



計算科学技術特論B

# 大規模量子化学計算(2)

小林 正人

(北海道大学理学研究院)

K-masato@mail.sci.hokudai.ac.jp

2014/07/10

# 講義概要

- 7/3 量子化学計算の概要と構成要素、高速化
  - ◆ 量子化学計算の目的と種類
  - ◆ 量子化学計算の手順、構成要素と高速化
  
- 7/10 大規模系に適用するための量子化学計算法
  - ◆ フラグメント分割に基づく方法
    - フラグメント分子軌道(FMO)法
    - 分割統治(DC)法
  - ◆ ラプラス変換MP2法
  - ◆ 2電子積分の密度フィッティング法
    - MP2計算への応用

# 量子化学計算にかかる時間と精度

分子の大きさの3乗に比例して計算時間増大


方法	Hartree-Fock (HF)法	密度汎関数理論(DFT)	MP2法 (摂動法)	CCSD法	CCSD(T)法
計算時間	$O(N^3)$	$O(N^3)$	$O(N^5)$	$O(N^6)$	$O(N^7)$
近似レベル	← 平均場理論 →		← 電子相関理論 →		
1000倍性能の計算機で	× 10	× 10	× 4.0	× 3.2	× 2.7
計算精度	定性的				正確

計算時間は

- ✓ 精度の低い理論でも $O(N^3)$
- ✓ 精度が上がるにつれて莫大に

『京』をただ使うだけでは  
大きな分子を扱えない

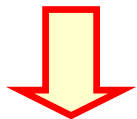
# 大規模量子化学計算手法

- 並列化とプログラムの工夫で頑張る
  - ◆ RSDFT (実空間密度汎関数理論)
  - ◆ ProteinDF (タンパク質密度汎関数プログラム)
- 密度行列を近似計算
  - ◆ エネルギー最小化法, 密度行列purification法
- 数学・アルゴリズムにより高速化
  - ◆ 積分のRI計算, Laplace変換MP2法
- 系を分割して計算を簡略化   $O(N) \sim$ 
  - ◆ フラグメント分子軌道(FMO)法
  - ◆ エロンゲーション法
  - ◆ 分割統治(DC)法

# 計算科学における分割統治(DC)法

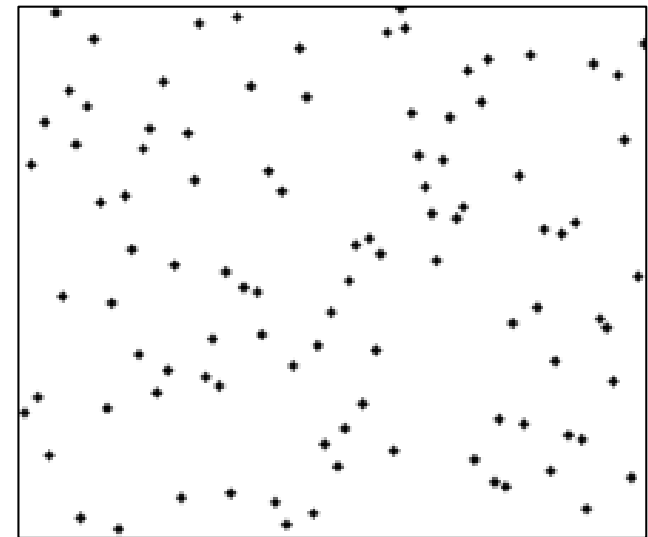
## ■ マージソート (フォン・ノイマン, 1945)

- ◆ 計算科学における最初の分割統治法
- ◆  $n$ 個のデータをソートするコスト:  $O(n \log n)$



## ■ さまざまなDCアルゴリズム

- ◆ 二分法 (求根, 探索)
- ◆ クイックソート
- ◆ カラツバ乗算法
- ◆ 高速フーリエ変換 (FFT)

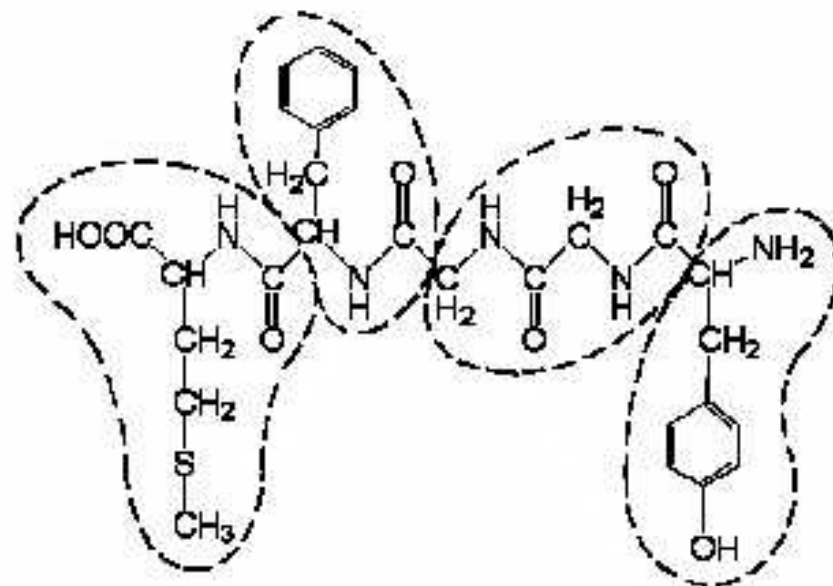


©Nuno Nogueira

分割部分は高並列化が可能  
(な場合が多い)

# フラグメント分子軌道(FMO)法

- 分子を(単結合で)切断
  - ◆ 結合に使われる電子は混成軌道を使って一方に寄せる (HOP)
- エネルギー等のプロパティは**多体展開**で求める



$$E_{\text{FMO1}} = \sum_I E_I \quad (\text{FMO1})$$

$$E_{\text{FMO2}} = \sum_I E_I + \sum_{I>J} (E_{IJ} - E_I - E_J) \quad (\text{FMO2})$$

$$E_{\text{FMO3}} = E_{\text{FMO2}} + \sum_{I>J>K} (E_{IJK} - E_{IJ} - E_{JK} - E_{IK} + E_I + E_J + E_K) \quad (\text{FMO3})$$

⋮

$E_I$ : モノマーのエネルギー  
 $E_{IJ}$ : ダイマーのエネルギー  
 $E_{IJK}$ : トリマーのエネルギー

- ◆ 多体展開を打ち切り  $\Rightarrow$  計算の大幅な高速化

# フラグメント分子軌道(FMO)法<sup>[1]</sup>

## ■ フラグメント $X$ [ $X = I$ (モノマー), $IJ$ (ダイマー), ...] の計算

◆ ハミルトニアン:  $\tilde{H}_{\mu\nu}^X = H_{\mu\nu}^X + V_{\mu\nu}^X + P_{\mu\nu}^X$  ( $\mu, \nu \in X$ )

■  $H_{\mu\nu}^X$ : フラグメント $X$ 自身のハミルトニアン

■  $V_{\mu\nu}^X$ :  $X$ の外側の電子・原子核からの静電ポテンシャル

◆  $V_{\mu\nu}^X = \sum_{K \neq X} \left[ \sum_{A \in K} \left\langle \mu \left| -\frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \right| \nu \right\rangle + \sum_{\lambda\sigma \in K} D_{\lambda\sigma}^K \Gamma_{\mu\nu, \lambda\sigma} \right]$

■  $P_{\mu\nu}^X$ :  $X$ に属していない混成軌道をプロジェクトアウト

◆ HF (KS) 方程式:  $\tilde{\mathbf{H}}^X \mathbf{C}^X = \mathbf{S}^X \mathbf{C}^X \boldsymbol{\varepsilon}^X$

■ 密度行列:  $D_{\mu\nu}^X = 2 \sum_i^{\text{occ}(X)} C_{\mu i}^X C_{\nu i}^X$

◆ 電子数(とスピン)はあらかじめ指定が必要

密度行列 $\mathbf{D}$ の自己無撞着な決定(SCC)が必要

[1] D.G. Fedorov and K. Kitaura eds., *The Fragment Molecular Orbital Method* (CRC Press, 2009).

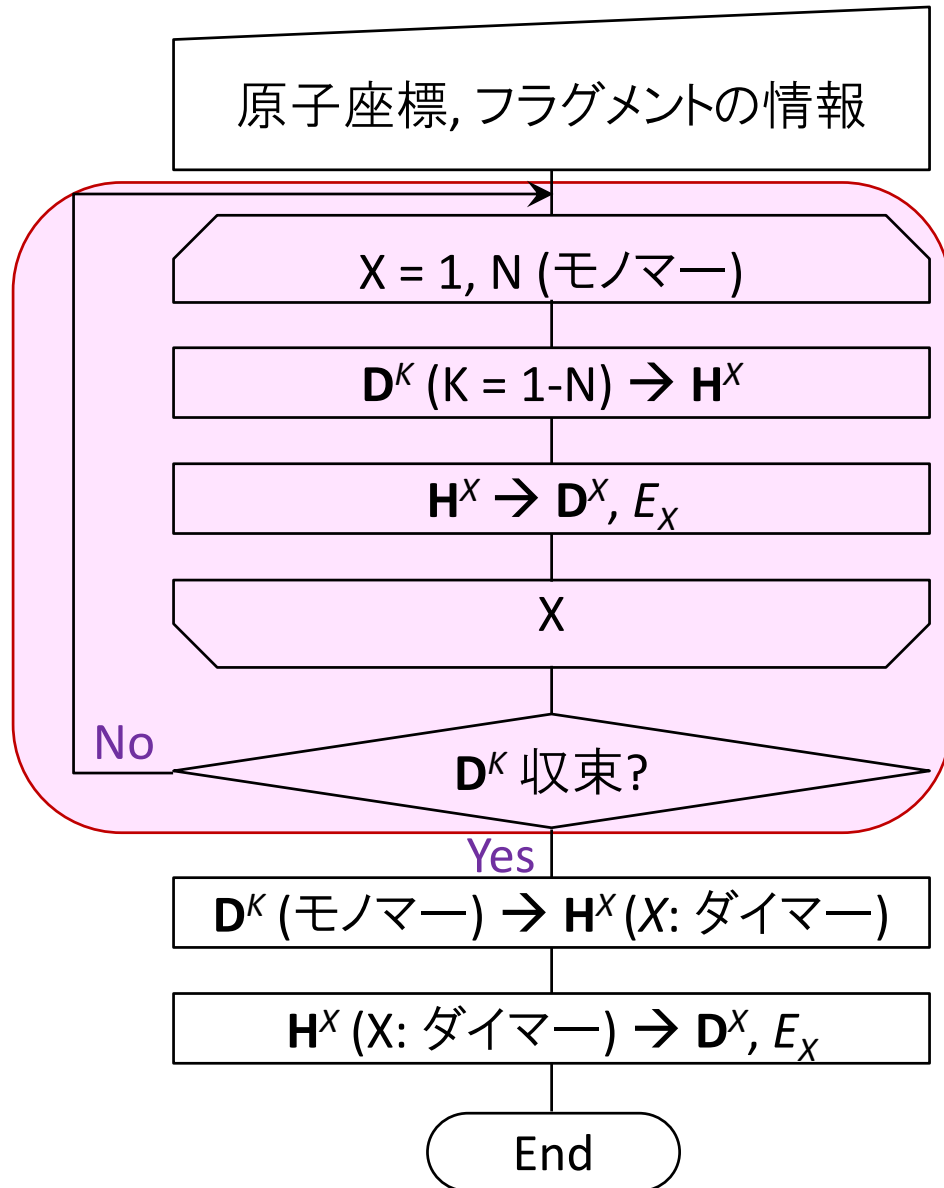
# FMO計算に用いられる近似

- FMO2の計算時間:  $O(N^2)$  [多体展開以外の近似なしの場合]
- カットオフ距離を使用して、いくつかの近似を導入
  - ◆ RESPPC: フラグメント間静電ポテンシャルを2電子積分を用いずにMulliken電荷で近似
  - ◆ RESDIM: ダイマー計算をあらわに実行せず、静電相互作用で近似
  - ◆ RCORSD [post HF電子相関計算を実行する場合]: ダイマーの電子相関を計算せずに無視
- SCC計算はモノマーに対してのみ実行

