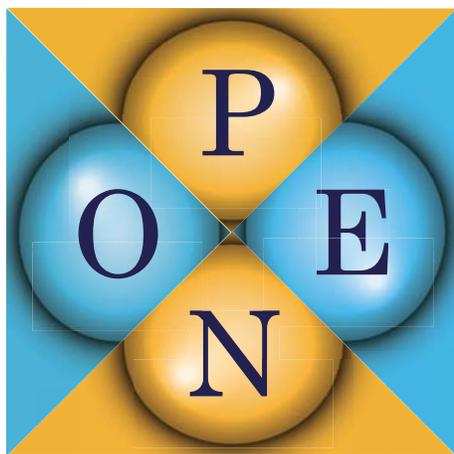


# OpenMX Ver. 3.9 ユーザーマニュアル



## 開発貢献者

尾崎泰助 (東京大学)  
木野日織 (物質・材料研究機構)  
J. Yu (ソウル大学校)  
M.J. Han (KAIST)  
大淵真理 (富士通研究所)  
石井史之 (金沢大学)  
澤田啓介 (金沢大学)  
久保田雄士 (金沢大学)  
水田耀ピエール (大阪大学)  
小鷹浩毅 (京都大学)  
山口直也 (金沢大学)  
澤端日華瑠 (金沢大学)  
T.B. Prayitno (Univ. Negeri Jakarta)  
大脇創 (日産アーク)  
T.V.T Duy (アイシン精機)  
宮田全展 (北陸先端科学技術大学院大学)  
G. Jiang (武漢科技大学)  
P.-H. Chang (George Mason Univ.)

寺澤麻子 (東京工業大学)  
合田義弘 (東京工業大学)  
H. Weng (CAS)  
椎原良典 (豊田工業大学)  
豊田雅之 (東京工業大学)  
奥野幸洋 (富士フイルム)  
R. Perez (UAM)  
P.P. Bell (UAM)  
M. Ellner (UAM)  
Yang Xiao (NUAA)  
伊藤篤史 (核融合科学研究所)  
大谷実 (産業技術総合研究所)  
河村光晶 (東京大学)  
吉見一慶 (東京大学)  
C.-C. Lee (Tamkang Univ.)  
Y.-T. Lee (Academia Sinica)  
福田将大 (東京大学)  
S. Ryee (KAIST)  
寺倉清之 (産業技術総合研究所)

December 9, 2022

# Contents

|       |                                  |    |
|-------|----------------------------------|----|
| 1     | OpenMX について                      | 8  |
| 2     | 関連文献                             | 11 |
| 3     | インストール                           | 13 |
| 3.1   | 必須ライブラリ                          | 13 |
| 3.2   | シリアル版                            | 13 |
| 3.3   | MPI 版                            | 13 |
| 3.4   | MPI/OpenMP ハイブリッド並列版             | 14 |
| 3.5   | FFTW3                            | 14 |
| 3.6   | その他のオプション                        | 15 |
| 3.6.1 | -Dnosse                          | 15 |
| 3.6.2 | -Dkcomp                          | 15 |
| 3.7   | プラットフォーム                         | 15 |
| 3.8   | インストールの際のヒント                     | 15 |
| 3.9   | make に関するオプション                   | 17 |
| 4     | テスト計算                            | 18 |
| 5     | 自動実行のテスト                         | 25 |
| 6     | 大規模な系の自動実行テスト                    | 28 |
| 7     | 入力ファイル                           | 30 |
| 7.1   | 例：メタン分子                          | 30 |
| 7.2   | キーワード                            | 31 |
| 8     | 出力ファイル                           | 46 |
| 9     | 汎関数                              | 49 |
| 10    | 基底関数                             | 50 |
| 10.1  | 概要                               | 50 |
| 10.2  | プリミティブ基底関数                       | 50 |
| 10.3  | データベース Ver. 2019 により提供される最適化基底関数 | 51 |
| 10.4  | ユーザーによる PAO の最適化                 | 53 |
| 10.5  | 空原子の配置                           | 53 |
| 10.6  | PAO および VPS ファイルを保存するディレクトリの指定   | 54 |
| 11    | 擬ポテンシャル                          | 57 |
| 11.1  | 従来型擬ポテンシャル                       | 57 |
| 11.2  | オープンコア型擬ポテンシャル                   | 58 |
| 11.3  | 内殻準位励起のための擬ポテンシャル                | 59 |

|           |                                     |           |
|-----------|-------------------------------------|-----------|
| <b>12</b> | <b>カットオフエネルギー：数値積分のための計算グリッドの設定</b> | <b>60</b> |
| 12.1      | 収束                                  | 60        |
| 12.2      | バルクのエネルギー曲線計算に関するヒント                | 61        |
| 12.3      | 構造格子の相対位置の固定                        | 61        |
| <b>13</b> | <b>SCF 収束</b>                       | <b>63</b> |
| 13.1      | 概要                                  | 63        |
| 13.2      | Kerker 因子の自動決定                      | 66        |
| 13.3      | SCF 収束パラメータの on-the-fly での調整        | 66        |
| <b>14</b> | <b>再スタート</b>                        | <b>67</b> |
| 14.1      | 概要                                  | 67        |
| 14.2      | MD および構造最適化中の外挿法                    | 67        |
| 14.3      | 再スタート計算用の入力ファイル                     | 68        |
| <b>15</b> | <b>構造最適化</b>                        | <b>69</b> |
| 15.1      | 最急降下法                               | 69        |
| 15.2      | EF、BFGS、RF、DIIS 最適化                 | 70        |
| 15.3      | RF 及び EF 最適化におけるヘシアン行列の初期値          | 71        |
| 15.4      | 制約条件付の構造最適化                         | 72        |
| 15.5      | 構造最適化の再スタート                         | 73        |
| <b>16</b> | <b>セル可変構造最適化</b>                    | <b>74</b> |
| 16.1      | 概要                                  | 74        |
| 16.2      | ストレステンソル                            | 76        |
| 16.3      | 格子ベクトルに対する拘束                        | 76        |
| 16.4      | エンタルピーの最適化                          | 77        |
| <b>17</b> | <b>分子動力学</b>                        | <b>79</b> |
| 17.1      | 定エネルギーの分子動力学 (NVE)                  | 79        |
| 17.2      | 速度スケールリングによる NVT 分子動力学 (NVT_VS)     | 79        |
| 17.3      | Nose-Hoover 法による NVT 分子動力学 (NVT_NH) | 80        |
| 17.4      | 多重熱浴分子動力学 (NVT_VS)                  | 82        |
| 17.5      | 制約条件付き分子動力学                         | 82        |
| 17.6      | 初速度                                 | 82        |
| 17.7      | ユーザーによる原子の質量の定義                     | 83        |
| 17.8      | ファイルフォーマットの変換: md2axsf              | 83        |
| <b>18</b> | <b>可視化</b>                          | <b>84</b> |
| <b>19</b> | <b>バンド分散</b>                        | <b>85</b> |
| <b>20</b> | <b>状態密度</b>                         | <b>88</b> |
| 20.1      | 通常の方法                               | 88        |
| 20.2      | 多数の k 点で計算する場合                      | 90        |

|      |                         |     |
|------|-------------------------|-----|
| 21   | 全エネルギーの軌道分解             | 92  |
| 22   | 軌道の最適化                  | 94  |
| 23   | $O(N)$ 法                | 98  |
| 23.1 | 分割統治法 (DC 法)            | 98  |
| 23.2 | 局在自然軌道による分割統治法 (DC-LNO) | 101 |
| 23.3 | $O(N)$ Krylov 部分空間法     | 105 |
| 23.4 | ユーザーによる FNAN+SNAN の定義   | 107 |
| 24   | MPI 並列化                 | 109 |
| 24.1 | $O(N)$ 計算               | 109 |
| 24.2 | クラスター計算                 | 109 |
| 24.3 | バンド計算                   | 109 |
| 24.4 | 完全な 3 次元並列化             | 111 |
| 24.5 | 使用可能な MPI プロセス数         | 111 |
| 25   | MPI/OpenMP ハイブリッド並列化    | 111 |
| 26   | 大規模計算                   | 112 |
| 26.1 | 通常の固有値解法                | 112 |
| 26.2 | $O(N)$ 法と通常の固有値解法の組み合わせ | 112 |
| 27   | 電場                      | 115 |
| 28   | 電荷ドーピング                 | 116 |
| 29   | 分数核電荷を持つ仮想原子            | 117 |
| 30   | LCAO 係数                 | 118 |
| 31   | 分子軌道                    | 119 |
| 32   | 電荷解析                    | 121 |
| 32.1 | Mulliken 電荷             | 121 |
| 32.2 | Voronoi 電荷              | 122 |
| 32.3 | 静電ポテンシャルフィッティング         | 122 |
| 33   | 自然電子密度解析                | 125 |
| 34   | ノンコリニア DFT              | 127 |
| 35   | 相対論的效果                  | 129 |
| 35.1 | 完全な相対論的扱い               | 129 |
| 35.2 | スピン軌道相互作用のコントロール        | 131 |
| 35.3 | 半相対論的扱い                 | 131 |

|           |   |            |
|-----------|---|------------|
| <b>36</b> | <b>軌道磁気モーメント</b>  | <b>132</b> |
| <b>37</b> | <b>DFT+<math>U</math> 法</b>                               | <b>134</b> |
| 37.1      | 標準設定  | 134        |
| 37.1.1    | DFT+ $U$ 法の選択: 単純化または一般化                                  | 134        |
| 37.1.2    | 二重計算に対する補正項の選択  | 135        |
| 37.1.3    | 軌道分極  | 138        |
| 37.2      | 付加機能  | 139        |
| 37.2.1    | 二つのスレーター積分の比の調整 ( $F^4/F^2$ )                             | 139        |
| 37.2.2    | 入力パラメタ $U$ からの $J$ と $F^4/F^2$ の見積もり                      | 139        |
| <b>38</b> | <b>ノンコリニアスピン方位に対する制約条件付き DFT</b>                          | <b>141</b> |
| <b>39</b> | <b>第二変分法: 磁気異方エネルギー (Magnetic Anisotropy Energy: MAE)</b> | <b>143</b> |
| <b>40</b> | <b>Zeeman 項</b>   | <b>146</b> |
| 40.1      | スピン磁気モーメントに対する Zeeman 項                                   | 146        |
| 40.2      | 軌道磁気モーメントに対する Zeeman 項                                    | 146        |
| <b>41</b> | <b>Berry 位相による巨視的分極の計算</b>                                | <b>148</b> |
| <b>42</b> | <b>交換結合パラメータ</b>  | <b>151</b> |
| 42.1      | 概要  | 151        |
| 42.2      | $J_x$ の計算   | 152        |
| 42.3      | OpenMX 計算による SCF 計算                                       | 152        |
| 42.4      | $J_x$ の設定ファイルの準備  | 153        |
| 42.5      | $J_x$ の実行と MPI 並列計算                                       | 154        |
| 42.6      | 実例  | 155        |
| <b>43</b> | <b>電気伝導計算</b>   | <b>159</b> |
| 43.1      | 概要  | 159        |
| 43.2      | ステップ 1: 電極部分の計算   | 161        |
| 43.3      | ステップ 2: NEGF 計算   | 162        |
| 43.4      | ステップ 3: 透過率、電流 (密度) と固有チャンネル                              | 167        |
| 43.4.1    | 透過率、電流、コンダクタンス  | 167        |
| 43.4.2    | 実空間における電流/スピン流密度  | 170        |
| 43.4.3    | 固有チャンネル解析   | 172        |
| 43.5      | ステップ 3 のみを再計算する方法   | 175        |
| 43.6      | ゼロバイアス下における周期系  | 177        |
| 43.7      | バイアス電圧効果の補間法  | 177        |
| 43.8      | NEGF の並列化   | 178        |
| 43.9      | ノンコリニア DFT 法に対する NEGF 法                                   | 178        |
| 43.10     | 実例  | 179        |
| 43.11     | NEGF の自動実行テスト   | 180        |

|   |            |
|---|------------|
| <b>44 最局在ワニエ関数</b>                        | <b>182</b> |
| 44.1 生成方法 . . . . .                       | 182        |
| 44.2 解析 . . . . .                         | 187        |
| 44.3 スプレッド関数の最適化過程の確認 . . . . .           | 188        |
| 44.4 MLWF の作成例 . . . . .                  | 191        |
| 44.5 出力ファイル . . . . .                     | 192        |
| 44.6 MLWF の自動実行テスト . . . . .              | 195        |
| <b>45 Wannier90 とのインターフェース</b>            | <b>196</b> |
| <b>46 対角化のための数値的に厳密な低次スケールリング法</b>        | <b>198</b> |
| <b>47 有効遮蔽媒質法 (ESM 法)</b>                 | <b>200</b> |
| 47.1 概要 . . . . .                         | 200        |
| 47.2 テスト計算例 . . . . .                     | 202        |
| <b>48 仕事関数の計算</b>                         | <b>204</b> |
| <b>49 NEB (Nudged elastic band) 法</b>     | <b>207</b> |
| 49.1 概要 . . . . .                         | 207        |
| 49.2 実行方法 . . . . .                       | 207        |
| 49.3 計算例と関連キーワード . . . . .                | 208        |
| 49.4 NEB 計算の再スタート . . . . .               | 212        |
| 49.5 ユーザー定義の初期経路 . . . . .                | 212        |
| 49.6 NEB 計算における SCF の確認 . . . . .         | 213        |
| 49.7 並列計算 . . . . .                       | 213        |
| 49.8 その他の注意 . . . . .                     | 214        |
| <b>50 Tersoff-Hamann 法による STM イメージ</b>    | <b>215</b> |
| <b>51 vdW 相互作用のための DFT-D2, DFT-D3 法</b>   | <b>216</b> |
| 51.1 DFT-D2 法 . . . . .                   | 216        |
| 51.2 DFT-D3 法 . . . . .                   | 217        |
| <b>52 バンドアンフォールディング法</b>                  | <b>219</b> |
| 52.1 バンド構造の解析 . . . . .                   | 219        |
| 52.2 バンド構造のアンフォールディング . . . . .           | 223        |
| 52.3 参照セルの原点 . . . . .                    | 227        |
| 52.4 アンフォールディングされたスペクトル重みの強度マップ . . . . . | 228        |
| 52.5 ノンコリニア DFT 計算の場合 . . . . .           | 229        |
| 52.6 入力ファイルの例 . . . . .                   | 229        |
| <b>53 k 空間でのスピネクスタの解析</b>                 | <b>231</b> |
| 53.1 概要 . . . . .                         | 231        |
| 53.2 FermiLoop: 一定エネルギー面での計算 . . . . .    | 233        |

|           |  |            |
|-----------|--|------------|
| 53.3      | GridCalc: k 点格子での計算                                  | 239        |
| 53.4      | BandDispersion: バンド分散関係の計算                           | 245        |
| 53.5      | MulPOnly: 利用者指定の k 点での計算                             | 251        |
| 53.6      | MulPCalc: 原子ごとに分割された k 空間スピン密度行列                     | 255        |
| 53.7      | kSpin の MPI 並列計算                                     | 259        |
| <b>54</b> | <b>スピン螺旋計算</b>                                       | <b>262</b> |
| <b>55</b> | <b>Fukui-Hatsugai-Suzuki の方法によるチャーン数とベリー曲率の計算</b>    | <b>264</b> |
| 55.1      | 概要   | 264        |
| 55.2      | 実例   | 264        |
| <b>56</b> | <b>Fukui-Hatsugai の方法による <math>Z_2</math> 不変量の計算</b> | <b>268</b> |
| 56.1      | 概要   | 268        |
| 56.2      | 実例   | 269        |
| 56.3      | 入力ファイル   | 273        |
| <b>57</b> | <b>内殻電子の絶対束縛エネルギー: XPS の内殻準位のエネルギー</b>               | <b>274</b> |
| 57.1      | 概要   | 274        |
| 57.2      | 気体系  | 275        |
| 57.3      | 固体系  | 278        |
| 57.4      | 入力ファイルの例   | 281        |
| <b>58</b> | <b>孤立系のイオン化ポテンシャルと電子親和力</b>                          | <b>283</b> |
| <b>59</b> | <b>光伝導度と誘電関数</b>                                     | <b>285</b> |
| 59.1      | 概要   | 285        |
| 59.2      | Si の場合   | 285        |
| 59.3      | 関連するキーワード  | 286        |
| 59.4      | ベンチマーク計算   | 289        |
| 59.5      | コード  | 291        |
| 59.6      | 実例   | 292        |
| 59.7      | 自動動作テスト  | 292        |
| <b>60</b> | <b>BoltzTraP とのインターフェース</b>                          | <b>294</b> |
| <b>61</b> | <b>“格子定数 vs. エネルギー” 曲線の計算</b>                        | <b>298</b> |
| 61.1      | “格子定数 vs. エネルギー” 曲線                                  | 298        |
| 61.2      | デルタ因子  | 299        |
| <b>62</b> | <b>フェルミ面</b>   | <b>300</b> |
| <b>63</b> | <b>2 つの Gaussian Cube ファイルの差の解析</b>                  | <b>301</b> |
| <b>64</b> | <b>2 つの幾何構造の差の解析</b>                                 | <b>302</b> |

|                                     |     |
|-------------------------------------|-----|
| 65 相互作用によって引き起こされる電荷密度の差の解析         | 304 |
| 66 セルサイズの自動決定                       | 306 |
| 67 開発者のためのインターフェース                  | 307 |
| 68 自動フォース・テスター                      | 309 |
| 69 自動メモリーリーク・テスター                   | 311 |
| 70 メモリ使用量の解析                        | 313 |
| 71 大規模ファイルのバイナリ形式での出力               | 314 |
| 72 Gaussian cube 形式から周期的 XSF 形式への変換 | 315 |
| 73 入力ファイルの例                         | 316 |
| 74 知られている問題点                        | 318 |
| 75 OpenMX フォーラム                     | 319 |
| 76 OpenMX に関するその他の情報源               | 320 |
| 77 他ツールへのリンク                        | 321 |
| 78 その他                              | 324 |

# 1 OpenMX について

OpenMX (Open source package for Material eXplorer) は密度汎関数理論 (DFT) [1]、ノルム保存型擬ポテンシャル [32, 33, 34, 35, 36] および擬原子基底関数 [41] に基づき、原子レベルから物質の第一原理シミュレーションを実行するためのソフトウェア・パッケージです。OpenMX で用いられている計算手法、アルゴリズム、またそのプログラム上での実装は並列コンピュータ上での MPI 並列や OpenMP/MPI ハイブリッド並列による大規模第一原理電子状態計算を実現するために、入念に設計されています。OpenMX において実現された DFT の効率的な実装により、固体、表面、界面、液体、そして低次元物質などの幅広い物質の電子構造、磁気構造、また幾何構造を第一原理的に計算することが可能です。数百コアを有する並列コンピュータを使用することにより、1000 個の原子で構成される系を通常対角化手法を用いて扱うことができます。さらに数千 CPU のコアを有する並列コンピュータを使用すれば、一万個以上の原子から構成される系の第一原理による電子状態計算さえも、OpenMX に実装されている  $O(N)$  法を用いて実行可能です。多くの元素に対して最適化された擬ポテンシャルおよび基底関数がデータベースとして整備されており、またその精度はベンチマーク計算によって検証されています。そのためユーザは自分でこれらのデータを準備する必要がなく、速やかに各自のシミュレーションを開始することができます。OpenMX には磁気特性、誘電特性、電気伝導特性、反応エネルギー障壁などの様々な物理的・化学的特性を計算するための機能が実装されており、量子力学の第一原理からナノスケールの物質群を深く理解するための強力なツールとして、幅広い活用が期待されます。これまでの応用計算の事例は OpenMX の Web サイト (<http://www.openmx-square.org/>) に記載されていますので、参考にして下さい。OpenMX の開発は、2000 年に尾崎グループにより開始され、その後、本マニュアルの冒頭のリストに挙げられている数多くの開発者が本オープンソースパッケージの開発に貢献してきました。プログラム・パッケージとソースコードは GNU General Public License version 3 (GPLv3) [102] に準じて配布されており、OpenMX の Web サイトからダウンロードすることができます。

OpenMX Ver. 3.9 の特徴と機能は以下のようになります。

- クラスタ計算法、バンド計算法、 $O(N)$  法、低次スケールリング法による全エネルギーと力の計算
- 交換相関エネルギーに対する局所密度近似 (LDA, LSDA) [2, 3, 4] および一般化勾配近似 (GGA) [5]
- DFT+ $U$  法 [20]
- ノルム保存型擬ポテンシャル [2, 33, 34, 36]
- 変分的に最適化された擬原子基底関数 [41]
- 擬ポテンシャル法の枠組みでの完全および半相対論的取り扱い [12, 32, 16]
- ノンコリニア DFT 法 [8, 9, 10, 11]
- ノンコリニア DFT 法におけるスピン磁気モーメントおよび軌道磁気モーメントの方位制御 [13]
- Berry 位相法による巨視的分極率の計算 [15]
- $O(N)$  分割統治法 (DC 法) [50]
- 局在化自然軌道による  $O(N)$  分割統治法 (DC-LNO 法) [51]

- $O(N)$ Krylov 部分空間法
- ELPA [39] による並列固有値ソルバ
- 単純混合法、RMM-DIIS 法 [58]、GR-Pulay 法 [57]、Kerker 法 [59]、Kerker 因子による重み付き RMM-DIIS 法 [58] による電荷密度の混合、ハミルトニアン行列に対する RMM-DIIS 法による電荷密度の混合 [58]
- 交換結合パラメータ [17, 18] の計算
- 有効遮蔽媒質法 (ESM) [125, 128]
- 走査型トンネル顕微鏡 (STM) シミュレーション [71]
- DFT-D2、DFT-D3 法による Van der Waals 相互作用 [135, 136, 137]
- バンドアンフォールディング法 [142]
- Nudged Elastic Band 法 (NEB 法) [72]
- 固体中の内殻軌道の絶対束縛エネルギーの計算 [88]
- 光伝導率と誘電関数 [98]
- 電荷ドーピング
- のこぎり波による均一電界の導入
- 完全および制限付き構造最適化
- 非平衡グリーン関数 (NEGF) 法による電気伝導計算 [73]
- 最大局所化ワニエ関数の計算
- NVE アンサンブル分子動力学
- 速度スケーリング法 [30] および Nose-Hoover 法 [31] による NVT アンサンブル分子動力学
- マリケン (Mulliken) 法、ポロノイ (Voronoi) 法、ESP 法による電荷・スピン数の解析
- 自然軌道解析 [7]
- 波動関数および電子・スピン密度分布の解析
- バンド計算によるバンド分散の解析
- 状態密度 (DOS) および射影 DOS の計算
- 柔軟なデータ書式 (flexible data format) による入力ファイル
- BoltzTrap へのインタフェース [100, 101]
- Wannier90 へのインタフェース [145]

- 電荷密度などを可視化するための XCrySDen へのインタフェース [105]
- 完全な動的メモリ割当て
- メッセージ・パッシング・インタフェース (MPI) による並列計算
- OpenMP による並列計算
- 開発者向けの利便性の高いユーザーインタフェース

コリニア DFT 法ではスカラー相対論的擬ポテンシャルが、またノンコリニア DFT 法では完全相対論的擬ポテンシャルが利用可能です。またスピンおよび軌道磁気モーメントを制御するために制限ノンコリニア DFT 法が実装されています。これらの手法は複雑なノンコリニア磁気構造やスピン軌道相互作用を調べるのに役に立ちます。通常対角化による計算は、数千コアまでの並列化が実現できる ELPA に基づく並列固有値ソルバ [39] 及び ScaLAPACK によって行われます。この機能により、通常対角化法を用いて 1000 個の原子からなる系の計算が可能になります。この高並列化対角化手法によってクラスター、分子、スラブおよび固体に対する大規模計算が可能ですが、線形スケールリング法および低次スケールリング法も固有値ソルバとして利用可能です。これらの低次スケールリング法に対しては計算精度と計算効率に対して注意深い検討を行うことで、1 万原子を越える系をも取り扱うことが可能です。結晶格子可変の構造最適化やバンドアンフォールディング法も有用な機能です。また OpenMX Ver. 3.9 の重要な新機能として、分割統治法と局在自然軌道に基づく新しい  $O(N)$  法と X 線光電子分光法 (XPS) で観測される内殻電子束縛エネルギーの絶対値計算が挙げられます。OpenMX の開発は継続して行われています。本オープンソースコードの発展に寄与して頂ける開発者の方のご参加を歓迎致します。

## 2 関連文献

OpenMX は GNU General Public License version 3 (GPLv3) [102] の条件のもとで配布されています。今後の更なる OpenMX の発展のために、OpenMX を用いて論文を執筆する際には、用いた機能に応じて以下の文献を引用して頂けると幸いです。

- OpenMX 全般  
T. Ozaki, Phys. Rev. B. **67**, 155108, (2003); T. Ozaki and H. Kino, Phys. Rev. B **69**, 195113 (2004); T. Ozaki and H. Kino, Phys. Rev. B **72**, 045121 (2005); K. Lejaeghere et al., Science 351, aad3000 (2016).
- 大規模並列計算  
T.V.T. Duy and T. Ozaki, Comput. Phys. Commun. **185**, 777 (2014); T.V.T. Duy and T. Ozaki, Comput. Phys. Commun. **185**, 153 (2014).
- $O(N)$  分割統治 局在自然軌道法  
T. Ozaki, M. Fukuda, G. Jiang, Phys. Rev. B **98**, 245137 (2018).
- $O(N)$  Krylov 部分空間法  
T. Ozaki, Phys. Rev. B **74**, 245101 (2006).
- 数値的に厳密な低次元スケーリング法  
T. Ozaki, Phys. Rev. B **82**, 075131 (2010).
- DFT+ $U$  法  
M.J. Han, T. Ozaki, and J. Yu, Phys. Rev. B **74**, 045110 (2006); S. Ryee and M.J. Han, J. Phys:Condens. Matter **30**, 275802 (2018). S. Ryee and M.J. Han, Scientific Reports **8**, 9559 (2018).
- 交換結合パラメータ  
M.J. Han, T. Ozaki, and J. Yu, Phys. Rev. B **70**, 184421 (2004); A Terasawa, M Matsumoto, T Ozaki, and Y Gohda, J. Phys. Soc. Jpn. **88**, 114706 (2019).
- 非平衡グリーン関数 (NEGF) 法  
T. Ozaki, K. Nishio, and H. Kino, Phys. Rev. **81**, 035116 (2010); T. Ozaki, Phys. Rev. B **75**, 035123 (2007).
- 有効遮蔽媒質 (ESM) 法  
T. Ohwaki, M. Otani, T. Ikeshoji, and T. Ozaki, J. Chem. Phys. **136**, 134101 (2012).
- ワニエ関数の生成  
H. Weng, T. Ozaki, and K. Terakura, Phys. Rev. B **79**, 235118 (2009).
- 自然原子軌道の生成  
T. Ohwaki, M. Otani, and T. Ozaki, J. Chem. Phys. **140**, 244105 (2014).

- バンドアンフォールディング法  
C.-C. Lee, Y. Yamada-Takamura, and T. Ozaki, *J. Phys.: Condens. Matter* **25**, 345501 (2013).
- XPS 束縛エネルギー  
T. Ozaki and C.-C. Lee, *Phys. Rev. Lett.* **118**, 026401 (2017).
- BoltzTraP 計算  
M. Miyata, T. Ozaki, T. Takeuchi, S. Nishino, M. Inukai, and M. Koyano, *Journal of Electronic Materials* **47**, 3254 (2017).
- フェルミ面  
M. Kawamura, *Comp. Phys. Comm.* **239**, 197 (2019).
- K 空間のスピントクスチャ解析  
H. Kotaka, F. Ishii, and M. Saito, *Jpn. J. Appl. Phys.* **52**, 035204 (2013).; N. Yamaguchi and F. Ishii, *Appl. Phys. Express* **10**, 123003 (2017).
- スピンスパイラル計算  
T.B. Prayitno and F. Ishii, *J. Phys. Soc. Jpn.* **87**, 114709 (2018); T.B. Prayitno and F. Ishii, *J. Phys. Soc. Jpn.* **88**, 054701 (2019).
- $Z_2$ , チャーン数とベリー曲率  
H. Sawahata, N. Yamaguchi, H. Kotaka, and F. Ishii, *Jpn. J. Appl. Phys.* **57**, 030309 (2018).
- OpenMX Viewer  
Y.-T Lee and T. Ozaki, *Journal of Molecular Graphics and Modelling* **89**, 192 (2019).

## 3 インストール

### 3.1 必須ライブラリ

OpenMX は、以下の 3 つのライブラリがインストール済みの Linux 環境にインストールすることができます。

- ScaLAPACK (および BLACS) (<http://www.netlib.org/>)
- FFTW (<http://www.fftw.org/>)
- MPICH2 や OpenMPI などの MPI ライブラリ

これらのライブラリ・パッケージが対象コンピュータにインストールされていない場合、OpenMX をインストールする前にこれらをインストールする必要があります。MPICH2 や OpenMPI などの MPI ライブラリは、OpenMX Ver. 3.9 のインストールに必須です。MPI ライブラリが利用可能できない場合、OpenMX Ver. 3.9 をインストールする次のステップに進むことはできません。OpenMX Ver. 3.9 は ScaLAPACK と BLACS を必要とし、LAPACK と BLAS とによる OpenMX Ver. 3.9 のコンパイルはサポートされません。代替策として、Intel Math Kernel Library (MKL) が ScaLAPACK と BLACS の代わりに利用できます。上記のライブラリ・パッケージがインストール済みの場合は、次のインストール手順に進んで下さい。openmx3.9.tar.gz をダウンロード後、以下のコマンドで解凍します。

```
% tar zxvf openmx3.9.tar.gz
```

解凍すると、「openmx3.9」ディレクトリの下に「source」、「work」、「DFT\_DATA19」の 3 つのディレクトリが作成されます。これらのディレクトリにはそれぞれソースコード、入力ファイル、Ver. 2019 の擬原子基底関数および擬ポテンシャルのデータファイルが含まれます。

### 3.2 シリアル版

OpenMX Ver. 3.9 ではシリアル版のインストールはサポートされていません。

### 3.3 MPI 版

MPI 版の OpenMX をインストールするには、「source」ディレクトリに移動し、「makefile」を編集し、CC、FC、LIB を設定することでコンパイラおよびライブラリを指定します。「makefile」のデフォルトの CC、FC、LIB 設定は以下の通りです。

```
MKLROOT = /opt/intel/mkl
CC = mpicc -O3 -xHOST -ip -no-prec-div -qopenmp -I/opt/intel/mkl/include/fftw
FC = mpif90 -O3 -xHOST -ip -no-prec-div -qopenmp
LIB= -L${MKLROOT}/lib/intel64 -lmkl_scalapack_lp64 -lmkl_intel_lp64 \
    -lmkl_intel_thread -lmkl_core -lmkl_blacs_open_mpi_lp64 \
    -lmpi_usempif08 -lmpi_usempi_ignore_tkr \
    -lmpi_mpifh -liomp5 -lpthread -lm -ldl
```

CC と FC は、それぞれ C と FORTRAN のコンパイラを指定し、LIB はリンクするライブラリを指定します。Ver. 3.6 までは FC の指定は不要でしたが、ELPA に基づく並列固有値ソルバ [39] の導入に伴い、Ver. 3.9 ではこれが必要となります。OpenMP が利用不可の環境では「-Dnoomp」オプションを指定して下さい。コンパイルとリンクが正しく行われ、十分に最適化された実行ファイルを作成するには、実行環境に適した CC、FC、LIB の設定を行うことが必要です。これらのオプションの設定後、以下のコマンドよりインストールを実行します。

```
% make install
```

コンパイルが正常に完了すると、「openmx」という実行ファイルが「work」ディレクトリの下に作成されます。OpenMX の実行効率を上げるために、必要に応じて実行環境に合わせてコンパイラやコンパイラオプションを変更することで、最適化された実行ファイルを生成することができます。「source」ディレクトリの「makefile」ファイル内には、CC、FC、LIB の設定例がいくつか用意されていますので参考にして下さい。

### 3.4 MPI/OpenMP ハイブリッド並列版

MPI/OpenMP ハイブリッド版を生成する場合には、「source」ディレクトリの「makefile」ファイル内の CC および FC に、OpenMP 並列化のコンパイラオプションを追加するだけです。MPI/OpenMP バージョンをインストールするには、「source」ディレクトリに移動し、「makefile」内の CC、FC、LIB を以下の例のように指定します。

icc の場合

```
CC = mpicc -O3 -xHOST -ip -no-prec-div -qopenmp -I/opt/intel/mkl/include/fftw  
FC = mpif90 -O3 -xHOST -ip -no-prec-div -qopenmp
```

OpenMP のコンパイラオプションはコンパイラに依存します。特に Intel コンパイラ用の「-qopenmp」に対応するオプションに注意してください。CC、FC、LIB を適切に指定した後、下記のコマンドでインストールを実行します。

```
% make install
```

コンパイルが正常に完了すると、「work」ディレクトリに「openmx」実行ファイルが生成されます。OpenMX の実行効率を上げるために、必要に応じて実行環境に合わせてコンパイラやコンパイラオプションを変更することで、最適化された実行ファイルを生成することができます。

### 3.5 FFTW3

Ver. 3.6 までの旧バージョンは FFTW2 と FFTW3 の両方に対応していましたが、OpenMX Ver. 3.9 では FFTW3 のみが使用可能です。makefile 中で以下のように指定することで FFTW3 とリンクすることができます。

```
LIB = -lfftw3
```

多くの利用者に対しては Intel MKL に含まれている FFTW3 の利用が推奨されます。

## 3.6 その他のオプション

### 3.6.1 -Dnosse

$O(N)$  クリロフ (Krylov) 部分空間法のルーチン (Krylov.c) は、ストリーミング SIMD 拡張命令 (SSE) を利用して最適化されています。デフォルトでは SSE を活用できるようにソースコードがコンパイルされます。お使いのプロセッサが SSE に対応していない場合、-Dnosse オプションを CC に含めて下さい。

### 3.6.2 -Dkcomp

富士通 (株) の SPARC プロセッサをご利用の場合は、CC および FC コンパイラオプションに-Dkcomp を含めて下さい。

## 3.7 プラットフォーム

OpenMX Ver. 3.9 は、これまで以下のプラットフォームで正常に作動することが確認されています。

- Intel Xeon 搭載クラスタ
- AMD EPYC 搭載クラスタ
- CRAY-XC40
- 富士通 FX100

## 3.8 インストールの際のヒント

OpenMX のインストールにおける問題のほとんどは、ScaLAPACK、BLACS およびこれらの代替ライブラリへのリンク時のエラーが原因となっています。多くの場合において MKL をリンクすることをお勧めします。これらのライブラリとリンクする例は「source」ディレクトリの「makefile」内に記載されています。

いくつかのポピュラーなプラットフォーム上でのインストールに関するヒントを以下に記します。OpenMX は、C と FORTRAN のコンパイラ、および ScaLAPACK、BLACS、FFTW3 ライブラリを必要とします。更に、C コンパイラはリンク時に使用されるため、ご使用のコンパイラに対応する FORTRAN ライブラリを明示的に指定しなければなりません。下記に一般的なコンパイラおよび ScaLAPACK、BLACS ライブラリにおけるインストール設定例を記載します。ここで、FFT ライブラリは /usr/local/fftw3/ にインストールされていると仮定しています。

- Intel の C および FORTRAN コンパイラ (icc、ifort)、ScaLAPACK と BLACS には MKL ライブラリを使用

```
MKLROOT = /opt/intel/mkl
CC = mpicc -O3 -xHOST -ip -no-prec-div -qopenmp -I/usr/local/fftw3/include
FC = mpif90 -O3 -xHOST -ip -no-prec-div -qopenmp
LIB= -L/usr/local/fftw3/lib -lfftw3 \
-L$MKLROOT/lib/intel64 -lmkl_scalapack_lp64 -lmkl_intel_lp64 \
-lmkl_intel_thread -lmkl_core -lmkl_blacs_open_mpi_lp64 \
-lmpi_usempif08 -lmmpi_usempi_ignore_tkr \
-lmpi_mpicfh -liomp5 -lpthread -lm -ldl
```

- GNU C および FORTRAN コンパイラ (gcc、g++、gfortran)、 ScaLAPACK と BLACS には MKL ライブラリを使用

```
MKLROOT = /opt/intel/mkl
CC = mpicc -O3 -xHOST -ip -no-prec-div -qopenmp -I/usr/local/fftw3/include
FC = mpif90 -O3 -xHOST -ip -no-prec-div -qopenmp
LIB= -L/usr/local/fftw3/lib -lfftw3 \
-L$MKLROOT/lib/intel64 -lmkl_scalapack_lp64 -lmkl_intel_lp64 \
-lmkl_intel_thread -lmkl_core -lmkl_blacs_open_mpi_lp64 \
-lmpi_usempif08 -lmmpi_usempi_ignore_tkr \
-lmpi_mpiofh -liomp5 -lpthread -lm -ldl
```

- GNU C および FORTRAN コンパイラ (gcc、g++、gfortran) ScaLAPACK と BLACS を使用

```
CC = mpicc -O3 -ffast-math -fopenmp -Dkcomp -I/usr/local/include -I/home/ytl/openmpi-3.0.1/mpi/include
FC = mpif90 -O3 -ffast-math -fopenmp -Dkcomp -I/home/ytl/openmpi-3.0.1/mpi/include
LIB= -L/usr/local/lib -lfftw3 -L/home/ytl/openmpi-3.0.1/mpi -lmmpi -lmmpi_mpiofh \
-L/home/ytl/Packages/lapack-3.7.0 -llapack -lrefblas -lgfortran
```

- FX100 計算機での FUJITSU コンパイラを使用

```
CC = mpifccpx -Kfast -Dnosse -Dkcomp
FC = mpifrtpx -Kfast -Dkcomp
LIB = -lfftw3 -SCALAPACK -SSL2BLAMP
```

その他のコンパイラと ScaLAPACK、BLACS ライブラリの組み合わせも同様に可能です。下記のコマンドを使用することで、MPI が利用するコンパイラ (Intel、PGI、GNU 等の) についての情報を表示することができます。

```
%mpicc -compile-info (MPICH の場合)
%mpicc -help (OpenMPI の場合)
```

C コンパイラが FORTRAN ライブラリの保存場所を見つけることができない場合、次のようなリンクエラーが生じることがあります。

Intel コンパイラの場合：

```
/usr/bin/ld: cannot find -lifcore
```

PGI コンパイラの場合：

```
/usr/bin/ld: cannot find -lpgf90
```

あるいは

```
-lpgf90_rpm1, -lpgf902, -lpgf90rtl, -lpgftnrtl
```

(-pgf90libs は、これらへのショートカットにすぎないため)

GNU コンパイラの場合 :

```
/usr/bin/ld: cannot find -lgfortran
```

これらのリンクエラーに関する問題を解決するには、以下の手順で FORTRAN ライブラリを明示的に指定する必要があります。まず FORTRAN コンパイラの保存場所は次のコマンドで確認することができます。

```
%which ifort (Intel コンパイラの場合)
/opt/intel/fce/10.0.026/bin/ifort
```

```
%which pgf90 (PGI コンパイラの場合)
/opt/pgi/linux86-64/7.0/bin/pgf90
```

```
%which gfortran (GNU コンパイラの場合)
/usr/bin/gfortran
```

次に、FORTRAN ライブラリは通常、上記/bin ディレクトリの親ディレクトリの下にある/lib ディレクトリにあるため、次のように LIB を指定します。

```
LIB= ... -L/opt/intel/fce/10.0.026/lib -lifcore (Intel コンパイラの場合)
LIB= ... -L/opt/pgi/linux86-64/7.0/lib -pgf90libs (PGI コンパイラの場合)
LIB= ... -L/usr/lib -lgfortran (GNU コンパイラの場合)
```

Intel Math Kernel Library については以下のウェブサイト (Intel Math Kernel Library Link Line Advisor) を利用できます。

```
https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor
```

この website を利用することで、使用環境に応じて、どの様にライブラリのリンクとコンパイルオプションを設定するべきかに関して詳細な情報が得られます。

### 3.9 *make* に関するオプション

すべてのオブジェクトファイル及び実行形式ファイルは以下のコマンドで削除できます。

```
% make clean
```

すべての実行形式ファイル (ポスト処理用コードも含む) は以下のコマンドでコンパイルされます。

```
% make all
```

## 4 テスト計算

インストールが正常に完了したら「work」ディレクトリに移動し、「Methane.dat」を入力ファイルとして下記のように「openmx」を実行して下さい。

```
% mpirun -np 1 openmx Methane.dat > met.std &
```

OpenMP/MPI ハイブリッドバージョンをお使いの場合は下記のコマンドで実行します。

```
% mpirun -np 1 openmx Methane.dat -nt 1 > met.std &
```

計算で用いた「Methane.dat」は、メタン分子の SCF 計算を実行するための入力ファイルです。構造最適化は行いません。この計算は 2.6 GHz Xeon 搭載マシンの単一コアで実行した場合、5 秒程度で終了しますが、実際の実行時間はコンピュータ環境に依存します。計算が正常に終了すると、次の 11 個のファイルと 1 個のディレクトリが「work」ディレクトリ内に作成されます。

|                |                                    |
|----------------|------------------------------------|
| met.std        | SCF 計算の標準出力                        |
| met.out        | 入力ファイルと標準出力                        |
| met.xyz        | 最終的な幾何構造                           |
| met.ene        | 各 MD ステップにおける計算値                   |
| met.md         | 各 MD ステップにおける幾何構造                  |
| met.md2        | 最終 MD ステップにおける幾何構造                 |
| met.cif        | Material Studio 用の初期構造の cif ファイル   |
| met.tden.cube  | Gaussian cube 形式の全電子密度             |
| met.v0.cube    | Gaussian cube 形式の Kohn-Sham ポテンシャル |
| met.vhart.cube | Gaussian cube 形式の Hartree ポテンシャル   |
| met.dden.cube  | 原子密度から計算した差電子密度                    |
| met_rst/       | 再スタートファイルを保存するディレクトリ               |

標準出力へ出力されるデータは「met.std」ファイルに格納されます。これは SCF 計算時の計算過程を知るために役立ちます。「met.out」ファイルには、全エネルギー、力、Kohn-Sham 固有値、Mulliken 電荷、SCF 計算の収束履歴、および計算時間などの結果が記載されています。参考のために「met.out」ファイルの一部を下記に抜粋します。14 回の SCF 反復でエネルギー固有値の変化が  $1.0e-10$  ハートリー (Hartree) 以下に収束していることが分かります。

```
*****
*****
                SCF history at MD= 1
*****
*****
SCF=   1  NormRD=  1.000000000000  Uele= -3.523169099731
SCF=   2  NormRD=  0.181517404404  Uele= -3.686855123738
SCF=   3  NormRD=  0.449067381009  Uele= -4.193062144919
SCF=   4  NormRD=  0.541215648203  Uele= -4.381387140154
```

```

SCF= 5 NormRD= 0.509921689399 Uele= -4.352426233337
SCF= 6 NormRD= 0.004026023243 Uele= -3.886371199720
SCF= 7 NormRD= 0.000838640096 Uele= -3.889312346884
SCF= 8 NormRD= 0.000420666755 Uele= -3.889396659132
SCF= 9 NormRD= 0.000241013350 Uele= -3.889362708861
SCF= 10 NormRD= 0.000573725679 Uele= -3.889427222948
SCF= 11 NormRD= 0.000000150516 Uele= -3.889316043314
SCF= 12 NormRD= 0.000000001917 Uele= -3.889316014533
SCF= 13 NormRD= 0.000000000005 Uele= -3.889316014156
SCF= 14 NormRD= 0.000000000001 Uele= -3.889316014146

```

また全エネルギー、化学ポテンシャル、Kohn-Sham 固有値、Mulliken 電荷、双極子モーメント、力、規格化座標、計算時間などが「met.out」ファイルに以下のように出力されます。

```
*****
```

```
      Total energy (Hartree) at MD = 1
```

```
*****
```

```
Uele.          -3.889316014146
```

```
Ukin.           5.533759381370
```

```
UHO.           -14.855519969177
```

```
UH1.            0.041396138425
```

```
Una.           -5.040606545149
```

```
Unl.           -0.134650846490
```

```
Uxc0.          -1.564720263874
```

```
Uxc1.          -1.564720263874
```

```
Ucore.         9.551521413583
```

```
Uhub.           0.000000000000
```

```
Ucs.           0.000000000000
```

```
Uzs.           0.000000000000
```

```
Uzo.           0.000000000000
```

```
Uef.           0.000000000000
```

```
UvdW           0.000000000000
```

```
Uch            0.000000000000
```

```
Utot.          -8.033540955187
```

```
Note:
```

```
Utot = Ukin+UHO+UH1+Una+Unl+Uxc0+Uxc1+Ucore+Uhub+Ucs+Uzs+Uzo+Uef+UvdW
```

```
Uene:  band energy
```

```
Ukin:  kinetic energy
```

UH0: electric part of screened Coulomb energy  
 UH1: difference electron-electron Coulomb energy  
 Una: neutral atom potential energy  
 Unl: non-local potential energy  
 Uxc0: exchange-correlation energy for alpha spin  
 Uxc1: exchange-correlation energy for beta spin  
 Ucore: core-core Coulomb energy  
 Uhub: DFT+U energy  
 Ucs: constraint energy for spin orientation  
 Uzs: Zeeman term for spin magnetic moment  
 Uzo: Zeeman term for orbital magnetic moment  
 Uef: electric energy by electric field  
 UvdW: semi-empirical vdW energy

(see also PRB 72, 045121(2005) for the energy contributions)

Chemical potential (Hartree) 0.000000000000

\*\*\*\*\*  
 \*\*\*\*\*  
 Eigenvalues (Hartree) for SCF KS-eq.  
 \*\*\*\*\*  
 \*\*\*\*\*

Chemical Potential (Hartree) = 0.000000000000  
 Number of States = 8.000000000000  
 HOMO = 4  
 Eigenvalues

|   | Up-spin           | Down-spin         |
|---|-------------------|-------------------|
| 1 | -0.69897506408475 | -0.69897506408475 |
| 2 | -0.41523055776668 | -0.41523055776668 |
| 3 | -0.41523055768741 | -0.41523055768741 |
| 4 | -0.41522182758055 | -0.41522182758055 |
| 5 | 0.21221759603691  | 0.21221759603691  |
| 6 | 0.21221759685634  | 0.21221759685634  |
| 7 | 0.21230533059490  | 0.21230533059490  |
| 8 | 0.24741918440773  | 0.24741918440773  |

\*\*\*\*\*  
 \*\*\*\*\*  
 Mulliken populations

\*\*\*\*\*  
 \*\*\*\*\*

Total spin S = 0.00000000000

|   |   | Up spin     | Down spin   | Sum         | Diff        |
|---|---|-------------|-------------|-------------|-------------|
| 1 | C | 2.509748760 | 2.509748760 | 5.019497520 | 0.000000000 |
| 2 | H | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| 3 | H | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| 4 | H | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| 5 | H | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |

Sum of MulP: up = 4.00000 down = 4.00000  
 total= 8.00000 ideal(neutral)= 8.00000

Decomposed Mulliken populations

| 1              | C        | Up spin     | Down spin   | Sum         | Diff        |
|----------------|----------|-------------|-------------|-------------|-------------|
|                | multiple |             |             |             |             |
| s              | 0        | 0.681737894 | 0.681737894 | 1.363475787 | 0.000000000 |
| sum over m     |          | 0.681737894 | 0.681737894 | 1.363475787 | 0.000000000 |
| sum over m+mul |          | 0.681737894 | 0.681737894 | 1.363475787 | 0.000000000 |
| px             | 0        | 0.609352701 | 0.609352701 | 1.218705403 | 0.000000000 |
| py             | 0        | 0.609305463 | 0.609305463 | 1.218610926 | 0.000000000 |
| pz             | 0        | 0.609352702 | 0.609352702 | 1.218705404 | 0.000000000 |
| sum over m     |          | 1.828010866 | 1.828010866 | 3.656021733 | 0.000000000 |
| sum over m+mul |          | 1.828010866 | 1.828010866 | 3.656021733 | 0.000000000 |

| 2              | H        | Up spin     | Down spin   | Sum         | Diff        |
|----------------|----------|-------------|-------------|-------------|-------------|
|                | multiple |             |             |             |             |
| s              | 0        | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| sum over m     |          | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| sum over m+mul |          | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |

| 3              | H        | Up spin     | Down spin   | Sum         | Diff        |
|----------------|----------|-------------|-------------|-------------|-------------|
|                | multiple |             |             |             |             |
| s              | 0        | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| sum over m     |          | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |
| sum over m+mul |          | 0.372562810 | 0.372562810 | 0.745125620 | 0.000000000 |

| 4 | H | Up spin | Down spin | Sum | Diff |
|---|---|---------|-----------|-----|------|
|   |   |         |           |     |      |

```

multiple
s      0      0.372562810  0.372562810  0.745125620  0.000000000
sum over m      0.372562810  0.372562810  0.745125620  0.000000000
sum over m+mul  0.372562810  0.372562810  0.745125620  0.000000000

```

```

5      H      Up spin      Down spin      Sum      Diff
multiple
s      0      0.372562810  0.372562810  0.745125620  0.000000000
sum over m      0.372562810  0.372562810  0.745125620  0.000000000
sum over m+mul  0.372562810  0.372562810  0.745125620  0.000000000

```

```

*****
*****
Dipole moment (Debye)
*****
*****

```

Absolute D 0.00000000

|             | Dx          | Dy          | Dz          |
|-------------|-------------|-------------|-------------|
| Total       | 0.00000000  | 0.00000000  | -0.00000000 |
| Core        | 0.00000000  | 0.00000000  | 0.00000000  |
| Electron    | 0.00000000  | 0.00000000  | -0.00000000 |
| Back ground | -0.00000000 | -0.00000000 | -0.00000000 |

```

*****
*****
xyz-coordinates (Ang) and forces (Hartree/Bohr)
*****
*****

```

<coordinates.forces

```

5
1      C      0.00000  0.00000  0.00000  0.000000000000  0.00...
2      H     -0.88998 -0.62931  0.00000 -0.064890985127 -0.04...
3      H      0.00000  0.62931 -0.88998  0.000000000002  0.04...
4      H      0.00000  0.62931  0.88998  0.000000000002  0.04...
5      H      0.88998 -0.62931  0.00000  0.064890985122 -0.04...

```

coordinates.forces>

```

*****
*****

```

Fractional coordinates of the final structure

\*\*\*\*\*  
 \*\*\*\*\*

|   |   |                    |                    |                    |
|---|---|--------------------|--------------------|--------------------|
| 1 | C | 0.0000000000000000 | 0.0000000000000000 | 0.0000000000000000 |
| 2 | H | 0.9110019000000000 | 0.9370688000000000 | 0.0000000000000000 |
| 3 | H | 0.0000000000000000 | 0.0629312000000000 | 0.9110019000000000 |
| 4 | H | 0.0000000000000000 | 0.0629312000000000 | 0.0889981000000000 |
| 5 | H | 0.0889981000000000 | 0.9370688000000000 | 0.0000000000000000 |

\*\*\*\*\*  
 \*\*\*\*\*

Computational Time (second)

\*\*\*\*\*  
 \*\*\*\*\*

Elapsed.Time.           4.600

|                            | Min_ID | Min_Time | Max_ID | Max_Time |
|----------------------------|--------|----------|--------|----------|
| Total Computational Time = | 0      | 4.600    | 0      | 4.600    |
| readfile =                 | 0      | 2.578    | 0      | 2.578    |
| truncation =               | 0      | 0.146    | 0      | 0.146    |
| MD_pac =                   | 0      | 0.000    | 0      | 0.000    |
| OutData =                  | 0      | 0.283    | 0      | 0.283    |
| DFT =                      | 0      | 1.591    | 0      | 1.591    |

\*\*\* In DFT \*\*\*

|                     |   |       |   |       |
|---------------------|---|-------|---|-------|
| Set_OLP_Kin =       | 0 | 0.052 | 0 | 0.052 |
| Set_Nonlocal =      | 0 | 0.039 | 0 | 0.039 |
| Set_ProExpn_VNA =   | 0 | 0.156 | 0 | 0.156 |
| Set_Hamiltonian =   | 0 | 0.663 | 0 | 0.663 |
| Poisson =           | 0 | 0.214 | 0 | 0.214 |
| Diagonalization =   | 0 | 0.005 | 0 | 0.005 |
| Mixing_DM =         | 0 | 0.000 | 0 | 0.000 |
| Force =             | 0 | 0.039 | 0 | 0.039 |
| Total_Energy =      | 0 | 0.256 | 0 | 0.256 |
| Set_Aden_Grid =     | 0 | 0.019 | 0 | 0.019 |
| Set_Orbitals_Grid = | 0 | 0.015 | 0 | 0.015 |
| Set_Density_Grid =  | 0 | 0.124 | 0 | 0.124 |
| RestartFileDFT =    | 0 | 0.004 | 0 | 0.004 |
| Mulliken_Charge =   | 0 | 0.000 | 0 | 0.000 |
| FFT(2D)_Density =   | 0 | 0.000 | 0 | 0.000 |

Others = 0 0.005 0 0.005

出力ファイル「met.tden.cube」、「met.v0.cube」、「met.vhart.cube」、および「met.dden.cube」には、それぞれ全電子密度、Kohn-Sham ポテンシャル、Hartree ポテンシャル、および構成孤立原子の電子密度の重ね合わせを基準とした差電子密度が Gaussian cube 形式で書き出されています。Gaussian cube 形式は広く使われているグリッド形式であるため、VESTA [103] や Molekel [104]、XCrySDen [105] などの無料の分子モデリングソフトウェアを用いて可視化することができます。後述の章にて可視化の例を示します。

## 5 自動実行のテスト

「テスト計算」の章で説明した計算に加え、OpenMX の主な機能が正常にインストールされているか確認したい場合には、自動実行のテストを実施することをお勧めします。この自動実行テストを行うには、以下のコマンドにより OpenMX を実行します。

MPI 並列化の場合：

```
% mpirun -np 8 openmx -runtest
```

MPI/OpenMP 並列化の場合：

```
% mpirun -np 8 openmx -runtest -nt 2
```

並列計算の実行の際に mpirun の他のオプションを指定することもできます。このテストにおいて、OpenMX は 14 個の入力テストファイルを計算し、結果を「work/input\_example」に格納されている参照データと比較します。比較結果（全エネルギーおよび力の差の絶対値）は「work」ディレクトリの「runtest.result」というファイルに格納されます。参照データは 2.6 GHz Xeon シングルプロセッサのコンピュータで計算されたものです。絶対差分が小数点以下 7 桁以内であればインストールは正常に行われたと判断できます。自動実行テストの結果、生成された「runtest.result」の例を以下に示します。

尾崎研究室のクラスタ計算機 mx17 (Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz),

icc version 16.0.2, compiler option -O3 -xHOST -ip -no-prec-div -qopenmp

20 processes (MPI) x 1 thread (OpenMP)

|    |                             |                        |                           |                            |
|----|-----------------------------|------------------------|---------------------------|----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 3.09  | diff Utot= 0.000000000021 | diff Force= 0.000000000000 |
| 2  | input_example/C60.dat       | Elapsed time(s)= 8.71  | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 3  | input_example/CO.dat        | Elapsed time(s)= 6.77  | diff Utot= 0.000000000000 | diff Force= 0.000000000004 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 7.15  | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 20.16 | diff Utot= 0.000000000003 | diff Force= 0.000000000047 |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 27.89 | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 3.45  | diff Utot= 0.000000000001 | diff Force= 0.000000000000 |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 3.75  | diff Utot= 0.000000000004 | diff Force= 0.000000000152 |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 2.93  | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 2.80  | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 8.84  | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 2.38  | diff Utot= 0.000000000013 | diff Force= 0.000000000001 |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 6.05  | diff Utot= 0.000000000001 | diff Force= 0.000000000000 |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 4.98  | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |

Total elapsed time (s) 108.96

この比較は参照用の計算で用いた同一の Xeon クラスタ上で 20 MPI プロセスを用いて実行されました。浮動小数点の演算はコンピュータ環境だけでなく並列計算で用いたプロセッサ数にも依存するため、上記の例では、同じコンピュータを使用したにも関わらず差が現れているのが分かります。各ジョブの経過時間も出力されているため、環境によって計算スピードの差を比較することも可能です。「work/input\_example」ディレクトリには、複数のプラットフォームで生成された「runtest.result」ファイルがあります。

参照データファイルを御自身で作成する場合には、以下のコマンドによって OpenMX を実行して下さい。

```
% ./openmx -maketest
```

これにより OpenMX は「work/input\_example」ディレクトリにある「\*.dat」入力ファイルに対して、対応する「\*.out」ファイルを同ディレクトリに生成します。新しい入力 dat ファイルを追加することで次の自動実行テスト時に、対応する「\*.out」を利用することができますが、この手続きにより以前の「work/input\_example」にある出力ファイルは上書きされてしまうことに注意して下さい。開発者やヘビーユーザがコードの信頼性を確認したい場合には「自動フォース・テスター」および「自動メモリリーク・テスター」の章もご参照下さい。

参考までにいくつかの計算機による 'runtest' の結果を以下に示します。

東京大学物性研究所 System B (sekirei) (Intel Xeon E5-2680v3 12core 2.5GHz)

icc version 18.0.5, compiler option: -O3 -xHOST -ip -no-prec-div -qopenmp -Dkcomp -fp-model precise  
6 processes (MPI) x 4 thread (OpenMP)

|    |                             |                        |                           |                             |
|----|-----------------------------|------------------------|---------------------------|-----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 3.69  | diff Utot= 0.000000000034 | diff Force= 0.000000000005  |
| 2  | input_example/C60.dat       | Elapsed time(s)= 11.54 | diff Utot= 0.000000000005 | diff Force= 0.000000000006  |
| 3  | input_example/CO.dat        | Elapsed time(s)= 6.14  | diff Utot= 0.000000000106 | diff Force= 0.0000000001979 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 5.80  | diff Utot= 0.000000000364 | diff Force= 0.000000000033  |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 58.73 | diff Utot= 0.000000000003 | diff Force= 0.000000000005  |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 48.60 | diff Utot= 0.000000000010 | diff Force= 0.000000000002  |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 3.39  | diff Utot= 0.000000000001 | diff Force= 0.000000000000  |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 9.05  | diff Utot= 0.000000000016 | diff Force= 0.000000000019  |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 3.04  | diff Utot= 0.000000000002 | diff Force= 0.000000000001  |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 2.69  | diff Utot= 0.000000000000 | diff Force= 0.000000000019  |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 10.50 | diff Utot= 0.000000000085 | diff Force= 0.000000000022  |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 2.05  | diff Utot= 0.000000000003 | diff Force= 0.000000000002  |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 6.41  | diff Utot= 0.000000000617 | diff Force= 0.000000000018  |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 5.59  | diff Utot= 0.000000000000 | diff Force= 0.000000000000  |

Total elapsed time (s) 177.21

東京大学物性研究所 System C (enaga) (Intel Xeon 6148 20core 2.4GHz)

icc version 18.0.5, compiler option: -O3 -xHOST -ip -no-prec-div -qopenmp -Dkcomp -fp-model precise  
5 processes (MPI) x 4 thread (OpenMP)

|    |                             |                        |                           |                             |
|----|-----------------------------|------------------------|---------------------------|-----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 2.92  | diff Utot= 0.000000000025 | diff Force= 0.000000000002  |
| 2  | input_example/C60.dat       | Elapsed time(s)= 9.47  | diff Utot= 0.000000000005 | diff Force= 0.000000000003  |
| 3  | input_example/CO.dat        | Elapsed time(s)= 5.71  | diff Utot= 0.000000000072 | diff Force= 0.0000000001573 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 5.46  | diff Utot= 0.000000000845 | diff Force= 0.000000000111  |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 40.05 | diff Utot= 0.000000000002 | diff Force= 0.000000000066  |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 37.81 | diff Utot= 0.000000000009 | diff Force= 0.000000000001  |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 2.96  | diff Utot= 0.000000000001 | diff Force= 0.000000000001  |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 5.76  | diff Utot= 0.000000000002 | diff Force= 0.0000000000140 |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 2.44  | diff Utot= 0.000000000000 | diff Force= 0.000000000000  |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 2.39  | diff Utot= 0.000000000002 | diff Force= 0.0000000003224 |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 10.08 | diff Utot= 0.000000000129 | diff Force= 0.000000000020  |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 1.88  | diff Utot= 0.000000000001 | diff Force= 0.000000000000  |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 6.09  | diff Utot= 0.000000000272 | diff Force= 0.000000000150  |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 4.02  | diff Utot= 0.000000000000 | diff Force= 0.000000000001  |

Total elapsed time (s) 137.06

北陸先端科学技術大学 hster (Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz)

icc compiler Ver. 14.0.2.144, compiler option: -openmp -O3 -xAVX -ip -no-prec-div  
20 processes (MPI) x 1 thread (OpenMP)

|    |                             |                        |                           |                            |
|----|-----------------------------|------------------------|---------------------------|----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 5.30  | diff Utot= 0.000000000038 | diff Force= 0.000000000003 |
| 2  | input_example/C60.dat       | Elapsed time(s)= 12.53 | diff Utot= 0.000000000001 | diff Force= 0.000000000002 |
| 3  | input_example/CO.dat        | Elapsed time(s)= 10.55 | diff Utot= 0.000000000047 | diff Force= 0.000000007948 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 10.74 | diff Utot= 0.000000000381 | diff Force= 0.00000000102  |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 27.48 | diff Utot= 0.000000000001 | diff Force= 0.000000000035 |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 38.56 | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 5.76  | diff Utot= 0.000000000001 | diff Force= 0.000000000000 |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 6.73  | diff Utot= 0.000000000003 | diff Force= 0.000000000073 |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 5.00  | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 4.86  | diff Utot= 0.000000000000 | diff Force= 0.000000000020 |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 13.97 | diff Utot= 0.000000000118 | diff Force= 0.000000000001 |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 4.36  | diff Utot= 0.000000000006 | diff Force= 0.000000000002 |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 9.83  | diff Utot= 0.000000000144 | diff Force= 0.000000000079 |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 8.39  | diff Utot= 0.000000000000 | diff Force= 0.000000000001 |

Total elapsed time (s) 164.04

### 北陸先端科学技術大学院大学 CRAY-XC40 (Intel Xeon E5-2695v4 2.1GHz)

icc version 17.0.7, compiler option: -Dxt3 -O3 -axCOMMON-AVX512,CORE-AVX512,CORE-AVX2,CORE-AVX-I,AVX,SSE4.2,SSE4.1,SSE3,SSSE3,SSE2 -qopenmp

18 processes (MPI) x 2 thread (OpenMP)

|    |                             |                        |                           |                            |
|----|-----------------------------|------------------------|---------------------------|----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 4.23  | diff Utot= 0.000000000040 | diff Force= 0.000000000002 |
| 2  | input_example/C60.dat       | Elapsed time(s)= 12.40 | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 3  | input_example/CO.dat        | Elapsed time(s)= 9.09  | diff Utot= 0.000000000150 | diff Force= 0.000000009551 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 8.56  | diff Utot= 0.000000000462 | diff Force= 0.000000000004 |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 20.81 | diff Utot= 0.000000000001 | diff Force= 0.000000000014 |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 31.99 | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 4.71  | diff Utot= 0.000000000001 | diff Force= 0.000000000002 |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 4.89  | diff Utot= 0.000000000032 | diff Force= 0.000000000004 |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 4.03  | diff Utot= 0.000000000001 | diff Force= 0.000000000002 |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 3.83  | diff Utot= 0.000000000001 | diff Force= 0.000000001042 |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 12.73 | diff Utot= 0.000000000064 | diff Force= 0.000000000029 |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 3.24  | diff Utot= 0.000000000004 | diff Force= 0.000000000001 |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 8.32  | diff Utot= 0.000000000576 | diff Force= 0.000000000032 |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 6.12  | diff Utot= 0.000000000000 | diff Force= 0.000000000001 |

Total elapsed time (s) 134.96

### 名古屋大学 FX100 (PRIMEHPC FX100, SPARC64b XIIfx, 2.2Gz)

mpifccpx, compiler option: -Kfast -Kopenmp -Dnosse -Dkcomp

16 processes (MPI) x 2 thread (OpenMP)

|    |                             |                        |                           |                            |
|----|-----------------------------|------------------------|---------------------------|----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 12.29 | diff Utot= 0.000000000003 | diff Force= 0.000000000005 |
| 2  | input_example/C60.dat       | Elapsed time(s)= 29.15 | diff Utot= 0.000000000158 | diff Force= 0.000000000050 |
| 3  | input_example/CO.dat        | Elapsed time(s)= 38.71 | diff Utot= 0.000000000125 | diff Force= 0.000000003104 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 25.43 | diff Utot= 0.000000001020 | diff Force= 0.000000000007 |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 84.38 | diff Utot= 0.000000000608 | diff Force= 0.000000073199 |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 80.54 | diff Utot= 0.000000000011 | diff Force= 0.000000015689 |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 14.78 | diff Utot= 0.000000000000 | diff Force= 0.000000000000 |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 14.52 | diff Utot= 0.000000000016 | diff Force= 0.000000000001 |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 12.78 | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 13.07 | diff Utot= 0.000000000001 | diff Force= 0.000000000023 |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 37.63 | diff Utot= 0.000000000153 | diff Force= 0.000000000000 |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 9.82  | diff Utot= 0.000000000007 | diff Force= 0.000000000002 |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 26.48 | diff Utot= 0.000000000209 | diff Force= 0.000000000058 |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 19.74 | diff Utot= 0.000000000000 | diff Force= 0.000000000001 |

Total elapsed time (s) 419.32

尾崎研究室 pauli (AMD EPYC 7351P, 2.4GHz)

gcc version 7.4.0, compiler option: -Dkcomp -O3 -march=znver1 -mtune=znver1 -mfma -mavx2 -m3dnow -fomit-frame-pointer -fopenmp  
12 processes (MPI) x 1 thread (OpenMP)

|    |                             |                        |                           |                             |
|----|-----------------------------|------------------------|---------------------------|-----------------------------|
| 1  | input_example/Benzene.dat   | Elapsed time(s)= 3.32  | diff Utot= 0.000000000039 | diff Force= 0.000000000002  |
| 2  | input_example/C60.dat       | Elapsed time(s)= 13.49 | diff Utot= 0.000000000013 | diff Force= 0.000000000006  |
| 3  | input_example/CO.dat        | Elapsed time(s)= 9.13  | diff Utot= 0.000000000064 | diff Force= 0.0000000000934 |
| 4  | input_example/Cr2.dat       | Elapsed time(s)= 8.43  | diff Utot= 0.000000002324 | diff Force= 0.000000000157  |
| 5  | input_example/Crys-MnO.dat  | Elapsed time(s)= 25.40 | diff Utot= 0.000000000003 | diff Force= 0.000000000070  |
| 6  | input_example/GaAs.dat      | Elapsed time(s)= 38.09 | diff Utot= 0.000000000002 | diff Force= 0.000000000001  |
| 7  | input_example/Glycine.dat   | Elapsed time(s)= 4.41  | diff Utot= 0.000000000001 | diff Force= 0.000000000003  |
| 8  | input_example/Graphite4.dat | Elapsed time(s)= 4.83  | diff Utot= 0.000000000015 | diff Force= 0.000000000011  |
| 9  | input_example/H2O-EF.dat    | Elapsed time(s)= 3.55  | diff Utot= 0.000000000000 | diff Force= 0.000000000001  |
| 10 | input_example/H2O.dat       | Elapsed time(s)= 2.95  | diff Utot= 0.000000000001 | diff Force= 0.000000000086  |
| 11 | input_example/HMn.dat       | Elapsed time(s)= 11.85 | diff Utot= 0.000000000113 | diff Force= 0.000000000001  |
| 12 | input_example/Methane.dat   | Elapsed time(s)= 2.72  | diff Utot= 0.000000000006 | diff Force= 0.000000000001  |
| 13 | input_example/Mol_MnO.dat   | Elapsed time(s)= 8.01  | diff Utot= 0.000000000326 | diff Force= 0.000000000050  |
| 14 | input_example/Ndia2.dat     | Elapsed time(s)= 5.93  | diff Utot= 0.000000000000 | diff Force= 0.000000000000  |

Total elapsed time (s) 142.11

## 6 大規模な系の自動実行テスト

より大規模な系のベンチマーク計算によって計算機性能を知りたい場合、以下のコマンドによってテスト計算を自動実行して下さい。

MPI 並列化の場合 :

```
% mpirun -np 112 openmx -runtestL
```

MPI/OpenMP 並列化の場合 :

```
% mpirun -np 112 openmx -runtestL -nt 2
```

これらのコマンドを実行すると、OpenMX は 16 個のテスト入力ファイルを実行し、その結果を「work/large\_example」内にある参照データと比較します。比較結果（全エネルギーおよび力の絶対差分）は、「work」ディレクトリの「runtestL.result」ファイルに格納されます。参照データは、2.6 GHz Xeon クラスタマシン上で 28 個の MPI プロセスを使用して計算されたものです。絶対差分が小数点以下 7 桁以内であればインストールは正常に行われたと判断できます。例として、自動実行テストで生成された「runtestL.result」を以下に示します。

|    |                                |                          |                           |                            |
|----|--------------------------------|--------------------------|---------------------------|----------------------------|
| 1  | large_example/5_5_13COb2.dat   | Elapsed time(s)= 52.78   | diff Utot= 0.000000000020 | diff Force= 0.000000000004 |
| 2  | large_example/B2C62_Band.dat   | Elapsed time(s)= 403.51  | diff Utot= 0.000000000001 | diff Force= 0.000000063810 |
| 3  | large_example/CG15c-DC-LNO.dat | Elapsed time(s)= 103.31  | diff Utot= 0.000000000269 | diff Force= 0.000000000551 |
| 4  | large_example/DIA512-1.dat     | Elapsed time(s)= 49.35   | diff Utot= 0.00000027379  | diff Force= 0.00000031436  |
| 5  | large_example/FeBCC.dat        | Elapsed time(s)= 80.54   | diff Utot= 0.000000000016 | diff Force= 0.000000000001 |
| 6  | large_example/GEL.dat          | Elapsed time(s)= 44.95   | diff Utot= 0.000000000009 | diff Force= 0.000000000004 |
| 7  | large_example/GFRAG.dat        | Elapsed time(s)= 27.68   | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 8  | large_example/GGFF.dat         | Elapsed time(s)= 643.36  | diff Utot= 0.000000000037 | diff Force= 0.000000000809 |
| 9  | large_example/MCCN.dat         | Elapsed time(s)= 82.04   | diff Utot= 0.00000005885  | diff Force= 0.00000003486  |
| 10 | large_example/Mn12_148_F.dat   | Elapsed time(s)= 74.25   | diff Utot= 0.000000000015 | diff Force= 0.000000000010 |
| 11 | large_example/N1C999.dat       | Elapsed time(s)= 1212.42 | diff Utot= 0.000000000035 | diff Force= 0.000000000390 |
| 12 | large_example/Ni63-O64.dat     | Elapsed time(s)= 70.90   | diff Utot= 0.000000000211 | diff Force= 0.000000000008 |
| 13 | large_example/Pt63.dat         | Elapsed time(s)= 58.76   | diff Utot= 0.00000001297  | diff Force= 0.000000000242 |
| 14 | large_example/SialicAcid.dat   | Elapsed time(s)= 16.75   | diff Utot= 0.000000000001 | diff Force= 0.000000000001 |
| 15 | large_example/ZrB2_2x2.dat     | Elapsed time(s)= 133.10  | diff Utot= 0.000000000044 | diff Force= 0.000000000020 |
| 16 | large_example/nsV4Bz5.dat      | Elapsed time(s)= 99.37   | diff Utot= 0.00000004771  | diff Force= 0.00000003167  |

Total elapsed time (s) 3153.08

この結果は 112 個の MPI プロセスを用いて Xeon クラスタ計算機で計算したものです。この自動実行テストは大量のメモリを必要とするため、少数の使用コア数で実行した際にはメモリの領域侵害を起こす場合があります。また上記の例では 112 コアを使用しても約 53 分の時間を要することが分かります。「大規模計算」の章にも別の大規模ベンチマーク計算の例の説明がありますので、参照して下さい。

## 7 入力ファイル

### 7.1 例：メタン分子

「work」ディレクトリにある「Methane.dat」入力ファイルを例として下記に示します。入力ファイルは柔軟なデータ書式 (flexible data format) で記述されます。この形式ではキーワードの後に対応するパラメータを指定します。キーワードを記載する順番は自由であり、また空白行やコメントも自由に使用できます。下記の例では特定の様式で書かれていますが、キーワードやオプション (パラメータ) は、大文字、小文字の区別をしておらず、ユーザーの好みに応じて、大文字、小文字、あるいは大文字小文字混在で指定することができます。

```
#
# File Name
#

System.CurrrentDirectory      ./      # default=./
System.Name                   met
level.of.stdout               1      # default=1 (1-3)
level.of.fileout              1      # default=1 (0-2)

#
# Definition of Atomic Species
#

Species.Number                2
<Definition.of.Atomic.Species
H   H5.0-s1                   H_PBE19
C   C5.0-s1p1                 C_PBE19
Definition.of.Atomic.Species>

#
# Atoms
#

Atoms.Number                  5
Atoms.SpeciesAndCoordinates.Unit  Ang # Ang|AU
<Atoms.SpeciesAndCoordinates
1  C      0.000000   0.000000   0.000000   2.0  2.0
2  H     -0.889981  -0.629312   0.000000   0.5  0.5
3  H      0.000000   0.629312  -0.889981   0.5  0.5
4  H      0.000000   0.629312   0.889981   0.5  0.5
5  H      0.889981  -0.629312   0.000000   0.5  0.5
Atoms.SpeciesAndCoordinates>
Atoms.UnitVectors.Unit        Ang # Ang|AU
<Atoms.UnitVectors
10.0  0.0  0.0
 0.0 10.0  0.0
 0.0  0.0 10.0
```

```

Atoms.UnitVectors>

#
# SCF or Electronic System
#

scf.XcType          GGA-PBE      # LDA|LSDA-CA|LSDA-PW|GGA-PBE
scf.SpinPolarization  off          # On|Off|NC
scf.ElectronicTemperature 300.0      # default=300 (K)
scf.energycutoff    120.0      # default=150 (Ry)
scf.maxIter         100          # default=40
scf.EigenvalueSolver cluster     # DC|Cluster|Band
scf.Kgrid           1 1 1        # means n1 x n2 x n3
scf.Mixing.Type      rmm-diis    # Simple|Rmm-Diis|Gr-Pulay|Kerker|Rmm-Diisk
scf.Init.Mixing.Weight 0.30       # default=0.30
scf.Min.Mixing.Weight 0.001      # default=0.001
scf.Max.Mixing.Weight 0.400      # default=0.40
scf.Mixing.History    7          # default=5
scf.Mixing.StartPulay 5          # default=6
scf.criterion        1.0e-10     # default=1.0e-6 (Hartree)

#
# MD or Geometry Optimization
#

MD.Type              nomd        # Nomd|Opt|NVE|NVT_VS|NVT_NH
                                # Constraint_Opt|DIIS
MD.maxIter           1          # default=1
MD.TimeStep          1.0        # default=0.5 (fs)
MD.Opt.criterion     1.0e-4     # default=1.0e-4 (Hartree/Bohr)

```

## 7.2 キーワード

各キーワードについて下記に説明していきます。この章ではOpenMXのすべてのキーワードは網羅しておらず、また各キーワードについては対応する章で更に詳しく説明しています。本マニュアルの最後に記載されているインデックスをご参照頂ければ、各キーワードに関連したページ番号が分かります。

### ファイル名

#### System.CurrentDir

このキーワードにより、出力ファイルの出力ディレクトを指定します。デフォルト値は「./」です。

#### System.Name

このキーワードで出力ファイルのファイル名を指定します。

#### DATA.PATH

VPS および PAO ディレクトリへのパスを下記のように指定します。

```
DATA.PATH ../DFT_DATA19 # default=../DFT_DATA19
```

絶対パス、相対パスとも指定可能です。デフォルト値は「../DFT\_DATA19」です。

#### level.of.stdout

このキーワードにより、標準出力への出力情報のレベルを制御します。「level.of.stdout=0」を指定した場合、最小限の情報が出力されます。「level.of.stdout=1」の場合、最小限の出力に加えて追加の情報が出力されます。「level.of.stdout=2」は開発者向けのオプションです。デフォルト値は1です。

#### level.of.fileout

このキーワードにより、出力ファイルへの出力情報のレベルを制御します。「level.of.fileout=0」を指定した場合、最小限の情報が出力されます（Gaussian cube ファイルおよびグリッドファイルの出力無し）。「level.of.fileout=1」は標準的な出力レベルです。「level.of.fileout=2」の場合、標準の出力に加えて追加の情報が出力されます。デフォルト値は1です。

## 原子種の定義

#### Species.Number

このキーワードにより、系に含まれる原子種の数指定します。

#### Definition.of.Atomic.Species

擬原子基底関数と擬ポテンシャルの両方のファイル名を明記することにより、原子種を指定して下さい。これらのファイルはそれぞれ「DFT\_DATA19/PAO」および「DFT\_DATA19/VPS」ディレクトリに置かなければなりません。例えば以下のように指定します。

```
<Definition.of.Atomic.Species
H   H5.0-s1>1p1>1      H_CA19
C   C5.0-s1>1p1>1      C_CA19
Definition.of.Atomic.Species>
```

原子種の指定は「<Definition.of.Atomic.Species」で始め、「Definition.of.Atomic.Species>」で終わらなければなりません。1番目の列では任意の原子種の名前を指定します。この名前は、後述する

「Atoms.SpeciesAndCoordinates」キーワードにて原子座標を指定するのに使います。2番目の列では擬原子基底関数の拡張子無しのファイル名と、プリミティブ基底関数の数および縮約された基底関数の数を指定します。例では基底関数は「H4.0-s1>1p1>1」となっていますが、H4.0が「DFT\_DATA19/PAO」ディレクトリにある擬原子基底関数の拡張子無しのファイル名を示し、s1>1は、1つのs軌道のプリミティブ軌道から1つの最適化された軌道が作られていることに対応します。つまり縮約は行わないことを意味します。縮約が無い場合、「s1>1」と記述する代わりに「s1」という簡単な記法を使用できます。つまり、「H4.0-s1p1」は「H4.0-s1>1p1>1」と同等ということです。3番目の列では擬ポテンシャルの拡張子無しのファイル名を指定します。このファイルは「DFT\_DATA19/VPS」ディレクトリ内に存在し

なければなりません。同一元素に対して異なる基底関数と擬ポテンシャルを割り当てることで、異なる原子種を定義することもできます。例えば複数の原子種を以下のように指定できます。

```
<Definition.of.Atomic.Species
H1  H5.0-s1p1      H_CA19
H2  H5.0-s2p2d1   H_CA19
C1  C5.0-s2p2     C_CA19
C2  C5.0-s2p2d2   C_CA19
Definition.of.Atomic.Species>
```

この柔軟性の高い記述法を活用し、系の特性を決定している領域に属する原子には高精度の基底関数を割り当て、不活性な領域に属する原子には低精度の基底関数を割り当てることも可能であり、計算量の削減には有効な手段です。

## 原子

### Atoms.Number

系の合計原子数を「Atoms.Number」キーワードで指定します。

### Atoms.SpeciesAndCoordinates.Unit

原子座標の単位を「Atoms.SpeciesAndCoordinates.Unit」キーワードで指定します。オングストロームの場合は「Ang」、原子単位の場合は「AU」を指定します。「FRAC」と記述することで、規格化座標を指定することも可能です。規格化座標の場合には、後述する「Atoms.UnitVectors」のキーワードで与えられた a、b、c ベクトルを基底として、座標を指定します。その際には 0.0~1.0 の範囲で座標が設定可能で、この範囲を超える座標は入力ファイルの読み込み後に自動的に調整されます。

### Atoms.SpeciesAndCoordinates

原子座標および各スピン毎の電子数を「Atoms.SpeciesAndCoordinates」キーワードで以下の例のように指定します。

```
<Atoms.SpeciesAndCoordinates
 1  C      0.000000   0.000000   0.000000   2.0  2.0
 2  H     -0.889981  -0.629312   0.000000   0.5  0.5
 3  H      0.000000   0.629312  -0.889981   0.5  0.5
 4  H      0.000000   0.629312   0.889981   0.5  0.5
 5  H      0.889981  -0.629312   0.000000   0.5  0.5
Atoms.SpeciesAndCoordinates>
```

記述は「<Atoms.SpeciesAndCoordinates」で始まり、「Atoms.SpeciesAndCoordinates>」で終わります。1 列目には原子を特定する連番を記述します。2 列目には事前に「Definition.of.Atomic.Species」の 1 列目で定義した原子種を指定します。3~5 列目には、それぞれ  $x$ 、 $y$ 、 $z$  座標を指定します。

「Atoms.SpeciesAndCoordinates.Unit」キーワードで「FRAC」を指定している場合は 3~5 列目はそれぞれ a、b、c 軸の規格化座標になり、値は 0.0~1.0 の範囲で指定します。この範囲を外れる数値は入

力ファイルが読み込まれる段階で自動的に調整されます。6 および 7 列目には、各原子のアップスピンとダウンスピン状態のそれぞれの初期電子数を設定します。アップスピン電子数とダウンスピン電子数の合計は原子の価電子の数に等しくなるようにします。原子の価電子数は表 1 と表 2 に記載されています。「LSDA-CA」, 「LSDA-PW」, あるいは「GGA-PBE」を使用してスピン分極系を計算する場合、6 および 7 列目の指定によって各原子に初期スピンを指定することができます。これにより強磁性状態や反強磁性状態などの計算が可能です。また SCF の収束を加速するためには、この初期スピン配置は基底状態に近い配置からスタートすることが重要です。

#### Atoms.UnitVectors.Unit

単位胞 (ユニットセル) におけるベクトルの単位を「Atoms.UnitVectors.Unit」キーワードで指定します。オングストロームの場合は「Ang」、原子単位の場合は「AU」を指定します。

#### Atoms.UnitVectors

単位胞のベクトル a、b、c を Atoms.UnitVectors キーワードで以下の様に指定します。

```
<Atoms.UnitVectors
  10.0  0.0  0.0
    0.0 10.0  0.0
    0.0  0.0 10.0
Atoms.UnitVectors>
```

記述は「<Atoms.UnitVectors」で始まり、「Atoms.UnitVectors>」で終わります。第 1~3 行はそれぞれ単位胞のベクトル a、b、c に対応します。クラスタ計算の際にこのキーワードが指定されないと周期的に配置された孤立系の間で基底関数が重ならないように、単位胞の大きさが自動的に決定されます。詳細に関しては「セルサイズの自動決定」の章もご参照下さい。

## SCF または電子系

#### scf.XcType

交換相関ポテンシャルを「scf.XcType」キーワードで指定します。OpenMX Ver. 3.9 においては「LDA」, 「LSDA-CA」, 「LSDA-PW」, 「GGA-PBE」から選択できます。ここで「LSDA-CA」は Ceperley-Alder [2] の局所スピン密度関数、「LSDA-PW」は、その GGA 形式において密度勾配をゼロとした Perdew-Wang 局所スピン密度関数 [4] です。「GGA-PBE」は Perdew らが提案する GGA 汎関数 [5] です。

#### scf.SpinPolarization

電子系の非スピン分極あるいはスピン分極は「scf.SpinPolarization」キーワードで指定します。スピン分極の計算を行う場合は「ON」を指定し、非スピン分極の計算を行う場合は「OFF」を指定します。「scf.XcType」キーワードで「LDA」を使用する場合は「scf.SpinPolarization」を「OFF」に設定して下さい。前述の 2 つのオプションの他、ノンコリニア DFT 計算を行う場合には「NC」というオプションを指定して下さい。この計算については「ノンコリニア DFT」の章もご参照下さい。

#### scf.partialCoreCorrection

交換相関エネルギーおよびそのポテンシャルの計算において部分内殻補正 (Partial Core Correction, PCC) を行う場合には、「scf.partialCoreCorrection」キーワードを指定します。これを「ON」に指定す

ると PCC を行い、「OFF」の場合には PCC を行いません。通常は、このフラグは「ON」にするべきです。何故ならば、PCC 擬ポテンシャルを使用する場合には PCC を行うべきであり、また非 PCC 擬ポテンシャルの場合でも PCC 電荷がゼロなので結果には影響を及ぼさないからです。デフォルトは「ON」です。

#### scf.Hubbard.U

LDA+ $U$  および GGA+ $U$  計算の場合、「scf.Hubbard.U」キーワードを「ON」に設定して下さい(ON|OFF)。デフォルトでは「OFF」に設定されています。

#### scf.Hubbard.Occupation

LDA+ $U$  法では、「scf.Hubbard.Occupation」キーワードを指定することで、「onsite」、「full」および「dual」の 3 つの占有数演算子から選択することができます。

#### Hubbard.U.values

下記のキーワードにより各原子種の軌道に対する実効的な  $U$  の値を定義します。

```
<Hubbard.U.values          #   eV
Ni  1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 4.0 2d 0.0
    0  1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.0
Hubbard.U.values>
```

記述は「<Hubbard.U.values」で始まり、「Hubbard.U.values>」で終わります。

「Definition.of.Atomic.Species」で指定した全ての基底軌道に対して、上記の形式にて実効的な  $U$  の値を設定しなければなりません。「1s」や「2s」はそれぞれ割り当てられた最初と二番目の  $s$  軌道を示し、「1s」の後にくる数字が最初の  $s$  軌道に対する実効的な  $U$  値 (eV) です。同様の規則が、 $p$  および  $d$  軌道にも当てはまります。

#### scf.Constraint.NC.Spin

スピン磁気モーメントのノンコリニア方位を制御する制約条件付 DFT 計算を行う場合には「scf.Constraint.NC.Spin」キーワードを「ON」に設定して下さい。このキーワードは「ON」か「OFF」に設定することができます。

#### scf.Constraint.NC.Spin.v

スピン磁気モーメントのノンコリニア方位を制御する制約条件付 DFT 計算を行う場合には、「scf.Constraint.NC.Spin.v」キーワードによってペナルティ関数の前因子 (eV) を指定します。

#### scf.ElectronicTemperature

「scf.ElectronicTemperature」キーワードにて電子温度 (K) を設定します。デフォルトでは 300 (K) に設定されています。

#### scf.energycutoff

「scf.energycutoff」キーワードで積分グリッド間隔を決定するカットオフエネルギーを指定します。この積分グリッドは差電子クーロンポテンシャルと交換相関ポテンシャルに対する行列要素の計算および高速フーリエ変換 (FFT) を用いた Poisson 方程式の解法のために使用されます。デフォルトでは 150 (Ryd) に設定されています。

#### scf.Ngrid

「scf.Ngrid」キーワードによって a、b、c 軸を離散化させる際のグリッド数を指定します。通常は「scf.energycutoff」で指定した値を使って実空間の離散化を行います。が、「scf.Ngrid」でグリッド数を指定した場合には、「scf.energycutoff」ではなくこちらが優先されます。

#### scf.maxIter

SCF の最大反復回数を「scf.maxIter」キーワードで設定します。SCF 反復ループは、収束条件が満たされない場合でもこのキーワードで指定した回数で終了します。デフォルト値は 40 です。

#### scf.EigenvalueSolver

固有値問題の計算手法を「scf.EigenvalueSolver」で指定します。O(N) 分割統治法は「DC」、局在自然軌道による O(N) 分割統治法は「DC-LNO」、O(N) クリロフ部分空間法は「Krylov」、数値厳密な低次スケール法は「ON2」、クラスタ計算は「Cluster」、バンド計算は「Band」を指定します。

#### scf.Kgrid

OpenMX では等間隔メッシュにより k 空間の第 1 ブリルアンゾーンを離散します。「scf.EigenvalueSolver」キーワードにより「Band」を指定した場合、k 空間の第 1 ブリルアンゾーンを離散化するための格子の数 (n1, n2, n3) を「scf.Kgrid」キーワードで指定しなければなりません。k 空間の逆格子ベクトル  $\tilde{a}$ 、 $\tilde{b}$ 、 $\tilde{c}$  を離散化するために「n1 n2 n3」のように指定して下さい。

#### scf.ProExpn.VNA

中性原子ポテンシャル VNA を射影演算子で展開する場合 [42] には、「scf.ProExpn.VNA」キーワードを「ON」に設定し、それ以外の場合は「OFF」を設定して下さい。デフォルトでは「ON」になっています。

```
scf.ProExpn.VNA      ON      # ON|OFF, default = ON
```

「scf.ProExpn.VNA=OFF」と設定されている場合、VNA ポテンシャルの行列要素は実空間の離散グリッドを用いて計算されます。

#### scf.Mixing.Type

「scf.Mixing.Type」キーワードによって、SCF 計算の次の反復ステップに入力される電子密度 (もしくは密度行列) を生成するための電子密度混合法を指定します。単純混合法 (Simple)、GR-Pulay 法 (Guaranteed-Reduction Pulay) [57]、RMM-DIIS 法 [58]、Kerker 法 [59]、RMM-DIISK 法 [58]、RMM-DIISV 法 [58] あるいは RMM-DIISH 法 [58] のいずれかを指定することができます。OpenMX の単純混合法は収束の速度を上げるために収束履歴を参照するように改良されています。「GR-Pulay」、「RMM-DIIS」、「Kerker」、「RMM-DIISK」、「RMM-DIISV」あるいは「RMM-DIISH」を使用する際には、以下の点に注意することで SCF 計算の高速化が可能です。

- Pulay 法に準拠する方法で混合を開始する前には、ある程度の SCF 収束が達成されている必要があります。そのため、やや大きめの「scf.Mixing.StartPulay」の値を使用します。「scf.Mixing.StartPulay」の適切な値は 10 ~ 30 です。
- 金属系の場合は「scf.ElectronicTemperature」で高温の電子温度を指定します。「scf.ElectronicTemperature」があまりに低いと数値的に不安定な挙動がよく見られます。
- 「scf.Mixing.History」の値を大きくします。ほとんどの場合、「scf.Mixing.History=30」が妥当な値です。

前述した混合法のうち、多くの系に対して最もロバスト性が高い手法は「RMM-DIISK」です。

#### **scf.Init.Mixing.Weight**

単純混合法、GR-Pulay 法、RMM-DIIS 法、Kerker 法、RMM-DIISK 法、RMM-DIISV 法、および RMM-DIISH 法で使用する初期の混合比を「scf.Init.Mixing.Weight」キーワードで指定します。有効な範囲は  $0 < \text{scf.Init.Mixing.Weight} < 1$  で、デフォルト値は 0.3 です。

#### **scf.Min.Mixing.Weight**

単純混合法および Kerker 混合法における混合比の下限を「scf.Min.Mixing.Weight」で指定します。デフォルトでは 0.001 に設定されています。

#### **scf.Max.Mixing.Weight**

単純混合法および Kerker 混合法における混合比の上限を「scf.Max.Mixing.Weight」で指定します。デフォルトでは 0.4 に設定されています。

#### **scf.Kerker.factor**

Kerker および RMM-DIISK 混合法における Kerker 因子を指定します。このキーワードが指定されない場合、適切な値が自動的に設定されます。詳しくは「SCF 収束」の章を参照して下さい。

#### **scf.Mixing.History**

GR-Pulay 法 [57]、RMM-DIIS 法 [58]、RMM-DIISK 法 [58]、RMM-DIISV 法 [58] 及び RMM-DIISH 法 [58] では、SCF の次の反復ステップにおける入力電子密度（ハミルトニアン）を、過去の SCF 反復の電子密度（ハミルトニアン）に基づき推定します。「scf.Mixing.History」キーワードは、この推定に使用する過去の SCF 反復ステップ数を指定します。例えば、「scf.Mixing.History」を 3 に設定した場合、6 回目の SCF 反復では過去の第 5、4、3 ステップの電子密度（ハミルトニアン）が考慮されます。30 程度が適切な選択値です。

#### **scf.Mixing.StartPulay**

GR-Pulay 法、RMM-DIIS 法、Kerker 法、RMM-DIISK 法、RMM-DIISV 法および RMM-DIISH 法を開始する SCF ステップを「scf.Mixing.StartPulay」で指定します。これらの方法を開始するまでの SCF ステップでは単純混合法あるいは Kerker 混合法が使用されます。デフォルトでは 6 に設定されています。

#### **scf.Mixing.EveryPulay**

Pulay タイプの混合法における残差ベクトルは、SCF ステップが進むに従い、相互に線形従属になる傾向にあり、このため収束が困難になります。線形従属を避ける方法のひとつは、Kerker 混合の合間に時々 Pulay タイプの混合を挿入するというものです。この方法における Pulay タイプ混合の使用頻度を、「scf.Mixing.EveryPulay」キーワードで指定します。例えば「scf.Mixing.EveryPulay=5」と設定した場合、SCF 反復 5 回につき 1 回の Pulay 混合が実施され、その他の反復では Kerker 法が使用されます。「scf.Mixing.EveryPulay=1」は従来の Pulay タイプ混合と同じものになります。このキーワード「scf.Mixing.EveryPulay」は、「RMM-DIISK」でのみサポートされており、デフォルト値は 1 です。

#### **scf.criterion**

SCF 計算の収束条件を「scf.criterion」キーワードで指定します（Hartree 単位）。SCF 反復は、 $dU_{\text{ele}} < \text{scf.criterion}$  という条件が満たされると終了します。ここで  $dU_{\text{ele}}$  とは、現在の SCF 反復とひとつ前の反復とのバンドエネルギーの絶対差です。デフォルト値は  $1.0 \times 10^{-6}$ （Hartree 単位）です。

### scf.Electric.Field

「scf.Electric.Field」キーワードによって、のこぎり波形で表される外部均一電場を導入することが可能です。例えば、a軸に沿って1.0 GV/m ( $10^9$  V/m)の電場を適用するには、次のように指定します。

```
scf.Electric.Field 1.0 0.0 0.0 # default=0.0 0.0 0.0 (GV/m)
```

電場の符号は電子に作用するものとして定義されています。デフォルト値は「0.0 0.0 0.0」です。

### scf.system.charge

電子および正孔のドーピングの量を「scf.system.charge」キーワードで指定します。プラス・マイナス符号はそれぞれ正孔と電子のドーピングを表します。デフォルト値は0です。

### scf.SpinOrbit.Coupling

スピン軌道相互作用を有効にする場合にはこのキーワードを「ON」に設定し、それ以外では「OFF」に設定して下さい。スピン軌道相互作用を使用する場合、ノンコリニアDFT法の枠組みにおいてj依存擬ポテンシャルを使用しなければなりません。これについては「相対論的效果」の章も参照して下さい。

## 1D FFT

### 1DFFT.EnergyCutoff

局在擬原子軌道と非局所ポテンシャルのプロジェクタの動径関数をフーリエ変換するエネルギー領域を「1DFFT.EnergyCutoff」キーワードで指定します。このエネルギー領域はフーリエ空間における動径方向の最大値となります。デフォルト値は3600 (Ryd)です。

### 1DFFT.NumGridK

0から1DFFT.EnergyCutoffまでのエネルギー領域は「1DFFT.NumGridK」キーワードで指定されるグリット数で離散化されます。局在擬原子軌道および非局所ポテンシャルのプロジェクタの動径関数のフーリエ変換された値は、k空間上の動径方向の関数として、0から1DFFT.EnergyCutoffまでの範囲で、このグリット上で保持されています。デフォルト値は900です。

### 1DFFT.NumGridR

実空間上の動径グリット数を「1DFFT.NumGridR」キーワードで指定します。この値は局在擬原子軌道および非局所ポテンシャルのプロジェクタの動径関数をフーリエ変換する際の数値積分に使用されません。デフォルト値は900です。

## 軌道最適化

### orbitalOpt.Method

軌道最適化の方法を「orbitalOpt.Method」キーワードで指定します。軌道最適化を行わない場合はこの値を「OFF」に設定して下さい。軌道最適化を行う場合は、次のオプションから選択します。「atoms」を指定した場合、各原子単位の基底関数が最適化されます。「species」を指定した場合、各原子種の基底関数が最適化されます。「atom」および「species」のどちらの場合でも、動径波動関数 $R$ は磁気量子数には依存しないという条件のもとで最適化されます。これにより最適化基底関数を用いた計算において、全エネルギーの回転不変性が保証されます。「atom」を指定した場合には最適化された基底関数は化学

的環境に依存するため、同一原子種であっても原子毎に異なります。一方、「species」を指定した場合には「Definition.of.Atomic.Species」で定義した同一名の原子種の基底関数は同一の軌道に最適化されます。ほぼ同様の化学的環境を持つ原子に対して同一の最適化基底を生成したい場合に、有用な方法です。

#### **orbitalOpt.scf.maxIter**

軌道最適化における SCF 反復の最大回数を「orbitalOpt.scf.maxIter」キーワードで指定します。

#### **orbitalOpt.Opt.maxIter**

軌道最適化の反復の最大回数を「orbitalOpt.Opt.maxIter」キーワードで指定します。軌道最適化の反復は、収束条件が達成しなかった場合でも、同キーワードで設定した回数で終了します。

#### **orbitalOpt.Opt.Method**

軌道最適化の収束方法として、2つの手法がサポートされています。「EF」は固有ベクトル追跡法、「DIIS」は反復部分空間における直接反転法です。それぞれのアルゴリズムは構造最適化のそれと同じです。「orbitalOpt.Opt.Method」キーワードでは「EF」あるいは「DIIS」を指定して下さい。

#### **orbitalOpt.StartPulay**

「orbitalOpt.StartPulay」キーワードで指定した最適化ステップから、準ニュートン法である「EF」または「DIIS」法を開始します。

#### **orbitalOpt.HistoryPulay**

準ニュートン法である「EF」および「DIIS」法において、次ステップでの縮約係数を推定するために参照する過去のステップ数を「orbitalOpt.HistoryPulay」キーワードで指定します。

#### **orbitalOpt.SD.step**

準ニュートン法である「EF」および「DIIS」法を開始するまでの最適化ステップは最急降下法が適用されます。最急降下法で使用する前因子は「orbitalOpt.SD.step」キーワードで指定します。多くのケースにおいて、「orbitalOpt.SD.step」の適切な値は 0.001 程度となります。

#### **orbitalOpt.criterion**

軌道最適化の収束条件 ( $(\text{Hartree}/\text{bohr})^2$ ) を「orbitalOpt.criterion」キーワードで指定します。「微分のノルム <orbitalOpt.criterion> という条件が満たされた時に反復ループが終了します。

#### **CntOrb.fileout**

最適化動径関数をファイルに出力したい場合は、「CntOrb.fileout」キーワードを「ON」にする必要があります。

#### **Num.CntOrb.Atoms**

最適化動径関数をファイルに出力する際の原子数を「Num.CntOrb.Atoms」キーワードで指定します。

#### **Atoms.Cont.Orbitals**

「Atoms.SpeciesAndCoordinates」キーワードの第 1 列で定義した原子の通し番号を用いて、最適化基底を出力する原子を「Atoms.Cont.Orbitals」キーワードで次のように指定します。

```
<Atoms.Cont.Orbitals
  1
  2
Atoms.Cont.Orbitals>
```

記述は「<Atoms.Cont.Orbitals」で始め、「Atoms.Cont.Orbitals>」で終わります。行の数は、「Atoms.Cont.Orbitals」で記述する数字と整合性がなければなりません。例えば、最適化擬原子軌道が「C\_1.pao」と「H\_2.pao」などとして保存された場合、その原子種名はキーワード「Definition.of.Atomic.Species」の設定における第一列の記号に対応し、記号の後の数字はキーワード「Atoms.SpeciesAndCoordinates」の設定における第一列の数値を意味します。これらの出力ファイル「C\_1.pao」と「H\_2.pao」は基底関数の入力データとして使用可能です。

## O(N) SCF 計算

### orderN.HoppingRanges

各原子を中心とする球の半径 (Å) を「orderN.HoppingRanges」キーワードで定義します。DC 法、DC-LNO 法および O(N) クリロフ部分空間法では、この球内に含まれる原子を選択することで切り取られたクラスタが構成されます。

### orderN.KrylovH.order

切り取られた各クラスタのハミルトニアンに対し、クリロフ部分空間の次元を「orderN.KrylovH.order」キーワードで指定します。

### orderN.KrylovS.order

後述のキーワードで「orderN.Exact.Inverse.S=off」に設定した場合、重なり積分の逆行列はクリロフ部分空間法を用いて近似されます。この時、切り取られた各クラスタに対する重なり行列のクリロフ部分空間法の次元を「orderN.KrylovS.order」キーワードで指定します。デフォルト値は、「orderN.KrylovH.order」で設定した値の 4 倍です。

### orderN.Exact.Inverse.S

キーワード「orderN.Exact.Inverse.S」を「on」に設定すると、切り取られた各クラスタの重なり行列の逆行列は厳密に評価されます。off に設定する場合は前述のキーワード「orderN.KrylovS.order」の項を参照して下さい。デフォルトでは「on」に設定されています。

### orderN.Recalc.Buffer

キーワード「orderN.Recalc.Buffer」を「on」に設定すると、バッファ行列は各 SCF 反復毎に再計算されます。「off」の場合にはバッファ行列は初回の SCF ステップで計算され、その後の SCF 反復では固定されます。デフォルトでは「on」に設定されています。

### orderN.Expand.Core

キーワード「orderN.Expand.Core」を「on」に設定すると、コア領域は半径  $1.2 \times r_{\min}$  の球の中にある原子から構成されます。ここで  $r_{\min}$  は、中心原子と最隣接原子間の距離です。このコア領域はクリロフ部分空間を生成するときの第 1 ステップで使用されるベクトル群を定義します。「orderN.Expand.Core」が「off」の場合、中心の原子がコア領域と見なされます。デフォルトでは「on」に設定されています。

## MD(分子動力学) あるいは構造最適化

## MD.Type

分子動力学計算あるいは構造最適化のタイプを指定します。現在使用できるオプションは以下の通りです。MD 無し (「Nomd」)、NVE アンサンブル MD (「NVE」)、速度スケーリング法による NVT アンサンブル MD (「NVT\_VS」) [30]、Nose-Hoover 法による NVT アンサンブル MD (「NVT\_NH」) [31]、複数熱浴付き MD (「NVT\_VS2」あるいは「NVT\_VS4」)、最急降下法による構造最適化 (「SD」)、DIIS 最適化法 (「DIIS」)、固有値ベクトル追跡法 (「EF」) [63]、及び合理的関数法 (「RF」) [64] です。その他にセル最適化のためのオプション (OptC1、OptC2、OptC3、OptC4、OptC5、OptC6、OptC7、RFC5、RFC6、RFC7) も利用可能です。詳細は「構造最適化」、「セル可変構造最適化」および「分子動力学」の章を参照して下さい。

## MD.Fixed.XYZ

次のキーワードを指定することで、構造最適化および分子動力学シミュレーションの際に任意の原子座標を初期座標に固定することができます。各原子の  $x$ 、 $y$ 、 $z$  座標に対して、以下のように設定します。

```
<MD.Fixed.XYZ
  1  1  1  1
  2  1  0  0
MD.Fixed.XYZ>
```

上記の例は 2 つの原子から構成される系の場合です。系に  $N$  個の原子がある場合、 $N$  個分の指定が必要です。第 1 列は「Atoms.SpeciesAndCoordinates」で個々の原子を特定するのに使用した連番です。第 2~4 列は、それぞれ  $x$ 、 $y$ 、 $z$  座標のフラグです。フラグが「1」だと座標は固定、「0」だと緩和されます。上記の例では、原子 1 の  $x$ 、 $y$ 、 $z$  座標は固定、原子 2 は  $x$  座標のみ固定されています。デフォルトでは全ての座標が緩和される指定になっています。このキーワードによる原子座標の固定は、すべての構造最適化および分子動力学の方法において有効です。

## MD.maxIter

MD 計算や構造最適化計算における最大の反復回数を「MD.maxIter」キーワードで指定します。

## MD.TimeStep

時間ステップ (fs) を「MD.TimeStep」キーワードで指定します。

## MD.Opt.criterion

「MD.Type」キーワードによって構造最適化の方法を選択している場合には、「MD.Opt.criterion」キーワードでその収束条件 (Hartree/Bohr) を設定します。原子にかかる力の最大絶対値が「MD.Opt.criterion」で指定した値より小さくなった場合に、構造最適化は終了します。

## MD.Opt.DIIS.History

「DIIS」、「EF」、「RF」による構造最適化を行う場合、「MD.Opt.DIIS.History」キーワードは構造最適化のために参照する過去のステップ数を指定します。デフォルト値は 3 です。

## MD.Opt.StartDIIS

「DIIS」、「EF」、「RF」による構造最適化を開始するステップを「MD.Opt.StartDIIS」キーワードで指定します。DIIS タイプの構造最適化を開始する以前のステップでは最急降下法が使用されます。デフォルト値は 5 です。

## MD.TempControl

キーワード「MD.TempControl」により NVT アンサンブルの分子動力学計算における原子運動の温度を指定します。「NVT\_VS」を選択した場合、原子運動の温度を下記の例のように制御できます。

```
<MD.TempControl
  3
  100  2  1000.0  0.0
  400 10   700.0  0.4
  700 40   500.0  0.7
MD.TempControl>
```

記述は「<MD.TempControl」で開始し、「MD.TempControl>」で終わります。最初の数字「3」は、続く温度指定の行数を指定します。例では3行あります。後続する行の第1列はMDステップ数を意味し、第2列は速度スケーリングを行うMDステップの間隔を指定します。例では、100ステップ目までは2ステップ毎に、100~400ステップ間は10ステップ毎に、400~700ステップ間は40ステップ毎に速度スケーリングを行います。第3、4列はそれぞれ温度(K)とスケーリングパラメータ $\alpha$ を指定します。詳細は「分子動力学」の章を参照して下さい。一方、NVT\_NHの場合、以下の記述で原子運動の温度を制御できます。

```
<MD.TempControl
  4
  1    1000.0
 100  1000.0
 400   700.0
 700   600.0
MD.TempControl>
```

記述は「<MD.TempControl」で開始し、「MD.TempControl>」で終わります。最初の数字「4」は、続く温度指定の行数を指定します。この例では4行あります。後続する行の第1、2列は、それぞれMDステップ数と原子運動の温度を指定します。指定されたMDステップ間の温度は線形補完されます。

## NH.Mass.HeatBath

キーワード「MD.Type」によって「NVT\_NH」を選択した場合、このキーワードで熱浴の質量を設定します。デフォルト値は20です。次元は長さ<sup>2</sup> × 質量です。この表記では長さにはボア半径、質量に統一原子質量単位（炭素原子の主同位体の質量を12.0とする単位）を用います。

## MD.Init.Velocity

分子動力学のシミュレーションでは、下記の記述により各原子に初期速度を与えることができます。

```
<MD.Init.Velocity
  1    3000.000  0.0  0.0
  2   -3000.000  0.0  0.0
MD.Init.Velocity>
```

これは2つの原子からなる系の例です。 $N$ 個の原子がある場合、同様に $N$ 行の記述が必要です。第1列は「Atoms.SpeciesAndCoordinates」キーワードで指定した原子の連番と同じです。第2、3、4列は、各原子の速度の $x$ 、 $y$ 、 $z$ 成分で、単位はm/sです。「MD.Init.Velocity」キーワードは、「MD.Fixed.XYZ」キーワードと併用できます。

## バンド分散

### Band.dispersion

バンド分散を評価する場合は「Band.dispersion」キーワードを「ON」に設定して下さい。

### Band.KPath.UnitCell

「Band.KPath.UnitCell」キーワードによって、下記の例のようにバンド分散の計算で使用する単位胞ベクトルを指定します。

```
<Band.KPath.UnitCell
 3.56 0.0 0.0
 0.0 3.56 0.0
 0.0 0.0 3.56
Band.KPath.UnitCell>
```

記述は「<Band.KPath.UnitCell」で開始し、「Band.KPath.UnitCell>」で終了します。

「Band.KPath.UnitCell」キーワードの記述がある場合、同キーワードで指定した単位胞ベクトルからバンド分散の計算に使用する逆格子ベクトルを算出します。「Band.KPath.UnitCell」を指定しない場合には、「Atoms.UnitVectors」キーワードで指定した単位胞ベクトルから計算した逆格子ベクトルが使用されます。面心立方格子構造（fcc）や体心立方格子構造（bcc）の場合には「Band.KPath.UnitCell」キーワードで逆格子ベクトルを再定義した方が、特殊 $k$ 点の指定が容易です。

### Band.Nkpath

バンド分散の計算における経路の数を「Band.Nkpath」で指定します。

### Band.kpath

バンド分散の経路を「Band.kpath」キーワードで下記のように指定します。

```
<Band.kpath
 15  0.0 0.0 0.0  1.0 0.0 0.0  g X
 15  1.0 0.0 0.0  1.0 0.5 0.0  X W
 15  1.0 0.5 0.0  0.5 0.5 0.5  W L
 15  0.5 0.5 0.5  0.0 0.0 0.0  L g
 15  0.0 0.0 0.0  1.0 0.0 0.0  g X
Band.kpath>
```

記述は「<Band.kpath」で始まり、「Band.kpath>」で終わります。行数は「Band.Nkpath」で指定した数と同じでなくてはなりません。第1列には経路上で固有値を計算する格子の数を指定します。続く

(n1, n2, n3) および (n1', n2', n3') は、逆格子ベクトルを基底とする第 1 ブリルアンゾーンの経路の開始および終了の k 点を指定します。「Band.KPath.UnitCell」キーワードを指定している場合、バンド分散を計算するための逆格子ベクトルは「Band.KPath.UnitCell」で設定した単位胞ベクトルを用いて計算されます。「Band.KPath.UnitCell」の定義が無い場合は「Atoms.UnitVectors」で指定した単位胞ベクトルを用いて逆格子ベクトルを計算します。最後の 2 つの英文字は経路の開始および終了の k 点の名前を指定します。

## 再スタート

### scf.restart

過去の計算で生成されたリスタートファイル「*System.Name\_rst/\**」を用いて SCF 計算を再スタートさせる場合には、このキーワード「scf.restart」を「ON」に設定して下さい。

## 分子軌道 (MO) の出力

### MO.fileout

分子軌道をファイルに出力したい場合は、キーワード「MO.fileout」を「ON」に設定して下さい。

### num.HOMOs

出力する最高被占分子軌道 (HOMO) の数を「num.HOMOs」キーワードで指定します。

### num.LUMOs

出力する最低空分子軌道 (LUMOs) の数を「num.LUMOs」キーワードで指定します。

### MO.Nkpoint

「MO.fileout=ON」および「scf.EigenvalueSolver=Band」を設定している場合、「MO.Nkpoint」キーワードで MO を出力する k 点の数を指定します。

### MO.kpoint

MO を出力する k 点を「MO.kpoint」キーワードで以下のように指定します。

```
<MO.kpoint
  0.0  0.0  0.0
MO.kpoint>
```

記述は「<MO.kpoint」で開始し、「MO.kpoint>」で終わります。k 点は逆格子ベクトルを基底して (n1, n2, n3) の 3 つの数字で指定します。ここで、逆格子ベクトルは「Band.kpath」の場合と同様な方法で生成されたものが使用されます。

## DOS と PDOS

### Dos.fileout

全状態密度 (DOS) および射影した部分状態密度 (PDOS) を評価する場合は「Dos.fileout」キーワー

ドを「ON」に設定して下さい。

#### Dos.Erange

DOS 計算におけるエネルギー範囲を「Dos.Erange」キーワードで下記のように指定します。

```
Dos.Erange          -10.0  10.0
```

ここで2つの値はそれぞれエネルギー範囲の下限と上限 (eV) です。

#### Dos.Kgrid

DOS 計算を行う上で、第1ブリリアンゾーンを離散化するために (n1, n2, n3) の格子点を「Dos.Kgrid」キーワードで指定します。

## 開発者向けインタフェース

#### HS.fileout

後処理計算において、Kohn-Sham ハミルトニアン行列、重なり行列、密度行列を利用する場合には「HS.fileout」キーワードを「ON」に設定して下さい。これによりバイナリ形式のデータが「*System.Name.scfout*」に格納されます。ここで、「*System.Name*」はキーワード「System.Name」で設定したファイル名です。このデータの利用に関しては「開発者向けインタフェース」の章で詳しく説明します。

## Voronoi 電荷

#### Voronoi.charge

Voronoi 電荷を計算する場合には「Voronoi.charge」キーワードを「ON」に設定して下さい。計算結果は「*System.Name.out*」ファイルに保存されます。ここで「*System.Name*」は「System.Name」で設定したファイル名です。

## 8 出力ファイル

「level.of.fileout=0」と設定した場合、以下のファイルが生成されます。ここで、「*System.Name*」はキーワード「System.Name」で設定されたファイル名です。

- *System.Name.out*  
SCF 計算の履歴、構造最適化の履歴、Mulliken 電荷、全エネルギー、および双極子モーメントが保存されます。
- *System.Name.xyz*  
MD または構造最適化により得られた最終的な幾何学的構造が保存されます。このファイルは OpenMX Viewer [152, 151] や XCrySDen [105] で可視化できます。
- *System.Name.bulk.xyz*  
「scf.EigenvalueSolver=Band」の場合、コピーされたセルの原子を含む原子座標が出力されます。このファイルは OpenMX Viewer [152, 151] や XCrySDen [105] で可視化できます。
- *System.Name\_rst/*  
再スタートを行うための一連のファイルを保管しているディレクトリです。
- *System.Name.md*  
各 MD ステップごとの原子座標が保存されます。このファイルは OpenMX Viewer [152, 151] で可視化できます。
- *System.Name.md2*  
最終 MD ステップにおける原子座標が保存されます。指定した原子種記号を用いて原子が指定されています。
- *System.Name.cif*  
cif 形式の初期原子座標が保存されます。
- *System.Name.ene*  
MD ステップごとの計算値が保存されます。保存されている各数値の内容は「iterout\_md.c」ルーチン中で確認できます。

「level.of.fileout=1」と指定した場合、「level.of.fileout=0」で生成されるファイルに加え、以下の Gaussian cube 形式のファイルが生成されます。ここで、「*System.Name*」はキーワード「System.Name」により指定されたファイル名です。

- *System.Name.tden.cube*  
Gaussian cube 形式で全電子密度が保存されています。
- *System.Name.sden.cube*  
「LSDA-CA」, 「LSDA-PW」, 「GGA-PBE」を使用したスピン分極計算が実行された場合に、スピン電子密度が Gaussian cube 形式で保存されています。

- *System.Name.dden.cube*

構成原子の孤立原子状態の電子密度の重ね合わせを基準として計算された差電子密度が Gaussian cube 形式で保存されています。

- *System.Name.v0.cube*

アップスピンの非局所ポテンシャルを除く Kohn-Sham ポテンシャルが Gaussian cube 形式で保存されています。「scf.ProExpn.VNA」で射影展開法がオンの場合は中性原子ポテンシャル VNA も除かれます。[http://www.openmx-square.org/tech\\_notes/tech\\_notes.html](http://www.openmx-square.org/tech_notes/tech_notes.html) にある技術ノートの「全エネルギーと力」も参照ください。

- *System.Name.v1.cube*

スピン分極計算における、ダウンスピンの非局所ポテンシャルを除く Kohn-Sham ポテンシャルが Gaussian cube 形式で保存されています。「scf.ProExpn.VNA」で射影展開法がオンの場合は中性原子ポテンシャル VNA も除かれます。[http://www.openmx-square.org/tech\\_notes/tech\\_notes.html](http://www.openmx-square.org/tech_notes/tech_notes.html) にある技術ノートの「全エネルギーと力」も参照ください。

- *System.Name.vhart.cube*

差電子密度から生じる差電子 Hartree ポテンシャルが Gaussian cube 形式で保存されています。[http://www.openmx-square.org/tech\\_notes/tech\\_notes.html](http://www.openmx-square.org/tech_notes/tech_notes.html) にある技術ノートの「全エネルギーと力」も参照ください。

「level.of.fileout=2」の場合、「level.of.fileout=1」の場合に生成されるファイルに加え、以下のファイルが生成されます。ここで、「*System.Name*」はキーワード「System.Name」で指定されたファイル名です。

- *System.Name.vxc0.cube*

アップスピンの交換相関ポテンシャルが Gaussian cube 形式で保存されています。

- *System.Name.vxc1.cube*

ダウンスピンの交換相関ポテンシャルが Gaussian cube 形式で保存されています。

- *System.Name.grid*

数値積分およびポアソン方程式の解に使用される実空間格子が保存されています。

「MO.fileout=ON」および「scf.EigenvalueSolver=Cluster」の場合、以下のファイルも生成されます。

- *System.Name.homo0\_0.cube*, *System.Name.homo0\_1.cube*, ...

Gaussian cube 形式で最高占有軌道 (HOMO) から順番に占有軌道を出力したファイルです。「homo」に続く最初の数字はスピン状態を意味します (up=0, down=1)。また 2 番目の数字は固有状態を表し、0、1、2 はそれぞれ HOMO、HOMO-1、HOMO-2 に対応します。

- *System.Name.lumo0\_0.cube*, *System.Name.lumo0\_1.cube*, ...

Gaussian cube 形式で最低非占有軌道 (LUMO) から順番に非占有軌道を出力したファイルです。「lumo」に続く最初の数字はスピン状態を意味します (up=0, down=1)。また 2 番目の数字は固有状態を表し、0、1、2 はそれぞれ LUMO、LUMO+1、LUMO+2 に対応します。

「MO.fileout=ON」および「scf.EigenvalueSolver=Band」の場合、以下のファイルが生成されます。

- *System.Name.homo0\_0\_0\_r.cube*, *System.Name.homo1\_0\_1\_r.cube*, ... *System.Name.homo0\_0\_0\_i.cube*,  
*System.Name.homo1\_0\_1\_i.cube*, ...

Gaussian cube形式で最高占有軌道 (HOMO) から順番に占有軌道を出力したファイルです。「homo」に続く最初の数字は、キーワード「MO.kpoint」により指定される k 点の番号を意味します。また 2 番目の数字はスピン状態です (up=0, down=1)。3 番目の数字は固有状態を表し、0、1、2 はそれぞれ HOMO、HOMO-1、HOMO-2 に対応します。「r」および「i」は波動関数の実数および虚数部を意味します。

- *System.Name.lumo0\_0\_0\_r.cube*, *System.Name.lumo1\_0\_1\_r.cube*, ... *System.Name.lumo0\_0\_0\_i.cube*,  
*System.Name.lumo1\_0\_1\_i.cube*, ...

