



4章

密度汎関数法計算の線形スケーリング法

4.1. Coulomb積分計算の高速化 SCF計算の高速化

SCFプロセスとその線形スケーリング法

1

Coulomb積分

形式的には $O(N^4)$ にスケールされるが、大規模分子計算では、標準的な積分スケーリングにより $O(N^2)$ にまでは近づく。

2

交換積分

同様に、標準的な積分スケーリングにより $O(N^2)$ になる。線形スケーリングについては、FMMにもとづく「近接領域交換(NFX)」や電子密度が核から遠距離で指数関数的に減衰することを仮定するOrder-N交換(ONX)などがあるが、決定的な方法はない。

3

行列の対角化

かなり大規模な分子の計算では律速になる（後述）。

4

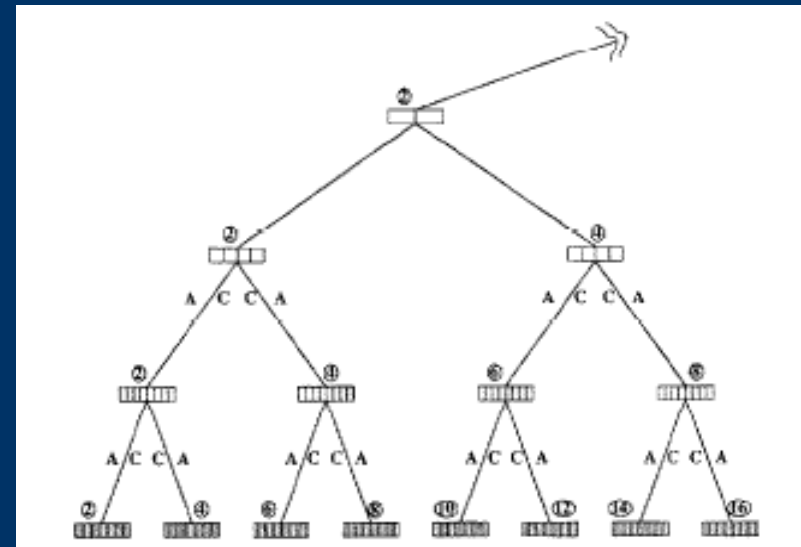
その他

それ以外の1電子積分などは、計算コスト的に重要でない。

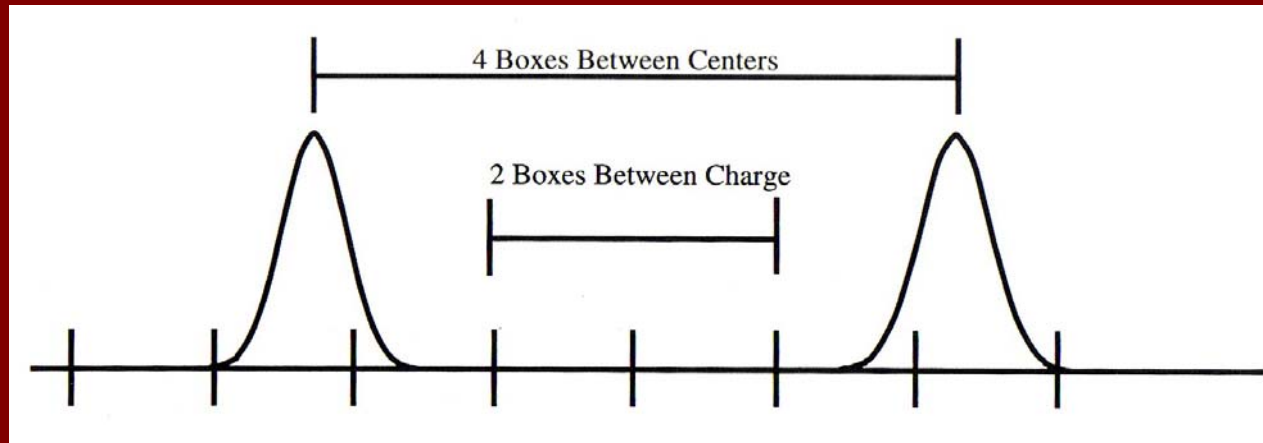
Fast Multipole法

FMMの原型 [Greengard and Rokhlin, J. Comput. Phys., 73, 325, 1987.]