計算時間を大幅に削減し、 $O(N)$ を実現



# FMO2計算手順



## モノマー-SCC

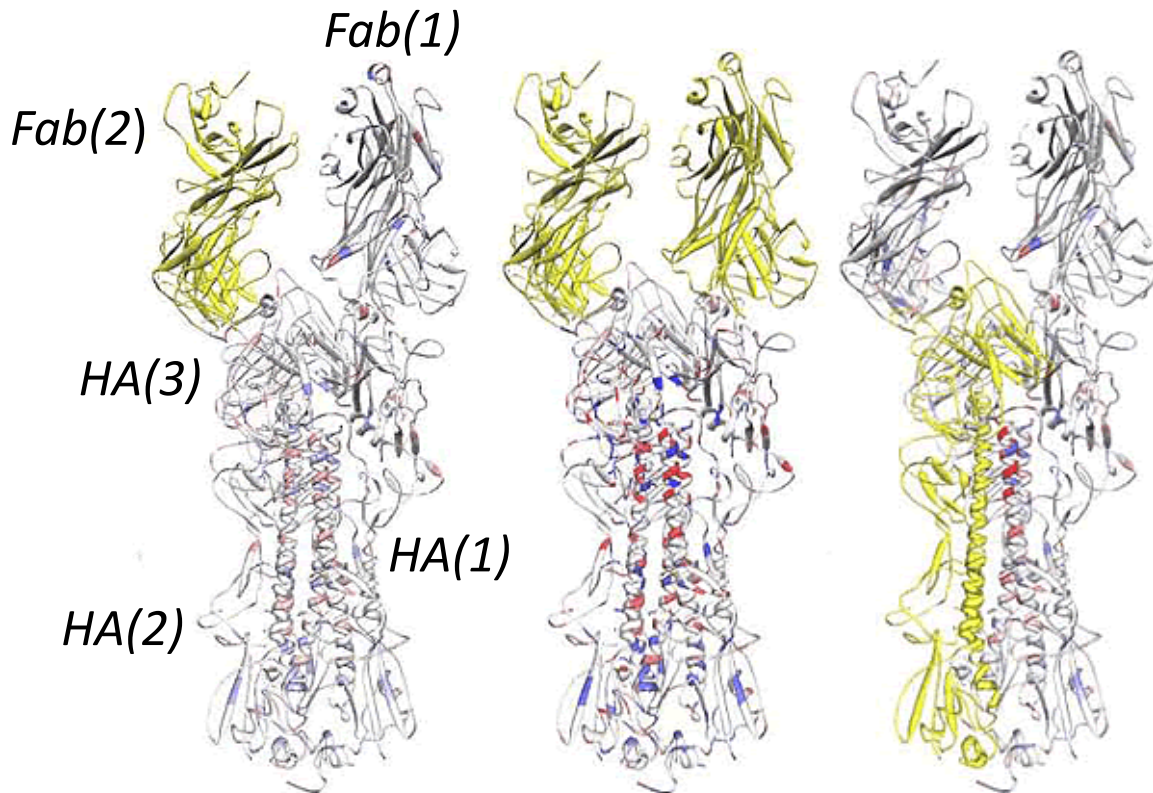
- ✓ 計算時間:  $O(N)$
- ✓ RESPPCにより計算時間のプレファクターを削減
- ✓ 並列化:  $D^K$ の通信が必要  
→ RESPPCにより簡略化

## ダイマー計算

- ✓ RESDIMにより $O(N)$ 化  
(静電相互作用計算のコストはnegligibly small)

# スパコンを用いた生体分子計算

- インフルエンザの抗原抗体複合系の計算<sup>[1]</sup>
  - ◆ 2350残基, 36000原子のMP2計算
  - ◆ フラグメントMO法による大規模相互作用解析計算



赤: 黄色との引力相互作用  
青: 黄色との斥力相互作用

阻害剤(タミフルなど)  
との相互作用計算  
にも応用

[1] S. Tanaka et al., Annual Report of the Earth Simulator Center 2010, 187.

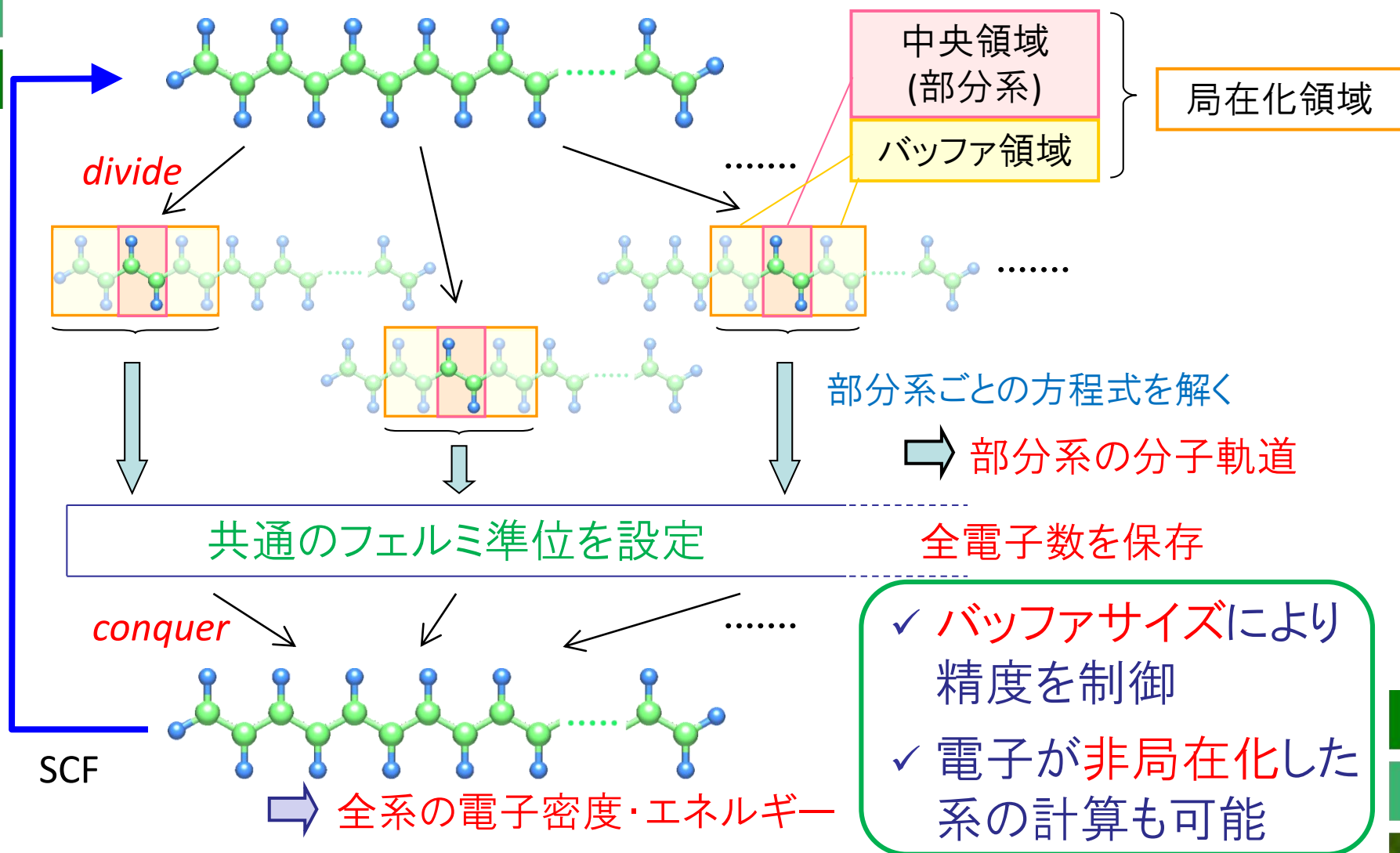
# 分割統治(DC)量子化学計算

- フラグメント(部分系)に分けて計算するのはFMOと同じ
- 部分系の周辺を**バッファ領域**として計算に含め、環境の効果を考慮
- 部分系の電子数は**フェルミ準位**を導入して自動決定(予め指定する必要なし)

SCFとpost-HF計算で取り扱いが大きく異なる

- 分割統治SCF法
  - ◆ **全系の密度行列**を部分系の寄与の和で表現
  - ◆ 分割行列を導入
- 分割統治post-HF電子相関計算
  - ◆ **部分系の相関エネルギー**を部分系の分子軌道で表現

# 分割統治量子化学計算のスキーム<sup>[1,2]</sup>



[1] MK and H. Nakai, in *Linear-Scaling Techniques in Computational Chemistry and Physics* (2011), pp.97-127.

[2] W. Yang and T.-S. Lee, *J. Chem. Phys.* **103**, 5674 (1995).

# DC-HF/DFT法

## ■ DC法: 密度行列を分割

$$D_{\mu\nu}^{\text{DC}} = \sum_{\alpha}^{\text{subsystem}} P_{\mu\nu}^{\alpha} D_{\mu\nu}^{\alpha}$$

$$P^{\alpha} = \begin{array}{|c|c|c|} \hline 0 & 1/2 & 0 \\ \hline 1/2 & 1 & 1/2 \\ \hline 0 & 1/2 & 0 \\ \hline \end{array} \left. \begin{array}{l} \text{バッファ領域} \\ \text{中央領域} \\ \text{バッファ領域} \end{array} \right\} \text{局在化領域 (部分系)}$$

## ◆ 部分系のMOから部分系の密度行列を構築

$$D_{\mu\nu}^{\alpha} \approx 2 \sum_p^{\text{MO}(\alpha)} f_{\beta}(\varepsilon_F - \varepsilon_p^{\alpha}) C_{\mu p}^{\alpha} C_{\nu p}^{\alpha} \quad \mu, \nu \in L(\alpha): \text{部分系}\alpha\text{の局在化領域の基底}$$

占有数をFermi関数で 各部分系で決定  $\mathbf{F}^{\alpha} \mathbf{C}^{\alpha} = \mathbf{S}^{\alpha} \mathbf{C}^{\alpha} \boldsymbol{\varepsilon}^{\alpha}$