Gaussian cube形式で最低非占有軌道 (LUMO) から順番に非占有軌道を出力したファイルです。「lumo」に続く最初の数字は、キーワード「MO.kpoint」により指定される k 点の番号を意味します。また 2 番目の数字はスピン状態です (up=0, down=1)。3 番目の数字は固有状態を表し、0、1、2 はそれぞれ LUMO、LUMO+1、LUMO+2 に対応します。「r」および「i」は波動関数の実数および虚数部を意味します。

「Band.Nkpath」が 0 以上で、かつ「scf.EigenvalueSolver=Band」の場合、以下のファイルも生成されます。

- *System.Name.Band*

バンド分散のデータファイルが保存されています。

「Dos.fileout=ON」の時、以下のファイルも生成されます。

- *System.Name.Dos.val*

状態密度を計算するための固有値のデータファイル。

- *System.Name.Dos.vec*

状態密度を計算するための固有ベクトルのデータファイル。

「scf.SpinPolarization=NC」および「level.of.fileout=1 もしくは 2」の場合、以下のファイルも生成されます。

- *System.Name.nco.xsf*

Mulliken 解析により各原子に射影されたノンコリニア軌道磁気モーメントが保存されたベクトルファイル。XCrySDen の「Display→Forces」を用いて可視化が可能です。

- *System.Name.nc.xsf*

Mulliken 解析により各原子に射影されたノンコリニアスピン磁気モーメントが保存されたベクトルファイル。XCrySDen の「Display→Forces」を用いて可視化が可能です。

- *System.Name.ncsden.xsf*

実空間格子上のノンコリニアスピン磁気モーメントが保存されたベクトルファイル。XCrySDen の「Display→Forces」を用いて可視化が可能です。

## 9 汎関数

OpenMX において、交換相関汎関数は局所密度近似 (LDA, LSDA) [2, 3, 4] もしくは一般化勾配近似 (GGA) [5] と呼ばれる近似手法によって取り扱われます。キーワード「scf.XcType」で、交換相関汎関数に対する近似の一つが選択できます。

```
scf.XcType          LDA          # LDA|LSDA-CA|LSDA-PW|GGA-PBE
```

現在、「LDA」、「LSDA-CA」、「LSDA-PW」、「GGA-PBE」が使用可能です。ここで、「LDA」と「LSDA-CA」は Ceperley-Alder の局所密度汎関数とそのスピン密度版であり [2]、「LSDA-PW」は Perdew-Wang による GGA の表現形式で密度勾配がゼロに設定された局所スピン密度汎関数です [4]。「GGA-PBE」は Perdew、Burke、Ernzerhof により提案された一般化勾配近似 (GGA) です [5]。この GGA は実空間での一次有限差分法を用いて実装されています。さらに、LDA+ $U$  (または GGA+ $U$ ) 汎関数も使用可能です。詳細は「LDA+ $U$ 」の章を参照して下さい。スピン (非) 分極およびノンコリニア計算を指定する関連キーワードは「scf.SpinPolarization」です。

```
scf.SpinPolarization  off          # On|Off|NC
```

スピン分極計算を実行する場合、「ON」に設定して下さい。非スピン分極計算を実行する場合には「OFF」に設定して下さい。キーワード「scf.XcType」に対して「LDA」を使用する場合には、キーワード「scf.SpinPolarization」は OFF にして下さい。これらのオプションの他に、ノンコリニア DFT 計算を実行する場合は「NC」に設定して下さい。この計算に関しては「ノンコリニア DFT」の章も参照して下さい。

## 10 基底関数

### 10.1 概要

OpenMX は一粒子 Kohn-Sham 波動関数を展開するために数値擬原子軌道 (PAOs) $\chi$  を基底関数として使用します。PAO 関数は動径関数  $R$  と実球面調和関数  $Y$  との積として次式で与えられます。

$$\chi(\mathbf{r}) = R(r)Y(\hat{\mathbf{r}}),$$

この動径関数  $R$  は数値の表として与えられ、実空間のカットオフ半径内で有限な関数です。つまり関数  $R$  は所定のカットオフ半径を超えるとゼロになります。ADPACK により計算された PAO 関数はプリミティブ関数と呼ばれます。このプリミティブ PAO 関数を出発点として、OpenMX の軌道最適化法を用いて最適化された PAO 関数を得ることが可能です [41]。これらの基底関数に関するデータはファイル拡張子「pao」を持ったファイルに保存されています。OpenMX の計算が実行される際には、これらのファイルに保存されている数値データが読み込まれています。任意の動径距離  $r$  に対する PAO 関数の値は補間法により得られます。ファイル拡張子「pao」を持ったファイルは、例えば「DFT\_DATA19/PAO」などのディレクトリに保存して下さい。このディレクトリは以下のキーワードにより指定します。ただしディレクトリ名中に「PAO」を含める必要はありません。

```
DATA.PATH ../DFT_DATA19 # default=../DFT_DATA19
```

絶対および相対パスの指定が可能であり、デフォルトは「../DFT\_DATA19」となっています。OpenMX の入力ファイルにおいて、基底関数は以下のように「Definition.of.Atomic.Species」キーワードにより指定されます。

```
<Definition.of.Atomic.Species
  H  H5.0-s2p1      H_PBE19
  C  C5.0-s2p1      C_PBE19
Definition.of.Atomic.Species>
```

ここで基底関数の略語記号、H5.0-s2p1 を導入します。H5.0 は PAO 関数の拡張子なしのファイル名を表します。このファイルはキーワード「DATA.PATH」により指定された「DFT\_DATA19/PAO」などのディレクトリに存在しなければなりません。5.0 は PAO 関数のカットオフ半径を意味します。また s2p1 は指定されたファイル中での二つの動径関数と一つの動径関数が  $s$ -軌道、 $p$ -軌道にそれぞれ割り当てられることを意味します。この場合、合計 5 つの PAO 基底関数 ( $2 \times 1 + 1 \times 3 = 5$ ) が「H」に対して割り当てられます。

OpenMX の Web サイト (<http://www.openmx-square.org/>) では最適化基底関数のデータベース Ver. 2019 が無償公開されています。従って一般のユーザーはこれらの最適化された基底関数を使用することが推奨されます。しかし上級者のために、プリミティブおよび最適化 PAO 関数について、続く節で説明します。

### 10.2 プリミティブ基底関数

閉じ込め擬ポテンシャル下で擬原子の基底状態および励起状態を計算することで、プリミティブ動径関数が生成されます [41]。この計算は ADPACK を用いて実行可能です。一例として炭素擬原子の  $s$ -軌道

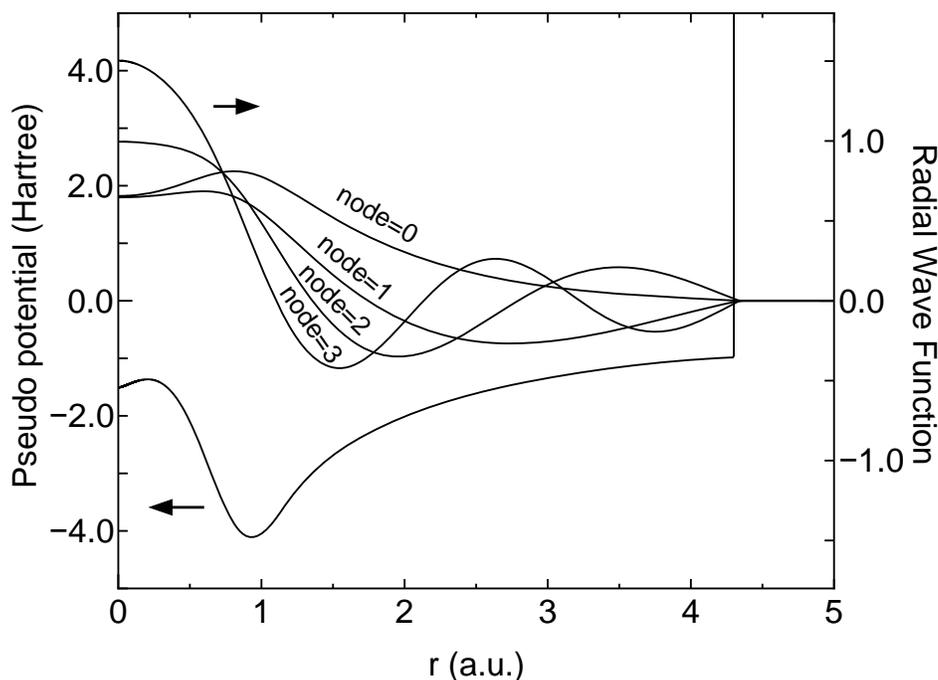


Figure 1: 閉じ込め擬ポテンシャル下で生成された炭素擬原子の  $s$ -軌道のプリミティブ動径関数。

のプリミティブ動径関数を図 1 に示します。これらの関数は数値の表として、ファイル拡張子「pao」を持つファイルに保存されています。図 1 から、基底状態はノードを持たず、第一励起状態は一つのノードを持っていることがわかります。より高い励起状態ではノード数は一つずつ増加しています。プリミティブ動径関数と実球面調和関数の積からプリミティブ基底関数が構成されます。さらに、このプリミティブ基底関数の線型結合によって一粒子 Kohn-Sham 波動関数が記述されます。計算の精度と効率は二つのパラメータ: カットオフ半径と基底関数の数により制御されます。一般的に図 2 に示すように、カットオフ半径と基底関数の数を増加させることで、収束解が得られます。しかしながら、大きなカットオフ半径を持ち、かつ多数の基底関数を使用する場合には、メモリー量や計算時間などの多大な計算リソースが必要となることに注意が必要です。各元素毎に適切なカットオフ半径と基底関数の数を決定する一般的な指針に関しては、参考文献 [41] で議論されています。この議論によれば、周期表の右側にある F や Cl などの元素に対して、十分な収束性を達成するためには高い角運動量をもつ基底関数が必要であり、周期表の左側にある Li や Na などの元素に対しては大きなカットオフ半径を使用すべきであるということがわかります。OpenMX の Web サイト (<http://www.openmx-square.org/>) では最適化基底関数のデータベース Ver. 2019 が無償公開されています。従って一般のユーザーはこれらの最適化された基底関数を使用することが推奨されます。

### 10.3 データベース Ver. 2019 により提供される最適化基底関数

最適化 PAO 関数はデータベース Ver. 2019 として、OpenMX の web サイト (<http://www.openmx-square.org/>) で一般公開されています。公開されている基底関数は軌道最適化法により生成され [41]、また一連のベンチマーク計算により十分にテストされていることから、一般ユーザーの最初の選択肢として、このデータベースの使用を推奨します。データベース Ver. 2019 内のほとんどの元素に対して、化学環境のトレーニングセットとして 3 種の系が選択され、選択された系に対して軌道最適化法により

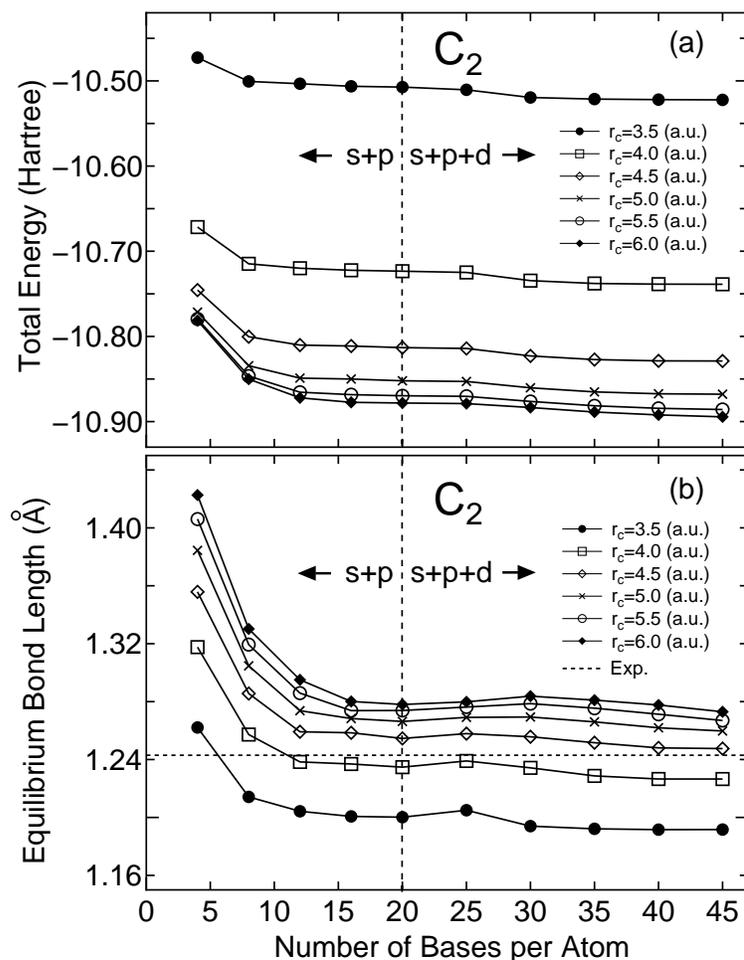


Figure 2: 炭素二量体の (a) 全エネルギーおよび (b) 平衡結合距離のカットオフ半径・基底関数の数に対する収束特性。

PAO 関数が変分原理に基づき最適化されています [41]。これらの最適化基底関数のセットは部分空間回転とグラム・シュミット直交化法の組み合わせ手法を用いて統合されており、一つの PAO ファイルに保存されています。最適化 PAO 関数はいくつかの異なる化学環境セットに対して最適化されているため、最適 PAO 関数の可搬性は高いと考えられます。実際、データベースのウェブサイト上にある一連のベンチマーク計算では、対応する全電子計算の結果とよく一致しています。これらベンチマーク計算を参考に、各元素に対しての適切な基底関数の選択 (カットオフ半径と基底関数の数) が可能でしょう。ベンチマーク計算に使われた入力ファイルもウェブサイトから入手することができますので、それらは OpenMX の計算に慣れるために有用だと思われます。

データベース Ver. 2019 の精度はデルタ因子により検証されています [40]。71 元素の平均デルタ因子は 1.774 meV/atom、標準偏差は 1.702 meV/atom となっており、これはデータベース Ver. 2019 の高い精度を示しています。従って一般のユーザーはこれらの最適化された基底関数を使用することが推奨されます。デルタ因子の計算に関しては、「格子定数 vs. エネルギー」の章も参照して下さい。

利用者の利便性のため、表 1 と 2 に基底関数の適切な選択方法が提供されています。この表には各擬ポテンシャルについて、三つの選択肢 *Quick*、*Standard* と *Precise* が用意されています。*Quick* は高速な計算ですが妥当な精度の計算が可能です。*Standard* はほとんどのユーザーに適切な選択肢となり、精度

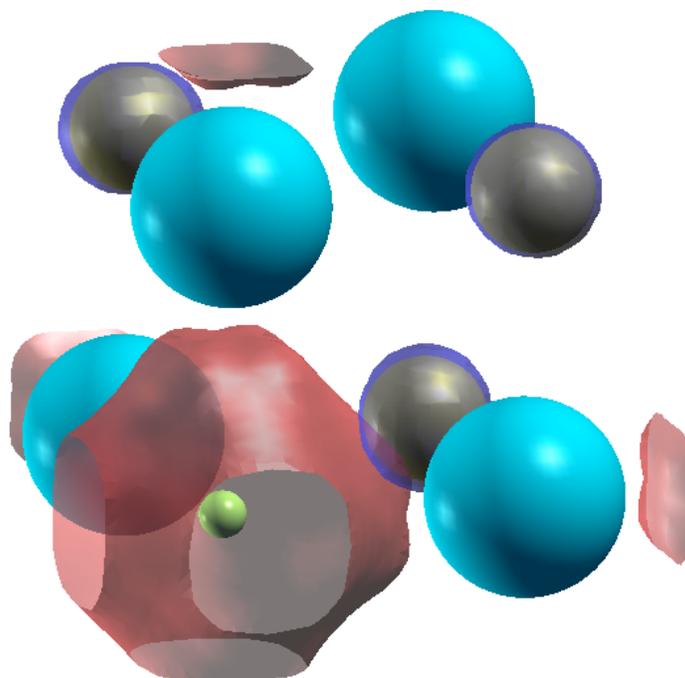


Figure 3: Cl サイト空孔を持った NaCl の最高占有状態の等値面 (F 点)。この等値面は Cl 空孔をもつ NaCl の F 中心を示している。等値面は XCrySDen を使用して、0.042 の等値 (isovalue) で描画 [105]。系の電荷を-1 として、キーワード「scf.system.charge」を使用して計算。水色および銀色はそれぞれナトリウムおよび塩素原子を表し、黄緑色の小丸は空原子の位置を示す。

と効率がバランスします。Precise はほとんどの場合で収束する結果に到達します。この表は基底関数の選択の指標を与えますが、当然ながら、基底関数は目的に応じて適切に選択されるべきです。

#### 10.4 ユーザーによる PAO の最適化

変分最適化法を用いれば、ユーザー自身で変分原理に基づき動径関数の形状を最適化することができます。詳細は「軌道の最適化」の章を参照して下さい。

#### 10.5 空原子の配置

プリミティブ関数および最適化 PAO 関数は通常、原子に割り当てられます。さらに、空原子を使って空孔領域に基底関数を割り当てることも可能です。データベースのウェブサイト (<http://www.openmx-square.org/>) には空原子「E」の項目があります。空原子「E」の擬ポテンシャルはフラットなゼロポテンシャルですが、この空原子「E」の擬ポテンシャルを使用することで、原子位置とは独立に、空間上の任意の場所に基底関数を配置することができます。それを行うために、空原子を以下のように定義します。

```
<Definition.of.Atomic.Species
H      H5.0-s2p1      H_PBE19
```

```

C      C5.0-s2p1      C_PBE19
EH     H5.0-s2p1      E
EC     C5.0-s2p1      E
Definition.of.Atomic.Species>

```

この例では、2種の空原子が「EH」および「EC」として定義されており、それぞれは「H5.0-s2p1」および「C5.0-s2p1」で指定される基底関数が割当てられています。従って空原子に対して、任意の基底関数が使用可能であり、また空原子の定義も空原子に対する擬ポテンシャルとして「E.vps」を使用するだけで、簡単に行うことができます。そして「EH」および「EC」はキーワード「Atoms.SpeciesAndCoordinates」によって、任意の場所に配置することができます。ここでは空原子に対する擬ポテンシャルはゼロであり、また空原子から系に供給される電子数もゼロとなります。空原子の配置により、counterpoise法 (CP) 法 [46, 47] を使って基底関数重なり誤差を推測できるだけでなく、線形結合擬原子軌道 (LCPAO) 法の範囲内で空孔状態や金属表面上の自由電子の状態を取り扱うこともできます。例として、Cl 空孔をもつ NaCl の F 中心の計算を図 3 に示します。Γ 点での最高占有状態がこの F 中心の状態であることがわかります。この計算は「work」ディレクトリ内の「NaClFC.dat」を使用して再現できます。

## 10.6 PAO および VPS ファイルを保存するディレクトリの指定

PAO および VPS ディレクトリへのパスは、以下のキーワードを使用して入力ファイル内で指定することができます。

```
DATA.PATH    ../DFT_DATA19    # default=../DFT_DATA19
```

絶対および相対パスの指定が可能です。基底関数と擬ポテンシャルのデータベースは複数のバージョンが公開されておりますが、いくつかの元素に対しては擬ポテンシャル中のセミコアの状態数がバージョン間で同一ではありません。従って、これらのデータを使用する際には、同一バージョンの PAO および VPS を使用することが推奨されます。こうした理由から、PAO および VPS ファイルの各バージョンは異なったディレクトリに保存するのが良いでしょう。その際に、このキーワードが有用になります。

Table 1: E から Kr までの PAO 基底関数の推奨される選択肢

| VPS       | Valence electrons | Quick         | Standard       | Precise         |
|-----------|-------------------|---------------|----------------|-----------------|
| E         | 0.0               | Kr10.0-s1p1   | Kr10.0-s2p1d1  | Kr10.0-s2p2d1f1 |
| H_PBE19   | 1.0               | H5.0-s2       | H6.0-s2p1      | H7.0-s2p2d1     |
| He_PBE19  | 2.0               | He8.0-s1p1    | He8.0-s2p1     | He10.0-s2p2d1   |
| Li_PBE19  | 3.0               | Li8.0-s3p1    | Li8.0-s3p2     | Li8.0-s3p2d1    |
| Be_PBE19  | 2.0               | Be7.0-s2p1    | Be7.0-s2p2     | Be7.0-s3p2d1    |
| B_PBE19   | 3.0               | B7.0-s2p2     | B7.0-s2p2d1    | B7.0-s3p2d2     |
| C_PBE19   | 4.0               | C6.0-s2p2     | C6.0-s2p2d1    | C6.0-s3p2d2     |
| N_PBE19   | 5.0               | N6.0-s2p2     | N6.0-s2p2d1    | N6.0-s3p2d2     |
| O_PBE19   | 6.0               | O6.0-s2p2     | O6.0-s2p2d1    | O6.0-s3p2d2     |
| F_PBE19   | 7.0               | F6.0-s2p2     | F6.0-s2p2d1    | F6.0-s3p3d2f1   |
| Ne_PBE19  | 8.0               | Ne9.0-s2p2    | Ne9.0-s2p2d1   | Ne9.0-s3p2d2    |
| Na_PBE19  | 9.0               | Na9.0-s3p2    | Na9.0-s3p2d1   | Na9.0-s3p2d2    |
| Mg_PBE19  | 8.0               | Mg9.0-s2p2    | Mg9.0-s3p2d1   | Mg9.0-s3p2d2    |
| Al_PBE19  | 3.0               | Al7.0-s2p1d1  | Al7.0-s2p2d1   | Al7.0-s3p2d2    |
| Si_PBE19  | 4.0               | Si7.0-s2p1d1  | Si7.0-s2p2d1   | Si7.0-s3p3d2    |
| P_PBE19   | 5.0               | P7.0-s2p2d1   | P7.0-s2p2d1f1  | P7.0-s3p2d2f1   |
| S_PBE19   | 6.0               | S7.0-s2p2d1   | S7.0-s2p2d1f1  | S7.0-s3p2d2f1   |
| Cl_PBE19  | 7.0               | Cl7.0-s2p2d1  | Cl7.0-s2p2d1f1 | Cl7.0-s3p2d2f1  |
| Ar_PBE19  | 8.0               | Ar9.0-s2p2d1  | Ar9.0-s2p2d1f1 | Ar9.0-s3p2d2f1  |
| K_PBE19   | 9.0               | K10.0-s3p2    | K10.0-s3p2d1   | K10.0-s3p2d2    |
| Ca_PBE19  | 10.0              | Ca9.0-s3p2    | Ca9.0-s3p2d1   | Ca9.0-s3p2d2    |
| Sc_PBE19  | 11.0              | Sc9.0-s2p2d1  | Sc9.0-s3p2d1   | Sc9.0-s3p2d2    |
| Ti_PBE19  | 12.0              | Ti7.0-s2p2d1  | Ti7.0-s3p2d1   | Ti7.0-s3p2d2f1  |
| V_PBE19   | 13.0              | V6.0-s2p2d1   | V6.0-s3p2d1    | V6.0-s3p2d2f1   |
| Cr_PBE19  | 14.0              | Cr6.0-s2p2d1  | Cr6.0-s3p2d1   | Cr6.0-s3p2d2f1  |
| Mn_PBE19  | 15.0              | Mn6.0-s2p2d1  | Mn6.0-s3p2d1   | Mn6.0-s3p2d2f1  |
| Fe_PBE19H | 16.0              | Fe5.5H-s2p2d1 | Fe5.5H-s3p2d1  | Fe5.5H-s3p2d2f1 |
| Fe_PBE19S | 14.0              | Fe6.0S-s2p2d1 | Fe6.0S-s3p2d1  | Fe6.0S-s3p2d2f1 |
| Co_PBE19H | 17.0              | Co6.0H-s2p2d1 | Co6.0H-s3p2d1  | Co6.0H-s3p2d2f1 |
| Co_PBE19S | 15.0              | Co6.0S-s2p2d1 | Co6.0S-s3p2d1  | Co6.0S-s3p2d2f1 |
| Ni_PBE19H | 18.0              | Ni6.0H-s2p2d1 | Ni6.0H-s3p2d1  | Ni6.0H-s3p2d2f1 |
| Ni_PBE19S | 16.0              | Ni6.0S-s2p2d1 | Ni6.0S-s3p2d1  | Ni6.0S-s3p2d2f1 |
| Cu_PBE19H | 19.0              | Cu6.0H-s2p2d1 | Cu6.0H-s3p2d1  | Cu6.0H-s3p2d2f1 |
| Cu_PBE19S | 11.0              | Cu6.0S-s2p1d1 | Cu6.0S-s3p2d1  | Cu6.0S-s3p2d2f1 |
| Zn_PBE19H | 20.0              | Zn6.0H-s2p2d1 | Zn6.0H-s3p2d1  | Zn6.0H-s3p2d2f1 |
| Zn_PBE19S | 12.0              | Zn6.0S-s2p1d1 | Zn6.0S-s3p2d1  | Zn6.0S-s3p2d2f1 |
| Ga_PBE19  | 13.0              | Ga7.0-s2p2d1  | Ga7.0-s3p2d2   | Ga7.0-s3p2d2f1  |
| Ge_PBE19  | 4.0               | Ge7.0-s2p1d1  | Ge7.0-s3p2d2   | Ge7.0-s3p2d2f1  |
| As_PBE19  | 15.0              | As7.0-s3p2d1  | As7.0-s3p2d2   | As7.0-s3p2d2f1  |
| Se_PBE19  | 6.0               | Se7.0-s3p2d1  | Se7.0-s3p2d2   | Se7.0-s3p2d2f1  |
| Br_PBE19  | 7.0               | Br7.0-s3p2d1  | Br7.0-s3p2d2   | Br7.0-s3p2d2f1  |
| Kr_PBE19  | 8.0               | Kr10.0-s2p2d1 | Kr10.0-s3p2d2  | Kr10.0-s3p2d2f1 |

Table 2: Rb から Bi までの PAO 基底関数の推奨される選択肢

| VPS      | Valence electrons | Quick          | Standard       | Precise         |
|----------|-------------------|----------------|----------------|-----------------|
| Rb_PBE19 | 9.0               | Rb11.0-s2p2d1  | Rb11.0-s3p2d2  | Rb11.0-s3p2d2f1 |
| Sr_PBE19 | 10.0              | Sr10.0-s2p2d1  | Sr10.0-s3p2d2  | Sr10.0-s3p3d2f1 |
| Y_PBE19  | 11.0              | Y10.0-s3p2d1   | Y10.0-s3p2d2   | Y10.0-s3p3d2f1  |
| Zr_PBE19 | 12.0              | Zr7.0-s3p2d1   | Zr7.0-s3p2d2   | Zr7.0-s3p2d2f1  |
| Nb_PBE19 | 13.0              | Nb7.0-s3p2d1   | Nb7.0-s3p2d2   | Nb7.0-s3p2d2f1  |
| Mo_PBE19 | 14.0              | Mo7.0-s3p2d1   | Mo7.0-s3p2d2   | Mo7.0-s3p2d2f1  |
| Tc_PBE19 | 15.0              | Tc7.0-s3p2d1   | Tc7.0-s3p2d2   | Tc7.0-s3p2d2f1  |
| Ru_PBE19 | 14.0              | Ru7.0-s3p2d1   | Ru7.0-s3p2d2   | Ru7.0-s3p2d2f1  |
| Rh_PBE19 | 15.0              | Rh7.0-s3p2d1   | Rh7.0-s3p2d2   | Rh7.0-s3p2d2f1  |
| Pd_PBE19 | 16.0              | Pd7.0-s3p2d1   | Pd7.0-s3p2d2   | Pd7.0-s3p2d2f1  |
| Ag_PBE19 | 17.0              | Ag7.0-s3p2d1   | Ag7.0-s3p2d2   | Ag7.0-s3p2d2f1  |
| Cd_PBE19 | 12.0              | Cd7.0-s3p2d1   | Cd7.0-s3p2d2   | Cd7.0-s3p2d2f1  |
| In_PBE19 | 13.0              | In7.0-s3p2d1   | In7.0-s3p2d2   | In7.0-s3p2d2f1  |
| Sn_PBE19 | 14.0              | Sn7.0-s3p2d1   | Sn7.0-s3p2d2   | Sn7.0-s3p2d2f1  |
| Sb_PBE19 | 15.0              | Sb7.0-s3p2d1   | Sb7.0-s3p2d2   | Sb7.0-s3p2d2f1  |
| Te_PBE19 | 16.0              | Te7.0-s3p2d2   | Te7.0-s3p2d2f1 | Te7.0-s3p3d2f1  |
| I_PBE19  | 7.0               | I7.0-s3p2d2    | I7.0-s3p2d2f1  | I7.0-s3p3d2f1   |
| Xe_PBE19 | 8.0               | Xe11.0-s3p2d1  | Xe11.0-s3p2d2  | Xe11.0-s3p2d2f1 |
| Cs_PBE19 | 9.0               | Cs12.0-s3p2d1  | Cs12.0-s3p2d2  | Cs12.0-s3p2d2f1 |
| Ba_PBE19 | 10.0              | Ba10.0-s3p2d1  | Ba10.0-s3p2d2  | Ba10.0-s3p2d2f1 |
| La_PBE19 | 11.0              | La8.0-s3p2d1f1 | La8.0-s3p2d2f1 | La8.0-s3p3d2f1  |
| Ce_PBE19 | 12.0              | Ce8.0-s3p2d1f1 | Ce8.0-s3p2d2f1 | Ce8.0-s3p3d2f1  |
| Pr_PBE19 | 13.0              | Pr8.0-s3p2d1f1 | Pr8.0-s3p2d2f1 | Pr8.0-s3p3d2f1  |
| Nd_PBE19 | 14.0              | Nd8.0-s3p2d1f1 | Nd8.0-s3p2d2f1 | Nd8.0-s3p3d2f1  |
| Pm_PBE19 | 15.0              | Pm8.0-s3p2d1f1 | Pm8.0-s3p2d2f1 | Pm8.0-s3p3d2f1  |
| Sm_PBE19 | 16.0              | Sm8.0-s3p2d1f1 | Sm8.0-s3p2d2f1 | Sm8.0-s3p3d2f1  |
| Dy_PBE19 | 20.0              | Dy8.0-s3p2d1f1 | Dy8.0-s3p2d2f1 | Dy8.0-s3p3d2f1  |
| Ho_PBE19 | 21.0              | Ho8.0-s3p2d1f1 | Ho8.0-s3p2d2f1 | Ho8.0-s3p3d2f1  |
| Lu_PBE19 | 11.0              | Lu8.0-s3p2d2   | Lu8.0-s3p2d2f1 | Lu8.0-s3p3d2f1  |
| Hf_PBE19 | 12.0              | Hf9.0-s3p2d2   | Hf9.0-s3p2d2f1 | Hf9.0-s3p3d2f1  |
| Ta_PBE19 | 13.0              | Ta7.0-s3p2d2   | Ta7.0-s3p2d2f1 | Ta7.0-s3p3d2f1  |
| W_PBE19  | 12.0              | W7.0-s3p2d2    | W7.0-s3p2d2f1  | W7.0-s3p3d2f1   |
| Re_PBE19 | 15.0              | Re7.0-s3p2d2   | Re7.0-s3p2d2f1 | Re7.0-s3p3d2f1  |
| Os_PBE19 | 14.0              | Os7.0-s3p2d2   | Os7.0-s3p2d2f1 | Os7.0-s3p3d2f1  |
| Ir_PBE19 | 15.0              | Ir7.0-s3p2d2   | Ir7.0-s3p2d2f1 | Ir7.0-s3p3d2f1  |
| Pt_PBE19 | 16.0              | Pt7.0-s3p2d2   | Pt7.0-s3p2d2f1 | Pt7.0-s3p3d2f1  |
| Au_PBE19 | 17.0              | Au7.0-s3p2d2   | Au7.0-s3p2d2f1 | Au7.0-s3p3d2f1  |
| Hg_PBE19 | 18.0              | Hg8.0-s3p2d2   | Hg8.0-s3p2d2f1 | Hg8.0-s3p3d2f1  |
| Tl_PBE19 | 19.0              | Tl8.0-s3p2d2   | Tl8.0-s3p2d2f1 | Tl8.0-s3p3d2f1  |
| Pb_PBE19 | 14.0              | Pb8.0-s3p2d2   | Pb8.0-s3p2d2f1 | Pb8.0-s3p3d2f1  |
| Bi_PBE19 | 15.0              | Bi8.0-s3p2d2   | Bi8.0-s3p2d2f1 | Bi8.0-s3p3d2f1  |

## 11 擬ポテンシャル

### 11.1 従来型擬ポテンシャル

OpenMX では、原子核のクーロンポテンシャルは、Morrison、Bylander、Kleinman により提案されたノルム保存擬ポテンシャル [36] で置き換えられます。これは Vanderbilt によるウルトラソフト擬ポテンシャル [37] のノルム保存型となっています。擬ポテンシャルは、ADPACK を使用して生成できますが、ユーザーの利便性のために、擬ポテンシャルのデータベース Ver. 2019 が OpenMX の web サイトで公開されています (ADPACK は原子の密度関数計算を行うためプログラムパッケージであり、OpenMX の web サイトで入手可能です)。データベースに保存されている擬ポテンシャルを使用する場合には、それらをディレクトリ「openmx3.9/DFT\_DATA19/VPS/」にコピーして下さい。ただし、ほとんどのデータは既に OpenMX Ver. 3.9. の配布パッケージ内にコピー済みです。これらのデータは自由に利用することが可能ですが、利用に伴う損害・不利益等については一切責任を負いかねますので、あらかじめご了承下さい。擬ポテンシャルの指定は、基底関数の指定と同様にキーワード「Definition.of.Atomic.Species」で行います。例えば次の様にして指定できます。

```
<Definition.of.Atomic.Species
  H   H6.0-s2p1      H_CA19
  C   C6.0-s2p2      C_CA19
Definition.of.Atomic.Species>
```

擬ポテンシャルのファイルは第 3 列で指定され、またそのファイルはディレクトリ「DFT\_DATA19/VPS」に存在する必要があります。さらに原子座標の指定の際、各原子のアップおよびダウンスピン状態に対する電子数の指定を以下のように行うことが必要です。

```
<Atoms.SpeciesAndCoordinates
  1  C      0.000000   0.000000   0.000000   2.0  2.0
  2  H     -0.889981  -0.629312   0.000000   0.5  0.5
  3  H      0.000000   0.629312  -0.889981   0.5  0.5
  4  H      0.000000   0.629312   0.889981   0.5  0.5
  5  H      0.889981  -0.629312   0.000000   0.5  0.5
Atoms.SpeciesAndCoordinates>
```

ここで第 6、7 列は各原子のそれぞれアップおよびダウンスピン状態に対する初期電子数を与えます。原子に対するアップおよびダウン電子数の合計は、擬ポテンシャルの生成時に考慮される価電子の数と等価でなければなりません。それぞれの擬ポテンシャルに含まれている荷電子数は擬ポテンシャルファイル「\*.vps」の中から見つけることができます。例えば、データベース Ver. 2019 内の炭素原子のファイル「C\_PBE19.vps」の中に次の行があります。

```
valence.electron      4.0
```

数字「4.0」は擬ポテンシャルの生成で考慮される電子数です。従って上記の例では、アップスピン電子数 (2.0) およびダウンスピン電子数 (2.0) の合計は「Atoms.SpeciesAndCoordinates」で指定する「C」に対して 4.0 となります。表 1 と 2 にはデータベース Ver. 2019 内で提供される擬ポテンシャルの価電子数も示されていますので、参考にして下さい。

ユーザー自身で ADPACK を使用して擬ポテンシャルを作成する場合には、以下の点に注意する必要があります。

- 非物理的なゴースト状態が生じていないかを確認すること。擬ポテンシャルの分離形式を使用しているため、ゴースト状態が生じる可能性があります。目的とする系の計算を行う前に、単純な系に対してベンチマーク計算を行い、擬ポテンシャルが適切かどうかを確認する必要があります。
- 部分内殻補正のために滑らかな内殻電子密度を作成すること。さもないと数値積分の積分グリッドに対して高いカットオフエネルギーが必要となり、計算コストが増大します。

さらなる詳細はプログラムパッケージ「ADPACK」のマニュアルを参照して下さい。ただし、高精度な擬ポテンシャルを作成するには、最初に考える以上に多くの経験が必要になることを明記しておきます。

## 11.2 オープンコア型擬ポテンシャル

ランタノイドの  $4f$  電子は多くの場合にスピン分極しており、多数 (majority) スピン状態がフェルミエネルギーより数 eV 低くなります。しかし、LDA や GGA ではこれらのバンド構造の特徴を再現することができません。 $4f$  電子に対してオンサイトクーロン反発を導入する plus  $U$  法を用いた計算によってこの問題を解決することも可能ですが、OpenMX ではより簡便な方法として  $4f$  電子のスピン分極を組み込んだオープンコア型擬ポテンシャルが利用可能です (ただし、2019 年版のデータベースでは一部の原子のみに対応しています)。ランタノイドに対するオープンコア型擬ポテンシャルでは、 $4f$  電子を内殻状態の一部として扱います。また部分内殻補正電荷は内殻状態として扱われている  $4f$  電子の電荷分布をよく再現するように生成されています。2019 年版データベースに収められている Nd\_CA13\_OC.vps と Nd\_PBE19\_OC.vps がこの方法で作成されたものです。

図 4 は、オープンコア擬ポテンシャルを用いた  $\text{CaCu}_5$  型構造の  $\text{NdCo}_5$  の計算結果を示しています。これを見ると、オープンコア擬ポテンシャルを用いることで、LDA+ $U$  法を用いた場合と定性的に同じ結果が得られていることが分かります。OpenMX でオープンコア擬ポテンシャルを用いる場合には、以

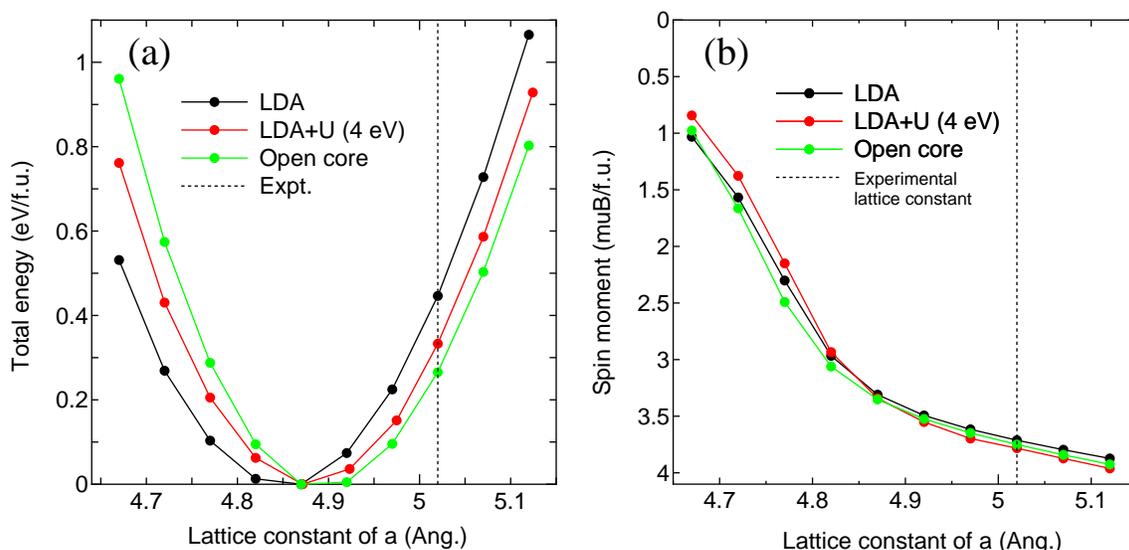


Figure 4: (a)  $\text{CaCu}_5$  構造を持つ  $\text{NdCo}_5$  の格子定数に対する全エネルギー変化。LDA 計算、 $4f$  状態に対して 4eV を使用した LDA+ $U$  計算、オープンコア擬ポテンシャルを用いた計算の三つの場合を比較。(b) (a) と同じ系におけるスピン磁気モーメント (化学式あたりの値を表示)。計算に用いた入力ファイルは、「work」ディレクトリの「NdCo5\_4f.dat」、「NdCo5\_4f+U.dat」、「NdCo5\_OC.dat」を参照のこと。

下のキーワードにより部分内殻補正電荷をスピン分極させることができます。

```
<scf.pcc.opencore
  Nd 1
  Co 0
scf.pcc.opencore>
```

これは NdCo<sub>5</sub> に対する例であり、1 列目は「Definition.of.Atomic.Species」キーワードにて定義した原子種名、2 列目はスピン方向を指定するフラグとなっています。部分内殻補正電荷は、*z* 軸に沿って「1」の場合には上向きに、「-1」の場合には下向きにスピン分極します。スピン分極させない場合には「0」を指定します。このキーワードを用いることで、サイトごとに 4*f* 電子の開殻状態のスピン方向を制御することが可能です。オープンコア擬ポテンシャルは、隣接原子間で 4*f* 状態と軌道が重なっていない場合に有効です。また化合物中での 4*f* 電子の占有状態が擬ポテンシャル生成時と大きく異なる場合には、オープンコア擬ポテンシャルの適用は正当化されません。

### 11.3 内殻準位励起のための擬ポテンシャル

内殻準位を含んだ完全な相対論的擬ポテンシャル (VPS) と擬原子軌道 (PAO) のデータベースがデータベース Ver 2019 として利用可能です。この擬ポテンシャルと擬原子軌道は内殻励起スペクトルの計算に対して使用されるものです。データベース Ver 2019 は以下のウェブサイトで利用できます。

[https://t-ozaki.issp.u-tokyo.ac.jp/vps\\_pao\\_core2019/](https://t-ozaki.issp.u-tokyo.ac.jp/vps_pao_core2019/)

B, C, N, O, Si, S, Ge 元素のデータが利用できます。固体や気体の内殻準位の絶対束縛エネルギーを計算するにはこれらの VPS と PAS のデータを使用しなければなりません。この計算で得られた絶対束縛エネルギーは X 線光電子分光 (XPS) で測定される束縛エネルギーと直接に比較できます。本計算に関しては内殻準位の絶対束縛エネルギー：XPS 内殻準位エネルギーの章も参照してください。

## 12 カットオフエネルギー：数値積分のための計算グリッドの設定

### 12.1 収束

計算精度と計算量は、数値積分およびポアソン方程式の解法で用いられるカットオフエネルギーにも依存します [42]。カットオフエネルギーはキーワード「scf.energycutoff」により制御されます。図 5 にカットオフエネルギーに対するメタン分子の全エネルギーの収束の様子を示します。ここで入力ファイルは「入力ファイル」の章で用いた「Methane.dat」です。カットオフエネルギーは基底関数に対してではなく、数値積分に対するものであることから、平面波基底系のようにカットオフエネルギーに応じて全エネルギーが高エネルギー領域から収束していくわけではありません。ほとんどの場合、150 から 200 Ryd のカットオフエネルギーが最適な選択となります。しかしながら、300 Ryd 以上のカットオフエネルギーを必要とする微妙な問題があることにも注意すべきです。極小点付近で非常に平坦なポテンシャル形状を計算する場合や、異なるスピン秩序間の小さなエネルギー差の計算がそのような微妙な問題となる可能性があります。

表 3 に水分子の構造パラメータと双極子モーメントのカットオフエネルギーへの依存性を示します。入力ファイルは「work」ディレクトリ内の「H2O.dat」です。およそ 90 Ryd で収束した結果が得られています。必要なカットオフエネルギー値は元素に依存しますが、多くの場合に 200 Ryd 程度で収束解が得られるでしょう。しかしながら御自身の計算対象の系の物理的特性が、どの様にカットオフエネルギーに依存しているのかチェックされることを推奨します。もう一方のカットオフエネルギー「1DFFT.EnergyCutoff」に対しては、通常、デフォルト値として 3600 (Ryd) を使用しており、この値で十分に収束しています。またその計算量も他の部分の計算と比較し、小さなものです。

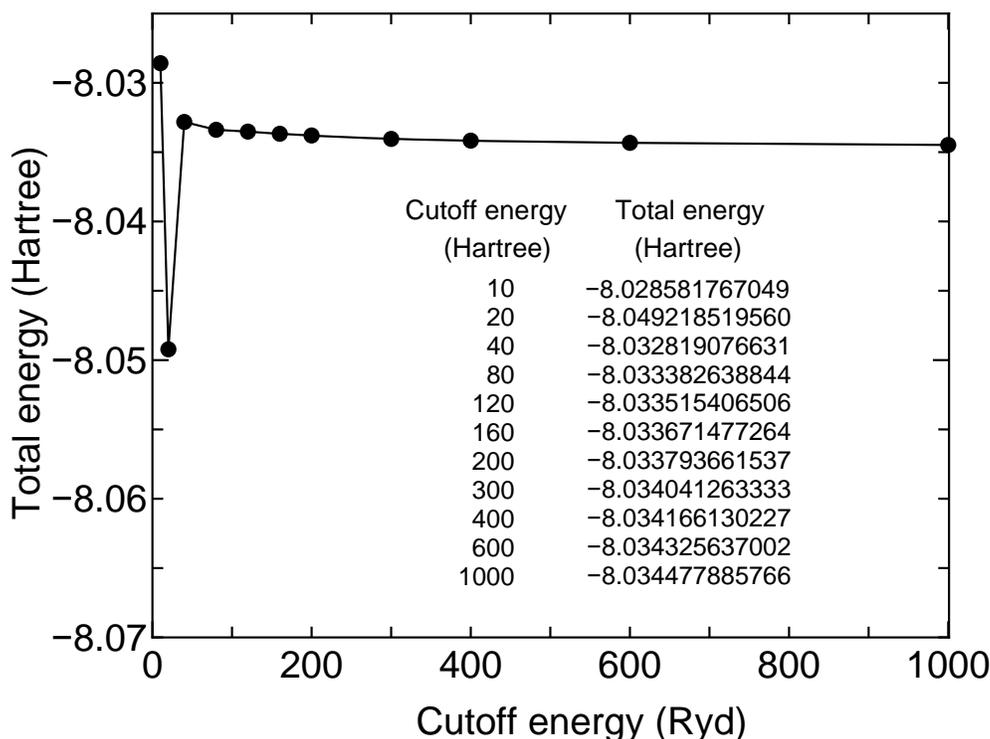


Figure 5: カットオフエネルギーに対するメタン分子の全エネルギーの収束性。

Table 3: カットオフエネルギーに対する水分子の構造パラメータと双極子モーメントの収束性。入力ファイルは「work」ディレクトリ内の「H2O.dat」。

| Ecut(Ryd) | r(H-O) (Å) | ∠ (H-O-H) (deg) | Dipole moment (Debye) |
|-----------|------------|-----------------|-----------------------|
| 60        | 0.970      | 103.4           | 1.838                 |
| 90        | 0.971      | 103.7           | 1.829                 |
| 120       | 0.971      | 103.7           | 1.832                 |
| 150       | 0.971      | 103.6           | 1.829                 |
| 180       | 0.971      | 103.6           | 1.833                 |
| Exp.      | 0.957      | 104.5           | 1.85                  |

## 12.2 バルクのエネルギー曲線計算に関するヒント

格子定数の関数としてバルク系のエネルギー曲線を計算する際には、エネルギー曲線に不連続性が現れる場合があります。一般に、この不連続性は格子定数の変化に伴い積分に用いられている実空間グリッドのグリッド数が変化することによるものです。例として、図 6 に bcc-Fe のエネルギー曲線を示しますが、カットオフエネルギーが固定であるとき (200 および 290 (Ryd) の場合)、エネルギー曲線が不連続 (それほど顕著ではありませんが) になっていることが分かります。不連続なエネルギーの飛びは計算グリッド数が変化する格子定数において発生しています。エネルギー曲線上の不連続性を避けるために、キーワード「scf.Ngrid」が利用可能です。

```
scf.Ngrid      32 32 32      # n1, n2, and n3 for a-, b-, and c-axes
```

このキーワードによって実空間グリッド数が明示的に指定された場合には、キーワード「scf.energycutoff」で指定したカットオフを参照することなしに、ここで与えたグリッド数で各単位胞ベクトルが離散化されます。図 6 から分かるように、 $32 \times 32 \times 32$  に固定されたグリッドでの計算では滑らかな曲線が得られています。

## 12.3 構造格子の相対位置の固定

OpenMX Ver. 3.9 では、キーワード「scf.energycutoff」もしくは「scf.Ngrid」で生成された実空間の積分グリッドは、一部のハミルトニアン行列要素の算出とポアソン方程式の解法に用いられています。ここで一部のハミルトニアン行列要素とは、差電子密度から生じる Hartree ポテンシャルと交換相関ポテンシャルに対する行列要素のことです。従って全エネルギーは原子座標とグリッド間の相対位置に依存します。原子座標の関数として相互作用エネルギーやエネルギー曲線を計算する場合には、それらの一連の計算に必要な全ての計算において、この相対位置を保持することが、誤差を軽減するために重要です。相対位置を保持するためには、グリッド生成のための原点の  $x$ 、 $y$ 、 $z$  成分を全ての計算において共通にする必要があります。標準出力には、次の様なグリッド生成時の原点となる  $x$ 、 $y$ 、 $z$  成分が「Grid\_Origin」として出力されています。

```
Grid_Origin xxx yyy zzz
```

一連の計算を行う際にある一つの計算からこの値「xxx yyy zzz」を取得した後は、その他の全ての計算において同一の値を使用しなければなりません。「Grid\_Origin」を明示的に指定するためには、次のキーワード「scf.fixed.grid」によって「xxx yyy zzz」を指定して下さい。

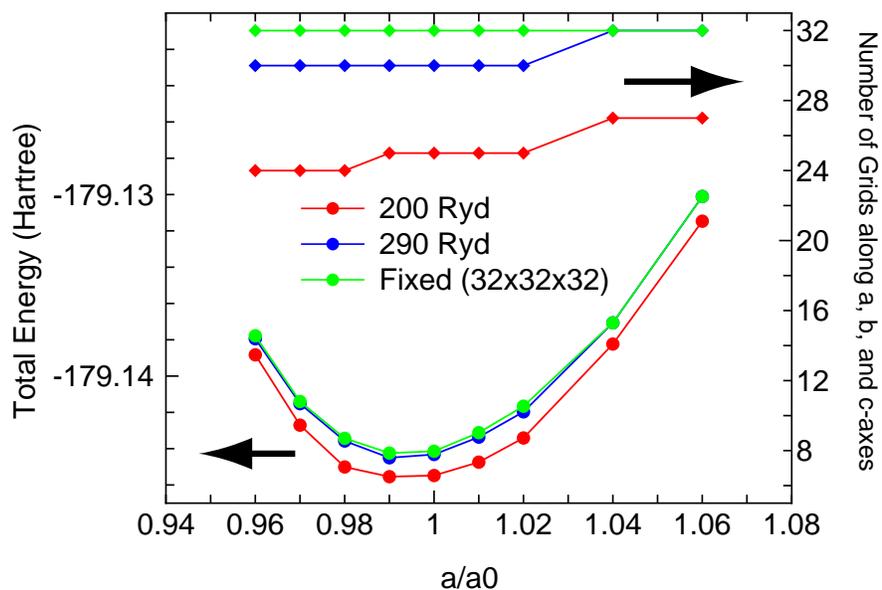


Figure 6: 格子定数の関数としての bcc 鉄の全エネルギー（実験による平衡格子定数  $a_0$  は  $2.87\text{\AA}$ ）。2 原子を含む立方単位胞により計算。入力ファイルは「work」ディレクトリ内の「Febcc2.dat」。

```
scf.fixed.grid xxx yyy zzz
```

この手続きにより、原子座標とグリッドとの相対位置をほぼ保持することが可能となり、積分グリッドの使用に伴う数値誤差が軽減されます。

さらに、前節「バルクのエネルギー曲線計算に関するヒント」で議論したように、一連の計算において格子定数が増える場合には、キーワード「scf.Ngrid」によりグリッド数を明示的に固定して下さい。

## 13 SCF 収束

### 13.1 概要

OpenMX Ver. 3.9 では、以下の 7 種類の電子密度混合法がキーワード「scf.Mixing.Type」により利用可能です。

- 単純混合 (Simple)  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight
- 反復部分空間における最小残差法 (RMM-DIIS) [58]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay
- Guaranteed reduction Pulay 法 (GR-Pulay) [57]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay
- Kerker 混合 (Kerker) [59]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Kerker.factor
- 電子密度に対する Kerker 計量付きの RMM-DIIS (RMM-DIISK) [58]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay, scf.Mixing.EveryPulay, scf.Kerker.factor
- Kohn-Sham ポテンシャルに対する Kerker 計量付きの RMM-DIIS (RMM-DIISV) [58]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay, scf.Mixing.EveryPulay, scf.Kerker.factor
- Kohn-Sham ハミルトニアンに対する RMM-DIIS (RMM-DIISH) [58]  
関連キーワード: scf.Init.Mixing.Weight, scf.Min.Mixing.Weight, scf.Max.Mixing.Weight, scf.Mixing.History, scf.Mixing.StartPulay, scf.Mixing.EveryPulay,

最初の 3 種類の方法においては、密度行列を混合することで入力密度行列が生成されます。密度行列は実空間での量とみなすことができ、(スピン)電子密度に容易に変換することが可能です。一方、続く 3 種類の方法 (Kerker, RMM-DIISK, RMM-DIISV) では、フーリエ空間で電子密度混合が行われます。最後の方法 (RMM-DIISH) は Kohn-Sham ハミルトニアンの混合が行われ、プラス U の計算や拘束条件付の計算に有効な方法となっています。「RMM-DIIS」, 「GR-Pulay」, 「RMM-DIISK」による混合法は全て Pulay 型混合法です。一般的に、大きなギャップを持つ系ではどの混合法を使用しても SCF 収束を達成することが可能です。しかし、小さなギャップを持つ系や金属系では、SCF 計算における電子密度の振動の問題 (charge sloshing) がしばしば深刻となることから、SCF 収束を達成することは容易ではありません。そのような困難な系に対応するために、「Kerker」, 「RMM-DIISK」および「RMM-DIISV」による 3 種の混合法が利用可能です。この 3 種の混合法は金属系の SCF 収束を達成するための有効な方法です。「Kerker」, 「RMM-DIISK」または「RMM-DIISV」を使用する際には、SCF 計算の収束を加速するために以下の処方が有用です。

- 「scf.Mixing.History」を増やす。30 から 50 の比較的大きな値によって収束しやすくなります。さらに「scf.Mixing.EveryPulay」は 1 に設定されなければなりません。
- 「scf.Mixing.StartPulay」に、大きめの値を使用する。Pulay 型の混合を開始する前に、ある程度の収束を達成しなければなりません。「scf.Mixing.StartPulay」の適切な値はおそらく 10 から 30 でしょう。
- 金属系の場合には「scf.ElectronicTemperature」に大きめの値を使用する。「scf.ElectronicTemperature」があまりに低い場合には、しばしば数値的不安定性が発生します。

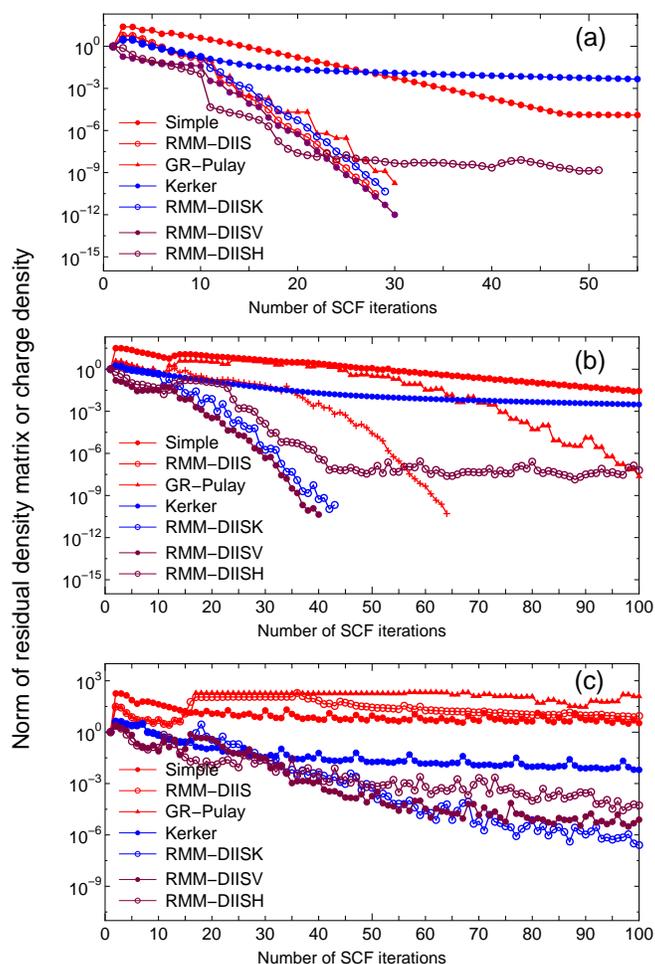


Figure 7: 5 種類の電子密度混合法における密度行列もしくは電子密度の残差ノルムの SCF 収束の過程。(a) シアル酸分子、(b) Pt<sub>13</sub> クラスタ、(c) Pt<sub>63</sub> クラスタ。入力ファイルは「work」ディレクトリ内の「SialicAcid.dat」, 「Pt13.dat」, 「Pt63.dat」。

SCF 計算における電子密度の振動は一般に電子密度の長波長成分から生じています。波数  $q$  の関数である Kerker 重み  $w_q$  を次式の内積計算に導入することで、この長波長成分からの振動を抑制することができます。ここで振動の抑制度合は Kerker 因子  $\alpha$  を調整することで制御され、その値はキーワード

「scf.Kerker.factor」で与えることができます。

$$\langle A|B \rangle = \sum_{\mathbf{q}} \frac{A_{\mathbf{q}}^* B_{\mathbf{q}}}{w_{\mathbf{q}}}$$

$$w_{\mathbf{q}} = \frac{|\mathbf{q}|^2}{|\mathbf{q}|^2 + q_0^2}$$

$$q_0 = \alpha |\mathbf{q}_{\min}|$$

ここで、 $\mathbf{q}_{\min}$  は 0 ベクトルを除く最小の大きさを持つ  $\mathbf{q}$  ベクトルです。大きな  $\alpha$  は電子密度の振動を大幅に抑制しますが、収束が遅くなる可能性があります。最適値は系に依存しますので、計算対象の系に適した値を、ユーザーが調整することが必要です。

さらに、「RMM-DIISK」の振る舞いは次のキーワードにより調整することができます。

```
scf.Mixing.EveryPulay    5    # default = 1
```

Pulay 型混合法の残差ベクトルは、混合ステップが累積するにつれてある特定の部分空間を張るだけとなり、このため、その部分空間に直交した線形独立な新しいベクトル成分が導入されないために収束が困難になってきます。この線形従属問題を回避する一つの方法は、Kerker 混合の合間に時折 Pulay 型混合を行うことです。この方法ではキーワード「scf.Mixing.EveryPulay」を使用して頻度を指定することができます。例えば「scf.Mixing.EveryPulay=5」の場合、5 回の SCF 反復ごとに Pulay 混合が、他のステップでは Kerker 型混合が行われます。「scf.Mixing.EveryPulay=1」は従来の Pulay 型混合に対応します。キーワード「scf.Mixing.EveryPulay」は「RMM-DIISK」に対してのみ使用でき、デフォルト値は「1」であることに注意して下さい。

SCF 収束性を向上させるための上記の処方箋は多くの場合に有効です。しかし、収束を加速させるために最も推奨されるのは以下の方法です。

- 「scf.Mixing.History」を増やす。30 から 50 の比較的大きな値によって収束しやすくなります。さらに「scf.Mixing.EveryPulay」は 1 に設定しなければなりません。

RMM-DIIS、RMM-DIISK、RMM-DIISV などの Pulay 型混合は準ニュートン法に基づいていることから、収束速度はいかに適切な近似ヘッセ (Hessian) 行列を見つけられるかに依存します。「scf.Mixing.History」を大きくするに従い、計算される近似ヘッセ行列の精度が向上する可能性があります。

図 7 は (a) シアル酸分子、(b) Pt<sub>13</sub> クラスタ、(c) Pt<sub>63</sub> クラスタに対する 7 種類の混合法の SCF 収束の比較を示しています。密度行列もしくは電子密度の残差ノルムは「System.Name.out」ファイル内で NormRD として記録されています。またこの計算で用いた入力ファイルは「work」ディレクトリ内の「SialicAcid.dat」、「Pt13.dat」、「Pt63.dat」です。図 7 に示された全ての系に対して「RMM-DIISK」と「RMM-DIISV」が頑健に動作していることがわかります。SCF 収束が極めて困難な場合には、大きな「scf.Kerker.factor」および小さな「scf.Max.Mixing.Weight」を用いた「Kerker」の使用が必要となりますが、多くの場合で、「RMM-DIISK」と「RMM-DIISV」が最良の選択となります。図 7 には示されていませんが、我々の経験上、plus  $U$  法と拘束法には「RMM-DIISH」が適切です。

## 13.2 Kerker 因子の自動決定

入力ファイルでキーワード「scf.Kerker.factor」が与えられていない場合、次式により Kerker 因子  $\alpha$  の適切な値が自動的に推定されます。

$$\alpha = \frac{0.5}{|\mathbf{b}_{\min}|^2} \left( 4 \frac{Dq}{Aq} + 1.0 \right),$$

このとき、

$$Aq = \frac{1}{3} (|\mathbf{b}_1|^2 + |\mathbf{b}_2|^2 + |\mathbf{b}_3|^2),$$

$$Dq = \frac{1}{3} \sum_{i < j} \left| |\mathbf{b}_i|^2 - |\mathbf{b}_j|^2 \right|,$$

となります。ここで、 $\mathbf{b}_i (i = 1, 2, 3)$  は逆格子ベクトルであり、 $\mathbf{b}_{\min}$  は  $\{\mathbf{b}\}$  内での最小ベクトルです。この式では系のサイズと異方性に対する  $\alpha$  の依存性が考慮されています。一連の数値計算から、推測した値はほとんどの場合に、適切に機能していることが分かっています。

## 13.3 SCF 収束パラメータの on-the-fly での調整

SCF 計算を行っている途中で SCF 収束に関する以下のパラメータを変更することが可能です。

```
scf.maxIter
scf.Min.Mixing.Weight
scf.Max.Mixing.Weight
scf.Kerker.factor
scf.Mixing.StartPulay
```

例えば、入力ファイルで次の 2 種類のキーワードを以下のように指定した場合には、

```
System.CurrentDirectory      ./      # default=./
System.Name                   c60
```

ディレクトリ「./」内に「c60\_SCF\_keywords」という名前のファイルを作成して下さい。このファイル中に例えば、以下のように記述します。

```
scf.maxIter                   100
scf.Min.Mixing.Weight         0.01
scf.Max.Mixing.Weight         0.10
scf.Kerker.factor             10.0
scf.Mixing.StartPulay         30
scf.criterion                 1.0e-6
```

OpenMX は SCF ステップ毎にファイル「c60\_SCF\_keywords」をチェックします。ファイルの読み込みに成功した場合には、標準出力に次のメッセージが表示されます。

```
The keywords for SCF iteration are renewed by ./c60_SCF_keywords.
```

また、キーワード「scf.maxIter」に対して負の値が与えられた場合には、OpenMX は終了します。大規模計算を行う場合、on-the-fly での SCF 収束パラメータの調整は有用でしょう。

## 14 再スタート

### 14.1 概要

状態密度、バンド分散、分子軌道などの計算を行う場合には予め自己無撞着計算を行っておき、二回目以降の計算では自己無撞着計算をスキップすれば、計算時間を節約することができます。この計算の再スタートを行うためのキーワードが「scf.restart」として用意されています。

```
scf.restart          on          # on|off,default=off
```

キーワード「scf.restart」を「on」にすると、最初の計算で作成された再スタート用のファイルが読み込まれ、二回目の計算での初期のハミルトニアンおよび(スピン)電子密度として使用されます。ここで二回目の計算の「System.Name」は最初の計算と同一でなければなりません。再スタートファイルは「work」ディレクトリ内の「System.Name\_rst」ディレクトリ内に保存されています。ここで *System.Name* は「System.Name」を意味します。「System.Name\_rst」内の再スタートファイルは密度行列混合法およびフーリエ空間混合法の両方に関係する全ての情報を含んでいます。従って、二回目の計算において他の混合法を使用することも可能です。再スタートの例として、図8にSCF計算の過程を示します。この計算は「work」ディレクトリ内にある入力ファイル「C60.dat」を用いて行ったものです。図8より、二回目の計算では再スタートファイルの使用によりSCF計算がすぐに終了していることが分かります。

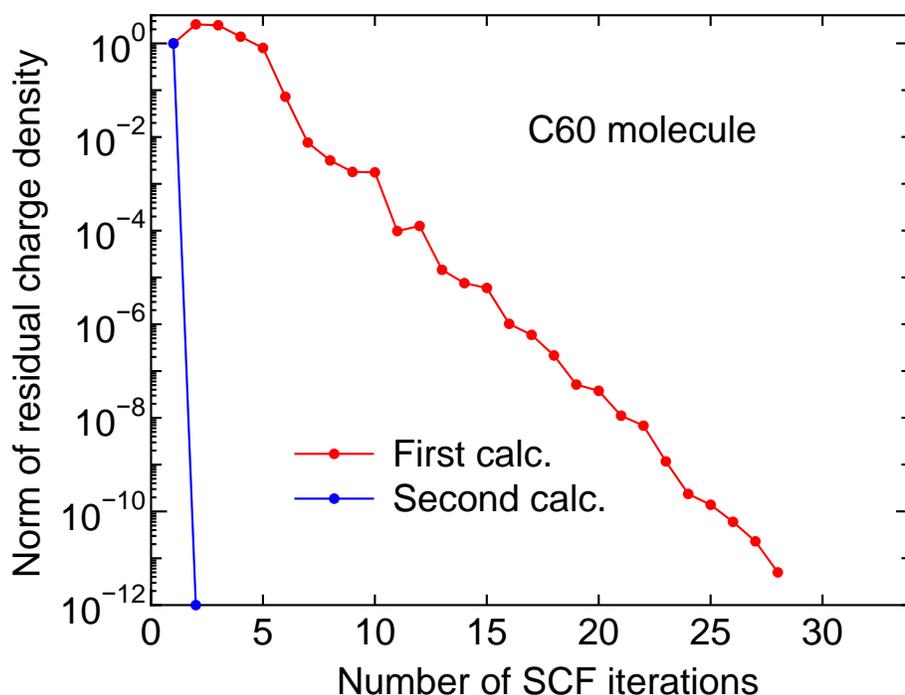


Figure 8: C<sub>60</sub> 分子の SCF 収束。二回目の計算では、最初の計算により生成された再スタートファイルを使用。入力ファイルは「work」ディレクトリ内にある「C60.dat」。

### 14.2 MD および構造最適化中の外挿法

構造最適化および分子動力学 (MD) シミュレーションでは、SCF の収束を加速するために、過去のステップで生成された再スタートファイルが自動的に利用されます [60, 61]。この方法は電子密度の外挿法に

基づいており、外挿に用いる過去のステップ数をキーワード「scf.ExtCharge.History」により設定できます。

```
scf.ExtCharge.History      2      # default=2
```

一連のベンチマーク計算から、「scf.ExtCharge.History」を2に設定すると収束が加速し、それ以上の大きな値を用いると数値的に不安定となることが分かっています。従って、デフォルト値の2を使用することを推奨します。

### 14.3 再スタート計算用の入力ファイル

構造最適化および分子動力学 (MD) シミュレーションの各ステップにおいて、再スタート計算用の入力ファイル「*System.Name.dat#*」が出力されます。「*System.Name.dat#*」には最終ステップでの原子座標が挿入されており、また「構造格子の相対位置の固定」の章で説明された「Grid-Origin」の設定も行われています。この入力ファイル「*System.Name.dat#*」を用いて構造最適化および分子動力学 (MD) シミュレーションを前回の最終ステップから継続することが可能です。

## 15 構造最適化

### 15.1 最急降下法

構造最適化の例を本章で説明します。初期構造として、「入力ファイル」の章で扱ったメタン分子を取り上げますが、メタン分子の炭素原子の  $x$  座標が以下のように  $0.3 \text{ \AA}$  に変更されています。

```
<Atoms.SpeciesAndCoordinates
  1  C      0.300000  0.000000  0.000000  2.0  2.0
  2  H     -0.889981 -0.629312  0.000000  0.5  0.5
  3  H      0.000000  0.629312 -0.889981  0.5  0.5
  4  H      0.000000  0.629312  0.889981  0.5  0.5
  5  H      0.889981 -0.629312  0.000000  0.5  0.5
Atoms.SpeciesAndCoordinates>
```

次に、キーワード「MD.type」を「Opt」に、キーワード「MD.maxIter」を200に設定します。「Opt」は可変の前因子を持った最急降下法です。メタン分子の構造最適化の収束履歴を図9(a)に示します。図では構造最適化のステップに対して原子にかかる力の絶対最大値を示しています。最急降下法では収束の状況に応じて前因子を変化させるため、前因子が大きくなりすぎて構造が停留点を越えていることが見て取れます。その後で完全な収束に至っています。この計算は「work」ディレクトリ内の「Methane2.dat」を使用することにより、再現することができます。またメタン分子の場合と同様、図9(b)に示すようにダイヤモンド構造のケイ素に対しても類似した挙動を見ることができます。

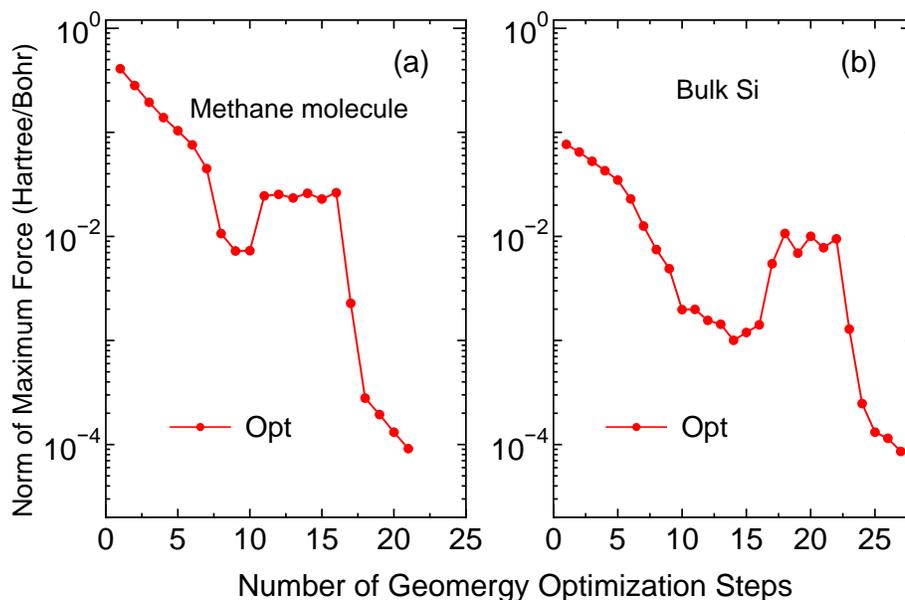


Figure 9: (a) メタン分子 と (b) ダイヤモンド構造のケイ素の原子の構造最適化における原子にかかる力の絶対最大値。初期構造は平衡構造から変形されたものを使用。入力ファイルはそれぞれ、「work」ディレクトリ内の「Methane2.dat」および「Si8.dat」。

## 15.2 EF、BFGS、RF、DIIS 最適化

「Opt」は頑健な方法ですが、一般的に収束速度は遅いと言えます。より高速な方法として、準ニュートン法に基づいた構造最適化手法が利用可能です。デカルト座標内で実行される固有ベクトル追跡 (EF) 法 [63]、Broyden-Fletcher-Goldfarb-Shanno (BFGS) 法 [65]、合理的関数 (RF) 法 [64]、そして反復部分空間 (DIIS) 法 [62] がサポートされています。EF および RF 法では、近似ヘシアンは BFGS 法により更新されます。キーワード「MD.Type」により、「Opt」、「EF」、「BFGS」、「RF」、「DIIS」の一つを選択して下さい。関連キーワードは次の通りになります。

|                     |        |                                 |
|---------------------|--------|---------------------------------|
| MD.Type             | EF     | # Opt DIIS BFGS RF EF           |
| MD.Opt.DIIS.History | 3      | # default=3                     |
| MD.Opt.StartDIIS    | 5      | # default=5                     |
| MD.Opt.EveryDIIS    | 200    | # default=200                   |
| MD.maxIter          | 100    | # default=1                     |
| MD.Opt.criterion    | 1.0e-4 | # default=0.0003 (Hartree/Bohr) |

特に、これらの準ニュートン法に基づいた構造最適化法は次の 2 つのキーワードにより制御されます。

|                     |   |             |
|---------------------|---|-------------|
| MD.Opt.DIIS.History | 3 | # default=3 |
| MD.Opt.StartDIIS    | 5 | # default=5 |

キーワード「MD.Opt.DIIS.History」は近似ヘシアンを更新するための過去の履歴ステップ数を指定します。デフォルト値は 3 です。また、「EF」、「BFGS」、「RF」、「DIIS」を開始する構造最適化ステップをキーワード「MD.Opt.StartDIIS」により指定します。これらの方法を開始する前の構造最適化ステップは、最急降下法により実行されます。デフォルト値は 5 です。

最適化の初期ステップは初期構造における力の最大値 (絶対値) を参照することで自動的に調整されます。図 10 は、分子やバルクにおいて力の最大値 (絶対値) が 0.0003 Hartree/Bohr 以下に到達するまでの構造最適化ステップ数を示しています。ここに示されるように、EF および BFGS 法も同様な性能を示しているものの、ほとんどの場合に RF 法が最もロバストで効率的な方法です。これらの計算に使用された入力ファイルと出力ファイルは「work/geoopt\_example/」ディレクトリ内にあります。

また、これらの準ニュートン法により、構造が最小停留点よりもむしろ鞍点に収束される可能性もあることに注意すべきです。これは、準ニュートン法の使用が開始された時の構造が変曲点に達しない場合に起こり得ます。そのような場合には、準ニュートン法に移る前に、構造を最急降下法により十分に最適化するべきです。対処方法は、「MD.Opt.StartDIIS」に対して大きな値を使用するだけです。十分な収束に対して多数の反復ステップが要求される系では、ファイル「System.Name.dat#」を使用して計算を再スタートすることも必要かも知れません。ここで「System.Name」は入力ファイルで指定した「System.Name」です。

一般的に、「scf.energy.cutoff」に大きな値を用いることでエネルギー曲線が滑らかになり、結果として収束性を早めることができます。これは分子固体などの弱く相互作用している系において特に顕著です。このような弱く相互作用している系においては、300-400Ry 程度の大きな「scf.energy.cutoff」の使用が推奨されます。

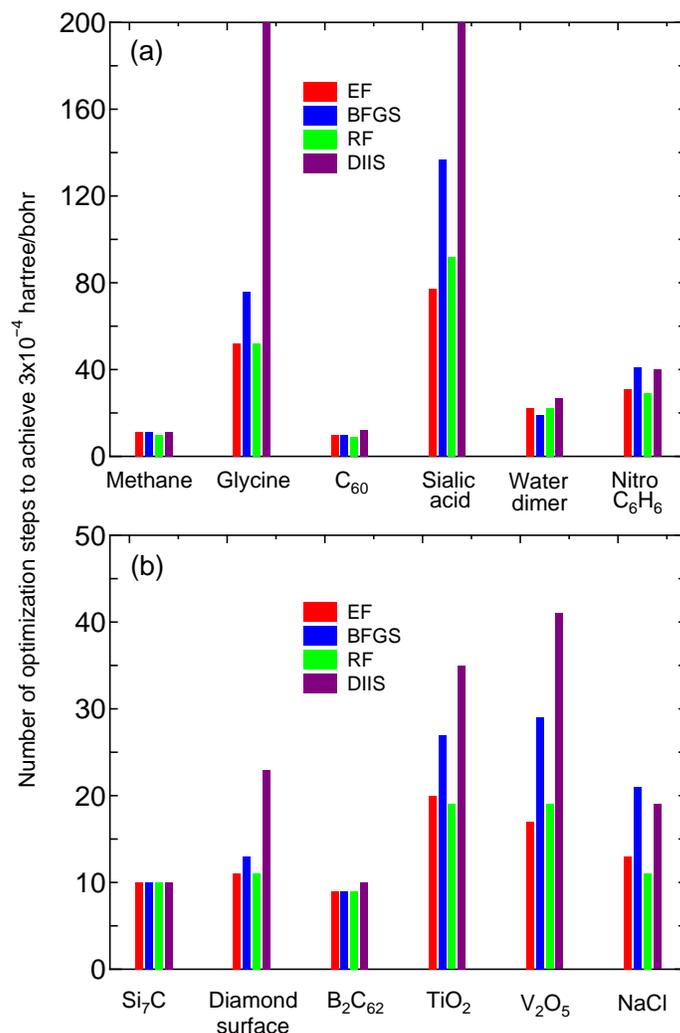


Figure 10: (a) 分子系および (b) バルク系に対する構造最適化の反復回数。収束判定条件は原子に働く力の絶対最大値が  $3 \times 10^{-4}$  Hartree/Bohr 以下。

### 15.3 RF 及び EF 最適化におけるヘシアン行列の初期値

RF 及び EF 最適化では 2 種類の初期ヘシアン行列が使用可能であり、以下のように設定します。

```
MD.Opt.Init.Hessian      Schlegel      # Schlegel|iden, default=Schlegel
```

デフォルトの「Schlegel」は Schlegel [66] によって提案された方法であり、結合の伸縮・屈曲・捻りからなる単純なモデルに基づき初期ヘシアン行列を見積もります (OpenMX Ver.3.9 では結合の伸縮のみを考慮して初期ヘシアン行列を構築します)。もう 1 つの「iden」では、ヘシアン行列を単位行列として計算を開始します。どちらの方法においても、ヘシアン行列は構造最適化ステップごとに Broyden-Fletcher-Goldfarb-Shanno (BFGS) 法 [65] を用いて更新されます。構造最適化手法として「BFGS」、「DIIS」を用いた場合には「Schlegel」は使用できないことに注意して下さい。一般的に「iden」よりも「Schlegel」の方が収束性がよいため、「Schlegel」をデフォルトとして設定しています。「EF」を用いた最適化における「Schlegel」と「iden」の比較を図 11 に示しましたが、ほとんどの分子系やバルク系において「Schlegel」の方が少ないステップ数で収束していることが分かります。

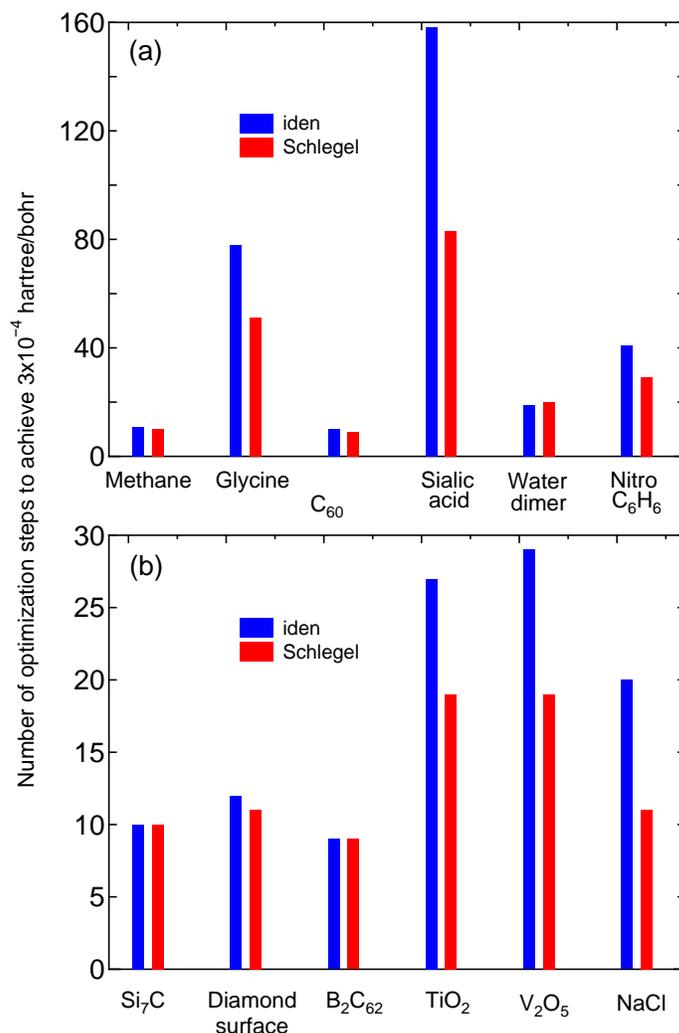


Figure 11: 初期ヘシアン行列を「iden」、「Schlegel」とした際の、力の最大値が  $3 \times 10^{-4}$  Hartree/Bohr を下回るまでに要した最適化ステップ数 (a) 分子系 (b) バルク系

#### 15.4 制約条件付の構造最適化

構造最適化の際に任意の原子座標を初期座標に固定することができます。この制約条件はキーワード「MD.Fixed.XYZ」を用いて各原子の  $x$ 、 $y$ 、 $z$  座標に対して、以下のように設定します。

```
<MD.Fixed.XYZ
  1  1  1  1
  2  1  0  0
MD.Fixed.XYZ>
```

上記の例は2つの原子から構成される系の場合です。系に  $N$  個の原子がある場合、 $N$  個分の指定が必要です。第1列は「Atoms.SpeciesAndCoordinates」で個々の原子を特定するのに使用した連番です。第2~4列は、それぞれ  $x$ 、 $y$ 、 $z$  座標のフラグです。フラグが「1」の場合には座標を固定、「0」の場合には緩和されます。上記の例では、原子1の  $x$ 、 $y$ 、 $z$  座標は固定、原子2は  $x$  座標のみが固定されています。

す。デフォルトでは全ての座標が緩和される指定になっています。このキーワードによる原子座標の固定は、すべての構造最適化および分子動力学の方法において有効です。制限条件付の構造最適化は大規模な系の局所構造の精密化にとって有用でしょう。

## 15.5 構造最適化の再スタート

最初の構造最適化の試行により、収束条件に達しなかった場合には、入力ファイル「*System.Name.dat#*」を使って構造最適化を再スタートすることができます。入力ファイル「*System.Name.dat#*」は構造最適化の各ステップで生成され、最終ステップでの幾何構造が挿入されています。構造最適化を再スタートさせるには最終ステップでの幾何構造を用いるだけでなく、最初の試行で計算された近似ヘシアンを用いることも必要です。OpenMX Ver. 3.9では、近似ヘシアンも構造最適化ステップ毎に保存され、「*System.Name.dat#*」により再スタートを行うときに自動的に再利用されます。そのため、「*System.Name.dat#*」を用いて複数の計算によって断続的に構造最適化が行われたとしても、構造最適化ステップのための反復回数は単一計算の場合と同じになります。実行時間に制限がある共用計算システムを用いて、大規模系を最適化する場合にはこの機能が有用となるでしょう。

## 16 セル可変構造最適化

### 16.1 概要

OpenMX Ver.3.9 ではセル可変の構造最適化を拘束条件あり/なしの双方で使用可能となっています。関連するキーワードは以下の通りです。

|                     |        |   |
|---------------------|--------|---|
| MD.Type             | RFC5   | # OptC1 OptC2 OptC3 OptC4 OptC5<br># OptC6 OptC7 RFC5 RFC6 RFC7 |
| MD.Opt.DIIS.History | 3      | # default=3   |
| MD.Opt.StartDIIS    | 5      | # default=5   |
| MD.Opt.EveryDIIS    | 200    | # default=200   |
| MD.maxIter          | 100    | # default=1   |
| MD.Opt.criterion    | 1.0e-4 | # default=0.0003 (Hartree/Bohr)                                 |

上記のキーワードは「構造最適化」セクションとまったく同じものなので、セル可変の構造最適化は従来の構造最適化と同様に制御可能です。セル可変の構造最適化は plus U 法を含めたコリニア計算のみに利用可能です。ただし DFT-D2 法及び DFT-D3 法による vdW 相互作用はサポートしていません。セル可変の構造最適化は「MD.Type」によって指定し、以下のオプションが選択可能です。

- OptC1  
初期内部座標を維持したまま、格子ベクトルに拘束をかけずに最適化を行います。最適化は、最急降下法によって行われます。
- OptC2  
初期内部座標を維持したまま、格子ベクトル間の角度を初期値に固定して最適化を行います。従って、格子ベクトルの長さのみが最適化されます。最適化は、最急降下法によって行われます。
- OptC3  
初期内部座標を維持したまま、格子ベクトル間の角度を初期値に固定し、格子ベクトルの長さも互いに等しい状態を保ったまま最適化を行います:  $|a_1| = |a_2| = |a_3|$ 。従って、格子ベクトルの長さのみが最適化されます。最適化は、最急降下法によって行われます。
- OptC4  
初期内部座標を維持したまま、格子ベクトル間の角度を初期値に固定し、格子ベクトルの長さを以下の拘束条件のもとで最適化します:  $|a_1| = |a_2| \neq |a_3|$ 。従って、格子ベクトルの長さのみが最適化されます。最適化は、最急降下法によって行われます。
- OptC5  
格子ベクトル、内部座標ともに拘束をかけずに同時に最適化を行います。最適化は、最急降下法によって行われます。
- OptC6  
格子ベクトル  $a_3$  を固定したまま、格子ベクトルと内部座標ともに同時に最適化を行います。最適化は、最急降下法によって行われます。

- OptC7

格子ベクトル  $a_2$  と  $a_3$  を固定したまま、格子ベクトルと内部座標ともに同時に最適化を行います。最適化は、最急降下法によって行われます。

- RFC5

格子ベクトル、内部座標ともに拘束をかけずに同時に最適化を行います。このオプションでは、RF法 [64] と DIIS 法 [62] の混合による内部座標の最適化が、BFGS 法 [65] を用いたヘシアン行列の更新と共に行われます。初期ヘシアン行列は「MD.Opt.Init.Hessian」によって「iden」もしくは「Schlegel」で指定し、デフォルトでは「Schlegel」が設定されています（詳細は「構造最適化」の章をご参照下さい）。

- RFC6

格子ベクトル  $a_3$  を固定したまま、格子ベクトルと内部座標ともに同時に最適化を行います。最適化は RFC5 と同じ方法で行われます。

- RFC7

格子ベクトル  $a_2$  と  $a_3$  を固定したまま、格子ベクトルと内部座標ともに同時に最適化を行います。最適化は RFC5 と同じ方法で行われます。

目的に応じて上記のオプションを選択して下さい。将来的には、上記以外の拘束条件も実装予定です。

セル可変の構造最適化の例として、ダイヤモンド構造の格子ベクトルと内部座標の同時最適化を以下に示します。計算は以下のコマンドによって行います。

```
% mpirun -np 16 openmx Cdia-RFC5.dat > Cdia-RFC5.std &
```

入力ファイル「Cdia-RFC5.dat」は「work/cellopt.example」ディレクトリに収められていますので、同じ計算を実施することが可能です。この計算例では、初期構造は以下のように歪んでいます。

```
Atoms.Number          2
Atoms.SpeciesAndCoordinates.Unit  frac # Ang|AU
<Atoms.SpeciesAndCoordinates
  1  C  0.10000000000000  0.00000000000000  -0.05000000000000  2.0  2.0
  2  C  0.25000000000000  0.25000000000000  0.25000000000000  2.0  2.0
Atoms.SpeciesAndCoordinates>
Atoms.UnitVectors.Unit          Ang # Ang|AU
<Atoms.UnitVectors
  1.6400  1.6400  0.0000
  1.6400  0.0000  1.6400
  0.0000  1.6400  1.6400
Atoms.UnitVectors>
```

クラスタ計算機 (Intel Xeon of 2.6 GHz) では 12 回の最適化ステップに 326 秒を要しました。全エネルギーと原子座標もしくは格子ベクトルに対する全エネルギーの最大勾配を「System.Name.out」にて確認できます。この計算例の場合には、「Cdia-RFC5.out」にて以下の情報を得ることができます。

```
*****
*****
History of cell optimization
```

\*\*\*\*\*  
 \*\*\*\*\*

| MD_iter | SD_scaling | Maximum force <br>(Hartree/Bohr) | Maximum step<br>(Ang) | Utot<br>(Hartree) | Enpy<br>(Hartree) | Volume<br>(Ang <sup>3</sup> ) |
|---------|------------|----------------------------------|-----------------------|-------------------|-------------------|-------------------------------|
| 1       | 1.25981732 | 0.16438857                       | 0.10583545            | -11.59621750      | -11.59621750      | 8.82188800                    |
| 2       | 1.25981732 | 0.08853079                       | 0.05902053            | -11.64994330      | -11.64994330      | 9.81261691                    |
| 3       | 1.25981732 | 0.04581932                       | 0.03054622            | -11.66453803      | -11.66453803      | 10.28662955                   |
| 4       | 1.25981732 | 0.02205340                       | 0.01470227            | -11.66928384      | -11.66928384      | 10.56026328                   |
| 5       | 3.14954331 | 0.01336972                       | 0.02228286            | -11.67121215      | -11.67121215      | 10.73689973                   |
| 6       | 3.14954331 | 0.00678359                       | 0.01130598            | -11.67332696      | -11.67332696      | 11.04288573                   |
| 7       | 3.14954331 | 0.00487464                       | 0.01195765            | -11.67421713      | -11.67421713      | 11.13669753                   |
| 8       | 3.14954331 | 0.00354039                       | 0.02370087            | -11.67479906      | -11.67479906      | 11.18107598                   |
| 9       | 3.14954331 | 0.00157491                       | 0.00373195            | -11.67534267      | -11.67534267      | 11.29495641                   |
| 10      | 3.14954331 | 0.00137813                       | 0.00160469            | -11.67537385      | -11.67537385      | 11.34330266                   |
| 11      | 3.14954331 | 0.00067979                       | 0.00165878            | -11.67538616      | -11.67538616      | 11.37836604                   |
| 12      | 3.14954331 | 0.00003708                       | 0.00000000            | -11.67538985      | -11.67538985      | 11.39519327                   |

最大勾配の絶対値が速やかに収束しており、閾値である 0.0003 Hartree/bohr に到達していることが見て取れます。「work/cellopt\_example」に、その他のセル可変構造最適化の例が含まれていますので参考にしてください。

## 16.2 ストレステンソル

ここまで「MD.Type」キーワードによるセル可変の構造最適化を説明してきましたが、これらの方法ではストレステンソルは歪みに対する全エネルギーの勾配から解析的に自動計算され、格子ベクトルに対する勾配として変換されていました。セル可変の構造最適化を実施せずに、格子ベクトルに対する全エネルギーの勾配を求めたい場合には、以下のキーワードを用いて下さい。

```
scf.stress.tensor on # on|off, default=off
```

「scf.stress.tensor」キーワードが「on」の場合には、「System.Name.out」に格子ベクトルに対する全エネルギーの勾配が出力されます。

## 16.3 格子ベクトルに対する拘束

「MD.Type」キーワードにて「OptC1」、「OptC2」、「OptC5」を指定した場合には、以下のように格子ベクトルのデカルト成分を初期値のまま固定することが可能です。設定するキーワードは以下となります。

```
<MD.Fixed.Cell.Vectors
  0 0 1
  0 0 0
  0 0 0
MD.Fixed.Cell.Vectors>
```

1行目のフラグは  $a_x, a_y, a_z$ , 2行目は  $b_x, b_y, b_z$ , 3行目は  $c_x, c_y, c_z$  に対応します。「1」の場合には成分を固定し、「0」の場合には可変とします。

## 16.4 エンタルピーの最適化

印加された圧力のもとで可変セルの最適化を実行できます。これは次式のエンタルピー  $H$  を最小化することです。

$$H = E + pV, \quad (1)$$

ここで  $E$  は単位胞あたりの全エネルギー、 $p$  は印加された圧力、そして  $V$  は単位胞の体積です。エンタルピーの最適化を実行するには入力ファイルに以下のキーワードを含めるだけです。

```
MD.applied.pressure      10.0      # in GPa, default=0
```

正の圧力はセルの圧縮に対応します。この機能はキーワード「MD.Type」に指定できる 'OptC1', 'OptC2', 'OptC3', 'OptC4', 'OptC5', 'OptC6', 'OptC7'、さらに 'RFC5', 'RFC6', and 'RFC7' と互換性があります。例として、ディレクトリ 'work' 内の入力ファイル Si8-pV.dat を用いてエンタルピーの最適化を実行できます。これは 10GPa 下での Si 結晶の最適化です。計算後、出力ファイル si8-pV.out に最適化の履歴が確認できます。

```
*****
History of cell optimization
*****
*****
*****
```

| MD_iter | SD_scaling | Maximum force <br>(Hartree/Bohr) | Maximum step<br>(Ang) | Utot<br>(Hartree) | Enpy<br>(Hartree) | Volume<br>(Ang <sup>3</sup> ) |
|---------|------------|----------------------------------|-----------------------|-------------------|-------------------|-------------------------------|
| 1       | 1.25981732 | 0.07663140                       | 0.05108760            | -32.84057849      | -32.47335956      | 160.10300700                  |
| 2       | 1.25981732 | 0.06717954                       | 0.04478636            | -32.84541333      | -32.48138995      | 158.70978745                  |
| 3       | 1.25981732 | 0.05879663                       | 0.03919775            | -32.84853574      | -32.48736913      | 157.46427382                  |
| 4       | 1.25981732 | 0.05131728                       | 0.03421152            | -32.85047522      | -32.49182806      | 156.36581813                  |
| 5       | 3.14954331 | 0.04468030                       | 0.07446716            | -32.85159836      | -32.49515060      | 155.40690918                  |
| 6       | 3.14954331 | 0.02956430                       | 0.04927383            | -32.85232293      | -32.50062291      | 153.33695214                  |
| 7       | 3.14954331 | 0.01960389                       | 0.03267316            | -32.85158714      | -32.50293764      | 152.00696345                  |
| 8       | 3.14954331 | 0.01318467                       | 0.02197446            | -32.85069024      | -32.50392104      | 151.18717226                  |
| 9       | 7.87385828 | 0.00909382                       | 0.03789092            | -32.84998761      | -32.50434789      | 150.69473500                  |
| 10      | 7.87385828 | 0.00253118                       | 0.00537839            | -32.84867882      | -32.50470324      | 149.96919000                  |
| 11      | 7.87385828 | 0.00198428                       | 0.03321825            | -32.84877730      | -32.50477155      | 149.98234416                  |
| 12      | 7.87385828 | 0.00271856                       | 0.01866538            | -32.84922787      | -32.50499775      | 150.08016284                  |
| 13      | 7.87385828 | 0.00086782                       | 0.00943670            | -32.84942256      | -32.50507226      | 150.13256385                  |
| 14      | 7.87385828 | 0.00077020                       | 0.00982293            | -32.84949585      | -32.50509162      | 150.15607426                  |
| 15      | 7.87385828 | 0.00020223                       | 0.00270074            | -32.84950610      | -32.50511244      | 150.15146767                  |
| 16      | 7.87385828 | 0.00005544                       | 0.00000000            | -32.84950546      | -32.50511390      | 150.15055140                  |

全エネルギーの減少よりむしろ体積の収縮によりエンタルピーが実際に最適化されることが確認できます。

さらに、以下のキーワードで直交結晶系において特定の軸のみに圧力をかけることもできます。

```
MD.applied.pressure.flag  1 1 1  # default=1 1 1
```

既定の設定は「111」で、これは a、b、そして c 軸に沿って等しく印加された等方的な圧力を意味します。このキーワードに「100」を指定すると、圧力は直方体結晶系の bc 面に垂直な方向である a 軸のみに沿って印加されます。この機能は直方体結晶系に有効です。この場合は以下のように非零の要素が対角要素となるように単位ベクトルを与える必要があります。

```
<Atoms.UnitVectors
 10.000  0.000  0.000
  0.000  8.000  0.000
  0.000  0.000 11.000
Atoms.UnitVectors>
```

直方体結晶系以外では他の格子ベクトルで定義される面に垂直な格子ベクトルに沿った非等方的な圧力を印加することになります。この状況は六方晶系の例で見られ、このキーワードを以下の様に指定します。

```
MD.applied.pressure.flag  0 0 1  # default=1 1 1
```

```
<Atoms.UnitVectors
 6.000  0.000  0.000
-3.000  5.196  0.000
 0.000  0.000 10.000
Atoms.UnitVectors>
```

## 17 分子動力学

OpenMX Ver 3.9 では次の 5 種類の分子動力学シミュレーションが実行可能です。

- 定エネルギーの分子動力学 (NVE)
- 速度スケーリングによる定温 (NVT) 分子動力学 (NVT\_VS)
- 各原子に対して独立に速度スケーリングを考慮した定温 (NVT) 分子動力学 (NVT\_VS2)
- 能勢-Hoover 法による定温 (NVT) 分子動力学 (NVT\_NH)
- 多重熱浴分子動力学 (NVT\_VS4)

各分子動力学シミュレーションの詳細を以下に説明します。

### 17.1 定エネルギーの分子動力学 (NVE)

キーワード「MD.Type」を「NVE」と指定することで、定エネルギーの分子動力学シミュレーションが実行可能です。

```
MD.Type          NVE          # NOMD|Opt|NVE|NVT_VS|NVT_VS2|NVT_NH
```

MDの各ステップで計算される数値は、出力ファイル「*System.Name.ene*」(*System.Name*は「System.Name」を意味します)に記録されます。詳細は、「source」ディレクトリ内のファイル「iterout\_md.c」に書かれています。いくつかの数値に関して、ここに示しておきます。

```
1:  MD step
2:  MD time (fs)
14: kinetic energy of nuclear motion, Ukc (Hartree)
15:  DFT total energy, Utot (Hartree)
16:  Utot + Ukc (Hartree)
17:  Fermi energy (Hartree)
```

出力ファイル「*System.Name.ene*」中で、最初の列はMD Step (MD ステップ)、2番目の列はMD time (MD 時間) 等に対応しています。

### 17.2 速度スケーリングによる NVT 分子動力学 (NVT\_VS)

キーワード「MD.Type」を「NVT\_VS」と指定することで、速度スケーリング法 [30] による NVT アンサンブル分子動力学シミュレーションが実行可能です。

```
MD.Type          NVT_VS      # NOMD|Opt|NVE|NVT_VS|NVT_VS2|NVT_NH
```

この NVT 分子動力学では原子運動の温度は、以下の書式で制御することができます。

```
<MD.TempControl
3
100  2  1000.0  0.0
400 10  700.0  0.4
```

```
700 40 500.0 0.7
MD.TempControl>
```

この書式の最初の行は「<MD.TempControl」であり、最後の行は「MD.TempControl>」です。最初の「3」は温度制御のために必要な行の数です。この場合には、引き続き3行で温度の指定を行っており、最初の列の数字はMDのステップ数を表し、2列目の数字は速度スケールが行われるMDステップの間隔を与えます。この計算例では、速度スケールは、MDステップが100回までは2回毎に行われ、MDステップの100回目から400回目までは10回ごとに行われ、さらにMDステップ400回から700回では40回ごとに行われます。3番目と4番目の列は、それぞれ設定温度  $T_{\text{give}}$  (K) とその区間でのスケール・パラメータ  $\alpha$  を表します。またその区間中における温度は線形補間によって与えられません。速度は次式を用いてスケールされます。

$$s = \sqrt{\frac{T_{\text{given}} + (T_{\text{calc}} - T_{\text{given}}) * \alpha}{T_{\text{calc}}}}$$

$$\mathbf{v}'_i = \mathbf{v}_i \times s$$

ここで、 $T_{\text{given}}$  および  $T_{\text{calc}}$  はそれぞれ設定温度と原子の運動から計算された温度です。「NVT\_VS」法では、全原子の速度を使って温度を計算します。一方、「NVT\_VS2」では、各原子の速度を用いて局所温度が評価され、速度スケールをその局所温度に基づき行います。「MD.TempControl」で指定された最後のMDステップが終了すると、NVT アンサンブルはNVE アンサンブルに切り替わります。各MDステップで計算された数値は、出力ファイル「*System.Name.ene*」（「*System.Name*」は「System.Name」を意味します）に記録されます。詳細は、「source」ディレクトリ内のファイル「iterout\_md.c」に書かれていますが、いくつかの数値に関して、ここに示しておきます。

```
1: MD step
2: MD time (fs)
14: kinetic energy of nuclear motion, Ukc (Hartree)
15: DFT total energy, Utot (Hartree)
16: Utot + Ukc (Hartree)
17: Fermi energy (Hartree)
18: Given temperature for nuclear motion (K)
19: Calculated temperature for nuclear motion (K)
22: Nose-Hoover Hamiltonian (Hartree)
```

出力ファイル「\*.ene」中で、最初の列はMD Step (MD ステップ)、2番目の列はMD time (MD 時間) 等に対応しています。例として、図 12 (a) にグリシン分子の速度スケールによるMD計算の結果を示します。分子の温度が設定温度の周辺で振動していることが分かります。分子動力学シミュレーションの可視化にはOpenMX Viewer [152, 151] やXCrySDen [105] が利用できます。これらのソフトウェアを用いて、出力ファイル「*System.Name.md*」を簡単にアニメーションとして可視化できます。

### 17.3 Nose-Hoover 法による NVT 分子動力学 (NVT\_NH)

キーワード「MD.Type」を「NVT\_NH」と指定することで、能勢-Hoover 法 [31] による NVT アンサンブル分子動力学シミュレーションが実行可能です。

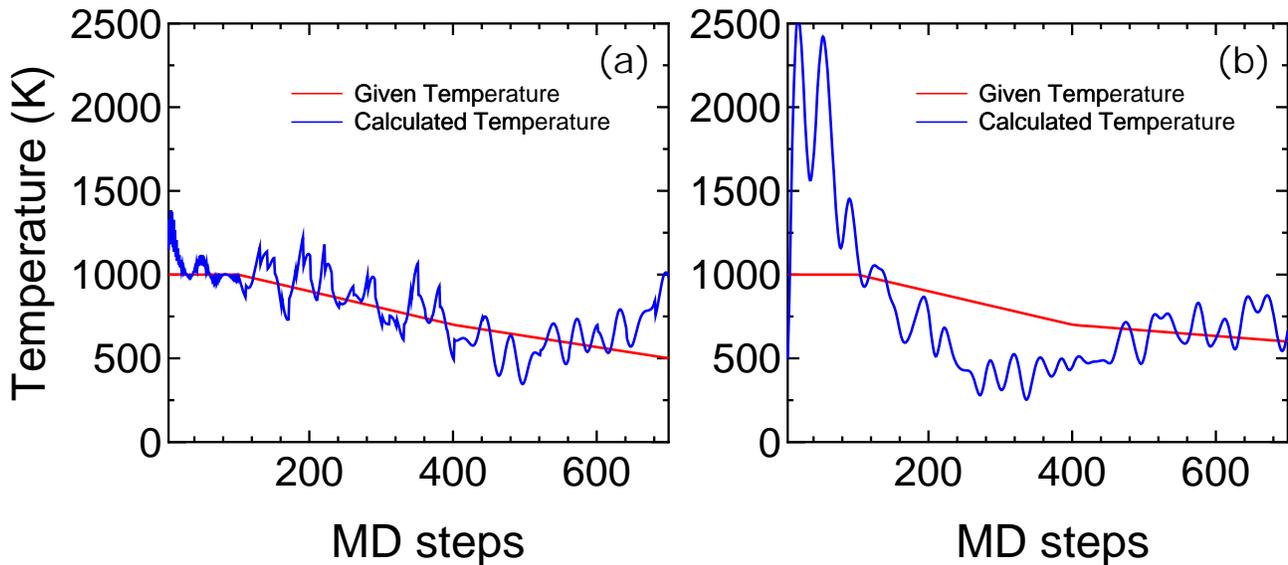


Figure 12: (a) 左図: 速度スケーリング NVT 分子動力学によるグリシン分子の設定温度と計算温度の時間変化。(b) 右図: Nose-Hoover NVT 分子動力学によるグリシン分子の設定温度と計算温度の時間変化。入力ファイルは、「work」ディレクトリ内の「Gly\_VS.dat」と「Gly\_NH.dat」。

```
MD.Type          NVT_NH      # NOMD|Opt|NVE|NVT_VS|NVT_NH
```

この NVT 分子動力学では原子運動の温度は、以下の書式で制御することができます。

```
<MD.TempControl
4
1  1000.0
100 1000.0
400  700.0
700  600.0
MD.TempControl>
```

この書式の最初の行は「<MD.TempControl」であり、最後の行は「MD.TempControl>」です。最初の「4」は温度制御のために必要な行の数です。この場合には、引き続き 4 行で温度の指定を行っており、最初の列の数字は MD のステップ数を表し、2 列目は原子運動の設定温度を表します。MD ステップの間の温度は線形補間により与えられます。能勢-Hoover 法 [31] による NVT アンサンブル分子動力学シミュレーションにおいても、速度スケーリング MD と同じキーワード「MD.TempControl」が使用されていますが、書式が異なることに注意して下さい。「MD.TempControl」の指定の他に、次のキーワードを用いて、熱浴の質量を指定して下さい。

```
NH.Mass.HeatBath      30.0      # default = 20.0
```

次元は長さ<sup>2</sup> × 質量です。この表記では長さにはボーア半径、質量に統一原子質量単位（炭素原子の主要同位体の質量を 12.0 とする単位）を用います。

「速度スケーリングによる NVT 分子動力学」で説明したように、MD ステップの各段階において計算

された数値は、出力ファイル名「*System.Name.ene*」に保存されます。例として、Nose-Hoover 法によるグリシン分子の計算結果を図 12 (b) に示します。原子運動の温度が設定温度の周辺で振動していることがわかります。また、分子力学の様子を可視化するには、NVT\_VS の場合と同様に、フリーソフト OpenMX Viewer [152, 151] や XCrySDen [105] を用いて、出力ファイル「*System.Name.md*」を簡単にアニメーション化できます。

#### 17.4 多重熱浴分子力学 (NVT\_VS)

OpenMX Ver. 3.9 では多重熱浴分子力学計算 (multi-heat bath molecular dynamics simulation) が実行可能です。グループ化した各原子の温度は速度スケール法 [30] による熱浴で制御します。この方法は次のキーワードで実行します。

```
MD.Type          NVT_VS4
```

グループ数はで指定し、

```
MD.num.AtomGroup  2
```

グループは次のようにで定義されます。

```
<MD.AtomGroup
  1  1
  2  1
  3  1
  4  2
  5  2
MD.AtomGroup>
```

最初の行は「<MD.AtomGroup」、最後の行は「MD.AtomGroup>」です。最初の列は原子を識別するための通し番号で、「Atoms.SpeciesAndCoordinates」で指定された番号に対応します。2 列目は各原子が属するグループを表す番号です。この番号は 1 から始まり、2、3、... という順序で指定しなければなりません。上の例は、系が 5 つの原子のみから構成され、2 つのグループに分けた場合です。Ver. 3.9 では、全グループの温度プロファイルは「速度スケール法による NVT 分子力学」の章で述べたように、「MD.Temp.Control」というキーワードで制御します。将来的には、グループ毎に独立に温度が制御できるような機能も導入する予定です。

#### 17.5 制約条件付き分子力学

原子座標を初期位置に固定して分子力学計算を行うための制約 MD シミュレーションが実行可能です。「制約条件付の構造最適化」の章で説明した方法で設定します。指定方法はそちらの章を参照して下さい。

#### 17.6 初速度

次のキーワードにより、分子力学計算において、各原子の初速度を指定することが可能です。

```
<MD.Init.Velocity
 1   3000.000  0.0  0.0
 2  -3000.000  0.0  0.0
MD.Init.Velocity>
```

この例は、2個の原子から構成される系の場合です。N個の原子の系の場合は、N行の指定行が必要となります。最初の列は、キーワード「Atoms.SpeciesAndCoordinates」の指定と同じ原子の通し番号です。2列目、3列目、4列目の数値は、それぞれ、各原子の初期速度のx、y、z成分です。速度の単位はm/sです。キーワード「MD.Init.Velocity」は「MD.Fixed.XYZ」と併用することが可能です。

## 17.7 ユーザーによる原子の質量の定義

OpenMXの分子動力学計算において、原子質量は「SetAtomWeight() of SetPara\_DFT.c」で定義されています。しかし、キーワード「Definition.of.Atomic.Species」を用いて、簡単に原子質量を変更することができます。その場合、以下の様に4列目の数値で原子質量を定義します。

```
<Definition.of.Atomic.Species
H   H5.0-s1          H_PBE19      2.0
C   C5.0-s1p1       C_PBE19     12.0
Definition.of.Atomic.Species>
```

4列目の数値が明示的に与えられない際には、デフォルトの原子質量が使用されます。この方法は分子動力学計算で原子質量の影響を調べる際に有用でしょう。また、とりわけ水素原子に対して重水素の質量を割り当てることで、より時間ステップを大きくすることができるようになります。原子質量の定義には、炭素原子の主同位体を12.0とする統一原子質量単位を用いています。

## 17.8 ファイルフォーマットの変換: md2axsf

分子動力学法や構造最適化では、「System.Name.md」に各ステップの構造が保存されます。「System.Name.md」はxyz形式のため、OpenMX Viewer [152, 151] や XCrySDen [105] で読み込むことも可能です。ただしXCrySDenでは周期的に拡張された格子を描画することができません (OpenMX Viewerでは「System.Name.md」に対して周期的に拡張された格子を描画できます)。周期的に拡張された格子を描画する際には、「xyz」から「axsf」形式への変換を以下の手順で行って下さい。まずmd2axsfをコンパイルします。

```
% gcc md2axsf.c -lm -o md2axsf
```

次に、「System.Name.md」の変換を行います。

```
% ./md2axsf System.Name.md System.Name.axsf
```

生成された「System.Name.axsf」ファイルはXCrySDenなどのソフトで可視化が可能です。

## 18 可視化

電子密度、分子軌道、ポテンシャル等のデータは Gaussian cube 形式でファイルに出力されます。図 13 に XcrySDen [105] を用いて可視化した等値面マップの例を示します。これらのデータは、Gaussian cube 形式で出力されているため、VESTA [103]、Molekel [104] や XCrySDen [105] 等の多くのソフトウェアで簡単に可視化できます。

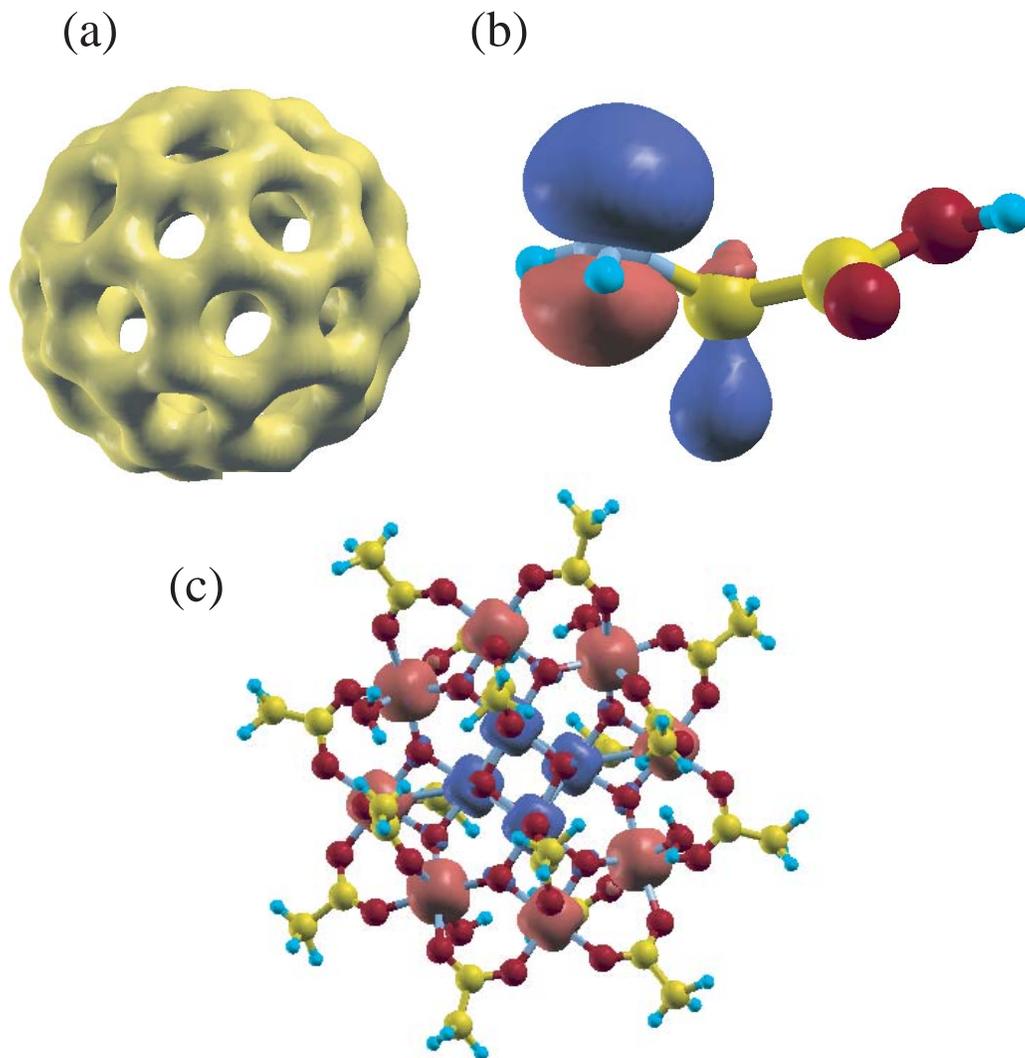


Figure 13: (a) C<sub>60</sub> 分子の全電子密度の等値面 (0.13 の等値を使用)。 (b) グリシン分子の最高被占軌道 (HOMO) の等値面 ( $|0.06|$  の等値を使用)。 (c) 分子磁石 ( $\text{Mn}_{12}\text{O}_{12}(\text{CH}_3\text{COO})_{16}(\text{H}_2\text{O})_4$  [106]) のスピン電子密度の等値面 ( $|0.02|$  の等値を使用)。

## 19 バンド分散

バンド分散は次の2つの段階を経由して、計算されます。

### (1) SCF 計算

ダイヤモンド構造の炭素を例として、バンド分散の計算方法を説明します。「work」ディレクトリ内のファイル「Cdia.dat」には、原子の座標、単位胞ベクトルおよび「scf.Kgrid」が次のように指定されています。

```
Atoms.Number          2
Atoms.SpeciesAndCoordinates.Unit  Ang # Ang|AU
<Atoms.SpeciesAndCoordinates
  1  C  0.000  0.000  0.000  2.0 2.0
  2  C  0.890  0.890  0.890  2.0 2.0
Atoms.SpeciesAndCoordinates>
Atoms.UnitVectors.Unit          Ang # Ang|AU
<Atoms.UnitVectors
  1.7800  1.7800  0.0000
  1.7800  0.0000  1.7800
  0.0000  1.7800  1.7800
Atoms.UnitVectors>

scf.Kgrid                7 7 7          # means n1 x n2 x n3
```

バンド分散の単位胞ベクトルとバンド分散を計算するためのk点の経路は次の書式で与えられます。

```
Band.dispersion          on          # on|off, default=off
<Band.KPath.UnitCell
  3.56  0.00  0.00
  0.00  3.56  0.00
  0.00  0.00  3.56
Band.KPath.UnitCell>
Band.Nkpath              5
<Band.kpath
  15  0.0 0.0 0.0  1.0 0.0 0.0  g X
  15  1.0 0.0 0.0  1.0 0.5 0.0  X W
  15  1.0 0.5 0.0  0.5 0.5 0.5  W L
  15  0.5 0.5 0.5  0.0 0.0 0.0  L g
  15  0.0 0.0 0.0  1.0 0.0 0.0  g X
Band.kpath>
```

次に OpenMX を以下の様に、通常実行します。

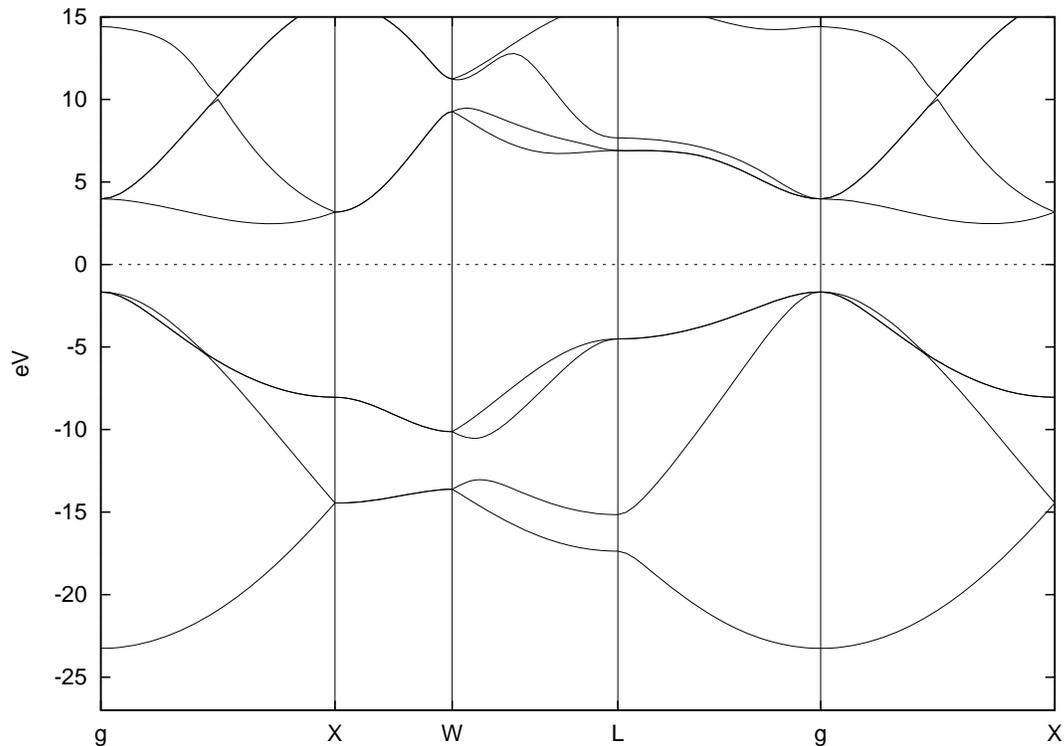


Figure 14: 炭素ダイヤモンドのバンド分散。入力ファイルは「work」ディレクトリ内の「Cdia.dat」。

```
% ./openmx Cdia.dat
```

