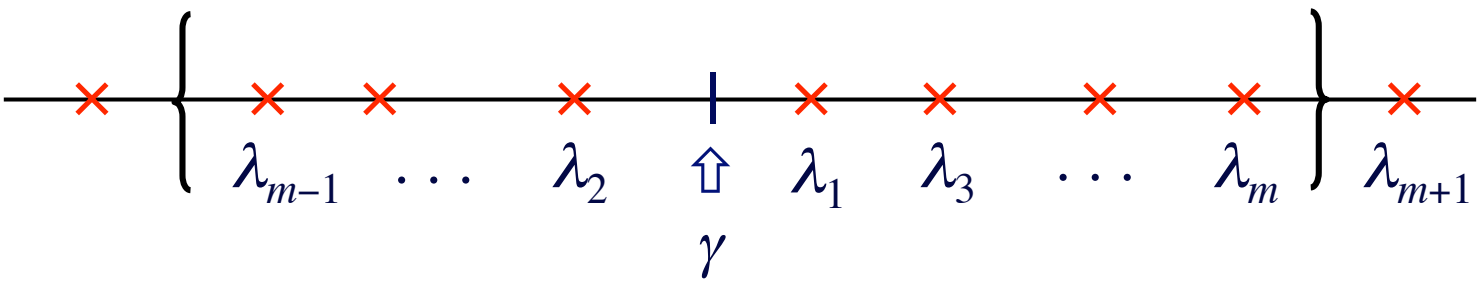
The generalized eigenvalue problem:

$$A\boldsymbol{x} = \lambda B\boldsymbol{x},$$

where $A, B \in \mathbb{R}^{n\times n}$, symmetric, and B is positive definite.

$(\lambda_j, \boldsymbol{u}_j)$: Eigenpair of the matrix pencil (A, B)

We find eigenpairs in a given interval:



Rayleigh-Ritz Procedure

Algorithm:

Inner Loop

1. Construct an orthonormal basis Q.
2. Form $A_Q = Q^{\mathrm{T}} A Q$ and $B_Q = Q^{\mathrm{T}} B Q$.
3. Compute the eigenpairs $(\theta_j, \boldsymbol{w}_j)$ $(1 \le j \le m)$ of (A_Q, B_Q).
4. Set $\boldsymbol{p}_j \leftarrow Q\boldsymbol{w}_j$, $j = 1, \ldots, m$.

Outer Loop

(A_Q, B_Q) : Projected pencil

θ_j : Ritz value

$\boldsymbol{p}_j$: Ritz vector

Contour Integral of Resolvent

To **avoid inner/outer loops**, we use a contour integral in **construction of a subspace**.

For a nonzero vector $\boldsymbol{v}$, let

$$\boldsymbol{s}_k := \frac{1}{2\pi \mathrm{i}} \int_\Gamma (z-\gamma)^k (zB - A)^{-1} B\boldsymbol{v} \mathrm{d}z,$$

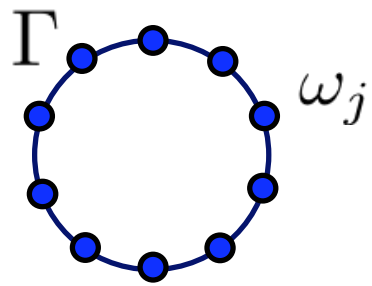
where $\Gamma \in \mathbb{C}$ is a Jordan curve that includes $\lambda_1, \ldots, \lambda_m$.

$$\boxed{\mathrm{span}(\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{m-1}) = \mathrm{span}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)}$$

[S and Tadano (2007)]

Approximation for Contour Integral

Γ : Circle with center γ and radius ρ



Equidistributed points on the circle:

$$\omega_j = \gamma + \rho e^{\frac{2\pi \mathrm{i}}{N}(j+1/2)}, \quad j = 0, 1, \ldots, N-1$$

$\boldsymbol{s}_k$ are approximated by the N-point trapezoidal rule:

$$\boldsymbol{s}_k \approx \frac{1}{N} \sum_{j=0}^{N-1} (\omega_j - \gamma)^{k+1} \boldsymbol{y}_j, \quad k = 0, \ldots, m-1$$

where

$$\boxed{\boldsymbol{y}_j = (\omega_j B - A)^{-1} \boldsymbol{v}, \quad j = 0, \ldots, N-1}$$

Contour Integral Rayleigh-Ritz Method

Algorithm of CIRR (Contour Integral Rayleigh-Ritz) method:

1. Set $\tau_j \leftarrow \exp(2\pi \mathrm{i}(j+1/2)/N)$, $j = 0, \ldots, N-1$.
2. Set $\omega_j \leftarrow \gamma + \rho\,\tau_j$, $j = 0, \ldots, N-1$.
3. Solve $(\omega_j B - A)\boldsymbol{y}_j = \boldsymbol{v}$ for $\boldsymbol{y}_j$, $j = 0, \ldots, N-1$.
4. Set $\hat{\boldsymbol{s}}_k \leftarrow \sum_{j=0}^{N-1} \tau_j^{k+1} \boldsymbol{y}_j$, $k = 0, \ldots, M-1$.
5. Construct an orthonormal basis Q from $\{\hat{\boldsymbol{s}}_0, \ldots, \hat{\boldsymbol{s}}_{M-1}\}$.
6. Form $A_Q = Q^{\mathrm{T}}(A - \gamma B)Q$ and $B_Q = Q^{\mathrm{T}} B Q$.
7. Compute the eigenpairs $(\theta_j, \boldsymbol{x}_j)$ $(1 \le j \le M)$ of (A_Q, B_Q).
8. Set $\hat{\lambda}_j \leftarrow \theta_j + \gamma$, $j = 1, \ldots, M$.
9. Set $\hat{\boldsymbol{u}}_j \leftarrow Q\boldsymbol{x}_j$, $j = 1, \ldots, M$.