→ Fermi準位は電子数の条件から決定

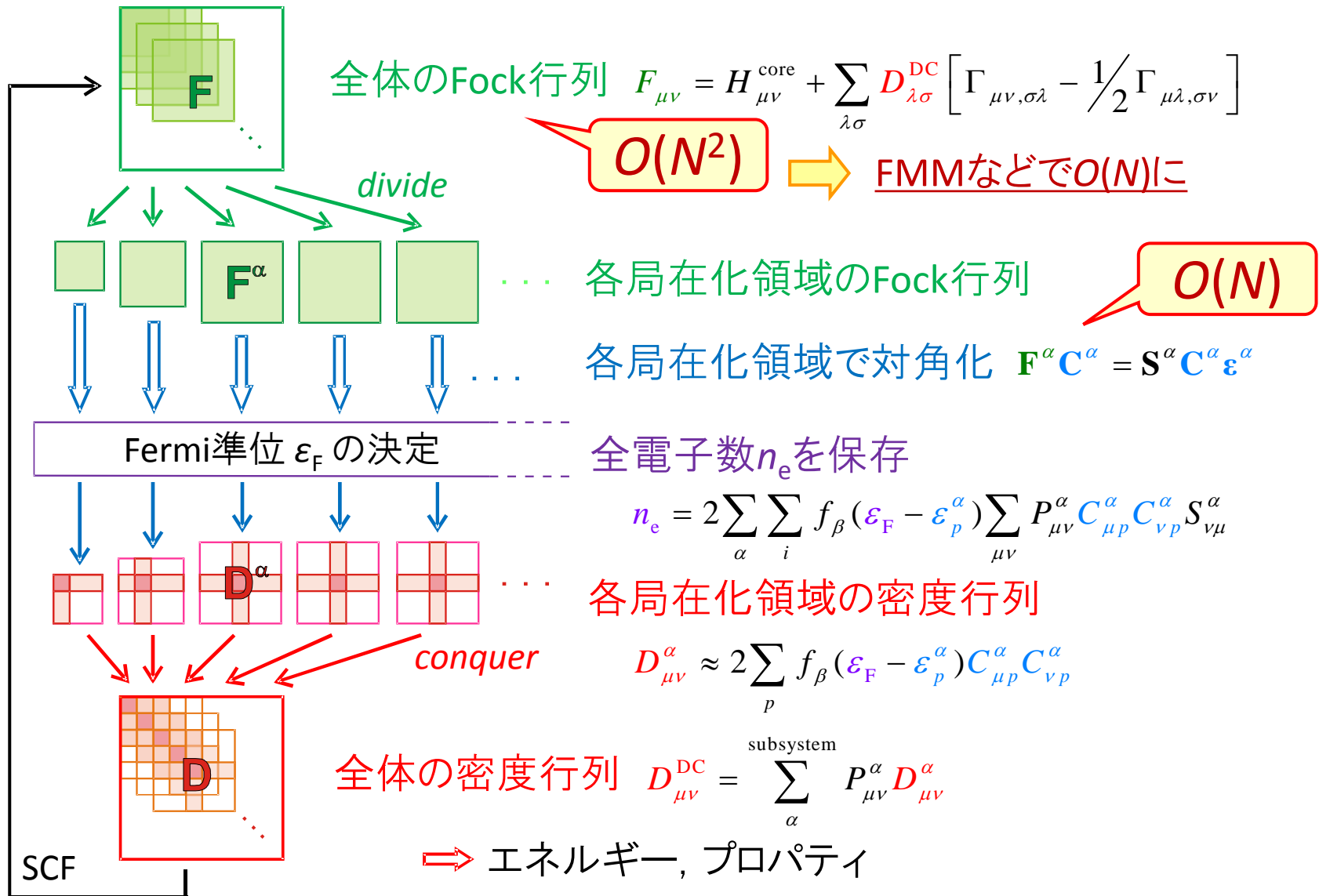
$$n_e = 2 \sum_{\alpha} \sum_p f_{\beta}(\varepsilon_F - \varepsilon_p^{\alpha}) \sum_{\mu\nu} P_{\mu\nu}^{\alpha} C_{\mu p}^{\alpha} C_{\nu p}^{\alpha} S_{\nu\mu}^{\alpha}$$

## ◆ Fock行列とエネルギーの表式は、通常と同じ

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} + \sum_{\lambda\sigma} D_{\lambda\sigma}^{\text{DC}} \left[ \Gamma_{\mu\nu,\sigma\lambda} - \frac{1}{2} \Gamma_{\mu\lambda,\sigma\nu} \right]$$

$$E_e^{\text{DC}} = \frac{1}{2} \text{Tr} \left[ \mathbf{D}^{\text{DC}} (\mathbf{H}^{\text{core}} + \mathbf{F}) \right] \quad (\text{Hartree-Fock})$$

# DC-HF/DFT計算スキーム



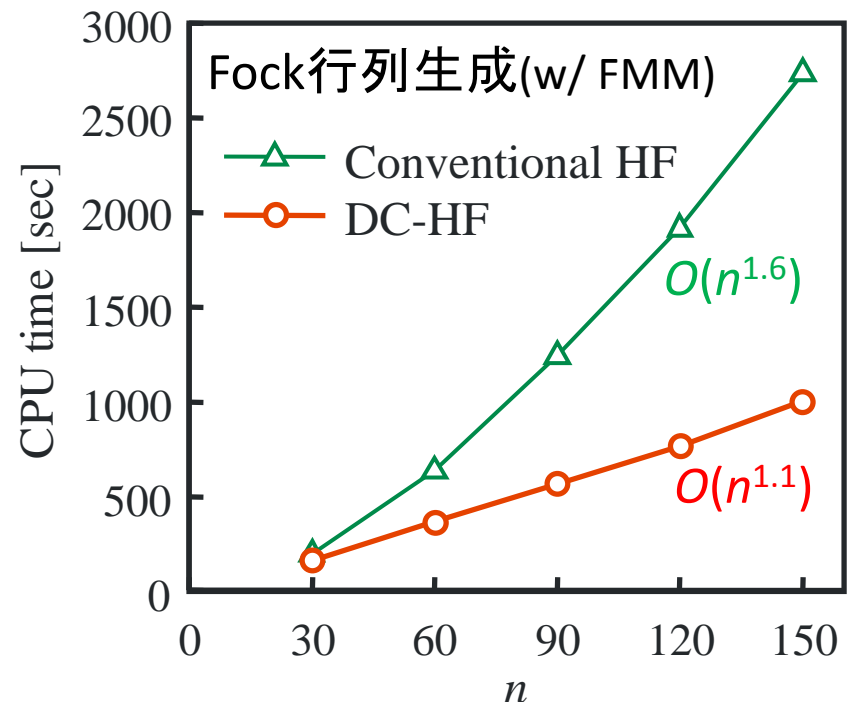
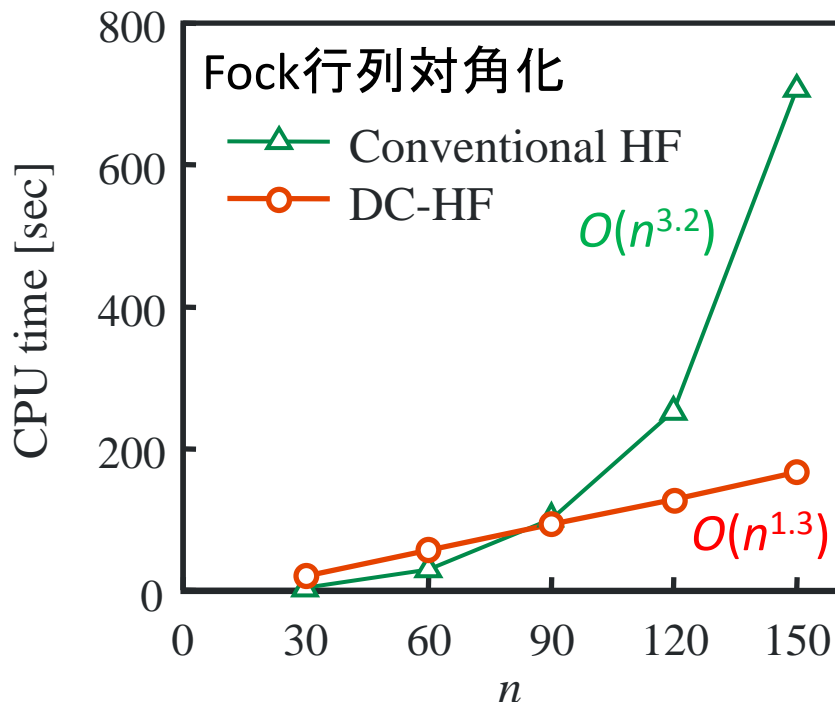
# DC-HF計算: CPU時間

## ■ 最初のSCF cycleにかかるCPU時間<sup>[1]</sup>

◆ ポリエン鎖  $C_nH_{n+2}$



基底関数: 6-31G\*\*  
中央領域:  $C_2H_{2(3)}$  (1 unit)  
バッファ: 6 units  
Pentium4 / 3.20 GHz



DC法によりほぼリニアスケーリングを達成

[1] M. Kobayashi and H. Nakai, in *Linear-Scaling Techniques in Computational Chemistry and Physics* (2011), pp. 97-127.

# Fermi準位の決定法

■ 電子数保存の条件:  $n_e = 2 \sum_{\alpha} \sum_i f_{\beta}(\epsilon_F - \epsilon_p^{\alpha}) \sum_{\mu\nu} P_{\mu\nu}^{\alpha} C_{\mu p}^{\alpha} C_{\nu p}^{\alpha} S_{\nu\mu}^{\alpha}$

◆  $\epsilon_F$ を求める非線形方程式

◆ 部分系の軌道の重み  $w_p^{\alpha} = \sum_{\mu\nu} P_{\mu\nu}^{\alpha} C_{\mu p}^{\alpha} C_{\nu p}^{\alpha} S_{\nu\mu}^{\alpha}$  を保存

◆ Fermi関数を計算する閾値を設定<sup>1</sup>

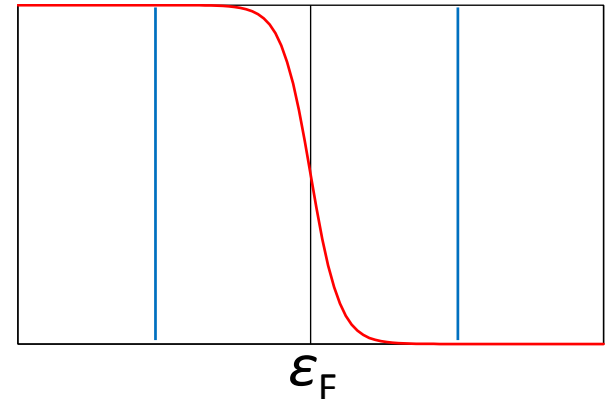
◆ Brent法で求根

■  $\epsilon_F$ の上限と下限を設定  
(一方は0または前回の結果を利用)<sup>0</sup>

■ 囲い込み、二分法、補間の組合せ

■ 計算時間は $O(N)$ だが、並列化は容易ではない

(繰り返しのたびに通信が必要)



$$f_{\beta}(x) = \left[ \exp\left(\frac{x}{k_b T}\right) + 1 \right]^{-1}$$



# Density functional tight binding (DFTB)法<sup>[1]</sup>

## ■ DFTB法: DFTをベースとした半経験的計算法

### ◆ DFTエネルギー

$$E[\rho] = \sum_i n_i \langle \psi_i | -\frac{1}{2} \nabla^2 + v(\mathbf{r}) + \frac{1}{2} \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' | \psi_i \rangle + E_{xc}[\rho] + \sum_{A < B}^N \frac{Z_A Z_B}{|\mathbf{R}_A - \mathbf{R}_B|}$$

⇓ Tight binding近似:  $\rho = \rho_0 + \delta\rho$

⇓ 交換相関項に対して2次のTaylor展開:  $E_{xc}[\rho_0 + \delta\rho]$

⇓ Charge dependent項に対してモノポール近似:  $\delta\rho_A \approx \Delta q_A F_{00} Y_{00}$

### ◆ DFTBエネルギー

DFT計算から決定するパラメータ

/  
Mulliken charge

$$E_2^{\text{TB}} = \sum_i \langle \psi_i | \hat{H}_0 | \phi_i \rangle + \frac{1}{2} \sum_{A,B} \Delta q_A \Delta q_B \gamma_{AB} + \sum_{A < B} E_{\text{rep}}(\mathbf{R}_{AB})$$

Charge independent

Charge dependent

Repulsive energy

2電子積分不要!

