|  |  |
| --- | --- |
| 曲がりくねった道と木々の写真  高校生向け原子シミュレーション入門書（仮）  授業が手につかなくなるくらい面白い世界へ | 要約  *「触媒反応シミュレーション、例えば、フィッシャー・トロプシュ反応触媒に対する好適な添加元素の探索を、あなた自身が、おうちのコンピュータでも出来るようになったとしたら、どう思うでしょうか？ 」ワクワクしたあなたは、ぜひこのテキストを読んでみて下さい。*  Masaki Open Lab.  [コースのタイトル] |

1. 本テキストのモチベーションと目標

「人口増加に伴う食糧危機を救うには、空中窒素の固定しかない。これこそ、化学者の最大かつ緊急の課題である」これは、19世紀末にイギリス化学会の会長であったクルックスが述べた言葉です[[1]](#footnote-1)。当時、農業に必要不可欠であった窒素肥料は主に天然のチリ硝石から作られていました。しかし、産業革命に伴う爆発的な人口の増加は、これら天然の窒素源をすさまじい速度で枯渇させていきました。そしてこの問題を解決するには、空気中の窒素を植物が利用可能な化合物に変換する必要がありました。

この言を受けて実際に開発されたのが、ハーバー・ボッシュ法（開発者2名の名前をとっています）でした。この反応は、空気中の窒素を水素と反応させてアンモニア（窒素肥料の原料）を合成するものであり、まさに空気からパンを生み出す魔法のようなプロセスでした（この手法が無ければ、現生人類の多くは生きることが出来なかっただろうとすら言われています）。

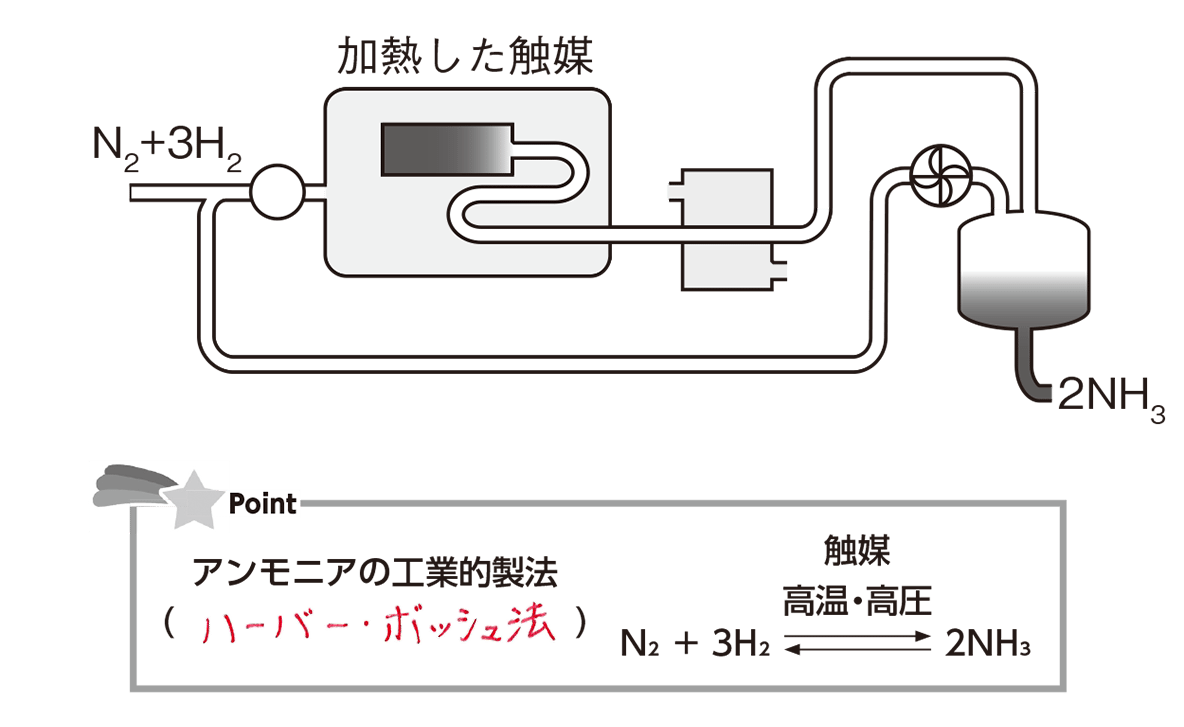


図 1　ハーバー・ボッシュ法[[2]](#footnote-2)