ここでは単一コアを使用した計算例を示しましたが、もちろん並列計算も実行可能です。この実行が正常に終了すると、「work」ディレクトリに「cdia.Band」というファイルが生成されます。キーワード「Band.KPath.UnitCell」が存在する場合にはこの単位胞ベクトルを用いて k 点経路の指定に用いられる逆格子ベクトル基底が計算されます。もしこのキーワード「Band.KPath.UnitCell」が指定されない場合には、「Atoms.UnitVectors」による単位胞ベクトルを用いて逆格子ベクトルを計算し、その目的に使用されます。

## (2) データを gnuplot 形式に変換する

「source」ディレクトリに「bandgnu13.c」というファイルがあります。このファイルを次のようにしてコンパイルします。

```
% gcc bandgnu13.c -lm -o bandgnu13
```

コンパイルが正常に終了すると、「source」ディレクトリに実行可能なファイル「bandgnu13」が生成されます。このファイルを「work」ディレクトリにコピーして下さい。実行ファイル「bandgnu13」を用いて、次のコマンドを実行することにより、ファイル「cdia.Band」は gnuplot 形式に変換されます。

```
% ./bandgnu13 cdia.Band
```

これで、「cdia.GNUBAND」、「cdia.BANDDAT1」(および「cdia.BANDDAT2」)の2または3個のファイルが生成されます。ファイル「cdia.GNUBAND」はgnuplotのスク립トであり、アップとダウンのスピ状態のデータファイル(それぞれ「cdia.BANDDAT1」および「cdia.BANDDAT2」)を読み込むためのファイルです。もし「LSDA-CA」、「LSDA-PW」、「GGA-PBE」のいずれかを用いてスピ分極計算を行った場合には、「\*.BANDDAT1」に加えてダウンスピ状態の「*System.Name*.BANDDAT2」が生成されます。ファイル「cdia.GNUBAND」は次のコマンドにより、gnuplotを使ってプロットすることができます。

```
% gnuplot cdia.GNUBAND
```

図 14 はここで説明した方法で計算した炭素ダイヤモンドのバンド分散を示しています。ただし「cdia.GNUBAND」内の y 軸の範囲は変更されています。化学ポテンシャルがエネルギーの原点となるように自動的に調整されていることに注意してください。

バンド分散を描く際の問題点はバンド分散の計算で使われる単位胞の選択法です。しばしば、SCF 計算で使用される単位胞と異なる単位胞を用いてバンド分散の計算における特殊 k 点の指定が行われます。このような場合は、バンド分散の計算で用いる単位胞をキーワード「Band.KPath.UnitCell」で定義する必要があります。「Band.KPath.UnitCell」を定義するとバンド分散の計算の逆格子ベクトルは「Band.KPath.UnitCell」で指定された格子ベクトルから計算されます。「Band.KPath.UnitCell」を定義しなければ、逆格子ベクトルは「Atoms.UnitVectors」で指定された格子ベクトルで計算され、バンド分散の計算に用いられます。体心、面心、底心、三方晶の場合はバンド分散の計算のための逆格子ベクトルはバンド構造計算の慣例に従いキーワード「Band.KPath.UnitCell」を用いて指定するべきです。

## 20 状態密度

### 20.1 通常の方法

Kohn-Sham 固有値の状態密度 (DOS) は次の 2 つの段階を経由して、計算されます。

#### (1) SCF 計算

ダイヤモンド構造の炭素を例として、DOS の計算方法を説明します。「work」ディレクトリ内のファイル「Cdia.dat」には、次の様に DOS 計算のためのキーワードが指定されています。

```
Dos.fileout          on
Dos.Erange          -25.0 20.0
Dos.Kgrid           12 12 12
```

キーワード「Dos.Erange」で指定された最初と 2 番目の数値は、それぞれ DOS 計算のためのエネルギー範囲 (eV) の下限と上限であり、エネルギーの原点 (0.0) は化学ポテンシャルに対応します。またキーワード「Dos.Kgrid」で指定された数値 (n1,n2,n3) は、DOS 計算において第 1 ブリルアンゾーンを離散化するためのグリッド数です。

次に OpenMX を以下の様に、通常実行します。

```
% ./openmx Cdia.dat
```

ここでは単一コアを使用した計算例を示しましたが、もちろん並列計算も実行可能です。計算が正常に終了すると、「work」ディレクトリに「cdia.Dos.val」および「cdia.Dos.vec」の 2 つのファイルが生成されます。「cdia.Dos.val」にはテキスト形式で固有値が、「cdia.Dos.vec」にはバイナリ形式で固有ベクトルが保存されています。この DOS 計算は、 $O(N)$  計算にも対応しており、この場合にはガウシアンブロードニング法 (Gaussian broadening method) が適用されます。

#### (2) DOS 計算

DOS 計算用のユーティリティプログラムをコンパイルして下さい。「source」ディレクトリに移動して、次のコマンドでコンパイルします。

```
% make DosMain
```

コンパイルが正常に終了すると、「source」ディレクトリ内に実行ファイル「DosMain」が生成されます。「DosMain」を「work」ディレクトリにコピーして、「work」ディレクトリに移動して下さい。このプログラム「DosMain」を使って先の 2 つのファイル「cdia.Dos.val」および「cdia.Dos.vec」から全状態密度 (DOS) と射影 DOS (PDOS) を以下の手順で計算します。

```
% ./DosMain cdia.Dos.val cdia.Dos.vec
```

その際に、プログラムから対話形式で次のように質問されることでしょう。

```
% ./DosMain cdia.Dos.val cdia.Dos.vec
Max of Spe_Total_CNO = 8
1 1 101 102 103 101 102 103
```

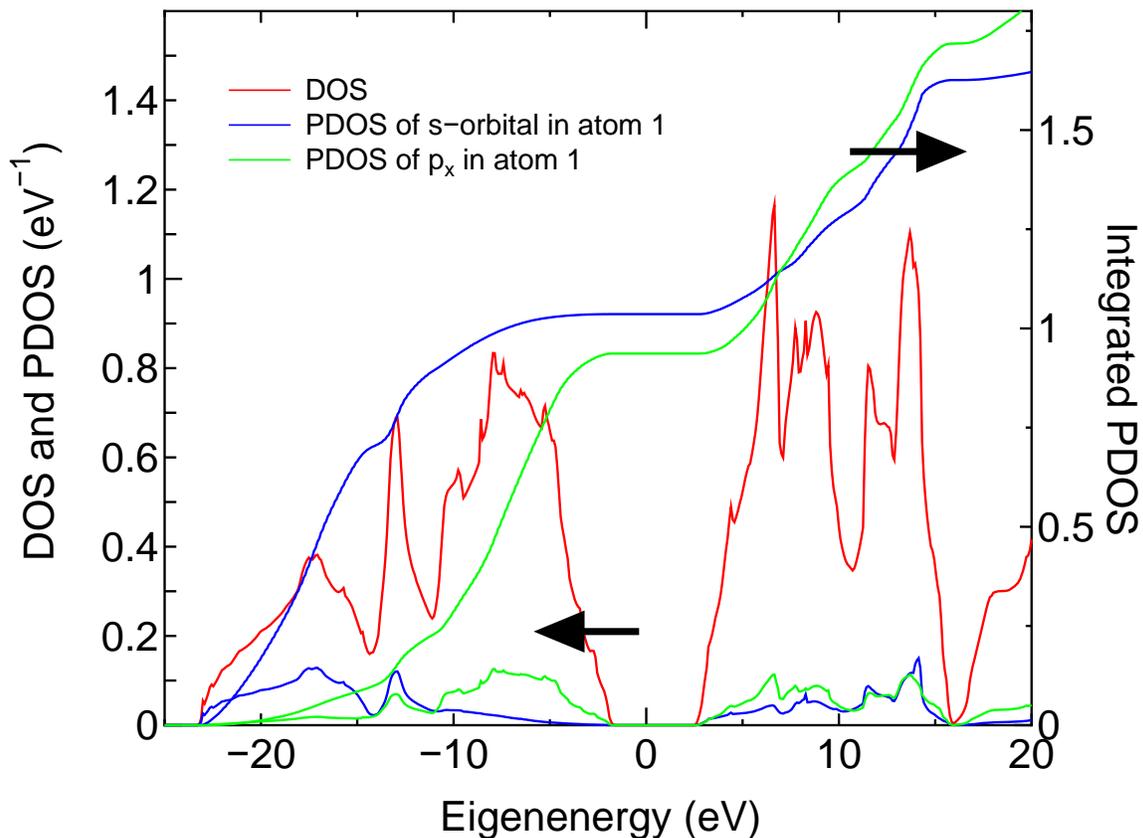


Figure 15: ダイヤモンド炭素の全状態密度 (DOS) と部分状態密度 (PDOS)、また PDOS の積分曲線。フェルミ準位をゼロに設定。s、p、d 軌道の間で電荷の再配分が起っているため、s および p 軌道のフェルミ準位までの PDOS の積分値は正確には 1 ではないことに注意。「work」ディレクトリにある入力ファイル「Cdia.dat」を用いて同じ計算が実行できる。

```

<cdia.Dos.val>
<cdia>
Which method do you use?, Tetrahedron(1), Gaussian Broadening(2)
1
Do you want Dos(1) or PDos(2)?
2

Number of atoms=2
Which atoms for PDOS : (1,...,2), ex 1 2
1
pdos_n=1
1
<Spectra_Tetrahedron> start
Spe_Num_Relation 0 0 1
Spe_Num_Relation 0 1 1
Spe_Num_Relation 0 2 101

```

```

Spe_Num_Relation 0 3 102
Spe_Num_Relation 0 4 103
Spe_Num_Relation 0 5 101
Spe_Num_Relation 0 6 102
Spe_Num_Relation 0 7 103
make cdia.PDOS.Tetrahedron.atom1.s1
make cdia.PDOS.Tetrahedron.atom1.p1
make cdia.PDOS.Tetrahedron.atom1.p2
make cdia.PDOS.Tetrahedron.atom1.p3
make cdia.PDOS.Tetrahedron.atom1

```

DOS の計算には四面体法 (tetrahedron method) [67] もしくはガウシアンブロードニング法 (Gaussian broadening method) が選択できます。またユーザーは DOS または PDOS を選ぶことができます。PDOS の計算を選択する場合は、PDOS を評価する原子を選択して下さい。この場合には、選択した原子の軌道 ( $s$ ,  $px(p1)$ ,  $py(p2)$ ,  $pz(p3)$ ,...) 上に射影した PDOS が各ファイルに出力されます。これらのファイルでは、最初と二番目の列は、エネルギー (eV) および DOS ( $\text{eV}^{-1}$ ) または PDOS ( $\text{eV}^{-1}$ ) の値であり、3 番目の列の数値は DOS または PDOS の積分値を示します。「LSDA-CA」、**「LSDA-PW」**、または「GGA-PBE」を用いたスピン分極計算を行った場合、これらのファイルの第 2 列および第 3 列の数値は、それぞれアップスピン状態およびダウンスピン状態の DOS と PDOS に対応し、第 4 列と第 5 列の数値はそれぞれに対応する積分値です。ガウシアンブロードニング法を使用した場合、ガウス分布のパラメーター値、 $a$  (eV) ( $\exp(-(E/a)^2)$  で定義されるガウス分布の幅を決めるパラメーター) を設定しなければなりません。ここで説明した手続きによって得られたダイヤモンド炭素の DOS および PDOS を図 15 に示します。

## 20.2 多数の k 点で計算する場合

多数の k 点を伴う大規模系の状態密度 (DOS) 計算では大容量サイズのメモリーが必要になり、「DosMain」を用いた後処理計算において、メモリ不足のために計算が異常終了してしまうことがあります。このような場合、on-the-fly のガウシアンブロードニング法が利用できます。この方法では、OpenMX の DOS 計算の際に、on-the-fly でガウシアンブロードニング法が適用され、DOS の計算が行われます。大きなデータ量となる波動関数の情報はファイル「*System.Name.DOS.vec*」に記録されません。この方法では、大容量のメモリーを必要としないので、大規模な系の状態密度計算が可能です。本方法を用いる場合には、入力ファイルに次の様にキーワードを指定します。

```

DosGauss.fileout      on          # default=off, on|off
DosGauss.Num.Mesh     200         # default=200
DosGauss.Width        0.2         # default=0.2 (eV)

```

この方法を使用する場合、キーワード「DosGauss.fileout」を「on」と指定します。そして、キーワード「DosGauss.Num.Mesh」では、キーワード「Dos.Erange」で指定したエネルギー範囲の分割数を指定します。キーワード「DosGauss.width」では、ガウス分布の  $\exp(-(E/a)^2)$  を定義する幅  $a$  を指定します。キーワード「DosGauss.fileout」と「Dos.fileout」は排他的で併用することはできませんので、注意して下さい。従って、このキーワードを使用する場合には、「Dos.fileout」は、次の様に「off」に設定します。

```

Dos.fileout           off         # on|off, default=off

```

また「Dos.fileout」および「DosGauss.file」の両方のキーワードに対して次の2つのキーワードが有効です。

```
Dos.Erange      -20.0  20.0  # default=-20 20
Dos.Kgrid       5 5 5    # default=Kgrid1 Kgrid2 Kgrid3
```

キーワード「DosGauss.fileout」では、ガウシアンブロードニング法によるDOSだけしか計算できません。四面体法によるDOSは計算できませんので、注意して下さい。「DosGauss.fileout」を用いた場合でも、「DosMain」による後処理の手続きは、四面体法によるDOSが計算不可であること以外は、これまで説明した方法と同様です。

## 21 全エネルギーの軌道分解

OpenMX Ver. 3.9 では、全エネルギーを各基底関数の寄与に分解することが可能です。これは、基底関数への射影によって行われます [68]。全エネルギーの軌道分解は、以下のキーワードを加えるだけで実施可能です。

```
Energy.Decomposition      on          # on|off, default=off
```

例として、メタン分子の全エネルギーの軌道分解を以下に示します。

```
% mpirun -np 5 openmx Methane_ED.dat > met_ed.std &
```

ここで使用している「Methane\_ED.dat」は「work」ディレクトリに収められています。計算が終わると「met\_ed.out」が得られ、以下の様な情報が書き出されます。

```
*****
*****
Decomposed energies in Hartree unit

Utot = Utot(up) + Utot(dn)
      = Ukin(up) + Ukin(dn) + Uv(up) + Uv(dn)
      + Ucon(up)+ Ucon(dn) + Ucore+UHO + Uvdw

Uele = Ukin(up) + Ukin(dn) + Uv(up) + Uv(dn)
Ucon arises from a constant potential added in the formalism

      up: up spin state, dn: down spin state
*****
*****

Total energy (Hartree) = -8.216132481346387

Decomposed.energies.(Hartree).with.respect.to.atom

      Utot      Utot(up)      Utot(dn)      Ukin(up)      Ukin(dn)      Uv(up)      ....
1   C      -6.132295762765    -3.066148    -3.066148    2.076016    2.076016    -2.957459    ....
2   H      -0.520959186503    -0.260480    -0.260480    0.300675    0.300675    -0.499086    ....
3   H      -0.520959174111    -0.260480    -0.260480    0.300675    0.300675    -0.499086    ....
4   H      -0.520959173764    -0.260480    -0.260480    0.300675    0.300675    -0.499086    ....
5   H      -0.520959184204    -0.260480    -0.260480    0.300675    0.300675    -0.499086    ....

Decomposed.energies.(Hartree).with.respect.to.atomic.orbital

      1   C      Utot      Utot(up)      Utot(dn)      Ukin(up)      Ukin(dn)      Uv(up)
multiple
none      -4.483770    -2.241885    -2.241885    0.000000    0.000000    0.000000    ....
s         0    -0.675699    -0.337849    -0.337849    0.203145    0.203145    -0.556473    ....
s         1     0.003690    0.001845    0.001845    0.036240    0.036240    -0.034310    ....
px        0    -0.325884    -0.162942    -0.162942    0.496144    0.496144    -0.673031    ....
py        0    -0.325912    -0.162956    -0.162956    0.496166    0.496166    -0.673068    ....
pz        0    -0.325884    -0.162942    -0.162942    0.496144    0.496144    -0.673031    ....
px        1     0.005096    0.002548    0.002548    0.107318    0.107318    -0.104552    ....
...
.
.
```

全エネルギーが各原子の各軌道からの寄与に分解されているのが分かります。不純物原子や空孔による局所的な摂動が隣接原子の安定性/不安定性にどのように影響するかを議論する際に、本機能は有用です。また、本機能は幅広い現象に対する物理的な描像を解析するのに役立つものと期待されます。しかしながら、この機能の公開はまだ試験的なものです。我々はこの機能を近い将来にさらに発展させることを考えています。

## 22 軌道の最適化

軌道最適化法 [41] を用いて基底の動径関数を変分的に最適化することができます。メタン分子 (入力ファイル「Methane\_OO.dat」) を例として、軌道最適化の手順を説明します。軌道最適化法では、最適化動径関数は、プリミティブ動径関数の線形結合で表されます。線形結合の係数は縮約係数と呼ばれ、この縮約係数が変分原理に基づき最適化されます。軌道最適化法でのプリミティブ動径関数と最適化動径関数の数は、次の様に指定します。

```
<Definition.of.Atomic.Species
  H   H5.0-s4>1           H_CA19
  C   C5.0-s4>1p4>1      C_CA19
Definition.of.Atomic.Species>
```

水素原子 (H) の  $s$  軌道に対しては、4つのプリミティブ動径関数の線形結合から1つの最適化動径関数が得られます。同様に、炭素原子 (C) の場合、4つのプリミティブ動径関数の線形結合から1つの  $s(p)$  軌道の最適化動径関数が得られます。さらに、以下のキーワードにより軌道最適化法が制御されます。

```
orbitalOpt.Method          species      # Off|Species|Atoms
orbitalOpt.Opt.Method      EF           # DIIS|EF
orbitalOpt.SD.step         0.001       # default=0.001
orbitalOpt.HistoryPulay    30          # default=15
orbitalOpt.StartPulay      10          # default=1
orbitalOpt.scf.maxIter     60          # default=40
orbitalOpt.Opt.maxIter     140         # default=100
orbitalOpt.per.MDIter      20          # default=1000000
orbitalOpt.criterion       1.0e-4      # default=1.0e-4

CntOrb.fileout             on          # on|off, default=off
Num.CntOrb.Atoms           2          # default=1
<Atoms.Cont.Orbitals
  1
  2
Atoms.Cont.Orbitals>
```

入力ファイル「Methane\_OO.dat」を用いて OpenMX を通常実行します。

```
% ./openmx Methane_OO.dat
```

この計算が正常に終了すると、ファイル「met\_oo.out」中に軌道最適化の履歴が記録されています。

```
*****
*****
History of orbital optimization   MD= 1
*****      Gradient Norm ((Hartree/borh)^2)      *****
Required criterion= 0.000100000000
```

\*\*\*\*\*

```
iter= 1 Gradient Norm= 0.057098961101 Uele= -3.217161102876
iter= 2 Gradient Norm= 0.044668461503 Uele= -3.220120116009
iter= 3 Gradient Norm= 0.034308306321 Uele= -3.223123238394
iter= 4 Gradient Norm= 0.025847573248 Uele= -3.226177980300
iter= 5 Gradient Norm= 0.019106400842 Uele= -3.229294858054
iter= 6 Gradient Norm= 0.013893824906 Uele= -3.232489198284
iter= 7 Gradient Norm= 0.010499500005 Uele= -3.235304178159
iter= 8 Gradient Norm= 0.008362635043 Uele= -3.237652870812
iter= 9 Gradient Norm= 0.006959703539 Uele= -3.239618540761
iter= 10 Gradient Norm= 0.005994816379 Uele= -3.241268535418
iter= 11 Gradient Norm= 0.005298095979 Uele= -3.242657118263
iter= 12 Gradient Norm= 0.003059655878 Uele= -3.250892948269
iter= 13 Gradient Norm= 0.001390201488 Uele= -3.255123241210
iter= 14 Gradient Norm= 0.000780925380 Uele= -3.255179362845
iter= 15 Gradient Norm= 0.000726631072 Uele= -3.255263012792
iter= 16 Gradient Norm= 0.000390930576 Uele= -3.250873416989
iter= 17 Gradient Norm= 0.000280785975 Uele= -3.250333677139
iter= 18 Gradient Norm= 0.000200668585 Uele= -3.252345643243
iter= 19 Gradient Norm= 0.000240367596 Uele= -3.254238199726
iter= 20 Gradient Norm= 0.000081974594 Uele= -3.258146794679
```

多くの場合、20~50回の反復計算で収束に達します。プリミティブ基底関数および最適化基底関数で計算されたメタン分子の全エネルギーを以下に示します。

```
Primitive basis orbitals
Utot = -7.992569945749 (Hartree)
```

```
Optimized orbitals by the orbital optimization
Utot = -8.133746986502 (Hartree)
```

軌道最適化によって全エネルギーは低下しており、変分的に基底関数が最適化されていることが分かります。図 16 に、プリミティブ基底および最適化基底を使って得られた分子およびバルクの全エネルギーの収束の様子を示します。ここで扱った全ての系に対して、最適化基底の収束特性が優れていることが分かります。上記の例でのメタン分子の場合、最適化された基底関数は「C\_1.pao」と「H\_2.pao」の2つのファイルに出力されます。これらのファイル「C\_1.pao」と「H\_2.pao」は擬原子基底関数の入力データとして、OpenMX の計算に対してそのまま使用できます。最適化された基底関数はファイルに出力されますので、計算しようとする系の基底関数を予め最適化しておくことが便利です。この際、軌道最適化法を適用する系としては、化学的に類似した小さな系を選択することを推奨します。

キーワード「orbitalOpt.Method」には、次の2つのオプションが用意されています。(1) それぞれの原子上の基底関数が完全に最適化される「atoms」、(2) それぞれの原子種の基底関数が最適化された「species」。

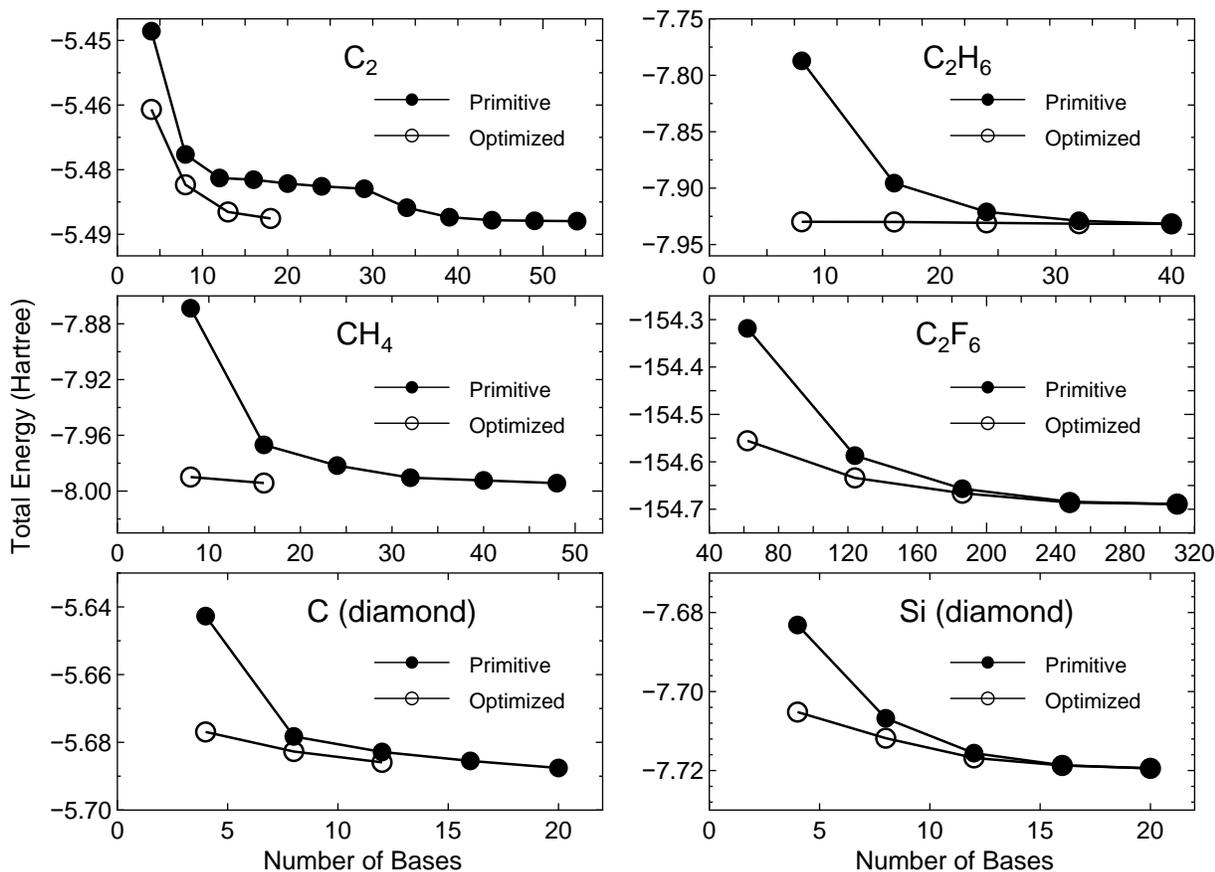


Figure 16: プリミティブ基底関数および最適化基底関数で計算された炭素 2 量体 (C<sub>2</sub>)、メタン分子 (CH<sub>4</sub>)、ダイヤモンド構造の炭素とケイ素、エタン分子 (C<sub>2</sub>H<sub>6</sub>)、ヘキサフルオロエタン (C<sub>2</sub>F<sub>6</sub>) の全エネルギー。全エネルギーと全基底数は、C<sub>2</sub>、ダイヤモンド構造の炭素およびケイ素の場合は 1 原子についての値、CH<sub>4</sub>、C<sub>2</sub>H<sub>6</sub> および C<sub>2</sub>F<sub>6</sub> の場合は 1 分子についての値。

- atoms

各原子単位で基底関数が最適化されます。動径波動関数  $R$  は磁気量子数には依存しませんので、系の回転に対する全エネルギーの不変性が保証されます。

- species

「Definition.of.Atomic.Species」で定義した同一名の原子種の基底関数は同一の軌道に最適化されます。動径波動関数  $R$  は磁気量子数には依存しませんので、系の回転に対する全エネルギーの不変性が保証されます。ほぼ同様の化学的環境を持つ原子に対して同一の最適化基底を生成したい場合に、有用な方法です。

「入力ファイル」の章でも同様な情報が記載されていますが、ユーザーの利便性のため、関連するキーワードの詳細を以下に列挙します。

#### orbitalOpt.scf.maxIter

軌道最適化における SCF 反復の最大回数を「orbitalOpt.scf.maxIter」キーワードで指定します。

#### orbitalOpt.Opt.maxIter

軌道最適化の反復の最大回数を「orbitalOpt.Opt.maxIter」キーワードで指定します。軌道最適化の反復は、収束条件が達成しなかった場合でも、同キーワードで設定した回数で終了します。

#### orbitalOpt.Opt.Method

軌道最適化の収束方法として、2つの手法がサポートされています。「EF」は固有ベクトル追跡法、「DIIS」は反復部分空間における直接反転法です。それぞれのアルゴリズムは構造最適化のそれと同じです。「orbitalOpt.Opt.Method」キーワードでは「EF」あるいは「DIIS」を指定して下さい。

#### orbitalOpt.StartPulay

「orbitalOpt.StartPulay」キーワードで指定した最適化ステップから、準ニュートン法である「EF」または「DIIS」法を開始します。

#### orbitalOpt.HistoryPulay

準ニュートン法である「EF」および「DIIS」法において、次ステップでの縮約係数を推定するために参照する過去のステップ数を「orbitalOpt.HistoryPulay」キーワードで指定します。

#### orbitalOpt.SD.step

準ニュートン法である「EF」および「DIIS」法を開始するまでの最適化ステップは最急降下法が適用されます。最急降下法で使用する前因子は「orbitalOpt.SD.step」キーワードで指定します。多くのケースにおいて、「orbitalOpt.SD.step」の適切な値は0.001程度となります。

#### orbitalOpt.criterion

軌道最適化の収束条件 ( $\text{Hartree/bohr}^2$ ) を「orbitalOpt.criterion」キーワードで指定します。「微分のノルム <orbitalOpt.criterion」という条件が満たされた時に反復ループが終了します。

#### CntOrb.fileout

最適化動径関数をファイルに出力したい場合は、「CntOrb.fileout」キーワードを「ON」にする必要があります。

#### Num.CntOrb.Atoms

最適化動径関数をファイルに出力する際の原子数を「Num.CntOrb.Atoms」キーワードで指定します。

#### Atoms.Cont.Orbitals

「Atoms.SpeciesAndCoordinates」キーワードの第1列で定義した原子の通し番号を用いて、最適化基底を出力する原子を「Atoms.Cont.Orbitals」キーワードで次のように指定します。

```
<Atoms.Cont.Orbitals
  1
  2
Atoms.Cont.Orbitals>
```

記述は「<Atoms.Cont.Orbitals」で始め、「Atoms.Cont.Orbitals>」で終わります。行の数は、「Num.CntOrb.Atoms」で記述する数字と整合性がなければなりません。例えば、最適化擬原子軌道が「C\_1.pao」と「H\_2.pao」などとして保存された場合、その原子種名はキーワード「Definition.of.Atomic.Species」での設定における第一列の記号に対応し、記号の後の数字はキーワード「Atoms.SpeciesAndCoordinates」の設定における第一列の数値を意味します。これらの出力ファイル「C\_1.pao」と「H\_2.pao」は基底関数の入力データとして使用可能です。

## 23 $O(N)$ 法

一般に行列対角化の計算時間は、使用する基底関数の数の 3 乗に比例します。そのため大規模な系の計算では、対角化が計算の律速となります。一方、近似手法である  $O(N)$  法の計算量は  $O(N)$  となり、大規模系の取り扱いが可能となります。しかし計算精度に関しては注意深い配慮が必要となります。OpenMX Ver. 3.9 では、分割統治法 (DC 法) [50] と局在自然軌道 (LNO) による分割統治法 (DC 法) [51]、 $O(N)$  Krylov 部分空間法 [43] の三つの  $O(N)$  法が利用可能です。これらの三つの  $O(N)$  法のなかでの我々の推奨は DC-LNO 法です。なぜなら、この方法は頑健で金属を含む幅広い物質に適用でき、これら三つの法のなかで並列化効率が最も高いと期待されるからです。以下の章では、それぞれの  $O(N)$  法について計算例を示しながら説明します。

### 23.1 分割統治法 (DC 法)

DC 法は数値的に安定で広範囲の物質に適用可能であり、また計算精度と計算効率を容易に制御できる方法です。本手法は特に共有結合性の物質群に適しています。この章では、DC 法を用いた  $O(N)$  計算について説明します。「work」ディレクトリ内の入力ファイル「DIA8\_DC.dat」において、キーワード「EigenvalueSolver」に対して「DC」を指定して下さい。

```
scf.EigenvalueSolver DC
```

入力ファイル「DIA8\_DC.dat」を用いて OpenMX を通常実行します。

```
% ./openmx DIA8_DC.dat
```

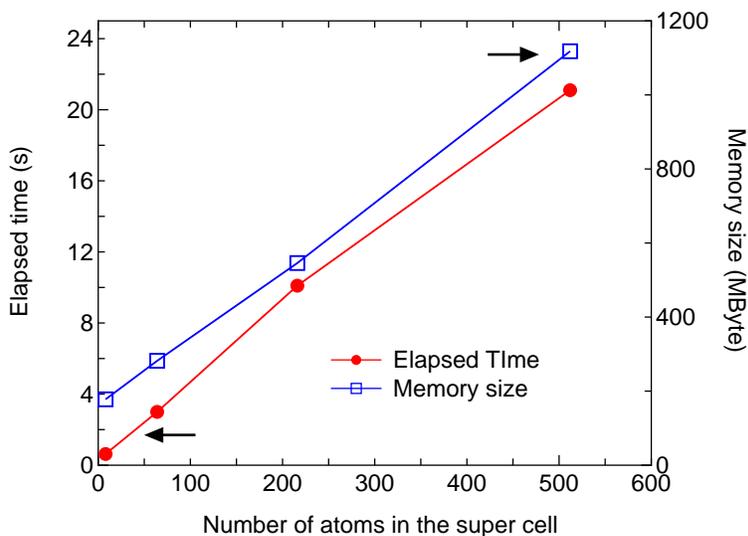


Figure 17: 分割統治法によるダイヤモンド炭素の計算時間とメモリ使用量。計算時間は 1 回の SCF ステップに対する対角化の所要時間、またメモリ使用量は 1 つの MPI プロセス当りを示す。計算には MPI 並列計算で 16 プロセスを使用。基底関数には C5.0-s1p1、DC 法では orderN.HoppingRanges=6.0 (Å) を使用。使用計算機は Xeon プロセッサ (2.6 GHz)。計算に用いた入力ファイルは「work」ディレクトリ内の「DIA8\_DC.dat」、「DIA64\_DC.dat」、「DIA216\_DC.dat」、「DIA512\_DC.dat」。

Table 4: 通常の対角化法と DC 法によって計算された  $C_{60}$  分子、小ペプチド分子 (パロルフィン (valorphin) [107]) シトシンとグアニンから構成される DNA の全エネルギーと計算時間。計算には最小基底関数を使用。DC の後のカッコ内の数値は、DC 計算に使われた「orderN.HoppingRanges」を示す。計算時間は Opteron PC クラスタ (48 MPI プロセス, 2.4 GHz) 用いて測定。入力ファイルは、「work」ディレクトリ内の「C60\_DC.dat」, 「Valorphin\_DC.dat」, 「CG15c\_DC.dat」。

|                            | Total energy (Hartree) | Computational time (s) |
|----------------------------|------------------------|------------------------|
| <b><math>C_{60}</math></b> |                        |                        |
| (60 atoms, 240 orbitals)   |                        |                        |
| Conventional               | -343.89680             | 36                     |
| DC (7.0)                   | -343.89555             | 37                     |
| <b>Valorphin</b>           |                        |                        |
| (125 atoms, 317 orbitals)  |                        |                        |
| Conventional               | -555.28953             | 81                     |
| DC (6.5)                   | -555.29019             | 76                     |
| <b>DNA</b>                 |                        |                        |
| (650 atoms, 1880 orbitals) |                        |                        |
| Conventional               | -4090.95463            | 576                    |
| DC (6.3)                   | -4090.95092            | 415                    |

この入力ファイルは 8 個の炭素原子を含むダイヤモンド格子の DC 計算を行うためのものです。構造最適化は行いません。計算時間は Xeon プロセッサ (2.6 GHz) を使用した場合におよそ 120 秒です。図 17 に、分割統治法によるダイヤモンド炭素の計算時間と計算メモリ量をスーパーセル内の炭素原子数の関数として示します。計算時間は 1 回の SCF ステップに対する対角化の所要時間、またメモリ使用量は 1 つの MPI プロセス当りを示します。図から、計算時間とメモリ使用量は原子数にほぼ比例することが分かります。DC 法の計算精度と計算効率はキーワード「orderN.HoppingRanges」で制御されます。

- orderN.HoppingRanges

各原子を中心とする球の半径 ( $\text{\AA}$ ) を「orderN.HoppingRanges」キーワードで定義します。DC 法、DC-LNO 法、および  $O(N)$  クリロフ部分空間法では、この球内に含まれる原子を選択することで切り取られたクラスターが構成されます。

系の原子数を  $N$  とすると、「orderN.HoppingRanges」で指定された半径の球で物理的に切り取られたクラスターが  $N$  個、構築されます。それぞれのクラスターに対して、独立に固有値問題を解き、中心原子への射影状態密度を計算します。その後、全ての原子からの射影状態密度を足し合わせることで、全系の状態密度が計算されます。「orderN.HoppingRanges」の適切な値は系によって異なりますが、分子系では計算精度と計算効率の妥協点として次の値が推奨されます。

orderN.HoppingRanges      6.0 - 7.0      # in Ang.

表 4 に、 $C_{60}$  分子と小ペプチド分子 (パロルフィン (valorphin) [107]) およびシトシンとグアニンから構成される DNA について、通常の対角化法と DC 法で計算した全エネルギーの比較を示します。DC 法で計算した全エネルギーの誤差は、どの系でも 1 原子当りミリハートリー程度であることが分かります。また本計算で用いた計算条件では原子数が 500 以上になると DC 法が通常の対角化法より高速となるこ

とが推定されますが、計算時間における通常対角化法と DC 法の交差点は、系と並列計算で使用するコア数に依存しています。

切り取られたクラスタのサイズに対する全エネルギーの精度の依存性を見るため、通常対角化と比較した全エネルギーの誤差を図 18 に示します。ここでは切り取られたクラスタの原子数をパラメーターとして (a) 有限ギャップを持った固体、(b) 金属、(c) 分子系の 3 つの場合を取り上げました。これから分かることは、(a) 有限ギャップを持った固体と (c) 分子系の場合にはその誤差がほぼ指数関数的に減少し、(b) 金属の場合には収束速度が相対的に遅いということです。

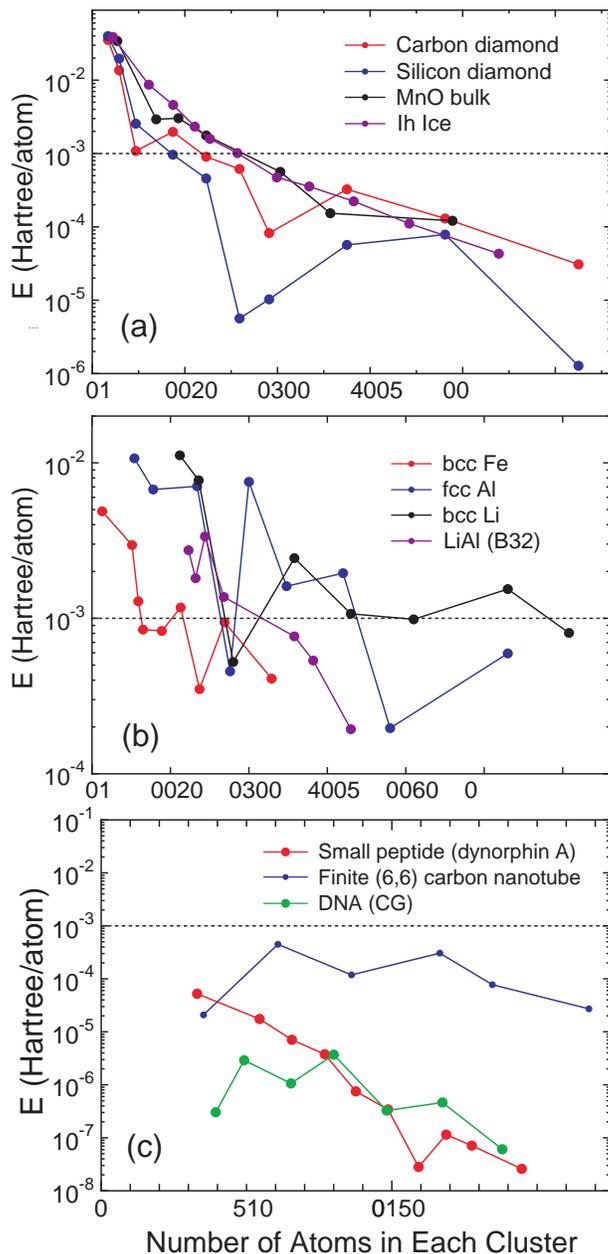


Figure 18: DC 法で計算された (a) 有限ギャップを持った固体、(b) 金属、(c) 分子系の全エネルギーにおける誤差。横軸は各切り取られたクラスタ内の原子数。水平方向の点線は「ミリハートリー (mili-Hartree)」の精度を示す。

## 23.2 局在自然軌道による分割統治法 (DC-LNO)

DC-LNO 法 [51] は DC 法の改良版です。占有状態の部分空間を張る局在自然軌道 (LNO) を導入することによって、計算効率が改善されます。本来の基底関数である PAO と比較して LNO の次元が低減されているため、結果として計算コストが削減されます。占有状態への射影演算子を局所固有値分解を用いて低ランク近似することで、LNO が生成されます。この LNO 生成する計算は逐次的でないため、非常に効率的に実行可能です。図 19(a) に示すように、切られたクラスター (truncated cluster) 内の遠方領域の原子に対して LNO が割り当てられ、また近傍では PAO を割り当てられます。このため計算精度を保ちながら、計算効率を向上させることが可能となります。本手法はギャップのある系のみならず、切られたクラスターが十分に大きければ (典型的には 200 原子以上を含むクラスター)、金属系にも適用可能です。

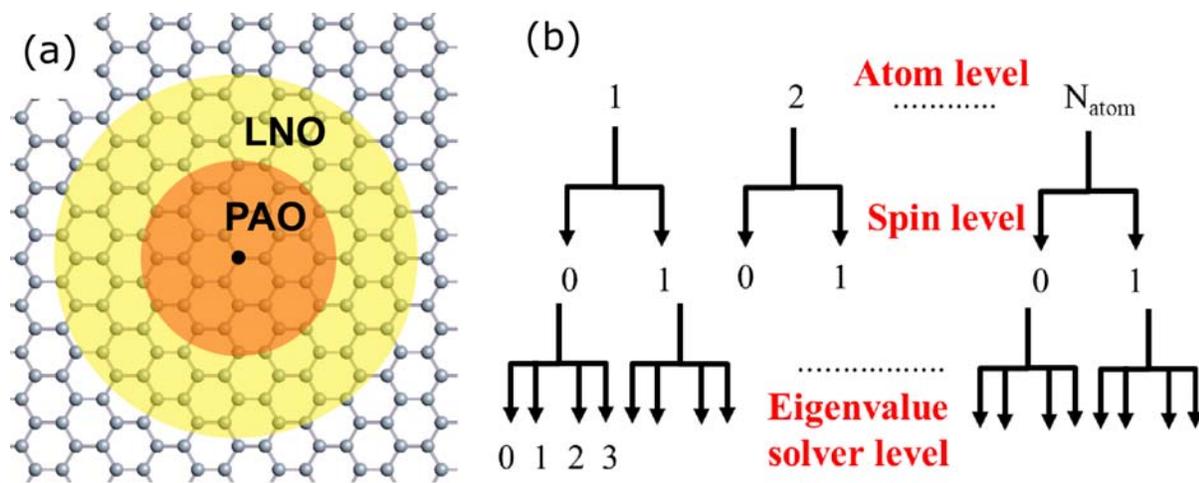


Figure 19: (a) DC-LNO 法における切られたクラスター (truncated cluster) の構築。中心原子からの近傍 (オレンジ色) と遠方の範囲 (黄色) の領域はそれぞれ PAO と LNO が割り当てられます。(b) DC-LNO 法の多階層並列化。

DC-LNO 法のオーダー ( $N$ ) 計算の最初のステップとして、ディレクトリ work に置かれた入力ファイルを用いて Si 結晶のオーダー ( $N$ ) 計算を以下のように実行できます。

```
% mpirun -np 112 ./openmx Si8-LNO.dat | tee si8-lno.std
```

この計算は Xeon クラスター計算機 (2.6 GHz) の 112 コアを用いて 66 秒で実行されました。また用いた入力ファイルはディレクトリ「work」中に「Si8-LNO.dat」として保存されています。図 19(b) に示すように、原子レベル、スピンレベル、対角化レベルの三段階の並列化が実装されています。そのため、例えば 1000 原子系であれば並列計算において 40000 CPU コアまでの加速が期待できます。1000 原子  $\times$  2 (スピン分極計算)  $\times$  20 CPU コア (ノードあたり) の積が 40000 となるためです。多階層並列化のベンチマーク計算は後で示します。ハイブリッド並列を実行する場合には、次のキーワードを設定して下さい。

```
scf.dclno.threading      on      # off|on
```

ハイブリッド並列では、並列化の最下層である対角化が OpenMP で並列化されます。この手法の計算の精度と効率は以下のキーワードで制御されます。

```

orderN.HoppingRanges    7.0    # 7.0 (Ang.)
orderN.LNO.Buffer        0.2    # default = 0.2
orderN.LNO.Occ.Cutoff    0.1    # default = 0.1

```

キーワード「orderN.HoppingRanges」の役割は DC 法でのそれと同一です。各原子に対する切られたクラスターはキーワード「orderN.HoppingRanges」で指定された半径の球内部の原子から構成されます。本パラメタの適切な選択は系によりますが、一連のベンチマーク計算からは 300 原子を含むように「orderN.HoppingRanges」を設定した場合には金属系に対しても十分な精度が確保されることが分かっています。

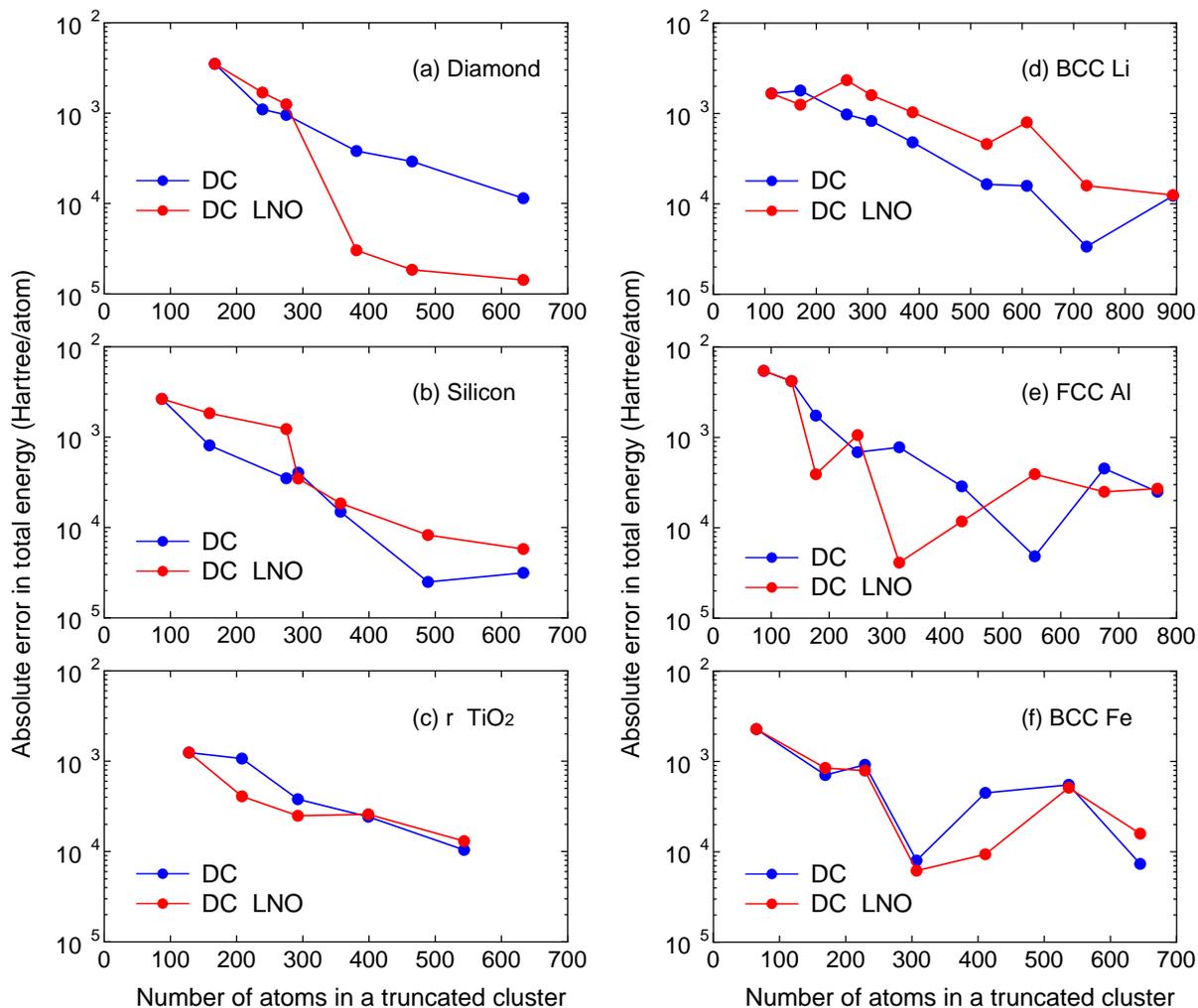


Figure 20: 切られたクラスター内の原子数に対する DC 法と DC-LNO 法の全エネルギーの絶対誤差 (Hartree/atom)。(a) ダイヤモンド、(b) ダイヤモンド構造のシリコン、(c) TiO<sub>2</sub> ルチル構造、(d) BCC 構造のリチウム、(e) FCC 構造のアルミニウム、(f) BCC 構造の鉄。どのケースでも実験の格子定数を使用。

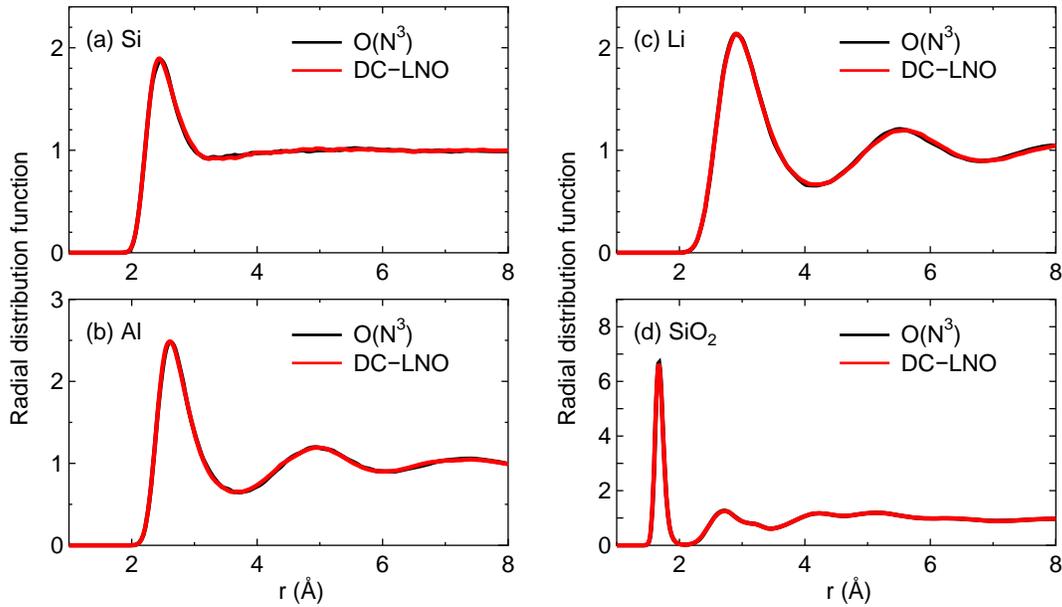


Figure 21: 通常のオーダー  $N^3$  対角化と DC-LNO 法によって計算された全動径分布関数 (RDF)。(a) 3500 K でのシリコン、(b) 2500 K でのアルミニウム、(c) 800 K でのリチウム、そして (d) 3000K での  $\text{SiO}_2$ 。MD シミュレーションは立方体のスーパーセルに対して時間刻み 2fs で 10ps の間で実行されました。シリコン、アルミニウム、リチウム、 $\text{SiO}_2$  の各スーパーセル (格子定数: 10.86, 12.15, 14.04, 14.25 (Å)) はそれぞれ 64, 108, 128, そして 192 個の原子を含む。シミュレーションの詳細は文献 [51] を参照のこと。

この設定は精度を重視したもので、効率性の面からは系に応じて検討が必要です。精度と効率のバランスの取れた選択は 200 から 300 原子程度にあると考えられます。LNO を割り当てる原子の領域はキーワード「orderN.LNO.Buffer」で制御されます。「orderN.LNO.Buffer=0.0」は全ての SNAN 原子に LNO を割り当てることを意味します。一方、「orderN.LNO.Buffer=1.0」の場合には全ての SNAN 原子に PAO を割り当てます。この取扱いは DC 法と同等です。SNAN については、「ユーザーによる FNAN+SNAN の定義」の 23.4 節を参照してください。「orderN.LNO.Buffer=1.0」の既定値は 0.2 ですが、0.1~0.2 の「orderN.LNO.Buffer」が精度と効率に関して適切な選択です。次式 [51] で定義される  $\Lambda_{0i}$  の固有値を確認することで各原子  $i$  の LNO が決定されます。

$$\Lambda_{0i} = \sum_{\mathbf{R}_j} \rho_{0i,\mathbf{R}_j} S_{\mathbf{R}_j,0i}, \quad (2)$$

ここで  $\rho_{0i,\mathbf{R}_j}$  と  $S_{\mathbf{R}_j,0i}$  はそれぞれ密度行列と重なり行列のブロック要素です。固有値は LNO に対する電子の占有数として解釈できますので、キーワード「orderN.LNO.Occ.Cutoff」で設定した占有数以上を持つ LNO を選択することにより、占有状態の部分空間を良く張った LNO が得られます。キーワード「orderN.LNO.Occ.Cutoff」を使用する代わりに各元素の LNO の数をキーワード「LNOs.Num」で直接に指定することもできます。例えば、元素種を以下のように定義した場合、

```
<Definition.of.Atomic.Species
Si    Si7.0-s2p2d1    Si_PBE19
H     H6.0-s2p1     H_PBE19
```

Definition.of.Atomic.Species>

LNO の数は以下の様に指定できます。

```
<LNOs.Num  
  Si    4  
  H     1  
LNOs.Num>
```

この場合、LNO の数は Si で 4、H で 1 にそれぞれ固定されます。構造最適化や分子動力学シミュレーションの際に LNO 数の変化を避けるため「orderN.LNO.Occ.Cutoff」よりも「LNOs.Num」の使用が推奨されます。DC 法と DC-LNO 法の比較を図 20 に示します。DC-LNO 法ではクラスター内の遠方領域において PAO の代わりに LNO が使用されていますが、ギャップのある系と金属系の両方においてその精度は DC 法に匹敵することが分かります。また DC-LNO 法の応用例として高温の分子動力学シミュレーションで得られたシリコン、アルミニウム、リチウム、SiO<sub>2</sub> の液体の動径分布関数 (RDF) を図 21 に示します。いずれの場合でも DC-LNO 法が通常のオーダー  $N^3$  対角化法の結果を良く再現していることが分かります。また得られた RDF は他の計算結果 [52, 53, 54, 55] とも整合しています。

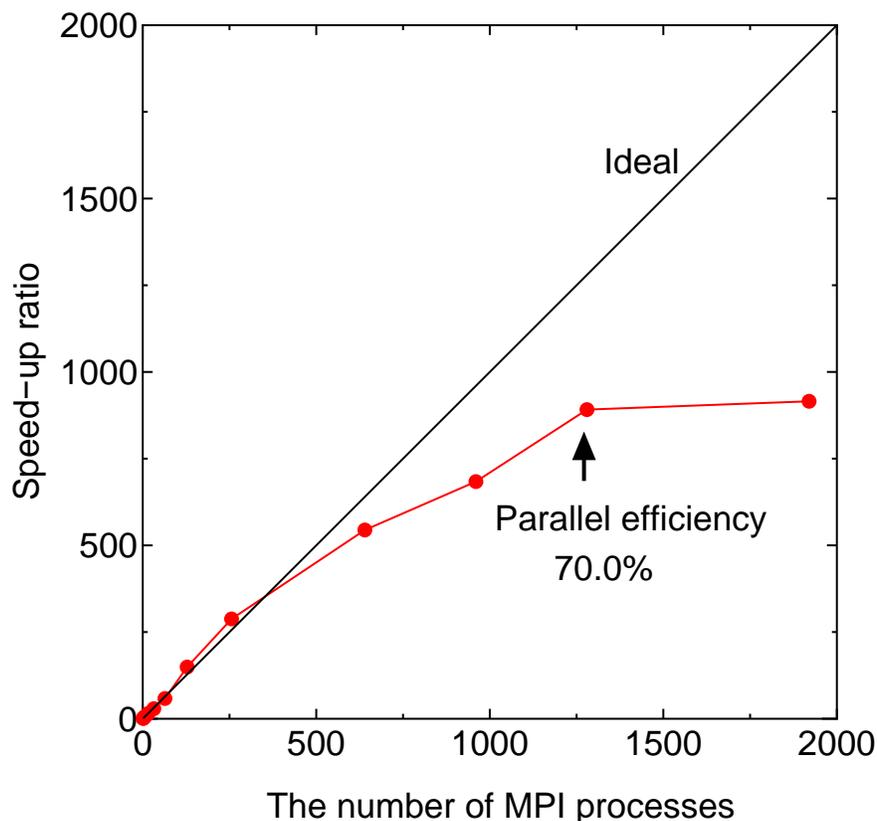


Figure 22: DC-LNO 法の MPI 並列化における速度向上比。ベンチマーク計算は 64 原子を含むダイヤモンドのスーパーセルに対して実施。8.0 Å のカットオフ半径を使用され、239 と 142 の原子が近傍領域と遠方領域に割り当てられ、結果として 3675 の行列次元を持ったクラスター問題になる。ベンチマーク計算の詳細は参考文献 [51] を参照のこと。

DC-LNO 法の MPI 並列計算における速度向上比を図 22 に示します。これはダイヤモンド (64 原子を含むスーパーセル) の非スピン分極計算の結果です。スピンインデックスの多重度は 1 のため、原子レベルの並列化の最大並列数である 64MPI プロセスまでほぼ理想的な振る舞いが見られます。64MPI プロセスを超えると原子レベルでの並列化に加えて対角化の並列化が考慮されます。128 と 256 の MPI プロセスを用いた場合には理想的な速度向上比を超えたスピードアップが観測されました。これはメモリ使用の削減によりキャッシュの効率的な利用によると推定されます。本ベンチマーク計算で用いた最大の 1280 MPI プロセスまで良いスケーリングが達成されています。1 MPI プロセスでの経過時間を基準として算出された並列化効率はおよそ 70% です。本ベンチマーク計算で使用した PC クラスターの各計算ノードは 20CPU コアから構成されるため、1280 (=64×20) MPI プロセスまで良いスケーリングが見られることは妥当な結果です。近年の超並列計算機の発展に伴って、多階層並列化は計算時間を短縮するために有効な手段であることが分かります。

### 23.3 $O(N)$ Krylov 部分空間法

DC 法は広範な系に適用でき、数値的にも安定な方法です。しかし金属系に対して高精度な結果を得るには、図 16 に示したように切り取られたクラスターのサイズを大きくする必要があり、その場合には計算量が増大します。計算量を削減する一つの方法は、切り取られたクラスターによって定義されるベクトル空間を次元数がより小さな Krylov 部分空間 [43] にマッピングすることです。次のキーワードにより  $O(N)$  Krylov 部分空間法を利用できます。

```
scf.EigenvalueSolver      Krylov
```

基本的に計算精度と計算効率は次の 2 つのキーワードで制御されます。

```
orderN.HoppingRanges      6.0
orderN.KrylovH.order       400
```

キーワード「orderN.HoppingRanges」は、DC 法の場合と同様に、各原子を中心とする球の半径を定義します。ハミルトニアンに対する Krylov 部分空間の次元は、キーワード「orderN.Krylov.order」で指定します。さらに次のキーワードで  $O(N)$  Krylov 部分空間法による計算の詳細が設定可能です。

- orderN.Exact.Inverse.S            on| off, default=on

キーワード「orderN.Exact.Inverse.S」を「on」に設定すると、切り取られた各クラスタの重なり行列の逆行列が厳密に評価されます。off に設定する場合はキーワード「orderN.KrylovS.order」の項を参照して下さい。デフォルトでは「on」に設定されています。

- orderN.KrylovS.order            1200, default=orderN.KrylovH.order×4

「orderN.Exact.Inverse.S=off」の場合、Krylov 部分空間法によって逆行列を近似します。この時、切り取られた各クラスタに対する重なり行列のクリロフ部分空間法の次元を「orderN.KrylovS.order」キーワードで指定します。デフォルト値は、「orderN.KrylovH.order」で設定した値の 4 倍です。

- orderN.Recalc.Buffer            on| off, default=on

キーワード「orderN.Recalc.Buffer」を「on」に設定すると、バッファ行列は各 SCF 反復毎に再計算されます。「off」の場合にはバッファ行列は初回の SCF ステップで計算され、その後の SCF 反復では固定されます。デフォルトでは「on」に設定されています。

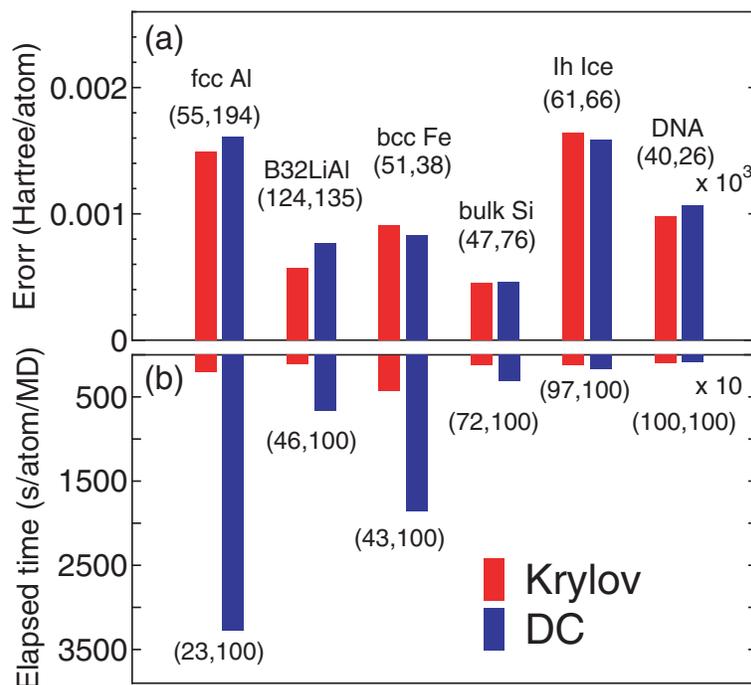


Figure 23: (a) Krylov 部分空間法と DC 法で計算された金属および絶縁固体の全エネルギー (Hartree/atom) における絶対誤差と (b) その計算時間 (s/atom/MD)。比較を容易にするため、1 MPI プロセスを用いて計算。図 (a) のカッコ内の数値はコア領域およびバッファ領域における平均原子数。図 (b) のカッコ内の数値は、切り取られたクラスター内の基底関数の数に対する Krylov 部分空間の次元の百分率を示す。

- orderN.Expand.Core            on| off, default=on

キーワード「orderN.Expand.Core」を「on」に設定すると、コア領域は半径  $1.2 \times r_{\min}$  の球の中にある原子から構成されます。ここで  $r_{\min}$  は、中心原子と最隣接原子間の距離です。このコア領域は Krylov 部分空間を生成するときの第 1 ステップで使用されるベクトル群を定義します。「orderN.Expand.Core」が「off」の場合、中心の原子がコア領域と見なされます。デフォルトでは「on」に設定されています。

一般にキーワード「orderN.Exact.Inverse.S」と「orderN.Expand.Core」は共有結合性物質に対しては「on」とした方が収束性が向上しますが、単純金属では「off」とした方が収束性が向上する場合があります。図 23 に、種々の物質に対し、Krylov 法と DC 法で計算した全エネルギーの絶対誤差を示します。特に金属系では DC 法と比較して Krylov 部分空間法の方が効率的であり、共有結合性とイオン性の増大とともに計算時間は同じ程度になることが分かります。

大規模な計算を実現するために、 $O(N)$  Krylov 部分空間法は並列化に配慮して実装されています。MPI による並列化の場合には原子数と同じ MPI プロセス数までは計算を加速させることができます。また OpenMP/MPI ハイブリッド並列によって OpenMP スレッドを使用すればさらに計算速度の向上が可能です。超並列計算機を用いて 10 万個以上の原子から構成される系の並列計算を実施した例を図 24 に示します [44, 45]。この例では、ベンチマーク系として 131072 個の炭素原子から構成されるダイヤモンドの計算を行い、131072 MPI プロセスを使用し、その並列化効率は 68%に達しています。

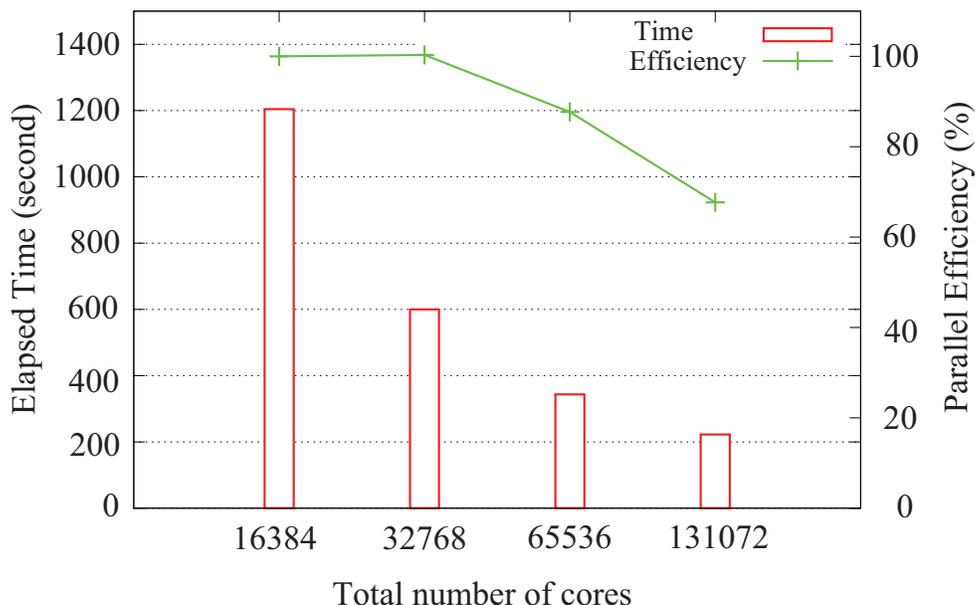


Figure 24: スーパーコンピュータ「京」上でのハイブリッド並列計算における  $O(N)$  Krylov 部分空間法の並列化効率。すべての場合で OpenMP の 8 スレッドを使用。ベンチマーク系として 131072 個の炭素原子から成るダイヤモンド構造を計算。

#### 23.4 ユーザーによる FNAN+SNAN の定義

OpenMX Ver. 3.9 で実装されている  $O(N)$  法では、切り取られたクラスター内の原子は第 1 隣接と第 2 隣接の 2 種類の原子に分類されます。中心原子上の基底関数のカットオフ半径  $r_0$  と隣接原子の基底関数のカットオフ半径  $r_N$  の和 ( $r_0 + r_N$ ) が、原子間距離より小さい場合、その原子は第 1 隣接原子に分類され、この条件を満たさないクラスター内の原子は第 2 隣接原子に分類されます。第 2 隣接原子の選択は、キーワード「orderN.HoppingRanges」によって決定されており、第 1 と第 2 隣接原子数は、それぞれ FNAN および SNAN として標準出力に書き出されています。さらに「orderN.HoppingRanges」とは独立に SNAN が決定するキーワード「orderN.FNAN+SNAN」が利用可能です。このキーワードによって (FNAN+SNAN) の数値を以下の様にユーザーが直接、与えることができます。

```
<orderN.FNAN+SNAN
 1  60
 2  65
 3  60
 4  50
 ..
 .
orderN.FNAN+SNAN>
```

この指定では、行数は原子数に等しくなければなりません。最初の列は、キーワード「Atoms.SpeciesAndCoordinates」の第 1 列と同一の通し番号であり、2 列目が (FNAN+SNAN) の値です。このキーワードを用いて、各原子に対して (FNAN+SNAN) が明示的に与えられた場合には、(FNAN+SNAN) の値が指定された値になるように各原子毎にクラスターの球半径が調整されます。キー

ワード「orderN.HoppingRanges」によってクラスターサイズを指定した場合の問題点は単位胞ベクトルを変化させた際に顕在化します。この場合、単位胞ベクトルを変化に伴い、大きく  $(FNAN+SNAN)$  が変化し、その結果としてエネルギー曲線に大きな飛びが生じます。キーワード「orderN.FNAN+SNAN」によって明示的に  $(FNAN+SNAN)$  を与えることで、この様なケースを避けることができます。

## 24 MPI並列化

大規模系の計算を実行するために、分散メモリ型並列計算機上での MPI 並列計算が実行可能です。

### 24.1 $O(N)$ 計算

$O(N)$  法は並列化に適したアルゴリズム構造を持っているため、高い並列化効率が期待されます。典型的な MPI の実行は次の様に行います。

```
% mpirun -np 4 openmx DIA512_DC.dat > dia512_dc.std &
```

「work」ディレクトリ中の「DIA512\_DC.dat」は、分割統治 (DC) 法を用いて 512 個の炭素原子からなるダイヤモンド格子の SCF 計算を実行するための入力ファイルです。MPI プロセス数に対する速度向上比を図 25 (a) に示します。この並列計算は CRAY-XC30 (Xeon プロセッサ, 2.6 GHz) 上で実行し、1MD ステップの経過時間から速度向上比が算出されました。128 個の MPI プロセスを用いて、その速度向上は約 84 倍であり、またプロセス数の増加に伴い、並列化効率は減少していくことが分かります。並列化効率が減少する理由は、計算負荷の大きな  $O(N)$  対角化計算において 1 個のプロセスに割り当てられる原子数が減少し、disk I/O のような並列化されていない部分の経過時間が顕在化するためです。また原子種や幾何学構造が一樣でない系では、負荷分散が崩れるために並列化効率が大きく減少する場合もあります。MD 計算や分子動力学計算では、動的に負荷分散を保つアルゴリズムが実装されており、可能な限り並列化効率の向上が図られています。並列化に関するさらなる情報については、「DC-LNO 法」と「 $O(N)$  Krylov 部分空間法」の章を参照して下さい。

### 24.2 クラスタ計算

クラスタ計算では、スピン多重度と固有状態のループ構造に対して MPI 並列化が実装されています。スピン多重度は非スピン分極計算およびノンコリニア計算では 1、スピン分極計算では 2 となります。スピン多重度がまず優先的に並列化され、多数の MPI プロセスが使用された際には、さらに固有状態が並列化されます。OpenMX Ver. 3.9 では、クラスタ計算の固有値問題は、高並列固有値ソルバー: ELPA [39] によって計算されます。ELPA1 または ELPA2 が次のキーワードで選択できます。

```
scf.eigen.lib      elpa1      # elpa1|elpa2, default=elpa1
```

既定の選択肢は ELPA1 です。ここでは提示しませんが、我々のベンチマーク計算は計算速度に関してほぼ同様な性能であることを示唆しています。図 25 (b) に、 $Mn_{12}$  単一分子磁石のスピン分極計算の所要時間と速度向上比を示します。使用した入力ファイル「Mn12.dat」は「work」ディレクトリに保存されています。速度向上比は、32 プロセスと 64 プロセスで、それぞれ 11 および 17 であることが分かります。

### 24.3 バンド計算

バンド計算ではスピン多重度、 $k$  点、固有状態の 3 つのループ構造に対して MPI 並列化が実装されています。スピン多重度は非スピン分極計算およびノンコリニア計算では 1、スピン分極計算では 2 となります。スピン多重度がまず優先的に並列化され、多数の MPI プロセスが使用された際には、さらに  $k$  点、固有状態の順番で並列化されます。また MPI プロセス数が (スピン多重度)  $\times$  ( $k$  点数) 以下の場合、各  $k$  点で化学ポテンシャルと密度行列の各計算のために 2 回の対角化が行われます。これは波動関数を

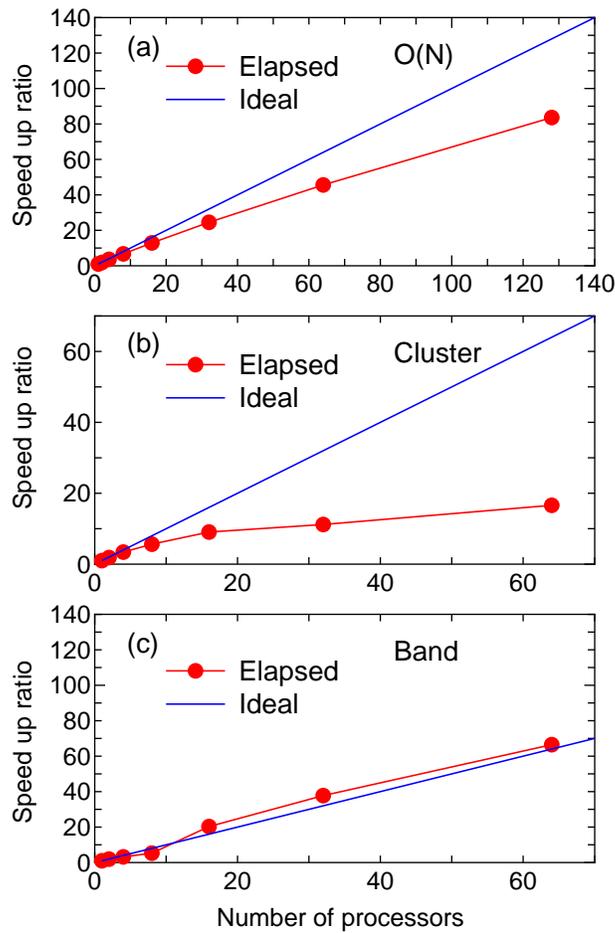


Figure 25: CRAY-XC30 (Xeon プロセッサ、2.6GHz) 上で MPI 並列計算を行った場合の、1MD ステップの所要時間における速度向上比。(a) DC 法によるダイヤモンド (512 炭素原子) の計算、(b) クラスタ法による  $Mn_{12}$  単一分子磁石 (148 原子) の計算、(c) バンド法によるダイヤモンド (64 炭素原子、 $3 \times 3 \times 3$  の k 点) の計算。比較のために、理想的な速度向上比に対応する直線も示す。

保持する際のメモリ使用量を削減するために、1 回目の対角化で化学ポテンシャルを探索し、2 回目の対角化で波動関数を保持することなく、直接に密度行列を計算しています。一方、MPI プロセス数が (スピン多重度)  $\times$  (k 点数) を越える場合、各プロセスは多数の k 点での波動関数をメモリ上に保持する必要がないため、対角化は一度だけで、計算効率が見かけの上で向上します。図 25 (c) に示す様に、プロセス数の関数として速度向上比を見た場合、バンド計算の並列化効率は良好であることが分かります。この計算は  $3 \times 3 \times 3$  個の k 点を考慮したダイヤモンド (64 炭素原子) の非スピン分極バンド計算を行ったもので、使用した入力ファイルは「work」ディレクトリにある「DIA64\_Band.dat」です。この場合のスピン多重度は 1 です。またコリニア計算ではブリルアンゾーンの半分の k 点のみが考慮されます。a 軸、b 軸、c 軸に対する全ての k 点数が奇数の場合には、 $\Gamma$  点が含まれますので、実際の計算が行われた k 点数は  $(3 \times 3 \times 3 - 1) / 2 + 1 = 14$  となります。プロセス数が 14 以上では、速度向上比が理想値を超えています。これは上記で説明したアルゴリズムの変更によるものです。クラスタ計算と同様に、バンド計算における固有値問題の解法には高並列固有値ソルバー: ELPA [39] を使用しています。ELPA1 または ELPA2 が次のキーワードで選択できます。

```
scf.eigen.lib      elpa1      # elpa1|elpa2, default=elpa1
```

既定の選択肢は ELPA1 です。ここでは提示しませんが、我々のベンチマーク計算は計算速度に関してほぼ同様な性能であることを示唆しています。

#### 24.4 完全な 3 次元並列化

OpenMX Ver. 3.6 以下のバージョンでは、単位胞の a 軸に対する 1 次元領域分割法によりデータを分散し、それに基づき MPI 並列化が実装されていました。一方、OpenMX Ver. 3.9 ではデータ分散のために、完全な 3 次元領域分割をサポートしています。従って、並列化の際に均等な負荷分散を実現するために単位胞をどのように指定したら良いかについて、考慮する必要がありません。OpenMX Ver. 3.9 では、単位胞ベクトルの選択に依存することなく、ほぼ同等の並列化効率を得られます。

#### 24.5 使用可能な MPI プロセス数

OpenMX Ver. 3.6 以下のバージョンでは、並列計算で使用できる MPI プロセス数の最大値は、系に含まれる原子数で制限されていました。OpenMX Ver. 3.9 ではこのような制限はありません。MPI プロセス数が原子数より大きい場合でも、可能な限り MPI 並列化が効率良く行われます。この機能は k 点数が系の原子数より大きい場合に特に有用となるでしょう。

## 25 MPI/OpenMP ハイブリッド並列化

MPI/OpenMP を用いたハイブリッド並列計算は次のコマンドで実行可能です。

```
% mpirun -np 32 openmx DIA512-1.dat -nt 4 > dia512-1.std &
```

「-nt」によって各 MPI プロセスに割り当られたスレッド数を与えます。もし「-nt」が指定されない場合は、スレッド数は 1 にセットされ、この場合には通常の MPI 並列化となります。OpenMX Ver. 3.9 の並列化は OpenMX Ver. 3.6 からは大きく変更されていますので、計算効率とメモリ使用の効率に関して、通常の MPI 並列化と比較して、ハイブリッド並列化が有効かどうかまだ十分なデータがありません。しかし予備的なベンチマーク計算では、ハイブリッド並列化は、メモリの利用に関しては効率が良く、一方で計算効率に関しては両者とも同程度であるという結果となっています。通常の並列化と OpenMP/MPI ハイブリッド並列化の適切な選択は、使用する計算機に依存していると考えられますので、ベンチマーク計算によって、確認されることを推奨します。

## 26 大規模計算

### 26.1 通常の固有値解法

数百個のプロセッサコアが利用可能であれば、通常対角化法で 1000 原子から構成される系の構造最適化が可能です。1000 原子系のベンチマーク計算は次のように「runtestL2」によって実行できます。

```
% mpirun -np 128 openmx -runtestL2 -nt 4
```

OpenMX は 7 個のテスト計算を実行し、計算結果を「work/large2\_example」に保存されている参照データと比較します。以下に示すのは、Xeon クラスター計算機で 640 個の MPI プロセスと 1 個の OpenMP スレッドを用いて実行した「runtestL2」の結果です。

|   |                                |                          |                           |                            |
|---|--------------------------------|--------------------------|---------------------------|----------------------------|
| 1 | large2_example/C1000.dat       | Elapsed time(s)= 777.60  | diff Utot= 0.000000007341 | diff Force= 0.000000008795 |
| 2 | large2_example/Fe1000.dat      | Elapsed time(s)= 8181.70 | diff Utot= 0.000000002241 | diff Force= 0.000000011061 |
| 3 | large2_example/GRA1024.dat     | Elapsed time(s)= 927.20  | diff Utot= 0.000000012903 | diff Force= 0.000000004981 |
| 4 | large2_example/Ih-Ice1200.dat  | Elapsed time(s)= 445.88  | diff Utot= 0.000000000216 | diff Force= 0.000000001451 |
| 5 | large2_example/Pt500.dat       | Elapsed time(s)= 2629.20 | diff Utot= 0.000000015832 | diff Force= 0.000000001879 |
| 6 | large2_example/R-TiO2-1050.dat | Elapsed time(s)= 844.58  | diff Utot= 0.000000002263 | diff Force= 0.000000001108 |
| 7 | large2_example/Si1000.dat      | Elapsed time(s)= 658.53  | diff Utot= 0.000000000404 | diff Force= 0.000000000908 |

Total elapsed time (s)            14464.69

これら全ての計算において、各原子の基底関数には価電子基底関数を 2 つ、分極基底関数を 1 つを割り当てていますので、十分な精度を持った計算となっています。「Pt500.dat」を除く、他の全ての系は 1000 原子以上から構成されています。ここでファイル名の最後の数字はそれぞれの系に含まれる原子数を示しています。上記の表に示された 1 MD ステップ当たりの経過時間は 30 分程度です。従って、数百個のプロセッサコアが利用可能であれば 1000 原子から構成される系の構造最適化が可能であることが分かります。この計算に用いた入力ファイルと出力ファイルはディレクトリ「work/large2\_example」に保存されています。出力ファイルから得られた情報に基づき、1 SCF ステップ当たりの経過時間をまとめたものを以下に示します。

| No. | Input file                     | SCF steps | Elapsed time(s)/SCF/spin | Dimension |
|-----|--------------------------------|-----------|--------------------------|-----------|
| 1   | large2_example/C1000.dat       | 53        | 14.7                     | 13000     |
| 2   | large2_example/Fe1000.dat      | 408       | 10.0                     | 13000     |
| 3   | large2_example/GRA1024.dat     | 72        | 12.9                     | 13312     |
| 4   | large2_example/Ih-Ice1200.dat  | 57        | 7.8                      | 9200      |
| 5   | large2_example/Pt500.dat       | 161       | 16.3                     | 12500     |
| 6   | large2_example/R-TiO2-1050.dat | 38        | 22.2                     | 15750     |
| 7   | large2_example/Si1000.dat      | 45        | 14.6                     | 13000     |

Kohn-Sham ハミルトニアン次元は 10000 のオーダーで、SCF ステップ当たりの所要時間は全ての系に対して 15 秒程度です。全所要時間の差は、主に SCF 反復回数差から生じていることが分かります。

### 26.2 $O(N)$ 法と通常の固有値解法の組み合わせ

$O(N)$  法によって 1000 個以上の原子を含む大規模な系を取り扱うことが可能ですが、OpenMX で実行される  $O(N)$  法では、波動関数に関する情報を得ることができません。大規模系に対する波動関数と対応する固有値を得るための簡便な方法は、最初に  $O(N)$  法を用いて自己無撞着な電子密度を求め、次にこの自己無撞着な電子密度の下で、一度だけ通常対角化計算を行うものです。この方法の例として、

564 個の炭素原子で構成される多重連結したカーボンナノチューブ (MCCN) の大規模計算を示します。まず最初に、16 MPI プロセス (Xeon, 2.6GHz) と  $O(N)$  Krylov 部分空間法を用いて MCCN の SCF 計算を実行しました。計算条件は以下の通りです。C5.0-s2p1 (基底関数)、「scf.energycutoff=130 Ryd」、「scf.criterion=1.0e-7」、「orderN.HoppingRange=6.5 Å」、「orderN.KrylovH.order=400」、および RMM-DIISK (混合法)。入力ファイルは「work」ディレクトリの中の「MCCN.dat」です。フーリエ表示での差分電子密度のノルムを SCF ステップの関数として、図 26 に示します。系の電子密度が収束するのに 56 SCF ステップが必要であり、また計算時間は約 7 分でした。この後、以下の様にキーワードを設定します。

```

scf.maxIter          1
scf.EigenvalueSolver Band
scf.Kgrid            1 1 1
scf.restart          on
MO.fileout           on
num.HOMOs            2
num.LUMOs            2
MO.Nkpoint           1
<MO.kpoint
  0.0  0.0  0.0
MO.kpoint>

```

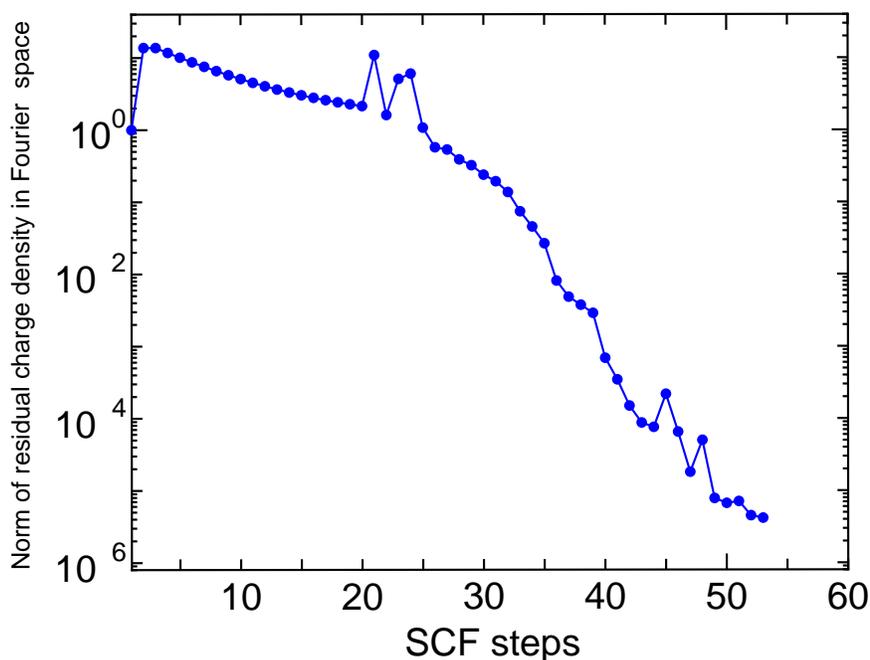


Figure 26: 564 個の炭素原子で構成される多重連結したカーボンナノチューブ (MCCN) の SCF 収束の過程。フーリエ表示での差分電子密度のノルムを SCF ステップに対して表示。入力ファイルは「work」ディレクトリ中の「MCCN.dat」。

次に波動関数を計算するために、16 MPI プロセス (Xeon, 2.6GHz) と通常対角化法を用いて、同じ系の計算を実行し、その計算時間は約 2 分でした。図 27 は上記の方法で計算した MCCN の  $\Gamma$  点での HOMO および LUMO の等値面図です。この計算例では、 $O(N)$  法と通常対角化法の計算時間の差はそれほど大きくありませんが、 $O(N)$  法と通常固有値解法の組み合わせる本手法は数千原子を含む大規模系に対して有用だと考えられます。

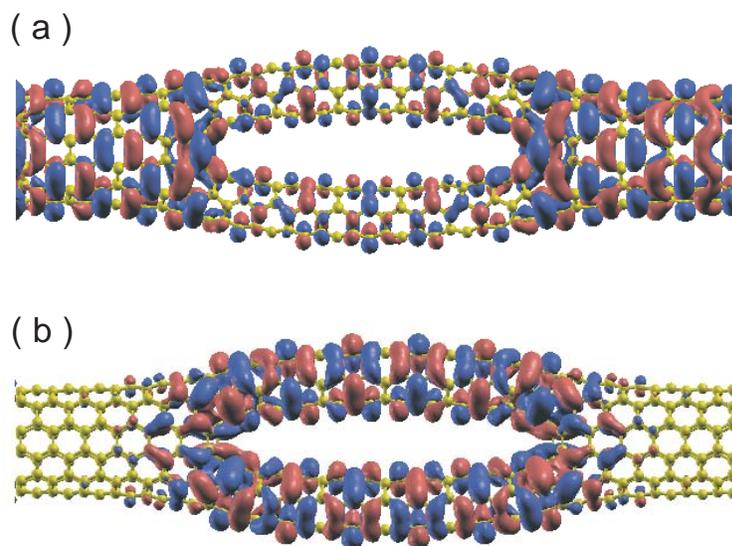


Figure 27: 564 個の炭素原子からなる多重連結カーボンナノチューブ (MCCN) の (a) 最高被占軌道 (HOMO) および (b) 最低空軌道 (LUMO) の等値面図。等値面值として  $|0.005|$  を使用。

## 27 電場

SCF 計算や構造最適化の際に、ノコギリ波による一様な外部電場を印加することができます。例えば、a 軸に沿って  $1.0 \text{ GV/m}$  ( $10^9 \text{ V/m}$ ) の電場を印加する場合、次の様にキーワード「scf.Electric.Field」を指定して下さい。

```
scf.Electric.Field 1.0 0.0 0.0 # default=0.0 0.0 0.0 (GV/m)
```

電場の符号は、電子に印加されるものとして定義されています。ノコギリ波による一様な外部電場が真空領域を持たない周期的なバルク系に印加されると、ポテンシャルの不連続面が導入されることとなります。そのため、数値的な不安定性が引き起こされる可能性があります。一方、分子系に対しては不連続面は真空領域に配置されますので、通常は数値不安定性の問題は起こりません。電場印加の例として、電場によって誘起されたニトロベンゼン分子中の全電荷の変化を図 28 に示します。 $-\text{NO}_2$  内の酸素とパラ炭素原子、それにパラ水素原子の間で大きな電子移動が起こっているのが分かります。入力ファイルは「work」ディレクトリ中の「Nitro\_Benzene.dat」です。図 28 に示した差電子マップに関しては「2つの Gaussian Cube ファイルの差の解析」の 63 章も参照して下さい。

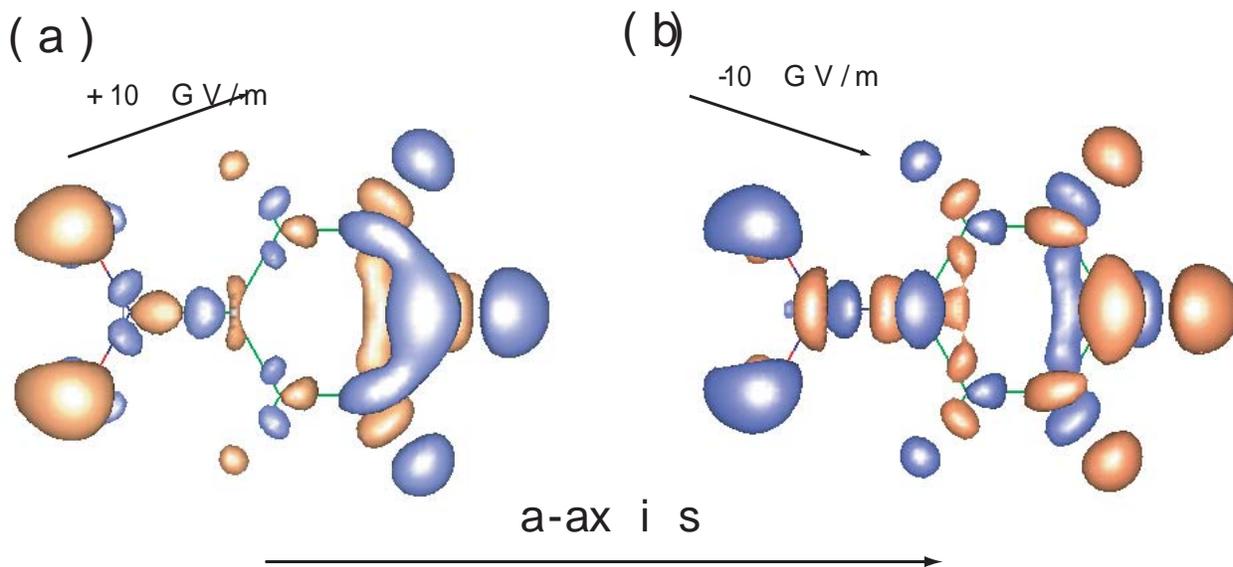


Figure 28: ゼロ外部電場印加の場合と、a 軸に沿って印加された外部電場 (a)  $10 \text{ GV/m}$ 、および (b)  $-10 \text{ GV/m}$  下でのニトロベンゼン分子の全電子密度の差。ここでオレンジ色と青色はそれぞれ、電子密度の増加および減少を示す。傾いた矢印は印加した電場の傾きを図示。入力ファイルは「work」ディレクトリ中の「Nitro\_Benzene.dat」。

## 28 電荷ドーピング

キーワード「scf.system.charge」により電子と正孔のドーピングが可能です。

```
scf.system.charge 1.0 # default=0.0
```

正と負の符号はそれぞれ正孔と電子のドーピングに対応しています。部分的な電荷のドーピングも可能です。OpenMX では Poisson 方程式の解法に FFT が用いられていますので、キーワード「scf.system.charge」で与えられる過剰電荷は逆電荷を持ったような背景電荷によって中性化されます。従って、異なる電荷状態間の全エネルギーを比較する場合には、全エネルギーにはバックグラウンドの電荷によって誘起される付加的な静電的相互作用が含まれていることに注意が必要です。電荷と孤立系を扱う正しい方法について章 58 で議論されています。

例として、正孔ドーブ、中性、電子ドーブの場合のカーボンナノチューブ (14 Å の有限長) のスピン密度を図 29 に示します。中性および電子ドーブのナノチューブの全スピンモーメントは 1.0 および 2.2 であり、一方、正孔をドーブしたナノチューブでは全スピンモーメントはほとんどゼロです。中性および電子ドーブのナノチューブではナノチューブ端のダングリングボンドによりスピン分極が発生しています。

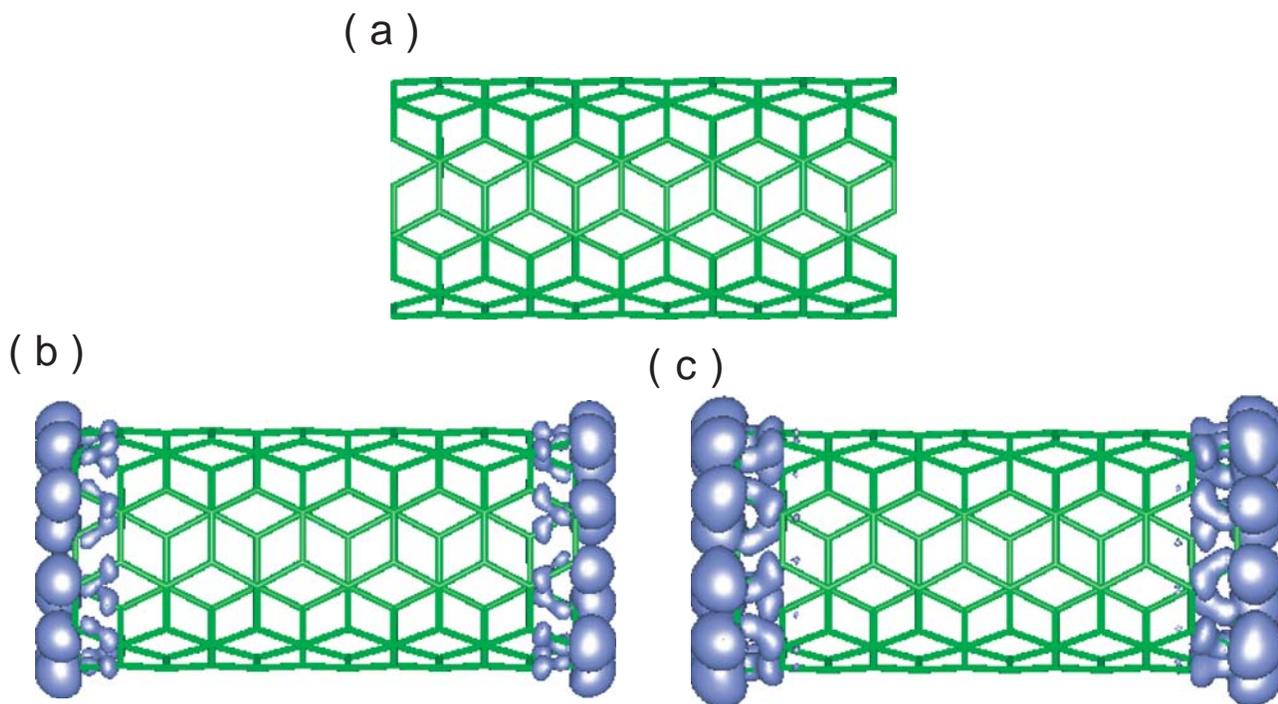


Figure 29: (a) 4 つの正孔をドーブした場合、(b) 中性の場合、そして (c) 4 つの電子をドーブした場合の、14 Å の有限長の (5,5) カーボンナノチューブのスピン密度。インプットファイルは「work」ディレクトリ中の「Doped\_NT.dat」。

## 29 分数核電荷を持つ仮想原子

分数核電荷を持つ仮想原子を導入した計算が実行できます。これは分数核電荷を持つ仮想原子の擬ポテンシャルを使用することで可能となります。仮想原子の擬ポテンシャルは ADPACK で生成できます。ADPACK での関連するキーワードは次の様に与えられます。

```
AtomSpecies          6.2
total.electron       6.2
valence.electron     4.2
<occupied.electrons
 1   2.0
 2   2.0  2.2
occupied.electrons>
```

上の例は炭素原子と窒素原子の間に位置付けられる仮想原子に対するものです。仮想原子の擬ポテンシャルに対応する基底関数は同じ分数核電荷を持つ仮想原子に対して生成しなければなりません。これは、\*.pao に保存されている原子の電子密度を用いて中性原子のポテンシャルが作成されるためです。例として、この方法をより計算した  $C_{7.8}N_{0.2}$  の DOS を図 30 に示します。入力ファイルはディレクトリ「work」中の「DIA8-VA.dat」です。この計算では、単位セル内の 8 個の炭素原子の内の 1 つを、有効核電荷 4.2 の仮想原子に置き換えました。これは  $C_{7.8}N_{0.2}$  の化学量論的化合物に相当しています。

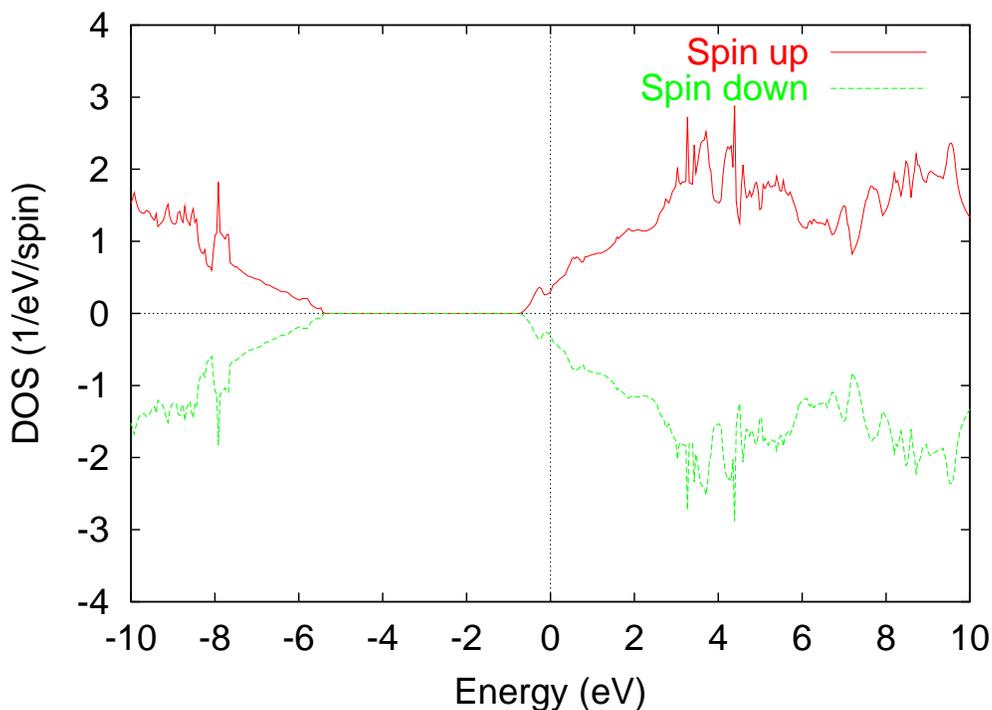


Figure 30: 仮想原子の擬ポテンシャルを使って計算した  $C_{7.8}N_{0.2}$  の状態密度 (DOS)。用いた入力ファイルは「work」ディレクトリ中の「DIA8-VA.dat」。

### 30 LCAO 係数

クラスター計算とバンド計算の両方で、LCAO 係数を解析することができます。クラスター計算では、キーワード「level.of.fileout」を2に設定すると、ファイル「*System.Name.out*」にLCAO 係数が加えられます。例として、「テスト計算」の章で扱っている「Methane.dat」のLCAO 係数を以下に示します。

```
*****
*****
Eigenvalues (Hartree) and Eigenvectors for SCF KS-eq.
*****
*****

Chemical Potential (Hartree) = 0.000000000000000
HOMO = 4

LCAO coefficients for up (U) and down (D) spins

          1 (U)   2 (U)   3 (U)   4 (U)   5 (U)   6 (U)
-0.69899 -0.41525 -0.41525 -0.41524  0.21215  0.21215

1  C 0 s      0.69137 -0.00000  0.00000  0.00000  0.00000  0.00000
   0 px      0.00000 -0.10055  0.63544  0.00033 -0.68649 -1.00467
   0 py      0.00000  0.00028 -0.00029  0.64331  0.00000 -0.00001
   0 pz     -0.00000  0.63544  0.10055 -0.00023 -1.00467  0.68649
2  H 0 s      0.12870  0.05604 -0.35474 -0.25425 -0.59781 -0.87489
3  H 0 s      0.12870 -0.35475 -0.05627  0.25420 -0.87488  0.59781
4  H 0 s      0.12870  0.35497  0.05604  0.25393  0.87488 -0.59781
5  H 0 s      0.12870 -0.05626  0.35497 -0.25388  0.59781  0.87488

          7 (U)   8 (U)
0.21223  0.24739

1  C 0 s      0.00000  1.90847
   0 px      0.00000  0.00000
   0 py     -1.21683 -0.00000
   0 pz     -0.00000  0.00000
....
...
```

バルクの計算でLCAO 係数を出力する場合にはキーワード「MO.fileout」を「ON」に設定して下さい。その際、キーワード「MO.kpoint」で指定されたk点におけるLCAO 係数がファイル「*System.Name.out*」に出力されます。クラスター計算でLCAO 係数を出力するためには「level.of.fileout」を「2」と設定しますが、バンド計算では、関連するキーワードは、「level.of.fileout」ではなく「MO.fileout」となります。

## 31 分子軌道

分子や結晶の Kohn-Sham 軌道を Gaussian cube 形式で出力できます。Gaussian cube 形式は広く用いられるデータ形式であるため、VESTA [103] や XCrySDen [105] などの多くのソフトウェアを用いて Kohn-Sham 軌道を可視化することが出来ます。関連するキーワードは以下で与えられます。

```
MO.fileout      on      # on|off, default=off
num.HOMOs       1      # default=2
num.LUMOs       1      # default=2
MO.Nkpoint      2      # default=1

<MO.kpoint
  0.0  0.0  0.0
  0.5  0.0  0.0
MO.kpoint>
```

cube ファイルを生成する際にはキーワード「MO.fileout」を「ON」として下さい。出力する最高占有の分子あるいは結晶の軌道 (HOMO) の数と最低非占有の分子あるいは結晶の軌道 (LUMO) の数はキーワード「num.HOMOs」と「num.LUMOs」でそれぞれ指定できます。バンド計算では HOMO と LUMO を計算する k 点をキーワード「MO.Nkpoint」と「MO.kpoint」で指定します。キーワード「MO.Nkpoint」は HOMO と LUMO を計算する k 点の数を与えます。キーワード「MO.kpoint」によって上に示したように k 点を露わに指定できます。k 点は「Atoms.UnitVectors」で与えられた単位セルベクトルの逆格子ベクトルに基づいて指定されます。出力される cube ファイルは以下のとおり要約されます。

クラスターの場合:

「MO.fileout=ON」かつ「scf.EigenvalueSolver=Cluster」ならば以下のファイルが生成されます。

- *System.Name.homo0\_0.cube*, *System.Name.homo0\_1.cube*, ...

HOMO が Gaussian cube 形式で出力されます。「homo」の下の最初の数字はスピン状態を表します (up=0, down=1)。二番目の数字は固有状態を表します。すなわち、0, 1, 2 がそれぞれ HOMO, HOMO-1, HOMO-2 に対応します。この総数はキーワード「num.HOMOs」で指定されます。

- *System.Name.lumo0\_0.cube*, *System.Name.lumo0\_1.cube*, ...

LUMO が Gaussian cube 形式で出力されます。「lumo」の下の最初の数字はスピン状態を表します (up=0, down=1)。二番目の数字は固有状態を表します。すなわち、0, 1, 2 がそれぞれ LUMO, LUMO+1, LUMO+2 に対応します。この総数はキーワード「num.LUMOs」で指定されます。

バルクの場合:

「MO.fileout=ON」かつ「scf.EigenvalueSolver=Band」ならば以下のファイルが生成されます。

- *System.Name.homo0\_0\_0\_r.cube*, *System.Name.homo1\_0\_1\_r.cube*, ... *System.Name.homo0\_0\_0\_i.cube*,  
*System.Name.homo1\_0\_1\_i.cube*, ...

HOMO が Gaussian cube 形式で出力されます。「homo」の下の最初の数字は k 点を意味し、この k 点はキーワード「MO.kpoint」で指定されます。二番目の数字はスピン状態を表します (up=0, down=1)。三番目の数字は固有状態を表します。すなわち、0, 1, 2 がそれぞれ HOMO, HOMO-1, HOMO-2 に対応します。この総数はキーワード「num.HOMOs」で指定されます。r と i は波動関数の実部と虚部を表します。

- *System.Name.lumo0\_0\_0.r.cube*, *System.Name.lumo1\_0\_1.r.cube*, ... *System.Name.lumo0\_0\_0.i.cube*, *System.Name.lumo1\_0\_1.i.cube*, ...

LUMO が Gaussian cube 形式で出力されます。「lumo」の下の最初の数字は k 点を意味し、この k 点はキーワード「MO.kpoint」で指定されます。二番目の数字はスピン状態を表します (up=0, down=1)。三番目の数字は固有状態を表します。すなわち、0, 1, 2 がそれぞれ LUMO, LUMO+1, LUMO+2 に対応します。この総数はキーワード「num.LUMOs」で指定されます。r と i は波動関数の実部と虚部と表します。

例として valorphin 分子の HOMO と LUMO を図 31 に示します。

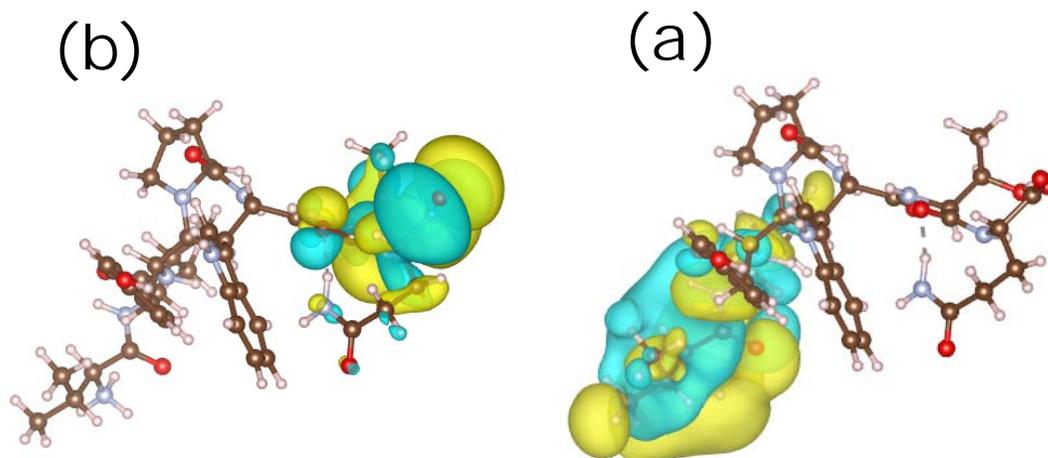


Figure 31: (a) valorphin 分子の HOMO、(b) valorphin 分子の LUMO。VESTA [103] による可視化。入力ファイルはディレクトリ work の Valorphin\_MO.dat。

## 32 電荷解析

それぞれの原子に有効電荷を割り当てるのは恣意性を含んだ難しい問題ですが、OpenMX では各原子の電荷状態を解析するために Mulliken 電荷解析、Voronoi 電荷解析、また静電ポテンシャルフィッティング法 (ESP) の 3 つの方法が利用可能です。

### 32.1 Mulliken 電荷

「テスト計算」の章で示したように Mulliken 電荷の計算結果はデフォルトで「*System.Name.out*」に出力されます。それぞれの原子に対して射影された Mulliken 電荷に加えて、「*System.Name.out*」ではそれぞれの基底関数に射影された Mulliken 電荷も出力されています。メタン分子の出力ファイル「*met.out*」に保存された結果を以下に示します。

Decomposed Mulliken populations

| 1              | C        | Up spin     | Down spin   | Sum         | Diff        |
|----------------|----------|-------------|-------------|-------------|-------------|
|                | multiple |             |             |             |             |
| s              | 0        | 0.598003833 | 0.598003833 | 1.196007667 | 0.000000000 |
| sum over m     |          | 0.598003833 | 0.598003833 | 1.196007667 | 0.000000000 |
| sum over m+mul |          | 0.598003833 | 0.598003833 | 1.196007667 | 0.000000000 |
| px             | 0        | 0.588514081 | 0.588514081 | 1.177028163 | 0.000000000 |
| py             | 0        | 0.588703212 | 0.588703212 | 1.177406424 | 0.000000000 |
| pz             | 0        | 0.588514081 | 0.588514081 | 1.177028162 | 0.000000000 |
| sum over m     |          | 1.765731375 | 1.765731375 | 3.531462749 | 0.000000000 |
| sum over m+mul |          | 1.765731375 | 1.765731375 | 3.531462749 | 0.000000000 |
| 2              | H        | Up spin     | Down spin   | Sum         | Diff        |
|                | multiple |             |             |             |             |
| s              | 0        | 0.409066346 | 0.409066346 | 0.818132693 | 0.000000000 |
| sum over m     |          | 0.409066346 | 0.409066346 | 0.818132693 | 0.000000000 |
| sum over m+mul |          | 0.409066346 | 0.409066346 | 0.818132693 | 0.000000000 |
| 3              | H        | Up spin     | Down spin   | Sum         | Diff        |
|                | multiple |             |             |             |             |
| s              | 0        | 0.409065912 | 0.409065912 | 0.818131824 | 0.000000000 |
| sum over m     |          | 0.409065912 | 0.409065912 | 0.818131824 | 0.000000000 |
| sum over m+mul |          | 0.409065912 | 0.409065912 | 0.818131824 | 0.000000000 |
| .....          |          |             |             |             |             |
| ....           |          |             |             |             |             |

ここで分かるように、Mulliken 電荷は全ての基底関数に対して分解されています。上記の表では分解された Mulliken 電荷は 2 種類の方法で加算されています。その一つは「sum over m」で、それぞれの動

径関数に対する磁気量子数についての加算を意味しています。もう一つは「sum over m+mul」で、これは磁気量子数と動径関数についての加算を意味しています。

## 32.2 Voronoi 電荷

各原子の Voronoi 電荷は、Voronoi 多面体内の電子密度を積分することで計算されます。OpenMX での Voronoi 多面体は Fuzzy セル分割法 [69] による連続的にぼかされた (smeared) 境界から構成されています。Voronoi 多面体は原子半径を考慮せずに構造だけで決定されるので、Voronoi 解析はしばしば過大あるいは過小評価した電荷を与えることに注意する必要があります。Voronoi 電荷を計算する場合には、入力ファイルに次のキーワード「Voronoi.charge」を設定して下さい。

```
Voronoi.charge      on      # on|off, default = off
```

メタン分子の場合、以下の Voronoi 電荷が「*System.Name.out*」に出力されます。

```
*****
*****
Voronoi charges
*****
*****
Sum of Voronoi charges for up      = 4.000000290723
Sum of Voronoi charges for down    = 4.000000290723
Sum of Voronoi charges for total   = 8.000000581446

Total spin magnetic moment (muB) by Voronoi charges = 0.000000000000

      Up spin    Down spin    Sum        Diff        Voronoi Volume (Ang.^3)
Atom=  1  1.129270484  1.129270484  2.258540969  0.000000000  2.355421391
Atom=  2  0.717682452  0.717682452  1.435364903  0.000000000  62.245579466
Atom=  3  0.717682452  0.717682452  1.435364903  0.000000000  62.245579466
Atom=  4  0.717682452  0.717682452  1.435364903  0.000000000  62.245579466
Atom=  5  0.717682451  0.717682451  1.435364903  0.000000000  62.245579466
```

炭素原子 (Atom = 1) は正に、水素原子 (Atom=2-5) は負に帯電しており、この結果は化学的な直観とは整合しません。しかし、Voronoi 解析は最密構造を持つバルク固体に対して有益な情報を与え、Voronoi 体積はバルク系の局所構造の解析にも有効です。

## 32.3 静電ポテンシャルフィッティング

小さな分子系に対しては、静電ポテンシャル (ESP) フィッティング法 [109, 110, 111] が有効電荷を決定するのに良く用いられています。多くの場合に、得られた結果は化学的な直観に整合しています。ただし、ESP フィッティング法では表面部分から離れた原子に対して、十分なサンプリング点が存在しないために、巨大分子やバルク系には適用できません。ESP フィッティング法では、最小二乗法を用いて有効点電荷を決定します。DFT 計算によって求めた静電ポテンシャルを可能な限り再現するように、最小二乗法を用いて有効点電荷を決定します。ESP フィッティング電荷は次の二段階で計算します。

### (1) SCF 計算

通常の SCF 計算後、次の 2 つのファイルが出力されます。

```
System.Name.out
System.Name.vhart.cube
```

二つのファイルを作成するための追加のキーワードは必要ありません。これらのファイルは通常の SCF 計算から得られる出力ファイルですが、キーワード「level.of.fileout」は 1 または 2 に設定する必要があります。

## (2) ESP フィッティング電荷

ESP フィッティング電荷を計算するためにプログラムコードをコンパイルします。ディレクトリ「source」に移動し、次のようにコンパイルします。

```
% make esp
```

コンパイルが正常に完了すると、ディレクトリ「work」に実行形式ファイル「esp」が作成されます。ESP フィッティング電荷はプログラム「esp」を用いて二つのファイル「System.Name.out」, 「System.Name.vhart.cube」から計算します。例えば、「テスト計算」の章で示したメタン分子の場合、ESP フィッティング電荷は次の様に計算されます。

```
% ./esp met -c 0 -s 1.4 2.0
```

拡張子なしのファイル名を指定するだけで十分ですが、二つのファイル「met.out」と「met.vhart.cube」は「work」ディレクトリに保存しておかなければなりません。オプションの「-c」と「-s」は制限条件と倍率 (scale factor) を指定するパラメータです。ソースコード「eps.c」のヘッダーの部分に次の説明が記載されています。

```
-c      constraint parameter
        '-c 0' means charge conservation
        '-c 1' means charge and dipole moment conservation
-s      scale factors for vdw radius
        '-s 1.4 2.0' means that 1.4 and 2.0 are 1st and 2nd scale factors
```

ESP フィッティング法は、電荷保存、並びに電荷と双極子モーメント保存の二つの制約条件下でフィッティングを行います。後者は DFT 計算によって求めた電荷と双極子モーメントを再現できますが、双極子モーメントの保存条件を導入すると、特に大きな分子に対して、物理的に容認できない点電荷が得られる可能性が生じます。従って前者の制約条件が推奨されます。サンプリング点としては二つのシェル構造で挟まれた領域内の実空間格子点が選択されます。シェル構造は各原子に割り当てられた球の足し合せから定義されます。球の半径は「-s」以降に与えた 1 番目および 2 番目の倍率 (scale factor) × ファン・デル・ワールス半径で与えられます [112]。

計算結果は次のように標準出力 (ユーザーのディスプレイ) に出力されます。

```
% ./esp met -c 0 -s 1.4 2.0
```

```
*****
*****
esp: effective charges by a ESP fitting method
Copyright (C), 2004, Taisuke Ozaki
```

This is free software, and you are welcome to  
redistribute it under the constitution of the GNU-GPL.