■ Fock行列  $F_{\mu\nu} = (H_0)_{\mu\nu} + \frac{1}{2} \sum_C \gamma_{AC} \gamma_{BC} \Delta q_C S_{\mu\nu}$

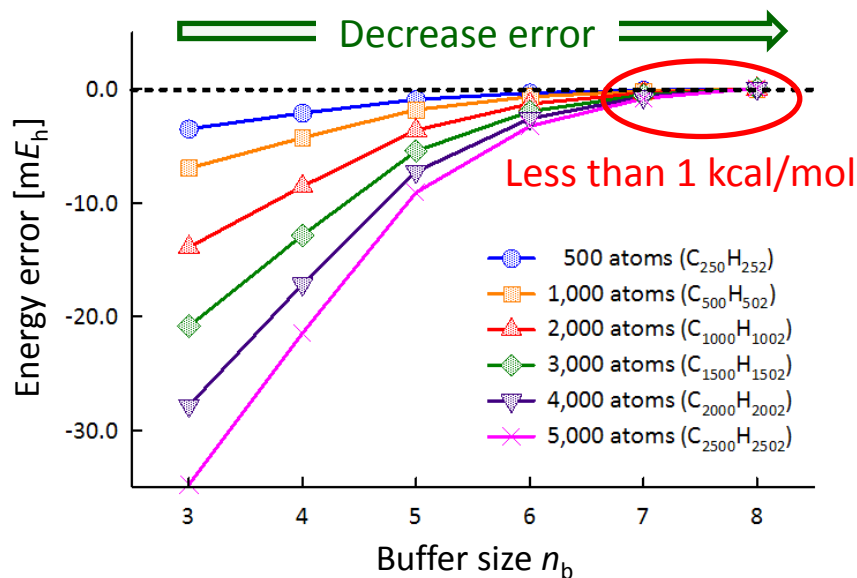
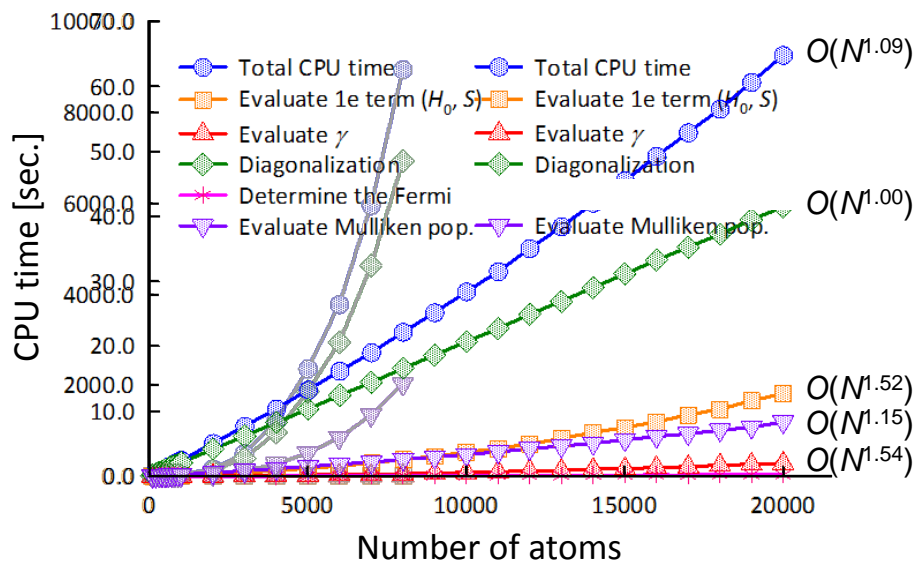
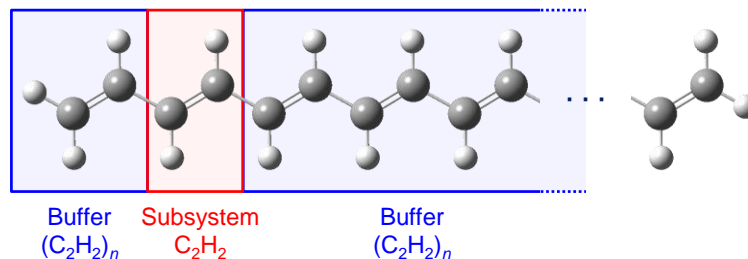
■ Mulliken電荷を自己無撞着に決定 (SCC)

[1] D. Porezag, Th. Frauenheim, Th. Köhler, G. Seifert, and R. Kaschner, *Phys. Rev. B* **51**, 12947 (1995).

# DC-DFTB計算のパフォーマンス

## ■ 計算時間の比較

### ◆ ポリエン鎖 $C_{2n}H_{2n+2}$



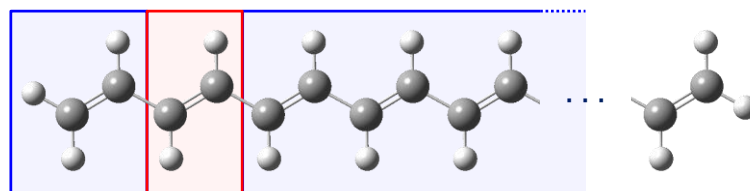
- ✓ DC法を用いて対角化の計算時間が $O(N)$ に
- ✓ エネルギー誤差ほとんどなしで計算可能
- ✓ エネルギー勾配計算も $O(N)$

〔 部分系:  $C_2H_{2(3)}$  (1 unit)  
バッファ:  $n_b$  units 〕

# DC-DFTB: 並列化効率

## ■ 並列化による高速化率

### ◆ ポリエン鎖 $C_{10000}H_{10002}$

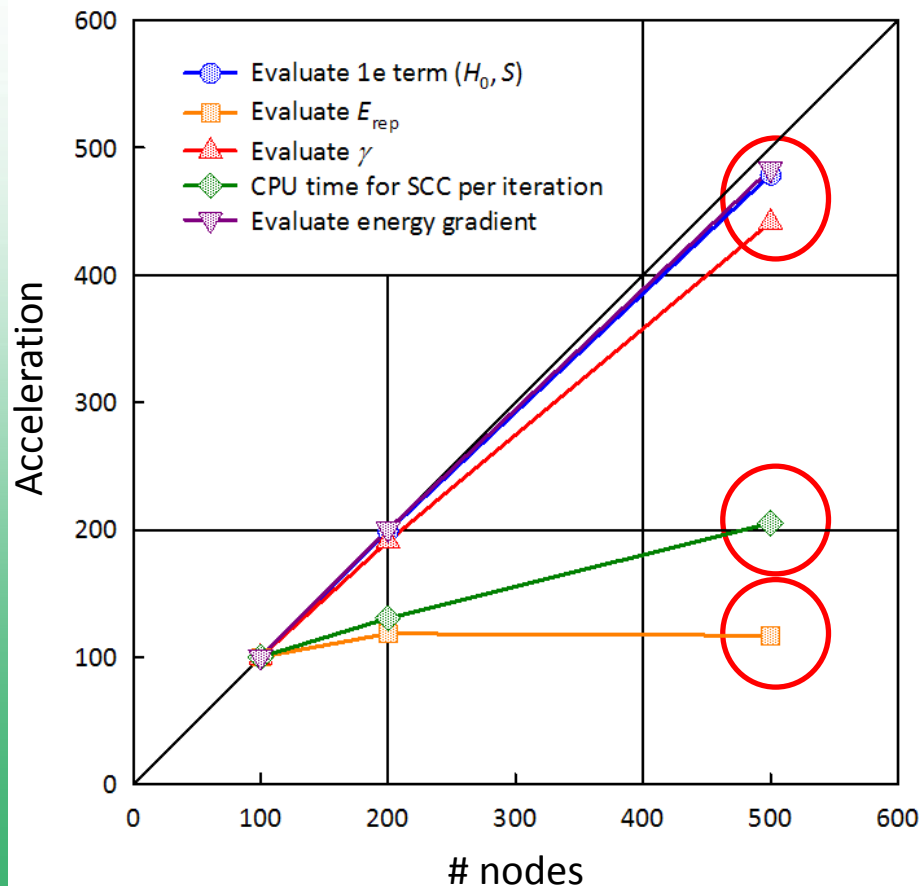


Buffer  
 $(C_2H_2)_n$

Subsystem  
 $C_2H_2$

Buffer  
 $(C_2H_2)_n$

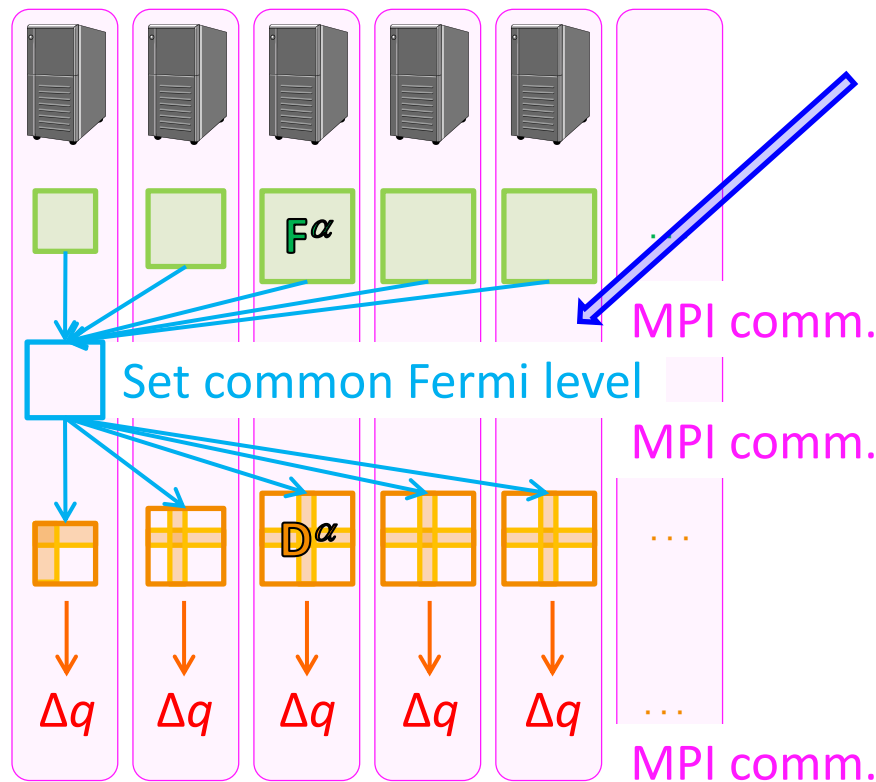
部分系:  $C_2H_{2(3)}$  (1 unit)  
バッファ: 8 units  
計算機: 京