1. 全粒子を正立方体の計算boxの中に入れる。
2. 計算boxを1/2に分割。分割前を親box、後を子boxと呼ぶ。
3. 分割を繰り返してtree構造を作る。2を行なった回数がレベル。
4. 最低レベルのboxに含まれる粒子数が一定になったら終了。
5. 各最低レベルのboxにある粒子の中の電荷の多極展開をboxの中心について行なう。
6. 「上方経路」: 各子boxの多極展開を変換演算子により親boxの中心に変換する(Pass 1)。その変換モーメントは、各親boxについて足され、貯められる。
7. レベル2まで6のプロセスを繰り返す。
8. 「下方経路」: 遠距離boxとの間の多極展開をそのboxの中心に関する局所的Taylor様展開に変える(Pass 2)。この局所展開を全ての遠距離boxについて計算し、足し合わせて、遠距離領域のポテンシャルとして算出される。



Fast Multipole法



9. もし、対象のboxの親boxがそのboxの親boxとよく離れていないならば、局所展開されずは無視される。この情報は親boxから子boxに受け継がれる(Pass 3)。このプロセスは最低レベルに達するまで行なわれる。

10. 最低レベルのboxにおいて、box内の各粒子で局所的Taylor展開を計算。これらの点での遠距離ポテンシャルを得る(Pass 4)。これは、そのboxから「よく離れた(well-separated)」全粒子の効果を表現。

11. ある粒子とそのboxあるいは近接box内にある粒子間の相互作用を計算。多極展開は妥当でないので、直接Coulomb相互作用を計算($O(N^2)$)。



点電荷分布に関するアルゴリズムの連続電荷分布への一般化が必要

連続FMM

連続FMM(CFMM)

[White, Johnson, Gill and Head-Gordon, CPL, 230, 8, 1994.]

改良点：「well-separated」の再定義
空間的広がりをもつ電荷分布を取り扱う
ため、「分布の程度(extent)」という指標
を導入して再定義。

Coulomb相互作用を近距離領域と遠距離
領域に分類

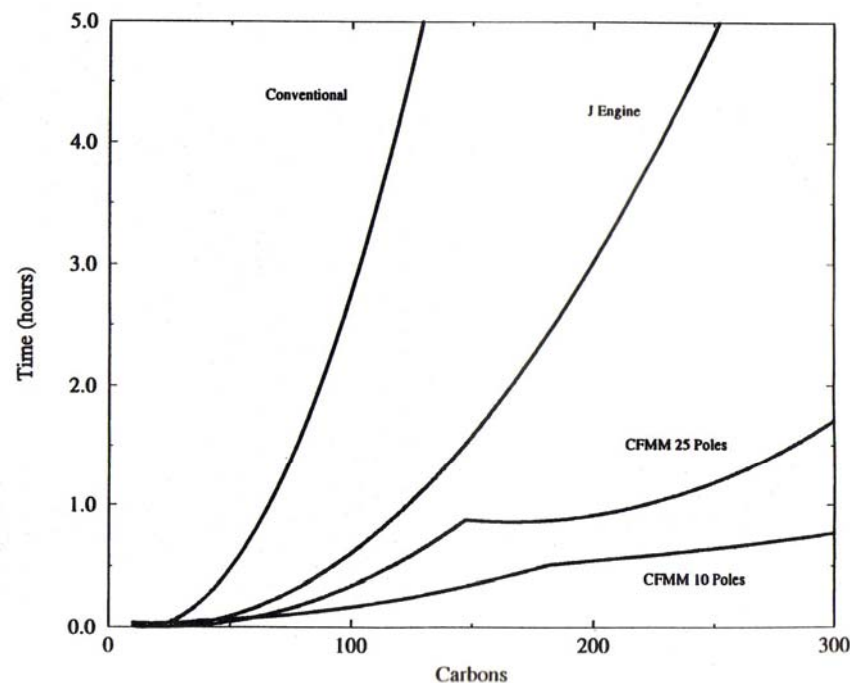
近距離領域では、そのままCoulomb反発
積分として計算
遠距離領域では、Legendre展開を用いて
取り扱う

$$r_{12}^{-1} = \frac{1}{r_1} \sum_{k=0}^{\infty} \left(\frac{r_2}{r_1} \right)^k P_k \left(\frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{r_1 \cdot r_2} \right)$$