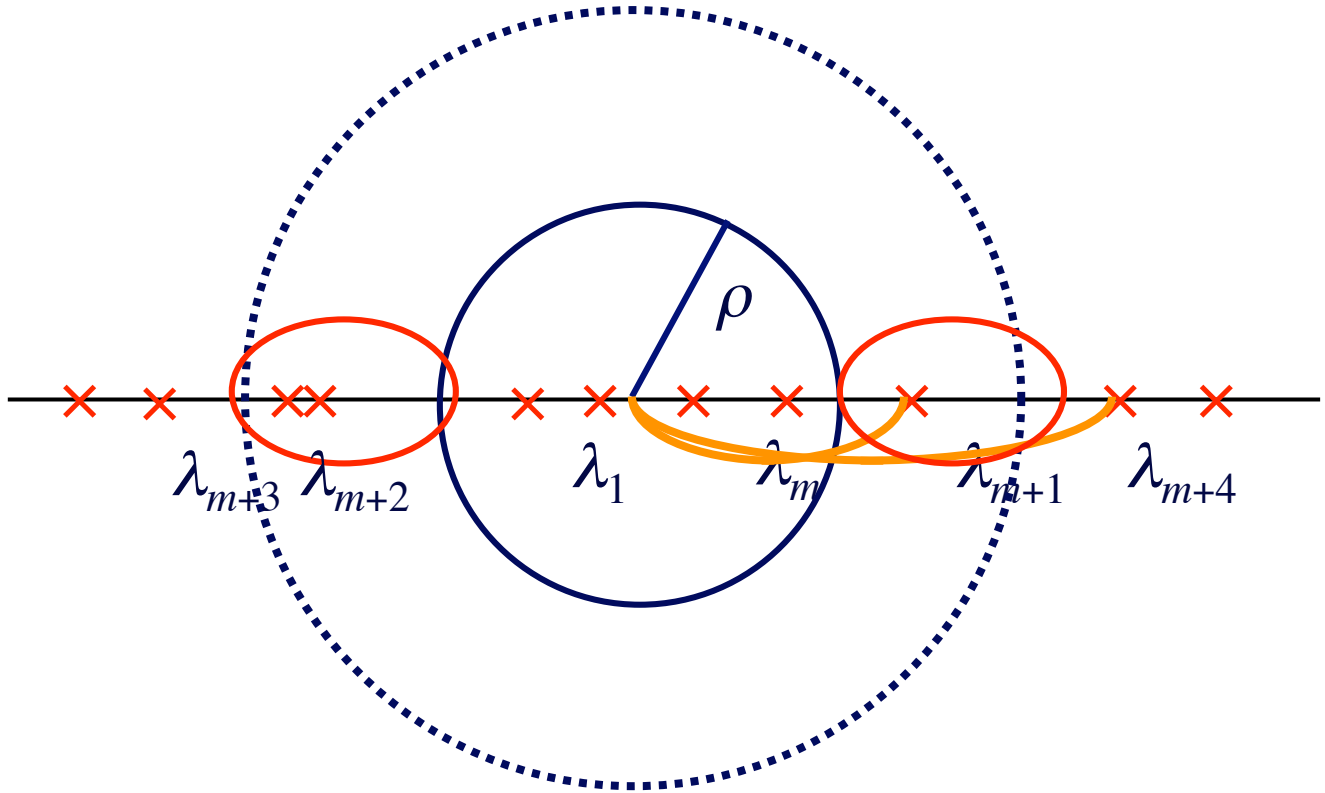
Construct a subspace

Rayleigh-Ritz procedure

Influence of Quadrature Error

Let $$\eta := \min_{j>m} \frac{\lambda_j - \gamma}{\rho} \qquad \eta \to \min_{j>m'} \frac{\lambda_j - \gamma}{\rho}, \quad m' > m$$

Then $$|\hat{\lambda}_j - \lambda_j| = O(\eta^{-N}), \quad 1 \le j \le m$$



Block Method

Block variant is also obtained by using a matrix

$$V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L]$$

instead of a vector $\boldsymbol{v}$.

$$\boldsymbol{s}_k := \frac{1}{2\pi \mathrm{i}} \int_\Gamma (z-\gamma)^k (zB - A)^{-1} B \boldsymbol{v} \mathrm{d}z$$