- ✓ 500ノード(4000コア)使用時の計算時間は2.2秒
- ✓  $H_0$ 、 $S$ 、 $\gamma$ の計算やgradient計算の効率は90%以上
- ✓ 反発エネルギー計算は全計算時間の1%未満
- ✓ ボトルネックは SCCエネルギー計算

# SCCエネルギー計算の並列化

## ■ 並列計算の手順



全部分系の軌道エネルギーを使ってFermi準位 $\epsilon_F$ の求根  
(Brent法等を使用)

$$n_e = \sum_{\alpha} \sum_p f_{\beta} (\epsilon_F - \epsilon_p^{\alpha}) w_p^{\alpha}$$

逐次計算が繰り返し  
計算の中に存在

500ノード使用時の計算時間

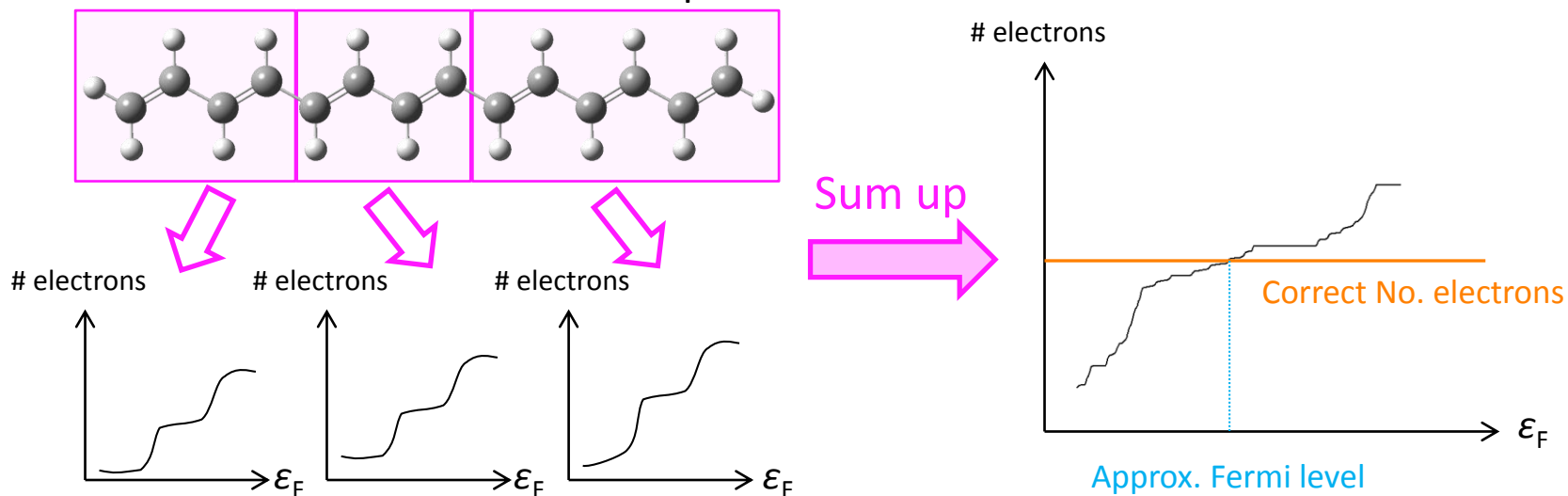
- ✓ SCCエネルギー計算: 1.10 s
- ✓ うちFermi準位決定: 0.78 s

⇒ 並列計算可能な新たなフェルミ準位  
計算アルゴリズムの開発

# 補間を用いた新たな $\epsilon_F$ 計算アルゴリズム

## ■ 新アルゴリズムの手順:補間(内挿)法を利用

1. Fermi準位の推定範囲を指定
2. 各ノードで推定範囲内を等間隔にした各 $\epsilon_F$ に対して、電子数を計算
3. MPI\_Reduceにより、 $\epsilon_F$ の上限・下限を決定

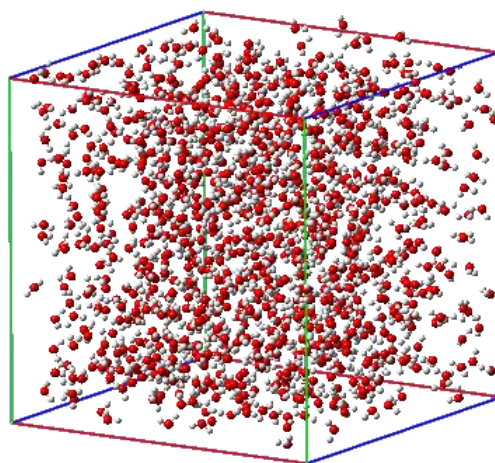


4. 適切な桁まで求めたら、Spline補間で $\epsilon_F$ を決定

# DC-DFTB-K: 並列計算パフォーマンス

## ■ 計算時間等の比較

- ◆ 水40000分子
- ◆ SCCエネルギー計算1回  
に対する計算

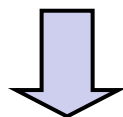


（部分系: 水1分子）  
バッファ: 5.5 Å  
計算機: 京

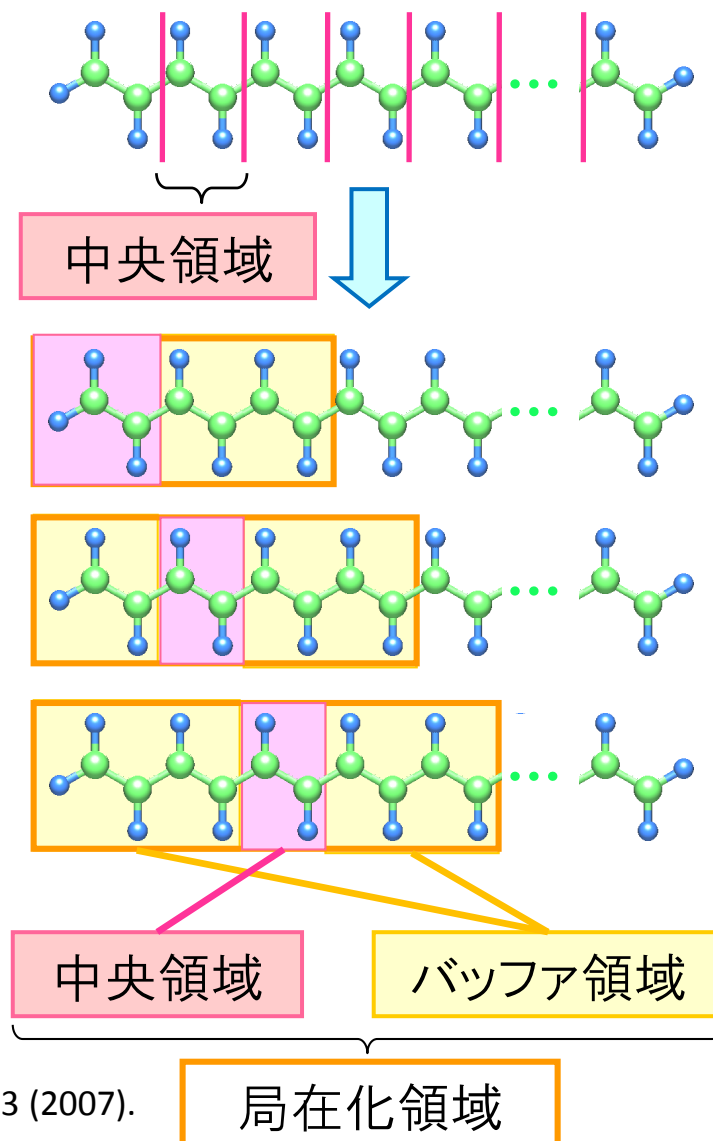
ノード数	コア数	実行時間 [s]	対ピーク 演算性能
2500	20000	1.1254	17.7%
5000	40000	0.6271	16.1%
10000	80000	0.4895	10.4%
効率		79.46%	

# 分割統治post-HF電子相関計算<sup>[1]</sup>

- 部分系の相関エネルギー
  - ◆ DC-HF計算で求められる部分系(局在化領域)の軌道から相関エネルギーを計算
  - ◆ 中央領域だけの相関エネルギーの見積もりが必要



エネルギー密度解析  
(EDA)<sup>[2]</sup>の利用



[1] M. Kobayashi, Y. Imamura, and H. Nakai, *J. Chem. Phys.* 127, 074103 (2007).

[2] H. Nakai, *Chem. Phys. Lett.* 363, 73 (2002).

# 分割統治post-HF電子相関計算

## ■ 相関エネルギー(Nesbetの定理)

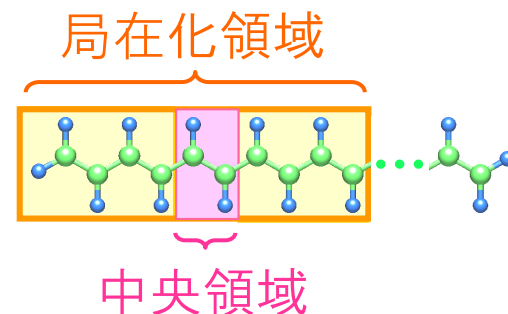
$$\Delta E = \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} (ia | jb) [2\tilde{t}_{ia,jb} - \tilde{t}_{ib,ja}]$$

## ■ 部分系の $\Delta E$ を部分系の軌道から計算

$$\Delta E^\alpha = \sum_{i,j}^{\text{occ}(\alpha)} \sum_{a,b}^{\text{vir}(\alpha)} w_{\text{occ}} \sum_{\mu \in S(\alpha)} C_{\mu i}^\alpha (\mu a^\alpha | j^\alpha b^\alpha) [2\tilde{t}_{ia,jb}^\alpha - \tilde{t}_{ib,ja}^\alpha]$$

EDA

$S(\alpha)$ : 部分系 $a$ の中央領域のAO



◆ MP2の場合:  $\tilde{t}_{ia,jb}^\alpha = -\frac{(i^\alpha a^\alpha | j^\alpha b^\alpha)}{\epsilon_a^\alpha + \epsilon_b^\alpha - \epsilon_i^\alpha - \epsilon_j^\alpha}$