\*\*\*\*\*  
\*\*\*\*\*

Constraint: charge

Scale factors for vdw radius 1.40000 2.00000

Number of grids in a van der Waals shell = 28464

Volume per grid = 0.0235870615 (Bohr<sup>3</sup>)

Success

Atom= 1 Fitting Effective Charge= -0.93558216739

Atom= 2 Fitting Effective Charge= 0.23389552572

Atom= 3 Fitting Effective Charge= 0.23389569182

Atom= 4 Fitting Effective Charge= 0.23389535126

Atom= 5 Fitting Effective Charge= 0.23389559858

Magnitude of dipole moment 0.0000015089 (Debye)

Component x y z 0.0000003114 -0.0000002455 -0.0000014558

RMS between the given ESP and fitting charges (Hartree/Bohr<sup>3</sup>)= 0.096515449505

### 33 自然電子密度解析

Weinhold [6] による自然結合軌道法では、原子有効電荷は自然占有数 (natural population, NP) と呼ばれ、原子に局在化した自然原子軌道 (natural atomic orbitals, NAO) から計算されます。OpenMX は NP 解析をサポートしており、truncated cluster 法をベースとした独自の手法 [7] により、大規模系においても選択した原子の NP 解析を効率的に実施することが可能です。NP の計算は SCF 計算の後に行われ、標準出力に結果が表示されます。NP 解析は以下の手順で行って下さい。

#### (a) 小規模計算の場合 (scf.EigenvalueSolver = Cluster)

NP の計算は以下のキーワード により制御します。

```
NBO.switch    on1
```

このオプションでは、フルサイズの密度行列を用いて NP の計算が行われます。エチレンカーボネート分子に対する例が「work/nbo\_example」ディレクトリ内に「EC\_NAO.dat」と「EC\_NAO.std」として収められていますので、ご参照下さい。

#### (b) 大規模計算の場合 (scf.EigenvalueSolver = Krylov)

「NBO.switch = on1」を用いた大規模系の NAO 計算は、密度行列のサイズが大きいためメモリ不足となる場合があります。その様な場合、大規模系の計算に対しては「NBO.switch」を以下の様に設定した計算を行って下さい。

```
NBO.switch    on2
```

$O(N)$  クリロフ部分空間法 (scf.EigenvalueSolver = Krylov) を用いた計算の際に、このオプションが使用可能です。 $O(N)$  の SCF 計算の後、指定した数原子に対して効率的に NP 計算を行います ( 化学反応のような重要なイベントを解析する際には、全原子についてではなく局所的な数原子に対して NP 計算をすれば十分なケースがほとんどです )。NP 解析の対象となる原子は、以下のように指定します。

```
NBO.Num.CenterAtoms    5
<NBO.CenterAtoms
  269
  304
  323
  541
  574
NBO.CenterAtoms>
```

キーワード「NBO.Num.CenterAtoms」によって、計算対象とする原子の数を指定して下さい。続けて「<NBO.CenterAtoms」と「NBO.CenterAtoms>」の間に計算対象とする原子の通し番号を記載して下さい。ここで用いる通し番号は、キーワード「Atoms.SpeciesAndCoordinates」にて指定したものです。「NBO.switch」が「on2」の場合にのみ「NBO.CenterAtoms」は有効であり、「on1」の場合には「NBO.CenterAtoms」は無視され全原子に対して NP 計算が行われます。大規模系に対する NP 解析の例として SiO<sub>2</sub>\_NAO.dat と SiO<sub>2</sub>\_NAO.std が「work/nbo\_example」ディレクトリに収められています。これらは、648 原子からなる SiO<sub>2</sub> のバルク系に対する計算例です。

### (c) NAO 計算に用いるパラメータ: NAO.threshold

NAO 計算の際には、占有された NAO と Rydberg (低占有) NAO を分離するプロセスがあります。分離する際の閾値として用いるキーワードが「NAO.threshold」であり、スピン軌道毎の電子数を指定します。デフォルト値は 0.85 として設定されており、多くの場合にこの値で問題ありません。不自然な占有 NAO 数が得られた場合には、「NAO.threshold」を調整して下さい。

### (d) 計算例

以下にエチレンカーボネート分子に対する NP 解析の計算結果を示します。まず、全ての原子に対する NP とその総和が標準出力に表示されます。

```
1  O :   6.46917105
2  C :   4.09908587
3  C :   4.09909317
4  O :   6.46902031
5  C :   3.14623972
6  O :   6.50714720
7  H :   0.80250093
8  H :   0.80249967
9  H :   0.80262024
10 H :   0.80262185
-----
Total   :  34.00000000

## Global atom num.: 1 ( O ) / NP =   6.4692
-----
NP in NAO      0.0013  1.6803  0.0046  1.6731  0.0055  1.3000  0.0073  1.7972
Energy (Hartree) 1.2529 -0.7196  0.4452 -0.3208  0.7108 -0.2935  0.4327 -0.3005
-----
1 s             -1.8000  1.2493 -0.0776  0.1255 -0.0125 -0.0027  0.0000  0.0000
2 s             1.9201  0.0014  0.0530 -0.0030  0.0163  0.0015 -0.0000 -0.0000
1 px            -0.4312  0.1031 -0.6568  1.0499  0.0431  0.0115 -0.0000 -0.0000
2 px            0.1913 -0.0062  1.7251  0.0215 -0.0500 -0.0052  0.0000  0.0000
1 py            -0.1040 -0.0056  0.0371  0.0110 -2.1699  1.0361 -0.0000 -0.0000
2 py            0.0573  0.0025 -0.0493 -0.0052  3.4499  0.1576  0.0000  0.0000
1 pz            0.0000 -0.0000 -0.0000 -0.0000 -0.0000 -0.0000 -0.6206  1.0131
2 pz            -0.0000 -0.0000  0.0000  0.0000  0.0000  0.0000  1.7312  0.0267
...
..
.
```

全ての原子に対する NP が表示された後、各原子の NAO に対する NP とエネルギー準位が表示され、各 NAO に対応する LCPAO 係数が表示されます。将来的には NBO の計算及び解析も OpenMX に実装される予定です。

## 34 ノンコリニア DFT

OpenMX ではノンコリニア DFT 法が利用可能です。各実空間グリッド上でスピン量子化軸に制限を持たない最も一般的な方法が実装されています。またスピン軌道相互作用 [8, 9, 10, 11, 16] も自己無撞着計算において取り入れることが可能です。ノンコリニア DFT 計算を実行する場合、キーワード「scf.SpinPolarization」を以下のように設定して下さい。

```
scf.SpinPolarization      NC      # On|Off|NC
```

この場合、Kohn-Sham 軌道は二成分スピノルによって表現されます。それぞれのサイトの初期スピンの方位は次の様に与えられます。

```
<Atoms.SpeciesAndCoordinates      # Unit=Ang
  1 Mn    0.00000  0.00000  0.00000  8.0  5.0  45.0  0.0  45.0  0.0  1 on
  2 O     1.70000  0.00000  0.00000  3.0  3.0  45.0  0.0  45.0  0.0  1 on
Atoms.SpeciesAndCoordinates>
```

- 1: 原子の通し番号
- 2: 原子種の名前
- 3: x-座標 (もしくは a 軸に対する規格化座標)
- 4: y-座標 (もしくは b 軸に対する規格化座標)
- 5: z-座標 (もしくは c 軸に対する規格化座標)
- 6: up スピン状態に対する初期占有数
- 7: down スピン状態に対する初期占有数
- 8: スピン磁気モーメントに対する局所磁場の Euler 角: theta
- 9: スピン磁気モーメントに対する局所磁場の Euler 角: phi  
8 と 9 番目の Euler 角は初期スピン密度の生成にも使用
- 10: 軌道磁気モーメントに対する局所磁場の Euler 角: theta
- 11: 軌道磁気モーメントに対する局所磁場の Euler 角: phi
- 12: 次のいずれかのキーワード「scf.Constraint.NC.Spin」、  
「scf.NC.Zeeman.Orbital」、  
もしくは「scf.NC.Zeeman.Orbital」が指定された際に、  
それぞれの原子に対して制約法を適用する場合には「1」、適用しない場合には「0」。
- 13: LDA(GGA)+U 法の際に、軌道分極を促進する場合には「on」、  
通常計算の場合には「off」。

スピン磁気モーメントと軌道磁気モーメントの方位に対する初期 Euler 角 ( $\theta$ ,  $\phi$ ) はそれぞれ 8 番目、9 番目および 10 番目、11 番目の列で与えられます。12 番目の列は制約法のスイッチで、それぞれの原子サイトにスピン磁気モーメントと軌道磁気モーメントの方位に対する制約汎関数を付加します。ここで、「1」は制約汎関数を付加、「0」は制約法無しを意味します。スピン磁気モーメントの方位に制約条件を付加する DFT 計算の詳細については「スピン磁気モーメント方位の制約 DFT」の 38 章を参照して下さい。最後の 13 番目の列は LDA(GGA)+U 法の際に、軌道分極を促進する手法へのスイッチで、軌道分極を促進する場合には「on」、通常計算の場合には「off」となります。図 32 はノンコリニア DFT 法で計算した MnO 分子中のスピン磁気モーメントをベクトル表示したものです。この計算は、「work」ディ

レクトリ内の入力ファイル「Mol\_MnO\_NC.dat」を使って、再現することができます。実空間におけるノンコリニアスピン磁気モーメントを可視化するために、次の二つのファイルが利用できます。

```
System.Name.nc.xsf
System.Name.ncsden.xsf
```

ここで、*System.Name* はユーザーが指定した「System.Name」です。「*System.Name.nc.xsf*」と「*System.Name.ncsden.xsf*」はXCrySDen でサポートされているベクトルファイルフォーマット (XSF) のファイルです。前者が Mulliken 解析によってそれぞれの原子に対して射影されたスピン磁気モーメントを、後者が実空間グリッド上のスピン磁気モーメントになります。この二つのファイルは、図 32 に示すようにXCrySDen の「Display → Forces」を使って可視化できます。

Mulliken 解析で計算されるそれぞれの原子のスピン磁気モーメントとその Euler 角は、ファイル「*System.Name.out*」内に次のように保存されています。

```
*****
*****
Mulliken populations
*****
*****

Total spin moment (muB)   4.998503442   Angles (Deg) 44.991211196   0.000000000

      Up      Down      Sum      Diff      theta      phi
1  Mn  9.59803  4.76902  14.36705  4.82901  44.99208  0.00000
2   O  3.40122  3.23173   6.63295  0.16949  44.96650 -0.00000
```

一般にノンコリニア DFT 法における SCF 計算はコリニア計算と比較して、収束が容易ではありません。これはスピン方位の変化に伴うエネルギーの変化が非常に小さいためです。スピン磁気モーメントや軌道磁気モーメントの方位に制約条件を課す制約法は SCF 計算の収束の加速に有効に働くことがあります。ノンコリニア DFT 法では、スピン軌道相互作用がサポートされていますが、コリニア DFT 法ではサポートされていません。スピン軌道相互作用に関しては「相対論的效果」の章を参照して下さい。

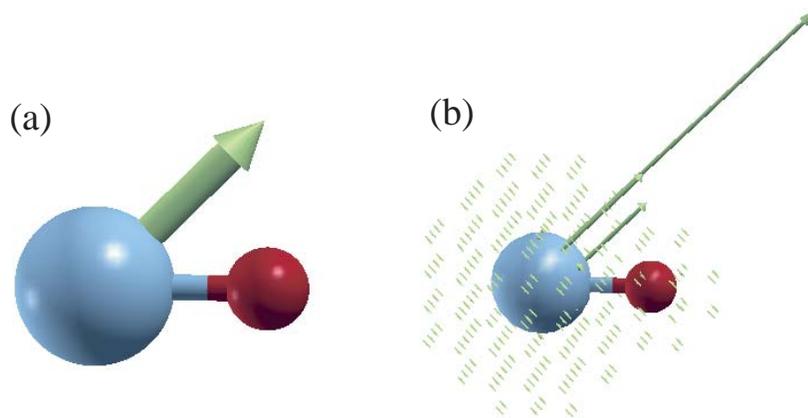


Figure 32: ノンコリニア DFT 法で計算した MnO 分子中のスピン磁気モーメント。(a) Mulliken 解析によってそれぞれの原子に対して射影されたスピン磁気モーメント。(b) 実空間グリッド上のスピン磁気モーメント。図は XCrySDen の「Display → Forces」により可視化。入力ファイルは「work」ディレクトリ内の「Mol\_MnO\_NC.dat」。

## 35 相対論的效果

OpenMX の WEB サイトで公開されている擬ポテンシャルは原子に対する Dirac 方程式を数値的に解くことで生成されています。相対論的擬ポテンシャルには質量項やポテンシャル勾配項から生じる相対論効果に加え、スピン軌道相互作用も含まれています。この相対論的擬ポテンシャルを用いて相対論効果を考慮することが可能です。ただし OpenMX の計算では波動方程式として Schrödinger 方程式と同等の微分演算子が用いられていますので、価電子に対する直接的な相対論効果は無視されています。直接的な効果は小さいことが分かっており、価電子に対する大部分の相対論効果は内殻電子を通し、間接的に生じます。間接的な相対論効果は擬ポテンシャルの中に含まれており、結局、内殻電子だけでなく OpenMX の計算で扱う価電子に対してもまた十分な精度で相対論効果が考慮されています。スピン軌道相互作用を量子数  $j$  の縮退度に応じて平均化する操作は「半相対論的扱い」として知られ、またスピン軌道相互作用を含めた場合には「完全な相対論的扱い」と呼ばれます。コリニア DFT 計算の際には「半相対論的扱い」のみが実行できます。またノンコリニア DFT 計算の際には「半相対論的扱い」と「完全な相対論的扱い」のどちらも実行可能です。

### 35.1 完全な相対論的扱い

擬ポテンシャル法の範疇で、スピン軌道相互作用を含む「完全な相対論効果」をノンコリニア DFT 計算において考慮することができます [12, 32, 16]。一方、スピン軌道相互作用の取扱いはコリニア DFT 計算ではサポートされていません。次の二つのステップで「完全な相対論効果」の考慮が可能です。

#### (1) $j$ 依存型擬ポテンシャルの生成

まず最初に ADPACK を使用し、 $j$  依存型擬ポテンシャルを作成して下さい。ユーザーの便宜のために、多くの元素に対する  $j$  依存型擬ポテンシャルがデータベース Ver. 2019 [149] として利用可能です。 $j$  依存型擬ポテンシャルを作成するための詳細は ADPACK のマニュアルを参照して下さい。

#### (2) SCF 計算

キーワード「Definition.of.Atomic.Species」の設定中で  $j$  依存型擬ポテンシャルを指定した場合、次の

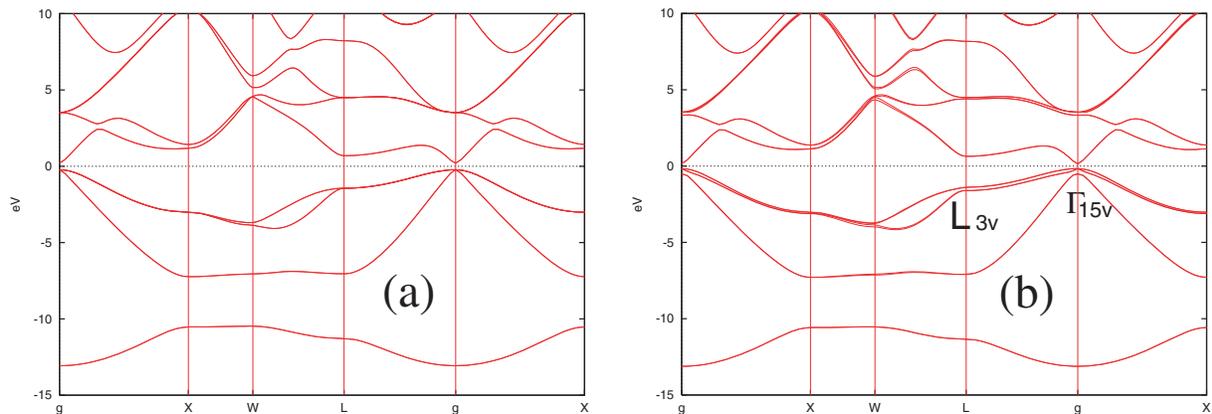


Figure 33: ノンコリニア DFT 法によって計算したスピ軌道相互作用が (a) 無し、(b) 有りの場合の GaAs 固体のバンド構造。Ver.2019 のデータベースから、基底関数として Ga7.0-s2p2d2 と As7.0-s2p2d2 を、また擬ポテンシャルとして Ga\_CA19.vps と As\_CA19.vps を使用。交換相関項は LDA を、scf.Kgrid と scf.energycutoff にはそれぞれ  $12 \times 12 \times 12$  と 140 (Ryd) を使用。また計算における格子定数は実験値 ( $5.65 \text{ \AA}$ )。入力ファイルは「work」ディレクトリの「GaAs.dat」。

キーワード「scf.SpinOrbit.Coupling」でスピ軌道相互作用が導入できます。

```
scf.SpinOrbit.Coupling      on      # On|Off, default=off
```

スピ軌道相互作用は、摂動法として取り扱われるのではなく、擬ポテンシャル法の枠組で自己無撞着に組み込まれます。このスピ軌道相互作用を介して、二成分スピノル中の  $\alpha$  および  $\beta$  スピン成分は直接に相互作用します。ノンコリニア DFT 計算において実空間でのスピ方位はスピ軌道相互作用を組み込んだ場合に意味を持ちます。スピ軌道相互作用が存在しない場合では、分子や結晶方位に対する相対的なスピ方位はエネルギー的に縮退しており、実空間でのスピ方位は一義的に決定されません。ただし相対的なスピ間の方位（例えば強磁性と反強磁性）はスピ軌道相互作用が含めない場合でも物理的に意味を持ち、エネルギー的な差異を生じさせますので誤解しないようにして下さい。スピ軌道分裂の例として、図 33 にノンコリニア DFT 法によって計算されたスピ軌道相互作用が (a) 無い場合と (b) ある場合の GaAs 固体のバンド構造を示します。入力ファイルは「work」ディレクトリの「GaAs.dat」です。図 33 (a) では分裂は見られませんが、図 33 (b) ではスピ軌道相互作用によっていくつかのバンドの縮退が解けていることが分かります。表 5 に二つの k 点、 $\Gamma$  点および L 点でのスピ軌道相互作用によるエネルギー分裂幅を示します。他の計算値および実験値と良い一致が見てとれます。

Table 5: GaAs 固体の  $\Gamma_{15v}$  および  $L_{3v}$  において計算されたスピ軌道分裂 (eV)。比較のために他の理論値 (LMTO: 参考文献 [113], PP: 参考文献 [114]) と実験値 (参考文献 [115]) も併記。入力ファイルは「work」ディレクトリ中の「GaAs.dat」。

| Level          | OpenMX | LMTO  | PP   | Expt. |
|----------------|--------|-------|------|-------|
| $\Gamma_{15v}$ | 0.344  | 0.351 | 0.35 | 0.34  |
| $L_{3v}$       | 0.213  | 0.213 | 0.22 |       |

## 35.2 スピン軌道相互作用のコントロール

OpenMX Ver. 3.9 では、データベース Ver. 2019 に保存された擬ポテンシャルを用いてスピン軌道相互作用の強さを制御することができます。スピン軌道相互作用の強弱を予め調整した擬ポテンシャルを特別に用意する必要はありません。それぞれの角運動量量子数に対するスケール係数を以下のようにキーワード「scf.SO.factor」で指定します。

```
<scf.SO.factor
  Ga  s 1.0 p 2.0 d 1.0 f 1.0
  As  s 1.0 p 2.0 d 1.0 f 1.0
scf.SO.factor>
```

1 列目はキーワード「Definition.of.Atomic.Species」で定義された原子名であり、それに続けて角運動量量子数のシンボル、係数を指定します。スケール係数「1.0」は実際の原子におけるスピン軌道相互作用に対応します。系内の全ての原子に対して、 $f$  軌道まで必ず指定して下さい。

## 35.3 半相対論的扱い

「半相対論的な取り扱い」を導入する方法の一つは、ADPACK によって作成された半相対論的擬ポテンシャルを使用することですが、より簡単な方法は  $j$  依存型擬ポテンシャルを用い、次のようにキーワード「scf.SpinOrbit.Coupling」を「off」にすることです。

```
scf.SpinOrbit.Coupling      off      # On|Off, default=off
```

この場合、OpenMX の計算において  $j$  依存型擬ポテンシャルが読み込まれた際に、自動的に  $j$  の縮退度を重みとして平均化されます。この平均化の操作によって  $j$  依存型擬ポテンシャルは半相対論的な擬ポテンシャルに変換されます。つまり、 $j$  依存型擬ポテンシャルは、相対論的および半相対論的取り扱いの両者に利用できることとなります。従って、相対論的な効果を考慮する場合には、半相対論的な擬ポテンシャルではなく、完全に相対論的な  $j$  依存型擬ポテンシャルを作成することをお勧めします。実際、図 33 (a) の計算は「scf.SpinOrbit.Coupling=off」とし、図 33 (b) と同じ擬ポテンシャルのファイルを用いて、実行したものです。

## 36 軌道磁気モーメント

各原子サイトでの軌道磁気モーメントは、ノンコリニア DFT 計算において、デフォルトで計算されます。局所的な軌道磁気モーメントはスピン軌道相互作用の結果として現れるため、スピン軌道相互作用が考慮された場合にのみ、この値は有限値となります [118, 119]。例として、ディレクトリ「work」内の入力ファイル「FeO\_NC.dat」を用いた酸化鉄固体のノンコリニア LDA+ $U$  計算 ( $U=5\text{eV}$ ) について説明します。LDA+ $U$  の計算については、「DFT+ $U$ 」の章を参照して下さい。計算した鉄のサイトの軌道磁気モーメントおよびスピン磁気モーメントを表 6 に示します。また以下のように「*System.Name.out*」内に軌道成分に分解した軌道磁気モーメントが出力されています。ここで「*System.Name*」は「System.Name」を意味します。

\*\*\*\*\*

\*\*\*\*\*

### Orbital moments

\*\*\*\*\*

\*\*\*\*\*

Total Orbital Moment (muB) 0.000001885 Angles (Deg) 126.954120326 185.681623854

|   |    | Orbital moment (muB) | theta (Deg) | phi (Deg) |
|---|----|----------------------|-------------|-----------|
| 1 | Fe | 0.76440              | 131.30039   | 51.57082  |
| 2 | Fe | 0.76440              | 48.69972    | 231.57071 |
| 3 | O  | 0.00000              | 40.68612    | 210.48405 |
| 4 | O  | 0.00000              | 48.18387    | 222.72367 |

### Decomposed Orbital Moments

|                                  |          | Orbital Moment(muB) | Angles (Deg)     |
|----------------------------------|----------|---------------------|------------------|
| 1                                | Fe       |                     |                  |
|                                  | multiple |                     |                  |
| s                                | 0        | 0.000000000         | 90.0000 0.0000   |
| sum over m                       |          | 0.000000000         | 90.0000 0.0000   |
| s                                | 1        | 0.000000000         | 90.0000 0.0000   |
| sum over m                       |          | 0.000000000         | 90.0000 0.0000   |
| px                               | 0        | 0.000055764         | 42.7669 270.0000 |
| py                               | 0        | 0.000046795         | 28.9750 180.0000 |
| pz                               | 0        | 0.000044132         | 90.0000 239.0920 |
| sum over m                       |          | 0.000120390         | 47.1503 239.0920 |
| px                               | 1        | 0.001838092         | 10.8128 -90.0000 |
| py                               | 1        | 0.001809013         | 3.5933 180.0000  |
| pz                               | 1        | 0.000362989         | 90.0000 251.7994 |
| sum over m                       |          | 0.003683170         | 11.3678 251.7994 |
| d3z <sup>2</sup> -r <sup>2</sup> | 0        | 0.043435663         | 90.0000 224.2874 |
| dx <sup>2</sup> -y <sup>2</sup>  | 0        | 0.066105902         | 24.3591 229.7056 |

Table 6: 遷移金属酸化物 MO (M=Mn, Fe, Co, Ni) のスピン磁気モーメント  $M_s(\mu_B)$  と軌道磁気モーメント  $M_o(\mu_B)$ 。LDA+ $U$  法 [20] を用いて計算し、M の最初の 1d 軌道に対して、Mn には 3.0 (eV)、Fe には 5.0 (eV)、Co には 7.0 (eV)、Ni には 7.0 (eV) の実効  $U$  値を使用。Mulliken 解析は多重基底関数の使用に際して大きめのスピンモーメントを与える傾向があるため、Voronoi 分割によって、局在スピンモーメントを計算。入力ファイルは「work」ディレクトリ中の「MnO\_NC.dat」, 「FeO\_NC.dat」, 「CoO\_NC.dat」, 「NiO\_NC.dat」。比較のために他の理論値 [70] と実験値 [70] も併記。

| Compound | $M_s$  |             | $M_o$  |             | Expt. in total |
|----------|--------|-------------|--------|-------------|----------------|
|          | OpenMX | Other calc. | OpenMX | Other calc. |                |
| MnO      | 4.519  | 4.49        | 0.004  | 0.00        | 4.79,4.58      |
| FeO      | 3.653  | 3.54        | 0.764  | 1.01        | 3.32           |
| CoO      | 2.714  | 2.53        | 1.269  | 1.19        | 3.35,3.8       |
| NiO      | 1.687  | 1.53        | 0.247  | 0.27        | 1.77,1.64,1.90 |

|                                    |   |             |          |          |
|------------------------------------|---|-------------|----------|----------|
| dxy                                | 0 | 0.361874370 | 80.4206  | 50.6465  |
| dxz                                | 0 | 0.397108491 | 144.2572 | -12.7324 |
| dyz                                | 0 | 0.427070801 | 138.9995 | 100.0151 |
| sum over m                         |   | 0.776513038 | 132.4577 | 51.6984  |
| d3z <sup>2</sup> -r <sup>2</sup>   | 1 | 0.000144144 | 90.0000  | 196.4795 |
| dx <sup>2</sup> -y <sup>2</sup>    | 1 | 0.000270422 | 31.2673  | 224.0799 |
| dxy                                | 1 | 0.003006770 | 85.5910  | 50.2117  |
| dxz                                | 1 | 0.002952926 | 139.3539 | -4.1301  |
| dyz                                | 1 | 0.003222374 | 134.0513 | 95.9246  |
| sum over m                         |   | 0.006795789 | 126.2536 | 52.1993  |
| f5z <sup>2</sup> -3r <sup>2</sup>  | 0 | 0.001903274 | 90.0000  | 33.4663  |
| f5xz <sup>2</sup> -xr <sup>2</sup> | 0 | 0.005186342 | 14.5594  | 118.0868 |
| f5yz <sup>2</sup> -yr <sup>2</sup> | 0 | 0.005258572 | 17.3323  | -35.0807 |
| fzx <sup>2</sup> -zy <sup>2</sup>  | 0 | 0.005477755 | 29.3372  | 224.9067 |
| fxyz                               | 0 | 0.004851020 | 10.1407  | 249.0607 |
| fx <sup>3</sup> -3*xy <sup>2</sup> | 0 | 0.002029489 | 84.1842  | -81.2087 |
| f3yx <sup>2</sup> -y <sup>3</sup>  | 0 | 0.001611593 | 82.6686  | 176.3172 |
| sum over m                         |   | 0.020307129 | 9.9551   | 249.3739 |
| .....                              |   |             |          |          |
| ...                                |   |             |          |          |

表 6 に示すように、OpenMX の計算結果は、一連の 3d 遷移金属酸化物のスピンおよび軌道磁気モーメントの他の計算結果と整合しています。しかし、軌道磁気モーメントの絶対値は基底関数や DFT+ $U$  法のオンサイトの  $U$  などの計算条件に大きく影響を受けます。軌道磁気モーメントの収束計算には分極関数を含むかなり大きな基底関数が必要であることにも注意すべきです。一方、スピン磁気モーメントの値は計算条件に対する変化が比較的、小さいことが経験上、分かっています。

## 37 DFT+ $U$ 法

3d 軌道や 4f 軌道などの局在軌道上での局所クーロン相互作用を考慮するために、DFT+ $U$  法が利用可能です。占有数演算子、汎関数形、ダブルカウント項の取り扱いに関して様々な表式の DFT+ $U$  法が実装されています。本機能はコリニア計算とノンコリニア計算の両方で利用できます。本機能を用いて論文等を出版される際には実装に関する論文 [20, 21, 22] を引用して下さい。本手法の理論と実装の詳細は参考文献 [20, 21, 22] と技術ノート [23, 24] を参照して下さい。

### 37.1 標準設定

#### 37.1.1 DFT+ $U$ 法の選択: 単純化または一般化

DFT+ $U$  法における占有数演算子、汎関数形、ダブルカウント項を選択するために次の二つのキーワードを指定します。

```
scf.Hubbard.U          on # on|off, default=off
scf.DFTU.Type         2  # 1:Simplified(Dudarev)|2:General, default=1
```

これらのキーワードはコリニア計算とノンコリニア計算 [21, 22] の両方で共通です。scf.DFTU.Type=1 は Dudarev ら [25] によるいわゆる「単純化回転不変形式」に相当し、ここで  $U$  (Hubbard  $U$ ) のみが役割を担います [20]。より一般的な DFT+ $U$  法では  $U$  だけでなく、 $J$  (Hunt のカップリング  $J$ ) も入力パラメタとなります。DFT+ $U$  法を使用する際にはキーワード「scf.SpinPolarization」は常に「on」または「nc」と設定して下さい。

占有数演算子 [20] は以下のキーワードで指定されます。

```
scf.Hubbard.Occupation dual # onsite|full|dual, default=dual
```

三つの占有数演算子 (onsite、full、dual) において、dual 演算子のみが総和測 (占有数行列のトレースが電子の総数を与える) を満たします。「onsite」、「full」、「dual」の演算子の詳細は参考文献 [20] を参照して下さい。

原子種をキーワード「Definition.of.Atomic.Species」で以下の様に定義した場合、

```
<Definition.of.Atomic.Species
Ni Ni6.0S-s2p2d2f1 Ni_CA13S
O  O5.0-s2p2d1      O_CA13
Definition.of.Atomic.Species>
```

各軌道の  $U$  と  $J$  の値 (eV 単位) はキーワード「Hubbard.U.values」により次の様に指定されます。

```
<Hubbard.U.values # eV
Ni 1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 5.0 2d 0.0 1f 0.0
O   1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.0
Hubbard.U.values>
```

```
<Hund.J.values # eV
Ni 1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.9 2d 0.0 1f 0.0
O   1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.0
Hund.J.values>
```

$U$  ( $J$ ) の指定は <Hubbard.U.values (<Hund.J.values)> から開始し、記述の最後は Hubbard.U.values (<Hund.J.values>) とします。全ての基底関数に対して、上記の形式で eV 単位で  $U$  と  $J$  を与えて下さい。「1s」や「2s」は第 1  $s$  軌道と第 2  $s$  軌道を意味します。「1s」の後ろの数値が第 1  $s$  軌道の  $U$  (または  $J$ ) の値です。同一の規則が  $p$  軌道や  $d$  軌道に適用されます。もし scf.DFTU.Type=1 ならば、 $U$  の値のみが読まれ、その際には  $J$  の値を指定する必要はありません。

### 37.1.2 二重計算に対する補正項の選択

二重計算 (double-counting, DC) に対する補正は DFT+ $U$  法を含む「埋め込み」法において共通して生じるものです。scf.DFTU.Type=2 を用いた場合、DC 項は以下のキーワードで指定しなければなりません。

```
scf.dc.Type          cFLL    # sFLL|sAMF|cFLL|cAMF, default=sFLL
```

上記では「cFLL」の DC 項が選択されます。「sFLL」、「sAMF」、「cFLL」、「cAMF」の指定において「c」と「s」はそれぞれ電荷 (スピン非分極) 密度とスピン密度 LDA/GGA を用いた密度汎関数法を意味します。そして、「FLL」と「AMF」はダブルカウント項の扱いにおける完全局在極限 (fully localized limit、FLL) と周辺平均場 (around mean-field、AMF) に対応します。各手法の詳細な定義と振る舞いは参考文献 [21, 22] を参照してください。cFLL または cAMF の DC 項が選択された場合、LDA (または GGA) のスピン密度交換相関ポテンシャルが SCF ループ中で考慮されないことに注意してください [21, 22]。scf.dc.Type=sFLL は Liechtenstein ら [26] によって提案された形式に対応します。単純化回転不変形式 (scf.DFTU.Type=1) が指定された際には sFLL の DC 項が暗黙に仮定されていることにも注意してください。どの種類の DFT+ $U$  法が指定されたのか、標準出力における以下のメッセージ (SCF ループの開始前) から確認することができます。

```
For scf.DFTU.Type=1,
```

```
*****
                DFT+U Type and DC
*****
                scf.DFTU.Type: 1(Simplified)
```

```
For scf.DFTU.Type=2 and scf.dc.Type=cFLL,
```

```
*****
                DFT+U Type and DC
*****
                scf.DFTU.Type: 2(General)   scf.dc.Type: cFLL
```

DFT+ $U$  計算例として、NiO 固体の状態密度を図 34 に示します。この計算では Ni の第 1  $d$  軌道に対して  $U = 5$  eV とし、また 0.5 もしくは 1.0 eV の  $J$  値を割り当てました。入力ファイルはディレクトリ「work」中の「NiO-cFLL.dat」と「NiO-sFLL.dat」です。 $d$  軌道への  $U$  の導入によりギャップが大きくなり、また scf.dc.Type の選択に応じて  $J$  の与える効果が異なっています。各軌道の占有数はファイル「System.Name.out」に出力されます。この出力形式はタイトル「Occupation Number in LDA+ $U$ 」で始まる軌道分解されたマリケン占有数のそれと同じです。NiO 固体 (「scf.dc.Type=cFLL」、 $U = 5$  eV、 $J = 0.5$  eV) の計算例を以下に示します。

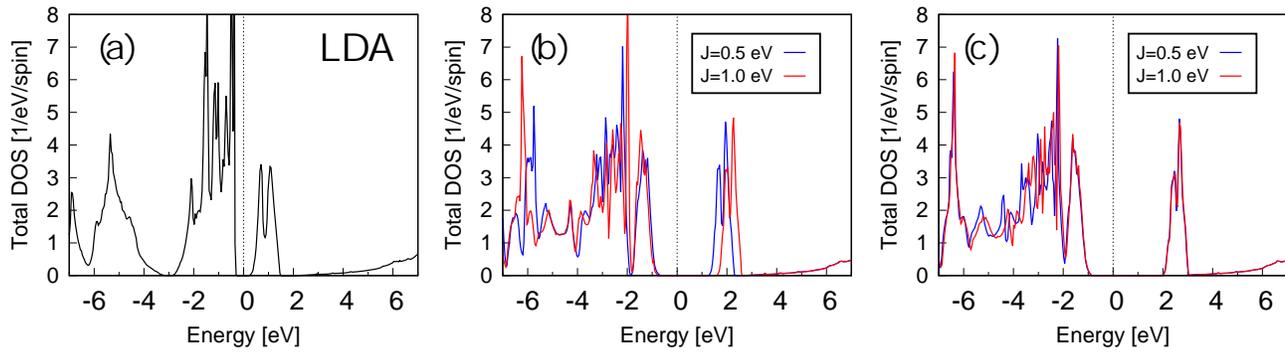


Figure 34: NiO における up スピンの状態密度。(a) LDA, (b) 「scf.dc.Type=cFLL」での DFT+ $U$ 、(c) 「scf.dc.Type=sFLL」での DFT+ $U$ 。 $U$  は 5 eV に固定。青色の線は  $J = 0.5$  eV、赤色の線は  $J = 1.0$  eV。

```
*****
*****
```

Occupation Number in LDA+U and Constraint DFT

Eigenvalues and eigenvectors for a matrix consisting  
of occupation numbers on each site

```
*****
*****
```

1 Ni

spin= 0

Sum = 8.708572022602

|                                  |   | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       |
|----------------------------------|---|---------|---------|---------|---------|---------|---------|---------|---------|
| Individual                       |   | -0.0041 | 0.0012  | 0.0012  | 0.0022  | 0.0040  | 0.0040  | 0.0044  | 0.0064  |
| s                                | 0 | 0.1792  | -0.0008 | -0.0000 | 0.0015  | -0.0000 | 0.0003  | 0.0124  | -0.0000 |
| s                                | 1 | -0.9756 | 0.0052  | 0.0000  | 0.0026  | 0.0000  | -0.0041 | -0.1251 | 0.0000  |
| px                               | 0 | 0.0006  | 0.0007  | -0.0012 | -0.0123 | 0.0003  | 0.0006  | -0.0033 | -0.0000 |
| py                               | 0 | 0.0006  | -0.0013 | -0.0000 | -0.0122 | 0.0000  | 0.0000  | -0.0033 | 0.0000  |
| pz                               | 0 | 0.0006  | 0.0007  | 0.0012  | -0.0123 | -0.0003 | 0.0006  | -0.0033 | 0.0000  |
| px                               | 1 | 0.0091  | 0.0053  | -0.0095 | -0.0867 | 0.0205  | 0.0152  | -0.0206 | 0.0026  |
| py                               | 1 | 0.0093  | -0.0116 | -0.0000 | -0.0870 | -0.0000 | -0.0207 | -0.0236 | -0.0000 |
| pz                               | 1 | 0.0091  | 0.0052  | 0.0095  | -0.0867 | -0.0205 | 0.0152  | -0.0206 | -0.0026 |
| d3z <sup>2</sup> -r <sup>2</sup> | 0 | 0.0002  | 0.0348  | 0.0604  | -0.0000 | -0.0020 | 0.0012  | 0.0001  | -0.0005 |
| dx <sup>2</sup> -y <sup>2</sup>  | 0 | 0.0004  | 0.0604  | -0.0348 | -0.0000 | 0.0011  | 0.0020  | 0.0001  | 0.0003  |
| dxy                              | 0 | -0.0001 | 0.0007  | 0.0012  | 0.0151  | 0.0367  | -0.0218 | 0.0097  | -0.0003 |

|                                    |   |         |         |         |         |         |         |         |         |
|------------------------------------|---|---------|---------|---------|---------|---------|---------|---------|---------|
| dxz                                | 0 | -0.0006 | -0.0015 | -0.0000 | 0.0167  | 0.0000  | 0.0417  | 0.0112  | 0.0000  |
| dyz                                | 0 | -0.0001 | 0.0007  | -0.0012 | 0.0151  | -0.0367 | -0.0218 | 0.0097  | 0.0003  |
| d3z <sup>2</sup> -r <sup>2</sup>   | 1 | -0.0025 | -0.4966 | -0.8626 | -0.0006 | 0.0295  | -0.0174 | -0.0006 | 0.0056  |
| dx <sup>2</sup> -y <sup>2</sup>    | 1 | -0.0042 | -0.8625 | 0.4967  | -0.0010 | -0.0170 | -0.0301 | -0.0010 | -0.0033 |
| dxy                                | 1 | 0.0136  | -0.0160 | -0.0276 | -0.5343 | -0.7016 | 0.4220  | -0.1326 | 0.0055  |
| dxz                                | 1 | 0.0225  | 0.0332  | 0.0000  | -0.5657 | -0.0000 | -0.7918 | -0.1607 | -0.0000 |
| dyz                                | 1 | 0.0136  | -0.0161 | 0.0275  | -0.5343 | 0.7016  | 0.4219  | -0.1325 | -0.0055 |
| f5z <sup>2</sup> -3r <sup>2</sup>  | 0 | -0.0029 | 0.0032  | 0.0065  | -0.0804 | -0.0514 | 0.0334  | -0.0282 | -0.0069 |
| f5xz <sup>2</sup> -xr <sup>2</sup> | 0 | 0.0017  | -0.0304 | -0.0148 | 0.0467  | -0.0113 | -0.0653 | 0.0174  | -0.4673 |
| f5yz <sup>2</sup> -yr <sup>2</sup> | 0 | 0.0013  | 0.0057  | -0.0294 | 0.0479  | 0.0517  | 0.0341  | 0.0272  | 0.4428  |
| fzx <sup>2</sup> -zy <sup>2</sup>  | 0 | -0.0001 | -0.0360 | 0.0237  | -0.0031 | -0.0256 | -0.0567 | 0.0001  | 0.5857  |
| fxyz                               | 0 | 0.1218  | -0.0003 | -0.0000 | 0.2573  | 0.0000  | 0.0172  | -0.9581 | 0.0000  |
| fx <sup>3</sup> -3*xy <sup>2</sup> | 0 | -0.0023 | -0.0195 | -0.0197 | -0.0655 | 0.0563  | -0.0083 | -0.0223 | -0.3532 |
| f3yx <sup>2</sup> -y <sup>3</sup>  | 0 | 0.0017  | 0.0072  | 0.0228  | 0.0618  | -0.0401 | 0.0441  | 0.0352  | -0.3430 |
|                                    |   | 9       | 10      | 11      | 12      | 13      | 14      | 15      | 16      |
| Individual                         |   | 0.0116  | 0.0117  | 0.0207  | 0.0207  | 0.0238  | 0.0972  | 0.1112  | 0.1114  |
| s                                  | 0 | -0.0003 | -0.0000 | 0.0000  | -0.0005 | -0.0075 | -0.0206 | -0.0000 | -0.0000 |
| s                                  | 1 | 0.0001  | 0.0000  | 0.0000  | -0.0013 | 0.0076  | 0.0102  | 0.0000  | -0.0000 |
| px                                 | 0 | -0.0005 | 0.0006  | -0.0024 | 0.0014  | 0.0043  | -0.0270 | 0.0291  | 0.0170  |
| py                                 | 0 | 0.0006  | 0.0000  | 0.0000  | -0.0027 | 0.0044  | -0.0279 | -0.0000 | -0.0338 |
| pz                                 | 0 | -0.0005 | -0.0006 | 0.0024  | 0.0014  | 0.0043  | -0.0270 | -0.0291 | 0.0171  |
| px                                 | 1 | 0.0229  | -0.0402 | 0.1442  | -0.0832 | -0.1073 | 0.5479  | -0.6901 | -0.4038 |
| py                                 | 1 | -0.0437 | 0.0000  | -0.0005 | 0.1632  | -0.1127 | 0.5594  | 0.0003  | 0.7916  |
| pz                                 | 1 | 0.0229  | 0.0402  | -0.1437 | -0.0841 | -0.1073 | 0.5478  | 0.6898  | -0.4043 |
| d3z <sup>2</sup> -r <sup>2</sup>   | 0 | 0.0053  | 0.0093  | 0.0012  | 0.0006  | -0.0001 | 0.0003  | -0.0202 | 0.0115  |
| dx <sup>2</sup> -y <sup>2</sup>    | 0 | 0.0092  | -0.0053 | -0.0007 | 0.0011  | -0.0002 | 0.0006  | 0.0117  | 0.0199  |
| dxy                                | 0 | -0.0033 | -0.0056 | -0.0049 | -0.0032 | -0.0237 | 0.0916  | 0.0095  | -0.0067 |
| dxz                                | 0 | 0.0069  | -0.0000 | -0.0000 | 0.0054  | -0.0236 | 0.0915  | 0.0000  | 0.0102  |
| dyz                                | 0 | -0.0033 | 0.0056  | 0.0049  | -0.0031 | -0.0237 | 0.0916  | -0.0095 | -0.0067 |
| d3z <sup>2</sup> -r <sup>2</sup>   | 1 | -0.0241 | -0.0404 | -0.0230 | -0.0138 | 0.0011  | -0.0001 | 0.0089  | -0.0052 |
| dx <sup>2</sup> -y <sup>2</sup>    | 1 | -0.0418 | 0.0233  | 0.0134  | -0.0238 | 0.0018  | -0.0002 | -0.0051 | -0.0090 |
| dxy                                | 1 | 0.0224  | 0.0367  | 0.0645  | 0.0399  | 0.1012  | -0.0657 | -0.0086 | 0.0058  |
| dxz                                | 1 | -0.0489 | 0.0000  | 0.0002  | -0.0737 | 0.1010  | -0.0652 | -0.0000 | -0.0096 |
| dyz                                | 1 | 0.0224  | -0.0367 | -0.0648 | 0.0395  | 0.1011  | -0.0657 | 0.0086  | 0.0058  |
| f5z <sup>2</sup> -3r <sup>2</sup>  | 0 | 0.0928  | 0.1648  | -0.6676 | -0.3997 | -0.5498 | -0.1226 | -0.1505 | 0.0868  |
| f5xz <sup>2</sup> -xr <sup>2</sup> | 0 | 0.4854  | 0.4030  | -0.3328 | 0.3674  | 0.3359  | 0.0744  | -0.0944 | -0.0506 |
| f5yz <sup>2</sup> -yr <sup>2</sup> | 0 | 0.1112  | 0.6352  | 0.1497  | -0.4674 | 0.3502  | 0.0772  | -0.0023 | 0.1046  |
| fzx <sup>2</sup> -zy <sup>2</sup>  | 0 | 0.6859  | -0.3821 | -0.0991 | 0.1577  | -0.0010 | -0.0008 | 0.0028  | 0.0032  |
| fxyz                               | 0 | 0.0052  | 0.0000  | 0.0000  | -0.0109 | 0.0195  | 0.0064  | 0.0000  | 0.0007  |
| fx <sup>3</sup> -3*xy <sup>2</sup> | 0 | 0.4934  | 0.1037  | 0.5899  | -0.2158 | -0.4352 | -0.0974 | 0.1173  | 0.0705  |
| f3yx <sup>2</sup> -y <sup>3</sup>  | 0 | 0.1435  | -0.4920 | -0.1130 | -0.6043 | 0.4520  | 0.0997  | 0.0019  | 0.1351  |

|            |   | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      |
|------------|---|---------|---------|---------|---------|---------|---------|---------|---------|
| Individual |   | 0.2342  | 0.9866  | 0.9949  | 0.9950  | 1.0070  | 1.0070  | 1.0101  | 1.0101  |
| s          | 0 | 0.9835  | -0.0030 | -0.0001 | 0.0000  | -0.0000 | 0.0001  | 0.0003  | 0.0000  |
| s          | 1 | 0.1796  | -0.0015 | -0.0000 | 0.0000  | 0.0000  | -0.0000 | -0.0000 | -0.0000 |
| px         | 0 | -0.0023 | -0.5138 | -0.3934 | -0.6753 | -0.0494 | -0.0317 | 0.1133  | -0.2016 |
| py         | 0 | -0.0021 | -0.5218 | 0.7787  | -0.0000 | -0.0000 | 0.0587  | -0.2264 | -0.0000 |
| pz         | 0 | -0.0023 | -0.5138 | -0.3933 | 0.6753  | 0.0494  | -0.0317 | 0.1132  | 0.2017  |
| px         | 1 | 0.0092  | -0.0625 | -0.0195 | -0.0329 | 0.0012  | 0.0006  | -0.0049 | 0.0083  |
| py         | 1 | 0.0095  | -0.0631 | 0.0376  | -0.0000 | 0.0000  | -0.0016 | 0.0097  | 0.0000  |
| pz         | 1 | 0.0092  | -0.0625 | -0.0195 | 0.0329  | -0.0012 | 0.0006  | -0.0049 | -0.0083 |
| .....      |   |         |         |         |         |         |         |         |         |
| ...        |   |         |         |         |         |         |         |         |         |

各原子サイトの占有数行列を対角化することで、固有値と固有ベクトルが得られます。占有数行列の固有値は物理的には局在軌道上の電子占有数となります。また固有ベクトルがその局在軌道に対応しています。

### 37.1.3 軌道分極

DFT+*U* 汎関数は、軌道占有の自由度に複数の極値を持っており、SCF 計算の結果は初期占有数近傍に存在する局所的な極値に収束することがあります。そのため収束した電子状態は最低エネルギーを持った基底状態であるとは限りません。軌道分極を持つ基底状態を見出すために、次のフラグによって軌道分極を明示的に促進することが可能です。

#### コリニア計算の場合

```
<Atoms.SpeciesAndCoordinates # Unit=AU
  1 Ni 0.0 0.0 0.0 10.0 6.0 on
  2 Ni 3.94955 3.94955 0.0 6.0 10.0 on
  3 O 3.94955 0.0 0.0 3.0 3.0 on
  4 O 3.94955 3.94955 3.94955 3.0 3.0 on
Atoms.SpeciesAndCoordinates>
```

#### ノンコリニア計算の場合

```
<Atoms.SpeciesAndCoordinates # Unit=AU
  1 Ni 0.0 0.0 0.0 10.0 6.0 40.0 10.0 0 on
  2 Ni 3.94955 3.94955 0.0 6.0 10.0 40.0 10.0 0 on
  3 O 3.94955 0.0 0.0 3.0 3.0 10.0 40.0 0 on
  4 O 3.94955 3.94955 3.94955 3.0 3.0 10.0 40.0 0 on
Atoms.SpeciesAndCoordinates>
```

軌道分極を促進させる場合には、最後の列で「on」と指定して下さい。通常の計算を行う場合には「off」と指定します。軌道分極の促進は各原子サイト毎に行われるため、全ての原子に対して「on」もしくは「off」を指定しなければなりません。指定を明示的に行わない場合には、「off」が設定されます。「on」の設定がされた場合は、最初の数回のSCFステップにおいて軌道分極が促進され、それ以降に続くSCFステップでは通常の計算が行われます。この取扱いにより軌道分極が促進され、多くの場合で基底状態が得られますが、逆に不安定な方向への収束を導く可能性もあり、万能な方法ではありません。詳細は文献 [20] を参照してください。また `scf.dc.Type=cFLL` と `cAMF` を使用する際には軌道分極のフラグの使用は推奨されません。

## 37.2 付加機能

標準的な DFT+ $U$  計算は上記で言及したキーワードで実施されます。ここでは、OpenMX で利用できる付加機能を説明します。これらの機能は `scf.DFTU.Type=2` の場合のみに利用できます。

### 37.2.1 二つのスレーター積分の比の調整 ( $F^4/F^2$ )

`scf.DFTU.Type=2` による一般化 DFT+ $U$  法では入力値  $U$  と  $J$  を用いてクーロン相互作用テンソルが生成されます。原子の球対称性を仮定し、スレーター積分 ( $F^0, F^2, F^4, \dots$ ) と Racah-Wigner 係数からこのテンソルが計算されます。 $d$  軌道に対する一般的な DFT+ $U$  法の取扱いでは比  $F^4/F^2$  が必要となり、通常は  $F^4/F^2 = 0.625$  が使用されます。ただし、この値は球対称な原子環境のみで有効であり、固体中ではこの値から逸脱する可能性があります。キーワード「`scf.Slater.Ratio`」により、ユーザーは各自の目的に応じて  $F^4/F^2$  を調整できます。

```
scf.Slater.Ratio          0.75      # default=0.625
```

### 37.2.2 入力パラメタ $U$ からの $J$ と $F^4/F^2$ の見積もり

Yukawa 型の遮蔽クーロンポテンシャルを仮定することで入力の  $U$  の値から  $J$  と  $F^4/F^2$  を見積もることができます。この目的のためにキーワード「`scf.Yukawa`」を「on」と設定します。

```
scf.Yukawa                on        # default=off
```

`scf.Yukawa=on` と設定することで、`<Hubbard.U.values ... Hubbard.U.values>` 中にある  $U$  の値だけが読み込まれ、`<Hund.J.values ... Hund.J.values>` 中にある  $J$  の値と `scf.Slater.Ratio` は無視されます。代わりに、 $J$  と  $F^4/F^2$  の値は入力の  $U$  の値に対応する Thomas-Fermi 遮蔽長を見積もることで自動的に生成されます [21, 22]。例として、以下のように定義された基底関数があるとします。

```
<Definition.of.Atomic.Species
  Ni Ni6.0S-s2p2d2f1 Ni_CA13S
  O  O5.0-s2p2d1      O_CA13
Definition.of.Atomic.Species>
```

`scf.Yukawa=on` かつ  $U$  の値がキーワード「`Hubbard.U.values`」により、以下の様に設定されるとします。

```
<Hubbard.U.values      # eV
  Ni 1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 6.4 2d 3.0 1f 0.0
```

```
0 1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.0
Hubbard.U.values>
```

この時、SCF ループが開始する前に標準出力に以下のメッセージが表示されます。

```
*****
Calculating Thomas-Fermi screening length
*****

<species: Ni, angular momentum= 2, multiplicity number= 0>
TF-screening-length lambda= 1.340787 1/au
Hubbard U= 6.400000 eV
Hund J= 1.154943 eV
Slater F0= 6.399932 eV
Slater F2= 9.321381 eV
Slater F4= 6.847820 eV
F4/F2= 0.734636

<species: Ni, angular momentum= 2, multiplicity number= 1>
TF-screening-length lambda= 0.271461 1/au
Hubbard U= 3.000000 eV
Hund J= 0.380184 eV
Slater F0= 3.000017 eV
Slater F2= 2.970544 eV
Slater F4= 2.352036 eV
F4/F2= 0.791786
```

これらの  $U$ 、 $J$ 、 $F^4/F^2$  の値は DFT+ $U$  計算の SCF ループ中で用いられます。この計算例で用いられた入力ファイル「NiO-Yukawa.dat」はディレクトリ「work」中に保存されていますので、参考にして下さい。

## 38 ノンコリニアスピン方位に対する制約条件付き DFT

ノンコリニア DFT 法において任意のスピン方位を持つ電子構造を計算するために制約条件付き DFT 法が利用可能です。制約条件付き DFT 法では、スピン方位と初期スピン方位の差がゼロでない限り系のエネルギーにペナルティーを与える汎関数が付加されます [13]。ノンコリニアスピン方位に対する制約条件付き DFT 法は次のキーワードで利用できます。

```
scf.Constraint.NC.Spin      on      # on|on2|off, default=off
scf.Constraint.NC.Spin.v   0.5     # default=0.0(eV)
```

キーワード「scf.Constraint.NC.Spin」を「on」もしくは「on2」とした場合にペナルティ関数が導入されます。「on」の場合、スピンの向きは初期の配向に拘束されますが、スピンの大きさには制約は課されません。全エネルギーの安定化するようにスピンの大きさは自動的に最適化されます。一方、「on2」の場合、スピンの向きと大きさの両方に制約条件が課され、「Atoms.SpeciesAndCoordinates」で指定した初期値から逸脱するとエネルギーにペナルティが加算されます。キーワード「scf.Constraint.NC.Spin.v」は導入した制約の強さを制御する際に使用します。各原子に対して制約を付与する場合には以下の様に指定します。

```
<Atoms.SpeciesAndCoordinates
  1  Cr  0.00000  0.00000  0.00000  7.0  5.0 -20.0  0.0  0.0  0.0  1  off
  2  Cr  0.00000  2.00000  0.00000  7.0  5.0  20.0  0.0  0.0  0.0  1  off
Atoms.SpeciesAndCoordinates>
```

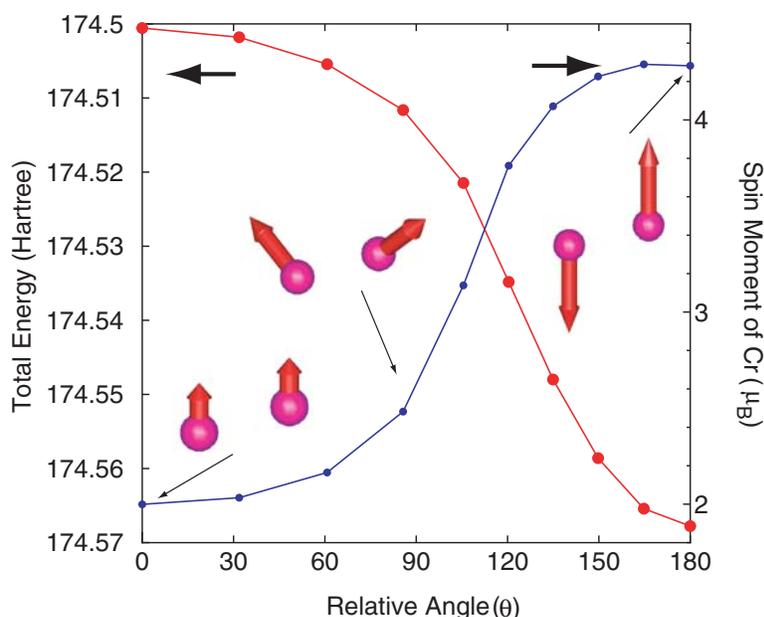


Figure 35: クロム二量体の全エネルギーと Cr 原子サイトの磁気モーメント。結合距離は 2.0 Å。入力ファイルは「work」ディレクトリ中の「Cr2\_CNC.dat」。

12 列目の「1」は制約が課されることを意味し、「0」は制約が課されないことを意味します。また本制約条件付き DFT は「DFT+U」の章で説明した DFT+U 計算と互換しており、同時に両手法を用いることが可

能です。本手法の適用例として、クロム二量体の二つの局所スピンの間の相対角度に対する全エネルギーと磁気モーメントの依存性を図 35 に示します。ディレクトリ「work」の入力ファイル「Cr2\_CNC.dat」を用いてこの計算を再現できます。

制約付き DFT 法を用いてスピン角度の関数として滑らかなエネルギー曲線を計算するための処方箋をここで説明します。多くの場合、局在磁気モーメントは  $3d$  電子のように局在した  $d$  軌道または  $f$  軌道への部分占有状態から生じます。局在軌道への電子の部分占有状態は占有数空間に多くの自由度を持ち、そのために多数の局所解を持っています。このような局所解を避けるため、まず容易軸に沿ったスピン方位から計算を開始するべきです。最初の計算が終了すると、リスタートファイルが生成されます。次の計算ではスピン方位を最初の計算と比較して、僅かに、例えば 30 度ほど回転させます。この際に最初の計算で生成されたリスタートファイルを読み込みます。同じ手順を計算したい全てのスピン回転角度の場合に適用します。各計算では前回のステップで生成されたリスタートファイルを利用して下さい。この段階的なアプローチは局所解の回避に有効な方法です。

## 39 第二変分法: 磁気異方エネルギー (Magnetic Anisotropy Energy: MAE)

前章ではスピン方位を制御する制約付き DFT 法を紹介しました。制約付き DFT 法を用いて磁性系の磁気異方性エネルギーの評価が可能です。この場合、自己無撞着に計算された全エネルギーから磁気異方性エネルギーが計算されます。ただし、その計算コストは大きく、また縮退する局在  $d$  軌道に対しては占有数空間における自由度のために局所解に陥り、正しく計算できない可能性があります。これらの問題を回避する手段の一つは第二変分法を用いることです。この方法ではまず最初にコリニア DFT 計算で SCF 電子密度を計算します。この際にリスタートファイルが生成されます。次にこのリスタートファイルを用いてノンコリニア DFT 法の枠組みで一回だけの対角化を行います。その際にスピン軌道相互作用 (SOI) を考慮し、対角化を実行します。この計算手法は第二変分法と呼ばれています。第二変分法は Harris 汎関数 [14] に基づくため、スピン回転とスピン軌道相互作用による摂動の効果はバンドエネルギーのみに生じます。全エネルギーはバンドエネルギーと二重計算項の和で記述することが可能ですが、Harris 汎関数を用いれば一回だけの対角化により、入力電子密度と SCF 電子密度の差の二次のオーダーの誤差で全エネルギーが評価できます。その際に二重計算項は入力電子密度のみから計算されるため、スピン回転とスピン軌道相互作用による摂動エネルギーはバンドエネルギーのみに含まれることとなります。従ってエネルギーの比較を行う際にはバンドエネルギーのみに着目すべきです。Harris 汎関数における二重計算項はスピン回転角に依存しません。OpenMX の出力では変化するように見えますが、これは各エネルギー項が入力電子ではなく出力電子によって計算されるためです (出力に惑わされないよう注意して下さい)。第二変分法を用いて、最初にコリニア DFT 法で強磁性状態を計算し、SCF 電子密度の結果を得ます。次に、SCF 電子密度が保存されたリスタートファイルを用いてノンコリニア DFT の枠組みでスピン軌道相互作用を考慮し、一回だけの対角化が実行します。リスタートファイルは以下のキーワードで読み込まれます。

```
scf.restart.filename      FePt
scf.restart               c2n
```

キーワード「scf.restart.filename」を用いて読み込むリスタートファイルが指定されます。キーワード「scf.restart」の「c2n」により、コリニア DFT 計算で生成されたリスタートファイルがノンコリニア DFT 計算用に変換されます。例として、「work」で利用可能な入力ファイル「FePt.dat」を用いて最初のコリニア計算を実行し、リスタートファイルを生成します。二回目の計算では、スピン方位を以下のキーワードで指定します。

```
scf.Restart.Spin.Angle.Theta  90.0
scf.Restart.Spin.Angle.Phi    0.0
```

この二つのキーワードはスピン方位のオイラー角 ( $\theta, \phi$ ) を指定します。実空間上の全てのグリッド上のスピン方位がオイラー角で指定された同一方向に並ぶこととなります。そのため、第二変分法による磁気異方エネルギー (MAE) の評価は強磁性体だけに適用可能であることに注意して下さい。第二変分法が MAE の評価にどのように適用されるのか、ここで一つの例を示します。図 36 は  $L1_0$  FePt パルクでのスピン回転角度に対する MAE 曲線を示しています。「Full SCF」の結果は 38 章で説明した制約付き DFT 法で得られました。入力ファイル「FePt-NC-SCF.dat」がディレクトリ「work」中に保存されていますので、ユーザーは同一の計算を実行することが出来ます。スピン軌道相互作用を考慮したノンコリニア DFT 計算により、それぞれのスピン回転角度において、完全な自己無撞着計算が実行されました。ここで電子温度は 300 K で  $k$  点数は  $17^3$  です。結果として、「Full SCF」では 2.78 meV/f.u. の MAE が得られました。第二変分法による二つの計算結果もまた図 36 に示されています。最初にディレ

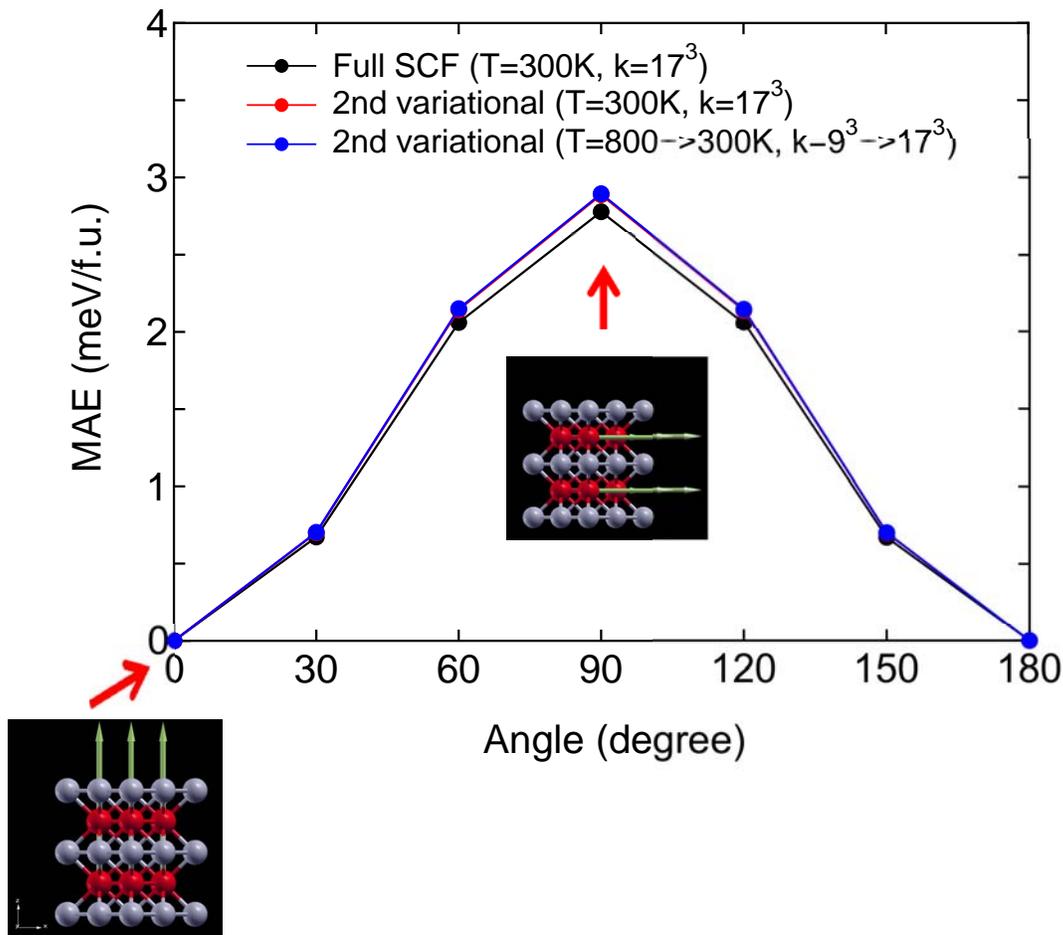


Figure 36:  $L1_0$  FePt バルクでのスピン回転角度に対する磁気異方性エネルギーの MAE 曲線。「Full SCF」に用いた入力ファイルは「FePt-NC-SCF.dat」、また入力ファイル「FePt.dat」と「FePt-NC.dat」は第二変分法による二つの計算に用いられました。これらの入力ファイルはディレクトリ「work」中にあります。

クトリ「work」の入力ファイル「FePt.dat」を用いてコリニア計算を実行し、リスタートファイルを生成します。次にディレクトリ「work」中の入力ファイル「FePt-NC.dat」を用いて一度だけの対角化を実行します。第二変分法による二つの計算において、一つは「Full SCF」と同一の電子温度 (300K) と同一の  $k$  点数が用いられた場合の結果です。この時、MAE は 2.89 meV/f.u. となりました。もう一方では最初の SCF 計算と二番目の計算の電子温度 ( $k$  点格子) にそれぞれ 800K ( $9^3$ ) と 300 K ( $17^3$ ) が用いられた場合の結果です。この場合は MAE は 2.90 meV/f.u. となりました。第二変分法による二つの計算はほとんど同じ結果を与えるので、SCF 電子密度を生成するためのコリニア計算とノンコリニア計算 (スピン軌道相互作用を含む) において異なる電子温度と  $k$  点数の使用が許容されるでしょう。SCF 計算に対しては高い電子温度と少数の  $k$  点数を割り当てることで計算効率の向上が期待でき、また二番目のノンコリニア計算に対しては低い電子温度と多数の  $k$  点数を用いて計算精度を向上させることが可能です。第二変分法では出力ファイルの「Utot」の代わりに「Uele」に着目すべきであることを再度、注意しておきます。第二変分法は Harris 汎関数に基づくため、MAE は「Uele」により評価されるべきです。他方「full SCF」での MAE は全エネルギーの「Utot」を用いて評価されます。MAE の評価におけ

る「full SCF」と第二変分法との差は約 0.1 meV/f.u. です。したがって、第二変分法は精度と効率の観点から MAE の評価への効率的なアプローチであると考えられます。

第二変分法は MAE の評価だけでなく、スピン軌道相互作用がどのようにバンド構造を変化させるか調べる際にも活用できます。つまりノンコリニア DFT 法による直接的な SCF 計算が困難な大規模系のバンド構造の解析にも適用できるでしょう。

## 40 Zeeman 項

スピンおよび軌道磁気モーメントに対して Zeeman 項を導入することができます。

### 40.1 スピン磁気モーメントに対する Zeeman 項

次のキーワードによって、スピン磁気モーメントに対する Zeeman 項が一樣磁場との相互作用として導入されます。

```
scf.NC.Zeeman.Spin          on          # on|off, default=off
scf.NC.Mag.Field.Spin       100.0       # default=0.0 (Tesla)
```

スピン磁気モーメントに対し Zeeman 項を導入する場合、キーワード「scf.NC.Zeeman.Spin」を「on」にします。一樣磁場の強度は、キーワード「scf.NC.Mag.Field.Spin」で、テスラ単位で指定します。本手法は制約条件付き DFT 法として利用することが可能で、各原子毎に磁場の方向を指定することができます。スピン磁気モーメントに対する磁場の方位は、キーワード「Atoms.SpeciesAndCoordinates」によって次のように指定します。

```
<Atoms.SpeciesAndCoordinates
 1 Sc  0.000  0.000  0.000   6.6 4.4  10.0 50.0  160.0 20.0  1  on
 2 Sc  2.000  0.000  0.000   6.6 4.4   80.0 50.0  160.0 20.0  1  on
Atoms.SpeciesAndCoordinates>
```

スピン磁気モーメントに対する磁場の方位を指定するために、第 8 列と 9 列で Euler 角 ( $\theta$ ,  $\phi$ ) を指定します。第 12 列は制約条件に対するスイッチです。「1」は制約条件有り、「0」は無しを意味しています。各原子サイトに対して印加磁場の方位を制御できるため、この方法は様々なノンコリニアスピン構造を研究する手段となります。ただし、キーワード「scf.NC.Zeeman.Spin」と「scf.Constraint.NC.Spin」は互いに両立できないことに注意して下さい。従って、「scf.NC.Zeeman.Spin」が「on」の場合、キーワード「scf.Constraint.NC.Spin」は次のように「off」としなければなりません。

```
scf.Constraint.NC.Spin      off          # on|off, default=off
```

この章で説明したスピン磁気モーメントの方位に対する Zeeman 項と、前章で説明した制約法は、どちらもスピン方位を制御する方法ですが、Zeeman 項を付加した場合には、スピン磁気モーメントの大きさが増加する可能性があることに注意して下さい。

### 40.2 軌道磁気モーメントに対する Zeeman 項

次のキーワードによって、軌道磁気モーメントに対する Zeeman 項が一樣磁場との相互作用として導入されます。

```
scf.NC.Zeeman.Orbital       on          # on|off, default=off
scf.NC.Mag.Field.Orbital    100.0       # default=0.0(Tesla)
```

軌道磁気モーメントに対して Zeeman 項を組み込む場合、キーワード「scf.NC.Zeeman.Orbital」を「on」にします。一樣磁場の強度は、キーワード「scf.NC.Mag.Field.Orbital」で、テスラ単位で指定できます。この方法は制約条件付き DFT 法として利用することが可能で、各原子毎に磁場の方向を指定すること

ができます。軌道磁気モーメントに対する磁場の方位は、キーワード「Atoms.SpeciesAndCoordinates」によって次のように指定します。

```
<Atoms.SpeciesAndCoordinates
 1 Sc 0.000 0.000 0.000 6.6 4.4 10.0 50.0 160.0 20.0 1 on
 2 Sc 2.000 0.000 0.000 6.6 4.4 80.0 50.0 160.0 20.0 1 on
Atoms.SpeciesAndCoordinates>
```

軌道磁気モーメントに対する磁場の方位を指定するために、第10列と11列でEuler角 ( $\theta$ 、 $\phi$ ) を指定します。第12列は制約条件に対するスイッチです。「1」は制約条件有り、「0」は無しを意味しています。各原子サイトに対して印加磁場の方位を制御できるため、この方法は様々なノンコリニアスピン構造を研究する手段となります。また軌道磁気モーメントに対する磁場の方位はスピン磁気モーメントの場合と同じである必要はありません。この機能を利用し、様々な磁気構造を計算することが可能です。

## 41 Berry 位相による巨視的分極の計算

Berry 位相の方法を用いて、バルクの巨視的電気分極を計算することができます [15]。例として、塩化ナトリウム中のナトリウム原子の Born 有効電荷を巨視的電気分極から計算する手順を説明します。

### (1) SCF 計算

最初に、「work」ディレクトリ中にある入力ファイル「NaCl.dat」を用いて、通常の SCF 計算を実行します。この際に、キーワード「HS.fileout」をオンにします。

```
HS.fileout          on      # on|off, default=off
```

SCF 計算が正常に完了すると、「work」ディレクトリ中に出力ファイル「nacl.scfout」が生成されます。

### (2) 巨視的分極の計算

巨視的分極はポストプロセスのプログラム「polB」を用いて計算します。この際に「nacl.scfout」が入力データとなります。まず「source」ディレクトリにおいて、「polB」を次のようにコンパイルします。

```
% make polB
```

コンパイルが正常に完了すると、「work」ディレクトリ中に実行形式ファイル「polB」が生成されます。次に、「work」ディレクトリに移動し、次のように実行します。

```
% polB nacl.scfout
もしくは
% polB nacl.scfout < in > out
```

後者の場合、テキストファイル「in」には次のデータが保存されています。

```
9 9 9
1 1 1
```

前者の場合、次のように会話形式でプログラムから質問されます。

```
*****
*****
polB:
code for calculating the electric polarization of bulk systems
Copyright (C), 2006-2007, Fumiyuki Ishii and Taisuke Ozaki
This is free software, and you are welcome to
redistribute it under the constitution of the GNU-GPL.
*****
*****
```

```
Read the scfout file (nacl.scfout)
Previous eigenvalue solver = Band
atomnum                    = 2
```

```

ChemP                = -0.156250000000 (Hartree)
E_Temp               = 300.000000000000 (K)
Total_SpinS         = 0.000000000000 (K)
Spin treatment       = collinear spin-unpolarized

```

```

r-space primitive vector (Bohr)
tv1= 0.000000  5.319579  5.319579
tv2= 5.319579  0.000000  5.319579
tv3= 5.319579  5.319579  0.000000

```

```

k-space primitive vector (Bohr^-1)
rtv1= -0.590572  0.590572  0.590572
rtv2= 0.590572 -0.590572  0.590572
rtv3= 0.590572  0.590572 -0.590572

```

Cell\_Volume=301.065992 (Bohr<sup>3</sup>)

Specify the number of grids to discretize reciprocal a-, b-, and c-vectors  
(e.g 2 4 3) 9 9 9

```

k1  0.00000  0.11111  0.22222  0.33333  0.44444  ...
k2  0.00000  0.11111  0.22222  0.33333  0.44444  ...
k3  0.00000  0.11111  0.22222  0.33333  0.44444  ...

```

Specify the direction of polarization as reciprocal a-, b-, and c-vectors  
(e.g 0 0 1 ) 1 1 1

逆格子ベクトルの離散化グリッド数と分極の方向を指定した後に、計算が以下のように進行します。

```

calculating the polarization along the a-axis ....
The number of strings for Berry phase : AB mesh=81

```

```

calculating the polarization along the a-axis .... 1/ 82
calculating the polarization along the a-axis .... 2/ 82
.....
...

```

```

*****
Electric dipole (Debye) : Berry phase
*****

```

Absolute dipole moment 163.93373639

| Background | Core | Electron | Total |
|------------|------|----------|-------|
|------------|------|----------|-------|

|    |             |             |             |             |
|----|-------------|-------------|-------------|-------------|
| Dx | -0.00000000 | 94.64718996 | -0.00000338 | 94.64718658 |
| Dy | -0.00000000 | 94.64718996 | -0.00000283 | 94.64718713 |
| Dz | -0.00000000 | 94.64718996 | -0.00000317 | 94.64718679 |

\*\*\*\*\*

Electric polarization (muC/cm<sup>2</sup>) : Berry phase

\*\*\*\*\*

|    | Background  | Core         | Electron    | Total        |
|----|-------------|--------------|-------------|--------------|
| Px | -0.00000000 | 707.66166752 | -0.00002529 | 707.66164223 |
| Py | -0.00000000 | 707.66166752 | -0.00002118 | 707.66164633 |
| Pz | -0.00000000 | 707.66166752 | -0.00002371 | 707.66164381 |

Elapsed time = 77.772559 (s) for myid= 0

$V_c$  を単位セルの体積、 $e$  を素電荷、 $\Delta u_\beta$  を  $\beta$  座標方向の変位、 $\Delta P_\alpha$  を  $\alpha$  座標方向の巨視的分極の変化とすると、Born 有効電荷  $Z_{\alpha\beta}^*$  は次のようにテンソルで定義されます。

$$Z_{\alpha\beta}^* = \frac{V_c \Delta P_\alpha}{|e| \Delta u_\beta}$$

上記の表式に従い、Born 有効電荷を計算する際には、ナトリウム原子の  $x$ 、 $y$ 、 $z$  座標を変化させながら、上の手順を少なくとも 2 回もしくは 3 回実行します。例えば  $x$  座標方向を変位させて分極を計算すると、次の結果が得られます。

Px = 94.39497736 (Debye/unit cell) at x= -0.05 (Ang)

Px = 94.64718658 (Debye/unit cell) at x= 0.0 (Ang)

Px = 94.89939513 (Debye/unit cell) at x= 0.05 (Ang)

従って、ナトリウム原子の Born 有効電荷は以下のように計算されます。

$$\begin{aligned} Z_{xx}^* &= \frac{(94.89939513 - 94.39497736)/(2.54174776)}{0.1/0.529177} \\ &= 1.050 \end{aligned}$$

塩化ナトリウム固体では、Born 電荷のテンソルの非対角項はゼロ、そして  $Z_{xx}^* = Z_{yy}^* = Z_{zz}^*$  となります。表 7 に、ここでの計算値と他の計算値 [116] および実験値 [117] との比較を示します。計算値と実験値が良く一致していることが分かります。巨視的分極の計算はコリニアおよびノンコリニア DFT 法の両者に対してサポートされています。また MPI プロセス数が系の原子数を上回る場合でも、プログラム「polB」は効率的な並列計算が可能です。

Table 7: Berry 位相による巨視的分極の計算から見積もられた NaCl 中の Na の Born 有効電荷。入力ファイルは「work」ディレクトリ中の「NaCl.dat」。比較のため、他の計算値 (FD: 参考文献 [116]) と実験値 (参考文献 [117]) も併記。

|       | OpenMX | FD   | Expt. |
|-------|--------|------|-------|
| $Z^*$ | 1.05   | 1.09 | 1.12  |

## 42 交換結合パラメータ

### 42.1 概要

摂動論に基づく Liechtenstein の方法 [17] を用いて局在したスピン間の交換結合パラメータ  $J_{ij}$  を計算できます。この計算はポストプロセスのプログラム「jx」を用いて実行します。OpenMX Ver 3.9 において本機能はクラスタやバルク系のコリニア DFT 計算のみに有効で、ノンコリニア DFT 計算には一般化されていません。本機能を用いて論文等を出版される場合は文献 [18, 19] を引用して頂けますと幸いです。

このプログラム「jx」は  $J_{ij}$  を計算する以下の三つの方法を提供します。

クラスタ系に対して、本プログラム「jx」は以下の表式に基づき原子サイト  $i$  と  $j$  の間の交換結合定数  $J_{ij}$  を計算します。

$$J_{ij} = \frac{1}{4} \sum_{n,n'} \frac{-f_{n\uparrow} + f_{n'\downarrow}}{\varepsilon_{n\uparrow} - \varepsilon_{n'\downarrow}} \times \sum_{\mu,\nu \in i} \sum_{\mu',\nu' \in j} C_{j\mu',n\uparrow} C_{i\nu,n\uparrow}^* [\hat{P}_i]_{\nu\mu} C_{i\mu,n'\downarrow} C_{j\nu',n'\downarrow}^* [\hat{P}_j]_{\nu'\mu'} \quad (3)$$

$$\hat{P}_i \equiv \hat{H}_{i\uparrow} - \hat{H}_{i\downarrow}, \quad (4)$$

$\varepsilon_{n\sigma}$  と  $C_{n\sigma}$  は Kohn-Sham 方程式の固有値と固有ベクトルです。ここで、波数  $\mathbf{k}$ 、バンドインデックス  $n$ 、スピンインデックス  $\sigma$  となります。また、 $[\hat{P}_i]_{\nu\mu}$  と  $[\hat{P}_j]_{\nu'\mu'}$  はサイト  $i$  と  $j$  のポテンシャル差演算子の部分行列をそれぞれ表します。

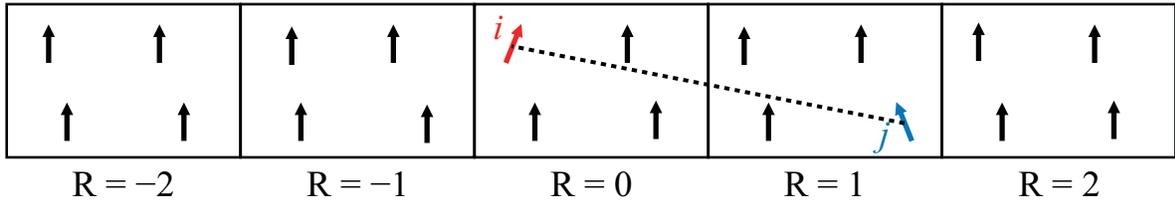
バルク系に対して、本プログラム「jx」は以下の表式に基づきセル  $\mathbf{0}$  と  $\mathbf{R}$  にそれぞれ位置する原子サイト  $i$  と  $j$  の間の交換結合定数  $J_{i\mathbf{0},j\mathbf{R}}$  を計算します。

$$J_{i\mathbf{0},j\mathbf{R}} = \frac{1}{2} \sum_{p=1}^{N_P} \tilde{R}_p \sum_{\mu,\nu \in i} \sum_{\mu',\nu' \in j} \text{Re} \left\{ [\hat{P}_i]_{\nu\mu} G_{i\mu,j\nu'}^+(\downarrow, \tilde{z}_p, \mathbf{R}) [\hat{P}_j]_{\nu'\mu'} G_{j\mu',i\nu}^+(\uparrow, \tilde{z}_p, -\mathbf{R}) \right\}. \quad (5)$$

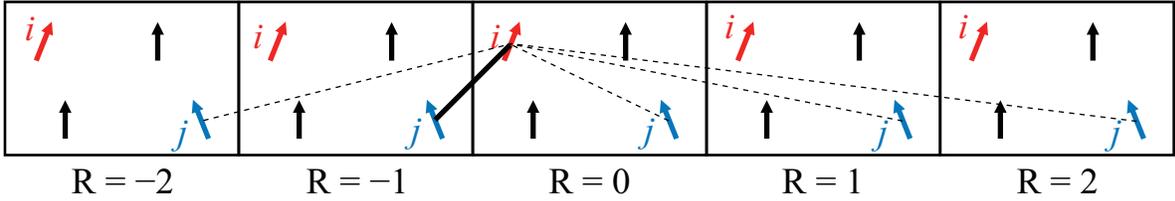
$\tilde{z}_p$  と  $\tilde{R}_p$  は近似されたフェルミ関数の極（複素平面の上側に位置する）と対応する留数です [74]。  $i$  と  $j$  は単位胞での原子のインデックスです。  $\mathbf{R}$  は原子  $j$  が位置するセルのインデックスです。グリーン関数  $G_{j\mu',i\nu}^+(\uparrow, \varepsilon, -\mathbf{R})$  と  $G_{i\mu,j\nu'}^+(\downarrow, \varepsilon, \mathbf{R})$  は以下の式で定義されます。

$$G_{j\mu',i\nu}^+(\uparrow, \varepsilon, -\mathbf{R}) \equiv \int d^3 \left( \frac{ka}{2\pi} \right) e^{i\mathbf{k}\cdot\mathbf{R}} \sum_n \frac{C_{j\mu',n\uparrow}(\mathbf{k}) C_{i\mu,n\uparrow}(\mathbf{k})}{\varepsilon + i\eta - \varepsilon_{n\uparrow}(\mathbf{k})} \quad (6)$$

$$G_{i\mu,j\nu'}^+(\downarrow, \varepsilon, \mathbf{R}) \equiv \int d^3 \left( \frac{ka}{2\pi} \right) e^{-i\mathbf{k}\cdot\mathbf{R}} \sum_{n'} \frac{C_{i\mu,n'\downarrow}(\mathbf{k}) C_{j\nu',n'\downarrow}(\mathbf{k})}{\varepsilon + i\eta - \varepsilon_{n'\downarrow}(\mathbf{k})}. \quad (7)$$



(a) Interaction between individual sites  $i$  and  $j$  located at specific cells.



(b) Interaction between periodic images of sites  $i$  and  $j$ .

Figure 37: 独立なサイト間と周期的なイメージ間での交換結合定数の模式図。

交換結合  $J_{ij}$  の  $R$  に渡る総和を考慮する場合には以下の表式を用いて計算することもできます。

$$\begin{aligned}
 J_{ij} &= \sum_{\mathbf{R}} J_{i0,j\mathbf{R}} \\
 &= \frac{1}{4} \int d^3 \left( \frac{ka}{2\pi} \right) \sum_{n,n'} \frac{-f_{n\uparrow}(\mathbf{k}) + f_{n'\downarrow}(\mathbf{k})}{\varepsilon_{n\uparrow}(\mathbf{k}) - \varepsilon_{n'\downarrow}(\mathbf{k})} \\
 &\quad \times \sum_{\mu,\nu \in i} \sum_{\mu',\nu' \in j} C_{j\mu',n\uparrow}(\mathbf{k}) C_{i\nu,n\uparrow}^*(\mathbf{k}) [\hat{P}_i]_{\nu\mu} C_{i\mu,n'\downarrow}(\mathbf{k}) C_{j\nu',n'\downarrow}^*(\mathbf{k}) [\hat{P}_j]_{\nu'\mu'} . \quad (8)
 \end{aligned}$$

式 (3), (5), (8) の詳細は文献 [18, 19] を参照してください。式 (5) と (8) の取扱いの相違は図 37 に模式的に示されています。図 37 (a) は式 (5) に相当する独立サイト間での相互作用の模式図です。一方、式 (8) に相当する周期イメージ間での相互作用は図 37 (b) に模式的に示されています。後節で説明されるように、実行時にオプション `Flag.PeriodicSum` を指定することでどちらかの計算が選択できます。

## 42.2 Jx の計算

ポストプロセスコード「jx」を用いて OpenMX の出力から  $J_{ij}$  を計算します。まずディレクトリ「source」に移動して、以下のようにコンパイルを実行して下さい。

```
% make jx
```

コンパイルが正常に完了すると、実行ファイル「jx」がディレクトリ「work」中に生成されます。

## 42.3 OpenMX 計算による SCF 計算

$J_{ij}$  計算の前に、`scfout` ファイルを生成するためにコリニア DFT 計算を実行して下さい。scfout ファイルを生成するために、OpenMX の入力ファイル中にキーワード「HS.fileout」を加えます。

```
HS.fileout          on          # on|off, default=off
```

「jx」で信頼のおける計算結果を得るためには比較的小さな基底関数セットを用いることが必要です。現在の実装では使用する擬原子基底関数が良く局在していることが前提となっており、多数の基底関数を含めた場合にはこの前提条件が満たされなくなるからです。例えば、Fe には s2p2d2、Nd には s2p1d1 が基底関数セットとして適切な選択となります。

#### 42.4 Jx の設定ファイルの準備

「jx」を実行するために設定ファイルを用意します。この設定ファイル中に様々な制御パラメータを指定します。設定ファイルは OpenMX の入力ファイルと同じ書式を持っています。

例として「jx.config」を以下に示します。「jx.config」はディレクトリ「work」中に保存されています。

```

Flag.PeriodicSum      off      # default - off
Num.Poles             60
Num.Kgrid             27 27 27
Num.ij.pairs         236
Bunch.ij.pairs       236
<ijpairs.cellid
  1 1 -2 -2 -2
  1 1 -2 -2 -1
  1 1 -2 -2 0
  ...
  ...
  2 2 0 -1 2
  2 2 0 0 -2
  2 2 0 0 -1
ijpairs.cellid>

```

それぞれのキーワードを以下に説明します。

```

Flag.PeriodicSum      off      # default - off

```

このフラグは交換結合をどのように計算するかを決定します。周期系でこのフラグが「on」の場合、本プログラム「jx」は式 (8) に基づいて  $J_{ij}$  を計算します。一方、「off」の場合には式 (5) に基づいて  $J_{i0,jR}$  を計算します。このフラグはクラスター計算では役割を持っていません。クラスター計算では、 $J_{ij}$  は式 (3) から計算されます。

```

Num.Poles             60

```

このキーワードでは式 (5) に現れるフェルミ関数の有限極近似 [74] に対する極の数  $N_P$  を指定します。極の数の増加に従い計算精度が向上しますが、計算時間は極の数に比例して増加します。適切な極の数は系によりますが、bcc 構造の Fe では 0.05meV の精度を達成するにはおよそ 60 程度です。300K の電子温度においては多くの系で 60 個の極で十分な精度が得られますが、一度、御自身でその収束性を確認されることをお勧めします。

```

Num.Kgrid            27 27 27

```

このキーワードは第一 Brillouin 領域の離散化に対する  $k$  点の数を指定します。ここで指定する  $k$  点数は SCF 計算の際と同じ値、もしくはそれ以上の値を指定して下さい。クラスター計算では自動的にガンマ点のみが考慮されますので、このキーワードに役割はありません。

```
Num.ij.pairs          236          # NOTE: Number of ij pairs.
```

このキーワードは計算する交換結合定数  $J$  の数を指定します。この値はキーワード <ijpairs.cellid と d ijpairs.cellid> の間の行数と同数でなければなりません。

```
<ijpairs.cellid
  1 1 -2 -2 -2
  1 1 -2 -2 -1
  1 1 -2 -2 0
  ...
  ...
  2 2 0 -1 2
  2 2 0 0 -2
  2 2 0 0 -1
ijpairs.cellid>
```

このフィールドでは計算する交換結合定数  $J$  のサイト  $i, j$  と  $J_{i0,j}\mathbf{R}$  のセルベクトル  $\mathbf{R} = l_1\mathbf{a}_1 + l_2\mathbf{a}_2 + l_3\mathbf{a}_3$  を指定します。ここで、 $\mathbf{a}_1, \mathbf{a}_2$  及び  $\mathbf{a}_3$  は OpenMX 入力での単位格子ベクトルです。このフィールドのデータの並び方は  $i j l_1 l_2 l_3$  の順です。クラスターあるいは周期イメージ計算では以下の様に別のフィールドを指定します。

```
<ijpairs.nocellid
  1 1
  1 2
ijpairs.nocellid>
```

これらの  $J_{ij}$  計算ではセルベクトルが現れないため、明示的に指定する必要がありません。

```
Bunch.ij.pairs       236          # default=Num.ij.pairs
```

既定の設定ではメモリ消費が多くなり、計算が出来ない場合があります。メモリ消費量を低減させるために、上記のキーワードが利用可能です。この値は Num.ij.pairs と同じか、より小さい値であるべきです。小さい「Bunch.ij.pairs」を用いるとメモリ消費量は低減されますが、計算時間が長くなる傾向があります。

## 42.5 Jx の実行と MPI 並列計算

ポストプロセスプログラム「jx」を以下の様に実行して下さい。

```
% ./jx fe2.scfout jx.config > jx.log
```

第一引数は OpenMX で生成された scfout ファイルです。第二引数は先に説明された設定ファイルです。周期系の計算に対しては MPI 並列化が実装されており、以下の様に実行可能です。

```
% mpirun -np 2 ./jx fe2.scfout jx.config > jx.log
```

一方で「cluster」計算に対しては「jx」は並列化されていません。並列化は「band」計算のみに実装されていますので、注意してください。またkベクトルに対する並列化のみが考慮されていることや、並列数がk点の総数より大きくなると計算資源の無駄となることにも注意しなければなりません。

## 42.6 実例

クラスター計算の実例としてディレクトリ「work」中に保存された入力ファイル「Fe\_Cluster\_jx.dat」を用いて以下のようにOpenMXの通常計算を実行します。

```
% mpirun -np 2 ./openmx Fe_Cluster_jx.dat
```

入力ファイル「Fe\_Cluster\_jx.dat」は鉄二量体のSCF計算のためのものです。この計算が通常通りに終了すると、scfoutファイル「Fe\_Cluster\_jx.scfout」が生成されます。次に、「jx」による計算を以下のように実行します。

```
% ./jx Fe_Cluster_jx.scfout jx_cluster.config
```

ここで「jx\_cluster.config」はディレクトリ「work」中に保存されています。すると、標準出力に以下のメッセージが表示されます。

```
*****
*****
jx: code for calculating an effective exchange coupling constant J
Copyright (C), 2003, Myung Joon Han, Jaejun Yu, and Taisuke Ozaki
      2019, Asako Terasawa and Taisuke Ozaki
This is free software, and you are welcome to
redistribute it under the constitution of the GNU-GPL.
*****
*****

Read the scfout file (Fe_Cluster_jx.scfout)
***
The file format of the SCFOUT file: 3
And it supports the following functions:
- jx
- polB
- kSpin
- Z2FH
- calB
***

Previous eigenvalue solver = Cluster
atomnum                    = 2
ChemP                      = -0.089740215968 (Hartree)
```

E\_Temp = 300.000000000000 (K)

Evaluation of J based on cluster calculation

| i | j | J [meV]           | J [mRy]          |
|---|---|-------------------|------------------|
| 1 | 1 | 1591.520791120630 | 116.974621661729 |
| 1 | 2 | 106.511867492210  | 7.828477938772   |
| 2 | 2 | 1591.520746009061 | 116.974618346089 |

Elapsed time = 0.036225 (s)

1番と2番の原子の間の交換結合定数  $J_{12}$  は 106.5 meV と計算されます。

バルク計算の実例として、ディレクトリ「work」中に保存された入力ファイル「Fe\_Bulk\_jx.dat」を用いて以下のように OpenMX の通常計算を実行します。

```
% mpirun -np 28 ./openmx Fe_Bulk_jx.dat
```

入力ファイル「Fe\_Bulk\_jx.dat」は鉄 (BCC 構造) の SCF 計算のためのものです。この計算が通常通りに終了すると、scfout ファイル「Fe\_Bulk\_jx.scfout」が生成されます。次に、「jx」による計算を以下のように実行します。

```
% mpirun -np 112 ./jx Fe_Bulk_jx.scfout Fe_Bulk_jx.config | tee jx.log
```

ここで「Fe\_Bulk\_jx.config」はディレクトリ「work」中に保存されています。すると、画面に以下のメッセージが表示されます。

```
*****
*****
jx: code for calculating an effective exchange coupling constant J
Copyright (C), 2003, Myung Joon Han, Jaejun Yu, and Taisuke Ozaki
      2019, Asako Terasawa and Taisuke Ozaki
This is free software, and you are welcome to
redistribute it under the constitution of the GNU-GPL.
*****
*****

Read the scfout file (Fe_Bulk_jx.scfout)
***
The file format of the SCFOUT file: 3
And it supports the following functions:
- jx
- polB
- kSpin
- Z2FH
- calB
```

\*\*\*

Previous eigenvalue solver = Band  
atomnum = 2  
ChemP = -0.205912787451 (Hartree)  
E\_Temp = 300.000000000000 (K)

Jij calculation for a periodic structure

Number of k-grids: 27 27 27

flag\_periodic\_sum = 0: coupling between site i at cell 0 and site j at cell R

Number of poles of Fermi-Dirac continued fraction (PRB.75.035123): 60

| i   | j | c1 | c2 | c3 | J [meV]         | J [mRy]         | time_eig [s] | ... |
|-----|---|----|----|----|-----------------|-----------------|--------------|-----|
| 1   | 1 | -2 | -2 | -2 | -0.845809571401 | -0.062165857439 | 0.51534      | ... |
| 1   | 1 | -2 | -2 | -1 | 0.274300677331  | 0.020160728111  | 0.00000      | ... |
| 1   | 1 | -2 | -2 | 0  | 0.036006012552  | 0.002646393135  | 0.00000      | ... |
| 1   | 1 | -2 | -2 | 1  | 0.274300705154  | 0.020160730156  | 0.00000      | ... |
| 1   | 1 | -2 | -2 | 2  | -0.845809596417 | -0.062165859278 | 0.00000      | ... |
| 1   | 1 | -2 | -1 | -2 | 0.274300737539  | 0.020160732536  | 0.00000      | ... |
| 1   | 1 | -2 | -1 | -1 | -0.206315672897 | -0.015163922403 | 0.00000      | ... |
| 1   | 1 | -2 | -1 | 0  | 0.149714301525  | 0.011003798302  | 0.00000      | ... |
| 1   | 1 | -2 | -1 | 1  | -0.206315540488 | -0.015163912672 | 0.00000      | ... |
| 1   | 1 | -2 | -1 | 2  | 0.274300804604  | 0.020160737465  | 0.00000      | ... |
| ... |   |    |    |    |                 |                 |              |     |
| ..  |   |    |    |    |                 |                 |              |     |
| 2   | 2 | 0  | -1 | 2  | 0.149714016159  | 0.011003777328  | 0.00000      | ... |
| 2   | 2 | 0  | 0  | -2 | 0.401809366424  | 0.029532443987  | 0.00000      | ... |
| 2   | 2 | 0  | 0  | -1 | 11.452192349598 | 0.841720620155  | 0.00000      | ... |

Elapsed time = 340.817975 (s)

図 38 (a) は計算で得られた Fe (BCC 構造) の交換結合定数  $J$  を距離の関数としてプロットしたものです。同様に (b) は HCP 構造の Co、(c) は FCC 構造の Ni、そして (d) は Fe 二量体の計算結果です。ここでの計算では、基底関数セットとして A6.0H-s2p2d2 (ここで A=Fe, Ni, Co)、交換相関汎関数として GGA-PBE が用いられました。またスピン配置は強磁性が仮定されています。計算に用いられた入力ファイルは「Fe\_Bulk\_jx.(dat,config)」、 「Co\_Bulk\_jx.(dat,config)」、 「Ni\_Bulk\_jx.(dat,config)」、 及び「Fe\_Cluster\_jx.(dat,config)」です。これらは入力ファイルはディレクトリ「work」中に保存されていますので、同一の計算を実行することが可能です。計算の詳細は文献 [19] を参照して下さい。

得られた交換結合定数を用いて平均場近似の枠組みで周期系のキュリー温度を計算することも可能です。一般の周期系のキュリー温度  $T_C$  は以下の固有値方程式の最大固有値として得られます。

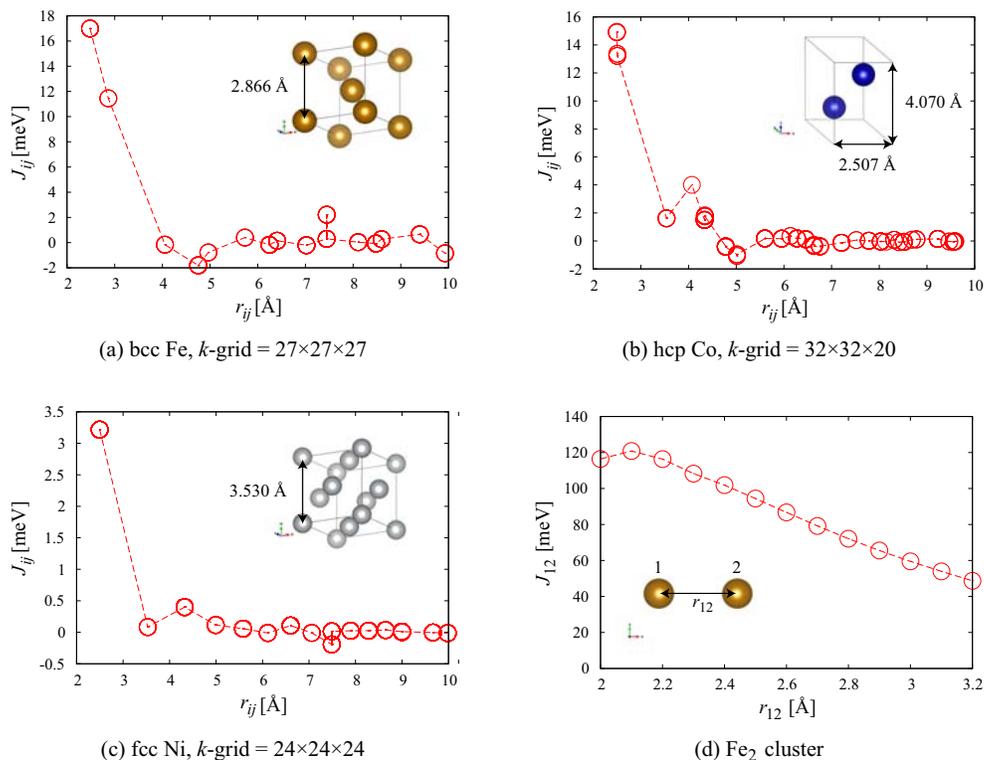


Figure 38: 距離の関数として計算された交換結合定数  $J$ 。(a) bcc Fe, (b) hcp Co, (c) fcc Ni, and (d) Fe dimer。この計算の入力ファイルは「Fe\_Bulk\_jx.(dat,config)」、「Co\_Bulk\_jx.(dat,config)」、「Ni\_Bulk\_jx.(dat,config)」、「Fe\_Cluster\_jx.(dat,config)」であり、これらはディレクトリ「work」中に保存されています。

$$T \langle \vec{s}_i \rangle_z = \frac{2}{3k_B} \sum_j \tilde{J}_{ij} \langle \vec{s}_j \rangle_z \quad (9)$$

$$\tilde{J}_{ij} \equiv J_{ij} - J_{i0,j0} \delta_{ij}. \quad (10)$$

ここで、 $J_{ij}$  と  $J_{i0,j0}$  の定義はそれぞれ式 (8) と (5) にあります。表 8 には BCC 構造の Fe、HCP 構造の Co、そして FCC 構造の Ni に対して式 (9) で計算されたキュリー温度を実験値とともに示します。

| System | $T_C$ [K]  |              |
|--------|------------|--------------|
|        | calculated | experimental |
| bcc Fe | 1321       | 1040         |
| hcp Co | 1640       | 1131         |
| fcc Ni | 445        | 627          |

Table 8: BCC 構造の Fe、HCP 構造の Co、そして FCC 構造の Ni に対して交換結合定数と平均場近似から計算されたキュリー温度とこれらの実験値。

## 43 電気伝導計算

### 43.1 概要

非平衡グリーン関数法 (non-equilibrium Green function method、NEGF 法) に基づき、分子、ナノワイヤ、超格子構造などの電子に由来する電気伝導特性を計算することが可能です。電気伝導計算の機能はコリニアとノンコリニア計算のどちらもサポートしています。その特徴と機能を以下に列挙します。

- ゼロおよび有限バイアス電圧下における二つの電極に接続した系の SCF 計算
- ゲートバイアス電圧下での SCF 計算
- DFT+ $U$  法と併用が可能
- スピン依存の透過率と電流
- 電流に垂直な方向に対して  $k$  分解された透過率と電流
- 電流-電圧曲線の計算
- 高精度・高効率な周回積分
- バイアス電圧効果の補間
- ゼロバイアス電圧下における周期系の迅速な透過率計算
- 固有チャネル解析 [141]
- 実空間における電流・スピン流 [139]

各機能の実装の詳細については、文献 [73] を参照して下さい。まずコリニア計算の場合について、各機能の使用法を説明します。その後、ノンコリニア計算に関して、補足説明します。

#### 考察する系

OpenMX Ver. 3.9 の実装では、図 39 (a) に示す系を NEGF 法により取り扱います。この系は左右の半無限電極部分とそれに接続する中心部からなり、また  $bc$  面上での二次元周期性が仮定されています。二次元周期性を考慮すると、この系は図 39 (b) に示す Bloch 波数ベクトル  $k$  に依存する次元問題に変換することができます。中心領域  $C_0$  と左右の領域  $L_0$  および  $R_0$  との境界面近くの電子構造の緩和を考慮するために、OpenMX Ver. 3.9 の実装では、領域  $C(\equiv L_0|C_0|R_0)$  のグリーン関数が自己無撞着に決定されます。現実装においては、ユニットセルの  $a$  軸が電子輸送の方向であると仮定されていることに留意して下さい。計算モデルの幾何構造を作成する際には、この仕様を守らなければなりません。これに関連して、「ステップ 1: リード線部の計算」の節も参照して下さい。

#### 計算の流れ

NEGF 計算は次の三段階で実行します。

ステップ 1 → ステップ 2 → ステップ 3

各ステップを以下に説明します。

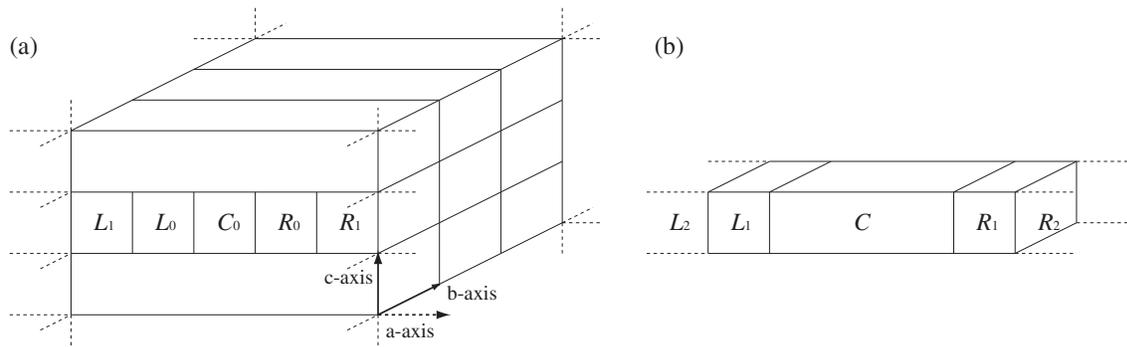


Figure 39: NEGF 法により取り扱う系の構成。a 軸方向の左右に半無限電極が配置され、また bc 面上に二次元周期境界条件を仮定。(b) bc 面内での周期性を考慮することにより、(a) に示す構成から一次元化された系。C 領域は、 $C_0$ 、 $L_0$  および  $R_0$  からなる拡張された中心領域を表す。

- ステップ 1

左右の電極部分のバンド構造計算を、プログラムコード「openmx」を用いて実行。得られた計算結果は、ステップ 2 の NEGF 計算における電極部分に使用。

- ステップ 2

図 39 に示した系に対して、ゼロまたは有限バイアス電圧下での NEGF 計算をプログラムコード「openmx」を用いて実行。この際に、ステップ 1 で計算した結果を電極部分の構築に使用。

- ステップ 3

ステップ 2 で得た結果を利用し、プログラムコード「openmx」を用いて、透過率、電流密度、スピン流密度、固有チャネルを計算。

例：炭素鎖

最初の試みとして炭素鎖を例にとり、上述の三つのステップについて説明します。

ステップ 1

```
./openmx Lead-Chain.dat | tee lead-chain.std
```

ステップ 1 の計算により、ファイル「negf-chain.hks」が生成されます。

ステップ 2

```
./openmx NEGF-Chain.dat | tee negf-chain.std
```

ステップ 2 の計算により、ファイル「negf-chain.tranb」が生成されます。

ステップ 3

openmx はステップ 2 の計算が完了すると自動的にステップ 3 の計算に移ります。ステップ 2 とステップ 3 の計算を分離して行いたい場合には、以下のように openmx を実行して下さい。

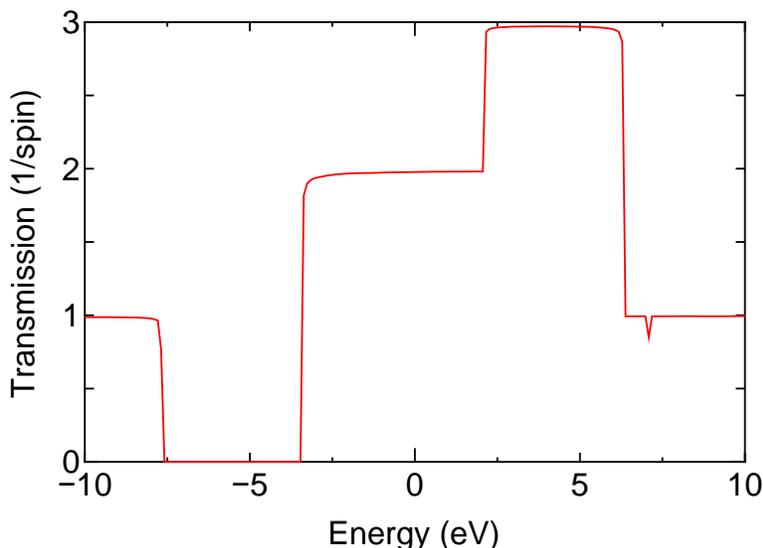


Figure 40: 炭素鎖の透過率のエネルギー依存性。エネルギーの原点は左電極の化学ポテンシャルに設定。

```
./openmx NEGF-Chain.dat | tee negf-chain.std
```

ステップ 3 では、openmx は「negf-chain.tranb」を読み込み、透過率、電流、固有チャネルの計算を行います。計算終了後に以下のファイルがステップ 3 で生成されます。

```
negf-chain.conductance      negf-chain.tranec0_0_0_2_r.cube
negf-chain.current         negf-chain.tranec0_0_0_3_i.cube
negf-chain.tran0_0         negf-chain.tranec0_0_0_3_r.cube
negf-chain.tranec0_0_0_0_i.cube  negf-chain.tranec0_0_0_4_i.cube
negf-chain.tranec0_0_0_0_r.cube  negf-chain.tranec0_0_0_4_r.cube
negf-chain.tranec0_0_0_1_i.cube  negf-chain.traneval0_0_0
negf-chain.tranec0_0_0_1_r.cube  negf-chain.tranevec0_0_0
negf-chain.tranec0_0_0_2_i.cube
```

ディレクトリ「work/negf\_example」中の入力ファイルを用いると、この計算を再現することができます。ステップ 3 の計算で得られた「negf-chain.tran0\_0」の六列目を四列目の関数としてプロットすると、図 40 に示す透過率曲線が得られます。

## 43.2 ステップ 1：電極部分の計算

ステップ 1 の計算は、通常のバンド構造計算と同じものです。ただし入力ファイルに二つのキーワード「NEGF.output\_hks」と「NEGF.filename.hks」を付け加えます。

```
NEGF.output_hks    on
NEGF.filename.hks  lead-chain.hks
```

上記のキーワードを付け加えることで、電極部分を構築するための情報がファイルに保存されます。ハミルトニアン行列要素、電子分布、および差電子 Hartree ポテンシャルが、キーワード「NEGF.filename.hks」

で指定されたファイルに保存されます。上記の場合には、ファイル「lead-chain.hks」が生成されます。「*NEGF.filename.hks*」で指定されたファイルは、ステップ 2 の計算で使用されることになります。現実装では電子輸送の方向は a 軸方向に仮定されていますので、バンド構造計算においては、この仕様を念頭において電極モデルを構築する必要があります。しかし、ユーザー自身で構造を回転させる必要はありません。格子ベクトルの指定を変更するだけで、簡単に電子輸送の方向を適切に設定することが可能です。例えば、次の格子ベクトルにおいて、

```
<Atoms.UnitVectors
  3.0  0.0  0.0
  0.0  3.0  0.0
  0.0  0.0 10.0
Atoms.UnitVectors>
```

a 軸としてベクトル (0.0, 0.0, 10.0) を指定したい場合には以下のようにベクトルの順番を入れ替えるだけで、電子輸送の方向を適切に設定できます。

```
<Atoms.UnitVectors
  0.0  0.0 10.0
  3.0  0.0  0.0
  0.0  3.0  0.0
Atoms.UnitVectors>
```

この様に指定すれば、(0.0, 0.0, 10.0) の方向が電子輸送の方向になります。上の例で示したように、格子ベクトルの順序を変更する際には、キーワード「scf.Kgrid」も同様に変更しなければならないことに注意して下さい。

ステップ 2 の計算において、表面グリーン関数を用いることにより、電極部分の半無限性を考慮しています。そのため、ステップ 2 の計算では a 軸方向に関しては波数空間の離散化を導入することなしに、半無限性を取り扱うことができるようになります。a 軸方向の半無限性の取扱いに関し、ステップ 1 とステップ 2 での計算の整合性を保つために、ステップ 1 のバンド構造計算では a 軸方向に多数の k 点を割り当てるようにして下さい。また、bc 面上の k 点数は、ステップ 1 とステップ 2 の計算において同一の値を使用して下さい。

### 43.3 ステップ 2 : NEGF 計算

#### A. 左電極 | デバイス | 右電極の設定

図 39 に示した領域  $L_0$ 、 $C_0$ 、 $R_0$  は次のようにして設定できます。

中心領域  $C_0$  の幾何学的構造は、キーワード「Atoms.Number」および「Atoms.SpeciesAndCoordinates」により、以下のように指定します。

```
Atoms.Number      18
<Atoms.SpeciesAndCoordinates
  1  C  3.000  0.000  0.000  2.0 2.0
  .....
 18  C 28.500  0.000  0.000  2.0 2.0
Atoms.SpeciesAndCoordinates>
```

左側の電極部分  $L_0$  の幾何学的構造は、キーワード「LeftLeadAtoms.Number」および「LeftLeadAtoms.SpeciesAndCoordinates」により、以下のように指定します。

```
LeftLeadAtoms.Number      3
<LeftLeadAtoms.SpeciesAndCoordinates
  1  C -1.500  0.000  0.000  2.0 2.0
  2  C  0.000  0.000  0.000  2.0 2.0
  3  C  1.500  0.000  0.000  2.0 2.0
LeftLeadAtoms.SpeciesAndCoordinates>
```

右側の電極部分  $R_0$  の幾何学的構造は、キーワード「RightLeadAtoms.Number」および「RightLeadAtoms.SpeciesAndCoordinates」により、以下のように指定します。

```
RightLeadAtoms.Number     3
<RightLeadAtoms.SpeciesAndCoordinates
  1  C 30.000  0.000  0.000  2.0 2.0
  2  C 31.500  0.000  0.000  2.0 2.0
  3  C 33.000  0.000  0.000  2.0 2.0
RightLeadAtoms.SpeciesAndCoordinates>
```

ここで示した例は、上述の節で取り上げた炭素鎖の場合のものです。中心領域  $C_0$  は 18 個の炭素原子から構成され、左側の領域  $L_0$  と右側の領域  $R_0$  は、それぞれ 3 個の炭素原子から成り、すべての隣接原子間距離は 1.5 Å です。デバイス領域  $C_0$  と電極  $L_0$  および  $R_0$  の幾何学的構造を設定することで、OpenMX は図 39 に示した拡張中心領域  $C(\equiv L_0|C_0|R_0)$  を構成します。中心領域  $C_0$  とその外側の  $L_0(R_0)$  の境界付近での電子構造の緩和を考慮するために、拡張中心領域  $C$  のグリーン関数が自己無撞着に決定されず、NEGF 法を用いて、拡張中心領域  $C$  のグリーン関数を計算するために、次の二つの条件を満たすように計算モデルを構築する必要があります [73]。

1. 領域  $C_0$  における局在基底軌道  $\phi$  は、領域  $L_0$  および  $R_0$  の局在基底軌道と重なるが、領域  $L_1$  および  $R_1$  と重なることはない。
2.  $L_i (R_i)$  領域における局在基底軌道  $\phi$  は、その最近接セル  $L_{i-1} (R_{i-1})$  および  $L_{i+1} (R_{i+1})$  より遠方のセル内の基底軌道と重なることはない。

OpenMX の基底関数は実空間において厳密に局在しているため [41, 42]、特定のカットオフ半径を持つ局在軌道を各領域に対して割り当てると、 $L_i$  および  $R_i$  のユニットセルの大きさを調整することにより、上記の二つの条件を常に満たすことができます。領域  $L_0$ 、 $C_0$ 、 $R_0$  の単位胞を指定する必要はありませんが、周期性を暗黙のうちに仮定していることに留意して下さい。半無限電極の構築は、ステップ 1 のバンド構造計算で用いた単位胞を利用して自動的に行われます。そのための情報はファイル「*NEGF.filename.hks*」に保存されています。また、図 39 に示した配置構造のため、左側と右側の電極の bc 面内の単位胞ベクトルは一致していなければなりません。また拡張中心領域  $C$  に対する bc 面内の単位胞ベクトルは、電極部分の単位胞ベクトルと同じであることが暗黙のうちに仮定されています。ユーザーは、こうした制約の枠内で、幾何構造を設定できます。

原子位置の指定で用いられる単位は、以下の様に指定します。

Atoms.SpeciesAndCoordinates.Unit Ang # Ang|AU

ただし、NEGF 計算では「Ang」または「AU」のみに対応しています。「FRAC」での指定はできませんので、注意して下さい。

どの様に OpenMX が指定した幾何学的な配置構造を解析したのか確認するために、標準出力にその情報が記載されています。上述の炭素鎖の例では以下の情報が出力されています。

```
<TRAN_Calc_GridBound>
```

```
*****
```

```
The extended cell consists of Left0-Center-Right0.
```

```
The cells of left and right reads are connected as.
```

```
...|Left2|Left1|Left0-Center-Right0|Right1|Right2...
```

```
Each atom in the extended cell is assigned as follows:
```

```
where '12' and '2' mean that they are in 'Left0', and
```

```
'12' has overlap with atoms in the Left1,
```

```
and '13' and '3' mean that they are in 'Right0', and
```

```
'13' has overlap with atoms in the 'Right1', and also
```

```
'1' means atom in the 'Center'.
```

```
*****
```

```
Atom1 = 12 Atom2 = 12 Atom3 = 12 Atom4 = 1 Atom5 = 1 Atom6 = 1 Atom7 = 1  
Atom8 = 1 Atom9 = 1 Atom10 = 1 Atom11 = 1 Atom12 = 1 Atom13 = 1 Atom14 = 1  
Atom15 = 1 Atom16 = 1 Atom17 = 1 Atom18 = 1 Atom19 = 1 Atom20 = 1 Atom21 = 1  
Atom22 = 13 Atom23 = 13 Atom24 = 13
```

$L_0|C_0|R_0$  からなる拡張された中心領域の原子には数値が割り当てられています。「12」および「2」は原子が  $L_0$  に属し、「12」は  $L_1$  内の原子と基底関数間の重なりを持つことを意味します。「13」および「3」は原子が  $R_0$  に属し、「13」は  $R_1$  にある原子と基底関数間の重なりを持つことを意味します。また「1」は原子が  $C_0$  に属していることを意味します。この出力を調べることで、計算する系の幾何学的な構造が適切に構築されているか否かを確認することができます。

## B. キーワード

ステップ 2 の NEGF 計算を実行するために、キーワード「scf.EigenvalueSolver」を次のように指定します。

```
scf.EigenvalueSolver      NEGF
```

NEGF 計算に関連するキーワードを以下に列挙します。

```
NEGF.filename.hks.l      lead-chain.hks  
NEGF.filename.hks.r      lead-chain.hks
```

```

NEGF.Num.Poles          100      # defalut=150
NEGF.scf.Kgrid          1 1      # defalut=1 1

NEGF.bias.voltage       0.0      # default=0.0 (eV)
NEGF.bias.neq.im.energy 0.01    # default=0.01 (eV)
NEGF.bias.neq.energy.step 0.02  # default=0.02 (eV)

```

各キーワードの内容を以下に説明します。

```

NEGF.filename.hks.l     lead-chain.hks
NEGF.filename.hks.r     lead-chain.hks

```

電極部分の情報を含むファイルは、上記の二つのキーワードにより指定できます。「NEGF.filename.hks.l」は左側、「NEGF.filename.hks.r」は右側の電極部分に対応します。

```

NEGF.Num.Poles          100      # defalut=150

```

平衡密度行列は、周回積分法により評価します [73, 74]。この方法で用いる極の数は、キーワード「NEGF.Num.Poles」により指定します。

```

NEGF.scf.Kgrid          1 1      # defalut=1 1

```

逆格子ベクトル  $\vec{b}$  および  $\vec{c}$  を離散化するための k 点数を、キーワード「NEGF.scf.Kgrid」により指定します。a 軸方向には周期性を持っていないため、a 軸に対しては指定する必要がありません。

```

NEGF.scf.Iter.Band      6        # defalut=6

```

SCF 計算の最初の数ステップでは、b 軸および c 軸と同様に a 軸方向にも周期性を仮定することにより、通常対角化法を用いた方が最終的な収束が加速されることが分かっています。自己無撞着な NEGF 法においてしばしば問題となる SCF 計算での収束困難性は、この方法によってかなり軽減します。通常対角化法を使用する SCF ステップ数を、キーワード「NEGF.scf.Iter.Band」により指定します。「NEGF.scf.Iter.Band」で指定した SCF ステップ数までは通常対角化法を用い、それ以降のステップにおいては、NEGF 法が適用されることとなります。デフォルト値は 6 です。

```

NEGF.bias.voltage       0.0      # default=0.0 (eV)

```

キーワード「NEGF.bias.voltage」により、左右の電極間に印加するソース・ドレインバイアス電圧を eV の単位で指定します。この eV の単位は電圧に換算するとボルト (V) に対応しています。ソースとドレイン間の電位差のみが物理的に意味を持つため、電位の差であるソース・ドレインバイアス電圧を本キーワードによって与えることとなります。

```

NEGF.bias.neq.im.energy 0.01    # default=0.01 (eV)
NEGF.bias.neq.energy.step 0.02  # default=0.02 (eV)

```

有限のソース・ドレインバイアス電圧を印加すると、密度行列の一部は非平衡グリーン関数から計算されません。非平衡グリーン関数は複素平面上で一般に解析的ではありませんので、平衡グリーン関数で用いられた周回積分法は使用できません。現在の実装では、非平衡グリーン関数は微小虚部を付与した実軸上で単純な矩形求積法を用いて評価しています。この際、虚数部はキーワード「NEGF.bias.neq.im.energy」により指定します。またステップ幅はキーワード「NEGF.bias.neq.energy.step」により、eVの単位で与えます。通常はデフォルト値で十分な精度が確保できますが、収束性の詳細な議論は文献 [73] を参照して下さい。非平衡グリーン関数を実軸上のエネルギー点数は標準出力およびファイル「*System.Name.out*」で確かめることができます。「NEGF-Chain.dat」の例では、バイアス電圧が 0.5V の際には 120 個のエネルギー点が割り当てられており、以下のように標準出力から確認できます。

```
Intrinsic chemical potential (eV) of the leads
Left lead:  -7.752843837400
Right lead: -7.752843837400
add voltage = 0.0000 (eV) to the left lead: new ChemP (eV):  -7.7528
add voltage = 0.5000 (eV) to the right lead: new ChemP (eV):  -7.2528

Parameters for the integration of the non-equilibrium part
lower bound:          -8.706843837400 (eV)
upper bound:          -6.298843837400 (eV)
energy step:          0.020000000000 (eV)
number of steps:      120
```

グリーン関数を実軸上のエネルギー点の総数は、非平衡グリーン関数を実軸上のエネルギー点と、平衡グリーン関数を実軸上の極の数の総和で与えられます。計算時間は、エネルギー点の総数に比例することに留意して下さい。

```
NEGF.Poisson.Solver      FD      # FD|FFT, default=FD
```

NEGF 法では差電子密度に対する静電ポテンシャルを 2 次元 FFT+1 次元有限差分法 (FD) [73] もしくは 3 次元 FFT 法 (FFT) [75] のどちらかの方法で評価します。このポアソン・ソルバーの選択はキーワード「NEGF.Poisson.Solver」で行います。非極性系についてはどちらの方法も同様の静電ポテンシャルを与えますが、極性系については両者の差異は大きくなります。前者の FD では、電極部分と中心領域の境界条件が厳密に満されるため、理論的にはより正しい方法です。ここでの境界条件とは、電極部分と中心領域の界面での静電ポテンシャルがステップ 1 の計算におけるポテンシャル値と同一であるというものです。SCF 収束性の観点からは、FD の収束性が悪く、一方、後者の FFT の方がより SCF 収束が容易であるという傾向が分かっています。デフォルトは FD です。

### C. SCF の収束条件

NEGF 法では、キーワード「scf.criterion」により与えられる SCF の収束条件は残差ノルム「NormRD」に適用されます。NEGF 法以外の場合には、SCF の収束条件は dUele に適用されます。

### D. ゲート・バイアス電圧

OpenMX Ver. 3.9 の実装では、次式で定義されるポテンシャルを加えることで、ゲート電圧  $V_g(x)$  が取り扱われます。

$$V_g(x) = V_g^{(0)} \exp \left[ - \left( \frac{x - x_c}{d} \right)^8 \right],$$

ここで、 $V_g^{(0)}$  はゲート電圧に対応する定数であり、キーワード「NEGF.gate.voltage」により指定できます。

```
NEGF.gate.voltage  1.0    # default=0.0 (in eV)
```

また  $x_c$  は領域  $C_0$  の中心位置、 $d$  は領域  $C_0$  の  $a$  軸方向のセルベクトルの長さです。式の形が示すように、印加されたゲート電圧は、主として拡張中心領域  $C$  内の領域  $C_0$  に作用します。このポテンシャルは、鏡像電荷によって生成されるポテンシャル形状に類似しています [76]。

## E. 状態密度 (DOS)

NEGF 計算において、状態密度 (DOS) は次のキーワードを設定することにより計算できます。

```
Dos.fileout          on          # on|off, default=off
NEGF.Dos.energyrange -15.0 25.0 5.0e-3 #default=-10.0 10.0 5.0e-3 (eV)
NEGF.Dos.energy.div  200         # default=200
NEGF.Dos.Kgrid       1 1         # default=1 1
```

NEGF 法において DOS を計算する際には、「Dos.fileout」を「on」にして下さい。また、DOS を計算するエネルギー範囲は、キーワード「NEGF.Dos.energyrange」により与えます。ここで、最初と二番目の数値はエネルギーの下限と上限で、三番目の数値は DOS を滑らかにするために用いる虚数値です。「NEGF.Dos.energyrange」により指定されたエネルギー範囲は、キーワード「NEGF.Dos.energy.div」で指定した数値で分割され、そのエネルギー点上で DOS が計算されます。逆格子ベクトル  $\vec{b}$  および  $\vec{c}$  を離散化するための  $k$  点のそれぞれの数を、キーワード「NEGF.Dos.Kgrid」により指定します。「NEGF.Dos.Kgrid」で与える数を「NEGF.scf.Kgrid」より大きな値にすることで計算精度が向上します。NEGF 計算が完了すると、「System.Name.Dos.val」と「System.Name.Dos.vec」の二つのファイルを生成します。その後の解析手順は、一般の場合と同様です。また、エネルギーの原点は左側の電極の化学ポテンシャルに設定されていることに留意して下さい。

## 43.4 ステップ 3: 透過率、電流 (密度) と固有チャネル

ステップ 2 とステップ 3 の計算を終了後、透過率、電流 (密度)、固有チャネルを計算できます。

### 43.4.1 透過率、電流、コンダクタンス

最初に、openmx を用いて透過率、電流、コンダクタンスを計算します。この計算に関連するキーワードは以下のとおりです。

```
NEGF.tran.Analysis   on          # default on
NEGF.tran.CurrentDensity on      # default on
NEGF.tran.energyrange -10 10 1.0e-3 # default=-10.0 10.0 1.0e-3 (eV)
```

```

NEGF.tran.energydiv      200      # default=200
NEGF.tran.Kgrid          1 1      # default= 1 1

```

- NEGF.tran.Analysis, NEGF.tran.Channel, NEGF.tran.CurrentDensity

NEGF.tran.Analysis を on にした場合、透過率、電流、コンダクタンスが計算されます。

- NEGF.tran.energyrange, NEGF.tran.energydiv

キーワード「NEGF.tran.energyrange」により透過率を計算するエネルギー範囲を指定します。最初と二番目の数値は、エネルギーの下限値と上限値で、三番目の数値は透過率を滑らかにするための虚数値です。「NEGF.tran.energyrange」により指定されるエネルギー範囲は、キーワード「NEGF.tran.energydiv」により指定した数値で分割され、そのエネルギー点上で透過率が計算されます。

- NEGF.tran.Kgrid

逆格子ベクトル  $\tilde{b}$  および  $\tilde{c}$  を離散化するための k 点の数のそれぞれの値を、「NEGF.tran.Kgrid」により指定します。「NEGF.tran.Kgrid」で与える値は「NEGF.scf.Kgrid」で与える値とは異なるものに設定することが可能であり、計算精度を向上には大きな値の使用が推奨されます。

透過率、電流、コンダクタンスの計算では、以下の様なメッセージが標準出力に表示されます。

```

*****
*****
Welcome to TRAN_Main_Analysis.
This is a post-processing code of OpenMX to analyze
transport properties such as electronic transmission,
current, eigen channel, and current distribution in
real space based on NEGF.
Copyright (C), 2002-2015, H. Kino and T. Ozaki
TRAN_Main_Analysis comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to
redistribute it under the constitution of the GNU-GPL.
*****
*****

```

Chemical potentials used in the SCF calculation

Left lead: -5.125617225230 (eV)

Right lead: -5.125617225230 (eV)

NEGF.current.energy.step 1.0000e-02 seems to be large for the calculation of current ...