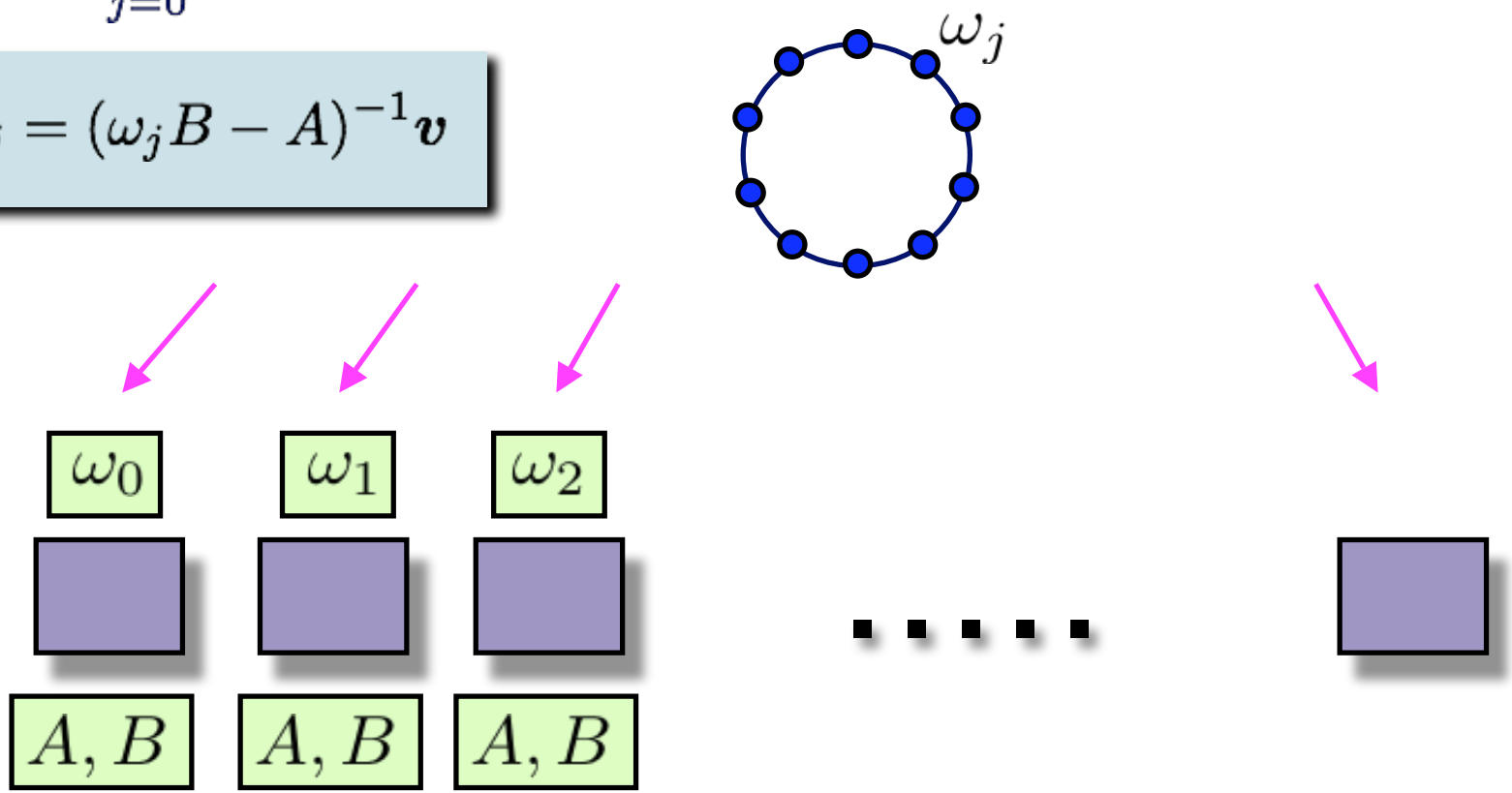
$$\Downarrow$$

$$S_k := \frac{1}{2\pi \mathrm{i}} \int_\Gamma (z-\gamma)^k (zB - A)^{-1} B V \mathrm{d}z$$

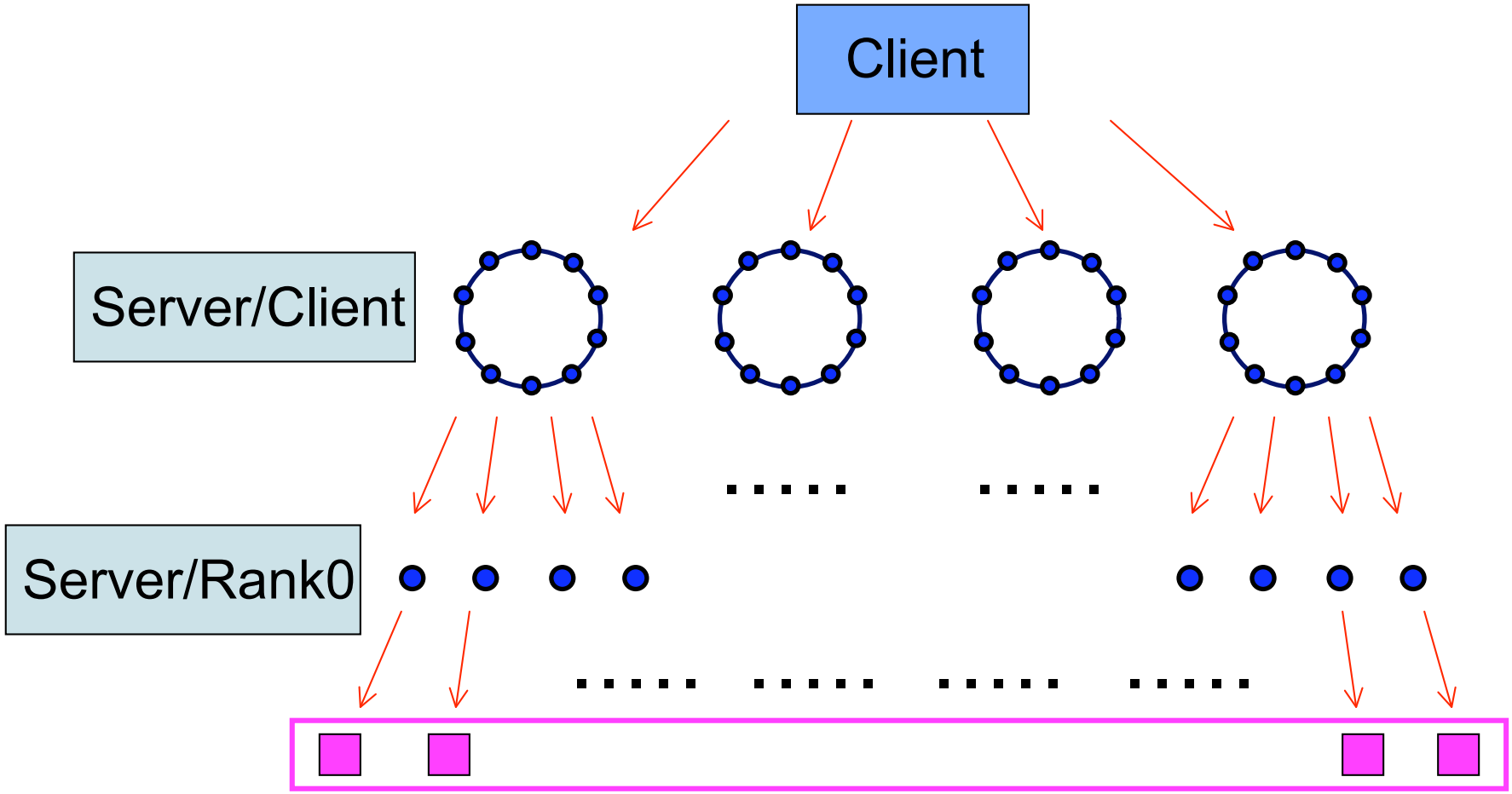
Parallel Implementation

$$\boldsymbol{s}_k \approx \frac{1}{N} \sum_{j=0}^{N-1} (\omega_j - \gamma)^{k+1} \boldsymbol{y}_j$$