◆ 全相関エネルギー = 部分系の相関エネルギーの和

$$\Delta E \approx \sum_{\alpha}^{\text{subsystem}} \Delta E^\alpha$$

DC-MP2法



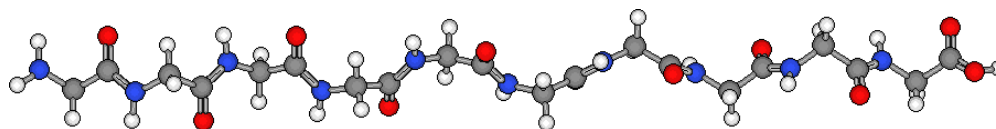
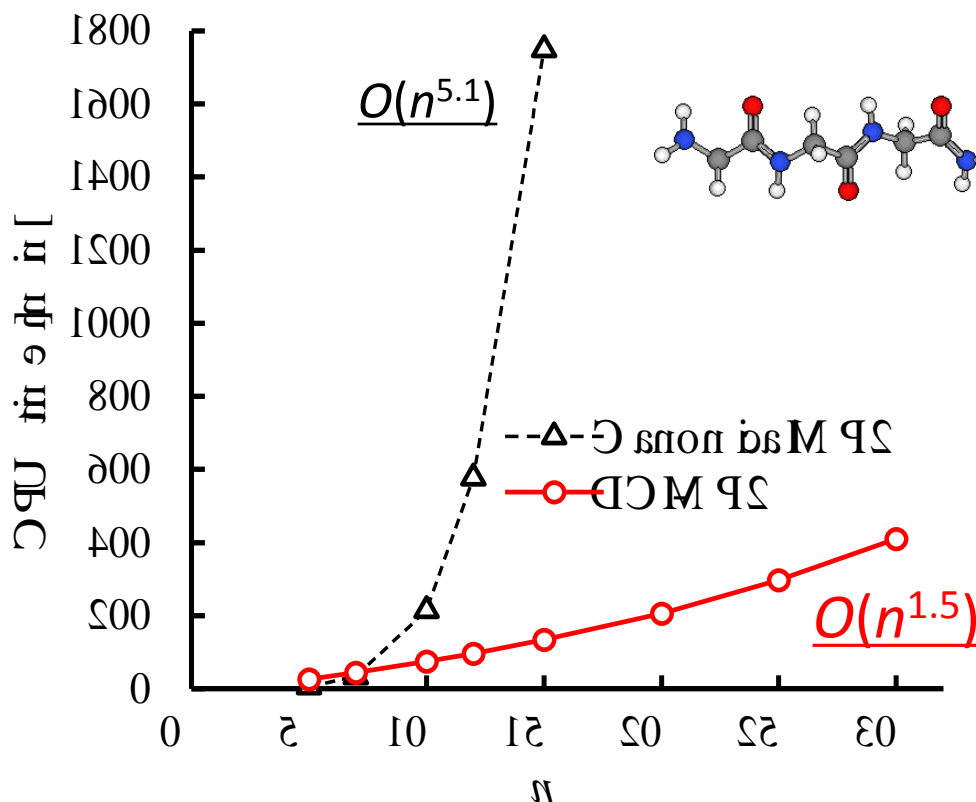
# DC-MP2法の計算時間

## ■ MP2計算にかかる時間

### ◆ $\beta$ ストランドグリシンペプチド (Gly)<sub>n</sub>

$r_b = 6.0 \text{ \AA}$   
部分系: 1原子  
MP2/6-31G

Xeon (Paxville) 2.8 GHz (1CPU)



- ✓ 計算時間: 系の大きさに対してほぼ線形
- ✓ 必要メモリ量: 系の大きさ  
にほぼ非依存  
(最大部分系の大きさに依存)

# DC-MP2計算の並列化

## ■ 部分系の相関エネルギー

- ◆ 重複を防ぐため、**中央領域**だけの相関エネルギーを算定

$$\Delta E^\alpha = \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \sum_{\mu \in \mathcal{S}(\alpha)} C_{\mu i}^\alpha \langle \mu j^\alpha | a^\alpha b^\alpha \rangle [2\tilde{t}_{ij,ab}^\alpha - \tilde{t}_{ij,ba}^\alpha]$$

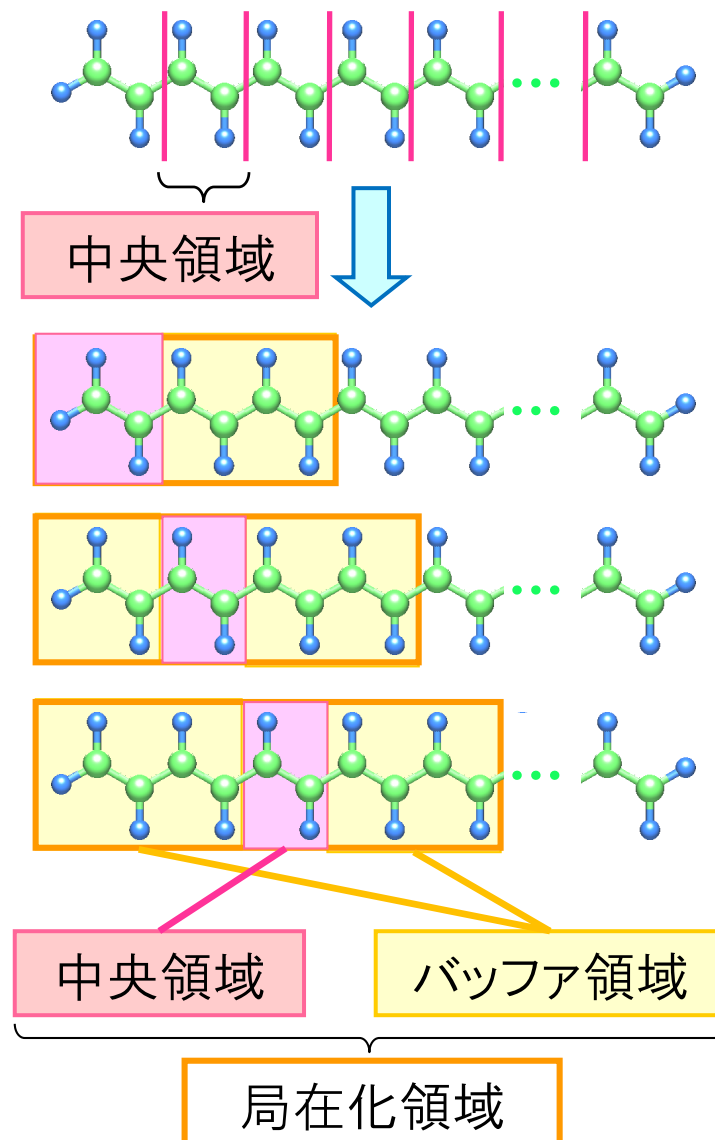
- ◆ 既存の高並列MP2アルゴリズムを利用可能

## ■ 全系の相関エネルギー

$$\Delta E \approx \sum_{\alpha}^{\text{subsystem}} \Delta E^\alpha$$

- ◆ 部分系ごとに独立の計算

2段階の並列化が可能!



# GDDIを用いた階層的並列処理

## ■ Generalized distributed data interface (GDDI)

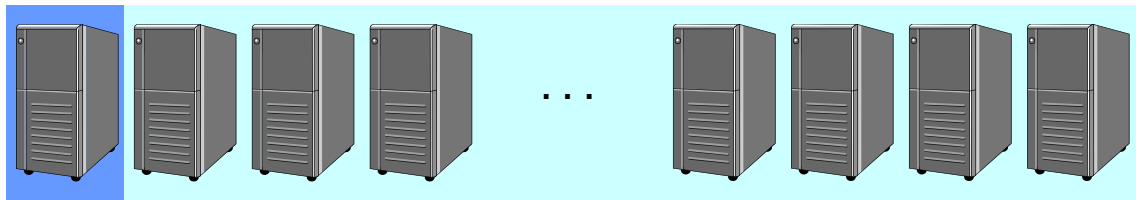
◆ GAMESSで多層並列処理を担うインターフェース

◆ 3種類の**スコープ**(MPIのコミュニケータ)を使用

(1) **DDI\_WORLD**: 全ノード = 1グループ (MPI\_COMM\_WORLD)

マスタ0

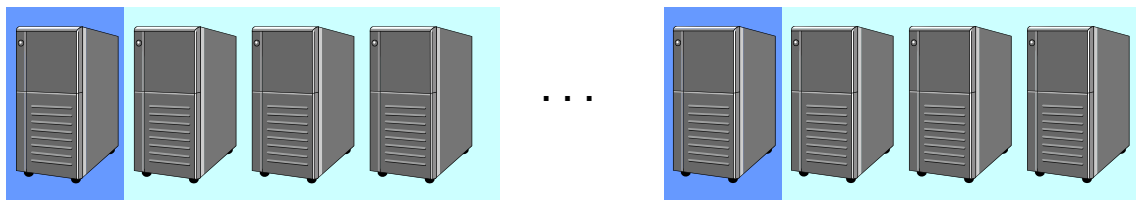
スレーブ



(2) **DDI\_GROUP**: グループ分割して並列処理 (グループ内並列)

マスタ0

マスタ (N-1)



グループ内では  
通常の並列処理

(3) **DDI\_MASTERS**: マスタ間で通信処理 (グループ間並列)

# DC-MP2の2段階並列アルゴリズム

部分系のエネルギー計算を各グループに割り振り:

**GDDI\_SCOPE(DDI\_GROUP)**

グループ0

$$\Delta E^{\alpha=1}$$



グループ (N-1)

$$\Delta E^{\alpha=N}$$



部分系の計算はグループ内で並列化



部分系のエネルギーの足合わせ: **GDDI\_SCOPE(DDI\_MASTERS)**

$$\Delta E = \sum_{\alpha}^{\text{subsystem}} \Delta E^{\alpha}$$



グループ0



グループ (N-1)

# GDDI DC-MP2計算の擬似コード<sup>[1]</sup>

```
1: 基底関数の数順に部分系をソート(ロードバランシングのため)
2: Call DDI_SCOPE (DDI_GROUP)
3: Call GDDICOUNT (-1, MYJOB)
4: EMP2TOT ← 0
5: Loop isub=1, nsub ; 部分系のループで並列化 (coarse-grain)
6:   Call GDDICOUNT (0, MYJOB)
7:   If (MYJOB=TRUE) Then
8:     EMP2 ← [MP2 correlation energy of isub subsystem]
           (グループ内 [fine-grain] 並列化を利用)
9:     EMP2TOT ← EMP2TOT + EMP2
10:   End If
11: End Loop
12: Call GDDICOUNT (1, MYJOB)
13: Call DDI_SCOPE (DDI_MASTERS)
14: Call DDI_GSUMF (EMP2TOT)
15: Call DDI_SCOPE (DDI_WORLD)
```

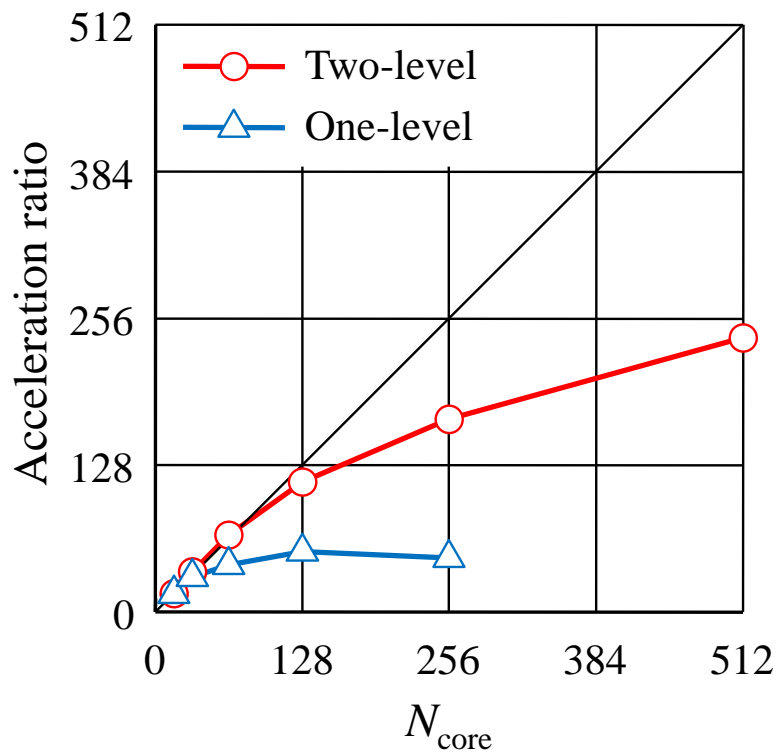
[1] M. Katouda, M. Kobayashi, H. Nakai, and S. Nagase, *J. Comput. Chem.* **32**, 2756 (2011).