翻って現代、私を含め皆さんが生きる世界では、地球温暖化による様々な影響をまざまざと見るようになりました。「気候変動に伴う人類の危機を乗り越えるには、CO2をはじめとする温室効果ガスの排出量をゼロに減らしていくことしかない。」これこそ、化学者の最大かつ喫緊の課題である―と誰かが言ったかは定かではありませんが、CO2を排出しない各種化学品の製造法を開発すること、またそのための技術開発に寄与することは、多くの化学者にとって21世紀における一大ミッションとなっているように思います。

ところで、先のボッシュ氏が実用化した別の製造プロセスとして、フィッシャー・トロプシュ反応があります。この反応は、石炭を加熱して得られる一酸化炭素と水素から炭化水素、すなわちガソリンや軽油といった化石燃料を合成するものであり、原油資源に恵まれなかった（けれども石炭資源は豊富であった）ドイツを（その功罪はさておいて）戦時下において生き永らえさせる技術の一つでした。そしてこのプロセスは現在、グリーン電力を用いた水の電気分解によるCO2フリー水素を原料とすることで、CO2を排出することなくガソリンをはじめとした燃料を合成できるプロセス、として見直しが進んでいます（図 2　グリーン水素を原料としたFT反応の利用フロー）[[3]](#footnote-3)。

ダイアグラム

自動的に生成された説明

図 2　グリーン水素を原料としたFT反応の利用フロー

ここまで述べてきた化学反応は、いずれも何等かの「触媒」上で起こるものです。高校化学の教科書にも記載があるとおり、触媒というのはそれが介在することによって化学反応を起こりやすくする―より正確には活性化エネルギーを低減し、同一の温度条件下における反応速度を向上する―ことができます。触媒の性能が優れていれば、より低い温度で（装置材料のコストダウンとエネルギー消費の低減）より速く（反応器の小型化、すなわちコストダウン）所望の生成物を得ることができます。これらは化学反応プロセスの経済性、つまりコスト競争力に直結することから、化学反応プロセスの開発競争はより優れた触媒の開発競争とも言えます。

触媒を開発する。一口に言えば簡単ですが、実際のところ一筋縄ではいかないものです。例えば先にあげたハーバー・ボッシュ法についていえば、触媒開発を担当したミタッシュは実用に資する触媒種の発見（これまた教科書に示される通り、その主成分は鉄となります）に至るまでに、2万種類にも上る物質を試したそうです。いやいや、そんなのは100年以上前のお話じゃないか、現代人にはコンピュータがある。コンピュータシミュレーションを使えば、触媒なんて簡単に設計できるのだろう。そう思うかもしれませんが、後々述べるように触媒上の反応を原子レベルで計算し、さらに欲をはって「よりよい触媒材料は何だろう」と探索するというのは、つい近年まで非常にコスト（お金と時間）のかかるものでした。

そんなわけで、2022年にPreferred Computational Chemistry社（PFCC）が超高速な汎用原子シミュレータ“Matlantis”を発表し、なかんずくその実用例として「コバルトを主成分とするフィッシャー・トロプシュ反応触媒に対する好適な添加元素の探索」が紹介されたとき、私はそれこそ信じられない、という驚きを隠せませんでした（図 3　MatlantisをFT反応触媒の添加物探索に用いた例）[[4]](#footnote-4)。そんなことが現実的な計算時間で手軽に検討できるとは、これまでの人生で夢にも思わなかったことだからです（多くの化学者からしても、おそらくそうであったことと思います―もちろん、先に挙げたコスト、すなわち時間と資金が無限大に用意されるのであれば別ですが）。PFCCが用いた技術は、ニューラルネットワークポテンシャル（NNP）というものです。NNPは、複雑かつコストの高い原子シミュレーションについての膨大なデータベースをあらかじめ人工知能に学習させることにより、原子シミュレーションの計算量を大幅に低減し高速化することができます（今話題のChatGPTがやっているのと、計算の種類としては類似です）。