The recommended Tran.current.energy.step is 0.0000e+00 (eV).

TRAN\_Channel\_kpoint 0 0.000000 0.000000

TRAN\_Channel\_energy 0 0.000000 eV

TRAN\_Channel\_Num 5

Parameters for the calculation of the current

lower bound: -5.125617225230 (eV)  
upper bound: -5.125617225230 (eV)  
energy step: 0.010000000000 (eV)  
imaginary energy 0.001000000000 (eV)  
number of steps: 0

calculating...

myid0= 0 i2= 0 i3= 0 k2= 0.0000 k3= -0.0000  
myid0= 1 i2= 0 i3= 0 k2= 0.0000 k3= -0.0000

Transmission: files

./negf-chain.tran0\_0

Current: file

./negf-chain.current

Conductance: file

./negf-chain.conductance

計算が終わると、以下の3つのファイルが生成されます。

negf-chain.tran0\_0, negf-chain.current, negf-chain.conductance:

- *System.Name.tran#\_%*

このファイルにはアップとダウンのスピンの状態に対する透過率が保存されます。4番目の列は、左側リード線部の化学ポテンシャルに対する相対的なエネルギーで、6番目と8番目の列は、それぞれアップとダウンスピンの状態に対する透過率です。「NEGF.tran.Kgrid」により与えられるk点の数を多く取ると、ファイル拡張子に「#」と「%」の異なる組を持つファイルが、k点毎に生成します。ファイルの数字とk点との対応はファイル内で確認できます。

- *System.Name.current*

このファイルには、アップスピン状態とダウンスピン状態に対するk分解された電流およびその平均値がアンペアの単位で保存されます。

- *System.Name.conductance*

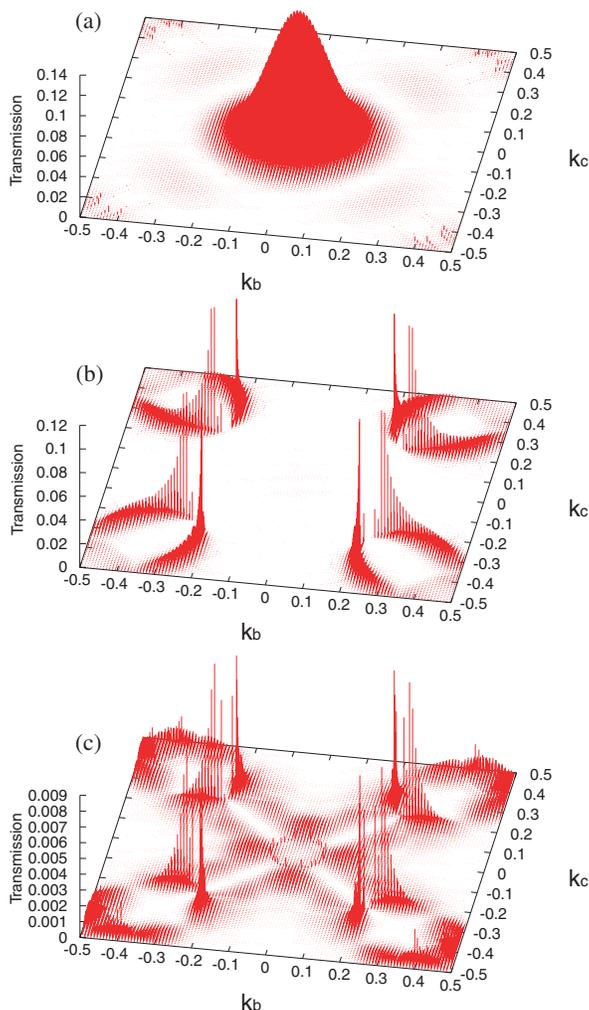


Figure 41: Fe|MgO|Feの化学ポテンシャルでのk分解透過率。(a) 平行スピ配置での多数スピ状態、(b) 平行スピ配置での少数スピ状態、(c) 反平行スピ配置でのスピ状態。各計算には、 $120 \times 120$ 個のk点を使用。

このファイルには、アップスピ状態とダウンスピ状態に対するk分解されたコンダクタンスとその平均値が量子化コンダクタンスの単位 ( $G_0 \equiv \frac{e^2}{h}$ ) で保存されています。コンダクタンス  $G$  は、左電極の化学ポテンシャル  $\mu_L$  での透過率  $T$  に、次のように比例します。

$$G = \frac{e^2}{h} T(\mu_L)$$

一例として、ファイル「*System.Name.conductance*」を用いて作成された Fe|MgO|Fe 構造のk分解透過率を図 41 に示します。

#### 43.4.2 実空間における電流/スピ流密度

ステップ 3 で、実空間における電流/スピ流密度が計算可能です。この計算に関連するキーワードは以下のとおりです。

```
NEGF.tran.CurrentDensity on # default on
```

- `NEGF.tran.CurrentDensity`

`NEGF.tran.CurrentDensity` が on の場合、実空間での電流密度が計算されます。

電流密度の計算の際には、`openmx` の計算から以下の標準出力が得られます。

Start Calculation of the currentdensity

Spin #0

Sum of current in real space [a.u.]

Left(ideal) : -9.10585e-06

Right(ideal): -9.10583e-06

Left(truncated ): -8.66971e-06

Right(truncated): -8.69926e-06

Spin #1

Sum of current in real space [a.u.]

Left(ideal) : -4.54540e-08

Right(ideal): -4.54544e-08

Left(truncated ): -4.19469e-08

Right(truncated): -4.27460e-08

Output: Currentdensity

Charge-current density along a-axis: `./negf-8zgnr-0.3.curden1.cube`

Spin-current density along a-axis: `./negf-8zgnr-0.3.scurden1.cube`

Charge-current density: `./negf-8zgnr-0.3.curden.xsf`

Spin-current density: `./negf-8zgnr-0.3.scurden.xsf`

Voronoi Charge-current density: `./negf-8zgnr-0.3.curden_atom.xsf`

Voronoi Spin-current density: `./negf-8zgnr-0.3.scurden_atom.xsf`

この計算例の場合には、以下の6つのファイルが作成されます。

`negf-8zgnr-0.3.curden.xsf`, `negf-8zgnr-0.3.scurden.xsf`,

`negf-8zgnr-0.3.curden1.cube`, `negf-8zgnr-0.3.scurden1.cube`,

`negf-8zgnr-0.3.curden_atom.xsf`, `negf-8zgnr-0.3.scurden_atom.xsf`,

これらのファイルには、以下の物理量が記載されています。

- `System.Name.curden.xsf`, `System.Name.scurden.xsf`

これらのファイルには実空間格子上的電流及びスピン流の密度が記載されます。XCrySDenの「Display→Forces」によって可視化可能です。

- `System.Name.curden1.cube`, `System.Name.scurden1.cube`

これらのファイルには、実空間格子上的電流及びスピン流の密度のa成分が記載されます。VESTAやXCrySDenにて可視化可能です。

- `System.Name.curden_atom.xsf`, `System.Name.scurden_atom.xsf`

これらのファイルには各原子の Voronoi 多面体にて平均化した電流及びスピンの密度が記載されます。XCrySDen の「Display→Forces」によって可視化可能です。

OpenMX Ver. 3.9 では、

```
NEGF.OffDiagonalCurrent on # default off
```

を設定することでノンコリニア磁性の計算も可能となっており、以下のファイルが作成されます。

- *System.Name.odcurden\_r.xsf*, *System.Name.odcurden\_i.xsf*
- *System.Name.odcurden1\_r.cube*, *System.Name.odcurden1\_i.cube*
- *System.Name.odcurden\_atom\_r.xsf*, *System.Name.odcurden\_atom\_i.xsf*

これらのファイルには電流密度の非対角スピン成分が記載されます。これらは一般的に複素数となるため、実部と虚部に分けて出力されます。

例として、図 42 に 0.3V のバイアス電圧下で反強磁性接合を持つグラフェンナノリボン (8-zigzag) の電流密度を示します。境界の近傍では、基底関数が適切に扱われていないため非物理的な振る舞いをしています。より正確な計算を実現するためには、より大きなスーパーセルモデルを用意する必要があります。

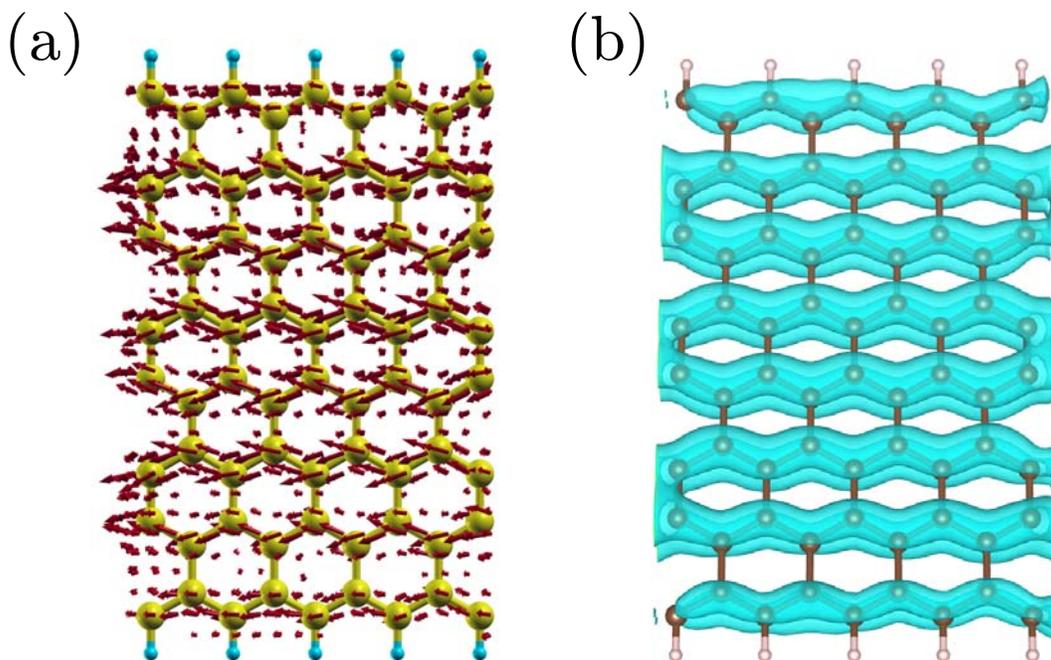


Figure 42: (a) 0.3V のバイアス電圧下で反強磁性接合を持つグラフェンナノリボン (8-zigzag) の電流密度。(b) (a) の a 成分の等値面。

#### 43.4.3 固有チャネル解析

ステップ 3 では、透過固有チャネルの計算も可能です。この計算に関連するキーワードは以下のとおりです。

```

NEGF.tran.Channel      on      # default on
NEGF.Channel.Nkpoint   1      # default=1
<NEGF.Channel.kpoint
0.0 0.0
NEGF.Channel.kpoint>
# default 0.0 0.0
NEGF.Channel.Nenergy   1      # default=1
<NEGF.Channel.energy
0.0
NEGF.Channel.energy>
# default 0.0
NEGF.Channel.Num      5      # default=5(for collinear), 10(for Non-collinear)

```

- `NEGF.tran.Channel`  
`NEGF.tran.Channel` が `on` の場合、固有チャンネルが計算されます。
- `NEGF.Channel.Nkpoint`, `<NEGF.Channel.kpoint, NEGF.Channel.kpoint>`  
これらのキーワードは、固有チャンネルを計算する  $k$  点を指定するのに用います。`<NEGF.Channel.kpoint` と `NEGF.Channel.kpoint>` の間に 1 行毎に  $k$  点を記載して下さい。  $k$  点の総数は `NEGF.Channel.Nkpoint` にて指定して下さい。  $k$  座標は 2 次元の規格化座標として記述し、透過方向に垂直な 2 つの逆格子ベクトルの係数として  $k$  を指定して下さい。
- `NEGF.Channel.Nenergy`, `<NEGF.Channel.energy, NEGF.Channel.energy>`  
これらのキーワードは、計算する固有チャンネルのエネルギーの指定に用います。`<NEGF.Channel.energy` と `NEGF.Channel.energy>` の間に 1 行毎にエネルギーを記載して下さい。エネルギーの総数は `NEGF.Channel.Nenergy` にて指定して下さい。エネルギーの単位は eV であり、左電極の化学ポテンシャルを基準に設定して下さい。
- `NEGF.Channel.Num`  
このパラメータでは、実空間表記での固有チャンネルの数を指定します。  $k$ 、エネルギー、スピンのそれぞれについて、透過固有値が降順に出力されます。ファイル形式は Gaussian cube 形式で、実部・虚部に分けて出力されます。

固有チャンネルの計算では、以下の様な標準出力が `openmx` より得られます。

```

*****
Calculation of transmission eigenchannels starts
*****

File index : negf-8zgnr-0.3.traneval#k_#E_#spin negf-8zgnr-0.3.tranevec#k_#E_#spin

myid0 = 0, #k : 0, N_{ort} / N_{nonort} : 380 / 380
PE 0 generates ./negf-8zgnr-0.3.traneval0_0_0 . Sum(eigenval) : 0.031643

```

PE 0 generates ./negf-8zgnr-0.3.traneval0\_0\_1 . Sum(eigenval) : 0.000508

Eigenchannel calculation finished

They are written in plottable files.

File index : negf-8zgnr-0.3.tranec#k\_#E\_#spin\_#branch\_r.cube(.bin)

negf-8zgnr-0.3.tranec#k\_#E\_#spin\_#branch\_i.cube(.bin)

```
./negf-8zgnr-0.3.tranec0_0_0_r.cube      ./negf-8zgnr-0.3.tranec0_0_0_0_i.cube
./negf-8zgnr-0.3.tranec0_0_0_1_r.cube   ./negf-8zgnr-0.3.tranec0_0_0_1_i.cube
./negf-8zgnr-0.3.tranec0_0_0_2_r.cube   ./negf-8zgnr-0.3.tranec0_0_0_2_i.cube
./negf-8zgnr-0.3.tranec0_0_0_3_r.cube   ./negf-8zgnr-0.3.tranec0_0_0_3_i.cube
./negf-8zgnr-0.3.tranec0_0_0_4_r.cube   ./negf-8zgnr-0.3.tranec0_0_0_4_i.cube
./negf-8zgnr-0.3.tranec0_0_1_0_r.cube   ./negf-8zgnr-0.3.tranec0_0_1_0_i.cube
./negf-8zgnr-0.3.tranec0_0_1_1_r.cube   ./negf-8zgnr-0.3.tranec0_0_1_1_i.cube
./negf-8zgnr-0.3.tranec0_0_1_2_r.cube   ./negf-8zgnr-0.3.tranec0_0_1_2_i.cube
./negf-8zgnr-0.3.tranec0_0_1_3_r.cube   ./negf-8zgnr-0.3.tranec0_0_1_3_i.cube
./negf-8zgnr-0.3.tranec0_0_1_4_r.cube   ./negf-8zgnr-0.3.tranec0_0_1_4_i.cube
```

この例では、以下の 22 個のファイルが生成します。

```
negf-8zgnr-0.3.treval0_0_0, negf-8zgnr-0.3.tranevec0_0_0,
negf-8zgnr-0.3.tranec0_0_0_0_r.cube - negf-8zgnr-0.3.tranec0_0_1_4_r.cube,
negf-8zgnr-0.3.tranec0_0_0_0_i.cube - negf-8zgnr-0.3.tranec0_0_1_4_i.cube
```

- *System.Name.traneval*{#k}\_{#E}\_{#s}

このファイルには、{#k}番目の k、{#E}番目のエネルギー及び {#s}番目のスピンのについて、全ての固有チャネルの透過固有値が記載されます。

- *System.Name.tranevec*{#k}\_{#E}\_{#s}

このファイルには、{#k}番目の k、{#E}番目のエネルギー及び {#s}番目のスピンのについて、固有チャネルの LCAO 成分が記載されます。

例えば、negf-chain.tranevec0\_0\_0 は以下のようになります。

```
*****
*****
Eigenvalues and LCAO coefficients
at the k-points specified in the input file.
*****
*****
```

```
# of k-point = 0
k2= 0.00000 k3= 0.00000
```

```
# of Energy = 0
e= 0.00000
```

Spin = Up

Real (Re) and imaginary (Im) parts of LCAO coefficients

|   |       | 1        |          | 2        |          | 3        |          | 4        |          |
|---|-------|----------|----------|----------|----------|----------|----------|----------|----------|
|   |       | Re       | Im       | Re       | Im       | Re       | Im       | Re       | Im       |
| 1 | C 0 s | -0.00000 | -0.00000 | -0.00000 | 0.00000  | 0.00000  | 0.00000  | -0.00000 | 0.00000  |
|   | 1 s   | -0.00000 | -0.00000 | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  | -0.00000 |
|   | 0 px  | -0.63002 | -1.49377 | -0.14466 | 0.00019  | 0.01644  | -0.00032 | -0.07885 | 0.00095  |
|   | 0 py  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | -0.00000 | 0.00000  | 0.00000  | 0.00000  |
|   | 0 pz  | 0.00000  | -0.00000 | 0.00000  | -0.00000 | -0.00000 | -0.00000 | -0.00000 | -0.00000 |
| 2 | C 0 s | -0.00000 | -0.00000 | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  | -0.00000 |
|   | 1 s   | 0.00000  | 0.00000  | -0.00000 | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  |
|   | 0 px  | 0.18040  | -0.03816 | -0.00452 | 0.00009  | -0.00545 | -0.00010 | -0.01970 | -0.00004 |
|   | 0 py  | -0.00000 | -0.00000 | -0.00000 | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  |
|   | 0 pz  | 0.00000  | -0.00000 | -0.00000 | 0.00000  | -0.00000 | -0.00000 | -0.00000 | -0.00000 |
| 3 | C 0 s | 0.00000  | 0.00000  | 0.00000  | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  |
|   | 1 s   | -0.00000 | -0.00000 | 0.00000  | 0.00000  | 0.00000  | -0.00000 | 0.00000  | -0.00000 |
|   | 0 px  | 2.06634  | 0.40490  | 0.11067  | 0.00023  | -0.06068 | 0.00009  | -0.06690 | -0.00042 |
|   | 0 py  | 0.00000  | 0.00000  | 0.00000  | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  |
|   | 0 pz  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | -0.00000 | -0.00000 | 0.00000  | -0.00000 |
| 4 | C 0 s | 0.00000  | 0.00000  | 0.00000  | 0.00000  | -0.00000 | -0.00000 | -0.00000 | -0.00000 |
|   | 1 s   | -0.00000 | -0.00000 | -0.00000 | -0.00000 | -0.00000 | 0.00000  | -0.00000 | 0.00000  |

- *System.Name.tranec*{#k}\_{#E}\_{#s}\_{#c}\_r.cube,  
*System.Name.tranec*{#k}\_{#E}\_{#s}\_{#c}\_i.cube

このファイルには、Gaussian cube形式で固有チャネルの実部・虚部が記載されます。VESTAやXcrySDenなどを用いて、等値面を描画できます。図43に0.3Vのバイアス電圧下で反強磁性接合を持つグラフェンナノリボン(8-zigzag)の例を示します。

### 43.5 ステップ3のみを再計算する方法

事前にopenmxを実行し、NEGF計算が終了している場合には、直接ステップ3の計算を実行することができます。このような場合には、以下のキーワードを設定して下さい。

```
NEGF.tran.SCF.skip on
```

openmxによって*System.Name.tranb*を読み込まれ、事前のステップ2の計算結果を用いて、透過率、固有チャネルが順次計算されます。他方、ステップ2の計算までで止めたい場合には、キーワードを以下のとおり設定して下さい。

```
NEGF.tran.SCF.skip off  
NEGF.tran.Analysis off  
NEGF.tran.Channel off
```

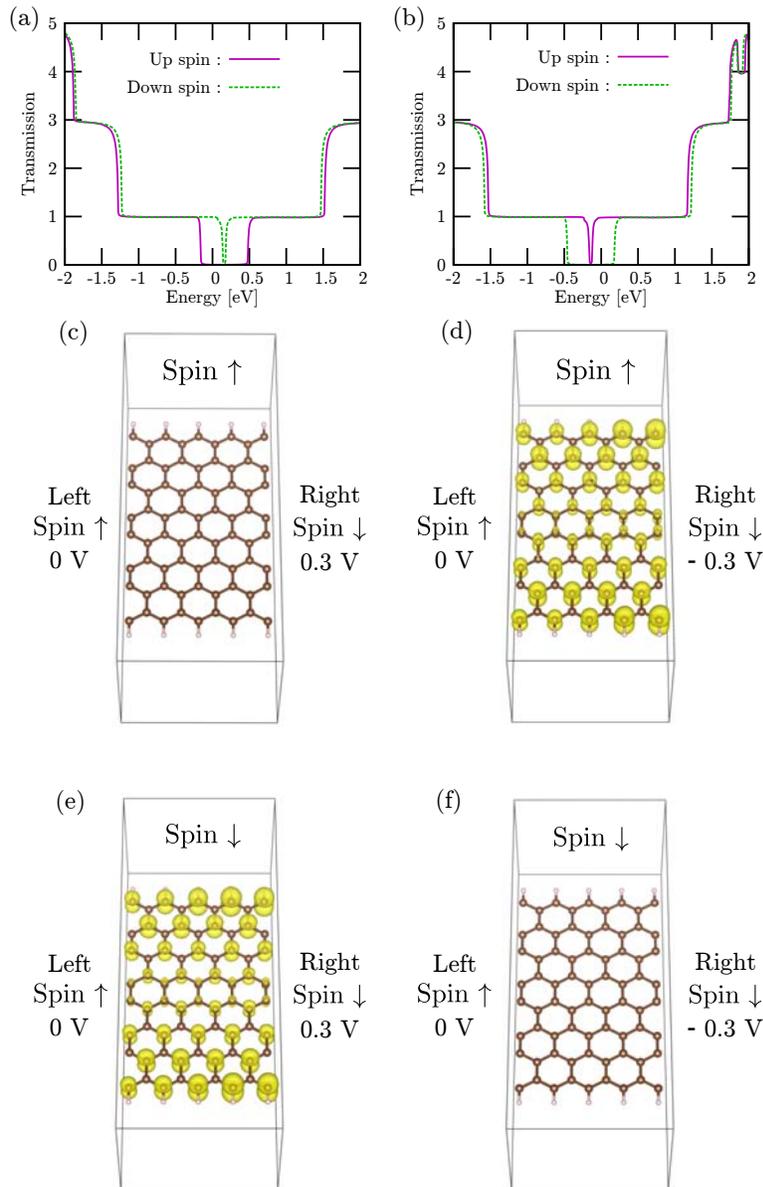


Figure 43: 反強磁性接合したグラフェンナノリボン (8-zigzag) の (a) スピン依存透過率 (バイアス電圧下: 0.3 V) (b) スピン依存透過率 (バイアス電圧下: -0.3 V) (c) 固有チャネル (バイアス電圧下: 0.3 V、エネルギー: 0 eV、スピン: ↑) (d) 固有チャネル (バイアス電圧下: -0.3 V、エネルギー: 0 eV、スピン: ↑) (e) 固有チャネル (バイアス電圧下: 0.3 V、エネルギー: 0 eV、スピン: ↓) (f) 固有チャネル (バイアス電圧下: -0.3 V、エネルギー: 0 eV、スピン: ↓) (スピンの↑は左領域、↓は右領域)。透過率及び固有チャネルは小さな値となっており、同じ値の等値面を描いてある。

### 43.6 ゼロバイアス下における周期系

bc面の周期性だけでなくa軸方向にも周期性を持った系の透過率は容易に計算可能です。3次元の周期性を持つ系の場合には、グリーン関数法を用いることなく、通常のバンド構造計算によって得られたハミルトニアンを利用し、ゼロバイアス電圧下での透過率が計算できます。この方法により、様々な幾何構造や磁気構造を持った系の輸送特性を低い計算コストで調べることが可能となります。とりわけ超格子構造などの計算には有用でしょう。この計算は、ステップ1のバンド構造計算において、キーワード「NEGF.Output.for.TranMain」を付け加えることで実行可能です。

```
NEGF.Output.for.TranMain    on
```

ステップ1の計算が正常に終了すると、ステップ3の計算に利用可能なファイル「\*.tranb」を生成されます。つまり、ステップ2の計算が省略されることとなります。

### 43.7 バイアス電圧効果の補間法

大規模な系のSCF計算を各バイアス電圧において実行するのは非常に計算時間を要します。バイアス電圧効果の補間法を用いてNEGF法による計算コストを低減することが可能です。次の手順により補間を行います。(i) 対象とするバイアス電圧領域から選択した2、3のバイアス電圧についてのSCF計算。(ii) 透過率および電流を計算するとき、中心散乱領域と右側電極部分に対するハミルトニアンのプロック要素、 $H_{\sigma,C}^{(k)}$ と $H_{\sigma,R}^{(k)}$ 、さらに化学ポテンシャル $\mu_R$ について次のような線形補間を行います。

$$\begin{aligned} H_{\sigma,C}^{(k)} &= \lambda H_{\sigma,C}^{(k,1)} + (1-\lambda) H_{\sigma,C}^{(k,2)}, \\ H_{\sigma,R}^{(k)} &= \lambda H_{\sigma,R}^{(k,1)} + (1-\lambda) H_{\sigma,R}^{(k,2)}, \\ \mu_R &= \lambda \mu_R^{(1)} + (1-\lambda) \mu_R^{(2)}, \end{aligned}$$

ここで、上付きの添字1および2は、事前にSCF計算を行ったバイアス電圧において、計算された、もしくは使用した量であることを意味します。補間の精度を保証するために、一般に内挿補間を行うべきであり、従って $\lambda$ は0から1の範囲の値に設定するべきです。

ステップ3の計算において、入力ファイルに次のキーワードを加えることにより補間を行います。

```
NEGF.tran.interpolate      on                # default=off, on|off
NEGF.tran.interpolate.file1 c1-negf-0.5.tranb
NEGF.tran.interpolate.file2 c1-negf-1.0.tranb
NEGF.tran.interpolate.coes  0.7 0.3         # default=1.0 0.0
```

補間を行う際には、キーワード「NEGF.tran.interpolate」を「on」に設定して下さい。上記の例では、キーワード「NEGF.tran.interpolate.file1」と「NEGF.tran.interpolate.file2」により指定されるファイル「c1-negf-0.5.tranb」と「c1-negf-1.0.tranb」には、それぞれ0.5Vおよび1.0Vのバイアス電圧下での計算結果が保存されます。キーワード「NEGF.tran.interpolate.coes」により重み0.7と0.3が指定されていますので、 $V = 0.7 \times 0.5 + 0.3 \times 1.0 = 0.65$  [V]における透過率と電流の値が本補間法により計算されます。

一次元炭素鎖の電流と透過率について、完全なSCF計算と補間法の比較を図44(a)および(b)に示します。補間法での計算では、0V、0.5Vおよび1.0Vの三つのバイアス電圧においてSCF計算を行い、

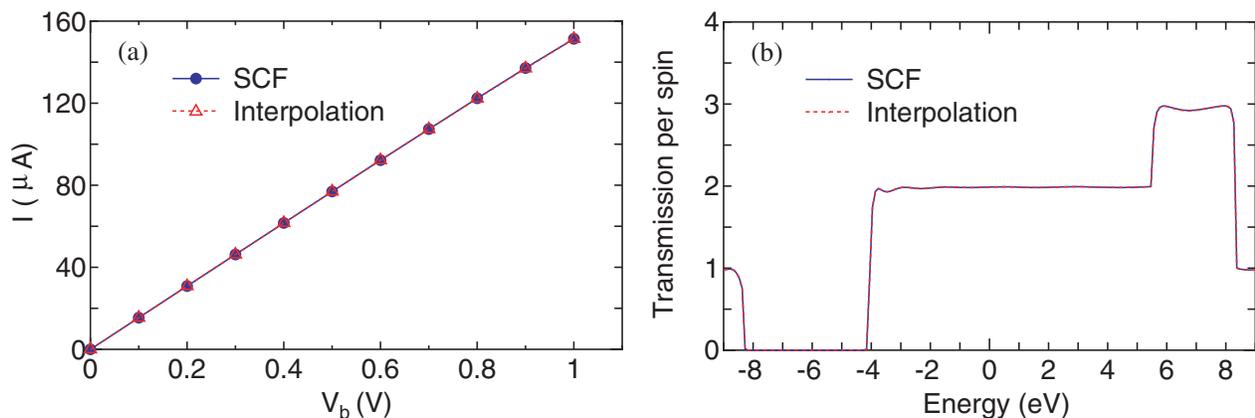


Figure 44: (a) SCF 計算（実線）と補間法（点線）により計算した一次元炭素鎖の電流、(b) SCF 計算（実線）と補間法（点線）により計算した 0.3V のバイアス電圧下での一次元炭素鎖の透過率。密度行列における非平衡成分の積分には虚数部 0.01eV と格子間隔 0.01eV を使用。

他のバイアス電圧における結果は補間で求めました。比較のために、補間法を用いずに完全な SCF 計算から求めた電流値も示しました。図 44 から、この簡易的な補間法は、電流および透過率のどちらにも非常に正確な結果を与えていることが確認できます。補間の際の SCF 計算で用いるバイアス電圧の適切な選択は系に依存しますが、この結果は、バイアス電圧の効果を計算精度を保ちつつ補間するために、本方法が非常に有用であることを示唆しています。

### 43.8 NEGF の並列化

NEGF 計算には MPI を用いた並列化が実装されています。MPI 並列化に加えて、MKL (Math Kernel Library) を用いると、グリーン関数を評価するときの行列乗算および逆行列計算を OpenMP により並列化することもできます。この場合、MPI/OpenMP によるハイブリッド並列化を実行可能で、さらに計算時間を短縮することができます。並列計算の実行方法は以前に述べたものと全く同じです。

図 45 に、NEGF 計算における OpenMP/MPI ハイブリッド並列の速度向上比を示します。これは 0.5 eV の有限バイアス電圧下にある 8-ジグザグ型グラフェンナノリボン (ZGNR) の密度行列の計算を行った場合の結果です。密度行列計算には 197 個のエネルギー点 (101 個は平衡密度行列、96 個は非平衡密度行列) を用いています。k 点サンプリングには  $\Gamma$  点のみを用い、スピン分極計算を行いました。エネルギー点、k 点、スピンの三つのループに関して 394 個の組み合わせを MPI により並列化しました。1 スレッドに相当するフラット MPI 並列化の速度向上比は 64 プロセスまで順当に向上しています。さらに、2 および 4 スレッドに相当するハイブリッド並列化により速度向上比が大きく改善されていることが分かります。64 個のプロセスと 4 スレッドに相当する 64 個のクアドコア・プロセッサを完全に用いた場合、速度向上比は約 140 であり、NEGF 法のスケラビリティが良いことを実証しています。詳細については文献 [73] を参照して下さい。OpenMX Ver. 3.9 では、MPI 並列化におけるプロセス数が原子数を超えても効率的に並列化が実行されることにも留意して下さい。

### 43.9 ノンコリニア DFT 法に対する NEGF 法

OpenMX Ver. 3.9 は、ノンコリニア DFT 法と組み合わせた NEGF 法に対応しています。ノンコリニア DFT 法の全ての機能と NEGF 法は整合性を持った実装となっています。スピン-軌道相互作用、DFT+U 法、そしてスピン磁気モーメントおよび軌道磁気モーメントの方位に対する制約法の全ての機能が NEGF

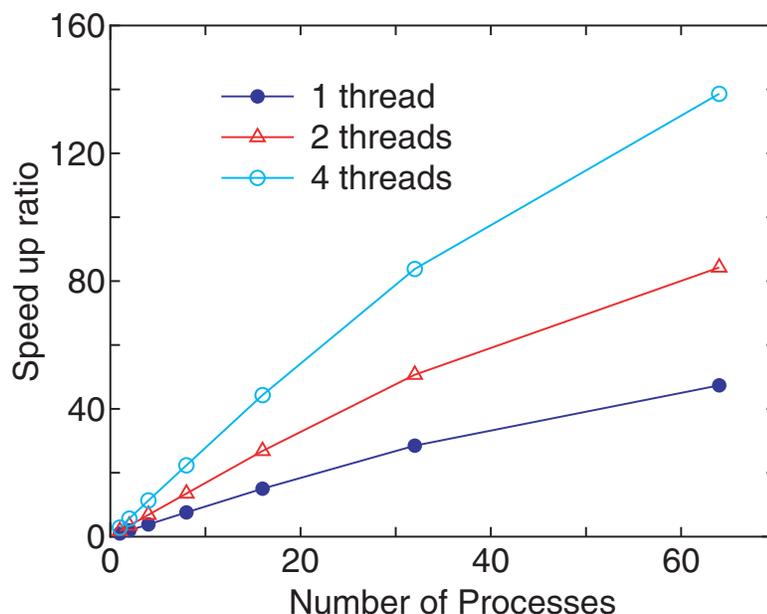


Figure 45: 8-ジグザググラフェンナノリボン (ZGNR) の密度行列の計算における OpenMP/MPI ハイブリッド並列の速度向上比。速度向上比は、 $T_1$  および  $T_p$  をシングルコア計算および並列計算の経過時間とすると、 $T_1/T_p$  により定義。プロセスおよびスレッドは MPI および OpenMP の並列計算で使用したコア数。並列計算は AMD opteron クワッドコア・プロセッサ (2.3 GHz) を搭載する CRAY-XT5 マシン上で実行。

法の実装と適合しています。従って、らせん状磁気構造を持つ磁区を介した電子輸送など、広範な問題を取り扱うことが可能であると期待されます。NEGF 計算の実行の方法は、基本的にコリニア DFT 法の場合と同一です。

ノンコリニア DFT 法と組み合わせた NEGF 計算の一例として、ジグザグ型グラフェン・ナノリボンの透過率の計算結果を図 46 に示します。ジグザグ端におけるスピン磁気モーメントは、左側および右側の電極部分のそれぞれにおいて、上方向および右方向に整列しています。電極部分の計算は、ステップ 1 においてスピン磁気モーメントの制約法を用いてノンコリニアバンド構造計算により行いました。次にステップ 2 の計算では制約条件を課しませんでした。ステップ 2 の SCF 計算の結果、図 46 (a) に示すように中心部分のスピン方向は徐々に回転していくことが分かります。この計算は、「work/negf\_example」ディレクトリに保存されている入力ファイル「Lead-L-8ZGNR-NC.dat」、「Lead-R-8ZGNER-NC.dat」、「NGEF-8ZGNR-NC.dat」により再現できます。また、同じディレクトリに金の一次元鎖のノンコリニア NEGF 計算の入力ファイルが保存されていますので、参考にして下さい。

#### 43.10 実例

NEGF 計算の参考例として、以下の 5 つの入力ファイルがディレクトリ「work/negf\_example」中に保存されています。

- ゼロバイアス電圧下にある一次元炭素鎖
  - ステップ 1: Lead-Chain.dat
  - ステップ 2: NEGF-Chain.dat
- ゼロバイアス電圧下にあるグラフェン・シート

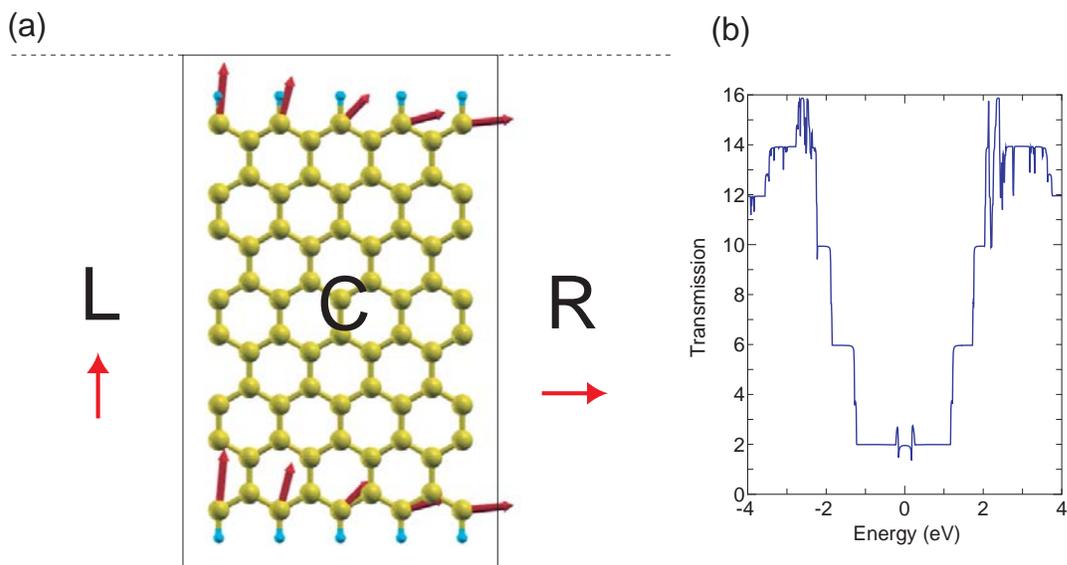


Figure 46: (a) 矢印で表したノンコリニアスピン方位を持つジグザグ型グラフェン・ナノリボン。矢印の長さは、スピン磁気モーメントの大きさに対応。ステップ 1 の計算において、スピン磁気モーメントの方位に対する制約法を適用。左側電極および右側電極において、ジグザグ端のスピン磁気モーメントがそれぞれ上方向および右方向となるように制約法を適用。(b) 図 (a) に示すチャンネル領域 C を通過する電子の透過率。

ステップ 1: Lead-Graphene.dat

ステップ 2: NEGF-Graphene.dat

- 0.3V の有限バイアス電圧下にある反強磁性接合を持つ 8-ジグザグ型グラフェン・ナノリボン

ステップ 1: Lead-L-8ZGNR.dat, Lead-R-8ZGNR.dat

ステップ 2: NEGF-8ZGNR-0.3.dat

- ゼロバイアス電圧下にあるノンコリニア磁気接合を持つ 8-ジグザグ型グラフェン・ナノリボン

ステップ 1: Lead-L-8ZGNR-NC.dat, Lead-R-8ZGNR-NC.dat

ステップ 2: NEGF-8ZGNR-NC.dat

- ノンコリニア NEGF 法で計算したゼロバイアス電圧下にある金の一次元鎖

ステップ 1: Lead-Au-Chain-NC.dat

ステップ 2: NEGF-Au-Chain-NC.dat

#### 43.11 NEGF の自動実行テスト

NEGF 計算に関連する機能が適切にインストールされていることを確認するために、NEGF 計算の自動実行テストを次のようにして行うことができます。

##### MPI 並列実行の場合

```
% mpirun -np 16 openmx -runtestNEGF
```

## MPI/OpenMP 並列実行の場合

```
% mpirun -np 8 openmx -runtestNEGF -nt 2
```

このテスト計算において、OpenMX はステップ 1 およびステップ 2 の計算を含む五つのテストケースを実行し、「work/negf\_example」に保存されている参照結果と比較します。比較結果（全エネルギー、力、平均電流密度及び固有透過率の和の絶対差異）は、「work」ディレクトリ内にファイル「runtestNEGF.result」として保存されます。参照結果は、2.6 GHz Xeon マシンの 16 MPI プロセスを用いて計算されたものです。絶対差異が少数点以下 7 桁以内であれば、インストールが正常であると判断されます。

## 44 最局在ワニエ関数

### 44.1 生成方法

OpenMX Ver. 3.9 を用いて最局在ワニエ関数 (MLWF) を生成することが可能です [122, 123]。MLWF を計算するためのキーワードと設定について以下に説明します。キーワードの様式は、OpenMX で本来用いている形式に厳密に従っています。具体的な例として、この章ではダイヤモンド構造のシリコンに関する計算結果を示します。計算はディレクトリ「work/wf\_example」中にある入力ファイル「Si.dat」を用いて、プログラムコード「openmx」を用いて再現できます。その他のポストプロセスコードは使用しません。通常の SCF 計算の収束解が得られた後に、再スタートファイルを用い、適切な MLWF が得られるまで、いくつかのパラメータを変更しながら以下に説明する手順を繰り返すことになります。

本機能を用いて発表を行う際には、文献 [77] を引用して頂けると幸いです。

#### MLWF を生成する

MLWF を生成するために、キーワード「Wannier.Func.Calc」を「on」と明示的に設定して下さい。デフォルト値は「off」です。

```
Wannier.Func.Calc      on          #default off
```

#### 生成する MLWF の数の指定

生成する MLWF の数を、キーワード「Wannier.Func.Num」により指定します。デフォルト値はありません。

```
Wannier.Func.Num      4          #no default
```

#### ブロッホ状態を選択するエネルギーウィンドウ

MLWF は、1 組のブロッホ状態から生成されます。固有エネルギーに対するエネルギー窓を指定することで、1 組のブロッホ状態が選択されます。参考文献 [123] に従って、二つのエネルギー窓を導入します。1 つは、外エネルギー窓とよばれ、下限を示す「Wannier.Outer.Window.Bottom」と上限を示す「Wannier.Outer.Window.Top」の 2 つのキーワードにより指定されます。もう 1 つは、内エネルギー窓で、下限を示す「Wannier.Inner.Window.Bottom」と上限を示す「Wannier.Inner.Window.Top」の 2 つの同様なキーワードにより指定されます。これらの 4 つの値はすべて、フェルミ準位からの相対的な値を eV 単位で与えます。内エネルギー窓は外エネルギー窓の領域内に設定する必要があります。内エネルギー窓を定義する下限値と上限値が等しいときには、内エネルギー窓は定義されていないことになり、計算には使用できません。外エネルギー窓にデフォルト値はなく、また内エネルギー窓の下限値と上限値のデフォルト値はともに 0.0 です。例えば、次の様に設定します。

```
Wannier.Outer.Window.Bottom  -14.0  #lower boundary of outer window, no default value
Wannier.Outer.Window.Top      0.0    #upper boundary of outer window, no default value
Wannier.Inner.Window.Bottom    0.0    #lower boundary of inner window, default value 0.0
Wannier.Inner.Window.Top      0.0    #upper boundary of outer window, default value 0.0
```

目的のバンドを含む 2 つのエネルギー窓を適切に設定するためには、MLWF の計算の前にバンド構造と状態密度の一方または両方を計算し、その範囲を事前に知っておく必要があります。

MLWF の局在化計算では再スタートファイルから重なり行列要素を取得し、計算を行います。重なり行列はある特定の外エネルギー窓に対して計算され、ファイルに保存されています。従って再計算の際に外エネルギー窓を再定義する場合には、事前に設定した外エネルギー窓の範囲内での変更が可能です。また計算を再スタートする際には、内エネルギー窓は外エネルギー窓の範囲内で自由に変更可能です。MLWF の両エネルギー窓の依存性を確かめる際には、上述の制約の範囲でエネルギー窓を調節して下さい。計算の再スタートさせる場合には、本節の後半にある「重なり行列の計算を省いた最適化の再スタート」を参照して下さい。

## MLWF の最初の推定

キーワード「Wannier.Initial.Guess」を「on」または「off」と設定することにより、MLWF の初期推定をするか否かを選択できます。デフォルト値は「on」です。多くの場合に MLWF の初期推定を行うことで、スプレッド関数を最小化する際の収束性が改善し、また局所的な解への収束を避けることが可能になります。初期推定を行う際には、生成する MLWF と同数の局在関数の組を定義しなければなりません。外エネルギー窓の内部のプロッホ波動関数は、局在関数の組に射影されます。従って、このような局在関数はプロジェクトとも呼びます。プロジェクトを指定するには、以下の手順が必要です。

### A. プロジェクトに用いる局在関数の定義

擬原子軌道 (PAO) をプロジェクトに用いますので、PAO の設定は基底関数の場合と同様です。ダイヤモンド構造のシリコンでの設定例は、以下のようになります。

```
Species.Number          2

<Definition.of.Atomic.Species
  Si      Si7.0-s2p2d1    Si_CA19
  proj1   Si7.0-s1p1d1f1 Si_CA19
Definition.of.Atomic.Species>
```

この例では、Si の PAO をプロジェクトとして使用し、原子種「proj1」を上記のように定義します。キーワードの対「<Definition.of.Atomic.Species」および「Definition.of.Atomic.Species>」内で、1 行目の Si 原子に加えて、プロジェクトに関する原子種「proj1」を定義します。この原子種「proj1」は「Si7.0-s1p1d1f1」と擬ポテンシャル「Si\_CA19」で定義されます。ただし、この行で定義された擬ポテンシャルは実際の計算では使用されません。単にデータ入力の一貫性を保つために与えています。どのような PAO もプロジェクトとして使用可能ですが、各 1-成分に対しては単一の動径軌道のみが指定可能であり、すべての場合に「s1p1d1f1」の指定を推奨します。

### B. プロジェクトの軌道、中心位置、配向の指定

対となるキーワード「<Wannier.Initial.Projectos」および「Wannier.Initial.Projectos>」を用いて、プロジェクト名、局在軌道関数、局在軌道の中心、軌道の方位を指定する局所的な  $z$  軸および  $x$  軸を指定します。設定例を以下に示します。

```

<Wannier.Initial.Projectors
proj1-sp3  0.250  0.250  0.250  -1.0  0.0  0.0    0.0  0.0 -1.0
proj1-sp3  0.000  0.000  0.000    0.0  0.0  1.0    1.0  0.0  0.0
Wannier.Initial.Projectors>

```

各行には次の項目が記載されています。例えば、第1行目では、原子種名「proj1」はキーワード対「Definition.of.Atomic.Species」で定義されています。プロジェクト名と選択された軌道をつなぐためにハイフン「-」を用います。「sp3」は、この原子種の  $sp_3$  混成軌道が生成するワニエ関数の初期推定として使用されることを意味します（利用可能な軌道およびその混成軌道を表9に示します）。この混成軌道からなるプロジェクトは、次に続く3つの数「0.25 0.25 0.25」で与えられる位置を中心として配置されます。これらの数値は、キーワード「Wannier.Initial.Projectors.Unit」により定義される単位で与えられています（説明は後述します）。その次の3つの数から成る2組の値は、局所座標系の  $z$  軸および  $x$  軸の方向をそれぞれ定義しています。ここで、各軸は  $xyz$  座標系の3つの成分により定義されるベクトルにより指定されます。この例の第1行目では、局所  $z$  軸は元の  $x$  軸に対して逆向きの「-1.0 0.0 0.0」で表されるベクトルで定義され、局所  $x$  軸はもとの  $z$  軸に対して逆向きの「0.0 0.0 -1.0」で表されるベクトルで定義されています。第2行目では、局所軸は元の座標系と同じです。

プロジェクトとして用いる軌道として、PAO 自身かまたはそれらの混成軌道も使用可能です。「sp3」により定義されるプロジェクトの総数は4個であること注意して下さい。同様に、「sp」および「sp2」により定義されるプロジェクトの総数は、それぞれ2個および3個です。サポートされているPAOおよびそれらの混成軌道のリストは表9にまとめられています。このリストに記載されていないプロジェクトは使用できませんので注意して下さい。

プロジェクトは、単位胞内のどこにでも配置することができます。その位置の指定には、単位胞ベクトルに相対的な規格化座標（FRAC）か、原子単位（AU）もしくはオングストローム（ANG）の単位で表したデカルト座標が使用できます。その選択はキーワード「Wannier.Initial.Projectors.Unit」で行います。

```

Wannier.Initial.Projectors.Unit    FRAC    #AU, ANG or FRAC

```

#### k 点グリッドおよび隣接する k 点を結ぶ b ベクトル

キーワード「Wannier.Kgrid」により、Monkhorst-Pack の  $k$  点グリッドを指定します。デフォルトの設定はありません。 $k$  空間における微分を計算する際に有限差分を用いるために、中心の  $k$  点からの距離に応じて殻（shell）ごとに隣接  $k$  点を結ぶ  $b$  ベクトルを探索します。探索する殻の最大数は、キーワード「Wannier.Maxshells」により指定します。デフォルト値は12で、適切な  $b$  ベクトルの組を見出すことができない場合には、最大数を増やして下さい。単位格子ベクトル間で大きなアスペクト比を持つ系の場合、問題が起きることがあります。その場合にはエラーメッセージが表示されます。しかし通常は、最大数12で有効に機能します。「Wannier.Kgrid」の適切な設定もまた、 $b$  ベクトルを見つけるときに役立ちます。ここで、各逆格子ベクトルに対する離散化のグリッド間隔は、互いにほとんど同等でなければなりません。

```

Wannier.MaxShells          12          # default value is 12.
Wannier.Kgrid              8 8 8        # no default value

```

Table 9: プロジェクタ用の軌道。軌道の方位は  $z$  軸および  $x$  軸を再定義することで回転可能。

| Orbital name | Number of included projector | Description   |
|--------------|------------------------------|---|
| s            | 1                            | $s$ orbital from PAOs   |
| p            | 3                            | $p_x, p_y, p_z$ from PAOs   |
| px           | 1                            | $p_x$ from PAOs   |
| py           | 1                            | $p_y$ from PAOs   |
| pz           | 1                            | $p_z$ from PAOs   |
| d            | 5                            | $d_{z^2}, d_{x^2-y^2}, d_{xy}, d_{xz}, d_{yz}$ from PAOs  |
| dz2          | 1                            | $d_{z^2}$ from PAOs   |
| dx2-y2       | 1                            | $d_{x^2-y^2}$ from PAOs   |
| dxy          | 1                            | $d_{xy}$ from PAOs  |
| dxz          | 1                            | $d_{xy}$ from PAOs  |
| dyz          | 1                            | $d_{xy}$ from PAOs  |
| f            | 7                            | $f_{z^3}, f_{xz^2}, f_{yz^2}, f_{zx^2}, f_{xyz}, f_{x^3-3xy^2}, f_{3yx^2-y^3}$ from PAOs  |
| fz3          | 1                            | $f_{z^3}$ from PAOs   |
| fxz2         | 1                            | $f_{xz^2}$ from PAOs  |
| fyz2         | 1                            | $f_{yz^2}$ from PAOs  |
| fzx2         | 1                            | $f_{zx^2}$ from PAOs  |
| fxyz         | 1                            | $f_{xyz}$ from PAOs   |
| fx3-3xy2     | 1                            | $f_{x^3-3xy^2}$ from PAOs   |
| f3yx2-y3     | 1                            | $f_{3yx^2-y^3}$ from PAOs   |
| sp           | 2                            | Hybridization between $s$ and $p_x$ orbitals, including $\frac{1}{\sqrt{2}}(s + p_x)$ and $\frac{1}{\sqrt{2}}(s - p_x)$   |
| sp2          | 3                            | Hybridization among $s, p_x,$ and $p_y$ orbitals, including $\frac{1}{\sqrt{3}}s - \frac{1}{\sqrt{6}}p_x + \frac{1}{\sqrt{2}}p_y, \frac{1}{\sqrt{3}}s - \frac{1}{\sqrt{6}}p_x - \frac{1}{\sqrt{2}}p_y$ and $\frac{1}{\sqrt{3}}s + \frac{2}{\sqrt{6}}p_x$  |
| sp3          | 4                            | Hybridization among $s, p_x, p_y$ and $p_z$ orbitals:<br>$\frac{1}{\sqrt{2}}(s + p_x + p_y + p_z), \frac{1}{\sqrt{2}}(s + p_x - p_y - p_z)$<br>$\frac{1}{\sqrt{2}}(s - p_x + p_y - p_z), \frac{1}{\sqrt{2}}(s - p_x - p_y + p_z)$   |
| sp3dz2       | 5                            | Hybridization among $s, p_x, p_y, p_z$ and $d_{z^2}$ orbitals:<br>$\frac{1}{\sqrt{3}}s - \frac{1}{\sqrt{6}}p_x + \frac{1}{\sqrt{2}}p_y,$<br>$\frac{1}{\sqrt{3}}s - \frac{1}{\sqrt{6}}p_x + \frac{1}{\sqrt{2}}p_y, \frac{1}{\sqrt{3}}s - \frac{2}{\sqrt{6}}p_x$<br>$\frac{1}{\sqrt{2}}p_z + \frac{1}{\sqrt{2}}d_{z^2}, -\frac{1}{\sqrt{2}}p_z + \frac{1}{\sqrt{2}}d_{z^2}$   |
| sp3deg       | 6                            | Hybridization among $s, p_x, p_y, p_z$ and $d_{z^2}, d_{x^2-y^2}$ orbitals:<br>$\frac{1}{\sqrt{6}}s - \frac{1}{\sqrt{2}}p_x - \frac{1}{\sqrt{12}}d_{z^2} + \frac{1}{2}d_{x^2-y^2},$<br>$\frac{1}{\sqrt{6}}s + \frac{1}{\sqrt{2}}p_x - \frac{1}{\sqrt{12}}d_{z^2} + \frac{1}{2}d_{x^2-y^2},$<br>$\frac{1}{\sqrt{6}}s - \frac{1}{\sqrt{2}}p_y - \frac{1}{\sqrt{12}}d_{z^2} - \frac{1}{2}d_{x^2-y^2},$<br>$\frac{1}{\sqrt{6}}s + \frac{1}{\sqrt{2}}p_y - \frac{1}{\sqrt{12}}d_{z^2} - \frac{1}{2}d_{x^2-y^2},$<br>$\frac{1}{\sqrt{6}}s - \frac{1}{\sqrt{2}}p_z + \frac{1}{\sqrt{3}}d_{z^2}, \frac{1}{\sqrt{6}}s + \frac{1}{\sqrt{2}}p_z + \frac{1}{\sqrt{3}}d_{z^2}$ |

## ワニエ関数の広がり (スプレッド) の最小化

もつれたバンド (entangled band) の場合 [123]、MLWF を見出すためには2つのステップが必要となります。最初のステップは、非孤立バンドのもつれを解くことによりスプレッド関数のゲージ不変部分を最小化することです。第2ステップは孤立バンドの場合と同様です [122]。ゲージ依存部分は、スプレッド関数の勾配に応じて、選択したプロッホ波動関数のユニタリー変換により最適化します。最初のステップでは、3つのパラメータを用いて MLWF を生成するための自己無撞着ループを制御します。それらは、SCF ループの最大数「Wannier.Dis.SCF.Max.Steps」、収束基準「Wannier.Dis.Conv.Criterion」、そして入出力部分空間プロジェクトの混合を制御するパラメータ「Wannier.Dis.Mixing.Para」です。

```
Wannier.Dis.SCF.Max.Steps      2000      # default 200
Wannier.Dis.Conv.Criterion     1e-12     # default 1e-8
Wannier.Dis.Mixing.Para       0.5       # default value is 0.5
```

第2ステップにおいては、3種類の最適化法が利用できます。1つ是最急降下法 (SD: steepest descent) で、2つ目は共役傾斜法 (CG: conjugate gradient) です。3つ目は、始めにSD法を用いその後CG法に切り替えるハイブリッド法です。キーワード「Wannier.Minimizing.Scheme」で、どの方法を使うかを指定します。0は簡易なSD法、1はCG法、2はハイブリッド法です。SD法のステップ長をキーワード「Wannier.Minimizing.StepLength」により指定します。CG法では、割線法 (secant method) を用いて最適ステップ長を求めます。最大割線ステップと初期ステップ長を、それぞれ「Wannier.Minimizing.Secant.Steps」と「Wannier.Minimizing.Secant.StepLength」により指定します。また最小化ステップの最大数および収束条件を、それぞれ「Wannier.Minimizing.Max.Steps」および「Wannier.Minimizing.Conv.Criterion」により指定します。

```
Wannier.Minimizing.Scheme      2         # default 0, 0=SD 1=CG 2=hybrid
Wannier.Minimizing.StepLength  2.0      # default 2.0
Wannier.Minimizing.Secant.Steps 5         # default 5
Wannier.Minimizing.Secant.StepLength 2.0     # default 2.0
Wannier.Minimizing.Conv.Criterion 1e-12    # default 1e-8
Wannier.Minimizing.Max.Steps    200      # default 200
```

ハイブリッド法でのSD法およびCG法の最大の最適化ステップ数は「Wannier.Minimizing.Max.Steps」で指定した値となります。

## 重なり行列の計算を省いた最適化の再スタート

一旦、重なり行列  $M_{mn}^{(k,b)}$  を計算し、ファイルに保存しておけば、再計算が容易に実行できます。事前に計算された重なり行列  $M_{mn}^{(k,b)}$  を利用し、再計算する場合には、キーワード「Wannier.Readin.Overlap.Matrix」を「on」と設定して下さい。

```
Wannier.Readin.Overlap.Matrix  on        # on|off, default is on
```

重なり行列の計算は時間を要しますので、このキーワードを指定することで、無駄な計算を省くことが可能となります。このキーワードが指定している場合にはプログラムコード「openmx」は、ファイルから固有エネルギーおよび固有状態と共に、重なり行列を読み取ります。外エネルギー窓とk点グリッドは、保存された重なり行列と固有値の計算時に用いたものと同じでなければならないことに注意し

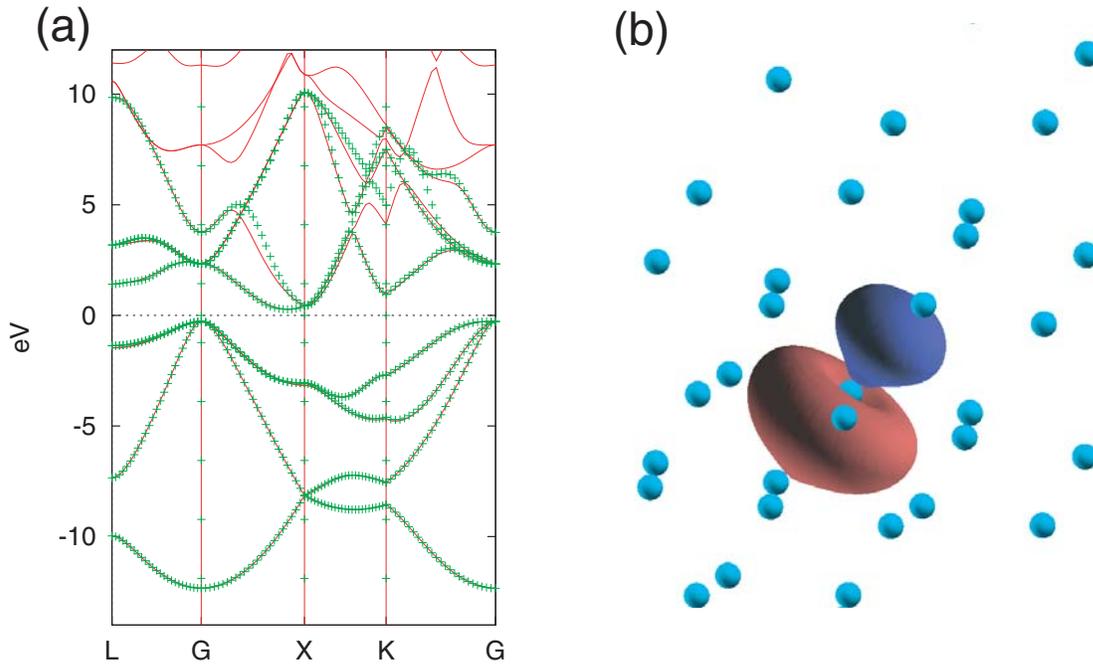


Figure 47: (a) ダイヤモンド構造におけるSiの補間バンド構造（シンボル線）と元のバンド構造（実線）との比較。(b) ダイヤモンド構造におけるSiのフェルミ準位近くの4つの価電子状態と4つの伝導状態からなる8つのMLWFの一つ。sp<sup>3</sup>混成の初期推定から計算したもの。

て下さい。読み込み時にそれらのパラメーターの一貫性がチェックされます。内エネルギー窓、MLWFの初期推定、また収束条件に関しては、最適化を再スタートする際に変更することができます。「Wannier.Readin.Overlap.Matrix」が「off」に設定されている場合には、重なり行列が計算されて自動的にファイルに保存されます。「System.Name」で定義された名前と「.mmn」という拡張子を持つファイルが生成されます。また固有エネルギーと固有状態も、拡張子「.eigen」を持ったファイルとして保存されます。

## 44.2 解析

### 補間バンド構造の表示

補間バンド構造を表示するためには、キーワード「Wannier.Interpolated.Bands」を「on」に設定して下さい。

```
Wannier.Interpolated.Bands      on      # on|off, default=off
```

k経路と経路に沿ったサンプリング密度などの他の必要な設定は、OpenMXにおけるバンド分散を表示するための設定と同様です。従って、補間バンド構造を作成するために、キーワード「Band.dispersion」を「on」に設定しなければなりません。収束後、補間バンド分散データは「.Wannier\_band」の拡張子を持つファイルに保存されます。このファイルは「.Band」ファイルと同じ形式です。一例として、ダイヤモンド構造におけるSiの補間バンド構造と元のバンド構造を図47(a)に示します。

## MLWF の描画

MLWF を描画するためには、キーワード「Wannier.Function.Plot」を「on」に設定して下さい。デフォルト値は「off」です。

```
Wannier.Function.Plot          on          # default off
Wannier.Function.Plot.SuperCells 1 1 1     # default=0 0 0
```

キーワード「Wannier.Function.Plot」を「on」に設定すると、すべての MLWF を描画するためのファイルが生成されます。MLWF は、例えば「.mlwfl\_4r.cube」のファイル拡張子を持つ Gaussian Cube 形式のファイルで保存されます。このファイルは、HOMO や LUMO といった分子軌道ファイルと同じ形式で名前が付けられています。「.mlwfl」の後の数字はスピン・インデックスを、次の文字は MLWF のインデックスで、最後の文字「r」は real を、「i」は imaginary を表し、それぞれ MLWF の実部と虚数部という意味です。ユーザーは、MLWF を描画するスーパーセルのサイズを設定できます。キーワード「Wannier.Function.Plot.SuperCells」によりそのサイズを指定します。上の例における「1 1 1」は、単位格子セルを中心に置き、a 軸、b 軸、c 軸方向のプラスおよびマイナス方向に 1 だけ拡大するという意味です。従って、この場合の MLWF は、 $27$  個  $= (1 \times 2 + 1) \times (1 \times 2 + 1) \times (1 \times 2 + 1)$  のセルからなる拡張セルに描画されます。図 47 (b) は、ダイヤモンド構造における Si のフェルミ準位近くの 4 つの価電子状態と 4 つの伝導状態からなる 8 つの収束した MLWF 内の一つを示したものです。

### 44.3 スプレッド関数の最適化過程の確認

最適化の過程は標準出力に書き出されます。以下の方法により、最適化の進行過程を確認して下さい。ここでは便宜上、標準出力はファイル「stdout.std」に保存されていると仮定します。以下で示す例は、「openmx\*/./work/wf\_example」中の入力ファイル「Si.dat」を用いて、ユーザー各自が同じ計算を実行することができます。

## DISE

次の様にして、バンドのもつれをほどく処理（最適化の第 1 段階）の自己無撞着ループが確認できます。

```
% grep "DISE" stdout.std

| Iter | Omega_I (Angs^2) | Delta_I (Angs^2) | ----> DISE
|   1  | 18.371525257652 | 18.371525257652 | ----> DISE
|   2  | 17.955767336391 | -0.415757921261 | ----> DISE
|   3  | 17.659503060694 | -0.296264275698 | ----> DISE
|   4  | 17.454033576174 | -0.205469484520 | ----> DISE
|   5  | 17.311180447271 | -0.142853128902 | ----> DISE
|   6  | 17.210945408916 | -0.100235038355 | ----> DISE
|   7  | 17.139778800398 | -0.071166608519 | ----> DISE
|   8  | 17.088603102826 | -0.051175697572 | ----> DISE
|   9  | 17.051329329614 | -0.037273773211 | ----> DISE
|  10  | 17.023842837298 | -0.027486492316 | ----> DISE
```

```

.....
.....
...
.

```

ここで、「Iter」は繰り返し数、「Omega\_I」はスプレッド関数のゲージ不変部分、「Delta\_I」は隣接する2つのステップ間の「Omega\_I」の差をそれぞれ意味します。キーワード「Wannier.Dis.Conv.Criterion」に与えた収束条件が「Delta\_I」に適用されます。

## CONV

次の様にして、スプレッド関数のゲージ依存部分の最適化（最適化の第2段階）過程が確認できます。

```
% grep "CONV" stdout.std
```

```

Opt Step |Mode of Gradient|d_Omega_in_steps|      d_Omega      | (in Angs^2) ---> CONV
| SD     1 | 6.52434844E-01 | 5.41612774E-04 | -5.41340331E-04| ---> CONV
| SD     2 | 6.51123660E-01 | 5.40524307E-04 | -5.40253165E-04| ---> CONV
.....
.....
| SD    200 | 4.77499752E-01 | 3.96392019E-04 | -3.96271308E-04| ---> CONV
|Opt Step |Mode of Gradient|      d_Omega      | (Angs^2) ---> CONV
| CG     1 | 8.61043764E-01 | -3.24716990E-01| ---> CONV
.....
.....
| CG    58 | 1.67083857E-12 | -5.37225101E-13| ---> CONV
| CG    59 | 5.44431651E-13 | -1.98972260E-13| ---> CONV
***** ---> CONV
CONVERGENCE ACHIEVED ! ---> CONV
***** ---> CONV
CONVERGENCE ACHIEVED ! ---> SPRD

```

ここで、「Opt Step」はSD法またはCG法のどちらかにおける最適化ステップであり、「Mode of Gradient」は、スプレッド関数の勾配の絶対値です。SD法では、ゲージ依存スプレッド関数における隣接する2つのステップ間の差が2つの異なる方法で計算されます。その結果が「d\_Omega\_in\_steps」および「d\_Omega」です。「d\_Omega\_in\_steps」は、

$$d\Omega = \epsilon \sum_k \|G^{(k)}\|^2,$$

であり、 $\epsilon$ はステップ長、 $G^{(k)}$ はスプレッド関数の勾配です。この式の詳細は参考文献 [122] に説明されています。一方、「d\_Omega」は

$$d\Omega = \Omega^{(n+1)} - \Omega^{(n)},$$

により与えられます。ここで、 $n$  は反復（繰り返し）数です。CG 法では、「d\_Omega」のみを評価します。キーワード「Wannier.Minimizing.Conv.Criterion」で与えられた収束条件が、「Mode of Gradient」に適用されます。

## SPRD

次の様にして、ワニエ関数の広がりの変化を確認できます。

```
% grep "SPRD" stdout.std
```

```
|Opt Step |      Omega_I |      Omega_D |      Omega_OD |      Tot_Omega | (in Angs^2) ----> SPRD
| SD   1 | 16.93053479 | 0.13727387 | 6.57748455 | 23.64529321 | ----> SPRD
| SD   2 | 16.93053479 | 0.13724827 | 6.57696989 | 23.64475295 | ----> SPRD
| SD   3 | 16.93053479 | 0.13722279 | 6.57645620 | 23.64421378 | ----> SPRD
| SD   4 | 16.93053479 | 0.13719743 | 6.57594347 | 23.64367569 | ----> SPRD
.....
.....
| SD 199 | 16.93053479 | 0.13399285 | 6.48989479 | 23.55442243 | ----> SPRD
| SD 200 | 16.93053479 | 0.13398326 | 6.48950811 | 23.55402616 | ----> SPRD
|Opt Step |      Omega_I |      Omega_D |      Omega_OD |      Tot_Omega | (Angs^2) ----> SPRD
| CG   1 | 16.93053479 | 0.15480701 | 6.14396737 | 23.22930917 | ----> SPRD
| CG   2 | 16.93053479 | 0.17172507 | 5.87830203 | 22.98056189 | ----> SPRD
| CG   3 | 16.93053479 | 0.17012089 | 5.78940789 | 22.89006357 | ----> SPRD
.....
.....
| CG  57 | 16.93053479 | 0.16557875 | 5.73752928 | 22.83364282 | ----> SPRD
| CG  58 | 16.93053479 | 0.16557876 | 5.73752928 | 22.83364282 | ----> SPRD
| CG  59 | 16.93053479 | 0.16557876 | 5.73752928 | 22.83364282 | ----> SPRD
***** ----> SPRD
CONVERGENCE ACHIEVED ! ----> SPRD
***** ----> SPRD
```

ここで、「Opt Step」はSD法またはCG法のどちらかにおける最適化ステップです。「Omega\_I」はスプレッド関数のゲージ不変部分です。「Omega\_D」はおよび「Omega\_OD」は、それぞれゲージ依存の対角成分および非対角成分からの寄与です。「Tot\_Omega」は、スプレッド関数の上記3つの成分の総和です。

## CENT

次の様にしてワニエ関数中心の変化を確認できます。

```
% grep "CENT" stdout.std
```

```
WF 1 ( 1.14164289, 1.14164298, 1.14164266) | 2.95573380 ---->CENT
WF 2 ( 1.55716251, 1.55716342, 1.14164203) | 2.95572597 ---->CENT
WF 3 ( 1.55716191, 1.14164295, 1.55716190) | 2.95572978 ---->CENT
WF 4 ( 1.14164389, 1.55716087, 1.55716055) | 2.95572957 ---->CENT
WF 5 ( 0.20775982, 0.20775967, 0.20775893) | 2.95572677 ---->CENT
WF 6 ( 0.20776045, -0.20775959, -0.20775914) | 2.95572605 ---->CENT
WF 7 (-0.20775851, 0.20775981, -0.20775888) | 2.95572925 ---->CENT
```

```

WF   8 (-0.20775787,-0.20775767, 0.20775933) | 2.95573335 --->CENT
Total Center ( 5.39761509, 5.39761243, 5.39760738) sum_spread 23.64583455 --->CENT
SD   1 -----> CENT
WF   1 ( 1.14164582, 1.14164592, 1.14164559) | 2.95566613 --->CENT
WF   2 ( 1.55715957, 1.55716049, 1.14164497) | 2.95565831 --->CENT
WF   3 ( 1.55715897, 1.14164588, 1.55715897) | 2.95566211 --->CENT
WF   4 ( 1.14164683, 1.55715794, 1.55715761) | 2.95566190 --->CENT
WF   5 ( 0.20775689, 0.20775673, 0.20775599) | 2.95565910 --->CENT
WF   6 ( 0.20775752,-0.20775666,-0.20775620) | 2.95565838 --->CENT
WF   7 (-0.20775558, 0.20775687,-0.20775594) | 2.95566158 --->CENT
WF   8 (-0.20775493,-0.20775474, 0.20775639) | 2.95566569 --->CENT
Total Center ( 5.39761509, 5.39761243, 5.39760738) sum_spread 23.64529321 --->CENT
SD   2 -----> CENT
.....
.....
CG   59 -----> CENT
WF   1 ( 1.14585349, 1.14584696, 1.14584386) | 2.85421846 --->CENT
WF   2 ( 1.55295615, 1.55294970, 1.14584792) | 2.85422167 --->CENT
WF   3 ( 1.55296133, 1.14584610, 1.55295139) | 2.85421070 --->CENT
WF   4 ( 1.14584053, 1.55296761, 1.55296391) | 2.85417080 --->CENT
WF   5 ( 0.20356211, 0.20355857, 0.20355600) | 2.85418933 --->CENT
WF   6 ( 0.20355119,-0.20355008,-0.20355192) | 2.85422458 --->CENT
WF   7 (-0.20355306, 0.20355395,-0.20355905) | 2.85420611 --->CENT
WF   8 (-0.20355603,-0.20356000, 0.20355520) | 2.85420117 --->CENT
Total Center ( 5.39761571, 5.39761281, 5.39760730) sum_spread 22.83364282 --->CENT

```

「SD」または「CG」で始まる行に、最適化法とそのステップ数が示されています。「WF」で始まる行は、各ワニエ関数の中心の  $(x, y, z)$  座標をÅ 単位で、またその広がりをÅ<sup>2</sup> 単位で示しています。すべてのワニエ関数中心と広がりの総和を、「Total Center」で始まる行に示しています。

#### 44.4 MLWF の作成例

様々な物質の MLWF の作成例として、ディレクトリ「work/wf\_example」に入力ファイルが用意されています。

- Benzene.dat  
ベンゼンの 6 個の  $\pi$  分子軌道から  $p_z$  軌道型のワニエ関数を生成します。
- GaAs.dat  
GaAs の 4 つの価電子バンドから最局在ワニエ関数を生成します。
- Si.dat  
Si の価電子バンドおよび伝導バンドの両方を含む 8 つのワニエ関数を生成します。初期値は  $sp_3$  混成軌道です。
- symGra.dat  
グラフェンシートワニエ関数を生成します。初期値は、炭素原子上の  $sp_2$  混成軌道と  $p_z$  軌道です。

- pmCVO.dat  
非磁性の立方ペロブスカイト  $\text{CaVO}_3$  に対する  $t_{2g}$  型のワニエ関数を生成します。
- NC\_CVO.dat  
「pmCVO.dat」の場合にスピン・軌道相互作用を含めた計算例です。
- GaAs\_NC.dat  
「GaAs.dat」の場合にスピン・軌道相互作用を含めた計算例です。
- VBz.dat  
バナジウム・ベンゼン無限鎖に対するワニエ関数を生成します。これは、参考文献 [77] で研究されたものです。

## 44.5 出力ファイル

MLWF の計算で生成される 4 つのファイルを以下に説明します。拡張子「.mmn」のファイルには重なり行列要素  $M_{mn}^{(k,b)}$  が保存されています。拡張子「.amn」のファイルには初期推定のための射影行列要素  $A_{mn}^{(k)}$  が保存されています。拡張子「.eigen」のファイルには各  $k$  点における固有エネルギーと固有状態が保存されています。拡張子「.HWR」のファイルにはサンプリングした  $k$  グリッドと共役な Wigner-Seitz スーパーセル内にある、一組の格子ベクトル上での MLWF 間のホッピング積分が保存されています。最適化計算を再スタートする際には、拡張子「.mmn」のファイルが読み込まれます。4 つのファイルのより詳しい情報を以下に説明します。

### A. ファイル「.mmn」の形式

このファイルの構造は、Wannier90 [145] のファイル構造に厳密に従っています。このファイルの最初の行は、第 2 行の数値についての説明です。第 2 行の数は、左から右に、それぞれ外エネルギー窓に含まれるバンド数 ( $N_{win}$ )、 $k$  点の数、 $b$  ベクトルの数、スピン成分の数です。次に続く各行は  $M_{mn}^{(k,b)}$  のデータブロックで、ループ形式で書き出されています。最も外側のループはスピン成分のループ、次は  $k$  点のループ、そして  $b$  ベクトルのループです。また最も内側のループは、それぞれバンド・インデックスの  $n$  および  $m$  です。各ブロックにおいて、最初の行は 5 つの数から構成されます。最初の 2 つの数は、現在の  $k$  点インデックスおよび隣接する  $k+b$  のインデックスです。次の 3 つの数は、単位セル  $k+b$  点の座標です。ブロックの 2 行目からは、各行列要素の実部と虚部です。各ブロックにおいては、 $N_{win} \times N_{win}$  個の複素数があります。例として、入力ファイル「Si.dat」を用いて生成された「.mmn」ファイルを以下に示します。

```
Mmn_zero(k,b). band_num, kpt_num, bvector num, spinsize
          10          512          8          1
1  512    0    0    0
0.571090282808  -0.819911068319
0.000031357498  -0.000045367307
-0.000149292597  0.000215591228
-0.003821911756  0.005522040495
0.028616452988  0.019804944108
```

```

0.003677357735    0.002544970842
-0.006610037555   -0.004574771451
-0.000950861169   -0.000658076633
-0.000000008855   0.000000005272
.....
.....
...

```

## B. ファイル「.amn」の形式

このファイルの構造は、Wannier90 [145] のファイル構造に厳密に従っています。ファイルの第1行はファイル全体の記述です。第2行の4つの数は、それぞれ外エネルギー窓内のバンドの数 ( $N_{win}$ )、k点の数、生成する MLWF の数、スピン成分の数です。同様に、データブロックはループ形式で書き出されています。最も外側のループはスピン成分で、その次が k 点、生成する MLWF、バンドの数です。このファイルの最初の行に示されているように、各ブロックにおいて、最初の3つの整数は、それぞれバンド・インデックス、MLWF インデックス、および k 点インデックスです。その後、行列要素の実部と虚部 (Hartree) が出力されています。例として、入力ファイル「Si.dat」を用いて生成された「.amn」ファイルを以下に示します。

```

Amn. Fist line BANDNUM, KPTNUM, WANNUM, spinsize. Next is m n k...
      10      512      8      1
1    1    1    0.053943539299    0.000161703961
2    1    1   -0.000525446164   -0.000000008885
3    1    1    0.002498021589    0.000000084311
... ..
... ..
10   1    1   -0.000000023582   -0.000000000069
1    2    1    0.053943534952    0.000161703965
2    2    1    0.033382665372    0.000000493665
3    2    1   -0.051189536188   -0.000001480360
.....
.....

```

## C. ファイル「.eigen」の形式

このファイルは、各 k 点での固有エネルギーと固有状態を保存しています。最初の行は、系のフェルミ準位です。二行目にはバンドの数が示されています。次のデータは、主に二つの部分から成ります。最初の部分は固有エネルギーで、次の部分は固有状態に対応しています。各部分において、最も外側のループはスピン成分です。次のループは k 点で、その後にはバンド・インデックス (指数、添字) が続きます。固有状態については、基底系に関して一つ以上の内部ループがあります。例として、入力ファイル「Si.dat」により生成されたファイルを以下に示します。

```

Fermi level -0.112747
Number of bands 10
1    1    -0.566228100179
2    1    -0.122518136808
3    1    -0.122518129040

```

```

4    1   -0.122518115949
5    1   -0.026598417854
... ..

```

```
WF kpt 1 (0.00000000,0.00000000,0.00000000)
```

```

1 1   0.4790338281  -0.0014359768
1 2   0.0440709749  -0.0001321095
1 3  -0.0000003333  -0.0000000000
.....
.....

```

#### D. ファイル「.HWR」の形式

このファイルには、中心 ( $\mathbf{R} = 0$ ) の単位セル内の  $m$  番目の MLWF である  $|m, 0\rangle$  と、 $\mathbf{R}$  の単位セル内の  $n$  番目の MLWF である  $|n, \mathbf{R}\rangle$  の間のホッピング積分が保存されています。次のルールに従い、行列要素  $\langle m, 0 | \hat{H} | n, \mathbf{R} \rangle$  は、書き出されます。最初の行は説明のみです。2 行目と 3 行目には、MLWF の数と Wigner-Seitz のスーパーセル内の格子ベクトルの数がそれぞれ書かれています。5、6、7 行目には単位セルベクトルが書かれています。8 行目には、コリニア計算かノンコリニア計算、またスピン分極の取扱いに関して説明されています。9 行目はフェルミ準位 (Hartree) です。10 行目からデータブロックが始まり、ループ形式で書き出されています。最も外側のループはスピン成分です。次のループは  $\mathbf{R}$  について、最後の 2 つは  $m$  と  $n$  のループです。各  $\mathbf{R}$  は各ブロックの第 1 行目に縮退数とともに記載されています。各行においては、 $m$  と  $n$  のインデックスが表示され、その後にホッピング積分の実部と虚部 (Hartree) が続きます。例として、入力ファイル「Si.dat」により生成されたファイルを以下に示します。

```
Real-space Hamiltonian in Wannier Gauge on Wigner-Seitz supercell.
```

```
Number of Wannier Function 8
```

```
Number of Wigner-Seitz supercell 617
```

```
Lattice vector (in Bohr)
```

```

5.10000   0.00000   5.10000
0.00000   5.10000   5.10000
5.10000   5.10000   0.00000

```

```
collinear calculation spinsize 1
```

```
Fermi level -0.112747
```

```

R (  -6   2   2 )   4
 1    1   -0.000078903162  -0.0000000003750
 1    2    0.000024237763  -0.000000000148
 1    3    0.000024237691  -0.000000000341
 1    4    0.000024238375   0.0000000004117
 1    5    0.000072656918  -0.000000000196
 1    6   -0.000022470544  -0.000000000859
 1    7   -0.000022481557   0.000000000750
 1    8   -0.000022492706   0.000000000148
 2    1    0.000024238091   0.000000000049
 2    2   -0.000078901874  -0.000000000011
 2    3    0.000024234912  -0.000000000023

```

.....  
.....  
...

#### 44.6 MLWF の自動実行テスト

MLWF 計算に関連する機能が適切にインストールされていることを確認するために、MLWF 計算の自動実行テストが次のように実行できます。

##### MPI 並列計算の場合

```
% mpirun -np 16 openmx -runtestWF
```

##### MPI/OpenMP 並列計算の場合

```
% mpirun -np 8 openmx -runtestWF -nt 2
```

このテスト計算において、OpenMX は 8 個のテストケースを実行し、計算結果を「work/wf\_example」に保存されている参照結果と比較します。比較結果（スプレッド関数と  $\Omega$  関数それぞれにおける絶対差異）が、「work」ディレクトリ中のファイル「runtestWF.result」に保存されます。参照結果は、Xeon のクラスターマシンを用いて計算されたデータです。絶対差異が小数点以下 7 桁以内であれば、インストールが正常であると判断されます。

## 45 Wannier90 とのインターフェース

OpenMX の計算結果を Wannier90 [145] を用いてさらに解析することが出来ます。Wannier90 [145] を用いて最局在ワニエ関数が計算できます。その他にワニエ関数に射影した DOS やバンド構造、フェルミ面、ベリー位相に関連する物性（異常ホール伝導度や光伝導率）などの物性が計算可能です。OpenMX の計算結果を Wannier90 に読み込ませるためには、OpenMX の入力ファイル中に以下の 2 つのキーワードを設定します。

```
Wannier.Func.Calc      on    # on|off, default=off
Wannier90.fileout     on    # on|off, default=off
```

このように設定した入力ファイルを用いて OpenMX を実行すると計算終了時に以下のメッセージが表示されます。

The input files for Wannier90,

```
System.Name.amn
System.Name.mmn
System.Name.eig
System.Name.win
```

are successfully generated.

計算終了後に、上記 4 つのファイルが生成されます。3 つのファイルは wannier90 の実行形式ファイル「wannier90.x」にて読み込まれるものであり、最後の「*System.Name.win*」は「wannier90.x」の計算を制御する入力ファイルです。図 48 に計算の模式図を示しました。重なり行列「*System.Name.mmn*」、

## Interface with Wannier90

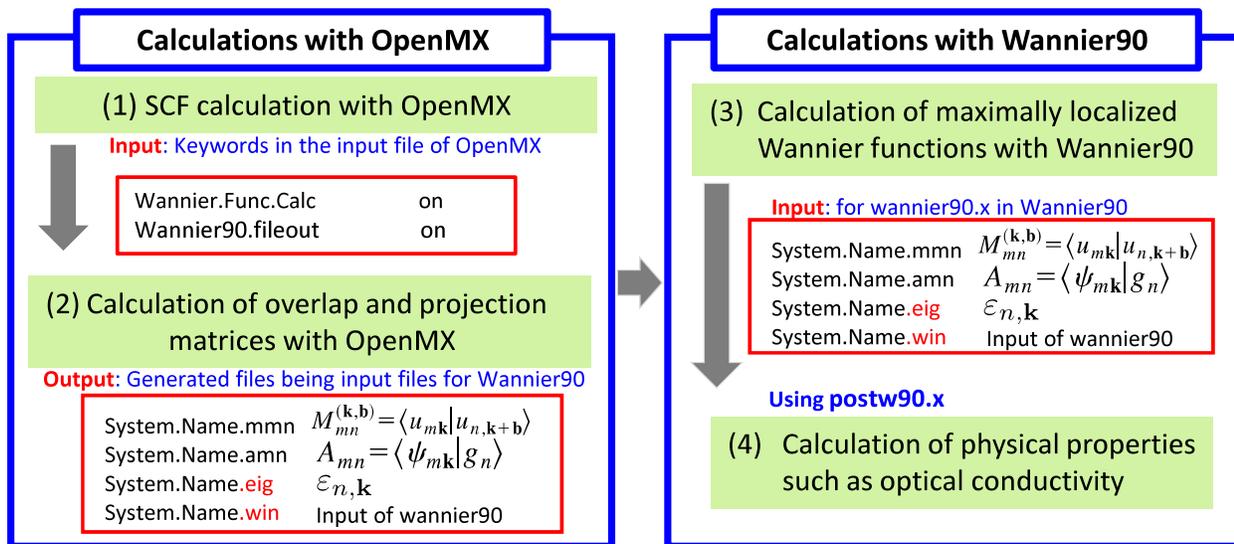


Figure 48: OpenMX と Wannier90 を用いた模式的な計算フロー

射影行列「*System.Name.amn*」と固有値「*System.Name.eig*」を用いて、「wannier90.x」では最局在ワニエ関数を計算します。最局在ワニエ関数が得られると、さまざまな物理量の計算が可能となります。「*System.Name.win*」には、デフォルトでは光伝導度を計算するが設定されています。

```
berry_task kubo
```

キーワード「*berry\_task*」を適切に変更することで、その他の物理量の計算が可能です。いくつかのオプション例を以下に示します。

```
berry_task kubo # optical conductivity
berry_task ahc # anomalous Hall conductivity
```

Wannier90 の詳しい解説は、Wannier90 [145] のウェブサイトをご参照下さい。

図 49 及び図 50 には、例として、ダイヤモンド構造をもつシリコンのゼーベック係数と、 $\text{SrVO}_3$  の光伝導度を示しました。入力ファイル「*Si-Wannier90.dat*」及び「*SrVO3-Wannier90.dat*」は「work」ディレクトリに収められています。

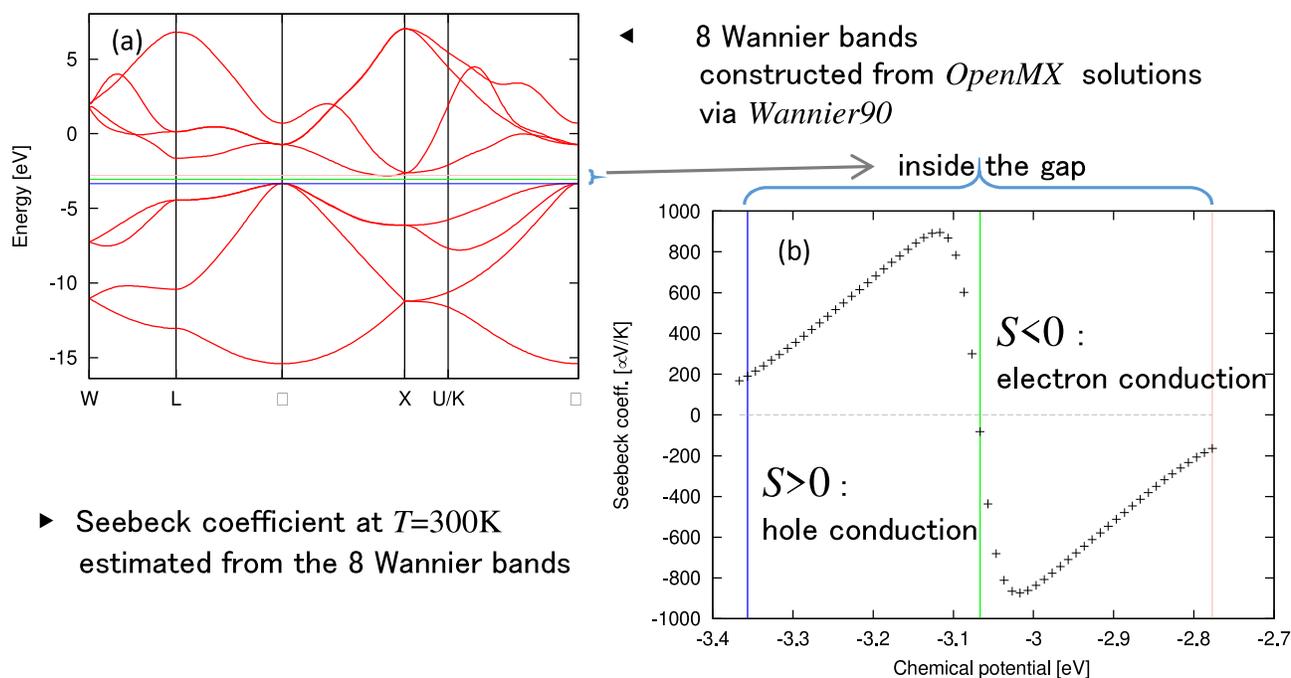


Figure 49: OpenMX と Wannier90 を用いて得られた 8 個のワニエ軌道によるダイヤモンド構造のシリコンの (a) バンド構造と (b)  $T=300\text{K}$  におけるゼーベック係数 (入力ファイル:「work」ディレクトリ内の「*Si-Wannier90.dat*」)。

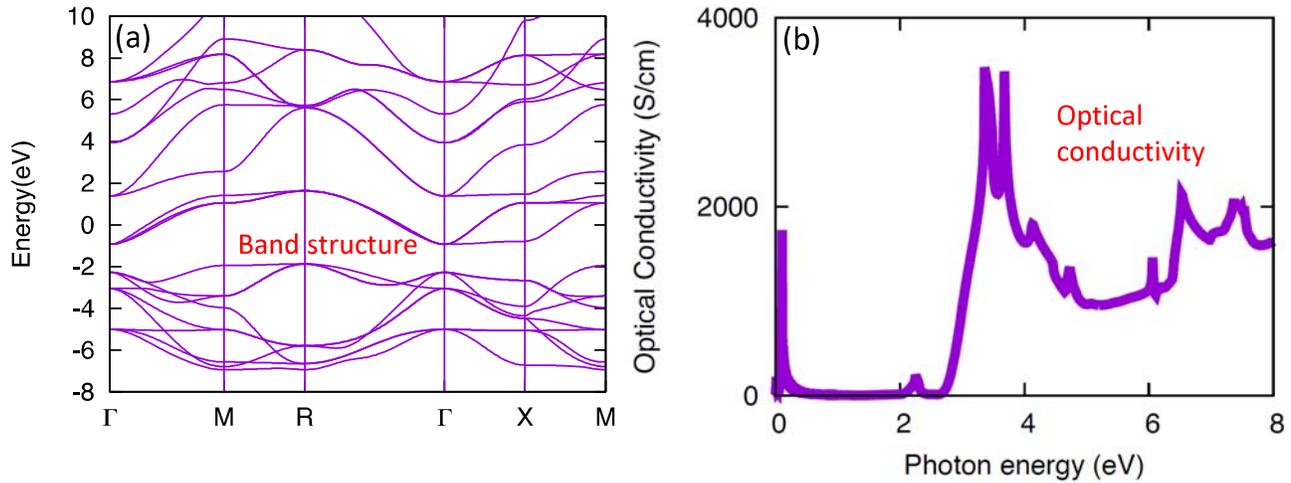


Figure 50: OpenMX と Wannier90 により得られた SrVO<sub>3</sub> の (a) バンド構造及び (b) 光伝導度。得られた光伝導度は過去の実験データ [147] と良く一致している。用いた入力ファイルは「work」ディレクトリ内の「SrVO3-Wannier90.dat」である。

## 46 対角化のための数値的に厳密な低次スケーリング法

大規模計算のために数値的に厳密な低次スケーリング法がサポートされています [124]。この方法の計算量は、 $N$  を基底関数の数とすると、1、2、3 次元の系に対してそれぞれ  $O(N(\log N)^2)$ 、 $O(N^2)$ 、 $O(N^{7/3})$  として増加します。 $O(N)$  法と異なり、本方法は低次のスケーリングにも関わらず通常の  $O(N^3)$  対角化法と同じく数値的に厳密な方法であり、一つの理論的な枠組で絶縁体から金属にも適用可能です。計算のデータ構造が適切に区分けされているため、図 51 に示すように大規模な並列処理に適しています。しかし計算量の前因子が大きくなるため、並列計算において多数の CPU コアを使用した場合にのみに、この方法が有利になります。多数の CPU コアを使って低次元の大規模系を計算する場合、この方法は適切な選択となるでしょう。この方法を選択するには、キーワード「scf.EigenvalueSolver」を以下の様に指定します。

```
scf.EigenvalueSolver    cluster2
```

本手法は、クラスター計算あるいはブリルアンゾーンでのサンプリングに対して  $\Gamma$  点だけが考慮された周期系のコリニア DFT 計算に対してのみサポートされています。全エネルギー計算ばかりでなく、力の計算も実装されていますので、幾何学構造の最適化を行うことができます。しかし状態密度計算や波動関数の計算は実装されていません。周回積分の極の数 [74] は次のキーワードで設定します。

Table 10: 数値的に厳密な低次スケーリング法と通常対角法の計算時間 (秒)。どちらの場合も、8 MPI プロセスを用いて C<sub>60</sub> 分子の全エネルギーを計算。入力ファイルは「work」ディレクトリの「C60\_LO.dat」。

| Method       | Total energy (Hartree) | Computational time (sec.) |
|--------------|------------------------|---------------------------|
| Low-order    | -343.896238929370      | 69.759                    |
| Conventional | -343.896238929326      | 2.784                     |

収束に必要な極の数は系の大きさに依らずに、系のスペクトル半径に依存しています [124]。電子温度が 300K 以上の場合には、100 の極を用いれば全エネルギーと力は十分に収束します。例として、「work」ディレクトリ中の入力ファイル「C60\_LO.dat」を用いた計算を示します。

```
% mpirun -np 8 openmx C60_LO.dat
```

表 10 に示すように、本法で計算された全エネルギーは、倍精度の範囲内で通常の対角化法とほぼ一致していますが、一方、計算時間に関しては、従来法と比較し、非常に長い時間を要しています。計算時間に関し、本法と通常の対角化法の交点は、系の次元にも依りますが並列処理に 100 個以上のコアを使う場合、原子数で 300 程度であると推測されます。

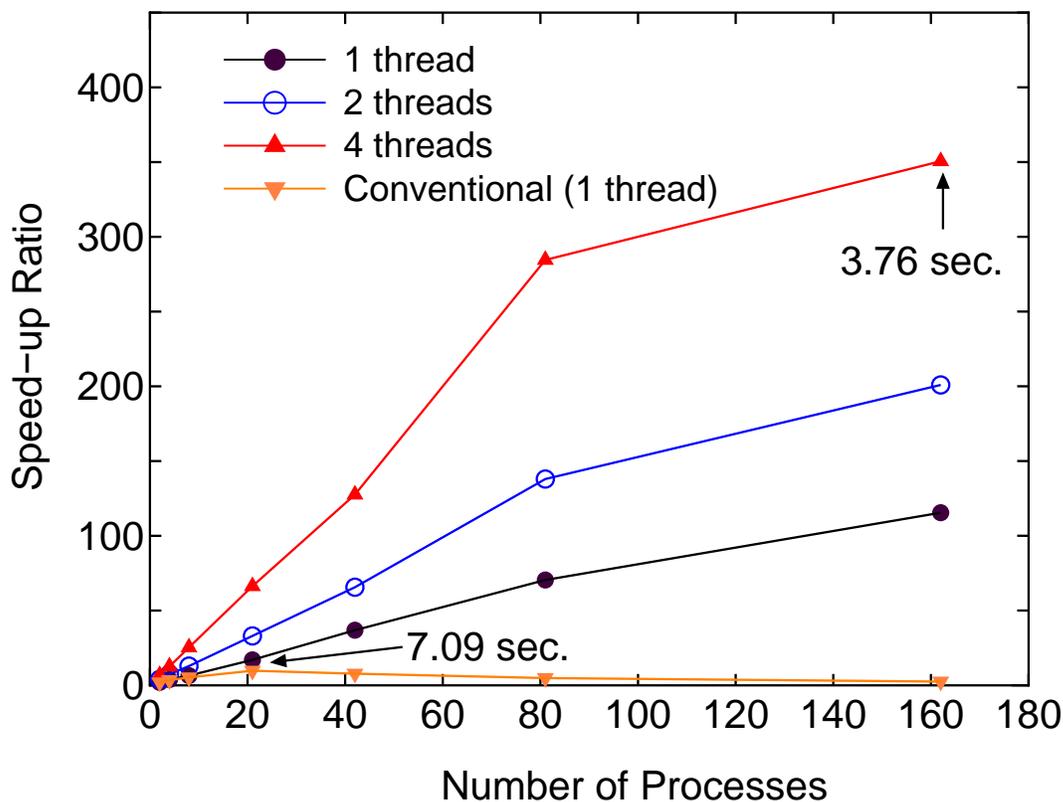


Figure 51: DNA の対角化 (SCF 計算) の OpenMP/MPI ハイブリッド並列計算の速度向上比。対角化には数値的に厳密な低次スケーリング法を使用。速度向上比は  $2T_2/T_p$  として定義。  $T_2$  は 2 MPI プロセスによる所要時間、  $T_p$  は並列計算の所要時間。これらの並列計算は AMD の Opteron クアッドコア・プロセッサ (2.3 GHz) からなる CRAY-XT5 マシン上で実施。電子温度を 700K とし、周回積分のための極の数は 80。比較のために、Householder 法と QR 法による通常の対角化法の並列計算の速度向上比も示す (シングルスレッドの場合)。また低次スケーリング法および通常法に対して、矢印で示した点での所要時間も示す。

## 47 有効遮蔽媒質法 (ESM 法)

### 47.1 概要

有効遮蔽媒質法 (ESM 法) は、電荷を有するスラブモデルあるいは電場印加条件下にあるスラブモデルについての第一原理計算手法です [125, 126, 127, 128]。本手法では、スラブの表面並行方向には 2 次元の周期境界条件を保ちつつ、垂直方向には 1 次元の境界条件を与え (図 52(a))、その 1 次元境界条件の下での Poisson 方程式をグリーン関数法によって解きます。実際には次のような半無限媒質を境界条件として導入することによって、孤立スラブ系、荷電スラブ系、および一様電場下にあるスラブ系を扱います。

- (a) 孤立スラブ系：真空 (比誘電率  $\epsilon = 1$ ) + 真空
- (b) 荷電スラブ系：真空 + 理想金属 (比誘電率  $\epsilon = \infty$ )
- (c) 一様電場下スラブ：理想金属 + 理想金属

ここで、「スラブ」とは、通常の表面モデルとして使われるスラブばかりでなく、二次元的に配置された分子群から構成される系も含まれます。孤立スラブモデルは分極した系の解析に、また荷電スラブモデルは電極表面のシミュレーションに適用できます。二枚の理想金属媒質の間に挟まれたままで電場が印加されたスラブモデルは、金属コンデンサ内にある物質のモデル化に該当します。OpenMX において、ESM 法の計算で使用する単位セルは次のように構築します (図 52 (a) 参照)。

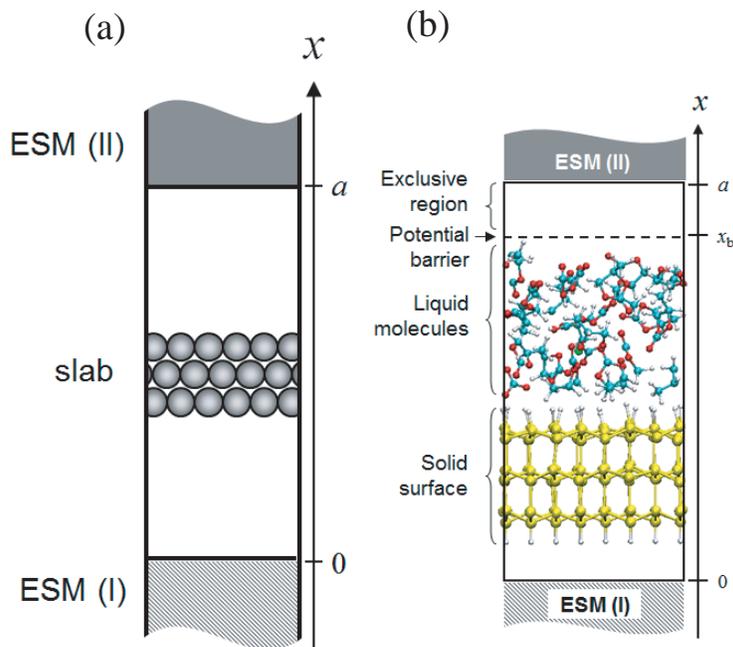


Figure 52: (a) 半無限媒質 (ESM) を伴うスラブの概略図。ESM(I) と ESM(II) はそれぞれ、 $x = 0$  と  $x = a$  ( $a$  は  $x$  軸方向のセルの長さ) におけるセル境界。(b) ESM 法を用いた固液界面モデル系の MD 計算の単位セルの例。スラブと ESM は  $y$ - $z$  面に平行に配置。

1. セルの a 軸は b-c 平面に対して垂直で x 軸に対して平行
2. 2 つの周期境界条件は y および z 軸方向に設定
3. ESM はセル境界 ( $x = 0, a$ ) に配置
4. x 軸の原点はセル境界に設定
5. x 軸の分率座標は、0 と 1 の間で指定

ESM 法に基づく計算は次のキーワードによって実行します。

```
ESM.switch          on3      # off, on1=v|v|v, on2=m|v|m, on3=v|v|m, on4=on2+EF
ESM.buffer.range    4.5      # default=10.0 (ang),
```

ここで、on1、on2、on3、on4 は ESM の組み合わせ、すなわちそれぞれ「真空+真空」、「理想金属+理想金属」、「真空+理想金属」、「理想金属+電場中の理想金属」を表しています。キーワード「ESM.buffer.range」はモデル内原子に対する排他的領域の幅 (Å 単位) を指定しており、これは波動関数と ESM の重なりを回避するために指定します。

1. ESM.switch = on1:

ESM(I) および ESM(II) は半無限真空です。この条件は、計算する系の全電荷が中性である場合にのみ適用されます。従って、キーワード「scf.system.charge」は必ずゼロに設定します。

2. ESM.switch = on2:

ESM(I) および ESM(II) は半無限の理想金属です。電荷を持つ系を取り扱うことができます。キーワード「scf.system.charge」は有限な値に設定します。

3. ESM.switch = on3:

ESM(I) および ESM(II) は、それぞれ、半有限の真空および理想金属です。電荷を持った系を取り扱うことができます。キーワード「scf.system.charge」は有限な値に設定します。

4. ESM.switch = on4:

ここでは on2 と同じ ESM の組み合わせが適用されており、系に電場が印加されます。次のキーワードを使うことで計算する系に一様電場を印加することができます。

```
ESM.potential.diff    1.0      # default=0.0 (eV),
```

ここで、下部の理想金属を基準に 2 枚の半無限理想金属媒質間の電位差 (eV 単位) を指定できます。電場はセル長 a と電位差によって決まります。

5. ESM 法を使った MD 計算

ESM 法を適用しながら固液界面系の MD 計算を行うことができます。表面モデルスラブと液体領域は図 52 (b) のように配置します。与えられた領域内に分子を制限するため、次のキーワードを使って障壁ポテンシャルを導入します (図 52 (b) 参照)。

```

ESM.wall.position      6.0      # default=10.0 (ang)
ESM.wall.height       100.0     # default=100.0 (eV),

```

ここで、「ESM.wall.position」はセル上端と障壁ポテンシャルの原点の間の距離  $a - x_b$ 、また「ESM.wall.height」は  $x = x_b + 1.0$  (Å) でのポテンシャルの高さ (ポテンシャルエネルギーの値) です。尚、MD を実行する間に表面モデルスラブの移動を防ぐために、そのスラブの最下層原子の座標を固定することをお勧めします。

## 6. ESM 法で扱われる軸の選択

図 52(a) に示したように、マニュアルでは ESM 法による扱いは  $x$  軸に適應されます。しかしながら、ESM 法で扱われる軸の選択はキーワード「ESM.direction」で以下のように変更できます。

```

ESM.direction          x          # x|y|z, default=x

```

既定の方向は図 52(a) のとおり  $x$  軸です。

## 47.2 テスト計算例

デモ計算として、図 53 (a) に「真空+理想金属」(ESM.switch=on3) の境界条件の下で、 $1 \times 1$  のアルミニウム終端 Si(111) スラブ中の余剰電荷  $\rho_{ex}$  分布を示します (このテスト計算の入力ファイル「AlSi111\_ESM.dat」は work ディレクトリにあります)。ドーブ電荷とそれに対応して発生する鏡像電荷との間の引力的相互作用により、スラブ内にあるドーブ電荷が理想金属側の表面に偏析している様子がわかります。図 53 (b) は、図 53 (a) に示した各条件での Hartree ポテンシャル変化  $\Delta V_H$  を示しています。ここで、Al-Si(111) スラブ内部のポテンシャルと、スラブと理想金属間の電場はドーブした電荷の量に従って変化します。

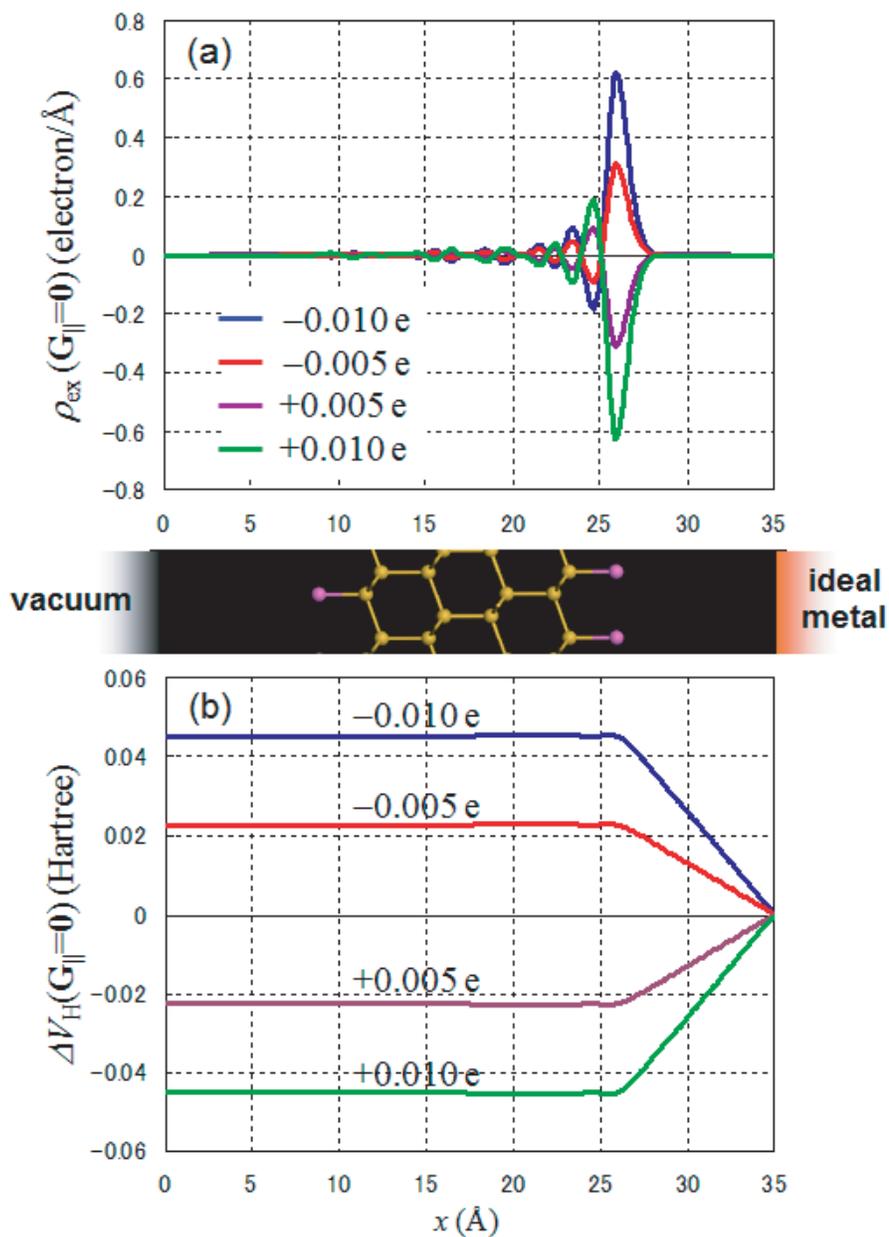


Figure 53: 真空+理想金属の条件における Si(111) スラブモデル。(a) Si(111) スラブ中の過剰電荷  $\rho_{\text{ex}}$  の分布。(b) Si(111) スラブのバイアスによる Hartree ポテンシャルの変化  $\Delta V_{\text{H}}$ 。ドーピングした電荷の量は  $-0.01$ 、 $-0.005$ 、 $+0.005$ 、 $+0.01 e$ 。各プロットは同じ ESM による中性のスラブを基準とした電荷または Hartree ポテンシャルの差の変化。

## 48 仕事関数の計算

`System.Name.v0.cube` または `System.Name.v1.cube` のどちらかを用いて、金属の仕事関数を計算できます。仕事関数は次式で定義されます。

$$\begin{aligned}\Phi &= (\phi_\infty + E[N - 1]) - E[N], \\ &= \phi_\infty - \mu,\end{aligned}\tag{11}$$

ここで  $E[N]$  と  $E[N - 1]$  はそれぞれ  $N$  電子系と  $(N - 1)$  電子系の全エネルギーです。 $\phi_\infty$  と  $\mu$  はそれぞれ表面からの無限遠でのポテンシャルと化学ポテンシャルです。式 (11) の第二行は Janak の定理 [89] から  $E[N - 1] - E[N] = \int dn \partial E / \partial n = -\mu$  として得られます。ここで、 $n$  はフェルミ面での一粒子固有状態の占有数です。 $dn = -ds/S$  はフェルミ面の面積  $S$  と無限小面積  $ds$  で定義され、表面積分はフェルミ面上で実施されます。仕事関数は表面に関連した量であるため、図 54(a) に示したスラブモデルを導入する必要があります。例として、アルミニウムスラブを考えます。ここで層の厚さは 5 であり、約 60 Å の真空が  $x$  軸に沿って章 47 での有効遮蔽媒質 (ESM) 法と共に考慮されます。ここで周期的スラブ間の相互作用を回避するため「ESM.switch=on1」が使われます。入力ファイル「Al111\_WorkFunc\_0E.dat」を用いて以下のとおり SCF 計算が実行できます。

```
% mpirun -np 28 ./openmx Al111_WorkFunc_0E.dat
```

計算が正常に完了すると、cube ファイル「Al111\_WorkFunc\_0E.v0.cube」が得られます。cube ファイルを解析するため、ディレクトリ「source」中の後処理コード「gcube2oned.c」が利用できます。このコードは以下のようにコンパイルできます。

```
% gcc gcube2oned.c -lm -o gcube2oned
```

実行コード「gcube2oned」を作業ディレクトリにコピーして下さい。「gcube2oned」を用いて三次元の cube データを選択した方向の一次元のデータに変換できます。以下のとおり、1 は  $a$  軸に相当し、残りの二次元の  $bc$  面上での積分が実行されます。

```
% gcube2oned Al111_WorkFunc_0E.v0.cube 1 > 1d_pot.txt
```