$r_1 > r_2$ (他の球と重なってない球内の粒
子) のときのみ収束

問題点：わずかに重なっても、有意な漸
近展開を算出

Fock行列のCoulomb項を
計算するのに要したCPU時間
の比較



KWIK法

KWIK法

[Dombroski, Taylor and Gill, JPC, 100, 6272, 1996;
Gill, CPL, 270, 193, 1997.]

r_{12}^{-1} を誤差関数で分割

$$r_{12}^{-1} \equiv S(r_{12}) + L(r_{12}) = \frac{1 - \operatorname{erf}(r_{12})}{r_{12}} + \frac{\operatorname{erf}(r_{12})}{r_{12}}$$

長距離部分をTaylor展開

$$L(r_{12}) = \sum_{k=0}^{\infty} \frac{L^{(2k)}(0)}{(2k)!} (r_1^2 + r_2^2 - 2r_1 \cdot r_2)^k$$

Coulombエネルギー（自己相互作用抜き≠）

$$\left\{ \begin{aligned} E_J &= \frac{1}{2} \langle \rho | S | \rho \rangle_{\neq} + \sum_{k=0}^{\infty} \frac{L^{(2k)}(0)}{(2k)!} W_{2k} \\ W_{2k} &= \frac{1}{2} \sum_{p=0}^k \sum_{q=0}^{k-p} \frac{(-2)^p k!}{p! q! (k-p-q)!} \left\langle r^{2(k-p-q)} (r_{\alpha} r_{\beta} \cdots)_p \right\rangle \left\langle r^{2q} (r_{\alpha} r_{\beta} \cdots)_p \right\rangle \end{aligned} \right.$$

W項の具体的な形

$$W_0 = (\mathbf{M}_0^2 - \sum Z_I^2) / 2, \quad W_2 = \mathbf{M}_0 \mathbf{M}_{ii} - \mathbf{M}_i \mathbf{M}_i, \quad \dots$$

$$\mathbf{M}_0 = \int \rho(\mathbf{R}) d^3 \mathbf{R}, \quad \mathbf{M}_i = \int \mathbf{r}_i \rho(\mathbf{R}) d^3 \mathbf{R}, \quad \mathbf{M}_{ij} = \int \mathbf{r}_i \mathbf{r}_j \rho(\mathbf{R}) d^3 \mathbf{R}$$

4.2. 行列対角化計算の高速化 密度汎関数法における計算オーダー

密度汎関数法計算の主要プロセスと計算オーダー

1

Coulomb積分計算

最近の方法論的發展により $O(N)$ に（既述）。

2

交換・相関積分計算

1. 求積法のグリッドの重率計算
2. 基底関数値計算
3. 電子密度計算
4. 交換相関汎関数計算

空間のcutoffで求積重率計算は $O(N)$ に。

それ以外も有効なスクリーニング法により $O(N)$ に（後述）。

3

Fock行列対角化計算

1. 原子軌道（AO）から規格直交基底へのFock行列の変換
2. 行列対角化
3. 固有ベクトルからAO基底への逆変換
4. 固有ベクトルから密度の形成

非常に大規模な系の計算でなければ律速にならないため、むしろ記憶容量 $O(N^2)$ が問題とされてきた。

密度行列の直接探索法

対角化の回避 [Milliam and Scuseria, JCP, 106, 5569, 1997;
X. Li, W. Nunes, and D. Vanderbilt, PRB, 37, 785, 1988.]