グラフ

自動的に生成された説明

図 3　MatlantisをFT反応触媒の添加物探索に用いた例

さて、前置きが長くなりましたがいよいよ本題に入りましょう。今まさにこの文章を読んでいるあなたへの質問です。

「ここまで述べてきたような計算、例えば、NNPを用いたコバルトを主成分とするフィッシャー・トロプシュ反応触媒（気体と固体という異なる相を含むため不均一触媒と呼ばれます）に対する好適な添加元素の探索を、**あなた自身が、おうちのコンピュータでも出来るようになったとしたら**、どう思うでしょうか？ 」

いかがでしょう。なんと言うか、夢がありませんか。この字句をみて少しでもワクワクした貴方には、この先を読み進めていただくことを是非、おすすめします。

本テキストは、フリーのNNPであるOCP[[5]](#footnote-5)を用いた触媒上における反応シミュレーションの実装を主題とします。まさに今述べた通り、コバルトを主成分とするフィッシャー・トロプシュ反応触媒に対する好適な添加元素の探索（より具体的には、添加元素としてバナジウムが有効であることの確認）が、おうちでも簡易的に出来ることを確かめる、というのが最終目標となります。必要な前提知識は、高校化学レベルを想定しています。ですから、触媒化学を専門としていない方や、まだ大学に入学していない方など、計算化学的手法に対し馴染みのない方（何を隠そう私自身がつい先日までそうだったわけですが）にも抵抗感を抱かずに進めていただけるよう、数式による説明は極力避け、定性的な考え方やストーリーが伝わるようにしたつもりですので、気楽に読んでいただければ[[6]](#footnote-6)幸いです。

グラフ, バブル チャート

自動的に生成された説明

図 4　Co（111面）を触媒表面とするFT反応におけるCO乖離反応のOCPによる解析例（淡赤：Co、灰大：V、灰小：C、赤：O）。これを実習形式で学ぶことが本テキストの目標。

1980年代に眞鍋淑郎氏が地球温暖化のメカニズムを提唱してから30年以上が過ぎました。2023年8月現在、気候変動は（勿論どのような時間スケールを適用するか、また誰を主体とするか、どのような経済的目論見を持っているかなどといった背景情報により、この見方は多分な影響をうけますし、メカニズムを提唱した眞鍋氏自身、最終的にこれらは政治の問題になるだろうことを90年代には予見しておられましたが）一種のPoint of no returnを過ぎたと考えています。人類がこの現象にどう立ち向かうかが試される時代、このテキストを通じて一人でも多くの方が当技術に興味を抱いていただき、そして今や触媒化学のシミュレーションが、市井の好奇心旺盛なあらゆる人類にとって、開かれたものとなったことに気づいていただければ（さらに贅沢を申し上げるならば、気づいて手足を動かした方と、いつかどこかで出会えるならば）、同じく化学の麓に立つものとしてこれほどの喜びはありません。

2023/8/7

Masaki Open Lab

もくじ

[1. 本テキストの目標 1](#_Toc142734434)

[2. 原子シミュレーションの歴史 9](#_Toc142734435)

[3. コンピュータを用意しよう 32](#_Toc142734436)

[4. 必要なソフトを整備しよう 38](#_Toc142734437)

[5. シミュレーション：基本 50](#_Toc142734438)

[6. シミュレーション：応用 60](#_Toc142734439)