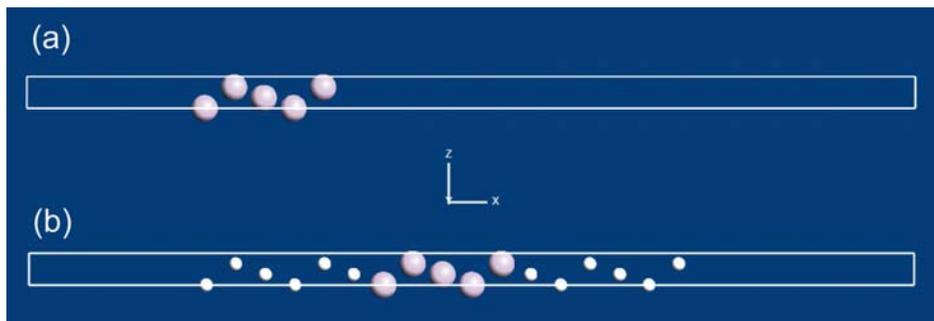


Figure 54: (a) Al 原子で構成されたスラブモデル。層の厚さは 5、また 60 Å の真空が挿入されている。(b) Al 原子で構成されたスラブモデル。層の厚さは 5、また 60 Å の真空が挿入され、さらに Al 層に加えて、真空領域の波動関数の裾が正確に記述できるように空原子から構成される 6 層が両表面に導入されている。

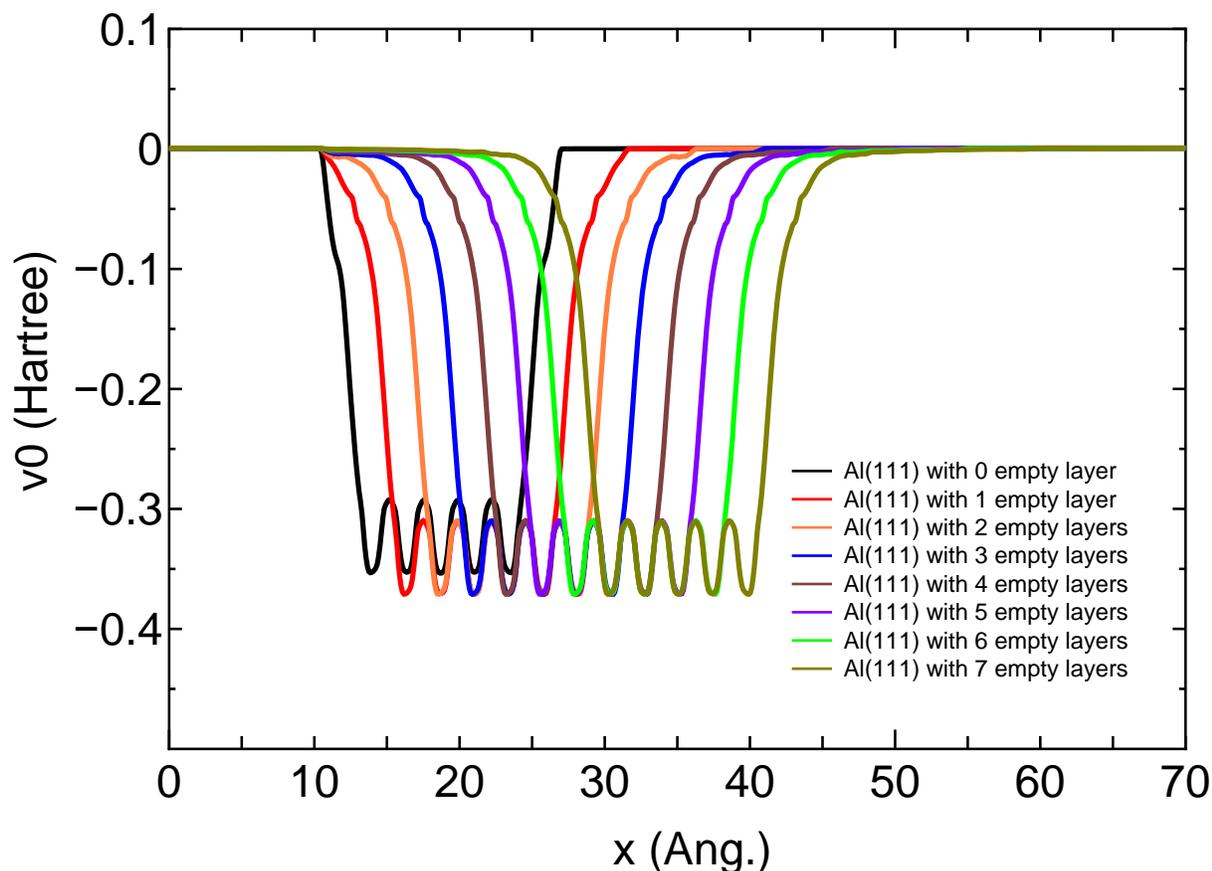


Figure 55: bc 面上で平均された a 軸に沿った一次元ポテンシャル (Hartree)。v0.cube ファイルからの取得。空原子を構成する層の数は両表面に対して 0 から 7 まで系統的に変化。計算に用いられた入力ファイルは Al111\_WorkFunc\_%E.dat、% は 0 から 7 まで。これらはすべてディレクトリ work で利用可能。

得られたファイル 1d\_pot.txt の第一、第二、第三列は格子点のシリアル番号、a 軸に沿った位置、そして bc 面上で平均された一次元ポテンシャル (Hartree) に対応します。b 軸または c 軸に沿った一次元ポテンシャルの場合は gcube2oned の引数に 2 または 3 を与えます。図 55 は Al(111) 面の表面に垂直な a 軸に沿った一次元ポテンシャルを示します。空原子を構成する層の数は両表面に対して 0 から 7 まで系統的に変化させました。図 54 に示したように真空領域に向かっての波動関数の裾が正確に記述できるように空原子から構成された層が考慮されます。計算に用いられた入力ファイルは「Al111\_WorkFunc\_%E.dat」であり、ここで % は 0 から 7 まで変化し、その値は空原子から構成された層数に対応します。これらはすべてディレクトリ「work」中で利用可能です。式 (11) のポテンシャル  $\phi_\infty$  は 70 Å 周辺でのポテンシャルから得られます。他方、化学ポテンシャル  $\mu$  は出力ファイルから得られます。70 Å 周辺でのポテンシャルはほとんどゼロのため、この場合は仕事関数は基本的に化学ポテンシャルから決定されます。図 56 は式 (11) を用いて計算された仕事関数を空原子から構成される空層数の関数としてプロットしたものです。Al(111) 表面の仕事関数が二つの空層で収束に至ることが分かります。これは収束結果を得るには二つの空層を加えることで十分であることを示唆します。

表 11 には五つの金属の仕事関数の計算結果と対応する実験値を示します。すべての計算で 7 つの層を導入しました。計算値と実験値の良い一致が見られます。計算に用いた入力ファイルは「Al111\_WorkFunc\_7E.dat」, 「Cu111\_WorkFunc\_7E.dat」, 「Ag111\_WorkFunc\_7E.dat」, 「Au111\_WorkFunc\_7E.dat」,

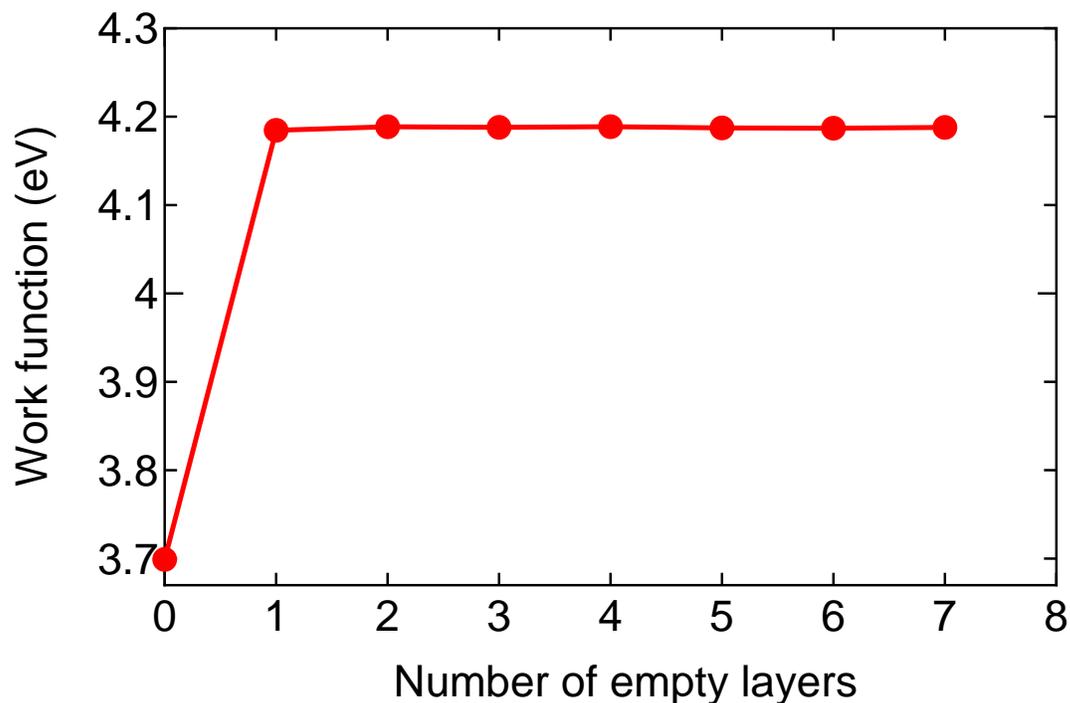


Figure 56: 空層数の関数としての Al(111) 表面の仕事関数 (eV)。数値は式 (11) に基づいて取得。計算は図 55 に掲載。

「Pd111\_WorkFunc\_7E.dat」で、これらは全てディレクトリ「work」中に保存されています。

ギャップのある系では式 (11) は厳密には有効ではありません。しかしながら、式 (11) は近似的な扱いとして使用できます。特に分極のある表面でのギャップのある系では、周期的スラブ間の相互作用を回避するため 47 章で説明された ESM 法を用いる必要があります。

Table 11: 五つの金属の仕事関数の計算結果と対応する実験値。報告された実験値は文献ごとにややばらつきがあることに注意。表にはその一つを掲載。

|         | OpenMX | Expt.           |
|---------|--------|-----------------|
| Al(111) | 4.19   | 4.26±0.03 [129] |
| Cu(111) | 4.74   | 4.94±0.03 [130] |
| Ag(111) | 4.51   | 4.46±0.02 [131] |
| Au(111) | 5.33   | 5.26±0.04 [132] |
| Pd(111) | 5.40   | 5.55±0.01 [133] |

## 49 NEB (Nudged elastic band) 法

### 49.1 概要

幾何学位相空間内の二つの安定構造をつなぐ最小エネルギー経路 (MEP) を調べるために、OpenMX Ver. 3.9 では参考文献 [134] に基づく NEB 法がサポートされています。実装の詳細を以下にまとめます。

- 参考文献 [134] の (8)-(11) 式に基づく正接の計算
- 参考文献 [134] の (4) 式に基づく垂直力の計算
- 参考文献 [134] の (12) 式に基づく平行力の計算
- DIIS 法+BFGS 法によるハイブリッド最適化法

NEB の機能は構造最適化とほぼ同様な手続で利用可能です。そのため、比較的容易に本機能を使いこなすことができるでしょう。その特徴を以下に列挙します。

- 構造最適化とほぼ同様な手続で使用可能
- ハイブリッド MPI/OpenMP 並列化
- 直線初期経路またはユーザー定義による初期経路
- 3 つのルーチンのみを付加

### 49.2 実行方法

NBE の計算は次の三つのステップで実行されます。

1. 反応物の構造最適化
2. 生成物の構造最適化
3. 反応物と生成物をつなぐエネルギー経路の最適化

数値誤差が相殺するように、3 つの計算において同一の計算パラメータを使用して下さい。ここで計算パラメータとは単位格子セル、カットオフエネルギー、基底関数、擬ポテンシャル、電子温度等のことです。ステップ 1 およびステップ 2 の計算の後、「\*.dat#」ファイルが作成されます。ファイル「\*.dat#」中に記載されている原子座標を利用して、容易にステップ 3 に必要な入力ファイルを作成できます。ステップ 3 に対する入力ファイルが作成できれば、NEB の計算は、次のように通常の OpenMX 計算の場合と同様に実行できます。

```
% mpirun -np 32 openmx input.dat -nt 4
```

### 49.3 計算例と関連キーワード

計算例として二つの入力ファイルが用意されています。

- C2H4\_NEB.dat  
2個のエチレン分子の環状付加反応によるシクロブタンの生成
- Si8\_NEB.dat  
ダイヤモンド構造のSi中での格子間水素原子の拡散

以下では入力ファイル「C2H4\_NEB.dat」を用いたNEB計算を例に説明します。

反応物と生成物の構造を設定

反応物の原子座標を次のように入力ファイル中で指定します。

```
<Atoms.SpeciesAndCoordinates
  1  C  -0.66829065594143   0.00000000101783  -2.19961193219289   2.0   2.0
  2  C   0.66817412917689  -0.00000000316062  -2.19961215251205   2.0   2.0
  3  H   1.24159214112072  -0.92942544650857  -2.19953308980064   0.5   0.5
  4  H   1.24159212192367   0.92942544733979  -2.19953308820323   0.5   0.5
  5  H  -1.24165800644131  -0.92944748269232  -2.19953309891389   0.5   0.5
  6  H  -1.24165801380425   0.92944749402510  -2.19953309747076   0.5   0.5
  7  C  -0.66829065113509   0.00000000341499   2.19961191775648   2.0   2.0
  8  C   0.66817411530651  -0.00000000006073   2.19961215383949   2.0   2.0
  9  H   1.24159211310925  -0.92942539308841   2.19953308889301   0.5   0.5
 10  H   1.24159212332935   0.92942539212392   2.19953308816332   0.5   0.5
 11  H  -1.24165799549343  -0.92944744948986   2.19953310195071   0.5   0.5
 12  H  -1.24165801426648   0.92944744880542   2.19953310162389   0.5   0.5
Atoms.SpeciesAndCoordinates>
```

生成物の原子座標を次のように入力ファイル中で指定します。

```
<NEB.Atoms.SpeciesAndCoordinates
  1  C  -0.77755846408657  -0.00000003553856  -0.77730141035137   2.0   2.0
  2  C   0.77681707294741  -0.00000002413166  -0.77729608216595   2.0   2.0
  3  H   1.23451821718817  -0.88763832172374  -1.23464057728123   0.5   0.5
  4  H   1.23451823170776   0.88763828275851  -1.23464059022330   0.5   0.5
  5  H  -1.23506432458023  -0.88767426830774  -1.23470899088096   0.5   0.5
  6  H  -1.23506425800395   0.88767424658723  -1.23470896874564   0.5   0.5
  7  C  -0.77755854665393   0.00000000908006   0.77730136931056   2.0   2.0
  8  C   0.77681705017323  -0.00000000970885   0.77729611199476   2.0   2.0
  9  H   1.23451826851556  -0.88763828740000   1.23464060936812   0.5   0.5
 10  H   1.23451821324627   0.88763830875131   1.23464061208483   0.5   0.5
 11  H  -1.23506431230451  -0.88767430754577   1.23470894717613   0.5   0.5
 12  H  -1.23506433587007   0.88767428525317   1.23470902573029   0.5   0.5
NEB.Atoms.SpeciesAndCoordinates>
```

## NEB 計算のためのキーワード

NEB 計算は次のようにキーワード「MD.Type」を設定することによって実行します。

```
MD.Type          NEB
```

経路中のイメージ数を以下のキーワードで指定します。

```
MD.NEB.Number.Images      8          # default=10
```

ここで、2つの終端構造(反応物と生成物)はイメージの数から除きます。  
バネ定数は次のように与えます。

```
MD.NEB.Spring.Const      0.1          # default=0.1(hartree/bohr^2)
```

ほとんどの場合、得られる経路はこの値に大きく依存しません。

MEPの最適化はハイブリッド最適化法(DIIS+BFGS)によって行います。この最適化法は次のキーワードによって制御されます。

```
MD.Opt.DIIS.History      4          # default=7
MD.Opt.StartDIIS         10         # default=5
MD.maxIter                100         # default=1
MD.Opt.criterion         1.0e-4       # default=1.0e-4 (Hartree/Bohr)
```

これらのキーワードの仕様は、構造最適化に対するものと同様なので、詳細についてはマニュアルの「構造最適化」の章をご覧ください。またキーワード「MD.Fixed.XYZ」で原子の位置を固定することも可能です。

## NEB 計算の実行

以下の様に、入力ファイル「C2H4\_NEB.dat」を用いて NEB 計算を実行できます。

```
% mpirun np 16 openmx C2H4_NEB.dat
```

計算が正常に完了すると、24個以上のファイルが作成されます。これらの出力ファイルを以下に列挙します。

```
c2h4.neb.opt          history of optimization for finding MEP
c2h4.neb.ene          total energy of each image
c2h4.neb.xyz          atomic coordinates of each image in XYZ format
C2H4_NEB.dat#         input file for restarting.
C2H4_NBE.dat_0        input file for the precursor
C2H4_NBE.dat_1        input file for the image 1
C2H4_NBE.dat_2        input file for the image 2
C2H4_NBE.dat_3        input file for the image 3
```

```

C2H4_NBE.dat_4      input file for the image 4
C2H4_NBE.dat_5      input file for the image 5
C2H4_NBE.dat_6      input file for the image 6
C2H4_NBE.dat_7      input file for the image 7
C2H4_NBE.dat_8      input file for the image 8
C2H4_NBE.dat_9      input file for the product
c2h4_0.out           output file for the precursor
c2h4_1.out           output file for the image 1
c2h4_2.out           output file for the image 2
c2h4_3.out           output file for the image 3
c2h4_4.out           output file for the image 4
c2h4_5.out           output file for the image 5
c2h4_6.out           output file for the image 6
c2h4_7.out           output file for the image 7
c2h4_8.out           output file for the image 8
c2h4_9.out           output file for the product

```

「c2h4.neb.opt」には、図 57 (a) に示すように、MEP の最適化過程の履歴が保存されています。以下の「c2h4.neb.opt」に示されるように、このファイルの最初の部分に説明が記載されています。

```

*****
*****
History of optimization by the NEB method
*****
*****

```

| iter  | SD_scaling | Maximum force <br>(Hartree/Bohr) | Maximum step<br>(Ang) | Norm<br>(Hartree/Bohr) | Sum of Total Energy of Images<br>(Hartree) |
|-------|------------|----------------------------------|-----------------------|------------------------|--|
| 1     | 0.37794520 | 0.12552539                       | 0.04583072            | 0.49503563             | -223.77727271                              |
| 2     | 0.37794520 | 0.08684953                       | 0.03163814            | 0.35379139             | -223.85742175                              |
| 3     | 0.37794520 | 0.05494411                       | 0.01922344            | 0.25668987             | -223.89831309                              |
| 4     | 0.37794520 | 0.03790970                       | 0.01234783            | 0.20282699             | -223.92042217                              |
| 5     | 0.45353424 | 0.02936250                       | 0.01326992            | 0.17349184             | -223.93482686                              |
| 6     | 0.45353424 | 0.02588308                       | 0.01169327            | 0.15249816             | -223.94772371                              |
| 7     | 0.45353424 | 0.02303223                       | 0.01039732            | 0.13836350             | -223.95785384                              |
| ..... |            |                                  |                       |                        |  |
| ...   |            |                                  |                       |                        |  |
| .     |            |                                  |                       |                        |  |

また「c2h4.neb.ene」と「c2h4.neb.xyz」は、図 57 (b) に示すように反応物からの距離 (Bohr) の関数としての全エネルギーの変化および構造の変化が保存されていますので、MEP の分析に利用できます。「c2h4.neb.ene」の内容は次のようなものです。

```

#
# 1st column: index of images, where 0 and MD.NEB.Number.Images+1 are the terminals

```

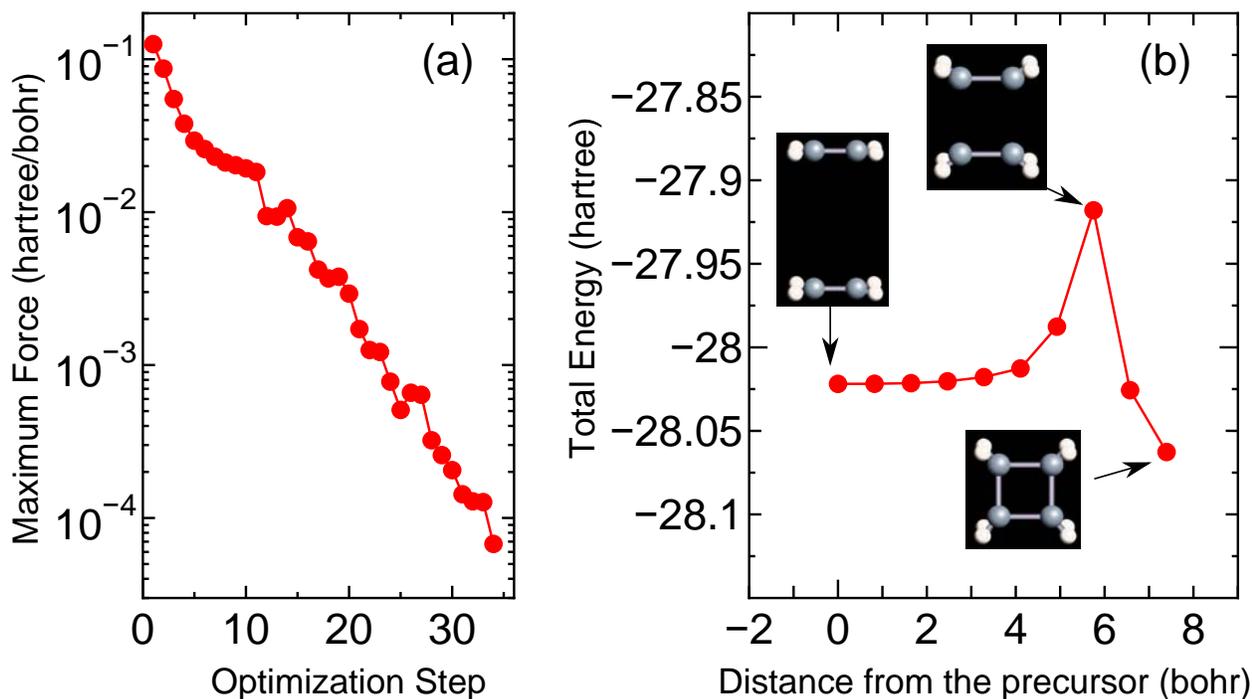


Figure 57: (a) 二個のエチレン分子の環状付加反応(シクロブタンの生成)のNEB計算の最適化履歴(c2h4.neb.opt)、(b) 反応物からの距離(Bohr)の関数としての2つのエチレン分子の全エネルギーの変化(c2h4.neb.ene)と、それに対応する最小エネルギー経路上のイメージの幾何構造(c2h4.neb.xyz)。本NEB計算の入力ファイルは「work」ディレクトリ内の「C2H4\_NEB.dat」。

```
# 2nd column: Total energy (Hartree) of each image
# 3rd column: distance (Bohr) between neighbors
# 4th column: distance (Bohr) from the image of the index 0
#
```

|   |              |            |            |            |
|---|--------------|------------|------------|------------|
| 0 | -28.02185123 | 0.00000000 | 0.00000000 | 1.33646479 |
| 1 | -28.02178507 | 0.82118927 | 0.82118927 | 1.33567761 |
| 2 | -28.02140083 | 0.82112464 | 1.64231391 | 1.33523542 |
| 3 | -28.02029258 | 0.82111520 | 2.46342911 | 1.33463918 |
| 4 | -28.01779519 | 0.82113225 | 3.28456136 | 1.33375873 |
| 5 | -28.01261498 | 0.82135735 | 4.10591871 | 1.33262670 |
| 6 | -27.98761576 | 0.82169347 | 4.92761218 | 1.34184319 |
| 7 | -27.91797754 | 0.82218705 | 5.74979923 | 1.51281867 |
| 8 | -28.02565242 | 0.82256542 | 6.57236464 | 1.55582513 |
| 9 | -28.06263668 | 0.82263897 | 7.39500361 | 1.55437554 |

ここで第1列はイメージの通し番号で、0と9はそれぞれ反応物と生成物に相当します。第2列はそれぞれのイメージの全エネルギーです。第3および第4列は、幾何学的位相空間内の2つの隣接イメージの間の距離(Bohr)と反応物からの距離(Bohr)です。それぞれのイメージの計算は、基本的に異なる入力ファイルによる独立したOpenMXの計算として実行されるため、「System.Name.dat\_#」ファイル

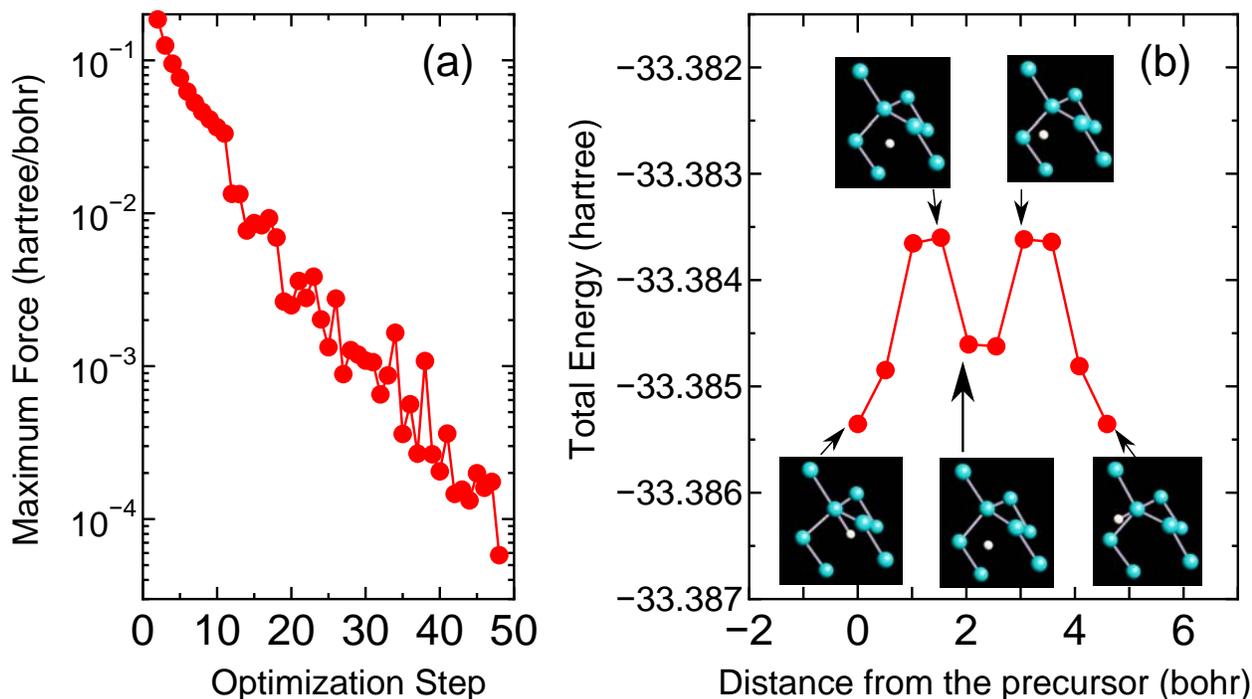


Figure 58: (a) ダイアモンド Si 中の格子間水素原子の拡散に対する NEB 計算の最適化履歴 (si8\_neb.neb.opt)、(b) 反応物からの距離 (Bohr) の関数としての全エネルギーの変化 (si8\_neb.neb.ene) と、それに対応する最小エネルギー経路上のイメージの幾何構造 (si8\_neb.neb.xyz)。本 NEB 計算の入力ファイルは「work」ディレクトリ内の「Si8\_NEB.dat」。

が自動生成されています。ここで「*System.Name*」は「*System.Name*」、「#」はそれぞれのイメージの通し番号です。対応する出力ファイル「*System.Name*\_*#*.out」も出力されますが、これは MEP 上で電子構造がどのように変化するのか、分析するために役立つでしょう。

「C2H4\_NEB.dat」と同様に、「Si8\_NEB.dat」でも NEB 計算を実行できます。計算が正常に終了すれば、図 58 に示す最適化の履歴と MEP に沿った全エネルギー変化が得られるでしょう。

#### 49.4 NEB 計算の再スタート

最適化ステップ数が最大値に達しても計算が収束しないことがしばしば発生します。この場合には、前のジョブの最後の最適化ステップから、新しいジョブとして最適化を続けて下さい。ファイル「*System.Name*.dat#」が最適化ステップ毎に生成されています。このファイルには、最終ステップでの全てのイメージの原子座標が設定されています。この入力ファイル「*System.Name*.dat#」を使って最適化計算を再スタートできます。

#### 49.5 ユーザー定義の初期経路

デフォルトで、反応物と生成物をつなぐ初期経路は、それらを結ぶ直線です。しかし、系によっては直線上に生成されたイメージの幾何学構造は原子間距離が近接してしまい、物理的には許容できないものになります。この場合、初期イメージの原子座標を明示的に与える必要があります。ユーザー自身で初期イメージの原子座標を明示的に与える方法は、実は再スタートの場合と同様です。どちらの場合でも次のキーワードによって、それぞれのイメージの原子座標を明示的に与えます。

```

<NEB1.Atoms.SpeciesAndCoordinates
  1 Si -0.12960866043083  0.13490502997627 -0.12924862991035  2.0  2.0
  2 Si -0.40252421446808  5.19664433048606  4.91248322056082  2.0  2.0
  ...
NEB1.Atoms.SpeciesAndCoordinates>

<NEB2.Atoms.SpeciesAndCoordinates
  1 Si -0.08436294149342 -0.02173837971883 -0.08374099211565  2.0  2.0
  2 Si -0.33677725120015  5.10216241168093  5.01087499461541  2.0  2.0
  ...
NEB2.Atoms.SpeciesAndCoordinates>

```

「MD.NEB.Number.Images」で指定した全てのイメージに対して、原子座標を与える必要があります。さらに、キーワードを「on」に設定して下さい。

```
scf.restart      on
```

## 49.6 NEB 計算における SCF の確認

NEB 計算での標準出力には、イメージ 1 の情報のみが表示されます。他のイメージについての情報は表示されません。SCF の反復計算が全てのイメージに対して収束するという保証はありませんので、ユーザー各自で SCF 計算の収束性を確認する必要があります。全てのイメージについて SCF 収束をモニタするために、一時ファイルが利用できます。NEB 計算では、それぞれのイメージに対して一つの入力ファイル「*System.Name.dat\_#*」が作成されます。ここで「#」は 0 から MD.NEB.Number.Images+1 までの値で、それぞれの入力ファイル中の「system.name」は「system.name\_#」に変更されています。従って、ファイル「system.name\_#.DFTSCF」をモニタすることによって各イメージの SCF 計算の収束状況を確認できます。

## 49.7 並列計算

NEB 計算の並列化の設定はプロセスとスレッドの数に応じて自動的に行われます。ただし MPI 並列化で適正な計算負荷を実現するためには、適切なプロセス数を設定する必要があります。ここで適切な MPI プロセス数とは、「MD.NEB.Number.Images」で与えられるイメージ数で割り切れる MPI プロセス数のことです。この場合には計算負荷が最適化されます。MPI 並列プロセス数は原子数を越えられません。他の機能の並列化と同様に、MPI/OpenMP によるハイブリッド並列化もサポートしています。

ほとんどの場合に、デフォルトの並列化の設定で問題なく動作しますが、大規模系の計算において少数の MPI プロセスを用いるとメモリー不足が重大な問題となり得ます。デフォルトの MPI 並列化では、イメージが優先的に並列化されます。MPI プロセス数がイメージ数を上回ると、次にそれぞれの各イメージ内の計算が並列化されます。その際に、メモリーもまた同様に並列化されます。そのため、デフォルトの並列化設定では、多数の MPI プロセスが利用できなければメモリー不足のために異常終了する可能性があります。この状況を避けるために次のキーワードが利用できます。

```
MD.NEB.Parallel.Number      3
```

この例では、イメージを3つのグループに分けてそれぞれで並列計算を行います。そのため、MPI プロセスを3つのグループに分類し、各 MPI グループを各イメージのグループの計算に割り当てます。並列化構造は2階層となっており、上の階層がイメージのグループの並列化、下の階層はそれぞれのイメージ内の計算の並列化です。全てのイメージの計算を完了するために、グループ化した計算を最低  $\lceil (\text{イメージ数}) / (\text{MD.NEB.Parallel.Number}) \rceil$  回繰り返します。「MD.NEB.Parallel.Number=1」と設定した場合には、イメージの並列化は行わずに、イメージ内の計算に全ての MPI プロセスが割り当てられることになり、この場合にメモリ使用量が最小になります。従って、この並列化法は大規模系の NEB 計算に有効でしょう。このキーワードが入力ファイルで指定されない場合には、デフォルトの並列化法が適用されます。

#### 49.8 その他の注意

バルク系に対しては「FRAC」の代わりに「Ang」または「AU」で原子座標を指定して下さい。「FRAC」で指定した場合には、原子の位置は0から1の間の規格化座標に自動的に変換された後に、それに対応するカルテシアン座標が生成されます。この場合、幾何空間中でイメージが連続的に配置されないことになり、NEB 計算が破綻します。これを避けるために、「Ang」または「AU」で原子座標を指定して下さい。

NEB の機能を実装するために、3つのルーチン、「neb.c」、「neb\_run.c」、「neb\_check.c」のみが追加されています。メインルーチンは「neb.c」です。関連する方法を実装する場合には、ルーチン「neb.c」を参照して下さい。

## 50 Tersoff-Hamann 法による STM イメージ

Tersoff-Hamann 法 [71] によって走査型トンネル顕微鏡 (STM) のイメージを計算することができます。この方法は、化学ポテンシャルから計ったエネルギーウィンドウ内の部分電荷密度の計算に他なりません。部分電荷密度は次のキーワードを使って計算します。

```
partial.charge          on          # on|off, default=off
partial.charge.energy.window 0.0      # in eV
```

ここで、二番目のキーワードにより化学ポテンシャル (プラスの値は伝導帯を、マイナスは価電子帯を意味します) から計ったエネルギーウィンドウ (eV 単位) を指定します。部分電荷密度の計算は状態密度 (DOS) の計算の際に実行されますので、次のキーワード も指定する必要があります。

```
Dos.fileout            on          # on|off, default=off
Dos.Erange             -20.0 20.0    # default = -20 20
Dos.Kgrid              5 5 5       # default = Kgrid1 Kgrid2 Kgrid3
```

計算が正常に終了すれば、Tersoff-Hamann 近似の範疇で STM シミュレーションに利用できる「*System.Name.pden.cube*」が生成されます。例として、本手法によって得られたグラフェン層の STM 像を図 59 に示します。

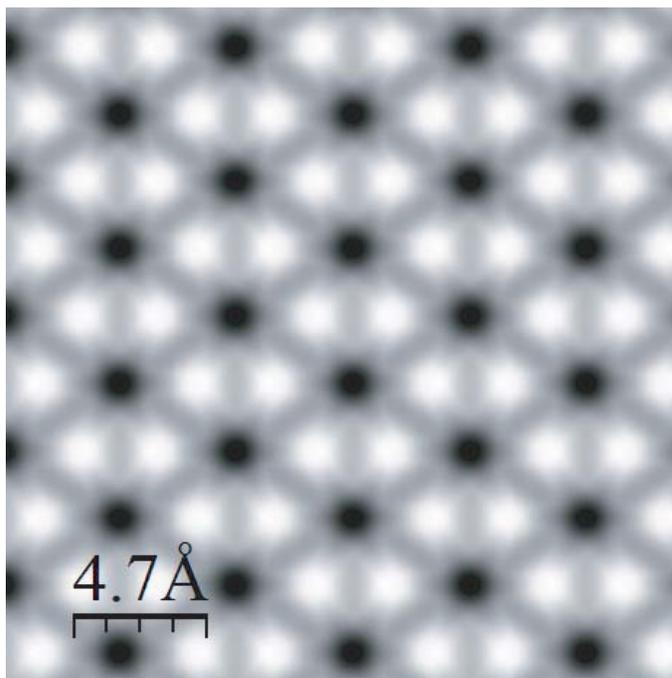


Figure 59: Tersoff-Hamann 近似によるグラフェン層の STM 像。計算では「partial.charge.energy.window」に 2eV に設定。入力ファイルはディレクトリ「work」中の「Graphene\_STM.dat」。ソフトウェア WSxM [153] を使い、cube ファイル「Graphene\_STM.pden.cube」を等値面値 0.0001 で可視化。

## 51 vdW 相互作用のための DFT-D2, DFT-D3 法

Grimme による DFT-D2 法 [135] 及び DFT-D3 法 [136, 137] により、vdW 相互作用を組み込むことができます。

### 51.1 DFT-D2 法

DFT-D2 法に関連するキーワードを以下に列挙します。

```
scf.dftD           on           # on|off, default=off
DFTD.Unit          Ang          # Ang|AU
DFTD.rcut_dftD    100.0        # default=100 (DFTD.Unit)
DFTD.d             20.0         # default=20
DFTD.scale6       0.75         # default=0.75
DFTD.IntDirection 1 1 1        # default=1 1 1 (1:on 0:off)
```

vdW 相互作用を組み込む際には、「scf.dftD」を「on」に設定して下さい。二体ポテンシャルのカットオフ半径はキーワード「DFTD.rcut\_dftD」で指定します。ここで、その単位は「DFTD.Unit」で与えます。Grimme の論文 [135] にある (12) 式の「d」値は「DFTD.d」によって指定します。そのデフォルト値は 20 です。Grimme の論文 [135] にある (11) 式の倍率は「DFTD.scale6」で指定します。PBE 関数を用いた場合のデフォルト値は 0.75 です。また本手法で導入された vdW 相互作用は「DFTD.IntDirection」によって a、b、もしくは c 軸方向で打ち切ることが可能で、ここで「1」は相互作用を含み、「0」は含まない場合です。また次のように、それぞれの原子の周期性を個別に制御できます。

```
<DFTD.periodicity
  1  1
  2  1
  3  1
  4  1
  ....
DFTD.periodicity>
```

ここで第 1 列は「Atoms.SpeciesAndCoordinates」と同様の通し番号、第 2 列はフラグで、対応する原子に対して「1」は周期的、「0」は非周期的であることを意味します。「0」を指定した原子は非周期的であると見なされ、周期セルとの相互作用は含まれません。

vdW 相互作用を実装する上での主な変更は二つのルーチン、「Total\_Energy.c」中の「Calc\_EdftD()」と「DFTDvdW\_init.c」に対して行われています。「DFTDvdW\_init.c」中では、vdW 補正のパラメータが容易に変更でき、「Total\_Energy.c」の「Calc\_EdftD()」ではどのように実装されているのか確認できます。

OpenMX では基底関数として局在軌道を用いるため、vdW 相互作用のような弱い相互作用の効果を調べる際には、基底関数重なり誤差 (BSSE) を考慮する事が重要です。BSSE を評価するために counterpoise 法 (CP 法) [46, 47] が使用できます。CP 法については「空原子の配置」の章を参照して下さい。

## 51.2 DFT-D3 法

Grimme らによる DFT-D3 法 [136, 137] により vdW 相互作用を取り入れることが可能です。関連パラメータは以下となります (デフォルトのパラメータは GGA-PBE 用のものが設定されています)。

```
scf.dftD          on          # on|off, default=off
version.dftD      3           # 2|3, default=2
DFTD3.damp        bj         # zero|bj, default=bj
DFTD.Unit         AU         # Ang|AU
DFTD.rcut_dftD   100.0       # default=100 (DFTD.Unit)
DFTD.cncut_dftD  40          # default=40 (DFTD.Unit)
DFTD.IntDirection 1 1 1      # default=1 1 1 (1:on 0:off)
```

DFT-D2 もしくは DFT-D3 計算を実施する際には「scf.dftD」を「on」にします。DFT-D2 では、「version.dftD」を「2」とし、DFT-D3 では「3」として下さい。DFT-D3 の減衰関数としてゼロ及び Becke-Johnson(BJ)[137] の双方が使用可能です。相互作用のカットオフ半径は「DFTD.rcut\_dftD」で指定し、配位数の計算に用いるカットオフは「DFTD.cncut\_dftD」です。「DFTD.Unit」で指定する単位は AU を推奨しています。「DFTD.IntDirection」では、a、b、c 軸方向の仮想原子との相互作用を加味するかどうかを指定し、「1」の場合には加味し、「0」の場合には無視します。各原子の周期性の制御は、DFT-D2 法と同様に以下の様に行います。

```
<DFTD.periodicity
 1  1
 2  1
 3  1
 4  1
....
DFTD.periodicity>
```

ここで第 1 列は「Atoms.SpeciesAndCoordinates」と同様の通し番号、第 2 列はフラグで、対応する原子に対して「1」は周期的、「0」は非周期的であることを意味します。「0」を指定した原子は非周期的であると見なされ、周期セルとの相互作用は含まれません。

DFT-D3 法のための主要な改変は、「DFTD3vdW\_init.c」と「Total\_Energy.c」内の「Calc\_EdftD()」のみとなっています。「DFTD3vdW\_init.c」において、vdW 相互作用パラメータを変更することが可能です。また、「Total\_Energy.c」内の「Calc\_EdftD()」を見ることで、どの様にエネルギーが計算されているか確認することができます。

その他の汎関数を使用する際には、以下のキーワードを設定して下さい。

```
DFTD.scale6      1          # default=0.75|1.0 (for DFT-D2|DFT-D3)
DFTD.scale8      0.7875     # default=0.722|0.7875 (for PBE with zero|bj damping)
DFTD.sr6         1.217     # default=1.217 (for PBE)
DFTD.a1          0.4289     # default=0.4289 (for PBE)
DFTD.a2          4.4407     # default=4.4407 (for PBE)
```

「s<sub>6</sub>」と「s<sub>8</sub>」は Grimme の論文 [136] 中の式 (3) に含まれるスケーリングファクターであり、「DFTD.scale6」と「DFTD.scale8」にて指定します。このスケーリングファクターは汎関数と減衰関数に依存して変

わります。また、ゼロ減衰関数を用いる場合には「sr6」、BJ減衰関数を用いる場合には「a1」と「a2」を設定する必要があります。これらのパラメータは Grimme の論文の式 (6) に含まれるものです。

DFT-D3 計算の例として、2つのベンゼン分子(平行に配置、 $D_{6h}$  対称)間の相互作用を、2分子間の距離の関数として計算したものを図 60 に示します。計算に用いた以下の入力ファイルは「work/DFT-D3」ディレクトリに収められています。

```
Dimer-Ben-10.0.dat  Dimer-Ben-3.88.dat  Dimer-Ben-4.5.dat  Mono-Ben-1.dat
Dimer-Ben-3.3.dat  Dimer-Ben-3.89.dat  Dimer-Ben-5.0.dat  Mono-Ben-2.dat
Dimer-Ben-3.4.dat  Dimer-Ben-3.8.dat  Dimer-Ben-6.0.dat  Mono-Ben.dat
Dimer-Ben-3.6.dat  Dimer-Ben-3.9.dat  Dimer-Ben-7.0.dat
Dimer-Ben-3.86.dat Dimer-Ben-4.0.dat  Dimer-Ben-8.0.dat
Dimer-Ben-3.87.dat Dimer-Ben-4.2.dat  Dimer-Ben-9.0.dat
```

1分子を「Mono-Ben.dat」を用いて最適化した後、2分子間の距離を変えた計算を「Dimer-Ben-#.dat (#=3.3-9.0)」を用いて実施しました。その際、ベンゼンの構造は1分子で最適化した構造に固定しています。「Mono-Ben-1.dat」及び「Mono-Ben-2.dat」では、1分子の計算を counterpoise 補正を行ったものになっています。最も安定な分子間距離が 3.87Å として得られており、これは Density fitted local second-order Møller Plesset perturbation theory (DF-LMP2) [138] による結果 (3.89 Å) とよく一致しています。counterpoise 補正を加えた相互作用エネルギーは 1.73 kcal/mol となっており、こちらも文献値 (1.7 kcal/mol) [138] とよく一致し、基底関数重なり誤差が無視できないものであることが分かります。

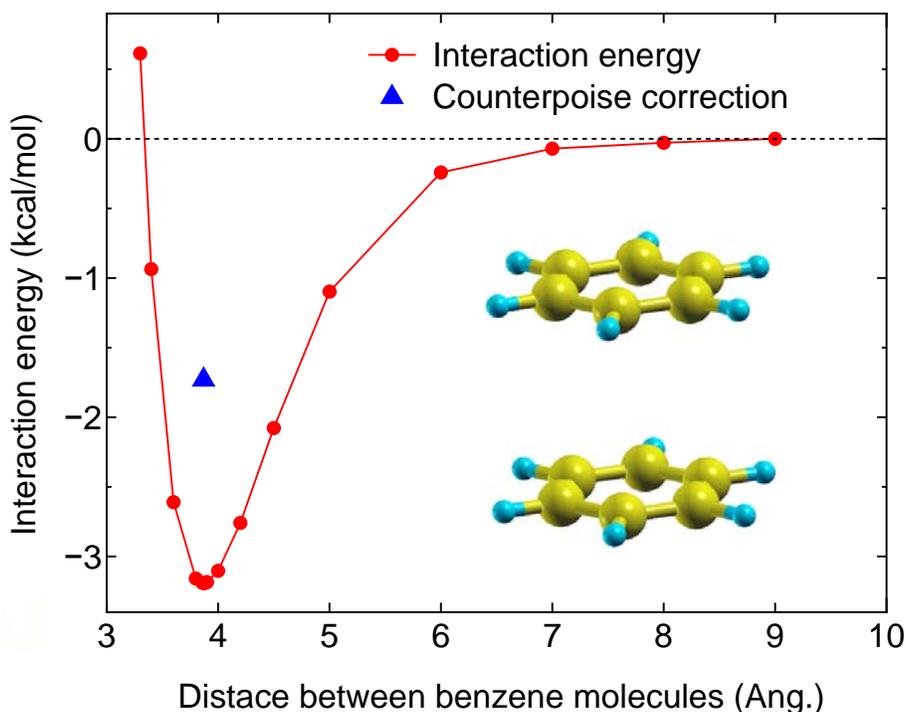


Figure 60: 平行に並んだベンゼン ( $D_{6h}$  対称性) 二量体における分子間の相互作用エネルギー。counterpoise 補正を加えた相互作用エネルギーを三角で示す。計算に用いた入力ファイルは、全て「work/DFT-D3」ディレクトリに収められている。

## 52 バンドアンフォールディング法

界面、不純物、欠陥や構造歪み等を含む不完全な系をスーパーセル法によって計算した場合、得られたバンド構造と角度分解光電子分光法 (ARPES) などにより得られたスペクトルとの比較において、実験で観測される周期性と計算で用いたスーパーセルの周期性が異なることが一般的です。スーパーセル法で得られたバンド構造は対応する小さなブリルアンゾーン内で折り畳まれており (folding)、この場合、実験で観測された周期性との整合性をとるために適切な単位胞に対するブリルアンゾーンを用いて再表示する必要があります。適切な単位胞の選び方は自明ではありませんが、OpenMX ではスーパーセル法で得られたバンド構造を、ユーザーが入力ファイルで指定した単位胞のブリルアンゾーンで展開する (unfolding、アンフォールディング) ことが可能です [142]。このバンドアンフォールディング法はコリニア計算、ノンコリニア計算の双方に対応しています。実験結果との比較を行う以外にも、バンドアンフォールディング法を用いることでそれぞれのバンドがどの擬原子軌道から構成されているか解析が行う事が可能となり、バンドの物理的背景や、構造不完全性の導入がもたらす影響を議論するのに有用な手法となっています。

以下では、一連の計算を通してバンドアンフォールディング法の機能を紹介します。

### 52.1 バンド構造の解析

まず、構造不完全性のないプリミティブセルの SiC に対するバンド計算において、それぞれのバンドを各擬原子軌道からの寄与に分解してみましょう。ここでは、バンドのアンフォールディングは行いません。SCF 計算は、以下の様に実行します。

```
% mpirun -np 16 openmx SiC_Primitive.dat > sic_primitive.std &
```

入力ファイル「SiC\_Primitive.dat」は「work/unfolding\_example」ディレクトリに含まれており、基底関数や構造は以下の様に指定されています。

```
Species.Number      3
<Definition.of.Atomic.Species
  C   C7.0-s2p2d1    C_PBE19
  Si  Si7.0-s2p2d1    Si_PBE19
  Te  Te11.0-s2p2d2f1 E
Definition.of.Atomic.Species>

Atoms.Number      3
Atoms.SpeciesAndCoordinates.Unit  FRAC  # Ang|AU
<Atoms.SpeciesAndCoordinates
  1  C   0.33333333  0.66666666  0.50000000  2.0  2.0
  2  Si  0.66666666  0.33333333  0.50000000  2.0  2.0
  3  Te  0.00000000  0.00000000  0.50000000  0.0  0.0
Atoms.SpeciesAndCoordinates>

Atoms.UnitVectors.Unit          Ang  # Ang|AU
<Atoms.UnitVectors
  3.0690  0.0000000000  0.000
```

```
-1.5345 2.6578319641 0.000
0.0000 0.0000000000 10.000
Atoms.UnitVectors>
```

伝導帯の記述を改善するために、長い裾の基底関数を有する空原子を、六角形の中央に配置しています。キーワード「Band.dispersion」を以下の様に設定して下さい。

```
Band.dispersion      on      # on|off, default=off
Band.Nkpath          3
<Band.kpath
60 0.33333333333 0.33333333333 0.00000000000 0.00000000000 0.00000000000 0.00000000000 K G
52 0.00000000000 0.00000000000 0.00000000000 0.50000000000 0.00000000000 0.00000000000 G M
30 0.50000000000 0.00000000000 0.00000000000 0.33333333333 0.33333333333 0.00000000000 M K
Band.kpath>
```

「sic\_primitive.BANDDAT1」により、図 61(a) のようにバンド構造を描画することが可能です。バンド構造の描画については「Band dispersion」の章をご覧ください。

### バンド構造の解析に関連するキーワード

バンドアンフォールディング法の計算にあたっては、以下のキーワードも設定する必要があります。

```
Unfolding.Electronic.Band      on      # on|off, default=off
Unfolding.LowerBound          -10.0   # default=-10 eV
Unfolding.UpperBound           6.0    # default= 10 eV

Unfolding.Nkpoint              4

<Unfolding.kpoint
K 0.33333333333 0.33333333333 0.00000000000
G 0.00000000000 0.00000000000 0.00000000000
M 0.50000000000 0.00000000000 0.00000000000
K 0.33333333333 0.33333333333 0.00000000000
Unfolding.kpoint>

Unfolding.desired_totalnkpt    30
```

上記のキーワードの内容を以下に説明します。

- **Unfolding.Electronic.Band**            on| off, default=off  
「on」にすることで、バンドアンフォールディング法を行います。デフォルトは「off」です。
- **Unfolding.LowerBound**  
「Unfolding.LowerBound」によって解析を行うバンドのエネルギー最小値を指定します。エネルギーはフェルミエネルギーを基準とし、デフォルト値は-10 eV となっています。
- **Unfolding.UpperBound**  
「Unfolding.UpperBound」では、解析を行うバンドのエネルギー最大値を指定します。エネルギーはフェルミエネルギーを基準とし、デフォルト値は 10 eV となっています。

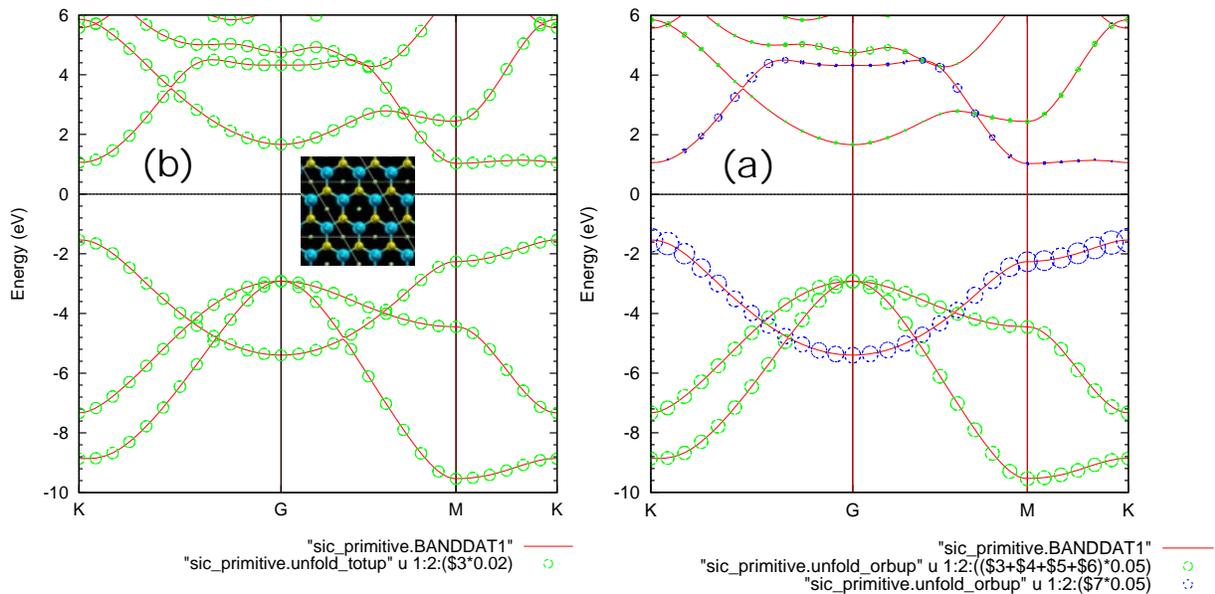


Figure 61: (a) プリミティブセルの SiC のバンド構造。赤線が通常の計算手法により求めたものであり、緑丸は全スペクトルの重みを示している。(b) 軌道分解されたスペクトルの重み。緑の丸印は炭素原子の  $s$ 、 $p_x$  と  $p_y$  のスペクトルの重みの総和を表し、紫の丸印は炭素原子の  $p_z$  軌道のスペクトルの重みの総和を、丸印の半径は重みの大きさを反映している。計算には「work/unfolding\_examples」内の「SiC\_Primitive.dat」を用いた。

- **Unfolding.Nkpoint**

「Unfolding.Nkpoint」では、「Unfolding.kpoint」で列挙する k 点の数を指定します。

- **Unfolding.kpoint**

「Unfolding.kpoint」では、「Unfolding.Nkpoint」にて指定した数だけ k 点を指定します。上記の例では 4 つの k 点が列挙されており、「K」から「G」, 「G」から「M」, 「M」から「K」のパスとなっています。このパスに沿ってバンドの解析が行われます。k 点を指定する際には、「Atoms.UnitVectors」で指定した単位胞の逆格子ベクトルに対する規格化座標を用いて下さい。「Unfolding.ReferenceVectors」によって、バンドアンフォールディング法に用いる格子ベクトルを変更することもできます。「Unfolding.ReferenceVectors」を指定しなかった場合には、「Atoms.UnitVectors」で指定した格子ベクトルが自動的に使用されることに注意して下さい。「Unfolding.ReferenceVectors」については、次の節にて説明します。

- **Unfolding.desired\_totalnkpt**

「Unfolding.kpoint」において指定されたパスは概ね等間隔に分割され、分割される間隔は、(k 空間における総パス長) / 「Unfolding.desired\_totalnkpt」で見積もられます。ただし、実際の間隔は「Unfolding.kpoint」にて指定した k 点が常に解析に含まれるように、自動的に調整されます。

## バンド構造の解析に関連した出力ファイル

SCF が収束すると、バンドの解析・アンフォールディングに関連した以下のファイルが生成されます。

- sic\_primitive.unfold\_totup

参考文献 [142] 中の式 (24) より得られる全スペクトルの重みが保存されています。1 列目は「Unfolding.kpoint」にて指定された最初の k 点からの距離 (Bohr<sup>-1</sup> 単位)、2 列目はフェルミエネルギーを基準とするエネルギー (eV 単位)、3 列目は全スペクトルの重みです。スピン分極した計算の場合には、「System.Name.unfold\_totdn」がダウンスピン用に追加で生成され、ノンコリニア計算の場合には「System.Name.unfold\_tot」という単一のファイルが生成されます。

- sic\_primitive.unfold\_orbup

参考文献 [142] 中の式 (26) より得られる軌道分解されたスペクトルの重みが保存されています。1 列目は「Unfolding.kpoint」にて指定された最初の k 点からの距離 (Bohr<sup>-1</sup> 単位)、2 列目はフェルミエネルギーを基準とするエネルギー (eV 単位)、3 列目は軌道分解されたスペクトルの重みです。軌道分解されたスペクトルの重みの出力順序は「System.Name.out」をご参照下さい。スピン分極した計算の場合には、「System.Name.unfold\_orbdn」がダウンスピン用に追加で生成され、ノンコリニア計算の場合には「System.Name.unfold\_orb」という単一のファイルが生成されます。

- sic\_primitive.unfold\_plotexample

上記のデータを gnuplot にて描画するための例が「System.Name.unfold\_plotexample」として生成されます。gnuplot のコマンドラインにおいて、以下の様に打ち込むことで描画可能です。

```
gnuplot> load 'sic_primitive.unfold_plotexample'
```

目的に応じて、適宜編集をして使用して下さい。

gnuplot を用いることで、「sic\_primitive.BANDDAT1」、「sic\_primitive.unfold\_totup」の描画を行うには、以下のコマンドを使用して下さい。

```
gnuplot> set style data lines
gnuplot> set zeroaxis
gnuplot> set key below
gnuplot> set ytics 1
gnuplot> set mytics 5
gnuplot> set xra [0.000000:1.708883]
gnuplot> set yra [-10.0:6.0]
gnuplot> set ylabel "eV"
gnuplot> set xtics ("K" 0.000000, "G" 0.722258, "M" 1.347753, "K" 1.708882)
gnuplot> p "sic_primitive.BANDDAT1","sic_primitive.unfold_totup" u 1:2:($3*0.02) w circle
```

上記のコマンドにより、図 61 (a) と同様の図を描くことができます。実線は「sic\_primitive.BANDDAT1」を、丸印は「sic\_primitive.unfold\_totup」を表しています。丸印の大きさはスペクトルの重みを反映しますが、今回はオリジナルセルのブリルアンゾーンにて得られたバンドを解析したため、全て同じ大きさとなっています。

次に、軌道分解されたスペクトルの重み「sic\_primitive.unfold\_orbup」を見てみましょう。上記と同様に、以下のコマンドで描画することが可能です。

```
gnuplot> p "sic_primitive.BANDDAT1","sic_primitive.unfold_orbup" u 1:2:((($3+$4+$5+$6)*0.05) w circle,
"sic_primitive.unfold_orbup" u 1:2:($7*0.05) w circle
```

このコマンドにより、図 61 (b) を得ることができます。この例では、炭素原子の  $s$ 、 $p_x$ 、 $p_y$  軌道のスペクトルの重みの総和を緑の丸印で、炭素原子の  $p_z$  軌道のスペクトルの重みを紫の丸印で描画しています。本解析を行うことで、 $\sigma$  バンドと  $\pi$  バンドを明確に区別することが可能となります。「sic\_primitive.unfold\_orbup」の形式については、上述の説明をご参照下さい。

## 52.2 バンド構造のアンフォールディング

ここでは、スーパーセル法によって得られたバンド構造をどの様に参照単位胞のブリルアンゾーンにアンフォールディングするか説明します。例として、ここでも不完全性のない 2 次元蜂の巣構造をもつ SiC を考えます。ただし、計算セルを以下の様に  $(2 \times 2)$  のスーパーセルに拡張したものを考えます。

```
Atoms.Number 12
Atoms.SpeciesAndCoordinates.Unit  FRAC  # Ang|AU
<Atoms.SpeciesAndCoordinates
  1 C    0.16666666  0.33333333  0.50000000  2 2
  2 C    0.66666666  0.33333333  0.50000000  2 2
  3 C    0.16666666  0.83333333  0.50000000  2 2
  4 C    0.66666666  0.83333333  0.50000000  2 2
  5 Si   0.33333333  0.16666666  0.50000000  2 2
  6 Si   0.83333333  0.16666666  0.50000000  2 2
  7 Si   0.33333333  0.66666666  0.50000000  2 2
  8 Si   0.83333333  0.66666666  0.50000000  2 2
  9 Te   0.00000000  0.00000000  0.50000000  0 0
 10 Te   0.50000000  0.00000000  0.50000000  0 0
 11 Te   0.00000000  0.50000000  0.50000000  0 0
 12 Te   0.50000000  0.50000000  0.50000000  0 0
Atoms.SpeciesAndCoordinates>
```

```
Atoms.UnitVectors.Unit          Ang  # Ang|AU
<Atoms.UnitVectors
  6.138  0.0000000000  0.00
 -3.069  5.3156639282  0.00
  0.000  0.0000000000 10.00
Atoms.UnitVectors>
```

2 次元 SiC のスーパーセルに対する SCF 計算は以下の様に行います。

```
% mpirun -np 16 openmx SiC_C_NSP_P.dat > sic_c_nsp_p.std &
```

入力ファイル「SiC\_C\_NSP\_P.dat」は「work/unfolding\_example」に収められています。SCF 計算が終わると、以下のアンフォールディング計算に必要なファイルが生成されます。

```
sic_c_nsp_p.unfold_totup
sic_c_nsp_p.unfold_orbup
sic_c_nsp_p.unfold_plotexample
```

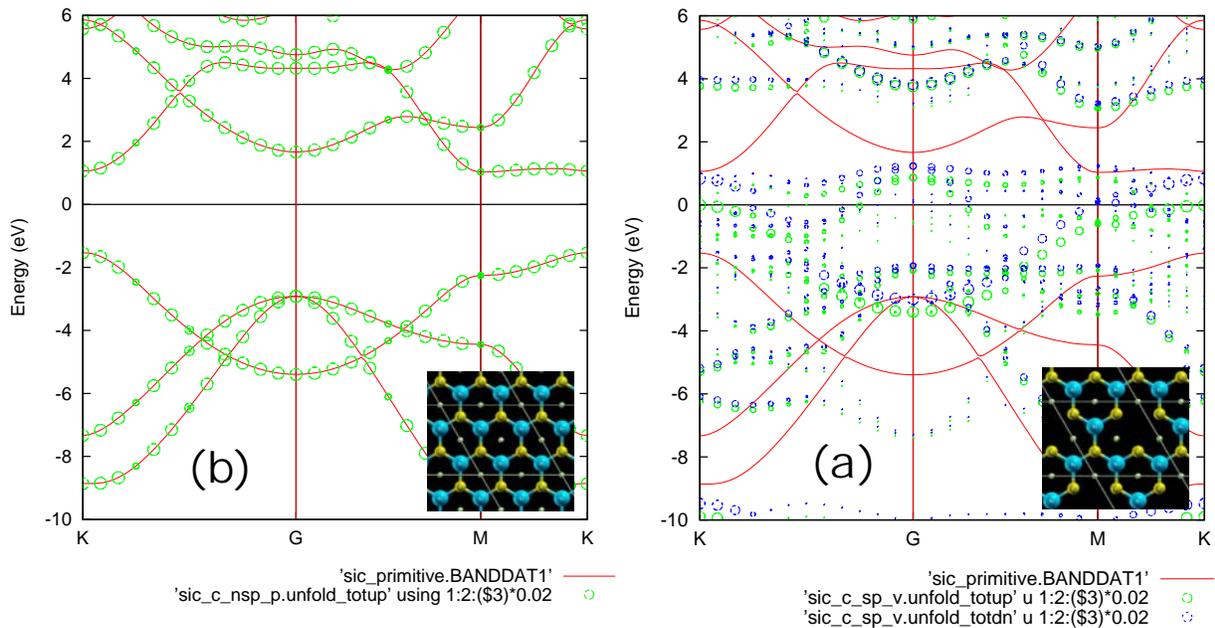


Figure 62: (a) 2次元蜂の巣構造のSiC (2 × 2) スーパーセルのバンド構造。赤線はプリミティブセルに対する通常の計算により得られたバンド構造であり、緑の丸印はアンフォールディングにより得られた全スペクトルの重みを表している。また、丸印の大きさは重みの強度を反映している。計算に用いた「SiC\_C\_NSP\_P.dat」は「work/unfolding\_example」に収められている。(b) Si欠陥を持つ2次元蜂の巣構造のSiC (2 × 2) スーパーセルのバンド構造。赤線はプリミティブセルに対する従来の計算により得られたバンド構造であり、緑と青の丸印はアンフォールディング法で得られたアップスピン及びダウンスピンの全スペクトルの重みを表している。また、丸の大きさは重みの強度を反映している。計算に用いた「SiC\_C\_SP\_V.dat」は「work/unfolding\_example」に収められている。

プリミティブセルで得られたバンド構造と重ねて「sic\_c\_nsp\_p.unfold\_totup」を描画することにより、図62(a)と同様の図を得ることができます。スーパーセルからアンフォールディングされたバンドが、プリミティブセルのバンドを完全に再現していることが確認できます。

バンドのアンフォールディングを実施するためには、前節で示したキーワードの他に、以下のキーワードを指定する必要があります。

```
<Unfolding.ReferenceVectors
  3.0690  0.0000000000  0.000
 -1.5345  2.6578319641  0.000
  0.0000  0.0000000000  10.000
Unfolding.ReferenceVectors>
```

```
<Unfolding.Map
  1  1
  2  1
  3  1
  4  1
```

```

5 2
6 2
7 2
8 2
9 3
10 3
11 3
12 3
Unfolding.Map>

```

上記キーワードの指定に関して以下に説明します。

- **Unfolding.ReferenceVectors**

「Unfolding.ReferenceVectors」では、参考文献 [142] の式 (24) に従ってバンドアンフォールディングを行う際に参照する単位胞を定義します。上記の例では、プリミティブセルを参照セルとしています。入力形式は「Atoms.UnitVectors」と同じであり、1行目、2行目、3行目に a-, b-, c-軸を記載して下さい。参照セルの単位は、スーパーセルで用いたものと同じであり、「Atoms.UnitVectors.Unit」で指定したものとなります。「バンド構造の解析」の節では、「Unfolding.ReferenceVectors」を設定しました。このような場合には、「Atoms.UnitVectors」が自動的に「Unfolding.ReferenceVectors」として設定されます。

- **Unfolding.Map**

「Unfolding.Map」では、スーパーセル中の原子を参照セル内の原子にどの様にマッピングするかを指定します。1列目には「Atoms.SpeciesAndCoordinates」にて指定した原子の通し番号を記述し、2列目には各原子が所属するグループの通し番号を記述します。上記の例では、原子 1 から 4 はグループ 1 に、原子 5 から 8 がグループ 2 に、原子 9 から 12 がグループ 3 に所属します。「Unfolding.Map」にて与えられた情報に基づいて、参考文献 [142] の式 (17) に従って各原子の再ラベリングを行います。グループの識別番号は、正の整数であれば「1」からスタートする必要はなく、昇順にする必要もありません。

ここでは、更なる例として、2次元蜂の巣構造の SiC (2×2) スーパーセルにおいて、Si 欠陥を導入した場合を紹介します。まず、以下の様に SCF 計算を行います。

```
% mpirun -np 16 openmx SiC_C_SP_V.dat > sic_c_sp_v.std &
```

入力ファイル「SiC\_C\_SP\_V.dat」は「work/unfolding\_example」ディレクトリに収められています。(2×2) スーパーセルから Si 原子を取り除くと、入力ファイルは以下の様になります。

```

Atoms.Number 11
Atoms.SpeciesAndCoordinates.Unit  FRAC  # Ang|AU
<Atoms.SpeciesAndCoordinates
  1 C    0.16666666  0.33333333  0.50000000  2.5 1.5
  2 C    0.66666666  0.33333333  0.50000000  2.5 1.5
  3 C    0.16666666  0.83333333  0.50000000  2.5 1.5
  4 C    0.66666666  0.83333333  0.50000000  2.5 1.5

```

```

5 Si  0.33333333  0.16666666  0.50000000  2.5 1.5
6 Si  0.83333333  0.16666666  0.50000000  2.5 1.5
7 Si  0.33333333  0.66666666  0.50000000  2.5 1.5
8 Te  0.00000000  0.00000000  0.50000000  0.0 0.0
9 Te  0.50000000  0.00000000  0.50000000  0.0 0.0
10 Te 0.00000000  0.50000000  0.50000000  0.0 0.0
11 Te 0.50000000  0.50000000  0.50000000  0.0 0.0

```

Atoms.SpeciesAndCoordinates>

スーパーセル内の原子を以下の様に参照セルにマッピングします。

<Unfolding.Map

```

1 1
2 1
3 1
4 1
5 2
6 2
7 2
8 3
9 3
10 3
11 3

```

Unfolding.Map>

SCF 計算が終わると、以下のアンフォールディング法に関連するファイルが生成されます。

```

sic_c_sp_v.unfold_totup
sic_c_sp_v.unfold_totdn
sic_c_sp_v.unfold_orbup
sic_c_sp_v.unfold_orbdn
sic_c_sp_v.unfold_plotexample

```

プリミティブセルで得られたバンド構造と併せて「sic\_c\_sp\_v.unfold\_totup」と「sic\_c\_sp\_v.unfold\_totdn」を gnuplot によりプロットすることで、図 62(b) を得ることができます。アンフォールディングされたスペクトルの重みを見ると、欠陥導入後も完全系における特徴を維持していることが分かります。さらに、化学ポテンシャルが大きく高エネルギー側にシフトしていることや、炭素原子のダングリングボンドによってスピン分極していることも見て取れます。これらの解析のとおり、不完全性の導入によって本来のバンド構造が受ける影響をバンドアンフォールディング法を用いることで評価することができます。

欠陥ではなく不純物を導入した場合には、不純物の識別番号を指定するだけで計算が可能です。SiC (2×2) スーパーセルの 13 番目の原子として不純物を導入した場合には、再ラベリングのマッピングルールを以下の様に定義します。

<Unfolding.Map

```

1 1
2 1

```

```
3 1
4 1
5 2
6 2
7 2
8 2
9 3
10 3
11 3
12 3
13 4
```

```
Unfolding.Map>
```

複数の不純物が存在する場合にも、上記と同様にマッピングルールを定義することができます。

### 52.3 参照セルの原点

参考文献 [142] の式 (17) を見ると、再ラベリングには  $R$  から  $R+r_0(M)$  と  $M$  から  $m'(M)$  の2つのマッピングルールが必要となっていることが分かります。ここで、 $R$  と  $r_0(M)$  はスーパーセルと参照セルの格子ベクトルであり、 $M$  と  $m'(M)$  は原子軌道のインデックスです。 $M$  から  $m'(M)$  の再ラベリングについては、すでに「Unfolding.Map」にてマッピングルールを定義しています。この節では、 $R$  から  $R+r_0(M)$  の再ラベリングをどの様に行うかを説明します。 $R$  から  $R+r_0(M)$  の再ラベリングは、実空間の原子座標に対して参照セルがどのように配置しているかに依存しており、これは参照セルの原点の取り方に依存しているとも考えることができます。一方、 $M$  から  $m'(M)$  の再ラベリングのマッピングルールは多くの場合変わらないため、アンフォールディングの重みは参照セルの原点の取り方には敏感ではありません。ただし、表面や大きく歪んだ系などにおいては原点の取り方によってマッピングルールが変化する場合がありますので注意が必要です。OpenMX のデフォルトでは、以下の2つのルールに基づいて原点を推定します。1つ目のルールは、「Unfolding.Map」にて同じ識別番号を付与された原子を各参照セルに可能な限り1つだけ割り当てられるように原点を選択するというものです。大きく歪んだ構造では、同一の識別番号でラベルされた原子を各参照セルに多重に割り当ててしまうことが起こりうるため、OpenMX の実装ではこのような状況を自動的に回避するような工夫がされています。2つ目のルールは、再ラベリングによって割り当てられた原子の数がゼロではない参照セルの総数を最小にするよう原点を決定するというものです。スーパーセル法で表現される表面系ではスラブ間に真空領域を有するので、割り当てられた原子数がゼロではない参照セルの総数が、原点の取り方によって変化する可能性があります。OpenMX の実装では自動的に割り当て原子数がゼロではない参照セルの総数を最小にするよう工夫されています。これら2つのルールによって、多くのユーザーの要件を満たすことができますが、原点を自ら管理したい場合もあります。この目的のために、以下のキーワードが利用可能です。

```
<Unfolding.ReferenceOrigin
 0.1 0.2 0.3
Unfolding.ReferenceOrigin>
```

これらには「Atoms.UnitVectors.Unit」にて定義した単位を用います。

## 52.4 アンフォールディングされたスペクトル重みの強度マップ

アンフォールディングされたスペクトル重みは強度マップとして可視化できます。重み  $w$  は以下の様に Lorentian 関数によって滑らかにされます。

$$L(k, E) = \frac{w}{(k/\Delta_k)^2 + (E/\Delta_E)^2 + 1}$$

$k$  は Bohr<sup>-1</sup> 単位の  $k$ -ベクトルの値であり、 $E$  は eV 単位のエネルギー、 $\Delta_k$  及び  $\Delta_E$  は滑らかにする度合です。強度マップの絶対値には物理的な意味がないことにご注意下さい。強度マップの可視化は以下の 3 ステップで実施可能です。

### (1) intensity\_map.c のコンパイル

「source」ディレクトリにおいて、以下の様に「intensity\_map.c」をコンパイルして下さい。

```
gcc intensity_map.c -lm -o intensity_map
```

生成された実行形式ファイル「intensity\_map」を作業ディレクトリにコピーして下さい。

### (2) 強度マップの作成

アンフォールディング計算の終了後、強度マップを作成するために必要なメッシュデータが書き込まれたファイルを生成することができます。ここでは、前節にて議論した Si 欠陥をもつ 2 次元蜂の巣構造の SiC (2×2) スーパーセルを例に説明します。アンフォールディング計算の入力ファイルは「SiC\_C\_SP\_V.dat」であり、メッシュデータファイル「sic-intmap.txt」を以下の様に生成します。

```
./intensity_map sic_c_sp_v.unfold_totup -c 3 -k 0.1 -e 0.1 -l -10 -u 6 > sic-intmap.txt
```

各引数の説明は以下となります。

- -c : 解析したいスペクトル重みが書かれた列番号
- -k :  $k$  ベクトルでの  $\Delta_k$  の値 (Bohr<sup>-1</sup>)
- -e :  $\Delta_E$  の値 (eV)
- -l : マップに描くバンドのエネルギー最小値 (eV)
- -u : マップに描くバンドのエネルギー最大値 (eV)

「*System.Name.unfold\_orbup*(dn)」を解析する際には、「*System.Name.out*」の擬原子軌道の通し番号を参考に、「-c」の列番号を設定して下さい。また、「-c」で指定された  $N_{\text{col}}$  は、「*System.Name.out*」の擬原子軌道の通し番号  $N_{\text{seq}}$  に関連していることに注意して下さい。両者の関係は「 $N_{\text{col}} = N_{\text{seq}} + 2$ 」です。

### (3) 強度マップの描画

gnuplot を用いて強度マップを描画可能です。入力ファイル「SiC\_C\_SP\_V.dat」を用いた計算では、以下の様に描画して下さい。

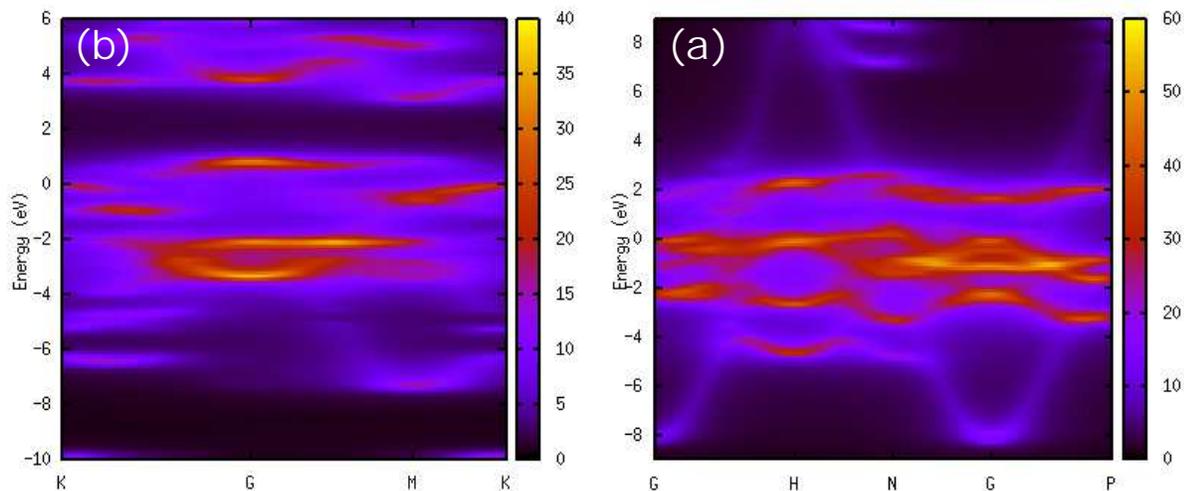


Figure 63: (a) アンフォールディングされた全スペクトル重み (アップスピン状態) の強度マップ。系は Si 欠陥をもつ 2 次元蜂の巣構造の SiC (2 × 2) スーパーセル。SCF 計算は「SiC\_C\_SP\_V.dat」において「Unfolding.desired\_totalnkpt」を 300 に設定し、 $\Delta_k = 0.1$  (Bohr<sup>-1</sup>)、 $\Delta_E = 0.1$  (eV) とした。(b) 構造揺らぎを持つ BCC 鉄のアンフォールディングされた全スペクトル重みの強度マップ。SCF 計算は「BCC\_Fe\_N\_SO\_Disorder.dat」において「Unfolding.desired\_totalnkpt」を 300 に設定し、 $\Delta_k = 0.1$  (Bohr<sup>-1</sup>)、 $\Delta_E = 0.1$  (eV) とした。

```
gnuplot> set yrange [-10.000000:6.000000]
gnuplot> set ylabel 'Energy (eV)'
gnuplot> set xtics('K' 0.000000,'G' 0.722259,'M' 1.347753,'K' 1.708883)
gnuplot> set xrange [0:1.708883]
gnuplot> set arrow nohead from 0,0 to 1.708883,0
gnuplot> set arrow nohead from 0.722259,-10.000000 to 0.722259,6.000000
gnuplot> set arrow nohead from 1.347753,-10.000000 to 1.347753,6.000000
gnuplot> set pm3d map
gnuplot> sp 'sic-intmap.txt'
```

これらのコマンドにより、図 63(a) を得ることができます。

## 52.5 ノンコリニア DFT 計算の場合

アンフォールディングバンドの機能は、スピン軌道相互作用を含めたノンコリニア DFT 計算や plus U 法の計算をサポートしています。これらの機能との併用に際して、追加のキーワードはありません。

## 52.6 入力ファイルの例

6 種類の計算例に対する入力ファイルが「work/unfolding\_example」に収められています。

- SiC\_Perfect.dat

不完全性のない 2 次元蜂の巣構造 SiC のプリミティブセルについて、各バンドを構成する擬原子軌道を解析する際の例です（コリニア、スピン分極なし）。

- SiC\_C\_NSP\_P.dat

不完全性のない 2 次元蜂の巣構造 SiC の  $(2 \times 2)$  スーパーセルについて、アンフォールディングバンド法を用いる際の例です（コリニア、スピン分極なし）。

- SiC\_C\_SP\_V.dat

Si 欠陥をもつ 2 次元蜂の巣構造 SiC の  $(2 \times 2)$  スーパーセルについて、アンフォールディングバンド法を用いる際の例です（コリニア、スピン分極あり）。

- SiC\_NC\_SO\_P.dat

不完全性のない 2 次元蜂の巣構造 SiC の  $(2 \times 2)$  スーパーセルについて、アンフォールディングバンド法を用いる際の例です（ノンコリニア計算でスピン軌道相互作用を含む）。

- SiC\_NC\_SO\_V.dat

Si 欠陥をもつ 2 次元蜂の巣構造 SiC の  $(2 \times 2)$  スーパーセルについて、アンフォールディングバンド法を用いる際の例です（ノンコリニア計算でスピン軌道相互作用を含む）。

- BCC\_Fe\_Perfect.dat

不完全性のない BCC 構造の Fe のプリミティブセルに対して、各バンドを構成する擬原子軌道を解析する際の例です（コリニア、スピン分極あり）。

- BCC\_Fe\_C\_SP\_Perfect.dat

不完全性のない BCC 構造の Fe の立法セルに対して、アンフォールディングバンド法を用いる際の例です。（コリニア、スピン分極あり）

- BCC\_Fe\_N\_SO\_Disorder.dat

構造揺らぎをもつ BCC 構造の Fe の立法セルに対して、アンフォールディングバンド法を用いる際の例です（ノンコリニア計算でスピン軌道相互作用を含む）。

## 53 k 空間でのスピネクスタの解析

Rashba 効果等で見られるバンド構造のスピンスplitはスピノ軌道相互作用の考慮で生じることがあります。スピンスplitは各固有状態・波数ベクトル  $k$  に分解され、固有状態・ $k$  で分解されたスピンスplitは、逆空間でのバンド分散関係におけるスピノ構造であり、スピネクスタと呼ばれます。スピノ軌道相互作用を含めたノンコリニア計算に対して、ポストプロセスコード「kSpin」を用いてスピネクスタが解析できます。固有状態・ $k$  で分解されたスピノ密度行列は、以後は  $k$  空間スピノ密度行列と呼ぶことにします。この  $k$  空間スピノ密度行列は二成分スピノルから計算され、 $2 \times 2$  行列の形式をとります。スピネクスタはこの  $2 \times 2$  行列から計算されます。ポストプロセスコード「kSpin」による解析では、スピネクスタに加えて、Rashba 効果などのスピノ軌道相互作用により生じる現象の物理的な起源を解析できるように  $k$  空間スピノ密度行列の情報も提供されます。固有状態・ $k$  への分解だけでなく、原子・擬原子軌道で分解されたスピネクスタの解析機能も「kSpin」を提供します。この機能により、スピンスplitにおいてどの原子・軌道が中心的役割を担うか、理解することができます。「kSpin」はラシュバ効果のみならずトポロジカル絶縁体や非ラシュバ型のスピンスplitにも応用できることにも注意して下さい。

以下に続く節では、計算例を示しながら、それらの解析に「kSpin」がどのように応用できるのか説明します。「kSpin」はノンコリニア計算の場合のみに使用可能で、コリニア計算には対応していないことに留意してください。ノンコリニア計算に関しては「ノンコリニア DFT」の章 34 を参照してください。

本機能を用いて発表を行う際には、文献 [78] を引用して頂けますと幸いです。また、本機能の実装に関する技術ノートが [http://www.openmx-square.org/tech\\_notes/note\\_kSpin-1.0.pdf](http://www.openmx-square.org/tech_notes/note_kSpin-1.0.pdf) にありますので、参照して下さい。

### 53.1 概要

本節では Au(111) 表面の簡単なモデルを用いて、スピネクスタの計算方法を説明します。以下の 2 もしくは 3 ステップの手順により、スピネクスタ及び  $k$  空間密度行列は解析されます。

#### 1. SCF 計算

ディレクトリ「work」に収容された入力ファイル「Au111Surface\_FL.dat」を用いて通常の SCF 計算を実行してください。そして、以下のキーワード「scf.SpinPolarization」と「HS.fileout」を以下のようそれぞれ「NC」と「ON」に設定して下さい。

```
scf.SpinPolarization      NC      # On|Off|NC
HS.fileout                 ON      # on|off, default=Off
```

また、スピネクスタを解析する場合にはキーワード「scf.SpinOrbit.Coupling」を「ON」に設定して下さい。

```
scf.SpinOrbit.Coupling    ON      # On|Off
```

正常に計算が完了すると、出力ファイル「Au111Surface.scfout」がディレクトリ「work」に生成されます。またバンド図を作成するためのキーワードを設定することで、本モデル Au(111) 表面のバンド構造が得られます (図 64 を参照)。

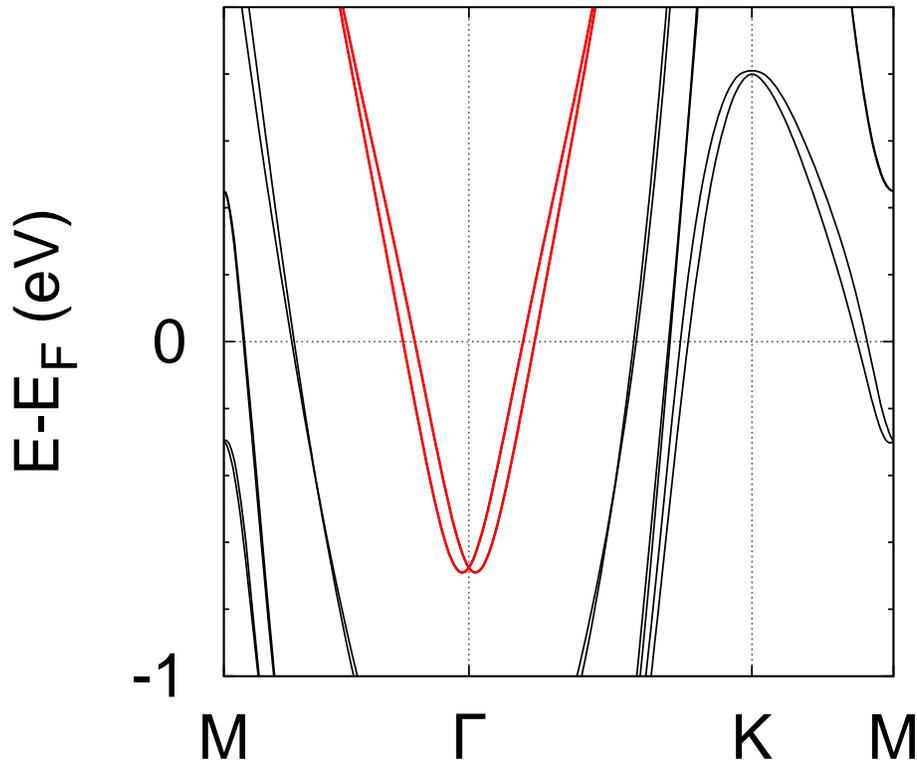


Figure 64: Au(111) 表面での Rashba スピン分裂を示すバンド構造。この図はバンドファイル「Au111Surface.Band」により得られました (バンド分散の 19 章を参照してください)。赤色の曲線は Rashba バンドを示し、バンドインデックスの 55 と 56 に相当します。Rashba バンドの色付けは GNUBAND ファイル「Au111Surface.GNUBAND」を変更するか、OMXTool [146] を利用することで実現できます。

## 2. スピントクスチャと k 空間スピン密度行列の計算

ここでは図 64 に示すスピン分裂した Rashba バンドの解析を行います。ポストプロセスコード「kSpin」により、スピントクスチャと k 空間スピン密度行列が計算できます。まず、ディレクトリ「source」中で以下のようにコンパイルすることで実行ファイルが得られます。

```
% make kSpin
```

コンパイルに成功すると、実行ファイル「kSpin」がディレクトリ「work」中に生成されます。次に、ディレクトリ「work」に移動し、以下のように実行してください。

```
% ./kSpin Au111Surface_FL.dat
```

入力ファイルにおいて「kSpin」への適切なキーワードを設定する必要があります。必要なキーワードに関しては後ほど説明します。「kSpin」には k 空間スピン密度行列を計算するために四つの方法が実装されています。*FermiLoop*, *GridCalc*, *BandDispersion* そして *MulPOnly* です。この四つの方法の詳細はそれぞれの節で説明されています。ここでの例「Au111Surface\_FL.dat」は *FermiLoop* に対するものであり、*FermiLoop* の 53.2 節にその解析手順が説明されています。

### 3. (オプション) k 空間スピン密度行列の解析

k 空間スピン密度行列及びその原子分解を行う解析機能が実装されています。原子分解から進んでさらに  $s$ ,  $p$ ,  $d$ , または  $f$  軌道の成分にも分解可能です。原子・擬原子軌道に分解された k 空間スピン密度行列を解析するためにポストプロセスコード「MulPCalc」が利用できます。本解析機能に関しては *MulPCalc* の 53.6 節を参照して下さい。

#### 53.2 FermiLoop: 一定エネルギー面での計算

ポストプロセスコード「kSpin」には k 空間スピン密度行列を計算する 4 つの方法があります。これはキーワード「Calc.Type」で指定します。ここでは 4 つの方法のうち、*FermiLoop* を説明します。*FermiLoop* はスピントクスチャと k 空間スピン密度行列を一定エネルギー面上 (例えばフェルミ面) で計算します。*FermiLoop* を用いたスピントクスチャの解析例として Au(111) 表面のモデルをここで説明します。*FermiLoop* は二つの段階で k 点を探索します。第一段階では計算すべきバンドを探すために大域的な探索を実施します。第二段階では三角メッシュを用いて一定エネルギー面上の k 点を特定します。

前節で議論した「Au111Surface\_FL.dat」を用いて OpenMX の SCF 計算を実行した後で「kSpin」の計算を行います。「kSpin」の実行に関連するキーワードを以下に列挙します。これらのキーワードは入力ファイル「Au111Surface\_FL.dat」中に記載されています。また入力ファイル「Au111Surface\_FL.dat」はディレクトリ「work」中に収容されています。

#### kSpin に関連するキーワードの一覧

|                  |                     |  |
|------------------|---------------------|--|
| Filename.scfout  | Au111Surface.scfout |  |
| Filename.outdata | Au111Surface_FL     |  |
| Calc.Type        | FermiLoop           | # FermiLoop, GridCalc,<br>BandDispersion, or MulPOnly<br>default: MulPOnly |
| Energy.Range     | 0.0 0.0             | # eV; default: 0.0 0.0   |
| Search.kCentral  | 0.0 0.0 0.0         | # default: 0.0 0.0 0.0   |
| Calc.Type.3mesh  | 2                   | # default: 1   |
| kRange.3mesh     | 0.2 0.2             | # default: 0.5 0.5   |
| k-plane.1stStep  | 21 21               | # default: 2 2   |
| k-plane.2ndStep  | 3 3                 | # default: 3 3   |
| Eigen.Brent      | On                  | # on off, default: On  |
| Trial.Brent      | 5                   | # default: 5   |
| Calc.Bandbyband  | Off                 | # on off, default: Off   |
| Calc.Band.Min    | 55                  |  |
| Calc.Band.Max    | 56                  |  |
| MulP.Vec.Scale   | 0.1 0.1 0.1         | # default: 1.0 1.0 1.0   |

#### キーワードの仕様

それぞれのキーワードの仕様を以下に説明します。

## Filename.scfout

「kSpin」によって読み込まれる scfout ファイルの名前を指定します。

## Filename.outdata

出力ファイルの名前を指定します。このキーワードは OpenMX 計算のキーワード「System.Name」に相当します。

## Calc.Type

*FermiLoop*, *GridCalc*, *BandDispersion*, もしくは *MulPOnly* のいずれかを選択します。デフォルトの設定は *MulPOnly* です。ここでは *FermiLoop* を選択します。

## Energy.Range

解析を実施するバンドを探索するためのエネルギー範囲を指定します。*FermiLoop* では上限値と下限値に対して同一の値を指定します。単位は eV です。もし異なる値が設定されると、それらの平均値が使用されます。デフォルト値は「-0.5 0.5」であり、これはフェルミ面に対応します。

## Search.kCentral

ある中心 k 点の周辺で一定エネルギー値を持つ k 点の探索が実施されます。本キーワードによりこの中心 k 点の指定を行います。一定エネルギー値はキーワード「Energy.Range」で指定します。k 点の指定方法はキーワード「Band.kpath」に従います。デフォルト値は「0.0 0.0 0.0」(つまり  $\Gamma$  点) です。

## Calc.Type.3mesh

スピントクスチャが計算される面を指定します。'1', '2', '3' の値はそれぞれ  $k_a k_b$ ,  $k_b k_c$ ,  $k_c k_a$  面に対応します。デフォルト値は 1 です。

## kRange.3mesh

k 点が計算される逆空間の二次元領域を二つの値で指定します。例えば、キーワード「Calc.Type.3mesh」の値が 1 ( $k_a k_b$  面) の際には、値「0.2 0.3」は  $-0.2 \leq k_a \leq 0.2$ ,  $-0.3 \leq k_b \leq 0.3$  の領域を指定します。k 点の指定の表記はキーワード「Band.kpath」に従います。デフォルト値は「0.5 0.5」(つまり第一ブリルアンゾーンの全体) です。

## k-plane.1stStep

キーワード「kRange.3mesh」で指定された領域を分割する格子点の数を与えます。これは第一段階の探索において用いられます。例えば、キーワード「Calc.Type.3mesh」の値が 1 ( $k_a k_b$  面) ならば、値「2 3」は  $k_a$  軸方向に 2 点の格子点、 $k_b$  軸方向に 3 点の格子点で分割することを意味します。もし、値が「1 1」でキーワード「Calc.Bandbyband」の値が「ON」の際には、*FermiLoop* は第一段階を省略します。これは多数の MPI プロセスを用いた大規模計算において有用です。これに関連して k-plane.2ndStep; Calc.Bandbyband; Calc.BandMin; Calc.BandMax も参照してください。デフォルト値は「2 2」です。

## k-plane.2ndStep

第二段階の探索における格子点の数を指定します。これは第一段階で分割された領域のうち、計算に必要な領域を三角メッシュを用いてさらに細かく分割します。表記はキーワード「k-plane.1stStep」と同一です。これに関連して「k-plane.1stStep」も参照してください。デフォルト値は「3 3」です。

## Eigen.Brent

指定された一定エネルギー面上 (つまり、キーワード「Energy.Range」の値) の k 点を探索するため *FermiLoop* に Brent 法を用いるなら「ON」を、用いない場合には「OFF」を指定します。もし値が「OFF」ならば、*FermiLoop* は線型補間法を用います。デフォルト値は「ON」です。これに関連して「Trial.Brent」も参照してください。

#### Trial.Brent

Brent 法 (*Eigen.Brent* を参照してください) のステップ数の最大値を指定します。キーワード「Eigen.Brent」の値が「ON」の際にこのキーワードは有効となります。デフォルト値は5です。これに関連して「Eigen.Brent」も参照してください。

#### Calc.Bandbyband

*FermiLoop* で与えられたバンドを計算させる場合は「ON」、さもなければ「OFF」を指定します (*Calc.BandMin*; *Calc.BandMax* を参照してください)。デフォルト値は「OFF」です。

#### Calc.BandMin

バンドインデックスを指定することで計算するバンドの下限値を指定します。キーワード「Calc.Bandbyband」の値が ON の場合にこのキーワードは有効です (*Calc.Bandbyband*; *Calc.BandMax* を参照してください)。バンドインデックスは *OMXTool* [146] を利用するか、または *BandDispersion* の機能 (章 53.4) により確認できます。

#### Calc.BandMax

バンドインデックスを指定することで計算するバンドの上限値を指定します。キーワード「Calc.Bandbyband」の値が ON の場合にこのキーワードは有効です (*Calc.Bandbyband*; *Calc.BandMax* を参照してください)。バンドインデックスは *OMXTool* [146] を利用するか、または *BandDispersion* の機能 (53.4 節) により確認できます。

#### MulP.Vec.Scale

スピネクスタを表現するベクトルを描く際の縮尺を指定します。例えば、値「0.1 0.2 0.3」は  $x$  軸に 0.1、 $y$  軸に 0.2、 $z$  軸に 0.3 の縮尺を指定します。このキーワードは「XXXXX.Pxyz\_YY」のみに影響します。(XXXXX はキーワード「Filename.outdata」の値、YY はバンドインデックス)。デフォルト値は「1.0 1.0 1.0」です。

## 計算

ポストプロセスコード「kSpin」によりスピネクスタ及び k 空間スピン密度行列が計算されます。ここではポストプロセスコード「kSpin」はディレクトリ「work」にあると仮定して説明を行います。ディレクトリ「work」に移動して、計算を以下のとおり実行します。

```
% ./kSpin Au111Surface_FL.dat
```

または、MPI 計算では、例えば、4MPI プロセスの場合は以下のとおり実行します。

```
% mpirun -np 4 ./kSpin Au111Surface_FL.dat
```

計算の進行に伴い、以下の標準出力が得られます。

```
*****
*****
kSpin:
code for evaluating spin related properties
in momentum space of solid state materials.
Copyright (C), 2019,
Hiroki Kotaka, Naoya Yamaguchi and Fumiyuki Ishii.
This software includes the work that is distributed
in version 3 of the GPL (GPLv3).
```

```
Please cite the following article:
H. Kotaka, F. Ishii, and M. Saito,
Jpn. J. Appl. Phys. 52, 035204 (2013).
DOI: 10.7567/JJAP.52.035204.
```

```
*****
*****
```

```
Input filename is "Au111Surface.scfout"
```

```
Start "FermiLoop" Calculation (5).
```

```
##### ORBITAL DATA #####
ClaOrb_MAX[0]: 2
ClaOrb_MAX[1]: 8
Total Band (2*n): 124
Central ( 0.000000 0.000000 0.000000)
#####
```

```
##### EIGEN VALUE #####
The number of BANDS 2 ( 55-> 56)
##### CONTOUR CALC #####
k-height : 0 0.000000
The number of BANDS 2 ( 55-> 56)
l= 55, k_points: 139 (array: 277)
l= 56, k_points: 115 (array: 229)
Total MulP data: 254
```

```
#####
```

```
#####
```

```
##### CALC TIME #####
```

```

Total Calculation Time: 26.851349 (s)
  Eigen Value Calc: 4.277838 (s)
l= 55:   Contour Calc: 11.349228 (s)
        MulP Calc: 1.120379 (s)
l= 56:   Contour Calc: 9.179602 (s)
        MulP Calc: 0.920524 (s)
#####
##### CALC TIME #####
Total Calculation Time: 26.869150 (s)
#####

```

上記のとおり計算が正常に完了すると、ディレクトリ「work」に以下の出力ファイルが生成されます。

```

Au111Surface_FL.FermiSurf_53
Au111Surface_FL.Pxyz_53
Au111Surface_FL.FermiSurf_54
Au111Surface_FL.Pxyz_54
Au111Surface_FL.FermiSurf_55
Au111Surface_FL.Pxyz_55
Au111Surface_FL.FermiSurf_56
Au111Surface_FL.Pxyz_56
Au111Surface_FL.AtomMulP
Au111Surface_FL.MulP_s
Au111Surface_FL.MulP_p
Au111Surface_FL.MulP_p1
Au111Surface_FL.MulP_p2
Au111Surface_FL.MulP_p3
Au111Surface_FL.MulP_d
Au111Surface_FL.MulP_d1
Au111Surface_FL.MulP_d2
Au111Surface_FL.MulP_d3
Au111Surface_FL.MulP_d4
Au111Surface_FL.MulP_d5
Au111Surface_FL.plotexample
Au111Surface_FL.atominfo
temporal_12345.input

```

例として、以下のコマンドを実行すると、図 65 に示す Au(111) 表面の Rashba スピン分裂によるスピントクスチャが得られます。これは典型的な Rashba 型のスピントクスチャです。

```
% gnuplot Au111Surface_FL.plotexample
```

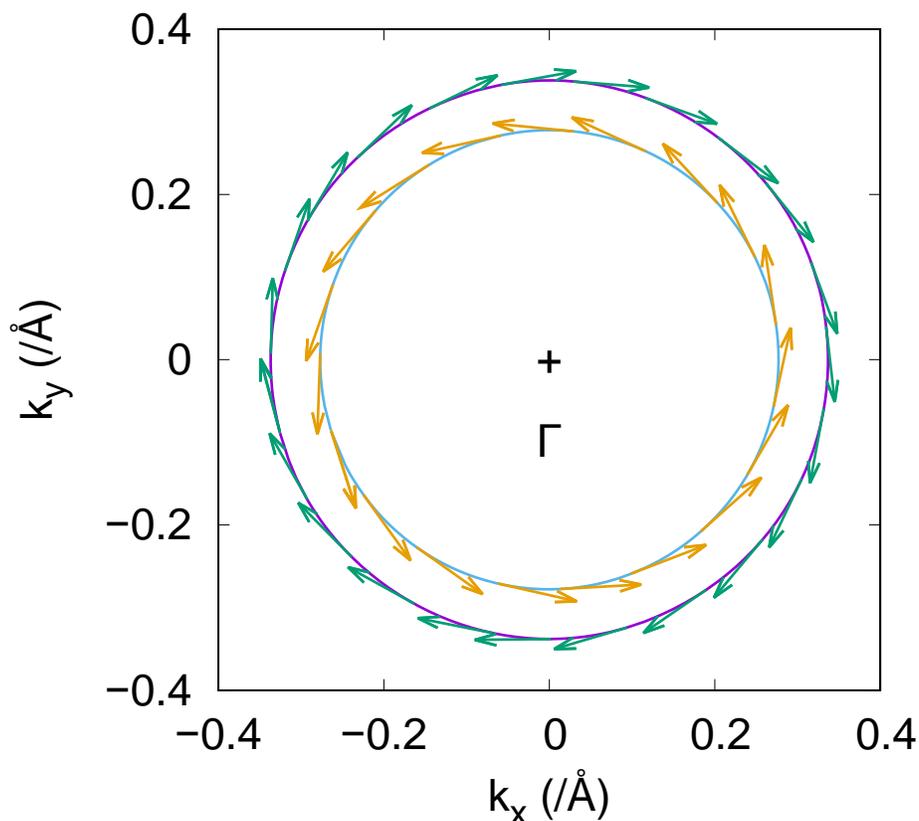


Figure 65: Au(111) 表面の  $\Gamma$  点周りでの Rashba スピン分裂を示すスピネクスタチャ。外側と内側はバンドインデックス 55 と 56 に対応。ベクトルは Pxyz\_YY ファイル「Au111Surface\_FL.Pxyz\_55」と「Au111Surface\_FL.Pxyz\_56」に記録された各  $k$  点でのパウリ行列固有ベクトルによる  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  の期待値。閉じた曲線は FermiSurf\_YY ファイル「Au111Surface\_FL.FermiSurf\_55」と「Au111Surface\_FL.FermiSurf\_56」のそれぞれに記録された一定エネルギー面。中心の十字点は  $\Gamma$  に対応。この図はプロット例のファイル「Au111Surface\_FL.plotexample」より作成。

## 出力ファイル

各出力ファイルの内容を以下に説明します。

### FermiSurf\_YY ファイル

このファイルは指定された一定エネルギー面上で探索されたバンド（バンドインデックス YY）の  $k$  点データを記録します。第一、第二、第三列はそれぞれ  $k_x$ ,  $k_y$ ,  $k_z$  成分に対応します。単位は  $\text{Bohr}^{-1}$  です。第四、第五列はそれぞれバンドインデックス、エネルギー（eV 単位）です。

### Pxyz\_YY ファイル

このファイルは FermiSurf\_YY ファイルに記録された各  $k$  点でのパウリ行列ベクトルの期待値のデータを記録します。第一、第二、第三列はそれぞれ  $k_x$ ,  $k_y$ ,  $k_z$  成分に対応します。単位は  $\text{Bohr}^{-1}$  です。第四、第五列、第六列はそれぞれ  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  の期待値です。単位は Bohr 磁子です。

### AtomMulP ファイル

このファイルは *MulPCalc* (53.6 節) により解析された原子寄与で分解された  $k$  空間スピン密度行列のデータを記録します。

#### MulP\_xx ファイル

このファイルは原子寄与で分解された  $k$  空間スピン密度行列の  $xx$  成分のデータを記録します。このファイルは *MulPCalc*53.6 で解析できます。

#### plotexample ファイル

このファイルは gnuplot スクリプトの例を提供します。

#### atominfo ファイル

このファイルは格子ベクトルと PAO の情報を提供します。

#### temporal\_12345.input

このファイルは scfout ファイルに記録された入力ファイルのコピーです。

### (オプション)k 空間スピン密度行列の解析

*MulPCalc* を用いて AtomMulP ファイルまたは MulP\_xx ファイルから  $k$  空間スピン密度行列を解析するためのデータが抽出できます。*MulPCalc* の入力ファイルの例が入力ファイル「Au111Surface\_FL.dat」の下部にあります。上記の計算後に  $k$  空間スピン密度行列の解析を以下の手順で行います。

```
% ./MulPCalc Au111Surface_FL.dat
```

*MulPCalc* に関するさらなる情報は 53.6 節も参照してください。

### その他の留意点

*FermiLoop* に対する適切なパラメーター設定を行うために、*GridCalc* が利用できます。*GridCalc* では逆空間内のより広い領域の探索に基づきスピネクスタチャを描くため、その結果を踏まえて *FermiLoop* の適切なパラメーターが設定できます。またキーワード「Eigen.Brent」を「OFF」に設定すると計算時間が短縮できます。その際には、細かいメッシュを用いて精度を向上させるためにキーワード「k-plane.1stStep」または「k-plane.2ndStep」の値を増加させて下さい。

### 53.3 GridCalc: $k$ 点格子での計算

ポストプロセスコード「kSpin」には  $k$  空間スピン密度行列を計算する 4 つの方法があります。これはキーワード「Calc.Type」で指定します。ここでは 4 つの方法のうち、*GridCalc* を説明します。*GridCalc* は指定  $k$  点格子上で各バンドのスピネクスタチャと  $k$  空間スピン密度行列を計算します。バンドはユーザー指定のエネルギー範囲内で行われます。エネルギー範囲を指定する代わりに計算するバンドを指定することもできます。*GridCalc* を用いたスピネクスタチャの解析例として Au(111) 表面のモデルをここで説明します。まず、ディレクトリ「work」中に収容された入力ファイル「Au111Surface\_GC.dat」を用いて OpenMX 計算を実行します。次にポストプロセスコード「kSpin」の計算を実行します。「kSpin」の実行に関連するキーワードを以下に列挙します。これらのキーワードは入力ファイル「Au111Surface\_GC.dat」中に記載されています。また入力ファイル「Au111Surface\_GC.dat」はディレクトリ「work」中に収容されています。

## kSpin に関連するキーワードの一覧

|                  |                     |  |
|------------------|---------------------|--|
| Filename.scfout  | Au111Surface.scfout |  |
| Filename.outdata | Au111Surface_GL     |  |
| Calc.Type        | GridCalc            | # FermiLoop, GridCalc,<br>BandDispersion, or MulPOnly<br>default: MulPOnly |
| Energy.Range     | -1.0 1.0            | # eV; default: 0.0 0.0   |
| Search.kCentral  | 0.0 0.0 0.0         | # default: 0.0 0.0 0.0   |
| Calc.Type.3mesh  | 2                   | # default: 1   |
| kRange.3mesh     | 0.5 0.5             | # default: 0.5 0.5   |
| k-plane.1stStep  | 14 14               | # default: 2 2   |
| Calc.Bandbyband  | Off                 | # default: Off   |
| Calc.Band.Min    | 55                  |  |
| Calc.Band.Max    | 56                  |  |
| MulP.Vec.Scale   | 0.1 0.1 0.1         | # default: 1.0 1.0 1.0   |

### キーワードの仕様

上記のキーワードは *FermiLoop* の場合と同一ですが、*GridCalc* の観点から各キーワードの仕様を以下に説明します。キーワード「Energy.Range」の振る舞いは *FermiLoop* でのそれと異なることに注意してください。

#### Filename.scfout

「kSpin」によって読み込まれる scfout ファイルの名前を指定します。

#### Filename.outdata

出力ファイルの名前を指定します。このキーワードは OpenMX 計算のキーワード「System.Name」に相当します。

#### Calc.Type

*FermiLoop*, *GridCalc*, *BandDispersion*, もしくは *MulPOnly* のいずれかを選択します。デフォルトの設定は *MulPOnly* です。ここでは *GridCalc* を選択します。

#### Energy.Range

解析を実施するバンドを探索するためのエネルギー範囲を指定します。単位は eV です。キーワード「Calc.Bandbyband」が「OFF」の場合、このキーワードが有効となります (Calc.Bandbyband を参照してください)。デフォルト値は「-0.5 0.5」です。つまりエネルギーの下限値が-0.5 (eV)、上限値は 0.5 (eV) となります。

#### Search.kCentral

ある中心 k 点の周辺で「Energy.Range」で指定されたエネルギー範囲内にある k 点の探索が実施されます。本キーワードによりこの中心 k 点の指定を行います。k 点の指定方法はキーワード「Band.kpath」に従います。デフォルト値は「0.0 0.0 0.0」(つまり  $\Gamma$  点) です。

### Calc.Type.3mesh

スピントクスチャが計算される面を指定します。'1', '2', '3' の値はそれぞれ  $k_a k_b$ ,  $k_b k_c$ ,  $k_c k_a$  面に対応します。デフォルト値は 1 です。

### kRange.3mesh

k 点が計算される逆空間の二次元領域を二つの値で指定します。例えば、キーワード「Calc.Type.3mesh」の値が 1 ( $k_a k_b$  面) の際には、値「0.2 0.3」は  $-0.2 \leq k_a \leq 0.2$ ,  $-0.3 \leq k_b \leq 0.3$  の領域を指定します。k 点の指定の表記はキーワード「Band.kpath」に従います。デフォルト値は「0.5 0.5」(つまり第一ブリルアンゾーンの全体) です。

### k-plane.1stStep

キーワード「kRange.3mesh」で指定された領域を分割する格子点の数を与えます。これは第一段階の探索において用いられます。例えば、キーワード「Calc.Type.3mesh」の値が 1 ( $k_a k_b$  面) ならば、値「2 3」は  $k_a$  軸方向に 2 点の格子点、 $k_b$  軸方向に 3 点の格子点で分割することを意味します。デフォルト値は「2 2」です。

### Calc.Bandbyband

*GridCalc* に与えられたバンドを計算させるなら「ON」、さもなければ「OFF」を指定します (Calc.BandMin; Calc.BandMax も参照してください)。デフォルト値は「OFF」です。

### Calc.BandMin

バンドインデックスを指定することで計算するバンドの範囲の下限值を指定します。キーワード「Calc.Bandbyband」の値が ON の場合にこのキーワードは有効です (Calc.Bandbyband; Calc.BandMax を参照してください)。バンドインデックスは OMXTool [146] を利用するか、または BandDispersion の機能 (章 53.4) により確認できます。

### Calc.BandMax

バンドインデックスを指定することで計算するバンドの範囲の上限値を指定します。キーワード「Calc.Bandbyband」の値が ON の場合にこのキーワードは有効です (Calc.Bandbyband; Calc.BandMax を参照してください)。バンドインデックスは OMXTool [146] を利用するか、または BandDispersion の機能 (章 53.4) により確認できます。

### MulP.Vec.Scale

スピントクスチャを表現するベクトルを描く際の縮尺を指定します。例えば、値「0.1 0.2 0.3」は  $x$  軸に 0.1、 $y$  軸に 0.2、 $z$  軸に 0.3 の縮尺を指定します。このキーワードは「XXXXX.Pxyz\_YY」のみに影響します。(XXXXX はキーワード「Filename.outdata」の値、YY はバンドインデックス)。デフォルト値は「1.0 1.0 1.0」です。

## 計算

ポストプロセスコード「kSpin」によりスピントクスチャ及び k 空間スピンドensity行列が計算されます。ここではポストプロセスコード「kSpin」はディレクトリ「work」にあると仮定して説明を行います。ディレクトリ「work」に移動して、計算を以下のとおり実行します。

```
% ./kSpin Au111Surface_GC.dat
```

または、MPI 計算では、例えば、4MPI プロセスの場合は以下のとおり実施します。

```
% mpirun -np 4 ./kSpin Au111Surface_GC.dat
```

計算の進行に伴い、以下の標準出力が得られます。

```
*****  
*****
```

```
kSpin:  
code for evaluating spin related properties  
in momentum space of solid state materials.  
Copyright (C), 2019,  
Hiroki Kotaka, Naoya Yamaguchi and Fumiyuki Ishii.  
This software includes the work that is distributed  
in version 3 of the GPL (GPLv3).
```

```
Please cite the following article:  
H. Kotaka, F. Ishii, and M. Saito,  
Jpn. J. Appl. Phys. 52, 035204 (2013).  
DOI: 10.7567/JJAP.52.035204.
```

```
*****  
*****
```

```
Input filename is "Au111Surface.scfout"
```

```
Start "GridCalc" Calculation (4).
```

```
##### ORBITAL DATA #####
```

```
ClaOrb_MAX[0]: 2  
ClaOrb_MAX[1]: 8  
Total Band (2*n): 124  
Central ( 0.000000 0.000000 0.000000)
```

```
#####
```

```
##### EIGEN VALUE #####
```

```
The number of BANDs 8 ( 49-> 56)  
Total MulP data:1568
```

```
#####
```

```
##### CALC TIME #####
```

```
Total Calculation Time: 3.656405 (s)  
Eigen Value Calc: 3.498849 (s)
```

```
#####  
##### CALC TIME #####  
Total Calculation Time: 3.670708 (s)  
#####
```

上記のとおり計算が正常に完了すると、ディレクトリ「work」に以下の出力ファイルが生成されます。

```
Au111Surface_GC.EigenMap_49  
Au111Surface_GC.Pxyz_49  
Au111Surface_GC.plotexample_49  
Au111Surface_GC.EigenMap_50  
Au111Surface_GC.Pxyz_50  
Au111Surface_GC.plotexample_50  
Au111Surface_GC.EigenMap_51  
Au111Surface_GC.Pxyz_51  
Au111Surface_GC.plotexample_51  
Au111Surface_GC.EigenMap_52  
Au111Surface_GC.Pxyz_52  
Au111Surface_GC.plotexample_52  
Au111Surface_GC.EigenMap_53  
Au111Surface_GC.Pxyz_53  
Au111Surface_GC.plotexample_53  
Au111Surface_GC.EigenMap_54  
Au111Surface_GC.Pxyz_54  
Au111Surface_GC.plotexample_54  
Au111Surface_GC.EigenMap_55  
Au111Surface_GC.Pxyz_55  
Au111Surface_GC.plotexample_55  
Au111Surface_GC.EigenMap_56  
Au111Surface_GC.Pxyz_56  
Au111Surface_GC.plotexample_56  
Au111Surface_GC.AtomMulP  
Au111Surface_GC.MulP_s  
Au111Surface_GC.MulP_p  
Au111Surface_GC.MulP_p1  
Au111Surface_GC.MulP_p2  
Au111Surface_GC.MulP_p3  
Au111Surface_GC.MulP_d  
Au111Surface_GC.MulP_d1  
Au111Surface_GC.MulP_d2  
Au111Surface_GC.MulP_d3  
Au111Surface_GC.MulP_d4  
Au111Surface_GC.MulP_d5
```

```
Au111Surface_GC.atominfo
temporal_12345.input
```

例として、以下のコマンドを実行すると、図 66(a) と (b) に示す Au(111) 表面のラシュバスピンスピン分裂のスピントクスチャが得られます。これは典型的な Rashba 型のスピントクスチャです。

```
% gnuplot Au111Surface_GC.plotexample_55
% gnuplot Au111Surface_GC.plotexample_56
```

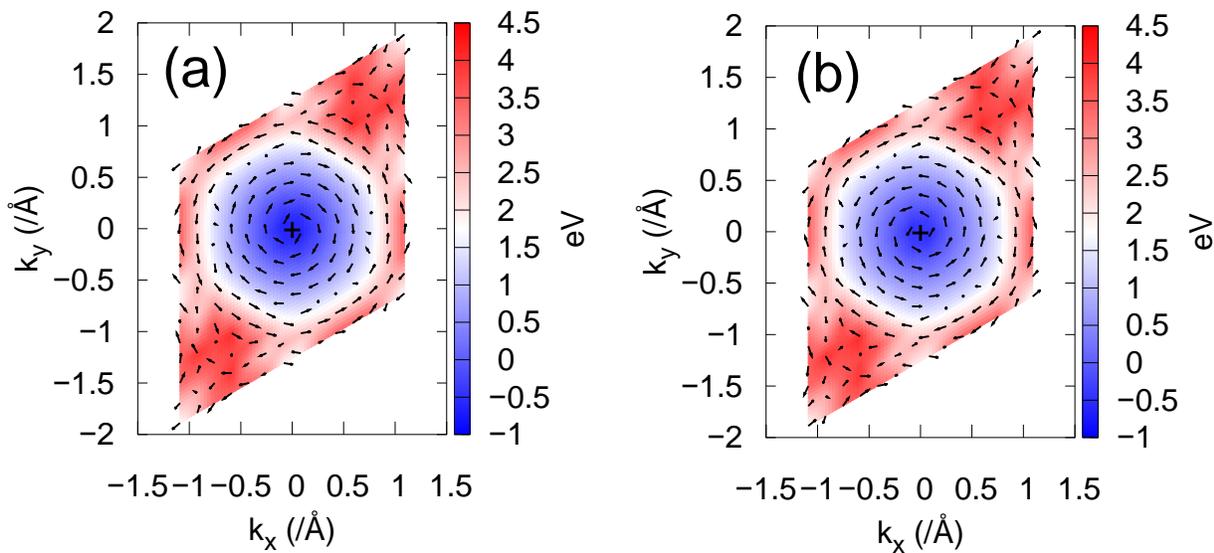


Figure 66: Au(111) 表面の  $\Gamma$  点周りでのラシュバスピンスピン分裂を示すスピントクスチャ。(a) バンドインデックス 55 に対応。(b) バンドインデックス 56 に対応。ベクトルは Pxyz\_YY ファイル「Au111Surface\_GC.Pxyz\_55」と「Au111Surface\_GC.Pxyz\_56」に記録された各  $k$  点でのパウリ行列固有ベクトルによる  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  の期待値。カラーマップは EigenMap\_YY ファイル「Au111Surface\_GC.EigenMap\_55」と「Au111Surface\_GC.EigenMap\_56」のそれぞれに記録された  $k$  点でのエネルギーでの高さ。中心の十字点は  $\Gamma$  に対応。この図はプロット例のファイル「Au111Surface\_GC.plotexample\_55」と「Au111Surface\_GC.plotexample\_56」により作成。

## 出力ファイル

各出力ファイルの内容を以下に説明します。

### EigenMap\_YY ファイル

このファイルはバンド (バンドインデックス YY) の  $k$  点データを記録します。第一、第二、第三列はそれぞれ  $k_x$ ,  $k_y$ ,  $k_z$  成分に対応します。単位は  $\text{Bohr}^{-1}$  です。第四列はエネルギー (eV 単位) です。

### Pxyz\_YY ファイル

このファイルは EigenMap\_YY ファイルに記録された各  $k$  点でのパウリ行列ベクトルの期待値のデータ

を記録します。第一、第二、第三列はそれぞれ  $k_x$ ,  $k_y$ ,  $k_z$  成分に対応します。単位は Bohr<sup>-1</sup> です。第四、第五列、第六列はそれぞれ  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  の期待値です。単位は Bohr 磁子です。

#### AtomMulP ファイル

このファイルは *MulPCalc* (53.6 節) により解析された原子寄与で分解された k 空間スピン密度行列のデータを記録します。

#### MulP\_xx ファイル

このファイルは原子寄与で分解された k 空間スピン密度行列の  $xx$  成分のデータを記録します。このファイルは *MulPCalc*53.6 で解析できます。

#### plotexample ファイル

このファイルは gnuplot スクリプトの例を提供します。

#### atominfo ファイル

このファイルは格子ベクトルと PAO の情報を提供します。

#### temporal\_12345.input

このファイルは scfout ファイルに記録された入力ファイルのコピーです。

### (オプション)k 空間スピン密度行列の解析

*MulPCalc* を用いて AtomMulP ファイルまたは MulP\_xx ファイルから k 空間スピン密度行列を解析するためのデータが抽出できます。*MulPCalc* の入力ファイルの例が入力ファイル「Au111Surface\_GC.dat」の下部にあります。上記の計算後に k 空間スピン密度行列の解析を以下の手順で行います。

```
% ./MulPCalc Au111Surface_GC.dat
```

*MulPCalc* に関するさらなる情報は 53.6 節も参照してください。

## 53.4 BandDispersion: バンド分散関係の計算

ポストプロセスコード「kSpin」には k 空間スピン密度行列を計算する 4 つの方法があります。これはキーワード「Calc.Type」で指定します。ここでは 4 つの方法のうち、*BandDispersion* を説明します。*BandDispersion* の機能を用いて、バンド分散関係上で各原子に分解された k 空間スピン密度行列をユーザー指定のエネルギー範囲内で計算できます。*BandDispersion* を用いたスピントクスチャの解析例として Au(111) 表面のモデルをここで説明します。まず、ディレクトリ「work」中に収容された入力ファイル「Au111Surface\_BD.dat」を用いて OpenMX 計算を実行します。次に、*MulPCalc*53.6 の機能を用いてスピントクスチャを計算できます。「kSpin」の実行に関連するキーワードを以下に列挙します。これらのキーワードは入力ファイル「Au111Surface\_BD.dat」中に記載されています。また入力ファイル「Au111Surface\_BD.dat」はディレクトリ「work」中に収容されています。

### kSpin に関連するキーワードの一覧

|                  |                     |
|------------------|---------------------|
| Filename.scfout  | Au111Surface.scfout |
| Filename.outdata | Au111Surface_BD     |

```

Calc.Type          BandDispersion  # FermiLoop, GridCalc,
                   BandDispersion, or MulPOnly
                   default: MulPOnly
Energy.Range       -1.0  1.0          # eV; default: 0.0  0.0
Band.Nkpath        2
<Band.kpath
  135  0.0  0.500000  0.000000  0.0  0.000000  0.000000  M  G
  135  0.0  0.000000  0.000000  0.0 -0.500000  0.000000  G -M
Band.kpath>

```

## キーワードの仕様

それぞれのキーワードの仕様を以下に説明します。

### Filename.scfout

「kSpin」によって読み込まれる scfout ファイルの名前を指定します。

### Filename.outdata

出力ファイルの名前を指定します。このキーワードは OpenMX 計算のキーワード「System.Name」に相当します。

### Calc.Type

*FermiLoop*, *GridCalc*, *BandDispersion*, もしくは *MulPOnly* のいずれかを選択します。デフォルトの設定は *MulPOnly* です。ここでは *BandDispersion* を選択します。

### Energy.Range

解析を実施するバンドを探索するためのエネルギー範囲を指定します。単位は eV です。デフォルト値は「-0.5 0.5」です。つまりエネルギーの下限値が-0.5 (eV)、上限値は 0.5 (eV) となります。

### Band.Nkpath

バンド分散を計算する k 点のパスの数を指定します。この表記は通常の OpenMX の計算と同一です。

### Band.kpath

バンド分散を計算する k 点のパスを指定します。この表記は通常の OpenMX 計算と同一です。

## 計算

ポストプロセスコード「kSpin」により原子ごとに分解された k 空間スピン密度行列が計算されます。ここではポストプロセスコード「kSpin」はディレクトリ「work」にあると仮定して説明を行います。ディレクトリ「work」に移動して、計算を以下のとおり実行します。

```
% ./kSpin Au111Surface_BD.dat
```

または、MPI 計算では、例えば、4MPI プロセスの場合は以下のとおり実行します。

```
% mpirun -np 4 ./kSpin Au111Surface_BD.dat
```

計算の進行に伴い、以下の標準出力が得られます。

```
*****  
*****
```

```
kSpin:  
code for evaluating spin related properties  
in momentum space of solid state materials.  
Copyright (C), 2019,  
Hiroki Kotaka, Naoya Yamaguchi and Fumiyuki Ishii.  
This software includes the work that is distributed  
in version 3 of the GPL (GPLv3).
```

```
Please cite the following article:  
H. Kotaka, F. Ishii, and M. Saito,  
Jpn. J. Appl. Phys. 52, 035204 (2013).  
DOI: 10.7567/JJAP.52.035204.
```

```
*****  
*****
```

```
Input filename is "Au111Surface.scfout"
```

```
Start "BandDispersion" Calculation (3).
```

```
line_Nk[1]: 0.665829  
( 0.000000, 0.500000, 0.000000 ) -> ( 0.000000, 0.000000, 0.000000 )  
line_Nk[2]: 1.331658  
( 0.000000, 0.000000, 0.000000 ) -> ( 0.000000, -0.500000, 0.000000 )
```

```
##### ORBITAL DATA #####
```

```
ClaOrb_MAX[0]: 2  
ClaOrb_MAX[1]: 8  
Total Band (2*n): 124
```

```
#####
```

```
Band.Nkpath: 2  
135 ( 0.000000, 0.500000, 0.000000 ) >>> ( 0.000000, 0.000000, 0.000000 ) M G  
135 ( 0.000000, 0.000000, 0.000000 ) >>> ( 0.000000, -0.500000, 0.000000 ) G -M  
l_min: 49 l_max: 56 l_cal: 8
```

```
Au111Surface_BD.Band49_1  
Au111Surface_BD.Band50_1  
Au111Surface_BD.Band51_1  
Au111Surface_BD.Band52_1  
Au111Surface_BD.Band53_1  
Au111Surface_BD.Band54_1
```

```

Au111Surface_BD.Band55_1
Au111Surface_BD.Band56_1
l_min: 49 l_max: 56 l_cal: 8
Au111Surface_BD.Band49_2
Au111Surface_BD.Band50_2
Au111Surface_BD.Band51_2
Au111Surface_BD.Band52_2
Au111Surface_BD.Band53_2
Au111Surface_BD.Band54_2
Au111Surface_BD.Band55_2
Au111Surface_BD.Band56_2
#####
Total MulP data:2176
##### CALC TIME #####
Total Calculation Time: 4.772406 (s)
#####

```

上記のとおり計算が正常に完了すると、ディレクトリ「work」に以下の出力ファイルが生成されます。

```

Au111Surface_BD.BAND
Au111Surface_BD.Band49_1
Au111Surface_BD.Band50_1
Au111Surface_BD.Band51_1
Au111Surface_BD.Band52_1
Au111Surface_BD.Band53_1
Au111Surface_BD.Band54_1
Au111Surface_BD.Band55_1
Au111Surface_BD.Band56_1
Au111Surface_BD.Band49_2
Au111Surface_BD.Band50_2
Au111Surface_BD.Band51_2
Au111Surface_BD.Band52_2
Au111Surface_BD.Band53_2
Au111Surface_BD.Band54_2
Au111Surface_BD.Band55_2
Au111Surface_BD.Band56_2
Au111Surface_BD.AMulPBand
Au111Surface_BD.AMulPBand_s
Au111Surface_BD.AMulPBand_p
Au111Surface_BD.AMulPBand_p1
Au111Surface_BD.AMulPBand_p2
Au111Surface_BD.AMulPBand_p3
Au111Surface_BD.AMulPBand_d

```

```

Au111Surface_BD.AMulPBand_d1
Au111Surface_BD.AMulPBand_d2
Au111Surface_BD.AMulPBand_d3
Au111Surface_BD.AMulPBand_d4
Au111Surface_BD.AMulPBand_d5
Au111Surface_BD.plotexample
Au111Surface_BD.atominfo
temporal_12345.input

```

例として、以下のコマンドを実行すると、図 67(a) に示す Au(111) 表面のラッシュバースピン分裂のスピントクスチャが得られます。

```
% gnuplot Au111Surface_BD.plotexample
```

バンド分散でのスピントクスチャの情報を得るため、次のように解析を実行します。53.6 *MulPCalc*.