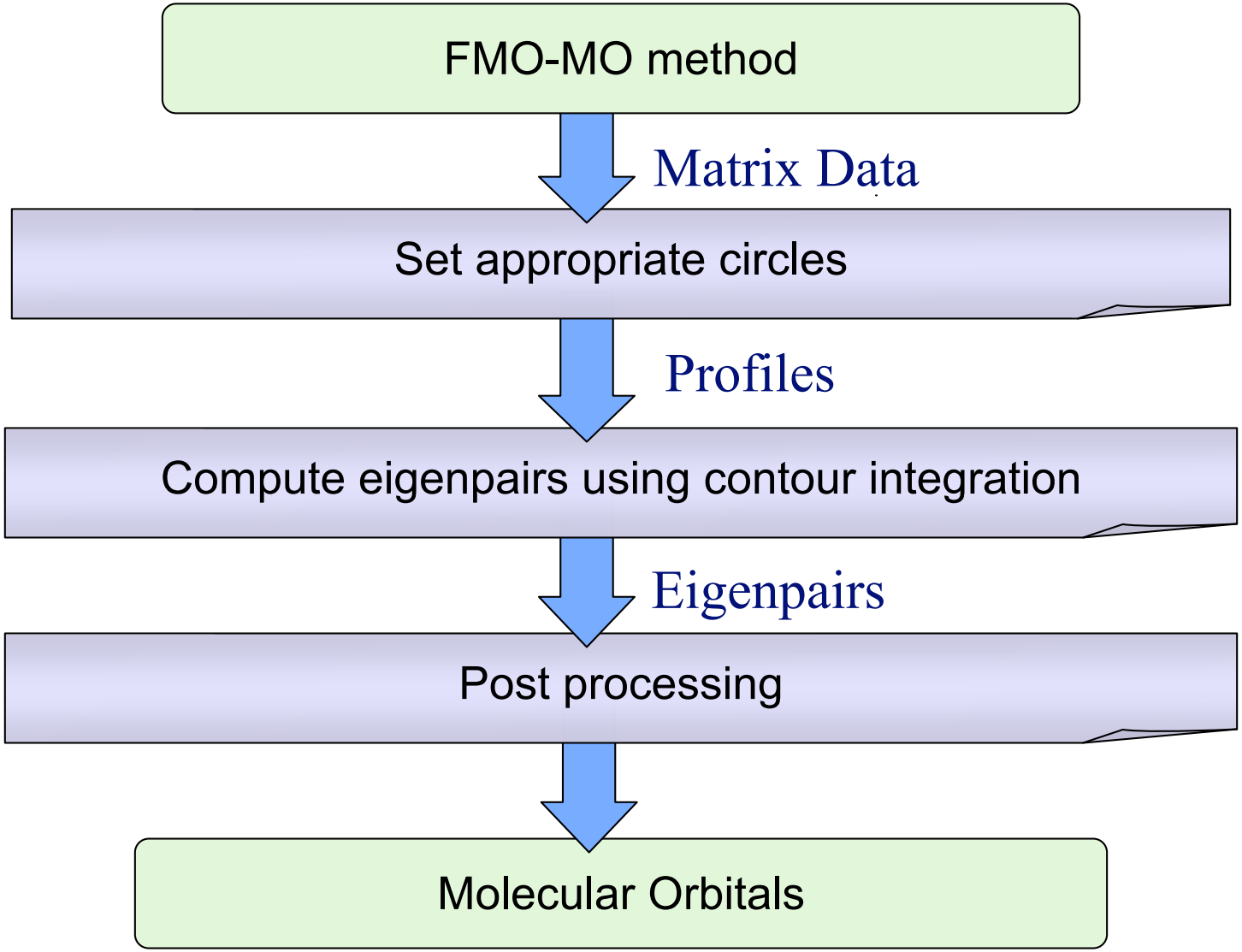
$$\boldsymbol{y}_j = (\omega_j B - A)^{-1} \boldsymbol{v}$$



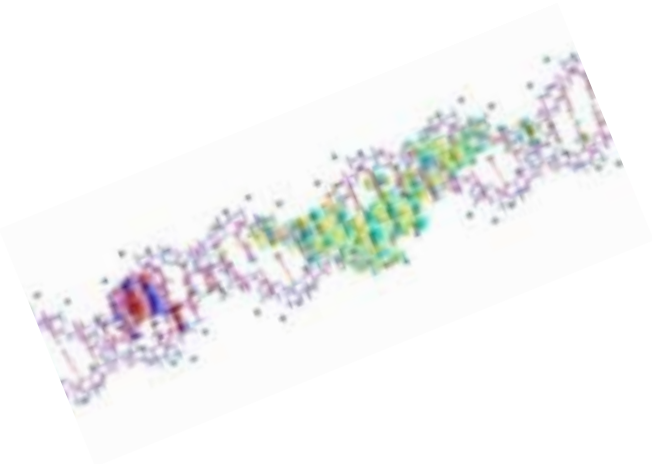
Parallel Implementation



Flow of the Eigensolver

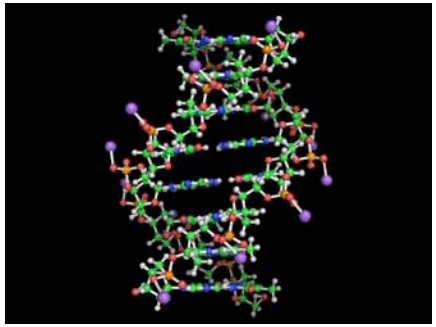


Contents



Numerical Example (1)

- Test problem:
  - Model of 8 DNA base pairs
  - Matrix size: 1,980 × 1,980
  - nnz: 728,080