# 2段階並列DC-MP2: 並列性能評価

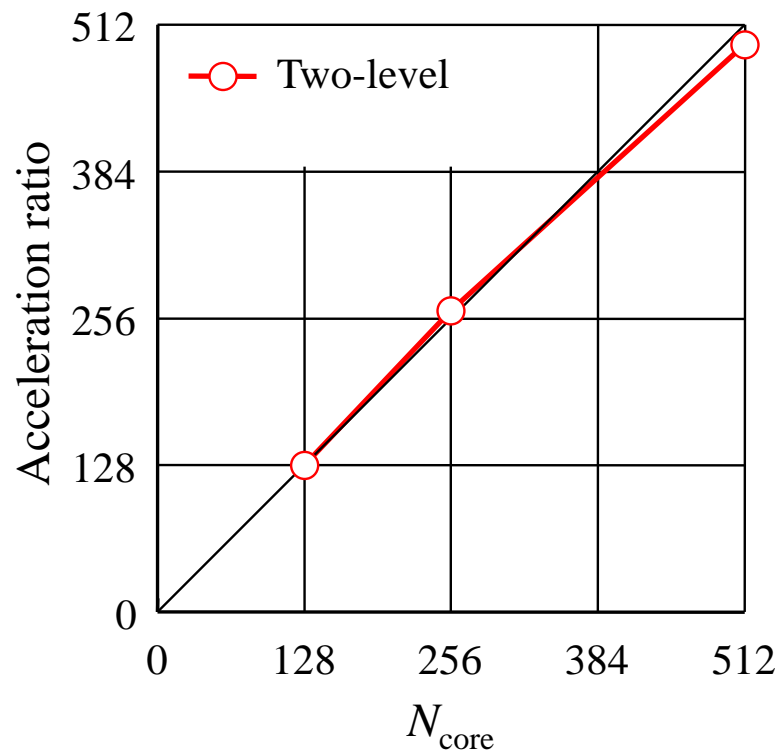
## ■ T2K-Tsukubaでの並列加速度 (OpenMP化はしていない)

基底関数: 6-31G\*  
中央領域: AUTO (4 Å)  
バッファ: 7 Å  
NGROUP =  $N_{\text{core}} / 16$

### – $\beta$ ストランド (Ala)<sub>20</sub>



### – (Ala)<sub>40</sub>



大規模系で特に高い並列計算効率を実現

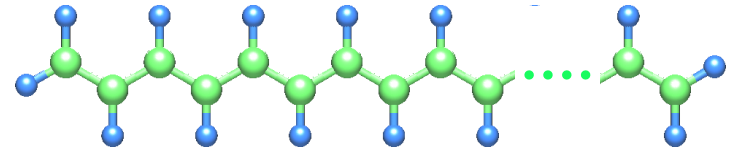
# 2段階並列DC-MP2:「京」での性能評価

## ■ 京での並列加速度

◆ ポリエン鎖  $C_{300}H_{302}$

◆ DC-MP2/6-31G\*

◆ MPI+ARMCI/OpenMP hybrid



部分系:  $C_2H_{2(3)}$  (1ユニット)  
バッファ領域: 左右8ユニット  
NGROUP =  $N_{\text{node}} / 18$   
SERIAL MP2アルゴリズム

$N_{\text{node}}$	$N_{\text{thread}}^*$	FLOPS	計算時間 [s]	$\alpha$ strong
72	504	6.01%	4845	
144	1008	5.80%	2478	98%
288	2016	5.45%	1246	97%
576	4032	3.99%	715	85%
1152	8064	2.36%	468	65%

\*各ノードでARMCIの通信スレッド立ち上がるため、1ノードにつき7スレッド利用

# 講義概要

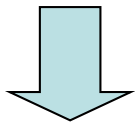
- 7/3 量子化学計算の概要と構成要素、高速化
  - ◆ 量子化学計算の目的と種類
  - ◆ 量子化学計算の手順、構成要素と高速化
  
- 7/10 大規模系に適用するための量子化学計算法
  - ◆ フラグメント分割に基づく方法
    - フラグメント分子軌道(FMO)法
    - 分割統治(DC)法
  - ◆ ラプラス変換MP2法
  - ◆ 2電子積分の密度フィッティング法
    - MP2計算への応用



# Laplace変換MP2法<sup>[1,2]</sup>

■ MP2エネルギー: 
$$\Delta E_{\text{MP2}} = \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \frac{(ia | jb) [2(ia | jb) - (ib | ja)]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

◆ 分母があるので、このままでは $O(N^4)$ よりも小さくできない



■ Laplace変換  $\frac{1}{x} = \int_0^\infty \exp(-xs) ds$  を利用

◆ 分子積分 $(ia | jb)$ も $\Gamma$ になおす [ $O(N^5)$ の積分変換を除去]

$$\Delta E_{\text{MP2}} = - \int_0^\infty \sum_{\gamma\delta\kappa\varepsilon} \sum_{\mu\nu\lambda\sigma} X_{\mu\gamma}(s) Y_{\nu\delta}(s) X_{\kappa\lambda}(s) Y_{\varepsilon\sigma}(s) \Gamma_{\gamma\delta,\kappa\varepsilon} [2\Gamma_{\mu\nu,\lambda\sigma} - \Gamma_{\mu\sigma,\lambda\nu}] ds$$

$$\equiv - \int_0^\infty \sum_{\mu\nu\lambda\sigma} (\underline{\mu}\bar{\nu} | \underline{\lambda}\bar{\sigma}) [2\Gamma_{\mu\nu,\lambda\sigma} - \Gamma_{\mu\sigma,\lambda\nu}] ds$$

$$\mathbf{X}(s) = \sum_i^{\text{occ}} e^{\varepsilon_i s} \mathbf{C}_i \mathbf{C}_i^T$$

$$\mathbf{Y}(s) = \sum_a^{\text{vir}} e^{-\varepsilon_a s} \mathbf{C}_a \mathbf{C}_a^T$$

◆ 積分(数値求積)が必要

[1] M. Häser, *Theor. Chim. Acta* **87**, 147 (1993).

[2] P. Y. Ayala and G. E. Scuseria, *J. Chem. Phys.* **110**, 3660 (1999).

# Laplace変換MP2: 求積法

- 最小二乗法で求積点を決定
    - ◆  $O(N^4)$ の点数に対して実行するのは非効率
  - Minimax法<sup>[1]</sup>
    - ◆ 求積誤差の最大値を最小にする
  - 一般的な求積法
    - ◆ Gauss-Laguerre
      - Exponentialに減衰する $[0, \infty]$ 積分に有効
    - ◆ Euler-Maclaurin法
      - 台形公式の誤差を見積もる方法
    - ◆ Romberg積分
      - Euler-Maclaurin法の誤差への外挿法
- 有限範囲への変数変換が必要

[1] A. Takatsuka, S. Ten-no, and W. Hackbusch, *J. Chem. Phys.* **129**, 044112 (2008).

# Laplace変換MP2: Euler-Maclaurin求積

## ■ 求積を台形公式で実行

$$\int_0^1 f_2(r) dr \approx \frac{1}{\tau+1} \left[ \sum_{k=1}^{\tau} f_2\left(\frac{k}{\tau+1}\right) + \frac{1}{2} [f_2(0) + f_2(1)] \right] \quad f_2(r) = e_2(s) \frac{ds}{dr}$$

◆ 積分範囲を有限にする変数変換が必要

## ■ Euler-Maclaurin法による誤差の見積もり

$$\Delta = \frac{f_2'(0)}{12(\tau+1)^2} - \frac{f_2'''(0)}{720(\tau+1)^4} + \frac{f_2^{(5)}(0)}{30240(\tau+1)^6} - \frac{f_2^{(7)}(0)}{1209600(\tau+1)^8} + \dots$$

$r=0$ でヤコビアンが0になるような変数変換を利用

$$(A) \quad s = \frac{r^2 - (r^3 + r^4 + r^5 + 0.9r^6)/4}{(1-r)^2} \quad \left( \frac{ds}{dr} \Big|_{r=0} = 0 \right)$$

$$(B) \quad s = \frac{r^3 - 0.9r^4}{(1-r)^2} + r^2 \tan(\pi r/2) \quad \left( \frac{ds}{dr} \Big|_{r=0} = 0, \quad \frac{d^2s}{dr^2} \Big|_{r=0} = 0 \right)$$

# Laplace変換MP2: 計算手順

- 求積点ごとに以下を実行 (求積点:  $s$ , 重み:  $w$ )

1. 行列 $\mathbf{X}(s)$ と $\mathbf{Y}(s)$ を求める  $\mathbf{X}(s) = \sum_i^{\text{occ}} e^{\varepsilon_i s} \mathbf{C}_i \mathbf{C}_i^T$   $\mathbf{Y}(s) = \sum_a^{\text{vir}} e^{-\varepsilon_a s} \mathbf{C}_a \mathbf{C}_a^T$

2. Schwarzのスクリーニングに用いる行列を求める

3. 各 $\kappa\varepsilon$ に対し、 $(\underline{\mu}\bar{\nu} | \kappa\varepsilon)$ を求めてディスクに保存

- $\Gamma$ を計算し、 $(\underline{\mu}\delta | \kappa\varepsilon) = \sum_{\gamma} X_{\mu\gamma} \Gamma_{\gamma\delta, \kappa\varepsilon}$ を足しこみ

- $(\underline{\mu}\bar{\nu} | \kappa\varepsilon) = \sum_{\delta} Y_{\nu\delta} (\underline{\mu}\delta | \kappa\varepsilon)$ を足しこみ

4.  $(\underline{\mu}\bar{\nu} | \underline{\lambda}\varepsilon) = \sum_{\kappa} X_{\lambda\kappa} (\underline{\mu}\bar{\nu} | \kappa\varepsilon)$ を足しこみ

5.  $(\underline{\mu}\bar{\nu} | \underline{\lambda}\bar{\sigma}) = \sum_{\sigma} Y_{\sigma\varepsilon} (\underline{\mu}\bar{\nu} | \underline{\lambda}\varepsilon)$ を足しこみ