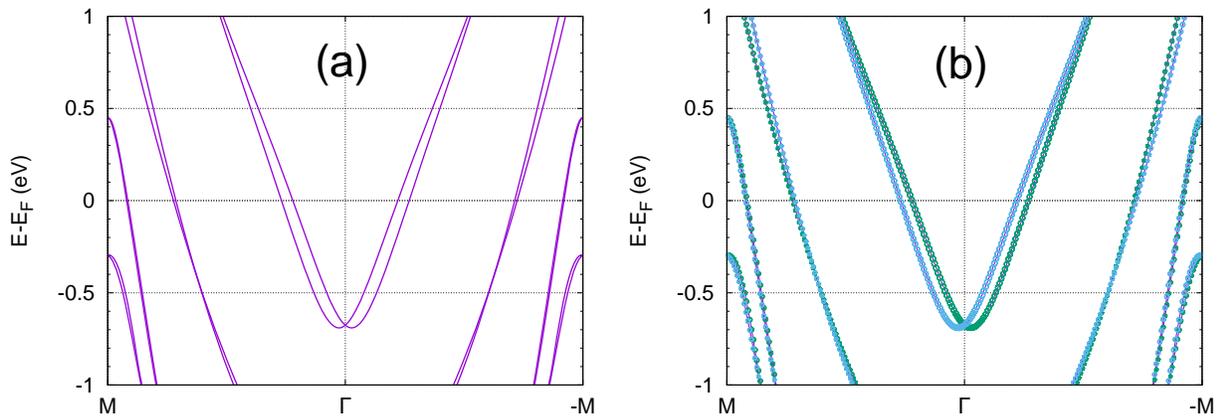


Figure 67: (a) Au(111) 表面のラッシュバースピン分裂を示すバンド分散。この図はバンドファイル「Au111Surface\_BD.BAND」とプロット例ファイル「Au111Surface\_BD.plotexample」により作成。(b) Au(111) 表面の各原子に分解された  $k$  空間スピン密度行列。この図は MulPop ファイル「Au111Surface\_BD\_MC.MulPop」により作成。円は各  $k$  点での  $\alpha$  と  $\beta$  のスピンの電子数の差を表す。半径と色はそれぞれその大きさと符合を反映。この図はプロット例のファイル「Au111Surface\_BD\_MC.plotexample」により作成。

## 出力ファイル

各出力ファイルの内容を以下に説明します。

### BAND ファイル

このファイルはバンド分散のデータを記録します。第一、第二列は  $k$  パス上の  $k$  点間の距離 (単位は  $\text{Bohr}^{-1}$ ) と固有エネルギー (単位は eV) に相当します。

### Band\_YY\_Z ファイル

このファイルは各ブランチ（バンドインデックス YY、k パスインデックス Z）のバンド分散のデータを記録します。このファイルの内容の表記は BAND ファイルと同一です。

#### AMulPBand ファイル

このファイルは原子ごとに分解された k 空間スピン密度行列のデータを記録します。そして、*MulPCalc*53.6 で解析できます。

#### AMulPBand\_xx ファイル

このファイルは原子ごとに分解された k 空間スピン密度行列の *xx* 成分のデータを記録します。このファイルは *MulPCalc*53.6 で解析できます。

#### plotexample ファイル

このファイルは gnuplot スクリプトの例を提供します。

#### atominfo ファイル

このファイルは格子ベクトルと PAO の情報を提供します。

#### temporal\_12345.input

このファイルは scfout ファイルに記録された入力ファイルのコピーです。

### (オプション) 原子ごとに分解された k 空間スピン密度行列の解析

*MulPCalc* を用いて AtomMulP ファイルまたは AMulPBand\_xx ファイルから原子ごとに分解された k 空間スピン密度行列を解析するためのデータが抽出できます。ディレクトリ「source」中で以下のようにコンパイルし、実行ファイルを作成して下さい。

```
% make MulPCalc
```

コンパイルが成功すると、実行ファイル「MulPCalc」がディレクトリ「work」に生成します。一例として、Au(111) 表面の Rashba スピン分裂のバンド分散に着目し、原子寄与に分解された k 空間スピン密度行列を解析します。最初に、以下のキーワードを入力ファイル「Au111Surface\_BD.dat」に追加します。

```
Filename.atomMulP Au111Surface_BD.AMulPBand # default: default
Filename.xyzdata Au111Surface_BD_MC # default: default
Num.of.Extract.Atom 3 # default: 1
Extract.Atom 1 2 3 # default: 1 2 ... (Num.of.Extract.Atom)
```

ポストプロセスコード「kSpin」による計算後に、原子寄与に分解された k 空間スピン密度行列を以下のように解析できます。

```
% ./MulPCalc Au111Surface_BD.dat
```

加えて、以下のキーワードを設定することで、図の見た目を調整することができます。

```
MulP.Vec.Scale 0.1 0.1 0.1 # default: 1.0 1.0 1.0
Data.Reduction 1 # default: 1
```

*MulPCalc* を実行後に、ディレクトリ「work」に以下の出力ファイルが得られます。

```
Au111Surface_BD_MC.MulPop
Au111Surface_BD_MC.MulPop49
Au111Surface_BD_MC.MulPop50
Au111Surface_BD_MC.MulPop51
Au111Surface_BD_MC.MulPop52
Au111Surface_BD_MC.MulPop53
Au111Surface_BD_MC.MulPop54
Au111Surface_BD_MC.MulPop55
Au111Surface_BD_MC.MulPop56
Au111Surface_BD_MC.plotexample
```

例として、以下のコマンドを実行すると、図 67(b) が得られます。これは原子寄与に分解された  $k$  空間スピン密度行列の情報を Au(111) 表面の Rashba スピン分裂バンド上に加味して表示したものです。

```
% gnuplot Au111Surface_BD_MC.plotexample
```

*MulPCalc* に関するさらなる情報は 53.6 節も参照してください。

### 53.5 MulPOnly: 利用者指定の $k$ 点での計算

ポストプロセスコード「kSpin」には  $k$  空間スピン密度行列を計算する 4 つの方法があります。これはキーワード「Calc.Type」で指定します。ここでは 4 つの方法のうち、*MulPOnly* を説明します。*MulPOnly* の機能を用いて指定する  $k$  点とバンドに対して  $k$  空間スピン密度行列を計算できます。*MulPOnly* と *MulPCalc* を用いたスピントクスチャの解析例として Au(111) 表面のモデルをここで説明します。まず、ディレクトリ「work」中に収容された入力ファイル「Au111Surface\_MO.dat」を用いて OpenMX 計算を実行します。次に、*MulPOnly* の機能を用いてスピントクスチャを計算できます。「kSpin」の実行に関連するキーワードを以下に列挙します。これらのキーワードは入力ファイル「Au111Surface\_MO.dat」中に記載されています。また入力ファイル「Au111Surface\_MO.dat」はディレクトリ「work」中に収容されています。

#### kSpin に関連するキーワードの一覧

|                     |                     |  |
|---------------------|---------------------|--|
| Filename.scfout     | Au111Surface.scfout |  |
| Filename.outdata    | Au111Surface_MO     |  |
| Calc.Type           | MulPOnly            | # FermiLoop, GridCalc,<br>BandDispersion, or MulPOnly<br>default: MulPOnly |
| Filename.kpointdata | kpoint.in           |  |

加えて、キーワード「Filename.kpointdata」で指定される名前のもう一つの入力ファイルが必要です。このファイルには  $k$  空間スピン密度行列を計算する  $k$  点を記述します。設定ファイルの例として、ディレクトリ「work」中にファイル「kpoint.in」が収容されています。そのフォーマットは以下の規則に従います。第一行に  $k$  空間スピン密度行列を計算する  $k$  点の数を指定します。次に、 $k$  点 (第一、第二、第三列がそれぞれ  $k_x, k_y, k_z$  に対応) とそのバンドインデックス (第四列) を指定します。逆空間の  $\Gamma$  点を中心とする円周上の  $k$  点で  $k$  空間スピン密度行列を計算する場合には、以下のコマンドを実行することで「kpoint.in」を生成できます。

```
rm -f kpoint.in && awk '{for (i=0; i<$1; i++){printf "%17.14f %17.14f %17.14f %d\n", 0, $2*cos(2*atan2(0, -1)/$1*i),
$2*sin(2*atan2(0, -1)/$1*i), $3 >> "buffer"; sum+=$1; if (!$1) {print sum; exit}}' -> N.in && cat N.in buffer >> kpoint.in && rm N.in buffer
```

その後で、次の値を入力して下さい。

```
40 0.18 55      <- # of k-points; radius; band index
40 0.15 56      <- # of k-points; radius; band index
0               <- exit
```

この手続きが正常に終了すると「kpoint.in」が生成されます。「kpoint.in」には以下の情報が保存されていますので、確認して下さい。

```
% cat kpoint.in
80
0.0000000000000000 0.1800000000000000 0.0000000000000000 55
0.0000000000000000 0.17778390130712 0.02815820370724 55
0.0000000000000000 0.17119017293313 0.05562305898749 55
0.0000000000000000 0.16038117435391 0.08171828995312 55
0.0000000000000000 0.14562305898749 0.10580134541265 55
0.0000000000000000 0.12727922061358 0.12727922061358 55
0.0000000000000000 0.10580134541265 0.14562305898749 55
0.0000000000000000 0.08171828995312 0.16038117435391 55
0.0000000000000000 0.05562305898749 0.17119017293313 55
0.0000000000000000 0.02815820370724 0.17778390130712 55
0.0000000000000000 0.0000000000000000 0.1800000000000000 55
0.0000000000000000 -0.02815820370724 0.17778390130712 55
...
..
```

## キーワードの仕様

それぞれのキーワードの仕様を以下に説明します。

### **Filename.scfout**

「kSpin」によって読み込まれる scfout ファイルの名前を指定します。

### **Filename.outdata**

出力ファイルの名前を指定します。このキーワードは OpenMX 計算のキーワード「System.Name」に相当します。

### **Calc.Type**

*FermiLoop*, *GridCalc*, *BandDispersion*, もしくは *MulPOnly* のいずれかを選択します。デフォルトの設定は *MulPOnly* です。ここでは *MulPOnly* を選択します。

### **Filename.kpointdata**

k 点とバンドインデックスのデータファイルの名前を指定します。

## 計算

ポストプロセスコード「kSpin」により原子ごとに分解された k 空間スピン密度行列が計算されます。ディレクトリ「work」に移動して、計算を以下のとおり実行します。

```
% ./kSpin Au111Surface_M0.dat
```

または、MPI 計算では、例えば、4MPI プロセスの場合は以下のとおり実行します。

```
% mpirun -np 4 ./kSpin Au111Surface_M0.dat
```

計算の進行に伴い、以下の標準出力が得られます。

```
*****
*****
kSpin:
code for evaluating spin related properties
in momentum space of solid state materials.
Copyright (C), 2019,
Hiroki Kotaka, Naoya Yamaguchi and Fumiyuki Ishii.
This software includes the work that is distributed
in version 3 of the GPL (GPLv3).

Please cite the following article:
H. Kotaka, F. Ishii, and M. Saito,
Jpn. J. Appl. Phys. 52, 035204 (2013).
DOI: 10.7567/JJAP.52.035204.
*****
*****

Input filename is "Au111Surface.scfout"

Start "MulPOnly" Calculation (6).

##### ORBITAL DATA #####
ClaOrb_MAX[0]: 2
ClaOrb_MAX[1]: 8
Total Band (2*n): 124
#####

##### CALC TIME #####
Total Calculation Time: 0.642631 (s)
#####
```

```
##### CALC TIME #####  
Total Calculation Time: 0.660566 (s)  
#####
```

上記のとおり計算が正常に完了すると、ディレクトリ「work」に以下の出力ファイルが生成されます。

```
Au111Surface_MO.AtomMulP  
Au111Surface_MO.MulP_s  
Au111Surface_MO.MulP_p  
Au111Surface_MO.MulP_p1  
Au111Surface_MO.MulP_p2  
Au111Surface_MO.MulP_p3  
Au111Surface_MO.MulP_d  
Au111Surface_MO.MulP_d1  
Au111Surface_MO.MulP_d2  
Au111Surface_MO.MulP_d3  
Au111Surface_MO.MulP_d4  
Au111Surface_MO.MulP_d5  
Au111Surface_MO.atominfo  
temporal_12345.input
```

## 出力ファイル

各出力ファイルの内容を以下に説明します。

### AtomMulP ファイル

このファイルは原子ごとに分解された  $k$  空間スピン密度行列のデータを記録します。*MulPCalc53.6* の機能により解析できます。

### MulP\_xx ファイル

このファイルは原子ごとに分解された  $k$  空間スピン密度行列の  $xx$  成分のデータを記録します。*MulP-Calc53.6* の機能により解析できます。

### atominfo ファイル

このファイルは格子ベクトルと PAO の情報を提供します。

### temporal\_12345.input

このファイルは scfout ファイルに記録された入力ファイルのコピーです。

## 原子ごとに分解された $k$ 空間スピン密度行列の解析

*MulPCalc* を用いて AtomMulPBand ファイルまたは AMulPBand\_xx ファイルから原子ごとに分解された  $k$  空間スピン密度行列を解析するためのデータが抽出できます。ディレクトリ「source」中で以下のようにコンパイルし、実行ファイルを作成して下さい。

```
% make MulPCalc
```

コンパイルが成功すると、実行ファイル「MulPCalc」がディレクトリ「work」に生成します。一例として、Au(111) 表面の Rashba スピン分裂のバンド分散に着目し、原子寄与に分解された k 空間スピン密度行列を解析します。最初に、以下のキーワードを入力ファイル「Au111Surface\_MO.dat」に追加します。

```
Filename.atomMulP Au111Surface_MO.AMulPBand # default: default
Filename.xyzdata Au111Surface_MO_MC # default: default
Num.of.Extract.Atom 3 # default: 1
Extract.Atom 1 2 3 # default: 1 2 ... (Num.of.Extract.Atom)
Calc.Type.3mesh 2 # default: 1
```

ポストプロセスコード「kSpin」による計算後に、原子寄与に分解された k 空間スピン密度行列を以下のように解析できます。

```
% ./MulPCalc Au111Surface_MO.dat
```

加えて、以下のキーワードを設定することで、図の見た目を調整することができます。

```
MulP.Vec.Scale 0.1 0.1 0.1 # default: 1.0 1.0 1.0
Data.Reduction 1 # default: 1
```

*MulPCalc* を実行後に、ディレクトリ「work」に以下の出力ファイルが得られます。

```
Au111Surface_MO_MC.MulPop
Au111Surface_MO_MC.MulPop55
Au111Surface_MO_MC.MulPop56
Au111Surface_MO_MC.plotexample
```

例として、以下のコマンドを実行すると、図 68 が得られます。これは Au(111) 表面のラッシュバースピン分裂のスピンテクスチャを表示したものです。

```
% gnuplot Au111Surface_MO_MC.plotexample
```

*MulPCalc* に関するさらなる情報は 53.6 節も参照してください。

### 53.6 MulPCalc: 原子ごとに分割された k 空間スピン密度行列

*MulPCalc* は「kSpin」で得られた AtomMulP ファイル、MulP\_XX ファイル、AMulPBand ファイル、または AMulPBand\_XX ファイルからマリケン密度を解析するためのデータを抽出します。まず、ディレクトリ「source」においてコンパイルを行い、実行ファイルを生成して下さい。

```
% make MulPCalc
```

コンパイルに成功すると、実行ファイル「MulPCalc」がディレクトリ「work」に生成されます。最初に、ディレクトリ「work」中に収容された入力ファイル「SiC\_Primitive\_BD.dat」を用いて OpenMX 計算を実行します。SCF 計算の後に以下のキーワードを設定し、*BandDispersion* 計算をポストプロセスコード「kSpin」を用いて実行します。

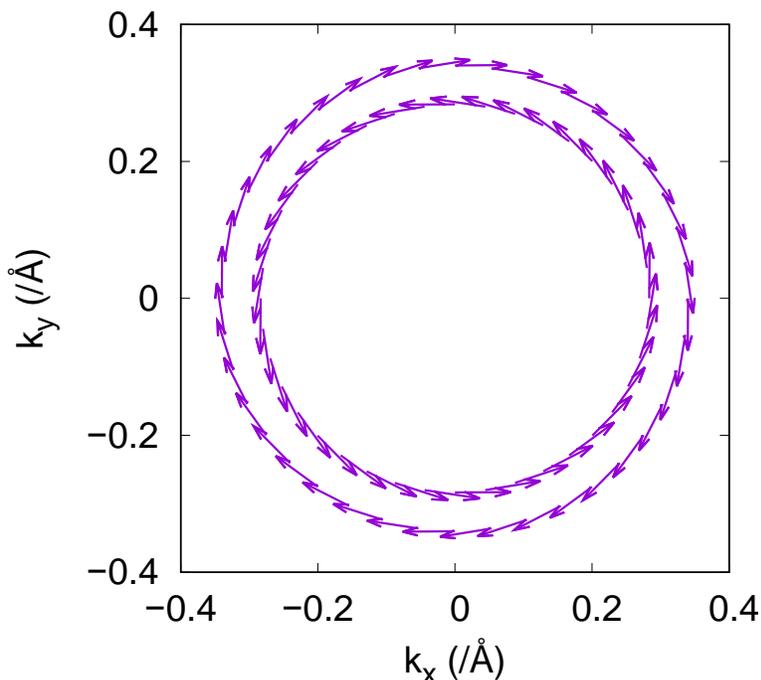


Figure 68: Au(111) 表面の Rashba 分裂に対する  $\Gamma$  点周辺のスピントクスチャ。外側と内側のスピントクスチャはそれぞれ 55 番と 56 番のバンドに対応。ベクトルは MulPop ファイル「Au111Surface\_MO\_MC.MulPop」に記録された各  $k$  点でのパウリ行列固有ベクトルによる  $\sigma_x, \sigma_y, \sigma_z$  の期待値。この図はプロット例のファイル「Au111Surface\_MO\_MC.plotexample」より作成。

```

Filename.scfout      sic_primitive.scfout # default: default
Filename.outdata     sic_primitive_BD      # default: default
Calc.Type            BandDispersion      # default: MulPOnly
Energy.Range         -10.0 6.0           # eV; default: 0.0 0.0

```

次に、*MulPCalc* を以下のように実行します。

```
% ./MulPCalc SiC_Primitive_BD.dat
```

炭素原子の  $p_z$  軌道のデータを抽出するために以下の設定を用います。

```

Filename.atomMulP    sic_primitive_BD.AMulPBand_p3 # default: default
Filename.xyzdata     sic_primitive_BD_MC_C_p3      # default: default
Num.of.Extract.Atom 1                               # default: 1
Extract.Atom         1                               # default: 1 2 ... (Num.of.Extract.Atom)

```

加えて、以下のキーワードを設定することで、図の見た目を調整することができます。

```

MulP.Vec.Scale      0.1 0.1 0.1                    # default: 1.0 1.0 1.0
Data.Reduction      2                               # default: 1

```

図 69 は *MulPCalc* を数回同様に実行し、MulPop ファイルの第 11、第 4、第 5 列を gnuplot の円スタイルを用いてプロットすることで得ることができます。

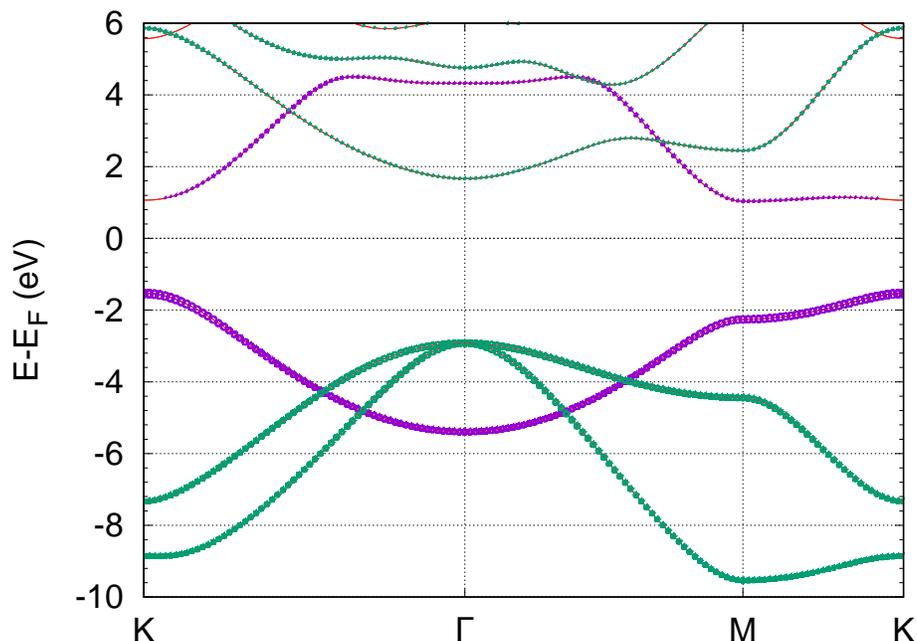


Figure 69: SiC プリミティブセル（不完全さの無い二次元八ニカム構造）の原子分割された k 空間 スピン密度行列。この系はバンドアンフォールディング法 52 章、バンド構造の解析 52.1 節で説明されたものである。緑色と紫色の円はそれぞれ炭素原子の  $s$ ,  $p_x$ ,  $p_y$  軌道を占有する電子数と  $p_z$  軌道を占有する電子数を表す。半径は電子数の大きさを反映。本データは AMulPBand\_XX ファイルの sic\_primitive\_BD.AMulPBand\_s, sic\_primitive\_BD.AMulPBand\_p1, sic\_primitive\_BD.AMulPBand\_p2, または sic\_primitive\_BD.AMulPBand\_p3 または sic\_primitive\_BD.AMulPBand\_p から *MulPCalc* を用いて抽出された。

もう一つの入力ファイル例「Au111Surface23\_FL.dat」がディレクトリ「work」に収容されています。これは清浄な Au(111) 表面のスラブモデルの計算を行うものです。この計算は他の計算例よりも多くの計算時間を要すことに注意して下さい。SCF 計算と後続の *FermiLoop* 計算によって、バンド分散とスピントクスチャが得られます。図 70(a) と 70(b) にその計算結果を図示します。ここで、二つの組の Rashba バンドが現れています。スラブモデルにおける上下の表面での表面状態が縮退しているためです。*MulPCalc* を用いれば片側の表面原子の寄与のみを解析することが可能となります。片側の表面からのスピントクスチャの寄与を図 70(c) に示します。この結果は *MulPCalc* の機能を用いて得られたものです。

それぞれのキーワードの仕様を以下に説明します。

### MulPCalc に関連するキーワードの一覧 (全 Calc.Type 共通)

#### Filename.atomMulP

AtomMulP ファイル、MulP\_XX ファイル、AMulPBand ファイル、または AMulPBand\_XX ファイルの名前を指定します。

#### Filename.xyzdata

出力ファイルの名前を指定します。キーワード「System.Name」に相当します。

#### **Num.of.Extract.Atom**

MulPCalc が抽出すべきマリケン密度のデータの原子の数を指定します。デフォルト値は 1 です。

#### **Extract.Atom**

MulPCalc が抽出すべきマリケン密度のデータの原子を指定します。デフォルト値は「1 2 ... キーワード「Num.of.Extract.Atom」の値」です。

#### **Data.Reduction**

MulPCalc が抽出すべきマリケン密度のデータの k 点の数を指定します。このキーワードは k 点の間引き、あるいはデータサイズの縮小に有用です。

#### **MulP.Vec.Scale**

スピネクスタチャを表現するベクトルを描く縮尺を指定します。例えば、値「0.1 0.2 0.3」は  $x$  軸に 0.1、 $y$  軸に 0.2、 $z$  軸に 0.3 の縮尺を指定します。

#### **Filename.outdata**

「kSpin」の計算と同一の設定として下さい。

#### **Calc.Type**

「kSpin」の計算と同一の設定として下さい。

### MulPCalc に関連するキーワードの一覧 (Calc.Type = FermiLoop or GridCalc)

#### **Search.kCentral**

「kSpin」の計算と同一の設定として下さい。

#### **Calc.Type.3mesh**

「kSpin」の計算と同一の設定として下さい。

#### **Energy.Range**

「kSpin」の計算と同一の設定として下さい。

### MulPCalc に関連するキーワードの一覧 (Calc.Type = BandDispersion)

#### **Energy.Range**

「kSpin」の計算と同一の設定として下さい。

#### **Band.Nkpath**

「kSpin」の計算と同一の設定として下さい。

#### **Band.kpath**

「kSpin」の計算と同一の設定として下さい。

### MulPCalc に関連するキーワードの一覧 (Calc.Type = MulPOnly)

### Calc.Type.3mesh

計算する面を指定します。1, 2, 3 の値はそれぞれ  $k_a k_b$ ,  $k_b k_c$ ,  $k_c k_a$  面に対応しています。デフォルト値は 1 です。

### 出力ファイル

「MulPCal」による計算が正常に完了すると、以下の出力ファイルが作業ディレクトリに生成されます。

#### MulPop ファイル

このファイルは各  $k$  点に対するデータを記録します。第 1、第 2、第 3 列はそれぞれ  $k$  点の  $\text{\AA}^{-1}$  単位での  $k_x$ ,  $k_y$ ,  $k_z$  成分に対応します。第 4 列は eV 単位でのエネルギーに対応し、第 5、第 6、第 7 列はそれぞれ電子の総数、 $\alpha$  スピンの数、 $\beta$  スピンの数に対応します。第 8、第 9、第 10 列はそれぞれ Bohr 磁子単位での  $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  の期待値に対応します。もし、キーワード「Calc.Type」の設定が「BandDispersion」ならば第 11 列は Bohr $^{-1}$  単位で  $k$  パスに沿った  $k$  点の距離に対応します。

#### MulPop\_YY ファイル

このファイルはバンドインデックス YY の各  $k$  点のデータを記録します。このファイルの内容の表記は MulPop ファイルと同一です。

#### plotexample ファイル

もしキーワード「Calc.Type」の値が「GridCalc」ならば、このファイルは gnuplot スクリプトとして活用できます。

#### plotexample\_YY ファイル

もしキーワード「Calc.Type」の値が「GridCalc」ならば、このファイルはバンドインデックス YY のバンドの gnuplot スクリプトとして活用できます。

## 53.7 kSpin の MPI 並列計算

ポストプロセスコード「kSpin」は MPI 並列計算が実行可能です。詳細を以下に説明します。

### FermiLoop

$k$  点メッシュ上の  $k$  点に対して MPI 並列化されていますが、第一段階の探索でメッシュ分割された領域において第二段階の探索が行われます。第二段階の探索は、第一段階の探索で計算に必要な領域に対して逐次的に行われるので、適切な MPI プロセスの数の最大値はキーワード「k-plane.2ndStep」の第一と第二の値の積です。計算の大部分は第二段階の探索に費やされることに注意してください。

### GridCalc

$k$  点メッシュ上の  $k$  点に対して MPI 並列化されているので、適切な MPI プロセスの数の最大値はキーワード「k-plane.1stStep」の第一と第二の値の積です。

### BandDispersion

各  $k$  点経路上の  $k$  点に対して MPI 並列化されているので、適切な MPI プロセスの数の最大値は各  $k$  点経路上で設定された  $k$  点数の最大値です。

## MulPOnly

k 点に対して MPI 並列化されているので、適切な MPI プロセスの数の最大値は指定された k 点の数です。

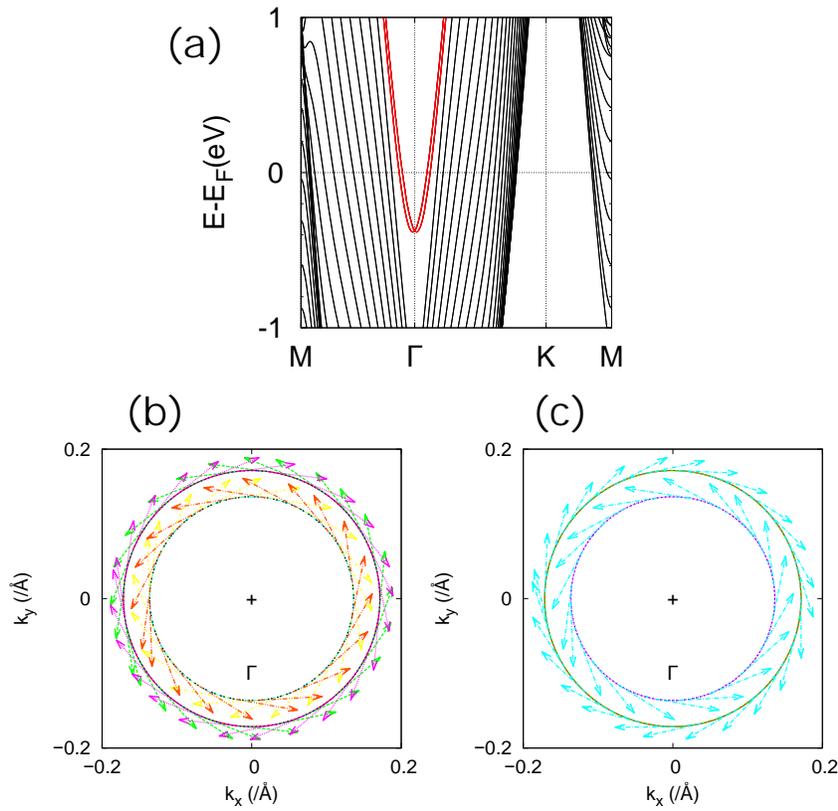


Figure 70: (a) Au(111) 表面のラシュバスピンスplitのバンド分散。バンドファイル「Au111Surface23.Band」をプロット。BandDispersion 53.4 節を参照のこと。赤色の曲線は Rashba バンドを表し、バンドインデックス 413, 414, 415, 416 に対応。色付けは GNUBAND ファイル「Au111Surface23.GNUBAND」を変更するか、OMXTool [146] を利用することで再現できます。Γ 点まわりでの Rashba スピンスplitのスピントクスチャ。(b) Au(111) の両表面の寄与、(c) Au(111) の片表面の寄与。ベクトルは各 k 点でのパウリ行列のベクトルの期待値を表し、以下の Pxyz\_YY ファイルにそれぞれ記録：Au111Surface23\_FL.Pxyz\_413, Au111Surface23\_FL.Pxyz\_414, Au111Surface23\_FL.Pxyz\_415, Au111Surface23\_FL.Pxyz\_416, Au111Surface23\_FL\_MC.Pxyz\_413, Au111Surface23\_FL\_MC.Pxyz\_414, Au111Surface23\_FL\_MC.Pxyz\_415, Au111Surface23\_FL\_MC.Pxyz\_416。閉曲線は一定エネルギー面を表し、以下の FermiSurf\_YY ファイルにそれぞれ記録：Au111Surface23\_FL.FermiSurf\_413, Au111Surface23\_FL.FermiSurf\_414, Au111Surface23\_FL.FermiSurf\_415, Au111Surface23\_FL.FermiSurf\_416, Au111Surface23\_FL\_MC.FermiSurf\_413, Au111Surface23\_FL\_MC.FermiSurf\_414, Au111Surface23\_FL\_MC.FermiSurf\_415, Au111Surface23\_FL\_MC.FermiSurf\_416。中心の十字点は Γ に対応。これらの図はそれぞれプロット例ファイル Au111Surface23\_FL.plotexample と Au111Surface23\_FL\_MC.plotexample により作成。

## 54 スピン螺旋計算

ノンコリニア DFT 法の枠組みにおいて (ただしスピン軌道相互作用なし) スピン螺旋構造を持つ磁性体の計算が可能です。スピン螺旋計算は一般化ブロッホの定理 [79, 80, 86, 87] に基づいています。スピン軌道相互作用は一般化ブロッホ定理で課される螺旋対称性を壊してしまうため、スピン軌道相互作用を含めたスピン螺旋計算はサポートされていないことに留意して下さい。本機能を用いて発表を行う際には、文献 [79, 80] を引用して頂けると幸いです。

本計算を実行する際には、以下のキーワードを設定してください。

```
scf.SpinPolarization      NC      # On|Off|NC
scf.Generalized.Bloch     on      # On|Off, default=off
scf.SpinOrbit.Coupling    off     # On|Off, default=off
```

スピン螺旋構造を指定するため、二つのパラメータを与える必要があります。第一のパラメータは円錐角です。2種類の螺旋構造が考えられ、一つは円錐螺旋 ( $0 < \theta < 90$ )、他方は平坦螺旋 ( $\theta = 90$ ) です。ここで  $\theta$  は円錐角度で、 $\theta$  は磁性原子に対して指定しなければなりません。図 71(a) に示されるように、OpenMX ではスピン磁気モーメントの方位を用いて円錐 (オイラー) 角  $\theta$  が指定されます。円錐 (オイラー) 角を指定する方法に関しては「ノンコリニア DFT」に関する 34 章に記載されています。例えば円錐 (オイラー) 角は以下のように指定できます。

```
<Atoms.SpeciesAndCoordinates
 1 Fe 0.0 0.0 0.0 10.0 6.0 90.0 0.0 0.0 0.0 1 off
Atoms.SpeciesAndCoordinates>
```

上記の例では、8 列目で  $\theta = 90$  度と指定することで、平坦螺旋構造を与えています。加えて、9 列目で  $\varphi$  の初期角度を指定することもできます。円錐角を固定するためには、「ノンコリニアスピン方位に対する制約条件付き DFT に関する」38 章の拘束法も利用可能です。

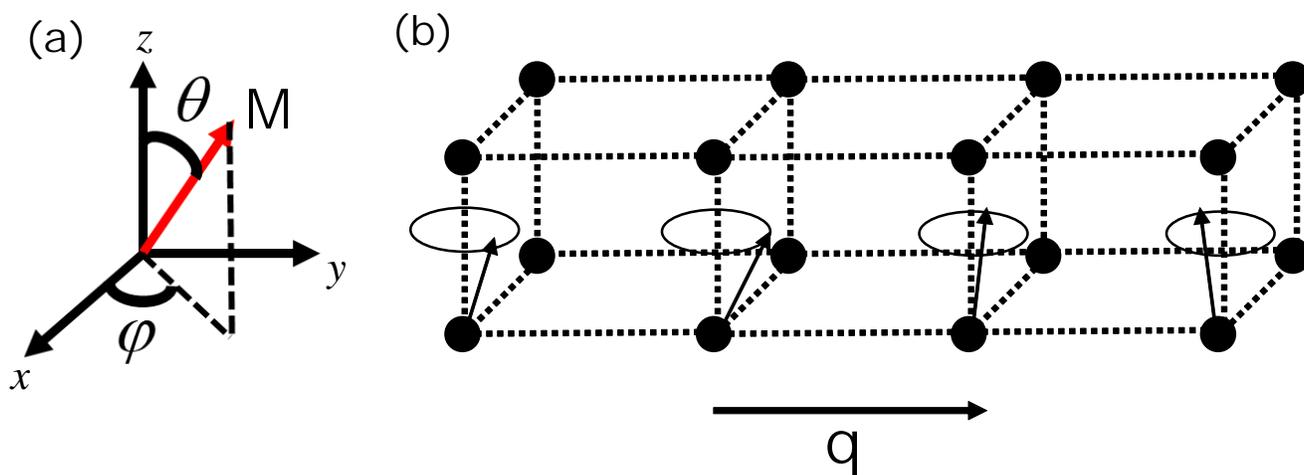


Figure 71: (a) 磁性原子の円錐 (オイラー) 角の定義。(b) サイト  $i$  の原子の磁性モーメント  $M_i = M_i \{ \cos(\mathbf{q} \cdot \mathbf{R}_i + \varphi_0) \sin(\theta_i) + \sin(\mathbf{q} \cdot \mathbf{R}_i + \varphi_0) \sin(\theta_i) + \cos(\theta_i) \}$ .

スピンの螺旋構造の第二のパラメータは螺旋ベクトル  $\mathbf{q}$  であり、次のキーワードで指定されます。

```
Spin.Spiral.Vector 0.0 0.0 0.0 # q1 q2 q3
```

上記のフォーマットにおいて、螺旋ベクトルは逆格子ベクトルで張られた規格化座標で指定されます。第1、第2、第3列により、逆格子ベクトル  $\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \tilde{\mathbf{a}}_3$  に対する螺旋ベクトルの成分を指定します。その際に、原子サイト  $i$  でのスピン角度は次式で与えられます。

$$M_i = M_i \{ \cos(\mathbf{q} \cdot \mathbf{R}_i + \varphi_0) \sin(\theta_i) + \sin(\mathbf{q} \cdot \mathbf{R}_i + \varphi_0) \sin(\theta_i) + \cos(\theta_i) \}.$$

格子ベクトルの並進に伴い、各原子サイトのスピン角度はこの式に基づき回転します。スピン螺旋計算の例として、図 72 に FCC 構造の Fe の計算結果を示します。直交座標 ( $2\pi/a$  の単位) で定義された  $(0, 0, 0.6)$  と  $(0.2, 0, 1)$  に安定な螺旋状態が出現します。スピン螺旋計算の信頼できる結果を得るためには適切なパラメータの設定が必要です。例えば  $k$  点の数、軌道の数、カットオフ半径、そしてカットオフエネルギーなどの選択です。これらの議論が文献 [87] にありますので、参照して下さい。また、キーワード「scf.Mixing.Type」により様々な電荷密度混合法が利用できますが、スピン螺旋計算には「RMM-DIISH」の使用を推奨します。一連のベンチマーク計算から、「RMM-DIISH」は他の混合法よりも速やかに収束することが分かっています。詳細な説明は SCF 収束の 13 章を参照して下さい。

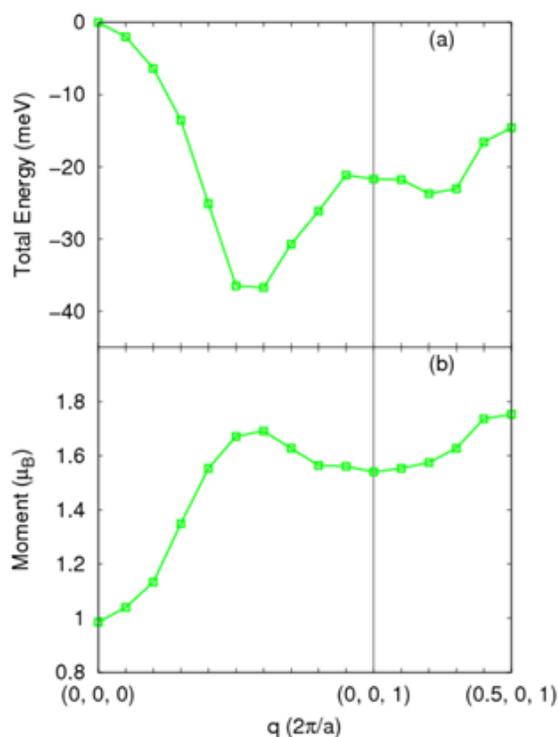


Figure 72: スピン螺旋計算によって得られた Fe (FCC 構造) の (a) 全エネルギーと (b) 磁気モーメント。  $\mathbf{q}$  は直交座標で定義された螺旋ベクトル。計算に用いた入力ファイル「Fefcc-SpinSpiral.dat」はディレクトリ「work」に収容。

## 55 Fukui-Hatsugai-Suzuki の方法によるチャーン数とベリー曲率の計算

### 55.1 概要

ポストプロセスコード「calB」を用いてバンドのチャーン数とベリー曲率が計算できます。この計算は Fukui-Hatsugai-Suzuki の方法 [81, 85] に基づいており、隣接  $k$  点の Kohn-Sham 軌道間の重なり行列要素を用いてチャーン数とベリー曲率が計算されます。本機能はノンコリニア DFT 計算のみに対応しており、コリニア DFT 計算では利用可能でないことに注意して下さい。本機能を用いて発表を行う際には、文献 [84] を引用して頂けると幸いです。

チャーン数は整数値のトポロジカル不変量であり、物質のバンドのトポロジジーを特徴づけます。有限のチャーン数  $C$  をもつ系では、次式で定義される異常ホール伝導度が誘起されます。

$$\sigma_{xy} = -\frac{e^2}{h} C \quad (C = 1, 2, 3, \dots)$$

ベリー曲率  $\mathbf{F}_n = \nabla \times \mathbf{A}_n$ ,  $\mathbf{A}_n = -i\langle u_{nk} | \frac{\partial}{\partial \mathbf{k}} | u_{nk} \rangle$  を用いて、チャーン数は次式で定義されます。

$$C = \frac{1}{2\pi} \sum_n^{\text{occ.}} \int F_{nz} dk^2$$

Fukui-Hatsugai-Suzuki の方法 [81] では、次式で定義される重なり行列  $U$  を用います。

$$U_{\Delta \mathbf{k}}(\mathbf{k}) = \det \langle u_n(\mathbf{k}) | u_m(\mathbf{k} + \Delta \mathbf{k}) \rangle$$

この重なり行列  $U$  が次式で定義されるベリー接続  $\mathbf{A}(\mathbf{k})$  とベリー曲率  $F(\mathbf{k})$  を計算するための中心的な役割を担います。

$$\begin{aligned} \mathbf{A}(\mathbf{k}) &= \text{Im} \log U_{\Delta \mathbf{k}}(\mathbf{k}), \\ F(\mathbf{k}) &= \text{Im} \log U_{\Delta k_1}(\mathbf{k}) U_{\Delta k_2}(\mathbf{k} + \Delta k_1) U_{\Delta k_1}^{-1}(\mathbf{k} + \Delta k_2) U_{\Delta k_2}^{-1}(\mathbf{k}) \end{aligned}$$

図 73 に示す様に、ベリー曲率は第一ブリルアンゾーンに導入された等間隔格子上での各「ブラケット」において以下の公式で計算されます。ここで、ブラケットはブリルアンゾーンのメッシュ状に切られた領域を意味します。

$$F(\mathbf{k}) = \text{Im} \log U_{12} U_{23} U_{34} U_{41}$$

ベリー曲率の周回積分の全ての寄与を総和して、チャーン数が計算されます。

$$C = \frac{1}{2\pi} \sum_{\mathbf{k}}^{\text{BZ}} F(\mathbf{k})$$

### 55.2 実例

例として、図 74 にグラフェンのベリー曲率を示します。ベリー曲率の大きさの絶対値はバンドギャップの逆数の平方におおよそ比例するので、スピン軌道相互作用を考慮すれば重いディラック点が出現する  $K$  点と  $K'$  点の周辺に大きなベリー曲率が出現します。ディレクトリ「work」に収容された入力ファイル「Graphene-Chern.dat」を用いて、計算の手続きを以下に説明します。

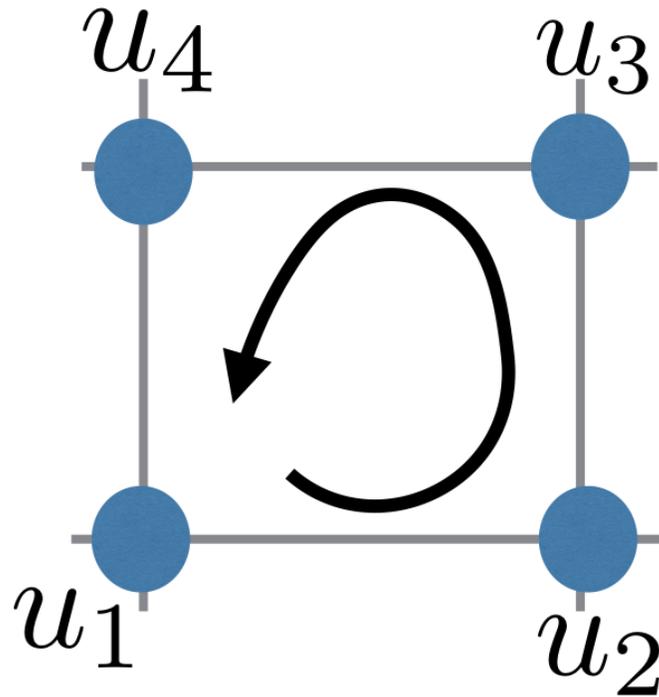


Figure 73: ベリー曲率の計算法、各ブラケットにおいて周回積分を実施。

## SCF 計算

ディレクトリ「work」に収容された入力ファイル「Graphene-Chern.dat」を用いてグラフェンに対して SCF 計算を実行します。キーワード「HS.fileout」を以下のように「on」に設定します。

```
HS.fileout    on    #on|off, default=off
```

SCF 計算が正常に終了すると、出力ファイル「Graphene-Chern.scfout」が得られます。

## チャーン数とベリー曲率の計算

次に、ポストプロセスコード「calB」を用いてチャーン数とベリー曲率の計算に進みます。ディレクトリ「source」に移動し、コード「calB」のコンパイルを以下の様に実行します。

```
% make calB
```

コンパイルが正常に完了するとディレクトリ「work」内に実行ファイル「calB」が生成されます。次に、ディレクトリ「work」に移動して、以下のように「calB」を実行します。

```
% ./calB graphene.scfout
```

or

```
% ./calB graphene.scfout < calB.in > calB.out
```

or

```
% mpirun -np 4 ./calB graphene.scfout < calB.in > calB.out
```

## calB のパラメタ

以下の内容を持つ入力ファイル「calB.in」を用意します。

```
1
0
0 0 1
100 100
```

各パラメータの意味を以下に説明します。

- 第一行に、すべてのバンドのベリー曲率の総和を計算するか、それとも分離したバンドについてベリー曲率を計算するか指定します。前者なら 1 を、後者なら 0 を設定します。後者の設定はバンドの集合が完全に他バンドから分離している場合のみに有効です。
- 第二行に、計算に考慮するバンドの数を指定します。バンド数は下端のバンドから数えます。0 の指定ですべての価電子バンドが考慮されます。
- 第三行に、逆空間で計算される平面を指定します。例えば、「0 0 1」は  $k_3 = 0$  平面の指定に相当します。
- 第四行に、メッシュの数を指定します。グラフェンの例では第三行で指定された  $k_3 = 0$  平面が  $100 \times 100$  のメッシュで分割されます。

## 出力ファイル

「calB」の計算の後に以下のファイルが生成されます。

- BerryC\*.dat

計算で得られたベリー曲率とチャーン数がこのファイルに記録します。ここでファイル名の BerryC の前の「\*」は含まれるバンドの数です。グラフェンの例では「\*」は「8」です。ベリー曲率の単位は  $\text{\AA}^{-2}$  です。グラフェンの例における「BerryC8.dat」の一部を以下に示します。

```
#Mesh Number:100*100
#Band Number:8
#ChernNumber = 0.000000
#k1          k2          F (ang-2)
0.005000 0.005000 -1.59799534e-05
0.015000 0.005000 1.94455630e-05
0.025000 0.005000 3.58476883e-05
....
..
```

以下のように Gnuplot を用いてデータを可視化できます。

```
% splot "BerryC8.dat" w l
```

図 74 は  $k_3 = 0$  平面でのベリー曲率を示します。

- temporal\_12345.input

このファイルは入力ファイルのコピーです。これは SCF 計算に用いられたもので、scfout ファイルから再構築されました。

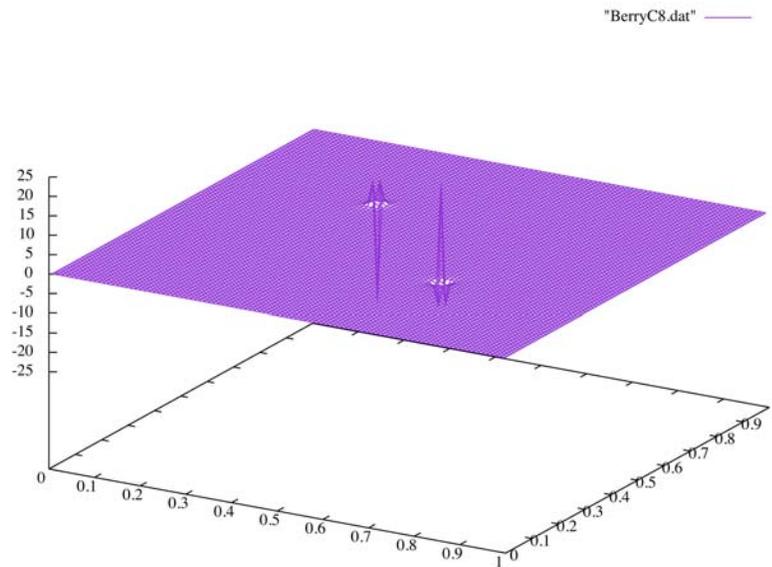


Figure 74: グラフェンのバンドのベリー曲率

## 56 Fukui-Hatsugai の方法による $Z_2$ 不変量の計算

### 56.1 概要

ポストプロセスコード「Z2FH」を用いて系の  $Z_2$  不変量が計算できます。Fukui と Hatsugai によって提案されたベリー位相に基づく方法 [81, 82] で系の  $Z_2$  不変量が計算されます。本機能はノンコリニア DFT 計算のみに対応しており、コリニア DFT 計算では利用可能でないことに注意して下さい。また、磁性系は現状の実装では扱うことが出来ません。本機能を用いて発表を行う際には、文献 [84] を引用して頂けますと幸いです。

$Z_2$  不変量は 0 または 1 の値となるトポロジカル不変量で、時間反転対称性がある非磁性な系で定義されます。 $Z_2 = 1$  と  $Z_2 = 0$  はそれぞれトポロジカル絶縁体と自明な絶縁体に相当します。 $Z_2$  不変量は次式で定義されます。

$$Z_2 = \frac{1}{2\pi} \sum_n^{\text{occ.}} \left( \int_{\partial B} \mathbf{A}_n \cdot d\mathbf{k} - \int_B F_{nz} dk^2 \right) \pmod{2}$$

ここで  $\mathbf{A}_n = -i \langle u_{n\mathbf{k}} | \frac{\partial}{\partial \mathbf{k}} | u_{n\mathbf{k}} \rangle$  はベリー接続と呼ばれ、そして、 $\mathbf{F}_n = \nabla \times \mathbf{A}_n$  はベリー曲率と呼ばれます。積分範囲  $B = [-\frac{\mathbf{G}_1}{2}, \frac{\mathbf{G}_1}{2}] \otimes [0, \frac{\mathbf{G}_2}{2}]$  はブリルアンゾーンの半分だけで十分です。系が時間反転対称性を持つため、トポロジカル不変量がブリルアンゾーンの半分で定義されるためです。積分を実行するため、重なり行列  $U$  を用います。これは Fukui, Hatsugai, Suzuki によって提案され  $U_{\Delta\mathbf{k}} = \det \langle u_n(\mathbf{k}) | u_m(\mathbf{k} + \Delta\mathbf{k}) \rangle$  と定義されます [81, 82]。ベリー接続とベリー曲率は各「プラケット」において以下のように計算されます。ここで、プラケットはブリルアンゾーンのメッシュ状に切られた領域を意味します。

$$\begin{aligned} A_{ab} &= \text{Im} \log U_{ab}, \\ F &= \text{Im} \log U_{12} U_{23} U_{34} U_{41}. \end{aligned}$$

そして、各プラケットの上での整数値の場合  $n (= 0, \pm 1)$  が以下の公式から計算できます。

$$n(\mathbf{k}) = \frac{1}{2\pi} (A_{12} + A_{23} + A_{31} + A_{41} - F)$$

ブリルアンゾーンの半分で全ての  $n$  を総和し、合計値の剰余 2 を考慮して、 $Z_2$  を得ます。 $Z_2$  不変量がゲージ非依存であることに注意しなければなりません。しかし、各  $n$  の値はゲージ依存で、これは計算機環境、コンパイラの最適化レベル、電子構造のわずかな差異によって変化します。A と F の計算の詳細は「Fukui-Hatsugai-Suzuki の方法によるチャーン数とベリー曲率の計算」の章を参照して下さい。 $Z_2$  不変量の計算はブリルアンゾーンの半分で周回積分で実行されるので、これは波動関数のゲージの任意性に依存します。それゆえ、ブリルアンゾーンの半分の境界でゲージを補正しなければなりません。図 75 に示すように、境界で以下の 3 種類のゲージ補正を考えます。

並進対称性 (赤色の部分)

$$|u_n(\mathbf{k} + \frac{\mathbf{G}_1}{2})\rangle = e^{i\mathbf{G}_1 \cdot \mathbf{r}} |u_n(\mathbf{k} - \frac{\mathbf{G}_1}{2})\rangle$$

時間反転対称性 (青色の部分)

$$|u_n(\mathbf{k})\rangle = \Theta |u_n(-\mathbf{k})\rangle = |u_n^{*\beta}(-\mathbf{k})\rangle |\alpha\rangle - |u_n^{*\alpha}(-\mathbf{k})\rangle |\beta\rangle$$

クラマース縮退 (黄色の部分)

$$|u_{2n}(\mathbf{k})\rangle = \Theta|u_{2n-1}(-\mathbf{k})\rangle$$

この計算では積分区間の半分で固有値問題を解きます。言い換えると、図 75 に示したようにブリルアンゾーンの四分の一です。

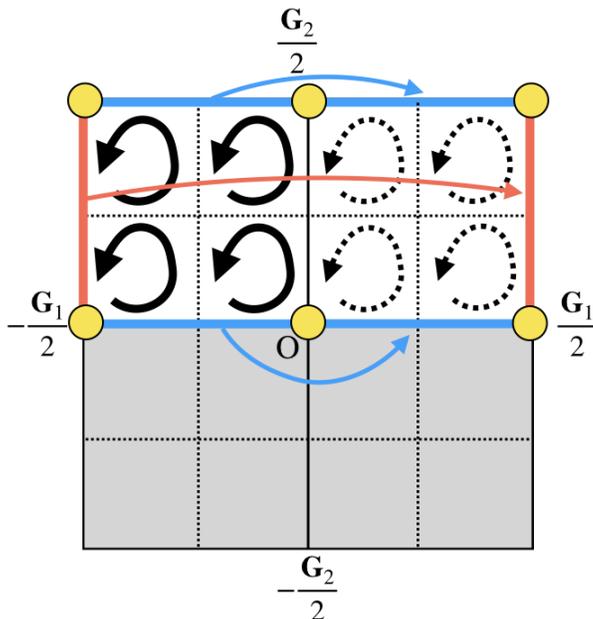


Figure 75: ブリルアンゾーンの半分でゲージを修正。波動関数のゲージを補正。赤色の部分では並進対称性を満たすようにゲージを補正、青色の部分では時間反転対称性を満たすようにゲージを補正、黄色の部分ではクラマース縮退を満たすようにゲージを補正。

他の領域での積分を実行する際には対称性的に対応するプラケットの波動関数のゲージを固定して波動関数を生成し、積分を実施します。

三次元系では、ブリルアンゾーンは  $k_n = 0, k_n = \frac{G_n}{2} (n = 1, 2, 3)$  の6面の時間反転不変な平面を持ちます。よって、図 76 に示すように6個の  $Z_2$  不変量  $(x_0, x_\pi, y_0, y_\pi, z_0, z_\pi)$  が定義できます。これらの不変量が以下の方程式を満たすことに注意してください。

$$x_0 + x_\pi = y_0 + y_\pi = z_0 + z_\pi \pmod{2}$$

それゆえ、4個の不変量だけが独立なパラメタとなります。この事実に基づき、三次元系の  $Z_2$  不変量は次式で定義されます。

$$Z_2 = (\nu_0, \nu_1, \nu_2, \nu_3) = (x_0 + x_\pi, x_\pi, y_\pi, z_\pi)$$

特に、 $\nu_0 = 1$  となる系は強いトポロジカル絶縁体と呼ばれます。なぜなら  $Z_2 = 1$  の状態がブリルアンゾーンのすべての方向に現れるからです。

## 56.2 実例

例として、三次元トポロジカル絶縁体の  $\text{Bi}_2\text{Se}_3$  の  $Z_2$  不変量を計算します [83]。

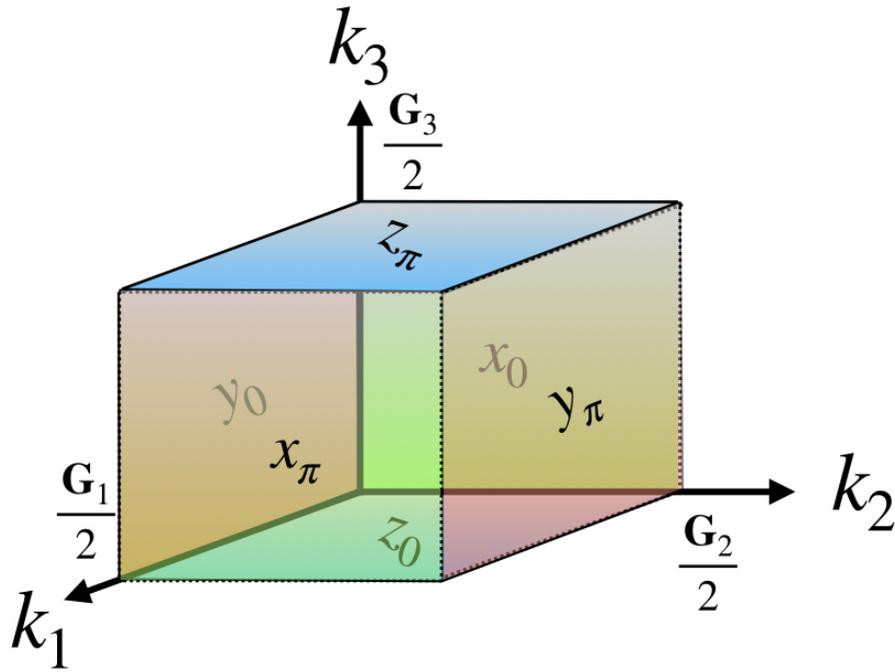


Figure 76: 三次元の逆格子空間で定義された6個の  $Z_2$  不変量。これらの不変量は  $x_0 + x_\pi = y_0 + y_\pi = z_0 + z_\pi \pmod{2}$  を満たす。そして、4個の不変量が独立なパラメータとして残る。

## SCF 計算

ディレクトリ「work」に收容された「Bi2Se3-Z2.dat」を用いて、以下のキーワードを設定し、SCF 計算を実行します。

```
HS.fileout      on      #on|off, default=off
```

SCF 計算が正常に終了すると、出力ファイル「Bi2Se3-Z2.scfout」が生成されます。

## $Z_2$ 不変量の計算

ポストプロセスコード「Z2FH」を用いた  $Z_2$  不変量の計算の前に、ディレクトリ「source」にてコードを以下のようにコンパイルします。

```
% make Z2FH
```

コンパイルの後に、実行ファイル「Z2FH」がディレクトリ「work」に生成されます。そして、以下のように  $Z_2$  不変量の計算に進みます。

```
% ./Z2FH Bi2Se3-Z2.scfout
```

```
or
```

```
% ./Z2FH Bi2Se3-Z2.scfout < Z2FH.in > Z2FH.out
```

```
or
```

```
% mpirun -np 4 ./Z2FH Bi2Se3-Z2.scfout < Z2FH.in > Z2FH.out
```

ファイル「Z2FH.in」は以下のように「Z2FH」の計算に必要なとされるパラメータを含んでいます。

```
5
1
1
```

上記の最初の実行例では、以下のようにプログラムから対話的に問われます。

```
*****
*****
Z2FH:
code for calculating the Z2 invariant of bulk systems
by Fukui-Hatsugai method.
Copyright (C), 2019, Hikaru Sawahata, Naoya Yamaguchi,
Fumiyuki Ishii and Taisuke Ozaki
This is free software, and you are welcome to
redistribute it under the constitution of the GNU-GPL.

Please cite the following article:
H. Sawahata, N. Yamaguchi, H. Kotaka and F. Ishii,
Jpn. J. Appl. Phys. 57, 030309 (2018).
*****
*****
Mesh1 Number(Half Direction):5
Calculate All plane?(0:No,1:Yes)1
Restart?[1:x0,2:xpi,3:y0....]1

Read the scfout file (Bi2Se3-Z2.scfout)
***
The file format of the SCFOUT file: 3
And it supports the following functions:
- jx
- polB
- kSpin
- Z2FH
- calB
***
```

## Z2FHのパラメータ

入力ファイル「Z2FH.in」のパラメータを以下に説明します。

- 第一行には、積分区間の半分におけるメッシュの数を指定します。言い換えると、第一ブリルアンゾーンの四分の一の空間です (図 75 を参照してください)。5 を指定すると、コードはブリルアンゾーンの四分の一において  $5 \times 5$  の  $k$  点メッシュで形成された正方形のプラケットで積分を実施します。

- 第二行には、計算する  $Z_2$  不変量を  $Z_2$  不変量が定義される 6 個の  $k$  平面の全てで実施するか否か指定します。もし  $(\nu_0, \nu_1, \nu_2, \nu_3)$  に対する 4 個の平面でのみ計算したいなら、0 を指定します。反対に 1 を指定すると、 $(x_0, x_\pi, y_0, y_\pi, z_0, z_\pi)$  に対する 6 個の平面のすべての計算に対応します。
- 第三行目には、計算をリスタートする  $k$  面を指定します。数値 1, 2, 3, 4, 5, 6 はそれぞれ  $k_1 = 0, k_1 = \frac{G_1}{2}, k_2 = 0, k_2 = \frac{G_2}{2}, k_3 = 0, k_3 = \frac{G_3}{2}$  面に対応します。

## 出力ファイル

ポストプロセスコード「Z2FH」による計算の後、以下のファイルが生成されます。

- Z2.dat

このファイルは  $Z_2$  不変量の計算結果を記録します。すべての  $k$  面で計算する場合 (第二行で 1 を指定した場合) は、以下のように 6 個の  $k$  面での  $Z_2$  不変量が第一行に並び、4 個の  $k$  面での  $Z_2$  不変量が第二行に並びます。

```
Z2 invariant:(x0,xpi,y0,ypi,z0,zpi)=(1.000000,-0.000000,1.000000,-0.000000,1.000000,-0.000000)
Z2 invariant:(nu0,nu1,nu2,nu3):(1,0,0,0)
```

- LCNum\*.dat

整数値の場合  $n(\mathbf{k})$  のデータファイル。ここで LCNum の前の\*は 1 から 6 まで走るインデックスです。数値 (1,2,3,4,5,6) はそれぞれ  $(x_0, x_\pi, y_0, y_\pi, z_0, z_\pi)$  に対応します。4 面のみを計算する場合は、 $(x_\pi, y_\pi, z_0, z_\pi)$  に対応する四つのファイル LCNum(2,4,5,6).dat が生成されます。

- LCNum\*.pl

Gnuplot 用のスクリプトファイル。ここで LCNum の前の\*は 1 から 6 まで走るインデックスです。数値 (1,2,3,4,5,6) はそれぞれ  $(x_0, x_\pi, y_0, y_\pi, z_0, z_\pi)$  に対応します。4 面のみを計算する場合は、 $(x_\pi, y_\pi, z_0, z_\pi)$  に対応する四つのファイル LCNum(2,4,5,6).pl が生成されます。整数値の場合  $n(\mathbf{k})$  を以下のように可視化できます。

```
% gnuplot LCNum1.pl
```

- temporal\_12345.input

このファイルは入力ファイルのコピーです。これは SCF 計算に用いられたもので、scfout ファイルから再構築されました。

例として、 $k_1 = 0$  での整数値の場合  $n(\mathbf{k})$  を図 77 に示します。これは上記で説明した手順で「gnuplot LCNum1.pl」として得られます。 $n(\mathbf{k})$  はゲージに依存する値なので、計算環境によっては異なる結果になるかもしれません。他方、 $Z_2$  不変量は再現されるはずで

### 56.3 入力ファイル

参考のため、四つの入力ファイル例がディレクトリ「work」に収容されています。

- Bi2Se3-Z2.dat  
Bi<sub>2</sub>Se<sub>3</sub> の強いトポロジカル絶縁体の場合
- Bi2Se3\_trivial-Z2.dat  
Bi<sub>2</sub>Se<sub>3</sub> の自明な絶縁体の場合
- Bi2Se3\_weak-Z2.dat  
Bi<sub>2</sub>Se<sub>3</sub> の弱いトポロジカル絶縁体の場合
- Bi111-Z2.dat  
Bi(111) スラブの二次元トポロジカル絶縁体の場合

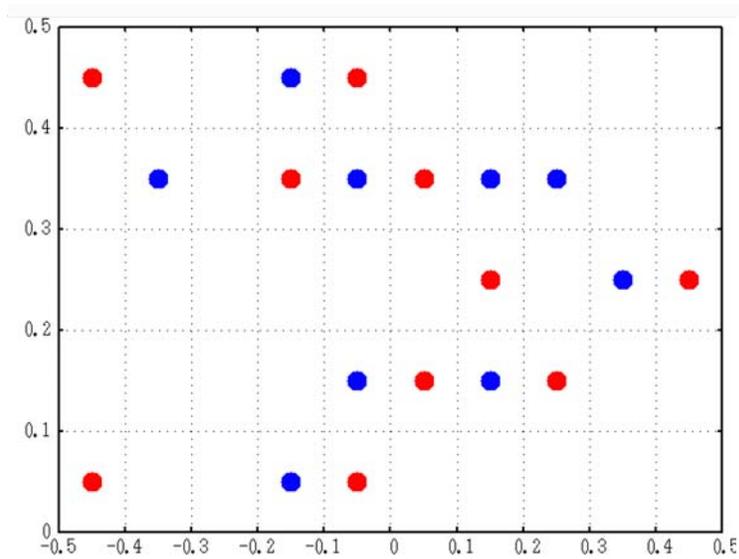


Figure 77: 整数値の場  $n(\mathbf{k})$  の例。赤い丸は+1を表し、青い丸は-1を表し、空白は0を表す。 $n(\mathbf{k})$  はゲージに依存する量なので、計算環境によっては異なる結果になるかもしれません。他方、 $Z_2$  不変量は再現されるはずでず。

## 57 内殻電子の絶対束縛エネルギー：XPSの内殻準位のエネルギー

### 57.1 概要

密度汎関数理論の枠内で金属及び絶縁体の内殻準位の絶対束縛エネルギーを計算できます。ペナルティ汎関数と厳密クーロン遮蔽法に基づき、内殻準位の絶対束縛エネルギーを計算する一般的な方法が実装されています [88]。得られた絶対束縛エネルギーは光電子分光 (X-ray Photoelectron Spectroscopy, XPS) 測定で得られた束縛エネルギーと直接に比較することが出来ます。この方法ではスーパーセル間の内殻ホールによる偽の相互作用は厳密クーロンカットオフ法で回避されます。他方、変分ペナルティ汎関数法により化学シフト、スピン軌道相互作用、交換相互作用による多重分裂を同等に扱うことを可能にします。金属、半導体、絶縁体の8つのケースの内殻電子の絶対束縛エネルギーの計算値は、実験値と比較して平均絶対誤差が 0.4 eV (平均相対誤差は 0.16%) であり、汎用的な計算手法であることが実証されています [88]。

図 78 に模式的に示すように XPS 測定での励起過程におけるエネルギー保存則を考慮し、気体や固体の内殻準位の絶対束縛エネルギーはそれぞれ次式で与えられます [88]。

$$E_b^{(\text{gas})} = E_f^{(0)}(N-1) - E_i^{(0)}(N), \quad (12)$$

$$E_b^{(\text{bulk})} = E_f^{(0)}(N-1) - E_i^{(0)}(N) + \mu_0, \quad (13)$$

ここで、 $E_i^{(0)}(N)$  と  $E_f^{(0)}(N-1)$  はそれぞれ DFT で計算された  $(N-1)$  電子系の始状態と  $N$  電子系の終状態の全エネルギーです。そして、 $\mu_0$  は始状態の計算から得られた化学ポテンシャルです。 $E_f^{(0)}(N-1)$  は文献 [88] で提案された方法で評価されます。他方、 $E_i^{(0)}(N)$  は通常のバンド計算から計算されます。式 (13) はギャップのある系、金属系のいずれでも成り立つ一般的な式ですが、金属では式 (13) は Janak

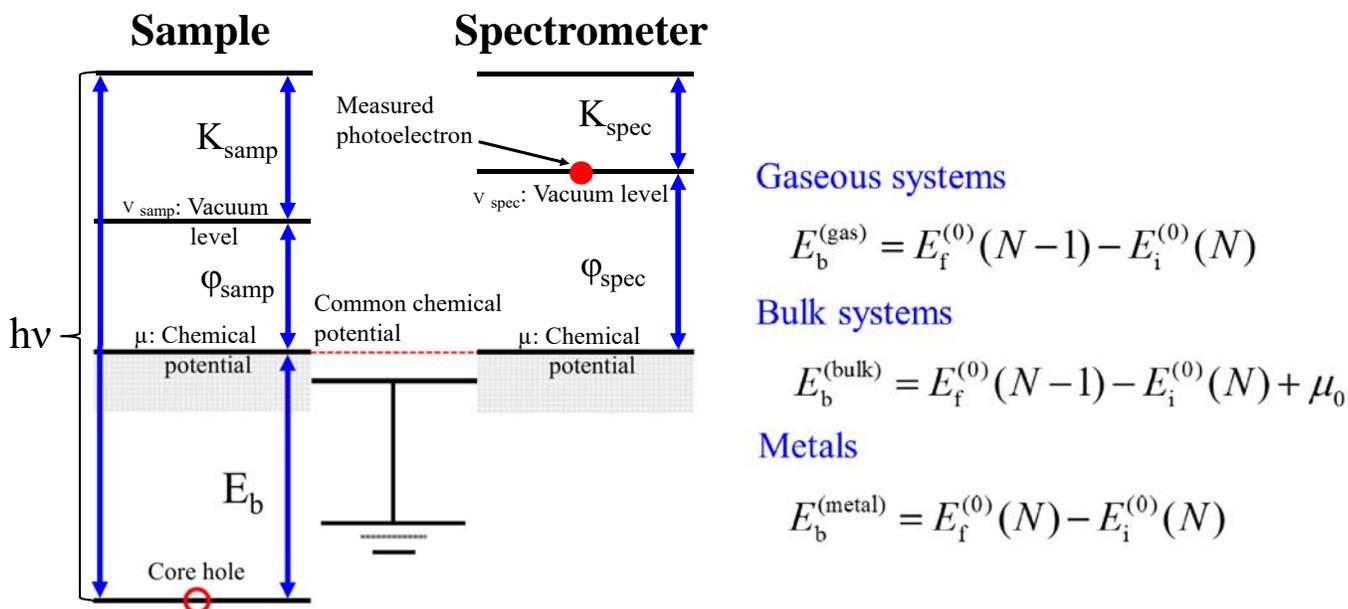


Figure 78: XPS 測定での試料と分光計に対する模式的なエネルギーダイアグラム。気体系、固体系、金属系での内殻準位の絶対束縛エネルギーの定義。

の定理 [89] を用いて以下のように変形できます。

$$E_b^{(\text{metal})} = E_f^{(0)}(N) - E_i^{(0)}(N), \quad (14)$$

終状態の計算に対して  $(N-1)$  電子系の代わりに  $N$  電子系の全エネルギー計算に置き換え可能となります。この表式は金属系の直観的理解によく合致しています。つまり金属系では内殻ホールの生成後の終状態を、完全な遮蔽が生じた状態として理解できることとなります。

## 57.2 気体系

内殻準位の絶対束縛エネルギーの計算方法を計算事例を紹介しながら、以下に説明します。

まず  $\text{C}_2\text{H}_2$  分子の炭素原子の  $1s$  状態の束縛エネルギーの計算を実行します。

始状態計算は以下のように実行されます。

```
% mpirun -np 4 ./openmx C2H2.dat
```

ここで入力ファイル「C2H2.dat」はディレクトリ「work」中に収容されています。始状態計算では以下のキーワードを指定する必要があります。

```
scf.coulomb.cutoff          on          # default = off
scf.SpinPolarization        on or nc  # default = off
```

「scf.coulomb.cutoff=on」の場合、厳密クーロン遮蔽法 [91] を用いて a, b, c 軸の三つの方向のすべてに沿ってスーパーセル間の古典的なクーロン相互作用がカットオフされます。始状態と終状態の計算のどちらの計算においても、キーワード「scf.SpinPolarization」に「on」または「nc」を指定する必要があります。終状態計算では内殻ホールの生成の後に系はスピン分極するため、始状態と終状態の計算の両方で同一のオプションを使用して下さい。内殻準位の絶対束縛エネルギーを計算するため内殻状態を含む擬ポテンシャルを使用して下さい。入力ファイル「C2H2.dat」では以下の擬ポテンシャルが指定されます。

```
<Definition.of.Atomic.Species
H   H7.0-s3p2          H_PBE19
C   C7.0_1s-s4p3d2     C_PBE19_1s
C1  C7.0_1s_CH-s4p3d2 C_PBE19_1s
Definition.of.Atomic.Species>
```

擬ポテンシャル C\_PBE19\_1s.vps は  $1s$ ,  $2s$ ,  $2p$  状態を価電子状態として含みます。C7.0\_1s.pao と C7.0\_1s-CH の基底セットをそれぞれ始状態と終状態の計算に用います。終状態の計算では内殻ホールによって動径波動関数が内殻ホールが無い状態に比べて大幅に変形しています。収束した計算結果を得るためには内殻ホールが存在した状態に対して最適化された基底セットを用いる必要があります。終状態計算の擬ポテンシャルと基底セットは以下のウェブサイトを利用可能です。

[https://t-ozaki.issp.u-tokyo.ac.jp/vps\\_pao\\_core2019/](https://t-ozaki.issp.u-tokyo.ac.jp/vps_pao_core2019/)

本ウェブサイト上では現在七つの元素 B, C, N, O, Si, S, Ge, Pt のみのデータが利用可能です。近い将来に他の元素の擬ポテンシャルと基底関数セットも公開する予定です。

始状態の計算では幾何学構造は以下のように指定されています。

```
Atoms.Number          4
Atoms.SpeciesAndCoordinates.Unit  Ang # Ang|AU
<Atoms.SpeciesAndCoordinates      # Unit=Ang.
  1  C   0.6005  0.000  0.000  3.0 3.0
  2  C  -0.6005  0.000  0.000  3.0 3.0
  3  H   1.8015  0.000  0.000  0.5 0.5
  4  H  -1.8015  0.000  0.000  0.5 0.5
Atoms.SpeciesAndCoordinates>
```

原子1の元素種は「C」で、この原子に対して始状態では C7.0\_1s.pao を基底セットに割り当てます。次に説明する終状態の計算では、原子1に対して 1s 状態の内殻ホールを生成することになります。

終状態 計算は以下のように実行します。

```
% mpirun -np 4 ./openmx C2H2-CH.dat
```

入力ファイル「C2H2-CH.dat」はディレクトリ「work」中に收容されています。終状態の計算では幾何構造は以下のように指定されています。

```
Atoms.Number          4
Atoms.SpeciesAndCoordinates.Unit  Ang # Ang|AU
<Atoms.SpeciesAndCoordinates      # Unit=Ang.
  1  C1  0.6005  0.000  0.000  3.0 3.0
  2  C  -0.6005  0.000  0.000  3.0 3.0
  3  H   1.8015  0.000  0.000  0.5 0.5
  4  H  -1.8015  0.000  0.000  0.5 0.5
Atoms.SpeciesAndCoordinates>
```

原子位置は始状態計算と厳密に同一です。これは励起過程の際での原子緩和は考慮されないことを意味します。原子1の元素種は「C1」で、この原子に対して終状態では基底関数セットとして「C7.0\_1s\_CH.pao」を割り当てていることに注意して下さい。終状態計算では以下のキーワードを指定する必要があります。

```
scf.system.charge      1.0    # default=0.0
scf.coulomb.cutoff     on     # default = off
scf.core.hole          on     # default = off
<core.hole.state
  1 s 1
core.hole.state>
```

分子系の終状態が  $(N - 1)$  電子系であることを考慮し、キーワード「scf.system.charge」で電子数を1つ減らします。次に、スーパーセル間の内殻ホール間の偽の相互作用を回避するため、キーワード「scf.coulomb.cutoff」を用いて厳密クーロンカットオフ法 [91] を適用します。もし、厳密クーロンカットオフ法が適用されない場合には周期的な帯電系でのクーロン発散を避けるため、電荷補償のために反対電荷を持つ様な電荷が自動的に導入されます。この扱いでは偽の相互作用が生じるため、始状態と終状態の間で全エネルギーの直接の比較ができなくなります。

Table 12: コリニア計算とノンコリニア計算における軌道インデックス

|           |                                  |                                   |                                    |                                   |                                |                                   |                                   |                               |
|-----------|----------------------------------|-----------------------------------|------------------------------------|-----------------------------------|--------------------------------|-----------------------------------|-----------------------------------|-------------------------------|
| コリニアの場合   |                                  |                                   |                                    |                                   |                                |                                   |                                   |                               |
| $s$       | 1: $s \uparrow$                  | 2: $s \downarrow$                 |                                    |                                   |                                |                                   |                                   |                               |
| $p$       | 1: $p_x \uparrow$                | 2: $p_y \uparrow$                 | 3: $p_z \uparrow$                  | 4: $p_x \downarrow$               | 5: $p_y \downarrow$            | 6: $p_z \downarrow$               |                                   |                               |
| $d$       | 1:<br>$d_{3z^2-r^2} \uparrow$    | 2:<br>$d_{x^2-y^2} \uparrow$      | 3:<br>$d_{xy} \uparrow$            | 4:<br>$d_{xz} \uparrow$           | 5:<br>$d_{yz} \uparrow$        |                                   |                                   |                               |
|           | 6:<br>$d_{3z^2-r^2} \downarrow$  | 7:<br>$d_{x^2-y^2} \downarrow$    | 8:<br>$d_{xy} \downarrow$          | 9:<br>$d_{xz} \downarrow$         | 10:<br>$d_{yz} \downarrow$     |                                   |                                   |                               |
| $f$       | 1:<br>$f_{5z^2-3r^2} \uparrow$   | 2:<br>$f_{5xz^2-xr^2} \uparrow$   | 3:<br>$f_{5yz^2-yr^2} \uparrow$    | 4:<br>$f_{zx^2-zy^2} \uparrow$    | 5:<br>$f_{xyz} \uparrow$       | 6:<br>$f_{x^3-3xy^2} \uparrow$    | 7:<br>$f_{3yx^2-y^3} \uparrow$    |                               |
|           | 8:<br>$f_{5z^2-3r^2} \downarrow$ | 9:<br>$f_{5xz^2-xr^2} \downarrow$ | 10:<br>$f_{5yz^2-yr^2} \downarrow$ | 11:<br>$f_{zx^2-zy^2} \downarrow$ | 12:<br>$f_{xyz} \downarrow$    | 13:<br>$f_{x^3-3xy^2} \downarrow$ | 14:<br>$f_{3yx^2-y^3} \downarrow$ |                               |
| ノンコリニアの場合 |                                  |                                   |                                    |                                   |                                |                                   |                                   |                               |
| $s$       | 1:<br>$J = 1/2$<br>$M = 1/2$     | 2:<br>$J = 1/2$<br>$M = -1/2$     |                                    |                                   |                                |                                   |                                   |                               |
| $p$       | 1:<br>$J = 3/2$<br>$M = 3/2$     | 2:<br>$J = 3/2$<br>$M = 1/2$      | 3:<br>$J = 3/2$<br>$M = -1/2$      | 4:<br>$J = 3/2$<br>$M = -3/2$     | 5:<br>$J = 1/2$<br>$M = 1/2$   | 6:<br>$J = 1/2$<br>$M = -1/2$     |                                   |                               |
| $d$       | 1:<br>$J = 5/2$<br>$M = 5/2$     | 2:<br>$J = 5/2$<br>$M = 3/2$      | 3:<br>$J = 5/2$<br>$M = 1/2$       | 4:<br>$J = 5/2$<br>$M = -1/2$     | 5:<br>$J = 5/2$<br>$M = -3/2$  | 6:<br>$J = 5/2$<br>$M = -5/2$     |                                   |                               |
|           | 7:<br>$J = 3/2$<br>$M = 3/2$     | 8:<br>$J = 3/2$<br>$M = 1/2$      | 9:<br>$J = 3/2$<br>$M = -1/2$      | 10:<br>$J = 3/2$<br>$M = -3/2$    |                                |                                   |                                   |                               |
| $f$       | 1:<br>$J = 7/2$<br>$M = 7/2$     | 2:<br>$J = 7/2$<br>$M = 5/2$      | 3:<br>$J = 7/2$<br>$M = 3/2$       | 4:<br>$J = 7/2$<br>$M = 1/2$      | 5:<br>$J = 7/2$<br>$M = -1/2$  | 6:<br>$J = 7/2$<br>$M = -3/2$     | 7:<br>$J = 7/2$<br>$M = -5/2$     | 8:<br>$J = 7/2$<br>$M = -7/2$ |
|           | 9:<br>$J = 5/2$<br>$M = 5/2$     | 10:<br>$J = 5/2$<br>$M = 3/2$     | 11:<br>$J = 5/2$<br>$M = 1/2$      | 12:<br>$J = 5/2$<br>$M = -1/2$    | 13:<br>$J = 5/2$<br>$M = -3/2$ | 14:<br>$J = 5/2$<br>$M = -5/2$    |                                   |                               |

キーワード「core.hole.state」により内殻ホールの生成を行う内殻状態を指定します。最初の数はキーワード「Atoms.SpeciesAndCoordinates」に指定される原子の通し番号です。二番目の記号は対象の  $l$  チャンネルを指定し、「 $s$ 」、「 $p$ 」、「 $d$ 」、「 $f$ 」を指定できます。最後の数字は軌道インデックスを指定し、1 から  $4l + 1$  までを指定できます。軌道インデックスと内殻状態の関係は表 12 に与えられています。内殻ホールの生成を行う対象となる内殻状態は価電子状態として擬ポテンシャルに含まれた  $l$  チャンネルの最下位の状態であることに注意してください。C<sub>2</sub>H<sub>2</sub> 分子の場合は、擬ポテンシャル C\_PBE19\_1s.vps は 1s 状態を価電状態として含みます。よって以下の指定は原子 1 の  $|1s \uparrow\rangle$  の状態に対して内殻ホールの生成を行う指定となります。

```
<core.hole.state
  1 s 1
core.hole.state>
```

$s$  状態が対象の状態に選ばれる場合は、「scf.SpinPolarization=on」もしくは「scf.SpinPolarization=nc」が指定可能です。一方、 $p,d,f$  状態を対象とする場合はスピン軌道相互作用を含めるため「scf.SpinPolarization=nc」が指定されなければなりません。

始状態と終状態の計算が終わると、出力ファイルから以下の全エネルギーが得られます。

```
Initial state: -76.787732114928 (Hartree)
```

```
Final state: -66.084858926233 (Hartree)
```

そして、式 (12) を用いて、束縛エネルギーは以下のとおり計算できます。

$$\begin{aligned} E_b^{(\text{gas})} &= E_f^{(0)}(N-1) - E_i^{(0)}(N), \\ &= -66.084858926233 - (-76.787732114928) = 10.702873188695(\text{Hartree}), \\ &\approx 291.24(\text{eV}). \end{aligned}$$

得られた 291.24 eV の値は実験値の 291.14 eV [90] と良く一致しています。計算の他の例題と計算に用いた入力ファイルはウェブサイト [https://t-ozaki.issp.u-tokyo.ac.jp/vps-pao\\_core2019/](https://t-ozaki.issp.u-tokyo.ac.jp/vps-pao_core2019/) で公開されていますので、参考にして下さい。

### 57.3 固体系

ここでは TiC 固体を例として、固体系における内殻準位の絶対束縛エネルギーの計算方法を説明します。始状態の計算は以下のように実行します。

```
% mpirun -np 112 ./openmx TiC216.dat | tee TiC216.std
```

ここで、特別なキーワードは指定しませんが、「scf.SpinPolarization=on」としてスピン分極計算が実行されます。ここでのスピン分極計算は終状態の計算と整合性を取るために実施されています。ただし計算結果は非磁性となります。ここで用いた入力ファイル「TiC216.dat」はディレクトリ「work」に収容されています。

終状態の計算は以下のように実行します。

```
% mpirun -np 112 ./openmx TiC216-CH3.dat | tee TiC216-CH3.std
```

ここで用いた入力ファイル「TiC216-CH3.dat」はディレクトリ「work」に収容されています。この入力ファイル中で原子種は以下のように定義されています。

```
<Definition.of.Atomic.Species
Ti Ti7.0-s3p2d2      Ti_PBE13
C  C6.0_1s-s3p2d1    C_PBE17_1s
C1 C6.0_1s-CH-s3p2d1 C_PBE17_1s
Definition.of.Atomic.Species>
```

そして、以下のように原子種「C1」が原子 5 に割り当てられます。

```

Atoms.Number          216
Atoms.SpeciesAndCoordinates.Unit  Ang # Ang|AU
<Atoms.SpeciesAndCoordinates
  1 Ti  0.000000000000  0.000000000000  0.000000000000  6.0 6.0
  2 Ti  2.163500000000  2.163500000000  0.000000000000  6.0 6.0
  3 Ti  0.000000000000  2.163500000000  2.163500000000  6.0 6.0
  4 Ti  2.163500000000  0.000000000000  2.163500000000  6.0 6.0
  5 C1  2.163500000000  0.000000000000  0.000000000000  3.0 3.0
  6 C   0.000000000000  2.163500000000  0.000000000000  3.0 3.0
  7 C   0.000000000000  0.000000000000  2.163500000000  3.0 3.0
  8 C   2.163500000000  2.163500000000  2.163500000000  3.0 3.0
  ....
  ..
Atoms.SpeciesAndCoordinates>

```

以下のキーワードを指定することで、原子5の1s状態に内殻ホールが生成されます。

```

scf.restart          on
scf.restart.filename TiC216
scf.coulomb.cutoff   on
scf.core.hole        on
scf.system.charge    0.0      # default=0.0

<core.hole.state
  5 s 1
core.hole.state>

```

終状態の計算において Hartree ポテンシャル  $V_H$  は二つの寄与から成ります [88]。次式で定義される周期部分  $V_H^{(P)}$  と非周期部分  $V_H^{(NP)}$  です。

$$V_H(\mathbf{r}) = V_H^{(P)}(\mathbf{r}) + V_H^{(NP)}(\mathbf{r}), \quad (15)$$

周期部分  $V_H^{(P)}$  は始状態計算で得られた電荷密度を用いて周期境界条件でポアソン方程式を介して計算されます。キーワード「scf.restart」と「scf.restart.filename」により始状態計算で得られた電荷密度が指定されます。非周期部分  $V_H^{(NP)}$  は差電荷密度  $\Delta\rho(\mathbf{r}) = \rho_f(\mathbf{r}) - \rho_i(\mathbf{r})$  に対して厳密クーロンカットオフ法 [91] を用いて計算されます。ここで  $\rho_f(\mathbf{r})$  と  $\rho_i(\mathbf{r})$  はそれぞれ終状態と始状態の電荷密度です。クーロン相互作用のカットオフ半径は最も短い格子ベクトルの長さの半分に設定されます。厳密クーロンカットオフ法を有効にするにはキーワード「scf.coulomb.cutoff」を「on」に設定します。内殻ホールはキーワード「scf.core.hole」と「core.hole.state」により生成されます。今回の場合では、原子5の1s状態に内殻ホールが生成されます。TiC 固体は金属であるためキーワード「scf.system.charge」は「0.0」に設定されていることに注意してください。エネルギーギャップのある系を扱う際には「scf.system.charge」を「1」に設定しなければなりません。

始状態と終状態の計算を終えると、出力ファイルから全エネルギーが得られます。

```
Initial state: -10499.900104007471 (Hartree)
```

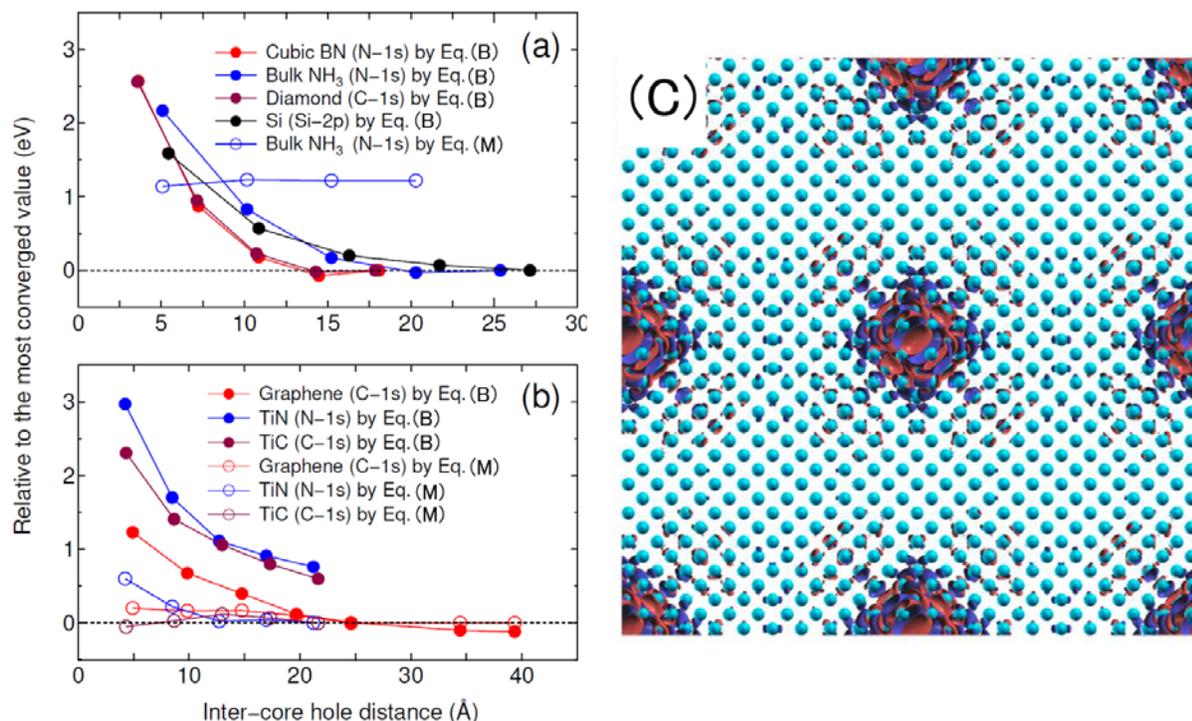


Figure 79: (a) ギャップのある系、(b) 半金属(グラフェン)と金属での内殻ホール間の距離の関数として、最も収束した結果からの相対値として計算された結合エネルギー。(a)と(b)の基準の結合エネルギーは各系の最大の単位胞に対して、それぞれ式(13)と(14)から計算。図中でEq.(B)とEq.(M)はそれぞれ式(13)と式(14)に対応。(c) シリコンでの2p状態での内殻ホールの生成により誘起された差電荷密度。単位胞は1000原子を含み、内殻ホール間距離は27.15 Å。

Final state: -10489.553360141708 (Hartree)

そして、式(14)を用いて、束縛エネルギーが以下のとおり計算できます。

$$\begin{aligned}
 E_b^{(\text{metal})} &= E_f^{(0)}(N-1) - E_i^{(0)}(N), \\
 &= -10489.553360141708 - (-10499.900104007471) = 10.346743865763(\text{Hartree}), \\
 &\approx 281.55(\text{eV}).
 \end{aligned}$$

得られた281.55eVの値は実験値の281.5eV [92]と比較して良く一致していることが分かります。

ギャップのある系の他の例として、固体シリコンの計算を説明します。始状態と終状態の計算を以下のように実行します。

```

% mpirun -np 256 ./openmx Si-4-SOI.dat | tee Si-4-SOI.std
% mpirun -np 256 ./openmx Si-4-CH-SOI1.dat | tee Si-4-CH-SOI1.std
% mpirun -np 256 ./openmx Si-4-CH-SOI6.dat | tee Si-4-CH-SOI6.std

```

「Si-4-SOI.dat」は始状態計算に対する入力ファイル、また「Si-4-CH-SOI1.dat」と「Si-4-CH-SOI6.dat」は終状態計算に対する入力ファイルで、それぞれ $2p_{3/2}(J=3/2, M=3/2)$ と $2p_{1/2}(J=1/2, M=-1/2)$ で指定される内殻準位にホールを導入しています。Si原子の2p状態のスピンの軌道相互作用を考慮するため、以下のキーワードを指定し、ノンコリニア計算を実行します。

```

scf.SpinPolarization      nc          # On|Off|NC
scf.SpinOrbit.Coupling    on          # On|Off, default=off

```

始状態と終状態の計算を終えると、出力ファイルから全エネルギーが得られます。

```

Initial state:             -34820.483255130872 (Hartree)
Final state for SOI1:     -34816.628201335407 (Hartree)
Final state for SOI6:     -34816.601864921540 (Hartree)

```

化学ポテンシャルは始状態計算から得られます。式 (13) を用いて、「Si-4-CH-SOI1.dat」と「Si-4-CH-SOI6.dat」の束縛エネルギーが以下のとおり計算できます。

For  $2p_{3/2}$

$$\begin{aligned}
 E_b^{(\text{bulk})} &= E_f^{(0)}(N-1) - E_i^{(0)}(N-1) + \mu_0, \\
 &= -34816.628201335 - (-34820.483255131) - 0.201641583 = 3.65341221(\text{Hartree}), \\
 &\approx 99.41(\text{eV}),
 \end{aligned}$$

For  $2p_{1/2}$

$$\begin{aligned}
 E_b^{(\text{bulk})} &= E_f^{(0)}(N-1) - E_i^{(0)}(N-1) + \mu_0, \\
 &= -34816.601864922 - (-34820.483255131) - 0.201641583 = 3.6797486(\text{Hartree}), \\
 &\approx 100.13(\text{eV}).
 \end{aligned}$$

$2p_{3/2}$  と  $2p_{1/2}$  に対する 99.41 と 100.13 eV の束縛エネルギー値は実験値の 99.2 と 99.8eV [92] と比較して良い一致が見られます。ここで、 $2p_{3/2}$  と  $2p_{1/2}$  の縮重度は表 12 に記載されるようにそれぞれ 4 と 2 です。ギャップのある系では単位胞サイズに対する絶対束縛エネルギーの収束性は遅いです。図 79(a) と (b) ではそれぞれギャップのある系と金属での相対束縛エネルギーを内殻ホール間距離の関数として示しています。ギャップのある系では収束が遅く、収束値を得るために大きなスーパーセルが必要であることを意味しています。一方、金属では小さな内殻ホール間距離において束縛エネルギーが速やかに収束していることが分かります。図 79(c) に  $2p$  状態への内殻ホールの生成により誘起された固体シリコンでの差電荷密度を示します。内殻ホールにより生成されたポテンシャルを遮蔽するため、内殻ホールから約  $7\text{\AA}$  の周辺まで電荷の再配置が発生します。ギャップのある系で見られた遅い収束は電荷の遠距離に渡る電荷の再配置が原因となっています。他方、金属では電荷の再配置は比較的、狭い範囲に限定され、このため図 79(b) に示したように早い収束を達成されています。

計算の他の例題と計算に用いた入力ファイルはウェブサイト

[https://t-ozaki.issp.u-tokyo.ac.jp/vps\\_pao\\_core2019/](https://t-ozaki.issp.u-tokyo.ac.jp/vps_pao_core2019/) で公開されていますので、参考にして下さい。また、本手法の応用例(シリコン、ポロフェン、グラフェン上に分散されたプラチナ単原子)が文献 [93, 94, 95] で議論されています。

## 57.4 入力ファイルの例

参考のため、三つの入力ファイル例がディレクトリ「work」に収容されています。

- C2H2.dat, C2H2-CH.dat

気体の  $\text{C}_2\text{H}_2$  分子の炭素原子の  $1s$  状態の絶対束縛エネルギーの計算

- TiC216.dat, TiC216-CH3.dat

金属の TiC 固体 (216 原子を含むスーパーセル) の炭素原子の  $1s$  状態の絶対束縛エネルギーの計算

- Si-4-SOI.dat, Si-4-CH-SOI1.dat, Si-4-CH-SOI6.dat

シリコン固体のシリコン原子の  $2p_{3/2}$  と  $2p_{1/2}$  状態の絶対束縛エネルギーの計算

本機能を用いた計算の他の例題と計算に用いた入力ファイルはウェブサイト

[https://t-ozaki.issp.u-tokyo.ac.jp/vps\\_pao\\_core2019/](https://t-ozaki.issp.u-tokyo.ac.jp/vps_pao_core2019/) にありますので参考にして下さい。

## 58 孤立系のイオン化ポテンシャルと電子親和力

デルタ SCF 法と厳密クーロンカットオフ法 [91] により、孤立系のイオン化エネルギーと電子親和力が計算できます。厳密クーロンカットオフ法を用いると周期境界条件においても帯電した孤立系の計算が可能となります。孤立系の基底状態とイオン化状態の二つの計算を実行し、その全エネルギー差からイオン化エネルギーと電子親和力を計算します。まず、孤立系のイオン化エネルギーの計算方法を説明します。最初の例は水分子です。孤立状態の計算を以下のように実行します。

```
% mpirun -np 3 ./openmx H2O+0.dat | tee h2o+0.std
```

入力ファイル「H2O+0.dat」はディレクトリ「work」中に収容されています。幾何構造は本計算の前に同一の計算条件で最適化されました。スーパーセル間の偽の相互作用を回避するため、厳密クーロンカットオフ法 [91] が以下のキーワードで指定されています。

```
scf.coulomb.cutoff          on          # default=off, on|off
```

厳密クーロンカットオフ法を用いてスーパーセル間の偽の相互作用を回避することは可能ですが、クーロン相互作用のカットオフ半径は最も短い格子ベクトルの長さの半分に設定されるため、中心単位胞でのクーロン相互作用が正しく計算できるように単位胞の長さは十分に大きく設定して下さい。

イオン化状態の計算は以下のように実行します。

```
% mpirun -np 3 ./openmx H2O+1.dat | tee h2o+1.std
```

入力ファイル「H2O+1.dat」はディレクトリ「work」中に収容されています。「H2O+0.dat」と比較して、以下のキーワードが変更・追加されています。

```
scf.system.charge          1.0          # default=0.0
scf.coulomb.cutoff          on          # on|off, default=off
scf.SpinPolarization        on          # On|Off|NC
```

キーワード「scf.system.charge」の設定により、孤立系は正に帯電します。「scf.coulomb.cutoff」を設定し、厳密クーロンカットオフ法を用いることで帯電した系のクーロン発散が回避されます。またイオン化することで系はスピン分極する可能性があるため、「scf.SpinPolarization」を「on」に設定します。二つの計算を終えた後に、以下のように出力ファイルから全エネルギーが得られます。

```
Ground state:              -17.477268421216 (Hartree)
Charged state of +1:       -17.010776518028 (Hartree)
```

そして、(+1 に帯電した全エネルギー) - (基底状態の全エネルギー) として定義されたイオン化ポテンシャル IP は以下のように計算されます。

$$\begin{aligned} \text{IP} &= -17.010776518028 - (-17.477268421216) = 0.466491903188(\text{Hartree}), \\ &\approx 12.69(\text{eV}). \end{aligned}$$

得られた 12.69 eV の値は実験値の 12.65 eV [96] と良く一致しています。イオン化ポテンシャルと同じように、(基底状態の全エネルギー) - (-1 に帯電した全エネルギー) として定義された孤立系の電子親和力も、以下の指定で計算できます。

```
scf.system.charge          -1.0          # default=0.0
```

Table 13: 計算された孤立系のイオン化ポテンシャルと電子親和力。すべての幾何構造は計算の前に同一の計算条件で最適化された。本計算に用いた全ての入力ファイルはディレクトリ「work」中に収容されている。

| イオン化ポテンシャル                    |            |          |                        |
|-------------------------------|------------|----------|------------------------|
| 系                             | 実験値 (eV)   | 計算値 (eV) | 入力ファイル                 |
| H <sub>2</sub> O              | 12.65 [96] | 12.69    | H2O+0.dat, H2O+1.dat   |
| C <sub>2</sub> H <sub>2</sub> | 11.43 [97] | 11.47    | C2H2+0.dat, C2H2+1.dat |
| C <sub>2</sub> H <sub>2</sub> | 10.55 [97] | 10.57    | C2H4+0.dat, C2H4+1.dat |
| O <sub>2</sub>                | 12.04 [97] | 12.85    | O2+0.dat, O2+1.dat     |
| CO                            | 14.01 [97] | 13.85    | CO+0.dat, CO+1.dat     |

| 電子親和力           |           |          |                      |
|-----------------|-----------|----------|----------------------|
| 系               | 実験値 (eV)  | 計算値 (eV) | 入力ファイル               |
| OH              | 1.81 [97] | 1.82     | OH-0.dat, OH-1.dat   |
| O <sub>2</sub>  | 0.41 [97] | -0.29    | O2-0.dat, O2-1.dat   |
| Cl <sub>2</sub> | 2.37 [97] | 0.96     | Cl2-0.dat, Cl2-1.dat |
| CN              | 3.88 [97] | 3.51     | CN-0.dat, CN-1.dat   |
| SiH             | 1.27 [97] | 1.17     | SiH-0.dat, SiH-1.dat |

「scf.system.charge=-1.0」を設定し、電子を一つ付加して負に帯電させます。

表 13 に孤立系のイオン化ポテンシャルと電子親和力のベンチマーク計算の結果を示します。イオン化ポテンシャルの計算結果は実験値と比較して、全般的に良く一致していますが、電子親和力の計算値に関しては実験値からの逸脱が見られます。特に O<sub>2</sub> と Cl<sub>2</sub> に対してその誤差が大きくなっています。

## 59 光伝導度と誘電関数

### 59.1 概要

OpenMX Ver. 3.9 では密度汎関数理論と Kubo-Greenwood 公式 [98] に基づき、光学伝導度と誘電関数が計算できます。ボルン近似の範疇で光学伝導度と誘電関数の複素テンソルが電場への線形応答として計算されます。これらの複素テンソルは周波数の依存性を持っています。吸収率、消滅率、透過率、反射率、屈折率はすべて光学伝導度から導かれるため、これらの物理量もまた同時に計算されます。本機能はコリニア計算のみに対応しています。ノンコリニア計算への拡張は近い将来に実装される予定です。また多階層の並列化が実装されているため、超並列計算機を用いれば 1000 原子を含む大規模な系の計算が可能です。

### 59.2 Si の場合

ディレクトリ「work」に收容された入力ファイル「Si2\_k50x50x50.dat」を用いて計算を説明します。この入力ファイルは単位胞に 2 原子を含む Si バルクの計算のためのものであり、SCF 計算と伝導度の計算に対してそれぞれ  $10 \times 10 \times 10$  k 点と  $50 \times 50 \times 50$  k 点が使用されます。計算は以下のように実行します。

```
% mpirun -np 112 ./openmx Si2_k50x50x50.dat
```

SCF 計算を正常に終えた後に光学伝導度の計算が始まります。その際、標準出力にはメッセージ「<Optical calculation start>」が表示されます。

```
***** MD= 1 SCF=17 *****
<Poisson> Poisson's equation using FFT...
<Set_Hamiltonian> Hamiltonian matrix for VNA+dVH+Vxc...
<Band> Solving the eigenvalue problem...
KGrids1: -0.45000 -0.35000 -0.25000 -0.15000 -0.05000 0.05000 0.15000 ....
KGrids2: -0.45000 -0.35000 -0.25000 -0.15000 -0.05000 0.05000 0.15000 ....
KGrids3: -0.45000 -0.35000 -0.25000 -0.15000 -0.05000 0.05000 0.15000 ....
<Band_DFT> Eigen, time=0.028573
<Band_DFT> DM, time=0.024604
  1 Si MulP 2.0000 2.0000 sum 4.0000
  2 Si MulP 2.0000 2.0000 sum 4.0000
Sum of MulP: up = 4.00000 down = 4.00000
              total= 8.00000 ideal(neutral)= 8.00000
<DFT> Total Spin Moment (muB) = 0.000000000000
<DFT> Mixing_weight= 0.020000000000
<DFT> Uele = -2.418066179485 dUele = 0.000000000118
<DFT> NormRD = 0.000000000011 Criterion = 0.000000001000

<Optical calculation start>
CDDF.KGrids1: -0.49000 -0.47000 -0.45000 -0.43000 -0.41000 -0.39000 -0.37000 ....
CDDF.KGrids2: -0.49000 -0.47000 -0.45000 -0.43000 -0.41000 -0.39000 -0.37000 ....
CDDF.KGrids3: -0.49000 -0.47000 -0.45000 -0.43000 -0.41000 -0.39000 -0.37000 ....
<Optical calculations end, time=24.31524 (s)>
<MD= 1> Force calculation
Force calculation #1
Force calculation #2
```

```

Force calculation #3
Force calculation #4
Force calculation #5
<MD= 1> Total Energy
Force calculation #6
....
...

```

標準出力から光学伝導度の計算に要した経過時間は約 24 秒であることが分かります。計算が終わった後に、以下の関連する出力ファイルが得られます。

```

Si2_k50x50x50.cd_re      real part of optical conductivity tensor
Si2_k50x50x50.cd_im      imaginary part of optical conductivity tensor
Si2_k50x50x50.df_re      real part of dielectric function tensor
Si2_k50x50x50.df_im      imaginary part of dielectric function tensor
Si2_k50x50x50.absorption  absorption tensor
Si2_k50x50x50.extinction  extinction tensor
Si2_k50x50x50.transmission transmission tensor
Si2_k50x50x50.reflection  reflection tensor
Si2_k50x50x50.refractive_index refractive index tensor

```

各ファイルの形式はファイルのヘッダー部で分かります。そこに各物理量の単位も記されています。例えば、「Si2\_k50x50x50.cd\_re」に保存された光学伝導度テンソル  $\sigma$  の実部は以下のとおりです。

```

# conductivity tensor (real part) , unit = Siemens/meter = Mho/meter = 1/(Ohm*meter)
# index: energy-grid=1, xx=2, xy=3, xz=4, yx=5, yy=6, yz=7, zx=8, zy=9, zz=10, trace=11
#energy-grid(eV)  xx      xy      xz      yx      yy      yz      zx      zy      zz      (xx+yy+zz)/3
0.00000  16877.3220211  -227.5621843  -227.5697597  -227.5625069  16877.3919038  -227.5042335  -227.5702078  -227.5041190  16877.3911375  16877.3683541
0.00100  16877.3325817  -227.5625199  -227.5700960  -227.5628426  16877.4024628  -227.5045682  -227.5705442  -227.5044537  16877.4016971  16877.3789139
0.00200  16877.3431423  -227.5628556  -227.5704323  -227.5631782  16877.4130218  -227.5049028  -227.5708805  -227.5047883  16877.4122567  16877.3894736
0.00300  16877.3602570  -227.5634100  -227.5709878  -227.5637327  16877.4301338  -227.5054554  -227.5714360  -227.5053409  16877.4293698  16877.4065869
0.00400  16877.3839257  -227.5641832  -227.5717624  -227.5645058  16877.4537989  -227.5062261  -227.5722106  -227.5061116  16877.4530363  16877.4302536
.....
...