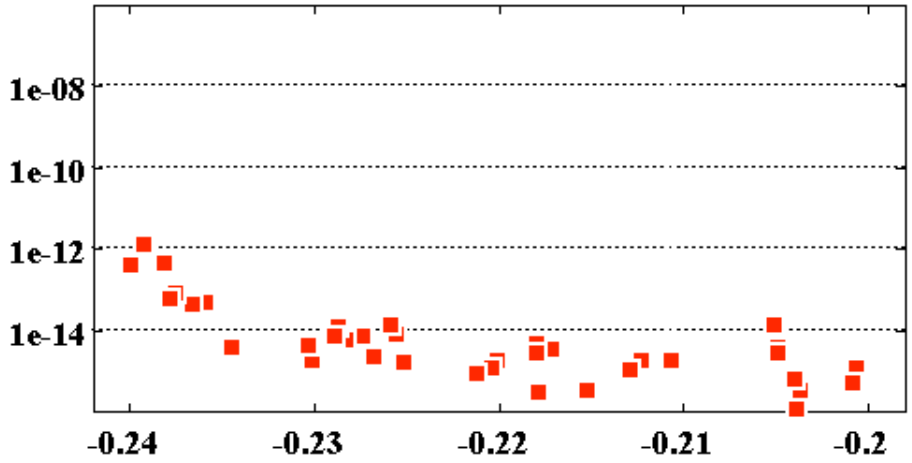


- Test Environment:
  - OS: MacOSX 10.5
  - CPU: Core 2 Duo 2.2GHz (2GB memory)
  - Software: MATLAB 7.5
  - Solver: UMFPACK (sparse direct solver)

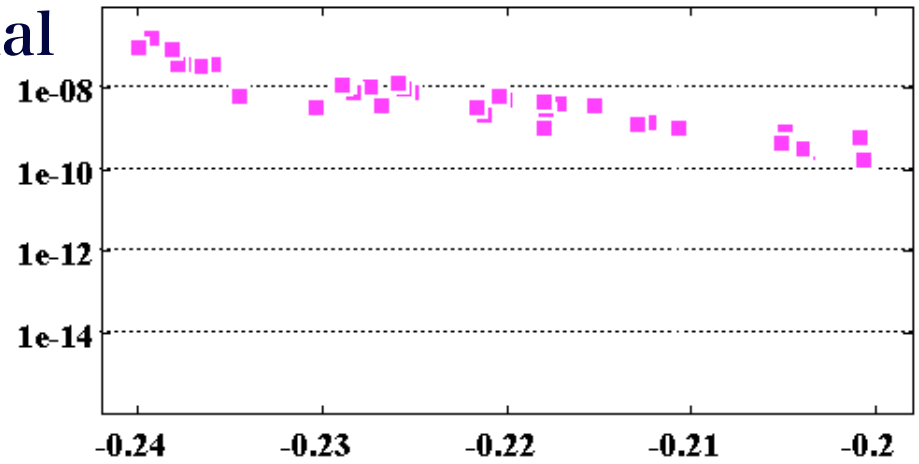
Numerical Example (1)

L= 12, N = 16, center = -0.22, radius = 0.02, 38 eigs

error



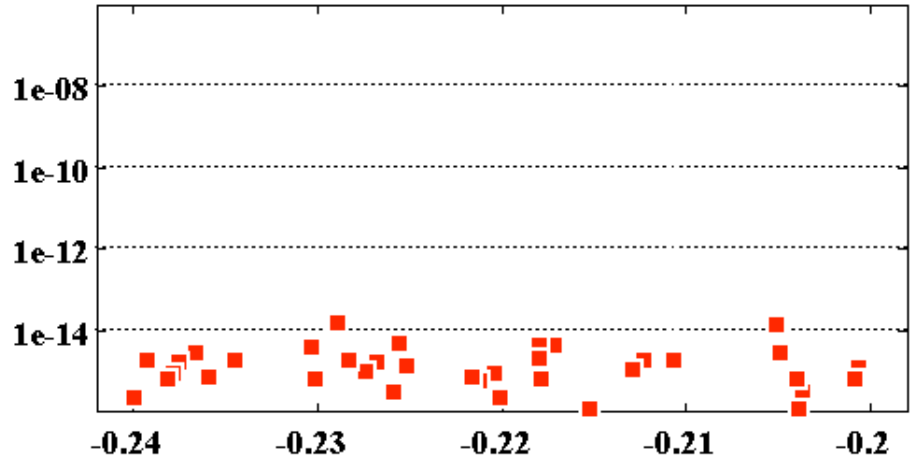
residual



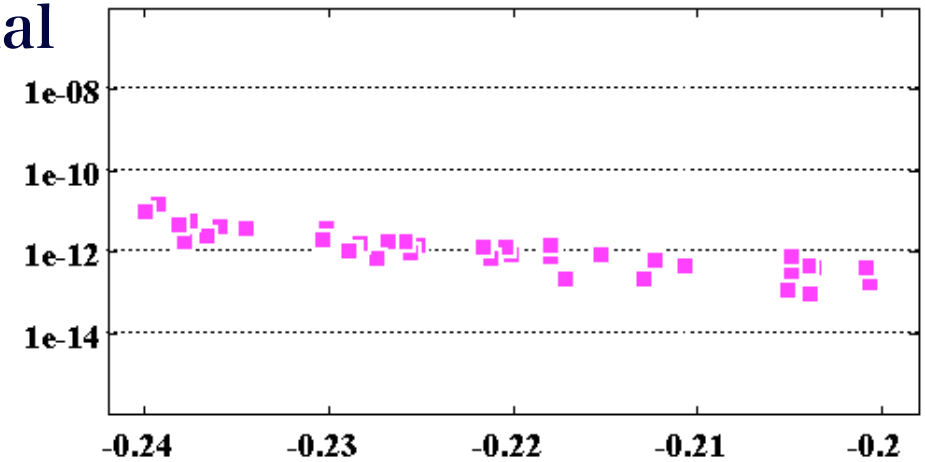
Numerical Example (1)