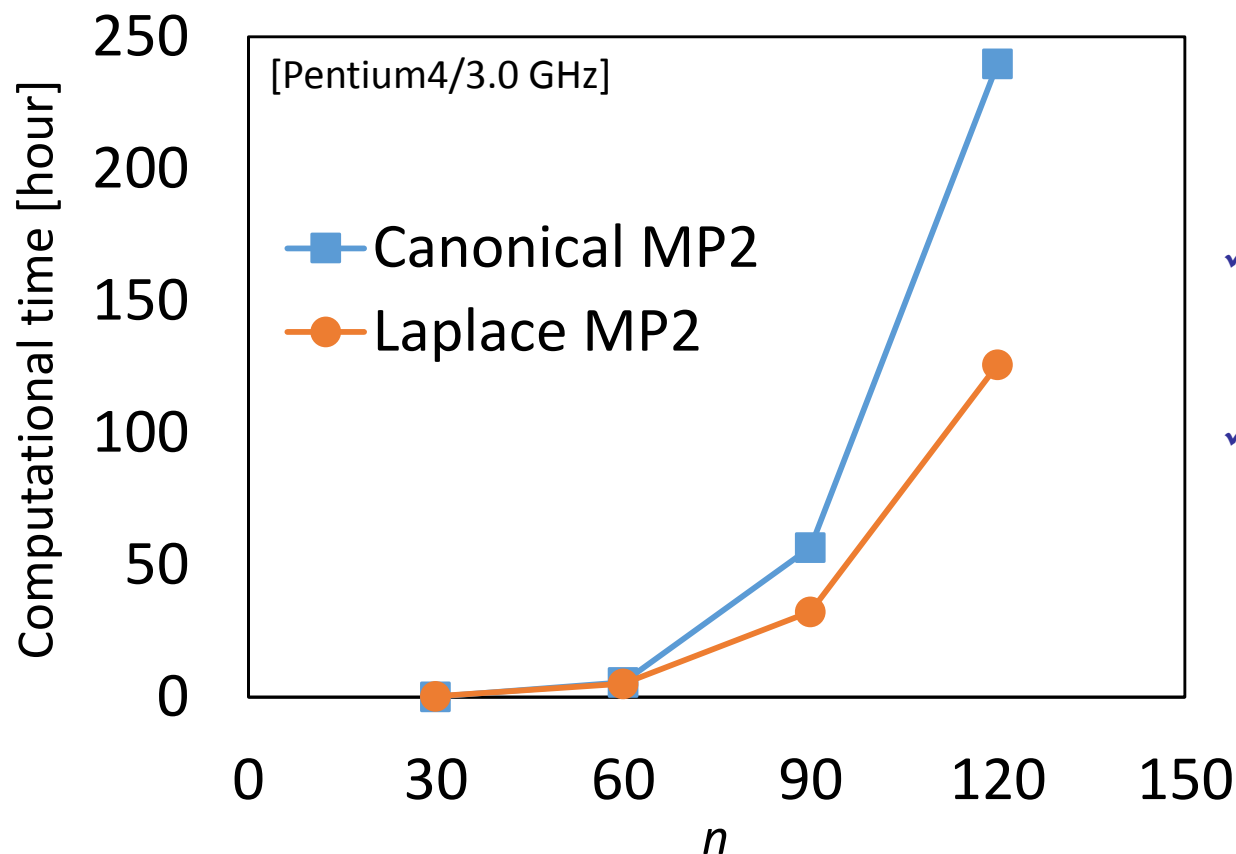
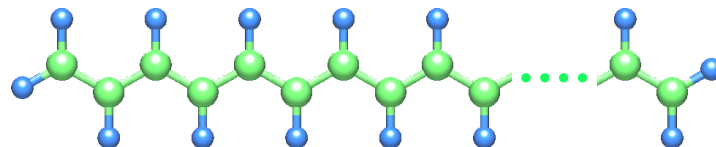
6.  $[2\Gamma_{\mu\nu, \lambda\sigma} - \Gamma_{\mu\sigma, \lambda\nu}]$ を求めてエネルギーに足しこみ

3~6でSchwarzの  
不等式を利用した  
スクリーニング

# Laplace変換MP2: 計算時間

■ MP2計算時間

◆ ポリエン鎖  $C_nH_{n+2}$



✓ 計算時間の削減に成功

✓  $n$ に対するスケールリングも改善

# Laplace変換MP2: 求積法の精度

## ■ MP2相関エネルギーの求積法依存性

◆ ベンゼン/6-31G\*

[Hartree]

求積法	求積点数	$E_{\text{corr}}$	(diff.)
Gauss-Laguerre	5	-0.733451	(+0.051311)
Euler-Maclaurin (A)	5	-0.770241	(+0.014521)
Euler-Maclaurin (B)	5	-0.784540	(+0.000221)
Romberg (A)	7	-0.784643	(+0.000118)
Romberg (B)	7	-0.783803	(+0.000958)
Canonical MP2		-0.784761	

✓ Euler-Maclaurin (B)やRombergが良い結果

✓ 誤差解析の結果にも対応

## 2電子積分の密度フィッティング(RI近似)

■ 2電子積分  $\Gamma_{\mu\nu,\lambda\sigma} = \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1)r_{12}^{-1}\phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2)$

◆ メモリにストアすることは困難 (4階テンソル)

■ 原子軌道の積  $\phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})$  を補助基底関数で展開

$$\phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r}) \approx \sum_m d_m^{\mu\nu} \varphi_m(\mathbf{r}) \equiv \theta_{\mu\nu}(\mathbf{r})$$

◆ 誤差の自己反発積分を最小化するように決定

$$\iint d\mathbf{r}_1 d\mathbf{r}_2 \frac{R_{\mu\nu}(\mathbf{r}_1)R_{\mu\nu}(\mathbf{r}_2)}{r_{12}} \rightarrow \text{Min} \quad R_{\mu\nu}(\mathbf{r}) = \psi_\mu(\mathbf{r})\psi_\nu(\mathbf{r}) - \theta_{\mu\nu}(\mathbf{r})$$

➡  $d_m^{\mu\nu} = \sum_n (m|n)^{-1}(n|\mu\nu)$

◆ まとめると  $\Gamma_{\mu\nu,\lambda\sigma} = \sum_{m,n} (\mu\nu|m)(m|n)^{-1}(n|\lambda\sigma)$

2階・3階のテンソルの積和

# 並列RI-MP2アルゴリズム<sup>[1]</sup>

■ MP2エネルギー: 
$$\Delta E_{\text{MP2}} = \sum_{i,j}^{\text{occ}} \sum_{a,b}^{\text{vir}} \frac{(ia | jb) [2(ia | jb) - (ib | ja)]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

◆  $(m | n)$ は正定値  $\rightarrow$  Cholesky分解  $(m | n) = \sum_l L_{ml} L_{ln}^T$  (LAPACK)

◆ 積分変換  $(ia | jb) = \sum_{\mu\nu\lambda\sigma} C_{\mu i} C_{\nu a} C_{\lambda j} C_{\sigma b} \Gamma_{\mu\nu,\lambda\sigma}$  を考慮

$$(ia | jb) = \sum_n B_n^{ia} B_n^{jb} \quad B_n^{ia} = \sum_l L_{nl}^{-1} \sum_{\mu\nu} C_{\mu i} C_{\nu a} (\mu\nu | l) \equiv \sum_l L_{nl}^{-1} (ia | l)$$

## ■ 並列RI-MP2アルゴリズム

1.  $(m | n)$ を計算し、Cholesky分解で $L^{-1}$ を計算して保存

2.  $(ia | l)$ を計算 ( $l$ に対して動的並列化)

3.  $B_n^{ia}$ を計算 ( $il$ に対して静的並列化)

4.  $(ia | jb)$ を計算 ( $ijl$ に対して静的並列化)、MP2エネルギー積算

$(ia | l)$ とBのデータ分散も

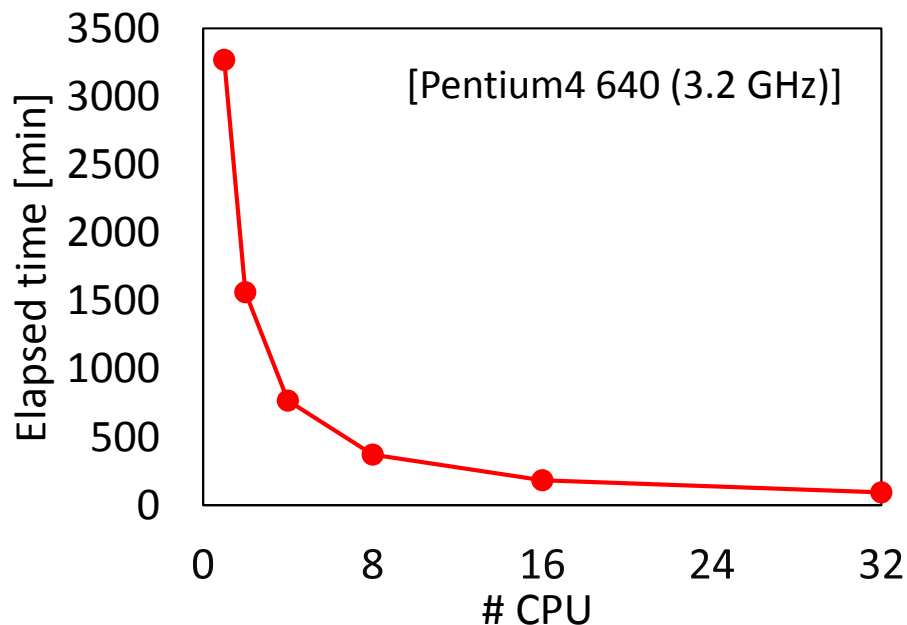
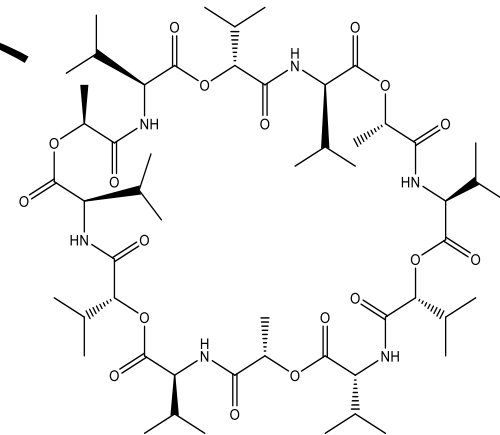
[1] M. Katouda and S. Nagase, *Int. J. Quantum Chem.* **109**, 2121 (2009).



# RI-MP2法: 計算コスト

## ■ MP2計算時間とメモリ容量<sup>[1]</sup>

◆ バリノマイシン(右図) / 6-311G\*\*



	メモリ [MB/node]	スクラッチ [GB]
RI-MP2	1206.1	32.4
Canonical	2401.4	639.3

✓リソース量を大幅削減

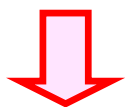
- ✓スーパーリニアスケーリングを達成
- ✓RI近似による誤差は $0.118 mE_h$

FMO法やDC法と組み  
合わせて利用し、  
更なる高速化も可能

[1] M. Katouda and S. Nagase, *Int. J. Quantum Chem.* **109**, 2121 (2009).

# 本日のまとめ(1)

- 「京」を使うだけでは大規模量子化学計算は不可能



- 計算コストのオーダーを削減するさまざまな手法

- ◆ フラグメント分割法 (FMO, DC)

- フラグメントの計算結果を足し合わせて全体の結果を得る
- 大規模並列化が可能

- ◆ Laplace変換MP2法

- 摂動論で現れるエネルギー分母をLaplace変換で消去
- Schwarz不等式等のカットオフを利用してオーダー削減

- ◆ 2電子積分の密度フィッティング法 (RI法)

- 4階のテンソルを3階以下のテンソルの積和で表現
- 計算リソースを大幅削減 (時間オーダーは不変)

# 本日のまとめ(2)

## ■ 並列化を見据えたアルゴリズムの改善

### ◆ DC法におけるFermi準位決定

- 計算時間はごく短い、DFTB法では問題に
- 一見非効率な方法も、並列化した場合には良いことも

### ◆ 2段階並列アルゴリズム

- 計算を大粒度で並列化し、その中身をさらに細粒度並列
- フラグメント分割計算では非常に有効

### ◆ データ分散が可能なアルゴリズム

- うまく組めばスーパーリニアな並列効率が出る場合も