物性物理における対角化回避 $O(N)$ 法

大量の平面波基底→対角化が影響
行列対角化回避の方法

1. 局所化基底関数
2. 直接的なエネルギー最小化
 - (1) 局所化分子軌道の探索
 - (2) 直接的な密度行列の探索



非直交基底に対する直接的エネルギー
最小化法を密度行列探索法により開発

密度行列とFock行列の規格直交基底への変換および逆変換

Cholesky分解：正行列を次数の異なる三角行列の積に分解($S=U^T U$ の対角化不要)

$$P_{\text{ortho}} = U P_{\text{AO}} U^T$$
$$F_{\text{ortho}} = U F_{\text{AO}} U^T$$

密度行列探索 (DMS) 法

N 表現可能性を課したエネルギー最小化

• N 表現可能な密度行列：

1. 電子数： $\text{Tr}(PS) = N$
2. ベキ等： $PSP = P$

P は密度行列、 S は重なり行列、 N は電子数

• 密度行列とFock行列

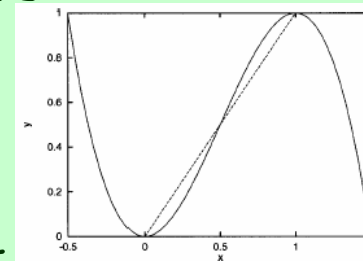
は交換可能：

$$SPF = FPS$$

McWeenyの純化法

$$P \sim = 3P^2 - 2P^3$$

もし P がベキ等に近
いなら、 $P \sim$ は P より
さらにベキ等に近くなる



共役勾配法と準Newton-Raphson法

共役勾配 (CG) 法

関数値とその勾配しか必要ないため最適

→ Newton-Raphson法などはHessian行列の計算・保存・逆変換が必要

CG法におけるエネルギー最小化のアルゴリズム

$$\mathbf{G}_0 = \mathbf{H}_0 = -\nabla E(\mathbf{P}_0)$$

do $i = 0$, maximum iterations

$$\mathbf{P}_{i+1} = \mathbf{P}_i + \lambda_i \mathbf{H}_i$$

$$\mathbf{G}_{i+1} = -\nabla E(\mathbf{P}_{i+1})$$

$$\gamma_i = \frac{(\mathbf{G}_{i+1} - \mathbf{G}_i) \cdot \mathbf{G}_{i+1}}{\mathbf{G}_i \cdot \mathbf{G}_i}$$

$$\mathbf{H}_{i+1} = \mathbf{G}_{i+1} + \gamma_i \mathbf{H}_i$$

enddo

準Newton-Raphson法のDIISを使う方法 [X. Li et al., JCP, 119, 7651, 2003.]

$$H_{\mu\nu\mu\nu} = d^2 E(\tilde{P}) / dP_{\mu\nu}^2$$

$$\begin{aligned} &= (3 - 2\mathbf{P}_{\mu\mu})\mathbf{F}_{\nu\nu} + (3 - 2\mathbf{P}_{\nu\nu})\mathbf{F}_{\mu\mu} \\ &\quad - 4(\mathbf{PF})_{\mu\mu} - 4(\mathbf{PF})_{\nu\nu} \end{aligned}$$