L = 16, N = 24, center = -0.22, radius = 0.02, 38 eigs

error



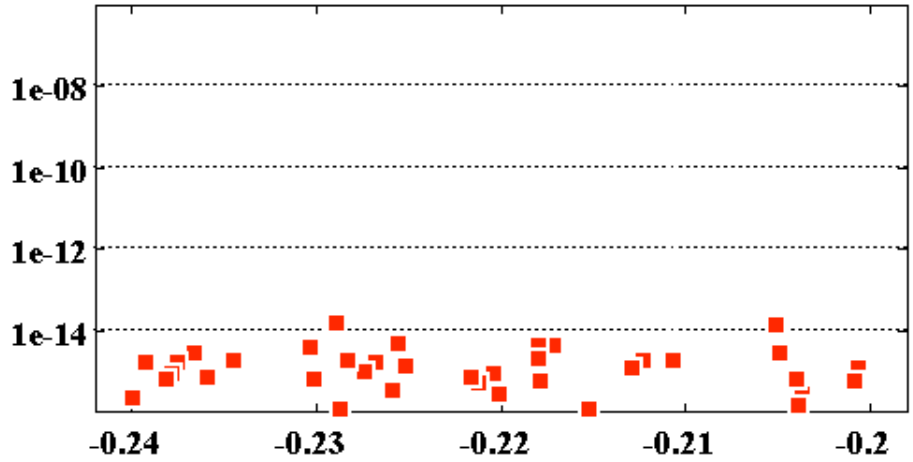
residual



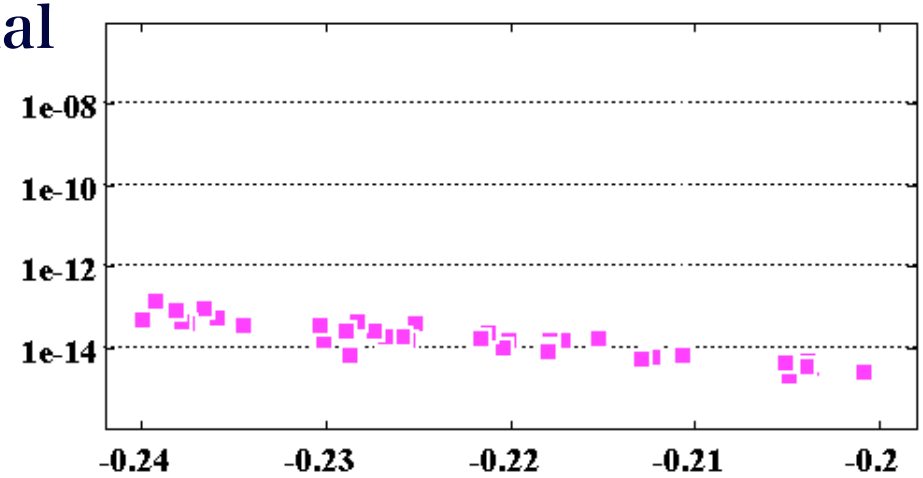
Numerical Example (1)

L = 20, N = 24, center = -0.22, radius = 0.02, 38 eigs

error

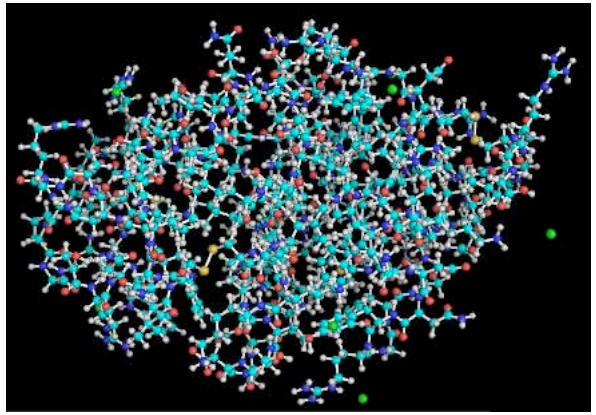


residual



Numerical Example (2)

- Test Problem:
  - Lysozime + H2O
  - Basis function: STO-3G
  - Size: 20,758 × 20,758
  - nnz: 20,064,444

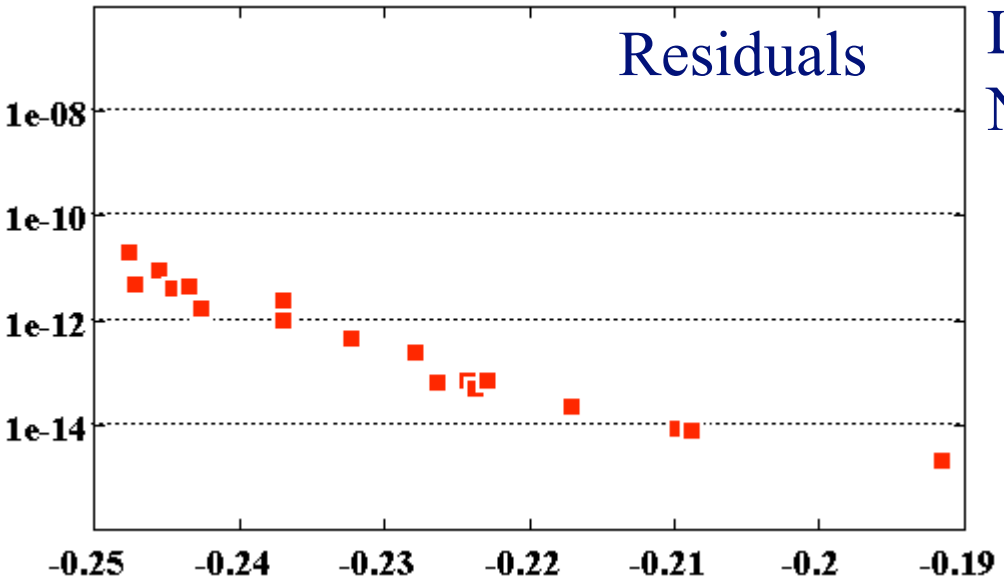


- Test Environment:
  - OS: MacOSX 10.5
  - CPU: Core 2 Duo 2.2GHz (2GB memory)
  - Compiler: icc 10.1, ifort 10.1
    Solver: COCG method [van der Vorst and Melissen (1990)]
  - Preconditioner: Complete Factorization for Approximate Matrix [Okada, S and Teranishi (2007)]
  - Sparse Direct Solver for Preconditioner: PARDISO