1. 原子シミュレーションの歴史

本章のまとめ：

・単一質点のシミュレーションはエネルギーと力が分かれば解ける（古典物理学）

・一方、原子や分子のシミュレーションは（単一原子、分子でも）難しい

・なぜならば、原子や分子は古典物理ではなく、量子性に支配されるから

・量子性に支配される原子や分子のエネルギーと力を求めるために、シュレディンガー方程式が作られた（量子力学）

・複数の原子核や電子を含む系について、シュレディンガー方程式を厳密に解くのは困難

・この複雑さを解消する目的で導入された手法のひとつが密度汎関数法

・ただし、密度汎関数法をもってしても、複雑な系（例：不均一触媒上における反応）を計算するのは大変だった

・そこで、様々な原子や分子に働く力を、古典物理的なパラメータ、すなわち力場に落とし込む（バネとおもりの系に無理やり近似する）分子（動）力学法が開発された

・分子力学法を用いれば、質点系で運動方程式を解くのと同じ要領で原子をシミュレーションできるという大きな利点があった

・しかし、分子力学法における力場は着目する系の種類（例：無機/有機）によって細かく異なるため、汎用的でないという欠点があった

・2022年、Facebookを中心とするグループにより、様々な固体触媒の表面上における様々な分子の吸着構造に関するビッグデータをAIに学習させることで、バルク結晶および不均一触媒上における反応に対し汎用的に使える力場（ニューラルネットワークポテンシャル）が公開された

* 1. 導入

この章ではまず、原子シミュレーションの歴史的経緯を大まかにまとめます。いやいや待ってくれ、シミュレーションの実装について学ぶというのに、今更どうして歴史を学ばなければいけないのだろう？　と気になった方も多かろうと思いますので少し説明をさせてください。

本テキストを手に取っていただいた皆さんは、仮に高校生であったとしたら、ある程度の確率で今後理科系の大学に進むことを想定されているのだろうと思われます。筆者の経験上、これから記載する「原子シミュレーションを支配する種々の法則群」については当然、多くの大学におけるカリキュラムに含まれています。ただ、初めてその法則群、中でも量子力学について講義を受けた時、多くの学生はこんな顔をすることになります。