E はエネルギー汎関数
 H_i は探索方向
 λ_i は最近接極小点へのステップ長さ

これだけでは $O(N)$ にならない

→ 行列のsparsityを利用

標準的な対角化法では行列のsparsityが不保持

疎行列計算法で重要な要素のみ保存して計算

→ 分布が疎な系なら $O(N)$ に

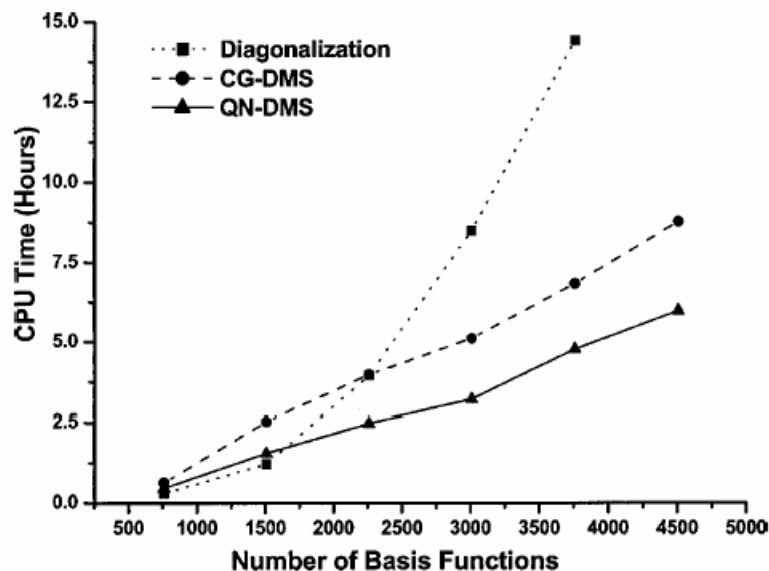
Hessian行列の計算・保存はネックではない

DIIS法を利用する利点は大きい

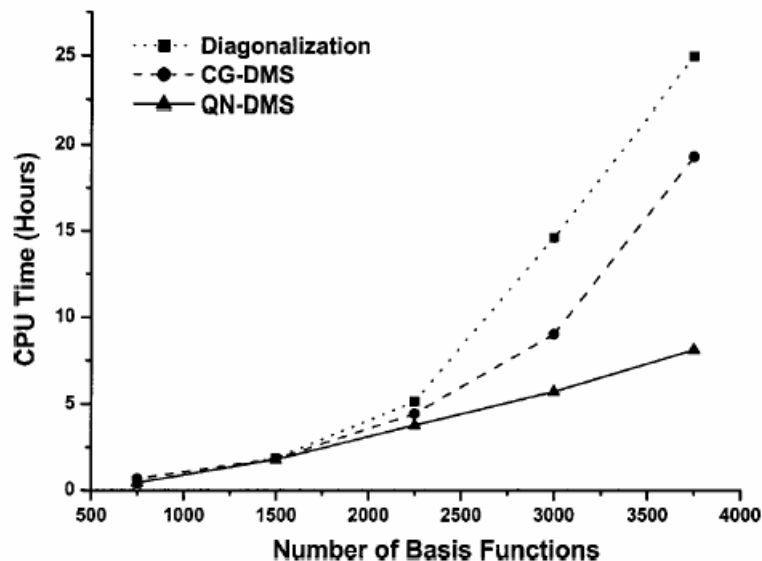
DIIS法に必要な誤差ベクトルの初期値として、

DMSで作れるHessian行列の対角項を利用

CPU時間の比較



2次元系であるポリグリシン鎖
のLDA/6-31G(d,p)計算
における行列対角化に要した
CPU時間



3次元系である水クラスタの
LDA/6-31G(d,p)計算
における行列対角化に要した
CPU時間

[X. Li et al., JCP, 119, 7651, 2003.]

4.3. 交換相関汎関数数値積分計算の効率化 密度汎関数法における計算オーダー

密度汎関数法計算の主要プロセスと計算オーダー

1

Coulomb積分計算

最近の方法論的発展により $O(N)$ に（既述）。

2

交換・相関汎関数数値積分計算

1. 求積法のグリッドの重率計算
2. 基底関数値計算
3. 電子密度計算
4. 交換相関汎関数計算

空間のcutoffで求積重率計算は $O(N)$ に。

それ以外も有効なスクリーニング法により $O(N)$ に。

3

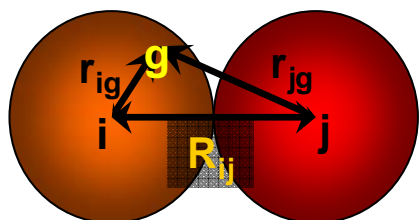
Fock行列対角化計算

1. 原子軌道（AO）から規格直交基底へのFock行列の変換
2. 行列対角化
3. 固有ベクトルからAO基底への逆変換
4. 固有ベクトルから密度の形成

物性物理で使われてきた行列対角化回避法で $O(N)$ 近くに（既述）。

重率の設定

数値積分グリッドの重率 [A. D. Becke, JCP, 88, 2547, 1988;
R. E. Stratmann, G. E. Scuseria and M. J. Frisch, CPL, 257, 213, 1996.]