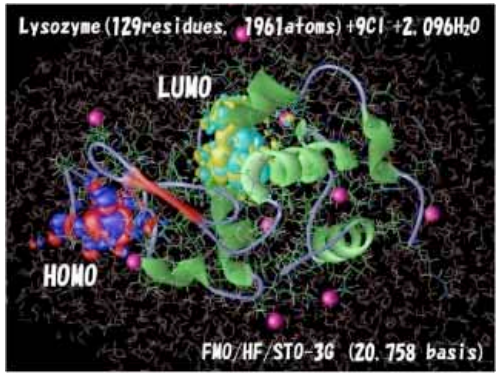
Numerical Example (2)

Center: -0.22 Radius: 0.03 18 eigs

Wall-clock time: 233.2 sec



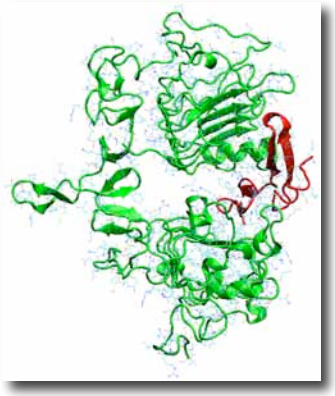
L = 12
N = 24



ARPACK+PARDISO: 316.1 sec, 20 eigs, max(res) = 6.6e-6
(Xeon 3.2GHz 2MB Memory)

Numerical Example (3)

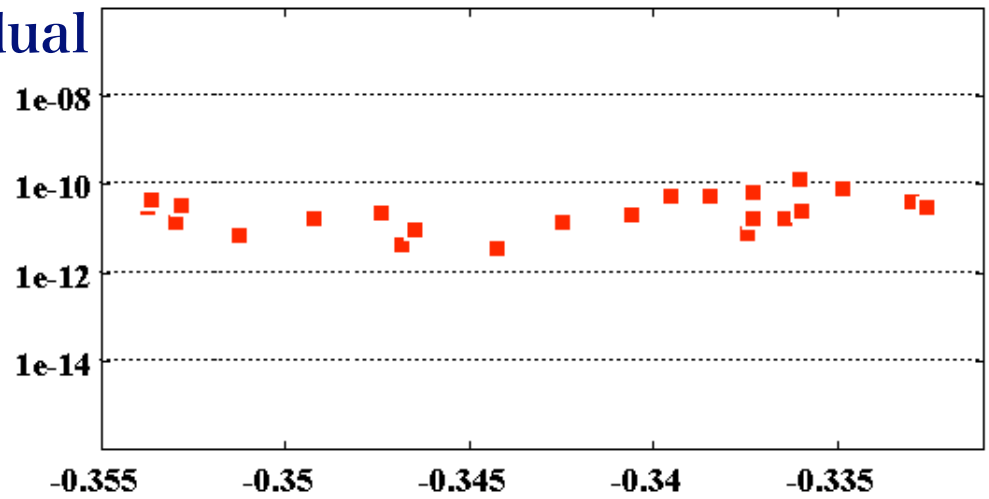
- Test Problems:
  - EGF (Epidermal Growth Factor)
  - Basis function: 6-31G
  - Size: 43,612 × 43,612
  - nnz: 73,175,935



- Test Environment:
  - OS: MacOSX 10.5
  - CPU: Core 2 Duo 2.2GHz (2GB memory)
  - Compiler: icc 10.1, ifort 10.1, MKL 10.0
  - Solver: COCG method [van der Vorst and Melissen (1990)]
  - Preconditioner: Complete Factorization for Approximate Matrix [Okada, S and Teranishi (2007)]
  - Sparse Direct Solver for Preconditioner: PARDISO

Numerical Example (3)