```

第一列は光子エネルギー (eV)、第二列から順に、 $\sigma_{xx}$ 、 $\sigma_{xy}$  などのテンソルの要素が記録されています。最後の列は対角要素  $\sigma_{xx}$ 、 $\sigma_{yy}$ 、 $\sigma_{zz}$  の平均値が与えられます。他の出力ファイルも Si2\_k50x50x50.cd\_re と同一の形式に従います。Si2\_k50x50x50.cd\_re, Si2\_k50x50x50.cd\_im, Si2\_k50x50x50.df\_re, Si2\_k50x50x50.df\_im の第一列を横軸、第二列を縦軸としてプロットすると図 80 に示す光学伝導度  $\sigma_{xx}$  と誘電関数  $\epsilon_{xx}$  が得られます。

### 59.3 関連するキーワード

光学伝導度テンソルと誘電関数の計算に関連する各キーワードの仕様は以下のとおりです。

#### CDDF.start

光学伝導度テンソルと誘電関数の計算を実施する場合に「on」と設定します。デフォルト値は「off」です。「off」の場合は伝導度テンソルと誘電関数の計算は行いません。

```
CDDF.start  on      # on|off, default=off
```

[ $\cdot 10^6$ ]

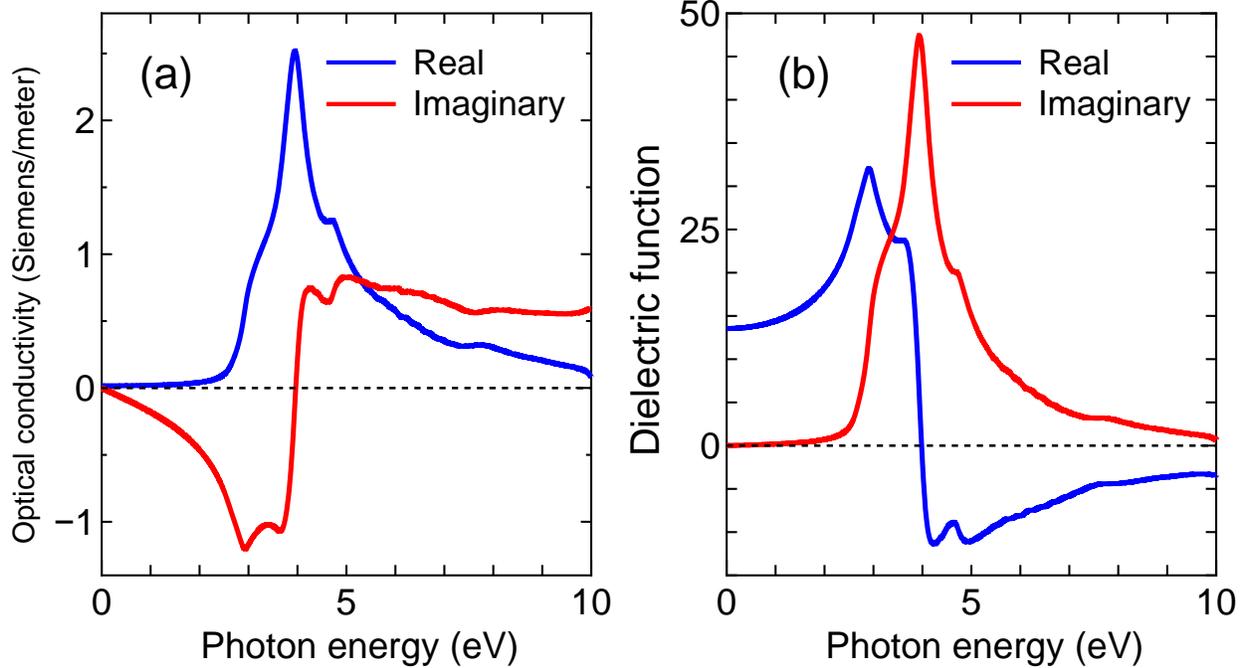


Figure 80: シリコンバルクの (a) 光学伝導度  $\sigma_{xx}$  と (b) 誘電関数  $\epsilon_{xx}$ 。 「Si2\_k50x50x50.cd\_re」, 「Si2\_k50x50x50.cd\_im」, 「Si2\_k50x50x50.df\_re」, 「Si2\_k50x50x50.df\_im」の各ファイルの第二列を垂直軸とし、第一列の関数としてプロット。計算に用いた入力ファイル「Si2\_k50x50x50.dat」はディレクトリ「work」で利用可能。

### CDDF.FWHM

光学伝導度と誘電関数の半値幅を設定します。デフォルト値は eV 単位で 0.2 です。本半値幅がローレンツ関数に適用されます。

```
CDDF.FWHM    0.2    # default = 0.2 (eV)
```

### CDDF.maximum\_energy

光学伝導度と誘電関数を出力する最大エネルギー値を設定します。デフォルト値は eV 単位で 10.0 です。エネルギーの範囲は 0.0 eV から開始します。

```
CDDF.maximum_energy    10.0    # default = 10.0 (eV)
```

### CDDF.additional\_maximum\_energy

計算する周波数領域を拡張するためのエネルギー範囲を設定します。光学伝導度と誘電関数の出力のエネルギー範囲は「CDDF.maximum\_energy」で指定しますが、ローレンツ関数の広がりのためにエネルギー範囲を超えた状態も寄与しうるため、「CDDF.additional\_maximum\_energy」を設定し、出力のエネルギー範囲を超えた状態を考慮します。例えば、「CDDF.maximum\_energy」が 10 eV で、

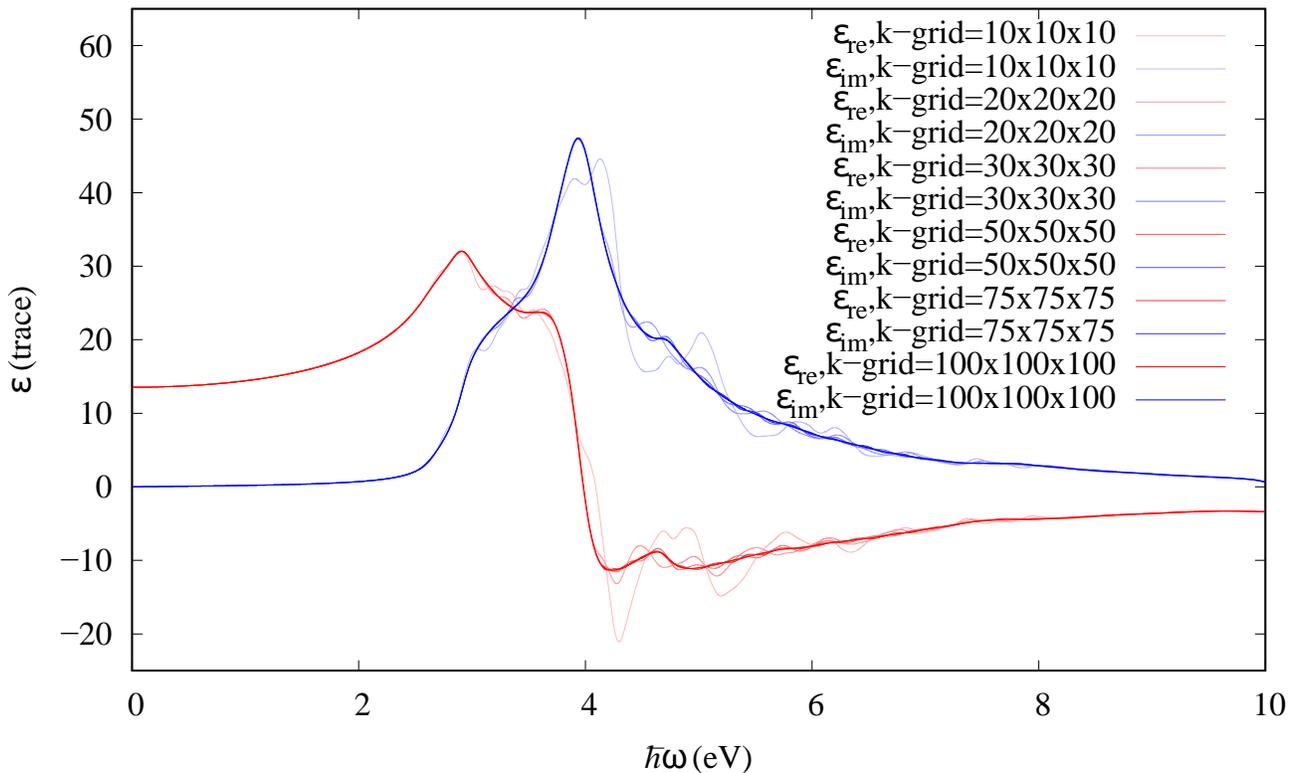


Figure 81: シリコン結晶の誘電関数 (2原子による最小セル)。異なる6種のk格子点を使用:  $10 \times 10 \times 10$ ,  $20 \times 20 \times 20$ ,  $30 \times 30 \times 30$ ,  $50 \times 50 \times 50$ ,  $75 \times 75 \times 75$ ,  $100 \times 100 \times 100$ 。青色と赤色の線はそれぞれ誘電関数の実部と虚部。CDDF.FWHM=0.2 eV を使用。

「CDDF.additional\_maximum\_energy」が 1.0 eV と設定された場合、計算には 11.0 (10.0+1.0) eV のエネルギー範囲が考慮されることとなります。デフォルト値は 0.0 eV です。

```
CDDF.additional_maximum_energy      1.0      # default = 0.0 eV
```

### CDDF.frequency.grid.total\_number

光学伝導度と誘電関数のエネルギーグリッド数を設定します。デフォルト値は 10000 です。またエネルギー間隔は ( Maximum energy - 0.0 ) / エネルギー格子の総数、つまり、(10.0 - 0.0)/10000 = 0.0010 (eV) と与えられます。

```
CDDF.frequency.grid.total_number    10000    # default = 10000
```

### CDDF.Kgrid

第一ブリルアンゾーンの積分グリッド数を3つの数字の組 (n1,n2,n3) で指定します。このグリッドが光学伝導度と誘電関数の計算に用いられます。逆格子ベクトルの  $\tilde{a}$ ,  $\tilde{b}$ ,  $\tilde{c}$  軸がそれぞれ n1, n2, n3 分割され、ブリルアンゾーンの積分が実行されます。「CDDF.Kgrid」は SCF 計算で用いた「scf.Kgrid」と同一である必要はありません。計算コストの削減のため、SCF 計算には粗いグリッドを用い、また光学伝導度と誘電関数の計算には細かいグリッドの使用が推奨されます。

Table 14: シリコン結晶の伝導度と誘電関数の計算時間

| # of Si atoms | Supercell | Diagonalization | k-Grid | Total time (s)<br>(CPUs=128) | Total time (s)<br>(CPUs=256) | Total time (s)<br>(CPUs=512) | Total time (s)<br>(CPUs=1024) | Total time (s)<br>(CPUs=2048) |
|---------------|-----------|-----------------|--------|------------------------------|------------------------------|------------------------------|-------------------------------|-------------------------------|
| 512 atoms     | 4x4x4     | Cluster         | 1x1x1  | 3367.16826                   | 1755.60797                   | 919.21912                    | 464.27761                     | 253.32210                     |
|               |           | ScaLAPACK       | 2x1x1  | 6819.30193                   | 3499.51872                   | 1838.64406                   | 948.87978                     | 513.79250                     |
| 1000 atoms    | 4x4x4     | Band            | 2x2x2  | 15300.58350                  | 10217.17765                  | 5953.19907                   | 3518.84650                    | 1747.20171                    |
|               |           | Cluster         | 1x1x1  |                              |                              | 6900.35370                   | 3511.85143                    | 1778.33693                    |
|               | 5x5x5     | ScaLAPACK       | 2x1x1  |                              |                              | 12994.17818                  | 6817.43990                    | 3460.76787                    |
|               |           | Band            | 2x2x2  |                              |                              | 43676.20392                  | 26055.12739                   | 13318.14587                   |

Table 15: シリコン結晶 (512 原子、 $4 \times 4 \times 4$  スーパーセル、 $1 \times 1 \times 1k$  格子) の伝導度と誘電関数の計算時間。128CPU に対する加速率を最後の列に記載。

| # of CPUs | Total time of calculating<br>conductivity and dielectric function (s) | Ratio of<br>total time by # of CPUs<br>to total time by 128 CPUs |
|-----------|---|--|
| 128       | 3367.16826  | 1.000  |
| 256       | 1755.60797  | 1.918  |
| 512       | 919.21912   | 3.663  |
| 1024      | 464.27761   | 7.252  |
| 2048      | 253.32210   | 13.292   |

## 59.4 ベンチマーク計算

いくつかのベンチマーク計算を以下に示します。

### Si

シリコン固体の誘電関数の実部を図 81 に示します。k 点数に対する誘電関数の依存性を調べるために系統的に k 点数を変化させて計算を実施しました。k 点数を  $10 \times 10 \times 10$  から  $100 \times 100 \times 100$  まで増加させるにつれ、誘電関数の実部が収束していきます。十分に収束した結果を得るには細かい k 点グリッドが必要であることが分かります。また表 14 と表 15 にシリコン固体の光学伝導度と誘電関数の計算時間と並列化効率を示します。1000 CPU コアを利用できる場合には 1000 原子を含む系も計算可能であることを示唆しています。

### $\beta$ -PVDF

$\beta$ -PVDF(ポリフッ化ビニリデン) の誘電関数の実部を図 82 に示します。k 点数に対する誘電関数の依存性を調べるために系統的に k 点数を変化させて計算を実施しました。収束解を得るために  $6 \times 9 \times 21$  の k 点数が必要であることが分かります。表 16 と表 17 に  $\beta$ -PVDF の光学伝導度と誘電関数の計算時間と並列化効率を示します。並列効率に良く、経過時間は 256 個の CPU コアを用いて 1 時間足らずであることが確認されました。参考のため、図 82 に  $\beta$ -PVDF の誘電関数の実部の  $xx$ ,  $yy$ ,  $zz$  成分を示します。

Table 16:  $3 \times 3 \times 5$  のスーパーセルに相当する 540 原子で構成された  $\beta$ -PVDF(ポリフッ化ビニリデン) の光学伝導度と誘電関数の計算時間。

| Diagonalization | k-Grid | Total time (s)<br>(CPUs=128) | Total time (s)<br>(CPUs=256) | Total time (s)<br>(CPUs=512) | Total time (s)<br>(CPUs=1024) | Total time (s)<br>(CPUs=2048) |
|-----------------|--------|------------------------------|------------------------------|------------------------------|-------------------------------|-------------------------------|
| Cluster         | 1x1x1  | 4215.67238                   | 2086.76841                   | 1061.06835                   | 565.92209                     | 329.42010                     |
| ScaLAPACK       | 1x1x2  | 3330.66855                   | 1711.98413                   | 877.61705                    | 599.73266                     | 330.11080                     |
| Band            | 2x2x2  | 12854.63816                  | 7244.26768                   | 3591.33618                   | 1866.03405                    | 998.07756                     |

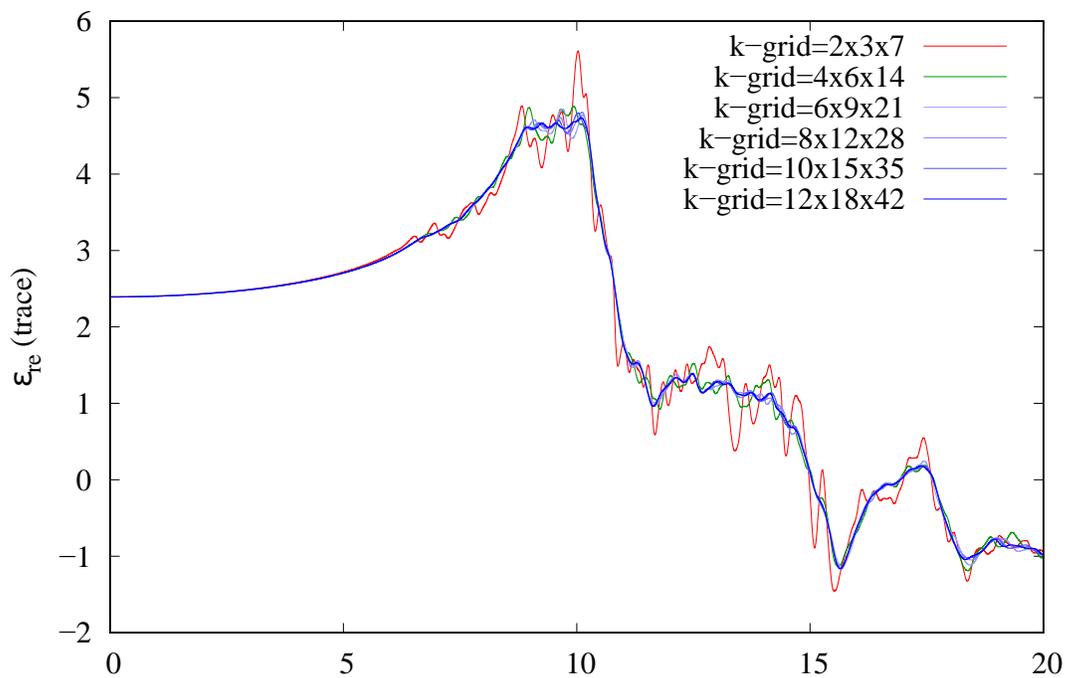


Figure 82:  $1 \times 1 \times 1$  セル (6 原子を含む) の  $\beta$ -PVDF (ポリフッ化ビニリデン) の誘電関数の実部。k 点を系統的に変化させて計算。4 CDFF.FWHM=0.2 eV を使用。

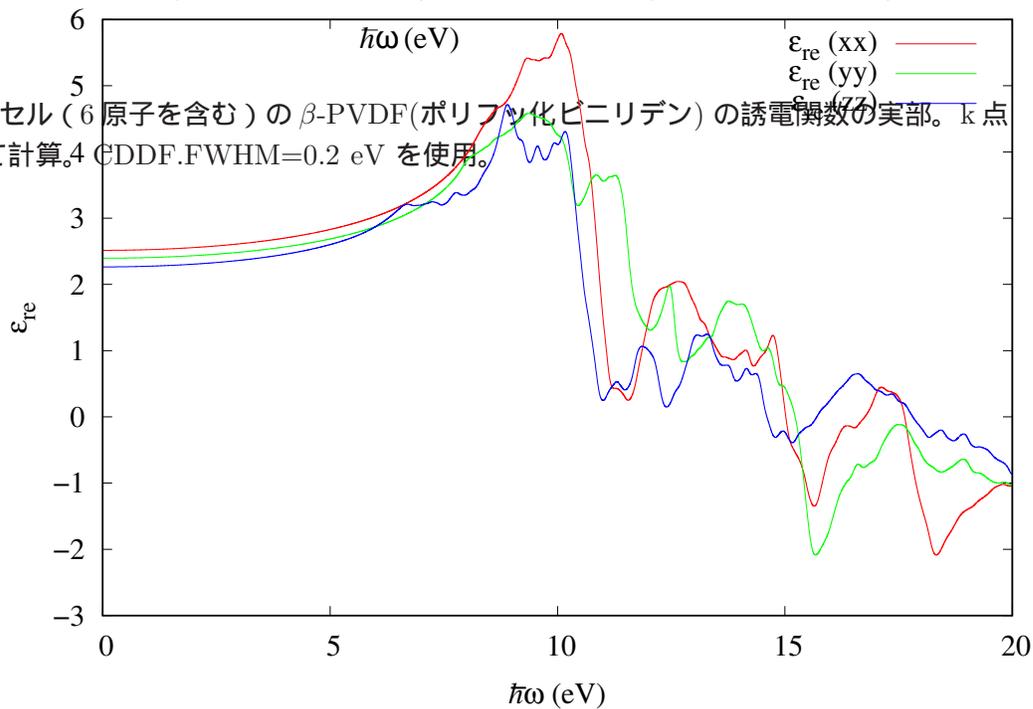


Figure 83:  $\beta$ -PVDF (ポリフッ化ビニリデン) の誘電関数の実部の  $xx$ ,  $yy$ ,  $zz$  成分 (6 原子、 $1 \times 1 \times 1$  セル)。

Table 17:  $3 \times 3 \times 5$  のスーパーセルに相当する 540 原子で構成された  $\beta$ -PVDF(ポリフッ化ビニリデン) の伝導度と誘電関数の計算時間。  $1 \times 1 \times 1$  の k 点を使用。 128CPU コアの結果を基準とした加速率を最後の列に記載。

| # of CPUs | Total time of calculating conductivity and dielectric function (s) | Ratio of total time by # of CPUs to total time by 128 CPUs |
|-----------|--|--|
| 128       | 4215.67238   | 1.000  |
| 256       | 2086.76841   | 2.020  |
| 512       | 1061.06835   | 3.973  |
| 1024      | 565.92209  | 7.449  |
| 2048      | 329.42010  | 12.797   |

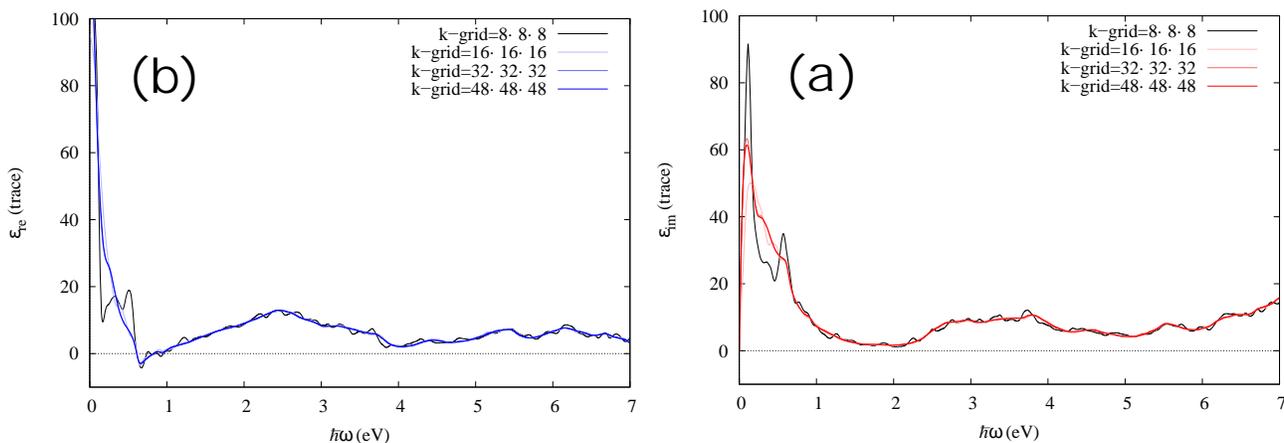


Figure 84:  $\text{VO}_2$  (R 相) の誘電関数の実部と虚部 (6 原子、  $1 \times 1 \times 1$  単位胞)。 k 点を系統的に変化させて計算。

### R 相の $\text{VO}_2$

$\text{VO}_2$  (R 相) の誘電関数の実部を図 84 に示します。 k 点数に対する誘電関数の依存性を調べるために系統的に k 点数を変化させて計算を実施しました。 収束解を得るために  $16 \times 16 \times 16$  の k 点数が必要であることが分かります。 表 18 と表 19 に  $\text{VO}_2$  (R 相) の伝導度と誘電関数の計算時間と並列化効率を示します。 並列効率は良く、 1 時間程度の経過時間で大規模系の計算が実現しています。

### 59.5 コード

光学伝導度と誘電関数の計算に関するルーチンを以下に記載します。

```

Band_DFT_Col_Optical.c      Band_DFT_Col_Optical_ScaLAPACK.c
Band_DFT_NonCol_Optical.c   Calc_optical.c
Cluster_DFT_Optical.c       Cluster_DFT_Optical_ScaLAPACK.c
DFT.c                        Get_Cnt_dOrbitals.c
Get_dOrbitals.c              Input_std.c

```

Table 18: R 相の VO<sub>2</sub> の伝導度と誘電関数の計算時間 (R 相、384 原子、4 × 4 × 4 スーパーセル)

| Diagonalization | k-Grid | Total time (s)<br>(CPUs=128) | Total time (s)<br>(CPUs=256) | Total time (s)<br>(CPUs=512) | Total time (s)<br>(CPUs=1024) | Total time (s)<br>(CPUs=2048) |
|-----------------|--------|------------------------------|------------------------------|------------------------------|-------------------------------|-------------------------------|
| Cluster         | 1x1x1  | 5328.56778                   | 2719.77511                   | 1382.56556                   | 704.32679                     | 360.64294                     |
| ScaLAPACK       | 1x1x2  | 5276.74287                   | 2755.58509                   | 1395.48266                   | 718.42770                     | 368.24621                     |
| Band            | 2x2x2  | 21031.96431                  | 10686.37446                  | 5586.12815                   | 2781.05698                    | 1431.91243                    |

Table 19: R 相の VO<sub>2</sub> の伝導度と誘電関数の計算時間 (R 相、384 原子、4 × 4 × 4 スーパーセル、1 × 1 × 1k 格子) 128CPU コアの結果を基準とした加速率を最後の列に記載。

| # of CPUs | Total time of calculating<br>conductivity and dielectric function (s) | Ratio of<br>total time by # of CPUs<br>to total time by 128 CPUs |
|-----------|---|--|
| 128       | 5328.56778  | 1.000  |
| 256       | 2719.77511  | 1.959  |
| 512       | 1382.56556  | 3.854  |
| 1024      | 704.32679   | 7.565  |
| 2048      | 360.64294   | 14.775   |

NabraMatrixElements.c

Set\_dOrbitals\_Grid.c

## 59.6 実例

参考のため、五つの入力ファイル例がディレクトリ「work/cddf\_examples」に収容されています。

- Febcc-Col\_k30x30x30.dat  
鉄 (BCC 構造) の光学伝導度と誘電関数の計算。30 × 30 × 30 k 点。
- Mn12.dat  
Mn12 分子磁石の光学伝導度と誘電関数の計算。1 × 1 × 1 k 点。
- Si2\_k10x10x10.dat  
シリコン固体の光学伝導度と誘電関数の計算。10 × 10 × 10 k 点。
- Si2\_k1xk1xk1.dat  
シリコン固体の伝導度と誘電関数の計算。1 × 1 × 1 k 点。
- VO2R-k16xk16x16.dat  
VO<sub>2</sub> (R 相) の光学伝導度と誘電関数の計算。16 × 16 × 16 k 点。

## 59.7 自動動作テスト

光学伝導度と誘電関数の計算の機能が正しくインストールされた否かを検査するため、自動動作テストを以下のように実施できます。

### MPI 並列での動作

```
% mpirun -np 112 ./openmx -runtestCDDF
```

## MPI/OpenMP 並列での動作

```
% mpirun -np 56 ./openmx -runtestCDDF -nt 2
```

OpenMX は五つのテストケースを実行し、work/cddf\_example に収容された参照結果と計算結果を比較します。比較 (誘電関数の差の絶対値) はディレクトリ「work」中の「runtestCDDF.result」に記録されます。参照結果は Xeon クラスタ計算機を用いて計算されたものです。差が 0.1 以下ならば、インストールは正常と考えられます。

## 60 BoltzTraP とのインターフェース

OpenMX Ver. 3.9 は BoltzTraP とのインターフェースを提供します。BoltzTraP [99] は Kohn-Sham 方程式のエネルギー固有値の波数依存性から、ボルツマンの理論に基づき電子輸送係数を計算するプログラムです。このインターフェース [100] を利用することで、ゼーベック係数、電子伝導度、電子熱伝導度、ホール係数などの物理量の計算が可能になります。この機能はコリニア計算のみならずノンコリニア計算にも対応しています。本機能を用いて発表を行う際には、文献 [100] を引用して頂けると幸いです。BoltzTraP2 とのインターフェースは将来のリリースで公開予定です。

OpenMX と BoltzTraP を橋渡しするインターフェースを利用するには、ディレクトリ「source」に収容された MX\_TRAP.sh をディレクトリ「work」にコピーして下さい。例として、非ドープ型シリコン固体の計算を紹介します。ディレクトリ「work」中の入力ファイル「Si\_BoltzTraP.dat」を用いて、SCF 計算を以下のとおり実行します。

```
% mpirun -np 28 ./openmx Si_BoltzTraP
```

SCF 計算が正常に終了した後に、出力ファイル「Si\_BoltzTraP.out」が得られます。その後で、MX\_TRAP.sh を以下のように実行します。

```
% sh MX_TRAP.sh
```

すると、現在のディレクトリにある out ファイルの名前が以下のように一覧で表示されます。

```
=====Outputfile List=====
Si_BoltzTraP.out
=====
```

```
Please enter the outputfile name ; Outputfile name =
```

「Outputfile name =」の後に、out ファイルのファイル名を入力し、Enter キーを押して下さい。計算例である「Si\_BoltzTraP.out」を以下のように入力してください。

```
Please enter the output file name: Outputfile name = Si_BoltzTraP.out
```

Enter キーを押すと、以下のメッセージが表示されます。

```
..
....
Nospin::kloop kx ky kz:1681/1686
Nospin::kloop kx ky kz:1682/1686
Nospin::kloop kx ky kz:1683/1686
Nospin::kloop kx ky kz:1684/1686
Nospin::kloop kx ky kz:1685/1686
Nospin::kloop kx ky kz:1686/1686
.energy file for BoltzTraP has been generated.

.struct file for BoltzTraP has been generated.
```

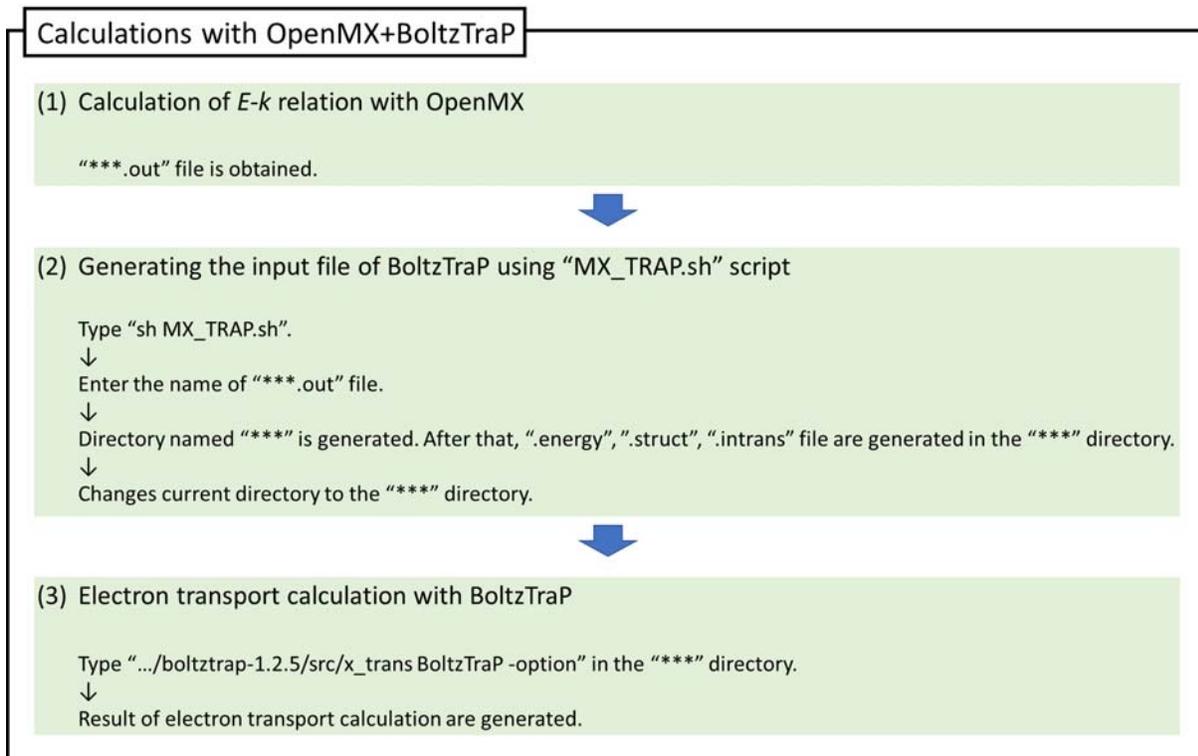


Figure 85: OpenMX と BoltzTraP による電子輸送計算の計算手順

.intrans file for BoltzTraP has been generated.

Conversion has been finished.

Directory is Si\_BoltzTraP

上記のメッセージが表示された後に、「Si\_BoltzTraP」と名付けられたディレクトリが生成されます。以下のように四つのファイル「Si\_BoltzTraP.out」、 「Si\_BoltzTraP.energy」、 「Si\_BoltzTraP.intrans」、 「Si\_BoltzTraP.struct」がディレクトリ「Si\_BoltzTraP」中に保存されます。

```
% cd Si_BoltzTraP
% ls
Si_BoltzTraP.out Si_BoltzTraP.energy Si_BoltzTraP.struct Si_BoltzTraP.intrans
```

「.out」ファイルは変換で得られた出力ファイルのコピーです。「.energy」ファイルはエネルギー固有値のファイルです。もし「.out」ファイルでのキーワード「scf.SpinPolarization」が「ON」ならば、「.energyup」と「.energydn」の二つのファイルが出力されます。他方、「.out」ファイルでのキーワード「scf.SpinPolarization」が「NC」ならば、「.energyso」のファイルが出力されます。「.struct」のファイルは単位格子ベクトルのファイルです。単位は自動的に原子単位系に変換されます。「.intrans」のファイルは電子輸送計算に必要な入力パラメタを記述します。詳しくは、BoltzTraP パッケージに付属の PDF マニュアルを参照してください。このファイルの第三行の第一項の数値の値は OpenMX で計算された化学ポテンシャル  $\mu$  です (単位は Ryd)。第五行にある 10 の値はバンドのフーリエ補間の係数で、既定で

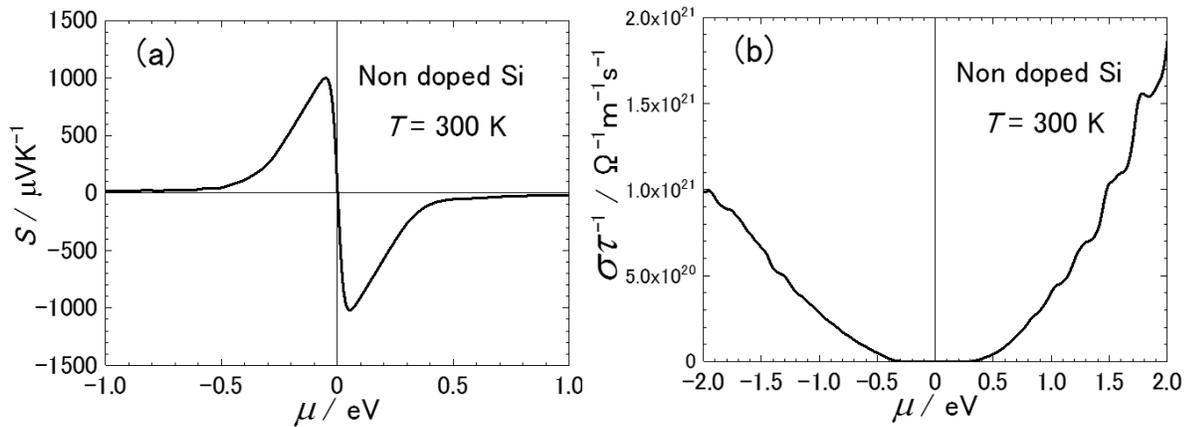


Figure 86: OpenMX と BoltzTraP によって計算されたダイヤモンド構造の非ドーブ型シリコン固体の (a) ゼーベック係数  $S$ , (b) 電子伝導率  $\sigma\tau_{\text{el}}^{-1}$ 。横軸は化学ポテンシャル。温度は 300K で、フーリエ補間の係数は 25 に設定。計算に用いた入力ファイル「Si\_BoltzTraP.dat」はディレクトリ「work」中に収容。

は 10 に設定されます。第八行の数値の値はフェルミ分布の電子温度です。第一項は最大温度です。第二項は温度依存性を計算する際の温度間隔です。もし温度依存性を計算したい場合には、第一項と同一の値を入力してください。既定では、「.out」ファイルの「scf.ElectronicTemperature」で指定された温度が入力されています。もし「scf.ElectronicTemperature」が指定された場合は、 $T = 300K$  が割り当てられます。

次に、ディレクトリ「Si\_BoltzTraP」に移動して、以下のとおり BoltzTraP とタイプしてください。これ以降の手順は通常の BoltzTraP 計算と同一です。それゆえ、詳細は BoltzTraP パッケージに付属の PDF マニュアルを参照してください。

```
% cd Si_BoltzTraP
% 'path to BoltzTrap'/boltztrap-1.2.5/sr/x_trans BoltzTraP
```

エネルギー固有値を保存されたファイルが「.energyup」、「.energydn」、「.energyso」の際には x\_trans BoltzTraP の後にそれぞれ「-up」、「-dn」、「-so」をオプションとして付加して実行して下さい。オプションで指定されたエネルギー固有値ファイルに対して電子輸送計算が実施されます。計算が正常に完了すると、以下のメッセージが表示されます。

```
===== BoltzTraP vs 1.2.5 =====
99.786 u 0.076 s 1: 40.04 99.8\% 0 + 0 k 0 + 1612 0 io 0 pf + 0 w
```

$x, y, z$  軸上での電子輸送特性の平均値が「Si\_BoltzTraP.trace」に記録されます。他方、 $x, y, z$  軸の電子輸送特性は「Si\_BoltzTraP.condtens」に記録されます。図 85 に OpenMX と BoltzTraP による電子輸送計算の計算手順を記載します。図 86 は非ドーブ型シリコン固体のゼーベック係数  $S$  と電子伝導度  $\sigma\tau_{\text{el}}^{-1}$  の化学ポテンシャル  $\mu$  依存性を示します。計算に用いた入力ファイル「Si\_BoltzTraP.dat」はディレクトリ「work」中に収容されています。参考文献 [100] の補足資料には本機能を用いた大規模なベンチマーク計算の結果が記載されています。このベンチマーク計算に含まれている GaCuS<sub>2</sub> 固体の計算結果を図 87 に示します。この計算ではスピン軌道相互作用を考慮し、ノンコリニア計算を実行しました。

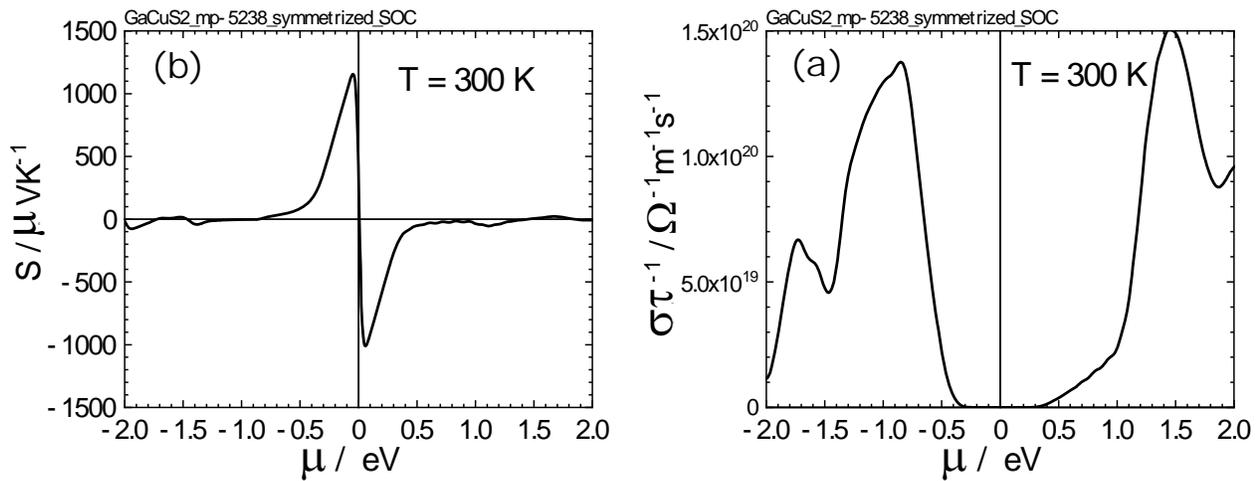


Figure 87: OpenMX と BoltzTraP によって計算された GaCuS<sub>2</sub> 固体の (a) ゼーベック係数  $S$ , (b) 電子伝導率  $\sigma\tau_{\text{el}}^{-1}$ 。横軸は化学ポテンシャル。温度は 300K で、フーリエ補間の係数は 10 に設定。計算に用いた入力ファイル「GaCuS2\_mp-5238\_symmetrized\_SOC.dat」はディレクトリ「work」中に収容。

本計算に用いた入力ファイル「GaCuS2\_mp-5238\_symmetrized\_SOC.dat」はディレクトリ「work」中に収容されています。

## 61 “格子定数 vs. エネルギー” 曲線の計算

### 61.1 “格子定数 vs. エネルギー” 曲線

次のキーワードを設定するにより” 格子定数 vs. エネルギー” 曲線を計算することができます。

|               |       |   |
|---------------|-------|---|
| MD.Type       | EvsLC | #   |
| MD.EvsLC.Step | 0.4   | # default=0.4%                            |
| MD.maxIter    | 32    | # default=1                               |
| MD.EvsLC.flag | 1 1 1 | # default=1 1 1                           |
|               |       | # (0: fixed, 1:expansion, -1:contraction) |

キーワード「MD.Type」を「EvsLC」に設定すると、単位胞ベクトル  $a$ 、 $b$ 、 $c$  を逐次的に変化させながら全エネルギーを計算することができます。単位胞ベクトルの変更は「MD.EvsLC.Step」で指定される一定比率で初期ベクトルを様に拡大・収縮させることで実施されます。計算ステップ数はキーワード「MD.maxIter」で指定します。もし格子ベクトルの一部を固定したい場合にはキーワード「MD.EvsLC.flag」を使用します。既定の設定は「1 1 1」であり、 $a$ 、 $b$ 、 $c$  軸のそれぞれの軸で様に拡大・収縮を実施します。このフラグを「0」に設定すると、対応する軸は初期値に固定されます。また「-1」に設定した場合には一様な収縮を意味します。計算が正常に終了すると、ファイル「*System.Name.EvsLC*」が生成されます。計算途中でもファイル「*System.Name.EvsLC*」には途中までの計算結果が出力されます。ここ

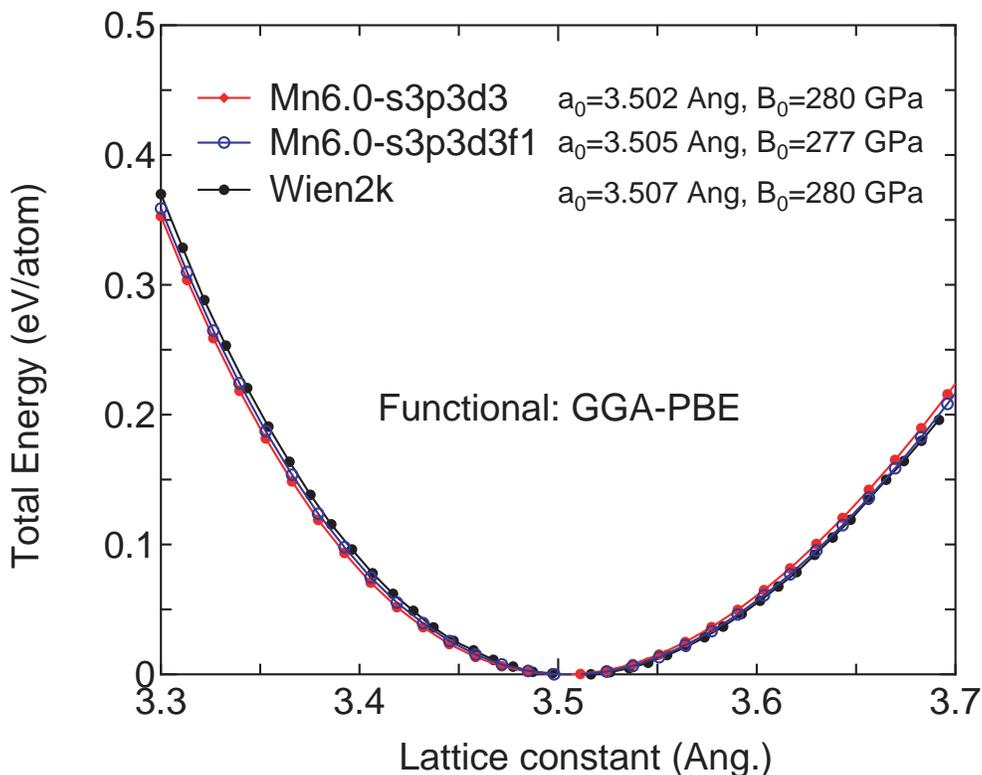


Figure 88: キーワード「EvsLC」によって計算された fcc 構造の Mn 固体の “格子定数 vs. エネルギー” 曲線。使用した入力ファイルはディレクトリ「work」中の「Mnfcc-EvsLC.dat」。図中の平衡格子定数と体積弾性率は、ウェブサイト [148] で提供されている「murn.f」コードを用い、Murnaghan 状態方程式にデータをフィッティングすることによって評価。



## 62 フェルミ面

XCrySDen [105] や FermiSurfer [143, 144] を用いてフェルミ面を可視化できます。まず、次のキーワードを設定して状態密度の計算を実行して下さい。

```
Dos.fileout           on           # on|off, default=off
Dos.Erange            -20.0  20.0        # default = -20 20
Dos.Kgrid             61 61 61     # default = Kgrid1 Kgrid2 Kgrid3
FermiSurfer.fileout   on           # default = off, on/off
```

計算が正常に終了すると、ファイル「*System.Name.FermiSurf0.bxsf*」が得られます。このファイルは XCrySDen[105] で読み込み可能です。ファイル「*System.Name.FermiSurf\_s0\_aA.frmsf*」は FermiSurfer [143, 144] で読み込み可能です。ここで、「*System.Name*」は「*System.Name*」であり、*A* は原子のシリアル番号です。FermiSurfer に対応するファイルを得るにはキーワード「FermiSurfer.fileout」を「on」に設定して下さい。「Dos.Fileout」と同様に、この目的のために「DosGauss.fileout」も使用可能ですが、FermiSurfer に対応するファイルは「DosGauss.fileout」の場合には生成されませんので注意して下さい。

スピン分極計算の場合には、「*System.Name.FermiSurf0.bxs*」と「*System.Name.FermiSurf1.bxs*」が生成され、それぞれ up スピンと down スピンに対応しています。また FermiSurfer を用いて各原子の寄与を解析するために、「*System.Name.FermiSurf\_s0\_aA.frmsf*」と「*System.Name.FermiSurf\_s1\_aA.frmsf*」が生成され、それぞれ up スピンと down スピンに対応しています。これらのファイルを FermiSurfer を用いて解析することで、各原子からの寄与を色分けしてプロットできます。例えば、原子 1 の原子軌道の特性をプロットするにはファイル「*System.Name.FermiSurf\_s0\_a1.frmsf*」を読み込みます。ノンコリニア計算の場合、ファイル「*System.Name.FermiSurf.bxs*」と「*System.Name.FermiSurf\_aA.frmsf*」が生成されます。また滑かなフェルミ面を描画するには、多数の k 点が必要になることに注意して下さい。例として、fcc 構造の Ca 固体のフェルミ面を図 89 に示します。この計算に使用された入力ファイルは、ディレクトリ「work」中の「Cafcc\_FS.dat」です。

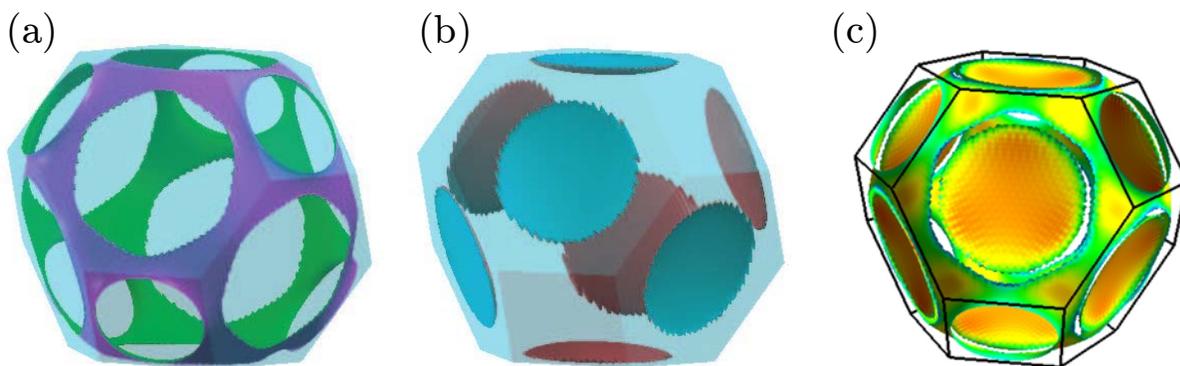


Figure 89: XCrySDen[105] で可視化した fcc 構造の Ca 固体のフェルミ面。2 種類のバンドがフェルミエネルギー（化学ポテンシャル）を横切るため、2 種類のフェルミ面を (a) と (b) に表示。(c) FermiSurfer [143, 144] で可視化されたフェルミ速度の色分けプロット。計算に使用された入力ファイルはディレクトリ「work」中の「Cafcc\_FS.dat」。

## 63 2つの Gaussian Cube ファイルの差の解析

ユーティリティツールによって、2つの Gaussian cube ファイルの差を解析することができます。構造変化やスピン磁気モーメント方位の変化に伴い、どのように全電子密度、スピン密度、ポテンシャルが変化しているのか解析することが可能です。

### (1) diff\_gcube.c のコンパイル

ディレクトリ「source」内に、「diff\_gcube.c」という名のファイルが保存されています。このファイルを次のコマンドでコンパイルします。

```
% gcc diff_gcube.c -lm -o diff_gcube
```

コンパイルが正常に終了すると、ディレクトリ「source」内に、実行可能なファイル「diff\_gcube」が生成されます。この実行可能ファイルをディレクトリ「work」にコピーして下さい。

### (2) 差の計算

「input1.cube」と「input2.cube」という名前の2つの Gaussian cube ファイルの差を求め、さらにその結果をファイル「output.cube」に出力したい場合、「diff\_gcube」を次のように実行します。

```
% ./diff_gcube input1.cube input2.cube output.cube
```

差は Gaussian cube 形式で「output.cube」という名前のファイルに出力されます。「output.cube」は、XCrySDen [105]、VESTA[103] や Molekel [104] 等の可視化ソフトウェアを用いて、容易に解析できます。実際に、「電場」の章の図 28 はこの方法で作成されました。

## 64 2つの幾何構造の差の解析

ユーティリティツールによって、3次元座標（デカルト座標）を保存している2つのxyzファイルの差を解析することができます。電場や置換基の影響により、どのように構造が変化しているのか解析することが可能です。幾何構造の差の解析には以下で説明する3種類の解析方法が利用できます。

2つの幾何構造間の標準偏差（RMSD）は次式で定義されます。

$$\text{RMSD} = \sqrt{\frac{\sum_i^{N_{\text{atom}}} (R_i - R_i^0)^2}{N_{\text{atom}}}}$$

2つの幾何構造間の平均偏差（MD）は次式で定義されます。

$$\text{MD} = \frac{\sum_i^{N_{\text{atom}}} |R_i - R_i^0|}{N_{\text{atom}}}$$

2つの幾何構造間の結合距離の平均偏差（MDBL）は次式で定義されます。

$$\text{MDBL} = \frac{\sum_i^{N_{\text{bond}}} |BL_i - BL_i^0|}{N_{\text{bond}}}$$

ここで、 $N_{\text{atom}}$  と  $N_{\text{bond}}$  は、それぞれ原子数、カットオフ半径内の結合長（BL）を持つ結合数です。また、それぞれの原子のxyz座標間の偏差ベクトルは、XCrySDen形式で「dgeo.vec.xsf」という名前のxsfファイルに出力されますので、2つの構造間の差を可視化できます。

以下に解析の手順を説明します。

### (1) diff\_gcube.c のコンパイル

ディレクトリ「source」内にファイル「diff\_geo.c」があります。このファイルを次のコマンドでコンパイルします。

```
% gcc diff_geo.c -lm -o diff_geo
```

コンパイルが正常に終了すると、ディレクトリ「source」内に実行ファイル「diff\_geo.c」が生成します。この実行ファイルをディレクトリ「work」にコピーして下さい。

### (2) 2つの幾何構造間の差の計算

diff\_geo.c のヘッダー部分に次のような「使用方法 (usage)」が記載されています。

```
usage:
    ./diff_geo file1.xyz file2.xyz -d rmsd

option
    -d rmsd      a root mean square of deviation
    -d md        a mean deviation
    -d mdbl 2.2  a mean deviation between bond lengths,
                 2.2 (Ang) means a cutoff bond length which
                 can be taken into account in the calculation
```

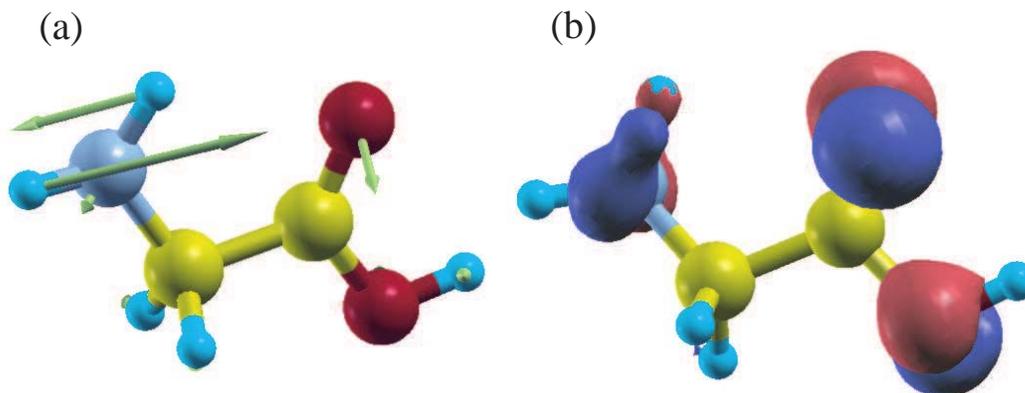


Figure 90: (a) 最適化構造間の原子座標の差に対応するベクトル。中性グリシン分子と電子を 1 個付加したグリシン分子の構造最適化をそれぞれ行い、最適化構造間の比較を実施。(b) 2 つの系の全電子密度の差。青色と赤色は、それぞれ全電荷密度の減少と増加を示す。図は XCrySDen を使って可視化。

2 つの幾何構造間の RMSD を知りたければ、次のように実行します。

```
% ./diff_geo file1.xyz file2.xyz -d rmsd
```

計算結果は標準出力に表示されます。また XCrySDen のベクトル形式で、各原子の 3 次元座標の差がファイル「dgeo\_vec.xsf」に保存されます。このファイルは XCrySDen 中の「Display→Forces」を使用して、可視化できます。MDBL を計算する場合には、カットオフ結合距離 (Å) を与えて下さい。RMSD 計算ではカットオフ半径内にある結合距離が考慮されます。図 90 (a) に最適化構造間の原子座標の差に対応するベクトルを示します。この計算では中性グリシン分子と電子を 1 個付加したグリシン分子の構造最適化をそれぞれ行い、最適化構造間の比較を行いました。また図 90 (b) に 2 つの系の全電子密度の差を示します。大きな電荷の変化に伴って、大きな構造変化が生じていることが分かります。この例から、電荷のドーピングや静電場によって構造変化がどのように起こるのかを知るために、このツールが有用であることが分かります。

## 65 相互作用によって引き起こされる電荷密度の差の解析

2つの系 A と B の間の相互作用によって引き起こされる（スピン）電子密度の再配分は、次の手順によって解析できます。

### (i) A と B から成る複合系の計算

A と B から成る複合系に対して、通常の計算を行い、（スピン）電子密度のための cube ファイルを生成します。このファイルを「AB.cube」とします。また標準出力から「Grid\_Origin」の情報を取得します。これは一様グリッドを生成する際の原点の  $x$ 、 $y$ 、 $z$  成分となります。

```
Grid_Origin xxx yyy zzz
```

この値は次の (ii) と (iii) で議論する計算において使用します。

### (ii) 系 A の計算

A のみを含んだ系の通常の計算を行います。ただし (i) の計算と同一の単位胞を使用して下さい。また計算条件も同一に設定する必要があります。また系 A の幾何構造は、計算 (i) の場合と同一でなければなりません。一様グリッドを生成する際の原点の  $x$ 、 $y$ 、 $z$  成分も (i) と同一に設定するために、入力ファイル中に次のキーワードを加えて下さい。

```
scf.fixed.grid xxx yyy zzz
```

ここで、「xxx yyy zzz」は (i) の「Grid\_Origin」から取得した座標です。このようにして、系 A に対する cube ファイルが得られます。これを「A.cube」とします。

### (iii) 系 B の計算

(ii) の計算と同様にして、系 B のみの計算も A と B から成る複合系の場合と同じ単位胞を使用し、また同一の計算条件で実行する必要があります。また、系 B の幾何構造は (i) の場合と同一でなければなりません。一様グリッドを生成する際の原点の  $x$ 、 $y$ 、 $z$  成分も (i) と同一に設定するために、入力ファイル中に次のキーワードを加えて下さい。

```
scf.fixed.grid xxx yyy zzz
```

ここで、「xxx yyy zzz」は (i) の「Grid\_Origin」から取得した座標です。このようにして、系 B に対する cube ファイルが得られます。これを「B.cube」とします。

### (iv) 2つのコードをコンパイルする

2つのコードを次のようにしてコンパイルします。

```
% gcc diff_gcube.c -lm -o diff_gcube
% gcc add_gcube.c -lm -o add_gcube
```

(v) (スピン) 電子密度の差を解析するための cube ファイルの作成

まず、次のコマンドで、系 A と B の 2 つの (スピン) 電子密度の重ねあわせの cube ファイルを作成します。

```
% ./add_gcube A.cube B.cube A_B.cube
```

ファイル「A\_B.cube」は 2 個の孤立系の (スピン) 電子密度の重ねあわせの cube ファイルです。次に、相互作用によって引き起こされる (スピン) 電子密度の差の cube ファイルを、次のようにして作成します。

```
% ./diff_gcube AB.cube A_B.cube dAB.cube
```

ファイル「dAB.cube」は、相互作用によって引き起こされる (スピン) 電子密度の差の cube ファイルであり、ここで「差」は  $(AB - A_B)$  を意味します。

## 66 セルサイズの自動決定

OpenMX では Poisson 方程式の解法は FFT を使用しています。そのため、孤立系を計算する場合でも周期境界条件のためにコピーされたイメージが周期的に配置されることになります。孤立系であることを保証するためには最低限、中心セルと隣接セル内の基底関数が重ならないようにスーパーセルを設定することが必要です。この条件の下、スーパーセルを設定すればバンド構造においてバンド分散は生じません。ただし古典的なクーロン相互作用は上記の条件下でスーパーセルを設定したとしても残っていることに注意して下さい。適切なスーパーセルはユーザー自身で設定する必要がありますが、スーパーセルの設定を自動で行うことも可能です。自動設定では、基底関数のカットオフ半径を考慮して、最小のセルサイズが決定されます。自動設定を行う場合には、キーワード「Atoms.UnitVectors」を入力ファイル中でコメントアウトして下さい。この場合、OpenMX は中心セルの基底関数と隣接セル内の基底関数が重ならないように、適切なセルサイズを自動的に決定し、また要求されるカットオフエネルギーを実現します。決定されたセルベクトルは、次のように標準出力に表示されます。この値を参考にユーザー自身でスーパーセルを設定仕直すことも可能でしょう。

```
<Set_Cluster_UnitCell> automatically determined UnitCell(Ang.)
<Set_Cluster_UnitCell> from atomic positions and Rc of PAOs (margin= 10.00%)
<Set_Cluster_UnitCell> 6.614718 0.000000 0.000000
<Set_Cluster_UnitCell> 0.000000 6.041246 0.000000
<Set_Cluster_UnitCell> 0.000000 0.000000 6.614718
```

```
widened unit cell to fit energy cutoff (Ang.)
A = 6.744142 0.000000 0.000000 (48)
B = 0.000000 6.322633 0.000000 (45)
C = 0.000000 0.000000 6.744142 (48)
```

## 67 開発者のためのインターフェース

開発者のためのインターフェースが用意されています。SCF 計算から得られた情報（ハミルトニアン、行列要素、重なり行列、密度行列、位置演算子、運動量演算子）はファイル「*System.Name.scfout*」に保存されます。このファイルに保存されたデータを開発者のポストプロセスコードから参照することができます。OpenMX Ver. 3.9 からは「*System.Name.scfout*」のデータ形式が変更されました。これに関連し、「SCF2File.c」、'read\_scfout.c'、'read\_scfout.h'、'analysis\_example.c' が変更されています。そのため、OpenMX の旧版で生成された scfout ファイルは Ver. 3.9 では解析できないことに注意して下さい。

ファイル「*System.Name.scfout*」に保存されたデータは以下の手順で利用可能です。

### 1. 'HS.fileout' の生成

入力ファイルにキーワード「HS.fileout」を設定します。

```
HS.fileout          on          # on|off, default=off
```

通常の OpenMX 計算を実行すると、データがファイル「*System.Name.scfout*」に出力されます。ここで「*System.Name*」は入力ファイルで指定された系の名前「System.Name」です。

### 2. analysis\_example の作成

ディレクトリ「source」内で、次のコマンドを実行します。

```
% make analysis_example
```

コンパイルが正常終了すると、ディレクトリ「work」内に実行ファイル「analysis\_example」が生成されます。

### 3. ./analysis\_example *System.Name.scfout*

ディレクトリ「work」に移動して、次のようにしてプログラムを実行します。

```
% ./analysis_example System.Name.scfout  
もしくは  
% ./analysis_example System.Name.scfout > HS.out
```

ハミルトニアン、重なり行列、密度行列等の要素が「HS.out」に書き出されます。

### 4. analysis\_example の説明

ファイル「analysis\_example」には、これらのデータについての詳しい説明があります。その一部を以下に示します。

```
*****  
You can utilize a filename.scfout which is generated by the SCF  
calculation of OpenMX by the following procedure:
```

1. Define your main routine as follows:

```
int main(int argc, char *argv[])
```

2. Include a header file, "read\_scfout.h", in your main routine (if you want, also in other routines) as follows:

```
#include "read_scfout.h"
```

3. Call a function, read\_scfout(), in the main routine as follows:

```
read_scfout(argv);
```

```
*****
```

'polB', 'jx', 及び'kSpin' はポストプロセスコードとしての本機能を利用して開発されました。

## 68 自動フォース・テスター

OpenMXには多くの機能が実装されていますので、それらが正しく実装されているのが保証することが必要です。実装の信頼性を確認する効果的な方法は、解析的に求めた力と数値計算によって求めた力を比較することです。プログラムに何らかのバグがあれば、それらは互いに異なるでしょう。これを実行するために、次のようにして自動テスターが利用可能です。

### 逐次計算

```
% ./openmx -forcetest 0
```

### 並列計算

```
% ./openmx -forcetest 0 "mpirun -np 4 openmx"
```

ここで「0」は、力の整合性を調べる際のエネルギー項のフラグです。フラグの種類は以下に示す8種類があり、各数字は次の項目に対応します。

| flag         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------------|---|---|---|---|---|---|---|---|---|
| Kinetic      | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Non-local    | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Neutral atom | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| diff Hartree | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Ex-Corr      | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| E. Field     | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Hubbard U    | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

ここで、「1」は力の整合性のチェックに含めるという意味です。ディレクトリ「work/force\_example」内には、力の整合性のチェックに使われる36個の入力ファイルが保存されています。テスト計算が終了すると、ディレクトリ「work」内にファイル「forcetest.result」が生成されます。比較結果の一部を以下に転載します。

```
force_example/C2_GGA.dat
```

```
flag= 0
```

```
Numerical force= -(Utot(s+ds)-Utot(s-ds))/(2*ds)
```

```
ds= 0.0003000000
```

```
Forces (Hartree/Bohr) on atom 1
```

|                 | x               | y               | z               |
|-----------------|-----------------|-----------------|-----------------|
| Analytic force  | -1.676203071292 | -1.397113794193 | -1.117456296887 |
| Numerical force | -1.676101156844 | -1.397036485449 | -1.117288361652 |
| diff            | -0.000101914447 | -0.000077308744 | -0.000167935235 |

```
force_example/C2_LDA.dat
flag= 0
Numerical force= -(Utot(s+ds)-Utot(s-ds))/(2*ds)
.....
....
```

## 69 自動メモリーリーク・テスター

OpenMX の実行時、使用メモリは必要に応じて動的に割り当てられています。動的メモリー割り当て (dynamic memory allocation) では、メモリーリークが発生する場合があります、MD ステップの増大とともに使用メモリが浪費され、最終的には異常終了する可能性が生じます。

OpenMX を次のように実行することで、メモリーリークをチェックすることが可能です。

### 逐次計算

```
% ./openmx -mltest
```

### 並列計算

```
% ./openmx -mltest "mpirun -np 4 openmx"
```

ディレクトリ「work/ml\_example」に保存されている 14 個のテスト用入力ファイルを用いて、メモリーリークが監視されます。この際に、プログラムコード「openmx」中の同じモニターポイントで実際に使用された VSZ および RSS を監視します。計算終了後、ファイル「mltest.result」が生成します。監視された VSZ と RSS は MD ステップの関数として、ファイル「mltest.result」中に次のよう出力されま

```
1 ml_example/DIA8.dat
```

|          |    | CPU (%) | VSZ (kbyte) | RSS (kbyte) |
|----------|----|---------|-------------|-------------|
| MD_iter= | 1  | 68.300  | 397396      | 123076      |
| MD_iter= | 2  | 96.400  | 436264      | 131916      |
| MD_iter= | 3  | 99.000  | 436264      | 131916      |
| MD_iter= | 4  | 97.900  | 436264      | 131916      |
| MD_iter= | 5  | 98.800  | 436264      | 131916      |
| MD_iter= | 6  | 99.300  | 436264      | 131916      |
| MD_iter= | 7  | 98.800  | 436264      | 131916      |
| MD_iter= | 8  | 99.200  | 436264      | 131916      |
| MD_iter= | 9  | 99.500  | 436264      | 131916      |
| MD_iter= | 10 | 99.100  | 436264      | 131916      |
| MD_iter= | 11 | 99.400  | 436264      | 131916      |
| MD_iter= | 12 | 99.500  | 436264      | 131916      |
| MD_iter= | 13 | 99.300  | 436260      | 131916      |
| MD_iter= | 14 | 99.500  | 436264      | 131916      |
| MD_iter= | 15 | 99.300  | 436264      | 131916      |
| MD_iter= | 16 | 99.500  | 436260      | 131916      |
| MD_iter= | 17 | 99.600  | 436264      | 131916      |
| MD_iter= | 18 | 99.400  | 436264      | 131916      |
| MD_iter= | 19 | 99.600  | 436264      | 133848      |

|          |    |        |        |        |
|----------|----|--------|--------|--------|
| MD_iter= | 20 | 99.400 | 436264 | 133848 |
| MD_iter= | 21 | 99.500 | 436264 | 133848 |
| MD_iter= | 22 | 99.600 | 436264 | 133848 |
| MD_iter= | 23 | 99.500 | 436264 | 133848 |
| MD_iter= | 24 | 99.500 | 436264 | 133848 |
| MD_iter= | 25 | 99.700 | 436264 | 133848 |
| MD_iter= | 26 | 99.500 | 436264 | 133848 |
| MD_iter= | 27 | 99.600 | 436264 | 133848 |
| MD_iter= | 28 | 99.500 | 436264 | 133848 |
| MD_iter= | 29 | 99.600 | 436264 | 133848 |
| MD_iter= | 30 | 99.600 | 436264 | 133848 |

2 ml\_example/DIA8\_DC.dat

|          |   | CPU (%) | VSZ (kbyte) | RSS (kbyte) |
|----------|---|---------|-------------|-------------|
| MD_iter= | 1 | 101.000 | 412508      | 136448      |
| MD_iter= | 2 | 100.000 | 516940      | 210312      |
| MD_iter= | 3 | 98.200  | 517016      | 210440      |
| MD_iter= | 4 | 98.900  | 517016      | 210440      |
| MD_iter= | 5 | 99.300  | 517016      | 210440      |
| MD_iter= | 6 | 99.500  | 517016      | 210440      |
|          |   | .....   |             |             |
|          |   | ....    |             |             |

## 70 メモリ使用量の解析

「\*.memory0」、「\*.memory1」、.....「\*.memory#」のファイルを解析することによってメモリの使用状況が分かります。ここで、「\*」はキーワード「System.Name」で指定されるファイル名であり、ファイル拡張子の最後の数（#）は、MPI 並列化におけるプロセス ID です。これらのファイルは、キーワード「memory.usage.fileout」を次のように設定することで出力されます。

```
memory.usage.fileout  on  # default=off, on|off
```

例として、「met.memory0」を下記に示します。

```
Memory: SetPara_DFT: Spe_PAO_XV          0.01 MBytes
Memory: SetPara_DFT: Spe_PAO_RV          0.01 MBytes
Memory: SetPara_DFT: Spe_Atomic_Den      0.01 MBytes
Memory: SetPara_DFT: Spe_PAO_RWF        0.57 MBytes
Memory: SetPara_DFT: Spe_RF_Bessel      1.03 MBytes
Memory: SetPara_DFT: Spe_VPS_XV         0.01 MBytes
Memory: SetPara_DFT: Spe_VPS_RV         0.01 MBytes
Memory: SetPara_DFT: Spe_Vna            0.01 MBytes
Memory: SetPara_DFT: Spe_VH_Atom        0.01 MBytes
Memory: SetPara_DFT: Spe_Atomic_PCC     0.01 MBytes
Memory: SetPara_DFT: Spe_VNL            0.11 MBytes
Memory: SetPara_DFT: Spe_VNLE           0.00 MBytes
Memory: SetPara_DFT: Spe_VPS_List       0.00 MBytes
.....
....
...
Memory: Poisson: array0                  4.00 MBytes
Memory: Poisson: array1                  4.00 MBytes
Memory: Poisson: request_send            0.00 MBytes
Memory: Poisson: stat_send                0.00 MBytes
Memory: Poisson: request_recv            0.00 MBytes
Memory: Poisson: stat_recv                0.00 MBytes
Memory: Force: Hx                         0.00 MBytes
Memory: Force: Hy                         0.00 MBytes
Memory: Force: Hz                         0.00 MBytes
Memory: Force: CDM0                       0.00 MBytes
Memory: Data_Grid_Copy_B2C_1: Work_Array_Snd_Grid_B2C 0.72 MBytes
Memory: Data_Grid_Copy_B2C_1: Work_Array_Rcv_Grid_B2C 0.72 MBytes
Memory: total                             256.99 MBytes
```

上記のファイルは、入力ファイル「Methane.dat」でキーワード「memory.usage.fileout」を設定、シングルプロセスで実行し、得られたものです。大部分の配列のメモリ使用量は、このファイルに記録されますが、リストは完全ではないことに注意して下さい。

## 71 大規模ファイルのバイナリ形式での出力

大規模計算では、大規模テキスト形式ファイル (例えば、cube ファイル) が生成されます。このようなファイルを出力する際の IO アクセスは、IO アクセスが高速でない計算機上では、長時間を要します。そのような場合には、大規模ファイルをバイナリ形式で出力することをお勧めします。バイナリ形式での出力はキーワード「OutData.bin.flag」を指定することで実行可能です。

```
OutData.bin.flag    on    # default=off, on|off
```

これにより、大規模ファイルは全てバイナリ形式で出力されます。デフォルトは「off」です。

出力されたバイナリファイルは、ディレクトリ「source」内のプログラム「bin2txt.c」を使用して変換されます。このプログラムは次のようにして、コンパイルできます。

```
gcc bin2txt.c -lm -o bin2txt
```

後処理として、以下のように実行すればバイナリ形式からテキスト形式に変換されます。

```
./bin2txt *.bin
```

この機能は IO アクセスが高速でない計算機では有用でしょう。

## 72 Gaussian cube 形式から周期的 XSF 形式への変換

VESTA の現行バージョン (2015/8 版) では、Gaussian cube 形式のファイルを周期的に拡張して描画できません。周期的に拡張した構造を描画するためには周期的 XSF 形式への変換が必要です。

OpenMX は Gaussian cube 形式で電荷/スピン密度や Kohn-Sham 軌道、固有チャネルなどを出力可能であり、「cube2xsf」を用いることで、Gaussian cube 形式から周期的 XSF 形式への変換が可能です。

「cube2xsf」は以下の様に使用して下さい。

```
$ cube2xsf a.cube
$ cube2xsf b.cube.bin
$ cube2xsf a.cube b.cube.bin
$ cube2xsf *.cube.bin
```

これらのコマンドにより拡張子「.xsf」を持つファイルが生成されます。

## 73 入力ファイルの例

参考のために、本マニュアル中の計算例に関する入力ファイルがディレクトリ「work」内に保存されています。入力ファイルの一覧を以下に示します。

### Molecules or clusters

|                   |   |
|-------------------|---|
| C60.dat           | SCF calc. of a C60 molecule                         |
| C60_DC.dat        | DC calc. of a C60 molecule                          |
| CG15c_DC.dat      | DC calc. of DNA                                     |
| Cr2_CNC.dat       | Constrained DFT calc. of a Cr2 dimer                |
| Doped_NT.dat      | SCF calc. of doped carbon nanotube                  |
| Fe2.dat           | SCF calc. of a Fe2 dimer                            |
| Gly_NH.dat        | Nose-Hoover MD of a glycine molecule                |
| Gly_VS.dat        | Velocity scaling MD of a glycine molecule           |
| H2O.dat           | Geometry opt. of a water molecule                   |
| MCCN.dat          | DC calc. of a multiply connected carbon nanotube    |
| Methane2.dat      | Geometry opt. of a distorted methane molecule       |
| Methane.dat       | SCF calc. of a methane molecule                     |
| Methane_00.dat    | Orbital optimization of a methane molecule          |
| Methane_ED.dat    | Total energy decomposition of a methane molecule    |
| Mn12.dat          | SCF calc. of a single molecular magnet, Mn12        |
| Mol_MnO_NC.dat    | Non-collinear SCF calc. of a MnO molecule           |
| Nitro_Benzene.dat | SCF calc. of a nitro benzene molecule under E-field |
| Pt13.dat          | SCF calc. of a Pt13 cluster                         |
| Pt63.dat          | SCF calc. of a Pt63 cluster                         |
| SialicAcid.dat    | SCF calc. of a sialic acid molecule                 |
| Valorphin_DC.dat  | DC calc. of valorphin molecule                      |
| Valorphin_MO.dat  | Molecular orbital calculations                      |
| C2H4_NEB.dat      | NEB calc. of C2H4 dimer                             |
| C60_L0.dat        | Low-order scaling calc. of a C60 molecule           |
| Fe_Cluster_jx.dat | jx calculation of a Fe dimer                        |
| H2O+0.dat         | SCF calc. of a neutral water molecule               |
| H2O+1.dat         | SCF calc. of a plus 1 water molecule                |
| C2H2.dat          | Initial state calculation of C2H2 for XPS           |
| C2H2-CH.dat       | Final state calculation of C2H2 for XPS             |

### Bulk

|                |   |
|----------------|---|
| Cdia.dat       | SCF calc. of bulk diamond                             |
| MnO_NC.dat     | Non-collinear SCF calc. of bulk MnO                   |
| FeO_NC.dat     | Non-collinear SCF calc. of bulk FeO                   |
| CoO_NC.dat     | Non-collinear SCF calc. of bulk CoO                   |
| NiO_NC.dat     | Non-collinear SCF calc. of bulk NiO                   |
| Crys-NiO.dat   | SCF calc. of bulk NiO                                 |
| NiO-cFLL.dat   | LDA+U calc. of NiO by the cFLL scheme                 |
| NiO-sFLL.dat   | LDA+U calc. of NiO by the sdcFLL scheme               |
| NiO-Yukawa.dat | LDA+U calc. of NiO to estimate J and $F^4/F^2$        |
| DIA64_Band.dat | SCF calc. of bulk diamond including 64 atoms          |
| DIA8_DC.dat    | DC calc. of bulk diamond including 8 atoms            |
| DIA64_DC.dat   | DC calc. of bulk diamond including 64 atoms           |
| DIA216_DC.dat  | DC calc. of bulk diamond including 216 atoms          |
| DIA512_DC.dat  | DC calc. of bulk diamond including 512 atoms          |
| DIA512-1.dat   | Krylov 0(N) calc. of bulk diamond including 512 atoms |
| Febcc2.dat     | SCF calc. of bcc Fe                                   |

Fefcc-SpinSpiral.dat SCF calc. of fcc Fe for spin spiral calculation  
 GaAs.dat Non-collinear calc. of bulk gallium arsenide  
 NaCl.dat SCF calc. of bulk NaCl  
 NaCl\_FC.dat SCF calc. of bulk NaCl with a Cl-site vacancy  
 Si8.dat Geometry opt. of distorted Si bulk  
 Si8-pV.dat Enthalpy opt. of Si bulk under 10 GPa  
 Si8-LNO.dat O(N) DC-LNO calc. of silicon crystal including 8 atom  
 NdCo5\_4f.dat E vs. lattice constant calc. of NcCo4 bulk including the 4f states  
 NdCo5\_4f+U.dat E vs. lattice constant calc. of NcCo4 bulk including the 4f states with plus U  
 NdCo5\_OC.dat E vs. lattice constant calc. of NcCo4 bulk with a Nd open core pseudopotential  
 Al-Si111\_ESM.dat ESM calc. of Al-Si interface  
 Cafcc\_FS.dat Fermi surface calc. of the fcc Ca bulk  
 Graphite\_STM.dat STM image of graphene  
 Mnfcc-EvsLC.dat E vs. lattice constant calc. of the fcc Mn bulk  
 Si8\_NEB.dat NEB calc. for hydrogen in Si  
 DIA8-VA.dat Virtual atom SCF calc. of diamond crystal  
 FePt.dat SCF calc. of L10-FePt within collinear DFT  
 FePt-NC-SCF.dat SCF calc. of L10-FePt within non-collinear DFT  
 FePt-NC.dat One-shot diagonalization of L10-FePt with SOI  
 Fe\_Bulk\_jx.dat jx calculation of bcc Fe  
 GaCuS2\_mp-5238\_symmetrized\_SOC.dat SCF calculation for BoltzTrap calculation  
 Si\_BoltzTraP.dat SCF calculation of Si bulk for BoltzTrap calculation  
 Si2\_k50x50x50.dat Optical conductivity for Si bulk  
 TiC216.dat Initial state calculation of TiC for XPS  
 TiC216-CH3.dat Final state calculation of TiC for XPS  
 Si-4-SOI.dat Initial state calculation of Si for XPS  
 Si-4-CH-SOI1.dat Final state ( $j=3/2$ ) calculation of Si for XPS  
 Si-4-CH-SOI6.dat Final state ( $j=1/2$ ) calculation of Si for XPS  
 Au111Surface\_FL.dat spin texture analysis of Au111 by the FermiLoop scheme  
 Au111Surface\_GC.dat spin texture analysis of Au111 by the GridCalc scheme  
 Au111Surface\_BD.dat spin texture analysis of Au111 by the BandDispersion scheme  
 Au111Surface\_MO.dat spin texture analysis of Au111 by the MulPOnly scheme  
 SiC\_Primitive\_BD.dat spin texture analysis of SiC by the BandDispersion scheme

## 74 知られている問題点

- 基底関数の過完備性

fcc、hcp、bccなどの結晶構造をもつ高密度のバルク系において、多くの基底関数を割り当てた場合には、基底関数が過完備になる可能性があります。そのような場合には、数値不安定性のために誤った固有値が生じる可能性が高まります。この過完備性を避けるために、最適化された少数の基底関数を使用することが推奨されます。またこの問題を避ける別の方法として、キーワード「scf.ProExpn.VNA」を「off」に設定することも有効です。

```
scf.ProExpn.VNA      off      # on|off, default = on
```

この場合、実空間の数値グリッドに対するカットオフエネルギーをキーワード「scf.energycutoff」によって大き目に設定する必要があるかもしれません。

- SCFの収束が困難なこと

複雑な（ノンコリニア）磁氣的構造、金属的な電子構造、その両者を合わせ持つ場合、またそれらの特徴を持った大規模系では、SCFの収束を得ることがかなり困難です。そのような場合には、電子密度を非常にゆっくりと混合せねばならず、それは残念ながら、収束解を得るために多くのSCFステップが必要となることを意味します。

- 構造最適化が困難なこと

分子系のように、相互作用が弱い系では、完全に最適化された構造を得ることは容易でなく、多数の反復ステップが必要となります。構造最適化のデフォルトの収束条件は最大  $10^{-4}$  Hartree/Bohr ですが、そのような場合には、妥協案として収束条件を  $10^{-4}$  から  $5 \times 10^{-4}$  に設定仕直すことも1つの方法です。

## 75 OpenMX フォーラム

OpenMX と ADPACK の技術的問題に関する議論のために、フォーラムが用意されています (<http://www.openmx-square.org/forum/patio.cgi>)。このフォーラムは OpenMX 利用上のヒントをユーザー間で共有することと、プログラムコード発展のための情報交換の場となることを期待して開設されています。フォーラムの利用上の注意点に関しては、<http://www.openmx-square.org/forum/note.html> を参照して下さい。

## 76 OpenMX に関するその他の情報源

いくつかのウェブサイトが OpenMX に関連する情報を提供しています。以下にその一覧を示します。

- Lecture materials at Oregon State University.  
<http://physics.oregonstate.edu/~tatej/COURSES/ph575/doku.php?id=openmx>
- Lecture materials at Tokyo Institute of Technology  
[https://www.slideshare.net/cms\\_initiative/open-mx-lecture](https://www.slideshare.net/cms_initiative/open-mx-lecture)
- Lecture materials at Kanazawa University  
<http://f-ishii.w3.kanazawa-u.ac.jp/ja/index.cgi?page=%B7%D7%BB%BB%CA%AA%C0%AD%B2%CA%B3%D82018>
- Tutorial by HPCI  
[http://www.hpci-office.jp/pages/appli\\_openmx](http://www.hpci-office.jp/pages/appli_openmx)
- Tutorial by Dr. Toyoda  
<https://sites.google.com/site/mtoyodacmp/openmx-memo>
- Tutorial by Dr. Ito  
<http://www-fps.nifs.ac.jp/ito/memo/openmx04.html>
- Information by MateriApps  
<https://ma.issp.u-tokyo.ac.jp/app/594>
- Information by Dr. Inukai  
<https://www5.hp-ez.com/hp/calculations/page114>
- Tutorial material by Prof. Kato  
[https://www.slideshare.net/cms\\_initiative/materiapps-openmx](https://www.slideshare.net/cms_initiative/materiapps-openmx)
- Benchmark by Dr. Larsson  
<https://www.nsc.liu.se/~pla/blog/2014/06/11/openmx/>
- Benchmark by HPC Technologies  
<https://www.hpc-technologies.co.jp/openmx-benchmarks>

## 77 他ツールへのリンク

他ツールへのリンクを本章にまとめます。

- トライアルユース

OpenMX をすばやく試してみたい場合には、以下のツールをお使いいただけます。

- MateriApps LIVE!:

[http://ma.cms-initiative.jp/en/whats-materiapps/try\\_apps/about-materiapps-live?set\\_language=en](http://ma.cms-initiative.jp/en/whats-materiapps/try_apps/about-materiapps-live?set_language=en)

MateriApps LIVE!は、ノートパソコンなどで計算材料科学に関するシミュレーションを無償でお試しいただける環境です。MateriApps アプリケーション、OS (Debian GNU / Linux) エディタ、ビジュアライゼーションツールなど、チュートリアルを開始するために必要なすべての環境が、USB メモリスティックで提供されています。OpenMX は "MateriApps LIVE!" のシミュレーションツールの1つとして利用可能ですので、"MateriApps LIVE!" を OpenMX の試用環境として使用することができます。

- バイナリ配布

以下から、LINUX 環境向けの OpenMX バイナリを入手可能です。

- Debian: <https://packages.debian.org/search?keywords=openmx>

- Ubuntu: <https://launchpad.net/ubuntu/+source/openmx>

- グラフィカルユーザーインターフェース (GUI) とジョブ管理環境

OpenMX 計算のための複数の GUI とジョブスケジューリング環境が以下から利用できます。

- ASE: <https://wiki.fysik.dtu.dk/ase/ase/calculators/openmx.html>

- sisl: <http://zerothi.github.io/sisl/docs/latest/api-generated/sisl.io.html>

- OMXTool: <https://github.com/Ncmexp2717/OMXTool>

- Winmostar: [https://winmostar.com/jp/manual\\_jp/V9/html/winmos/solid/winmos\\_openmx.html](https://winmostar.com/jp/manual_jp/V9/html/winmos/solid/winmos_openmx.html)

- 一般的な可視化ツール

OpenMX では cube、md、xyz、xsf、axsf、cif 形式のファイルを出力されます。これらのファイルは下記に列挙する多くのソフトウェアで可視化できます。

- OpenMX Viewer: <http://www.openmx-square.org/viewer/>

- XCrySDen: <http://www.xcrysden.org/>

- VESTA: <http://jp-minerals.org/vesta/en/>

- Molekel: <http://www.cscs.ch/molekel/>

- フェルミ面の可視化

フェルミ面は FermiSurfer で可視化できます。

- FermiSurfer: <http://fermisurfer.osdn.jp/>

- 分子動力学計算結果の解析

以下のツールを用いて、OpenMX の分子動力学シミュレーションによって生成されたトラジェクトリーを解析できます。

– ASAP: <http://www.mch.rwth-aachen.de/>

- OpenMX の Kohn-Sham ハミルトニアン の 読込演算 ツール

OpenMX の Kohn-Sham ハミルトニアン を 読込み ・ 演算 する ための ツール が Artem Pulkin 博士 によって 公開 されて います 。 以下 の ウェブ サイト を ご 参照 下さい 。

– openmx-hks: <https://github.com/pulkin/openmx-hks>

- Tight Binding Studio

OpenMX コード を 含む 第一 原理 計算 と 連携 して タイト バインディング モデル を 構築 する ソフトウェア パッケージ 。

– Tight Binding Studio: <https://tight-binding.com/>

- 熱電特性

OpenMX の 計算 結果 を 利用 し、熱電特性 が 計算 する 外部 ツール が 以下 で 公開 されて います 。

– BoltzTraP: <https://doi.org/10.1016/j.cpc.2006.03.007>

– QTWARE: <http://www.rs.tus.ac.jp/takahiro/QTWare.html>

- Jx: 磁気相互作用を計算するオープンソースソフトウェア。OpenMX コードとのインターフェースを持つ。

OpenMX コード 本体 を 用い て も 同様 の 計算 が 実行 可能 ですが、Jx コード は KAIST の M.J. Han 教授 の グループ によって 開発 されて いる ポスト プロセス コード であり、独立 に 開発 ・ 公開 されて います 。

– Jx: <https://doi.org/10.1016/j.cpc.2019.106927>

- ワニエ関数を用いた物性解析

最局在ワニエ関数を用いることで、様々な物性を効率的かつ正確に計算することができます。OpenMX は、ポストプロセスコード「Wannier90」とのインターフェースを提供しています。

– Wannier90: <http://wannier.org/>

Wannier90 では、第一原理 SCF 計算の結果を用いて最局在ワニエ関数を構築し、ワニエ射影 DOS 及びバンド構造、フェルミ面、ベリー位相関連特性（異常ホール伝導度、光伝導率、軌道磁化）及び熱電特性の計算が可能です。詳細は「Wannier90 とのインターフェース」の章をご参照下さい。

- フォノン関連特性

フォノン分散や熱伝導率などのフォノン関連特性を計算するため OpenMX は以下のウェブサイト で 解説 される ALAMODE と 連携 できます 。

- ALAMODE: <https://alamode.readthedocs.io/en/latest/tutorial.html>
- DCore: DFT コードとのインターフェースを持つ DMFT ソルバー  
動的平均場理論 (DMFT) に基づく DCore と OpenMX のインターフェース。詳細は以下のウェブサイトをご参照下さい。
  - DCore: [https://issp-center-dev.github.io/DCore/master/tutorial/srvo3\\_openmx/openmx.html#](https://issp-center-dev.github.io/DCore/master/tutorial/srvo3_openmx/openmx.html#)

## 78 その他

### プログラム

本プログラムパッケージは C 言語と F90 言語で記述されたルーチン、1 つの makefile で構成されています。

makefile,

また 34 個のヘッダーファイルが含まれます。

```
BandDispersion.h Circular_Search.h EigenValue_Problem.h Eigen_HH.h GetOrbital.h
Inputtools.h Tools_BandCalc.h Tools_Search.h exx.h exx_debug.h exx_def_openmx.h
exx_file_eri.h exx_file_overlap.h exx_index.h exx_interface_openmx.h
exx_log.h exx_rhox.h exx_step1.h exx_step2.h exx_vector.h exx_xc.h f77func.h
jx.h jx_LNO.h jx_config.h jx_quicksort.h jx_tools.h jx_total_mem.h
lapack_prototypes.h mimic_sse.h openmx_common.h read_scfout.h tran_prototypes.h
tran_variables.h
```

全部で 350 個のプログラムルーティンから構成されており、その内訳を以下に記載します。

```
ADIIS_Mixing_DM.c ADenBand.c Allocate_Arrays.c AngularF.c BandDispersion.c Band_DFT_Col.c
Band_DFT_Col_NEGF.c Band_DFT_Col_Optical_ScaLAPACK.c Band_DFT_Dosout.c Band_DFT_MO.c
Band_DFT_NonCol.c Band_DFT_NonCol_GB.c Band_DFT_NonCol_Optical.c Band_DFT_kpath.c
Band_DFT_kpath_LNO.c Band_Dispersion.c Bench_MatMul.c BentNT.c BroadCast_ComplexMatrix.c
BroadCast_ReMatrix.c Calc_optical.c Circular_Search.c Cluster_DFT_Col.c Cluster_DFT_Dosout.c
Cluster_DFT_LNO.c Cluster_DFT_NonCol.c Cluster_DFT_ON2.c Cluster_DFT_OptOrb.c
Cluster_DFT_Optical.c Cluster_DFT_Optical_ScaLAPACK.c Cluster_DFT_ScaLAPACK.c Cont_Matrix0.c
Cont_Matrix1.c Cont_Matrix2.c Cont_Matrix3.c Cont_Matrix4.c Contract_Hamiltonian.c
Contract_iHNL.c Coulomb_Interaction.c Cutoff.c DFT.c DFTD3vdW_init.c DFTDvdW_init.c
DIIS_Mixing_DM.c DIIS_Mixing_Rhok.c Divide_Conquer.c Divide_Conquer_Dosout.c
Divide_Conquer_LNO.c DosMain.c Dr_KumoF.c Dr_RadialF.c Dr_VH_AtomF.c Dr_VNAF.c
EGAC_DFT.c Eff_Hub_Pot.c EigenBand_lapack.c EigenValue_Problem.c Eigen_HH.c Eigen_PHH.c
Eigen_PReHH.c Eigen_lapack.c Eigen_lapack2.c Eigen_lapack3.c Embedded_GFM.c EulerAngle_Spin.c
FT_NLP.c FT_PA0.c FT_ProExpn_VNA.c FT_ProductPA0.c FT_VNA.c FermiLoop.c File_CntCoes.c
Find_CGrids.c Force.c Force_test.c Free_Arrays.c Fuzzy_Weight.c GR_Pulay_DM.c Gaunt.c
Gauss_Legendre.c Generate_Wannier.c Generating_MP_Special_Kpt.c GetOrbital.c
Get_Cnt_Orbitals.c Get_Cnt_dOrbitals.c Get_OneD_HS_Col.c Get_Orbitals.c Get_dOrbitals.c
GridCalc.c Hamiltonian_Band.c Hamiltonian_Band_NC.c Hamiltonian_Band_NC_Hs2.c
Hamiltonian_Cluster.c Hamiltonian_Cluster_Hs.c Hamiltonian_Cluster_NC.c
Hamiltonian_Cluster_NC_Hs2.c Hamiltonian_Cluster_SO.c Hamiltonian_NC_Hs2.c
Init_List_YOUSO.c Initial_CntCoes.c Initial_CntCoes2.c Input_std.c Inputtools.c
Inputtools_kSpin.c Kerker_Mixing_Rhok.c Krylov.c KumoF.c LNO.c LU_inverse.c
Lapack_LU_inverse.c MD_pac.c MTRAN_EigenChannel.c Make_Comm_Worlds.c Make_FracCoord.c
Make_InputFile_with_FinalCoord.c Maketest.c Matrix_Band_LNO.c Memory_Leak_test.c
Merge_LogFile.c Mio_tester.c Mio_tester2.c Mixing_DM.c Mixing_H.c Mixing_V.c
MulPCalc.c MulPOnly.c Mulliken_Charge.c NBO_Cluster.c NBO_Krylov.c NabraMatrixElements.c
Nonlocal_Basis.c Nonlocal_RadialF.c Occupation_Number_LDA_U.c Opt_Contraction.c
OpticalConductivityMain.c Orbital_Moment.c OutData.c OutData_Binary.c Output_CompTime.c
Output_Energy_Decomposition.c Overlap_Band.c Overlap_Band_NC_Ss2.c Overlap_Cluster.c
Overlap_Cluster_LNO.c Overlap_Cluster_NC_Ss2.c Overlap_Cluster_Ss.c PhiF.c
```

Poisson.c Poisson\_ESM.c Population\_Analysis\_Wannier.c Population\_Analysis\_Wannier2.c  
 Pot\_NeutralAtom.c PrintMemory.c PrintMemory\_Fix.c QuickSort.c RF\_BesselF.c  
 RadialF.c ReLU\_inverse.c RestartFileDFT.c Runtest.c SCF2File.c SetPara\_DFT.c  
 Set\_Aden\_Grid.c Set\_Allocate\_Atom2CPU.c Set\_ContMat\_Cluster\_LNO.c Set\_CoreHoleMatrix.c  
 Set\_Density\_Grid.c Set\_Hamiltonian.c Set\_Initial\_DM.c Set\_Nonlocal.c Set\_OLP\_Kin.c  
 Set\_OLP\_p.c Set\_Orbitals\_Grid.c Set\_ProExpn\_VNA.c Set\_Vpot.c Set\_XC\_Grid.c  
 Set\_dOrbitals\_Grid.c Show\_DFT\_DATA.c SigmaEK.c Simple\_Mixing\_DM.c Smoothing\_Func.c  
 Spherical\_Bessel.c Stress.c Stress\_test.c TRAN\_Add\_ADensity\_Lead.c TRAN\_Add\_Density\_Lead.c  
 TRAN\_Allocate.c TRAN\_Allocate\_NC.c TRAN\_Apply\_Bias2e.c TRAN\_Band.c TRAN\_Band\_Col.c  
 TRAN\_CDen\_Main.c TRAN\_Calc\_CentGreen.c TRAN\_Calc\_CentGreenLesser.c TRAN\_Calc\_CurrentDensity.c  
 TRAN\_Calc\_GridBound.c TRAN\_Calc\_Hopping\_G.c TRAN\_Calc\_OneTransmission.c TRAN\_Calc\_SelfEnergy.c  
 TRAN\_Calc\_SurfGreen.c TRAN\_Calc\_SurfGreen\_Sanvito.c TRAN\_Channel\_Functions.c TRAN\_Channel\_Output.c  
 TRAN\_Check\_Input.c TRAN\_Check\_Region.c TRAN\_Check\_Region\_Lead.c TRAN\_Credit.c TRAN\_DFT.c  
 TRAN\_DFT\_Dosout.c TRAN\_DFT\_NC.c TRAN\_Deallocate\_Electrode\_Grid.c TRAN\_Deallocate\_RestartFile.c  
 TRAN\_Distribute\_Node.c TRAN\_Input\_std.c TRAN\_Input\_std\_Atoms.c TRAN\_Input\_std\_Atoms0.c  
 TRAN\_Input\_std\_Atoms2.c TRAN\_Main\_Analysis.c TRAN\_Main\_Analysis\_NC.c TRAN\_Output\_HKS.c  
 TRAN\_Output\_HKS\_Write\_Grid.c TRAN\_Output\_Trans\_HS.c TRAN\_Poisson.c TRAN\_Print.c TRAN\_Print\_Grid.c  
 TRAN\_Read.c TRAN\_RestartFile.c TRAN\_Set\_CentOverlap.c TRAN\_Set\_CentOverlap\_NC.c  
 TRAN\_Set\_Electrode\_Grid.c TRAN\_Set\_IntegPath.c TRAN\_Set\_MP.c TRAN\_Set\_SurfOverlap.c  
 TRAN\_Set\_SurfOverlap\_NC.c TRAN\_Set\_Value.c TRAN\_adjust\_Ngrid.c Tetrahedron\_Bloch1.c Timetool.c  
 Tools\_BandCalc.c Tools\_Search.c Total\_Energy.c Unfolding\_Bands.c VH\_AtomF.c VNAF.c  
 Voronoi\_Charge.c Voronoi\_Orbital\_Moment.c XANES0.c XC\_CA\_LSDA.c XC\_Ceperly\_Alder.c XC\_EX.c  
 XC\_PBE.c XC\_PW92C.c Z2FH.c add\_gcube.c analysis\_example.c bandgnu13.c bin2txt.c calB.c  
 check\_lead.c cube2xsf.c damping.c deri\_dampingF.c diff\_gcube.c diff\_geo.c dtime.c esp.c  
 example\_mpi\_spawn.c expao.c exx.c exx\_debug.c exx\_file\_eri.c exx\_file\_overlap.c exx\_index.c  
 exx\_interface\_openmx.c exx\_log.c exx\_rhox.c exx\_step1.c exx\_step2.c exx\_vector.c exx\_xc.c  
 find\_Emin.c find\_Emin0.c find\_Emin2.c find\_Emin\_withS.c frac2xyz.c gcube2oned.c gen\_defile.c  
 init.c init\_alloc\_first.c intensity\_map.c io\_tester.c iterout.c iterout\_md.c jx.c jx\_LNO.c  
 jx\_band\_indiv.c jx\_band\_psum.c jx\_cluster.c jx\_config.c jx\_quicksort.c jx\_tools.c kSpin.c  
 lapack\_dstedc1.c lapack\_dstedc2.c lapack\_dstedc3.c lapack\_dstegr1.c lapack\_dstegr2.c  
 lapack\_dstegr3.c lapack\_dsteqr1.c lapack\_dstevx1.c lapack\_dstevx2.c lapack\_dstevx3.c  
 lapack\_dstevx4.c lapack\_dstevx5.c malloc\_multidimarray.c md2axsf.c mimic\_sse.c mpao.c  
 mpi\_multi\_world.c mpi\_multi\_world2.c mpi\_non\_blocking.c neb.c neb\_check.c neb\_run.c  
 openmx.c openmx\_common.c outputfile1.c pdb2pao.c polB.c read\_scfout.c readfile.c rmmmpi.c  
 rot.c test\_mpi.c test\_mpi2.c test\_mpi3.c test\_mpi4.c test\_openmp.c test\_openmp2.c  
 test\_openmp3.c tp.c truncation.c unit2xyz.c xyz2spherical.c zero\_cfrac.c zero\_fermi.c  
 elpa1.f90 get\_elpa\_row\_col\_comms.f90 solve\_evp\_complex.f90 solve\_evp\_real.f90

さらに、次のライブラリーパッケージがリンクされています。

lapack,  
 blas,  
 fftw,  
 MPICH2 or OpenMP

## プログラムパッケージの著作権

このプログラムパッケージの配布は、GNU General Public License version 3 (GPLv3)[102] の規約に従っています。またオリジナルプログラムの作者である尾崎泰助がこのプログラムパッケージのオリジナルバージョンの著作権を所有しています。我々はこのプログラムパッケージのユーザーの利用に関して、

いかなる保証もできません。しかしプログラムのバグを報告して頂ければ、問題解決に向けて可能な限り、利用者と共に協力して作業を行う所存です。

## 謝辞

本プログラムの作成にあたり、有益な示唆とコメントをいただいた JRCAT、RICS-AIST、JAIST、そして ISSP の皆様に感謝いたします (T.O.)。また、次の国家プロジェクトによって支援を受けました。

- NEDO のナノ機能合成技術プロジェクト (SYNAF-NEDO) [154]
- 科学技術振興機構の計算科学技術活用型特定研究開発推進事業 (ACT-JST) [155]
- 文部科学省の最先端・高性能汎用スーパーコンピュータの開発利用プロジェクト (NAREGI) [156]
- 科学技術振興機構の戦略的創造研究推進事業 (CREST-JST) [157]
- 文部科学省科学研究費補助金「新学術領域研究」コンピューティクスによる物質デザイン [158]
- 文部科学省「HPCI 戦略プログラム (SPIRE)」分野 2;新物質・エネルギー創成, 計算物質科学イニシアティブ (CMSI) [159]
- NEDO の超先端材料超高速開発基盤技術プロジェクト [160]
- 文部科学省「ポスト京重点課題 7・次世代の産業を支える新機能デバイス・高性能材料の創成」プロジェクト [161]

ご協力を深く感謝申し上げます。

## References

- [1] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964); W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
- [2] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett., 45, 566(1980); J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
- [3] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).
- [4] J. P. Perdew and Y. Wang, Phys.Rev.B **45**, 13244 (1992).
- [5] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
- [6] A.E. Reed, L.A. Curtiss, and F. Weinhold, Chem. Rev. **88**, 899 (1988); E. D. Glendening, C. R. Landis, and F. Weinhold, WIREs Comput. Mol. Sci. **2**, 1 (2012).
- [7] T. Ohwaki, M. Otani, and T. Ozaki, J. Chem. Phys. **140**, 244105 (2014).
- [8] U. Von. Barth and L. Hedin, J. Phys. C: Solid State Phys. **5**, 1629 (1972).
- [9] J. Kübler, K-H. Höck, J. Sticht, and A. R. Williams, J. Phys. F: Met. Phys. **18**, 469 (1988).

- [10] J. Sticht, K-H. Höck, and J. Kübler, *J. Phys.: Condens. Matter* **1**, 8155 (1989).
- [11] T. Oda, A. Pasquarello, and R. Car, *Phys. Rev. Lett.* **80**, 3622 (1998).
- [12] A. H. MacDonald and S. H. Vosko, *J. Phys. C: Solid State Phys.* **12**, 2977 (1979).
- [13] Ph. Kurz, F. Forster, L. Nordstrom, G. Bihlmayer, and S. Blugel, *Phys. Rev. B* **69**, 024415 (2004).
- [14] J. Harris, *Phys. Rev. B* **31**, 1770 (1985).
- [15] R. D. King-Smith and D. Vanderbilt, *Phys. Rev. B* **47**, 1651 (1993).
- [16] G. Theurich and N. A. Hill, *Phys. Rev. B* **64**, 073106 (2001).
- [17] A. I. Liechtenstein, M. I. Katsnelson, V. P. Antropov, and V. A. Gubanov, *J. Mag. Mag. Mat.* **67**, 65 (1987).
- [18] M. J. Han, T. Ozaki, and J. Yu, *Phys. Rev. B* **70**, 184421 (2004).
- [19] A. Terasawa, M. Matsumoto, T. Ozaki, and Y. Gohda, *J. Phys. Soc. Jpn.* **88**, 114706 (2019).
- [20] M. J. Han, T. Ozaki, and J. Yu, *Phys. Rev. B* **74**, 045110 (2006).
- [21] S. Ryee and M.J. Han, *J. Phys.:Condens. Matter* **30**, 275802 (2018).
- [22] S. Ryee and M.J. Han, *Scientific Reports* **8**, 9559 (2018).
- [23] [http://www.openmx-square.org/tech\\_notes/tech6-1\\_0.pdf](http://www.openmx-square.org/tech_notes/tech6-1_0.pdf)
- [24] [http://www.openmx-square.org/tech\\_notes/DFTU\\_notes\\_OpenMX.pdf](http://www.openmx-square.org/tech_notes/DFTU_notes_OpenMX.pdf)
- [25] S.L. Dudarev, G.A. Botton, S.Y. Savrasov, C.J. Humphreys, and A.P. Sutton, *Phys. Rev. B* **57**, 1505 (1998).
- [26] A.I. Liechtenstein, V.I. Anisimov, and J. Zaanen, *Phys. Rev. B* **52**, R5467 (1995).
- [27] A. N. Yaresko, V. N. Antonov, and P. Fulde, *Phys. Rev. B* **67**, 155103 (2003).
- [28] L. Vaugier, H. Jiang, and S. Biermann, *Phys. Rev. B* **86**, 165105 (2012).
- [29] F. Bultmark, F. Cricchio, O. Grånäs, and L. Nordström, *Phys. Rev. B* **80**, 035121 (2009).
- [30] L. V. Woodcock, *Chem. Phys. Lett.* **10**, 257 (1971).
- [31] S. Nose, *J. Chem. Phys.* **81**, 511 (1984); S. Nose, *Mol. Phys.* **52**, 255 (1984); G. H. Hoover, *Phys. Rev. A* **31**, 1695 (1985).
- [32] G. B. Bachelet, D. R. Hamann, and M. Schluter, *Phys. Rev. B* **26**, 4199 (1982).
- [33] N. Troullier and J. L. Martine, *Phys. Rev. B* **43**, 1993 (1991).
- [34] L. Kleinman and D. M. Bylander, *Phys. Rev. Lett.* **48**, 1425 (1982).
- [35] P. E. Blochl, *Phys. Rev. B* **41**, 5414 (1990).

- [36] I. Morrison, D.M. Bylander, L. Kleinman, Phys. Rev. B **47**, 6728 (1993).
- [37] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).
- [38] H.J. Monkhorst and J.D. Pack, Phys. Rev. B **13**, 5188 (1976).
- [39] T. Auckenthaler, V. Blum, H.-J. Bungartz, T. Huckle, R. Johanni, L. Kraemer, B. Lang, and H. Lederer, P. R. Willems, Parallel Computing 27, 783 (2011).
- [40] K. Lejaeghere, V. Van Speybroeck, G. Van Oost, and S. Cottenier, Critical Reviews in Solid State and Materials Sciences 39, 1 (2014); K. Lejaeghere et al., Science 351, aad3000 (2016).
- [41] T. Ozaki, Phys. Rev. B. **67**, 155108, (2003); T. Ozaki and H. Kino, Phys. Rev. B **69**, 195113 (2004).
- [42] T. Ozaki and H. Kino, Phys. Rev. B **72**, 045121 (2005).
- [43] T. Ozaki, Phys. Rev. B **74**, 245101 (2006).
- [44] T.V.T. Duy and T. Ozaki, Comput. Phys. Commun. 185, 777 (2014).
- [45] T.V.T. Duy and T. Ozaki, Comput. Phys. Commun. 185, 153 (2014).
- [46] S.F. Boys and F. Bernardi, Mol. Phys. **19**, 553 (1970).
- [47] S. Simon, M. Duran, and J.J. Dannenberg, J. Chem. Phys. **105**, 11024 (1996).
- [48] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias and J. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992) and references therein.
- [49] O. F. Sankey and D. J. Niklewski, Phys. Rev. B. **40**, 3979 (1989)
- [50] W. Yang, Phys.Rev.Lett. **66**, 1438 (1991)
- [51] T. Ozaki, M. Fukuda, G. Jiang, Phys. Rev. B **98**, 245137 (2018).
- [52] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
- [53] V. Recoules and J.-P. Crocombette, Phys. Rev. B **72**, 104202 (2005).
- [54] J. A. Anta and P. A. Madden, J. Phys.: Condens. Matter **11**, 6099 (1999).
- [55] M. Kim, K. H. Khoo, and J. R. Chelikowsky, Phys. Rev. B **86**, 054104 (2012).
- [56] P. Ordejon, E. Artacho, and J. M. Soler, Phys. Rev. B. **53**, 10441 (1996)
- [57] D. R. Bowler and M. J. Gillan, Chem. Phys. Lett. **325**, 475 (2000).
- [58] G. Kresse and J. Furthmeuller, Phys. Rev. B. **54**, 11169 (1996)
- [59] G. P. Kerker, Phys. Rev. B **23**, 3082 (1981).
- [60] T. A. Arias, M. C. Payne, and J. D. Joannopoulos, Phys. Rev. B **45**, 1538 (1992).

- [61] D. Alfe, *Comp. Phys. Commun.* **118**, 32 (1999).
- [62] P. Csaszar and P. Pulay, *J. Mol. Struct. (Theochem)* **114**, 31 (1984).
- [63] J. Baker, *J. Comput. Chem.* **7**, 385 (1986)
- [64] A. Banerjee, N. Adams, J. Simons, R. Shepard, *J. Phys. Chem.* **89**, 52 (1985)
- [65] C. G. Broyden, *J. Inst. Math. Appl.* **6**, 76 (1970); R. Fletcher, *Comput. J.* **13**, 317 (1970); D. Goldfarb, *Math. Comp.* **24**, 23 (1970); D. F. Shanno, *Math. Comp.* **24**, 647 (1970).
- [66] H.B. Schlegel, *Theoret. Chim. Acta (Berl.)* **66**, 333 (1984); J.M. Wittbrodt and H.B. Schlegel, *J. Mol. Struct. (Theochem)* **398-399**, 55 (1997).
- [67] P. E. Blochl, O. Jepsen and O. K. Andersen, *Phys. Rev. B* **49**, 16223 (1994).
- [68] The details of the implementation will be published elsewhere.
- [69] A. D. Becke and R. M. Dickson, *J. Chem. Phys.* **89**, 2993 (1988).
- [70] A. Svane and O. Gunnarsson, *Phys. Rev. Lett.* **65**, 1148 (1990).
- [71] J. Tersoff and D. R. Hamann, *Phys. Rev. B* **31**, 805 (1985).
- [72] G. Henkelman and H. Jonsson, *J. Chem. Phys.* **113**, 9978 (2000).
- [73] T. Ozaki, K. Nishio, and H. Kino, *Phys. Rev.* **81**, 035116 (2010).
- [74] T. Ozaki, *Phys. Rev. B* **75**, 035123 (2007).
- [75] M. Brandbyge, J.-L. Mozos, P. Ordejon, J. Taylor, and K. Stokbro, *Phys. Rev. B* **65**, 165401 (2002)
- [76] G. C. Liang, A. W. Ghosh, M. Paulsson, and S. Datta, *Phys. Rev. B.* **69**, 115302 (2004).
- [77] H. Weng, T. Ozaki, and K. Terakura, *Phys. Rev. B* **79**, 235118 (2009).
- [78] H. Kotaka, F. Ishii, and M. Saito, *Jpn. J. Appl. Phys.* **52**, 035204 (2013).; N. Yamaguchi and F. Ishii, *Appl. Phys. Express* **10** 123003 (2017).
- [79] T.B. Prayitno and F. Ishii, *J. Phys. Soc. Jpn.* **87**, 114709 (2018).
- [80] T.B. Prayitno and F. Ishii, *J. Phys. Soc. Jpn.* **88**, 054701 (2019).
- [81] T. Fukui, Y. Hatsugai and H. Suzuki *J. Phys. Soc. Jpn. J. Phys. Soc. Jpn.* **74**, 1674 (2005).
- [82] W. Feng, J. Wen, J. Zhou, D. Xiao, and Y. Yao, *Comput. Phys. Commun.* **183**, 1849 (2012).
- [83] T. Kato, H. Kotaka, and F. Ishii, *JPS Conf. Proc.* **5**, 011022 (2015).
- [84] H. Sawahata, N. Yamaguchi, H. Kotaka, and F. Ishii, *Jpn. J. Appl. Phys.* **57**, 030309 (2018).
- [85] <https://t-ozaki.issp.u-tokyo.ac.jp/meeting16/OMX-Sawahata-2016Nov.pdf>

- [86] L.M. Sandratskii, *Adv. Phys.* **47**, 91 (1998).
- [87] V.M. García-Suárez, C.M. Newman, C.J. Lambert, J.M. Pruneda, and J. Ferrer, *J. Phys.: Condens. Matter* **16**, 5453 (2004).
- [88] T. Ozaki and C.C. Lee, *Phys. Rev. Lett.* **118**, 026401 (2017).
- [89] J.F. Janak, *Phys. Rev. B* **18**, 7165 (1978).
- [90] W.L. Jolly, K.D. Bomben, C.J. Eyermann, *At. Data Nul. Data Tables* **31**, 433 (1984).
- [91] M. R. Jarvis, I. D. White, R. W. Godby, and M. C. Payne, *Phys. Rev. B* **56**, 14972 (1997).
- [92] C. D. Wagner, W. M. Riggs, L. E. Davis, J. F. Moulder, and G. E. Mullenberg, *Handbook of X-Ray Photoelectron Spectroscopy* (Perkin-Elmer, Minnesota, 1979).
- [93] C.-C. Lee, J. Yoshinobu, K. Mukai, S. Yoshimoto, H. Ueda, R. Friedlein, A. Fleurence, Y. Yamada-Takamura, and T. Ozaki, *Phys. Rev. B* **95**, 115437 (2017).
- [94] C.-C Lee, B. Feng, M. D'angelo, R. Yukawa, R-Y Liu, T. Kondo, H. Kumigashira, I. Matsuda, and T. Ozaki, *Phys. Rev. B* **97**, 075430 (2018).
- [95] K. Yamazaki, Y. Maehara, C-C Lee, J. Yoshinobu, T. Ozaki, and K. Gohara, *J. Phys. Chem. C* **122**, 27292 (2018).
- [96] K.B.Snow and T.F. Thomas, *Int. J. Mass Spectrom. Ion Processes* **96**, 49 (1990).
- [97] M. Ernzerhof and G.E. Scuseria, *J. Chem. Phys.* **110**, 5029 (1999).
- [98] [http://www.openmx-square.org/tech\\_notes/Dielectric\\_Function\\_YTL.pdf](http://www.openmx-square.org/tech_notes/Dielectric_Function_YTL.pdf)
- [99] G.K.H. Madsen and D.J. Singh, *Comput. Phys. Commun.* **175**, **67** (2006).
- [100] M. Miyata, T. Ozaki, T. Takeuchi, S. Nishino, M. Inukai, and M. Koyano, *J. Elec. Mater.* **47**, 3254 (2017).
- [101] [https://www.imc.tuwien.ac.at//forschungsbereich\\_theoretische\\_chemie/forschungsgruppen/prof\\_dr\\_gkh\\_madsen\\_theoretical\\_materials\\_chemistry/boltztrap/](https://www.imc.tuwien.ac.at//forschungsbereich_theoretische_chemie/forschungsgruppen/prof_dr_gkh_madsen_theoretical_materials_chemistry/boltztrap/)
- [102] <http://www.gnu.org/>
- [103] <http://jp-minerals.org/vesta/en/>
- [104] <http://www.cscs.ch/molekel/>
- [105] <http://www.xcrysden.org/>
- [106] T. Lis, *Acta Crystallogra. B* **36**, 2042 (1980).
- [107] T. P. Davis T. J. Gillespie, F. Porreca, *Peptides* **10**, 747 (1989).
- [108] A. Goldstein, S. Tachibana, L. I. Lowney, M. Hunkapiller, and L. Hood, *Proc. Natl. Acad. Sci. U. S. A.* **76**, 6666 (1979).

- [109] U. C. Singh and P. A. Kollman, *J. Comp. Chem.* **5**, 129(1984).
- [110] L. E. Chirlian and M. M. Francl, *J. Com. Chem.* **8**, 894(1987).
- [111] B. H. Besler, K. M. Merz Jr. and P. A. Kollman, *J. Comp. Chem.* **11**, 431(1990).
- [112] <http://www.webelements.com/>
- [113] M. Cardona, N. E. Christensen, and G. Gasol, *Phys. Rev. B* **38**, 1806 (1988).
- [114] G. Theurich and N. A. Hill, *Phys. Rev. B* **64**, 073106 (2001).
- [115] *Physics of Group IV Elements and III-V Compounds*, edited by O.Madelung, M.Schulz, and H. Weiss, Landolt-Büornstein, New Series, Group 3, Vol. 17, Pt.a (Springer, Berlin, 1982).
- [116] T. Ono and K. Hirose, *Phys. Rev. B* **72**, 085105 (2005).
- [117] W. N. Mei, L. L. Boyer, M. J. Mehl, M. M. Ossowski, and H. T. Stokes, *Phys. Rev. B* **61**, 11425 (2000).
- [118] I. V. Solovyev, A. I. Liechtenstein, K. Terakura, *Phys. Rev. Lett.* **80**, 5758.
- [119] K. Knopfle, L. M. Sandratskii, and J. Kubler, *J. Phys:Condens. Matter* **9**, 7095 (1997).
- [120] I. S. Dhillon and B. N. Parlett, *SIAM J. Matrix Anal. Appl.* **25**, 858 (2004).
- [121] J. J. M. Cuppen, *Numer. Math.* **36**, 177 (1981); M. Gu and S. C. Eisenstat, *SIAM J. Mat. Anal. Appl.* **16**, 172 (1995).
- [122] N. Mazari and D. Vanderbilt, *Phys. Rev. B* **56**, 12 847 (1997).
- [123] I. Souza, N. Marzari and D. Vanderbilt, *Phys. Rev. B* **65**, 035109 (2001).
- [124] T. Ozaki, *Phys. Rev. B* **82**, 075131 (2010).
- [125] M. Otani and O. Sugino, *Phys. Rev. B* **73**, 115407 (2006).
- [126] O. Sugino, I. Hamada, M. Otani, Y. Morikawa, T. Ikeshoji, and Y. Okamoto, *Surf. Sci.* **601**, 5237 (2007).
- [127] M. Otani, I. Hamada, O. Sugino, Y. Morioka, Y. Okamoto, and T. Ikeshoji, *J. Phys. Soc. Jpn.* **77**, 024802 (2008).
- [128] T. Ohwaki, M. Otani, T. Ikeshoji, and T. Ozaki, *J. Chem. Phys.* **136**, 134101 (2012).
- [129] R.M. Eastment and C.H.B. Mee, *J. Phys. F: Metal Phys.* **3**, 1738 (1973).
- [130] P.O. Gartland, S. Berge, and B.J. Slagsvold, *Phys. Rev. Lett.* **28**, 738 (1972).
- [131] M. Chelvayohan and C.H.B. Mee, *J. Phys. C: Solid State Phys.* **15**, 2305 (1982).
- [132] G.V. Hansson and S.A. Flodström, *Phys. Rev. B* **18**, 1572 (1978).

- [133] G.D. Kubiak, J. Vac. Sci. Technol. A **5**, 731 (1987).
- [134] G. Henkelman and H. Jonsson, J. Chem. Phys. **113**, 9978 (2000).
- [135] S. Grimme, J. Comput. Chem. **27**, 1787 (2006).
- [136] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, J. Chem. Phys. **132**, 154104 (2010).
- [137] S. Grimme, S. Ehrlich and L. Goerigk, J. Comput. Chem. **32**, 1456 (2011).
- [138] J.G. Hill, J.A. Platts, and H.-J. Werner, Phys. Chem. Chem. Phys. **8**, 4072 (2006).
- [139] C. Li, L. Wan, Y. Wei, and J. Wang, Nanotechnology **19**, 155401 (2008).
- [140] L. Zhang, B. Wang, and J. Wang, Phys. Rev. B **84**, 115412 (2011).
- [141] M. Paulsson and M. Brandbyge, Phys. Rev. B **76**, 115117 (2007).
- [142] C.-C. Lee, Y. Yamada-Takamura, and T. Ozaki, J. Phys.: Condens. Matter **25**, 345501 (2013).
- [143] M. Kawamura, Comp. Phys. Comm. **239**, 197 (2019).
- [144] <http://fermisurfer.osdn.jp/>
- [145] <http://www.wannier.org/>
- [146] <https://github.com/Ncmexp2717/OMXTool>
- [147] H. Makino, I. H. Inoue, M. J. Rozenberg, I. Hase, Y. Aiura, and S. Onari, Phys. Rev. B **58**, 4384 (1998).
- [148] [http://www.fhi-berlin.mpg.de/th/fhi98md/Murn/readme\\_murn.html](http://www.fhi-berlin.mpg.de/th/fhi98md/Murn/readme_murn.html)
- [149] <http://www.openmx-square.org/>
- [150] <http://www.netlib.org/lapack/>
- [151] <http://www.openmx-square.org/viewer/>
- [152] Y.-T Lee and T. Ozaki, Journal of Molecular Graphics and Modelling **89**, 192 (2019).
- [153] <http://www.nanotec.es/>
- [154] <http://www.nanoworld.jp/synaf/>
- [155] <http://act.jst.go.jp/>
- [156] <http://ccinfo.ims.ac.jp/nanogrid/>
- [157] <http://www.jst.go.jp/>
- [158] <http://computics-material.jp/index-e.html>
- [159] <http://www.cms-initiative.jp/ja>

- [160] A project commissioned by the New Energy and Industrial Technology Development Organization of Japan (NEDO) Grant (P16010).
- [161] <https://cdmsi.issp.u-tokyo.ac.jp/>