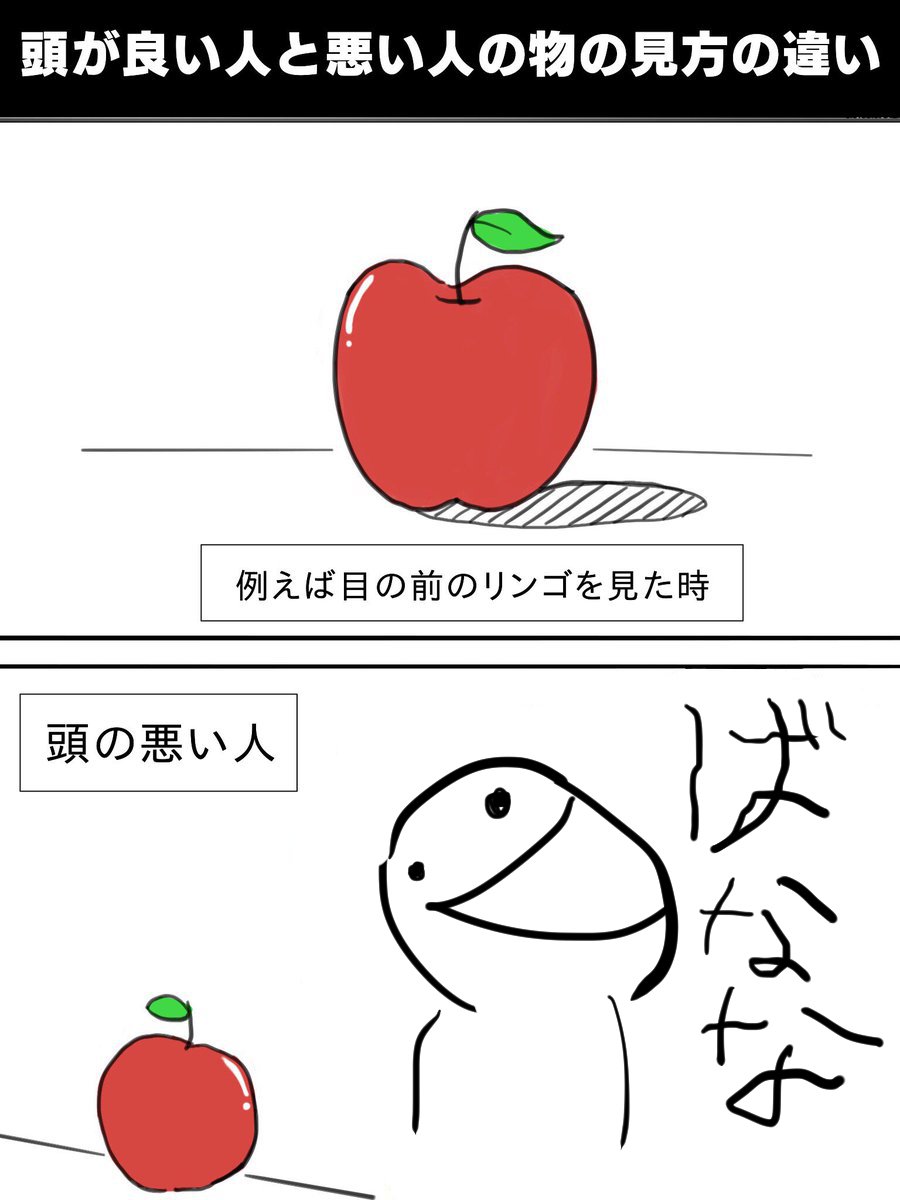


図 5　初めて量子力学を目にした学生のリアクション例[[7]](#footnote-7)

これはどうしてかと言いますと、まずは高校化学において暗記させられてきた原子の構造というのは、本質的に量子力学の内容を含むにもかかわらず、その背景にまで立ち入った説明を通常は受けてこなかったこと、またさらには、ようやっと大学で手にした量子力学の教科書は、古典論から量子力学の発展に至る歴史的経緯を無視した[[8]](#footnote-8)ものが多いために、高校過程で学んできた暗記事項と、大学で学ぶ量子力学との間にブリッジがかかりづらくなってしまうというのが本質的な原因であると私は考えています。以下に、筆者が好んで読んでいる教科書の文章を引用させていただきます。

*“たしかに、あたかも実験事実や理論的要求からの論理的必然として量子論が作られていったかのように書いてある書物は少なくないが、それを本当に論理的必然だと思ってしまったとしたら、誤読である。そこに挙げられている実験事実や理論的要求だけからは、古典論がダメであることは言えても、それにとって代わる理論体系としては、量子論が唯一の可能性とは決して言えない。（中略）幸か不幸か、比較的少数の実験事実しかなかった時代につくられた量子論の基本的枠組みが、その後の数多くの実験事実を説明・予言するのにも、全く無修正で済んでしまった。（中略）つまり、量子論は、実験事実や理論的要求からの論理的必然として出てきたものではないのだが、今のところ、なぜか驚異的にうまくいっている理論体系なのである。[[9]](#footnote-9)”*

ざっくばらんに申し上げれば、原子を支配する法則を理解するには、まず高校化学で原子の構造について学習した際、それが古典的に見て如何に破綻しているかという点に唖然とし、これにとって代わる量子力学がどのように登場したかを見聞きし、それが直感的には理解に苦しむ理論体系であるにせよ、[[10]](#footnote-10)恐ろしく現象をよく言い当てているという事実に対し、当時の科学者と同様に驚愕する、という心理的追体験を経なければ中々受け止めがたいように思います。従いまして、当テキストでは数式を避けながら、なるべくこの追体験が出来るということを重視して、量子力学を巡る科学史を紹介することにしました。なお、さらなる興味を持たれた方は、脚注に記載の参考図書を読み込むことで学習が深まるような構成としています。

また、「おうちで気軽に触媒上における反応シミュレーションを行うスキルを身に付ける」という題目は計算化学の歴史上、きわめてイレギュラーな物言いでもあります。というのは、初めに述べた通り原子シミュレーションというのは（特にそれが密度汎関数法を指すのであれば）つい最近まで、実用に供するにはかなりコストがかかる手法である、というのが一般常識であったからです。今回皆さんに実習していただくNNPの登場により状況は激変したものの、NNPが依拠する密度汎関数法による計算自体については、それがコストの高い手法であるということと、またそれが極めて難解な理論に基づいているという現実に変わりはありません。従ってそのイレギュラーさを知っておいていただくことは、将来何等かの形で化学に本格的にかかわる上で非常によいことだと考えています[[11]](#footnote-11)。