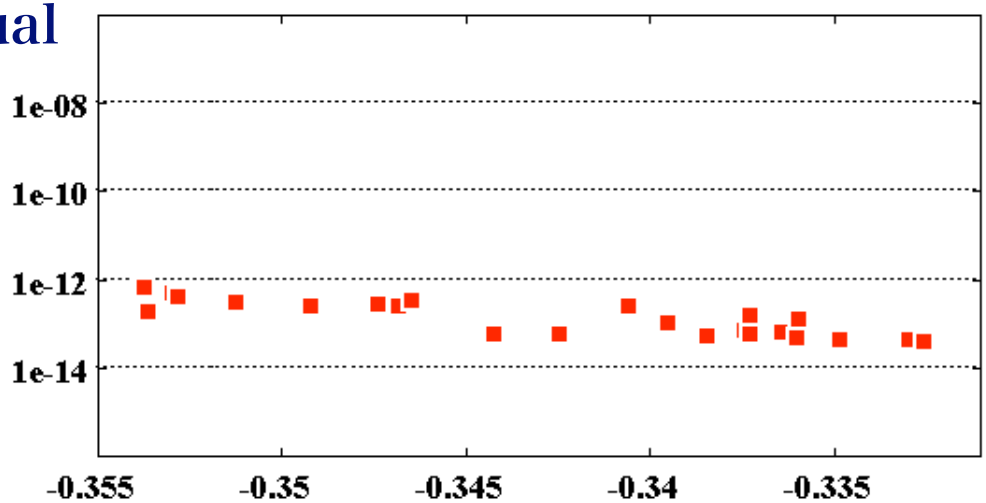
Residual



L = 8
N = 24

1583.1 sec

Residual



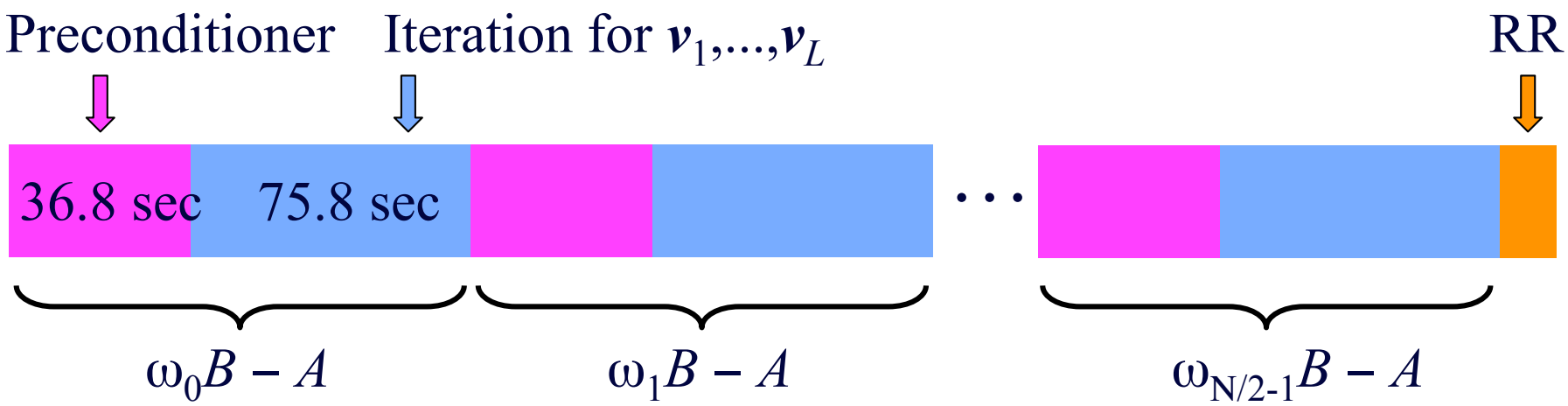
L = 12
N = 24

2017.7 sec

Numerical Example (3)

Timing result (serial case):

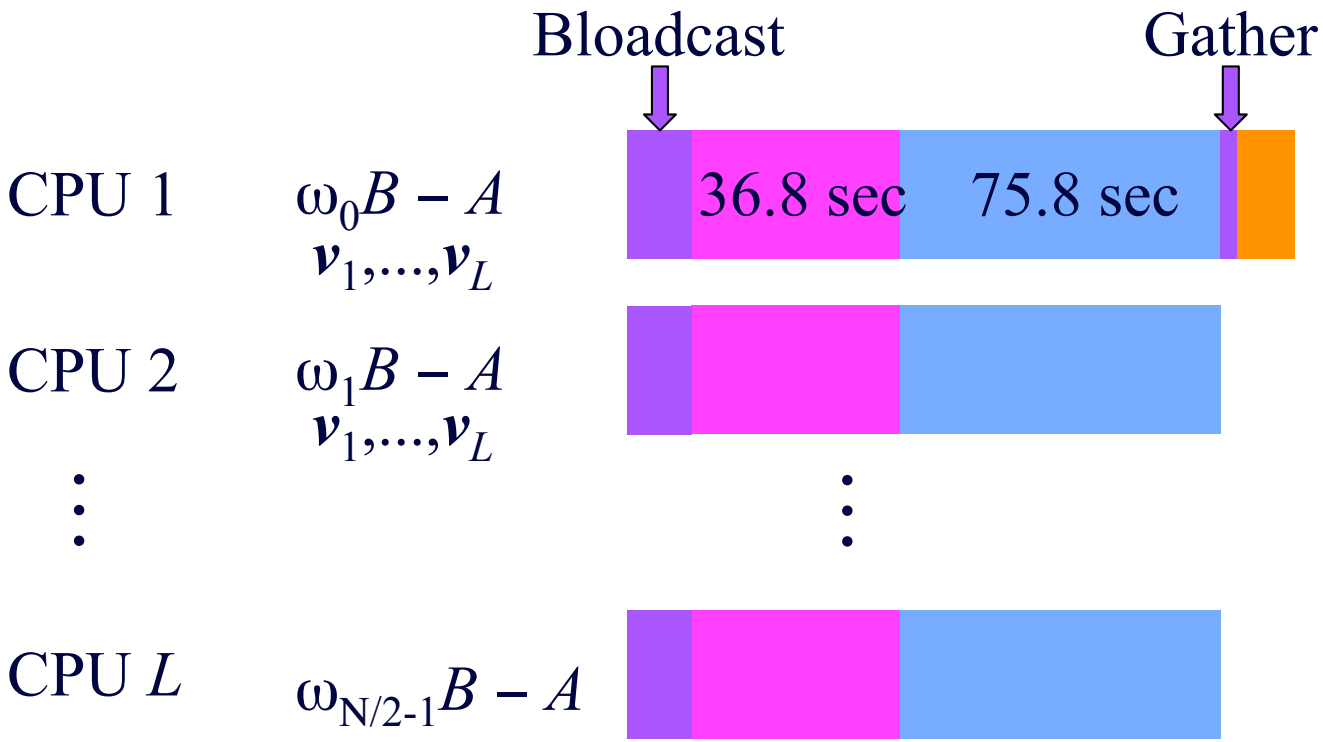
L = 12 N = 24 2017.7 sec



Numerical Example (3)

Timing result estimation (parallel case 1):

L = 12 N = 24



Numerical Example (3)

Timing result estimation (prallel case 2):

L = 12 N = 24

CPU 1 $\omega_0 B - A$

$\boldsymbol{v}_1$

CPU 2 $\omega_0 B - A$

$\boldsymbol{v}_L$

⋮ ⋮

CPU L*(N/2) $\omega_{N/2-1} B - A$

$\boldsymbol{v}_L$

Summary

- A Rayleigh-Ritz type method using the contour integral was proposed.
- This method finds limited number of eigenpairs in a given interval.
  - Efficient for molecular orbital computation.
  - Easy to implement for distributed computing.
- Find good preconditioner.
- Application for other problems.
  (Not only for SPD case)