3次元積分の数値計算

$$I \equiv \int F(\mathbf{r}) d^3\mathbf{r}$$

$$= \sum_k^{\text{atoms}} \sum_g^{\text{grid}} \omega_g p_k(\mathbf{r}_g) F(\mathbf{r}_g)$$

Becke fuzzy cell重率

共焦楕円座標: $\mu_{ij} = (r_{ig} - r_{jg}) / R_{ij}$

奇パリティの3次多項式

$$h(\mu_{ij}) = \frac{3}{2}\mu_{ij} - \frac{1}{2}\mu_{ij}^3 \quad \begin{cases} h(1) = 1, & h(-1) = -1, \\ h'(1) = 0, & h'(-1) = 0 \end{cases}$$

原子k中のグリッド点 \mathbf{r}_g の規格化重率

$$p_k(\mathbf{r}_g) = w_k(\mathbf{r}_g) / \sum_j w_j(\mathbf{r}_g)$$

$$w_k(\mathbf{r}_g) \equiv \prod_{i \neq k}^{\text{atoms}} s(\mu_{ij}) = \prod_{i \neq k}^{\text{atoms}} \frac{3}{2} [1 - g(\mu_{ij})]$$

$$g(\mu_{ij}) = h\{h[h(\mu_{ij})]\}$$

Stratmann-Scuseria重率

セル関数gの再定義

$$g(\mu_{ij}) = z(\mu_{ij}; a) \quad \begin{cases} -1 & (\mu_{ij} \leq -a) \\ -a < \mu_{ij} < a \\ 1 & (\mu_{ij} \geq a) \end{cases}$$

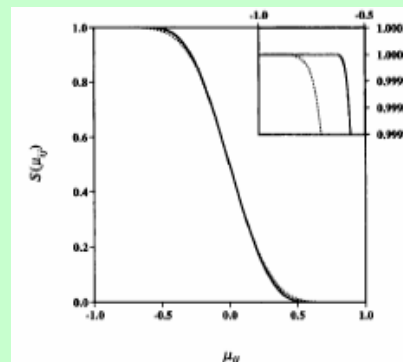
$$\begin{cases} z(\mu_{ij} = a) = 1, & z(\mu_{ij} = -a) = -1, \\ z'(\mu_{ij} = a) = 0, & z'(\mu_{ij} = -a) = 0 \end{cases}$$

新しい多項式(±aで導関数がゼロ)

$$z(\mu_{ij}; a) = \frac{1}{16} \left[35(\mu_{ij}/a) - 35(\mu_{ij}/a)^3 + 21(\mu_{ij}/a)^5 - 5(\mu_{ij}/a)^7 \right]$$

$m_{ij} = \pm a$ で2次、
3次の導関数が
ゼロになる関数

a=0.64が経験的
な最適値



基底関数やグリッドによるスクリーニング

汎関数数値積分計算のプロセスと線形スケーリングの手法

1

重率計算によるスクリーニング

各グリッドで重率ゼロの原子を計算から除外（既述）。

2

Gauss型基底関数のtailの減衰の利用（B. G. JohnsonのPhD論文）

グリッド点ごとに基底関数値を考え、数値的に重要なものだけ計算。各基底関数の有効範囲を閾値 ε で決まる半径の球とする。

1. 各グリッド点で $|\phi_\alpha(r)| > \varepsilon$ の基底関数のリストを作成する。
2. ある原子の全基底関数が与えられたグリッド点でのリストからはずれるなら、その原子の重率をゼロとして除外する。