* 1. 原子をシミュレートすることの難しさ（古典物理学の破綻）

高校物理の授業で運動方程式（*F* = *ma*）を学んだことを覚えておられる方は多かろうと思います。釈迦に説法ではありますが、この方程式を使うと、質点系の運動を数学的に記述し、予測することができます。我々がスマートフォンやテレビでゲームを楽しむ際には、ごく単純に言えばこの運動方程式に[[12]](#footnote-12)基づいたモデリングを計算機内で繰り返し高速に行うことにより、例えばレーシングカーの動きや、TPSにおける主人公のアクションを逐次シミュレートしています。この物理は一般的に古典物理学とか、ニュートン物理学と呼ばれています。

夜の街の道路を走る車のcg

中程度の精度で自動的に生成された説明

図 6　グランツーリスモ7[[13]](#footnote-13)のプレイ画像。筆者は雪道での安全な運転原理を学ぶという目的で導入しましたが、ドライビングという物理現象をさながら現実のように表現するさまには舌を巻きました。

一方、高校化学で学ぶ原子の構造についてはどうでしょうか。通常であれば、ラザフォード型の原子モデルや、難関大を目指す方であればs軌道、p軌道、はたまたsp2、sp3混成といった軌道について暗記することを求められたかと思います[[14]](#footnote-14)。一方で、原子や分子がどのように運動するか、はたまた原子を構成する電子がどのような運動をするかといった事項については、おそらくほとんど触れられたことはなかったでしょう。これは、原子の構造というものが本質的に古典物理学と少なからず矛盾する難しさを持つからなのです。

例として、ラザフォード型の水素原子を考えてみましょう。水素原子は一つの陽子と電子からなり、電子はK殻に格納されます。より原子量の大きい元素においては、K殻は2個、L殻は8個というように、電子が順番に格納されていく…というように習いましたね。しかし、高校物理で電子の円運動を学んだ皆さんは少し奇妙に思わないでしょうか。「電子が入る軌道が決まっているというのは一体どういう原理なのだろう？」と。その疑問はもっともで、高校で学ぶ古典物理の中には、電子が入る軌道を定めるべき原理は見当たりません。そもそも、電子は原子の中でどのような状態にあるのでしょうか？　最も直感的な答えは、K殻という軌道を電子が（陽子との間に引力を感じながら）ぐるぐると公転するというものです（さもなければ電子は陽子にひきつけられてしまいますから）。しかし、この古典的な理解には大きな問題点があります。と言うのも、電荷をもつ粒子が円運動すると電磁波の形でエネルギーを放出してしまい、公転半径がどんどん小さくなって陽子に引きずり込まれてしまうからです。さらに、混成軌道まで話が進んでしまうと、一体どうしてそんなへんてこな軌道に電子が入るのか、そもそもその軌道の中で電子は一体どこにいるのかといったことはさっぱり分かりません（教科書には「電子はこの軌道内に確率分布する」という、これまた古典物理しか学ばない高校生から見て理解に苦しむ表現が記載されています）。こうした現象について理解するためには、舞台を19世紀末のヨーロッパに戻し、「光の単位」についてお話を始める必要がございます。

* 1. 黒体放射：古典物理学が崩れた日

みなさんにはまず初めに、筆者一押しの参考文献[[15]](#footnote-15)から冒頭の一節を紹介します。

*“量子の物語の幕が開くのは、19世紀の末である。当時、電子、エックス線、放射能が発見され、原子は本当に存在するのかという問題を巡って論争が起こっていたにもかかわらず、物理学者の多くは、大きな発見はすでに成し遂げられ、もう何も残っていないと信じていた。アメリカの物理学者アルバート・マイケルソンは、1899年ンに次のように断じた。「物理学の重要な基本法則や実験事実は全て発見され、しっかりと確立されているので、今後何か新しい発見がなされ、すでに確立されたことが打倒される可能性はきわめて低い。将来的に何か発見をするためには、小数点以下6桁目を探さなければならないだろう。」”*

19世紀末のドイツでは、白熱電球の登場により電気産業が幕開けを迎えていました。より効率のよい電球の開発を行うには、とりもなおさず光度の単位系を定める必要がありました。そこで科学者たちが着目した現象が、黒体放射と呼ばれるものでした。例えば炭火を思い浮かべていただくと、着火した炭は赤色の光を放ちますね。これは、「物体はその素材によらず（素材によらないので、この議論を行う際には放射体のことを黒体と呼びます）、所定の温度において同一の放射を持つ」という法則によるものです。この現象は当時すでに知られており（なにしろきわめて身近な現象ですから）、これを定式化できれば光度を定量化する上で一つのベンチマークになることが期待されたことから、研究が急速に進み始めました。その中でも有力視されていたものとして、ヴィーンの法則が挙げられます*[[16]](#footnote-16)*。

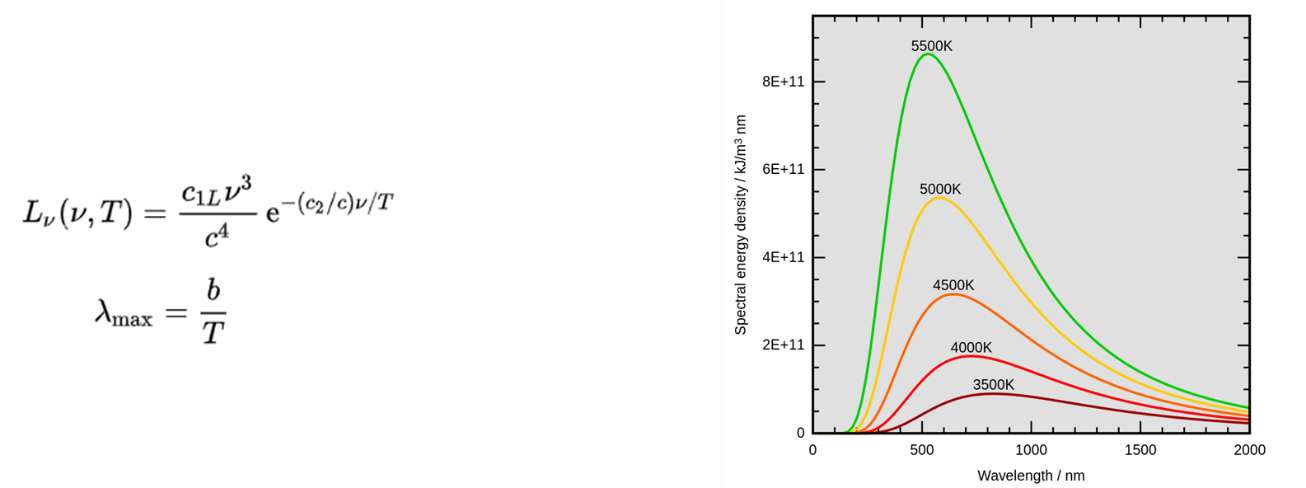


図 7　黒体放射に関するヴィーンの法則16

ヴィーンの法則は、黒体放射が最大となる光の波長は温度の逆数に比例する、というきわめてシンプルなものであり、実験結果とも「おおむね」合うことから広く受け入れられていました。ただし、長波長側（温度が低い側）において実験結果と無視できない差を生じるというのが課題でした。

そこで、ドイツの科学者プランクは長波長側でも合うようなヴィーンの法則の修正式を一つ、作ってみることにしました。それが、プランクの法則と呼ばれるものです。プランクの法則は、とりあえず作ってみたにしては現象をきわめてよく再現することから、当時の科学者にも特に抵抗なく受け入れられました。しかし、プランク当人にとっては「さてとりあえず作ってみたものの、この式は一体どうやって導出したものか？」という問題が残りました。