3

グリッド圧縮

空間的に近いグリッド点の座標をそれぞれ小バッチにまとめ、連続したメモリの番地に割り当てる。

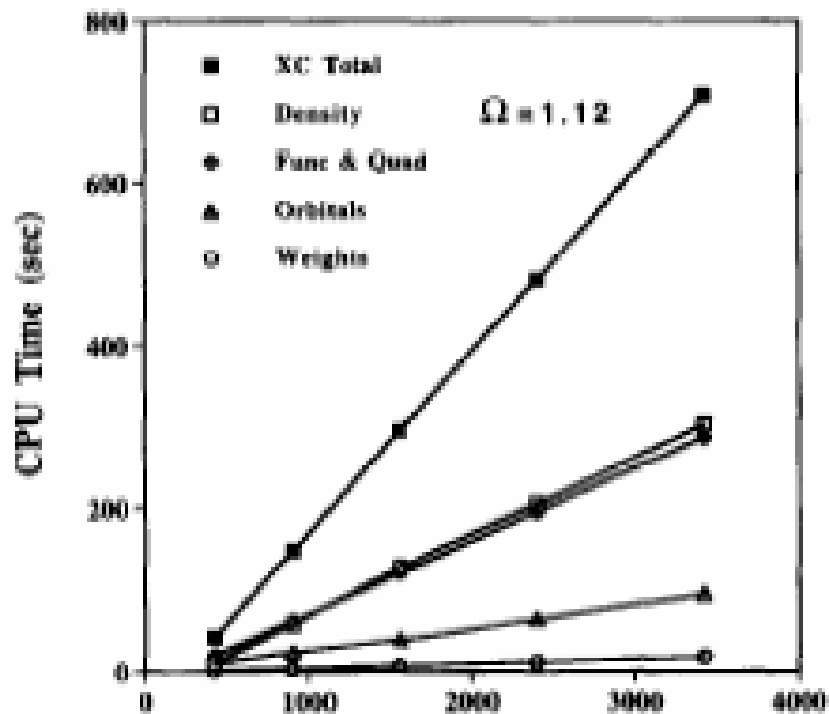
圧縮の際に、無視できる重率をもつグリッド点は除外。

→全グリッド点数が減る。＝「グリッド圧縮」

結果的に、メモリ容量の削減だけでなく、残りのプロセスのスピードアップや効率化にも有効。

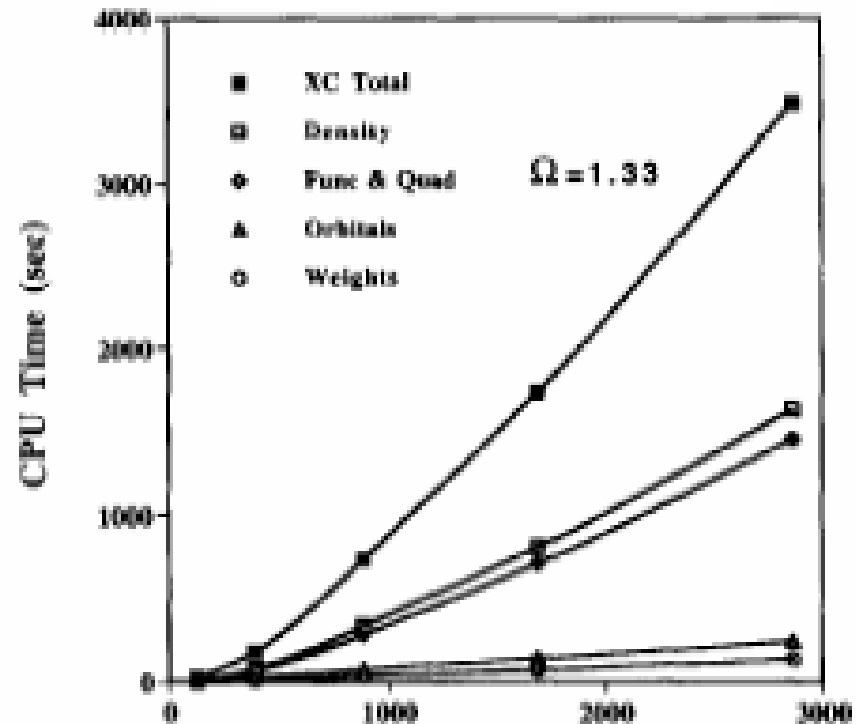
CPU時間

グラフィート板 (2次元)



Number of Basis Functions

ダイヤモンド塊 (3次元)



Number of Basis Functions

[R. E. Stratmann, G. E. Scuseria and M. J. Frisch, CPL, 257, 213, 1996.]