プランクはこの問題に対し、ボルツマンによるエントロピー最大化の統計的導出を利用することによって解決しました。つまり、黒体表面を架空の振動子の集まりとし（なにしろ素材によらないというのが黒体放射ですから）、熱平衡状態＝エントロピー最大という熱力学上の仮定を用いれば、プランクの法則を導出できることを示したのです[[17]](#footnote-17)。しかしながら、その過程で彼は2つの信仰転換を余儀なくされました。1つは、彼自身は信じていなかった原子論の推進者であるボルツマンに自身の導出における助けを求めたこと、そしてもう1つは量子化でした。

プランクによる導出の一部を図 8に示します。この式によれば、黒体放射におけるエネルギーは架空の振動子の振動数に比例します（*ε = hν*）。高校物理を学んだ方なら一目瞭然ですが、振動数というのは自然数ですから、この式がある以上、エネルギーはとびとびの値しか取れません。このとびとびの値しか取れない、というのを量子化と呼びます。

テキスト, 手紙

自動的に生成された説明

図 8　プランクの法則の導出論文（一部抜粋）17

量子化という概念がどれほど奇妙であったか、たとえ話で言うならば、ある日突然「これから速度の単位は整数になったから。1 km/h、2 km/hというのはあるけど、1.5 km/hというのは存在しないから。」とされるのと同じようなものですから、古典物理の世界に慣れた人々（この中には高校生も勿論含まれるわけです）からすると相当な抵抗感を覚える概念であったことは想像に難くありません。なお当のプランクはどうであったかと言いますと、彼自身にとってより重要であったのは振動数νではなく、比例定数の*h*でした。プランク定数として知られるこの定数は、未来永劫消えることのない絶対的な値であり、それを発見できたことの喜びの方が彼にとっては大きかったのです。そして、この振動数というのはあくまでも黒体という架空の系に対して、架空の振動子を仮定したために生じたものに過ぎないだろう。いつかは誰かが、よりスマートな解をもたらしてくれるだろう、と彼は信じて疑いませんでした15。

* 1. 光の量子化：架空の振動子から気体分子へ

ところが、プランクの期待は20世紀の初頭に早くも裏切られることになりました。先ほど示した通り、プランクは黒体を「架空の振動子」の集まりと考えたため、そこで生じるエネルギーの量子化というのはあくまでも架空の（便宜上の）ものであるという建付けをとりました。しかし、アインシュタインは異なるアプローチをとりました。彼は、黒体内部は架空の振動子ではなく、気体分子からなるという仮定をおき、改めて黒体放射のエントロピー[[18]](#footnote-18)がもつ体積依存性を調べました。すると、得られた答えは実在気体におけるエントロピーの挙動と一致することが分かりました。すなわち、黒体を構成するのは架空の振動子ではなくて、実在気体分子であること、したがって、量子化されていたのは架空の振動子ではなく実在する分子のエネルギーであり、その放射である光もまた量子化されているということが示されてしまいました。

* 1. ラザフォード型モデル

ちょうどそのころ、英国のラザフォードは原子の構造について有名なラザフォード型モデルを提唱しました。彼はガイガーおよびマースデンと協力して実験を行い、α粒子（ヘリウムの原子核）を薄い金箔に衝突させたところ、そのほとんどが透過する一方で、ごく一部が非常に大きく散乱することを突き止めました。この結果から、彼は原子のほとんどが空っぽであり、中心部に原子核が存在するという仮説を立てたのです（この内容については高校化学の教科書にも記載がありますね）。ラザフォード型の原子模型は、多くの人々に受け入れられる原子モデルとなったのは確かですが、先に示した通り「電子はどうしてエネルギーを放射せず、また原子核にひきつけられてしまいもせず安定に存在するのか」という問いには答えられませんでした。



図 9　ガイガー・マースデンの実験[[19]](#footnote-19)

* 1. 原子内部の構造を支配する物理：量子力学の登場

ラザフォードの弟子であったボーアという科学者は、このような矛盾に対して「電子が安定的に存在するのは、陽子にひきつけられて消えることもなく、電磁波を放射してエネルギーを失うこともない、量子化された（どこにいてもいいわけでなく、特定の条件を満たした）特殊な軌道に存在するからだ」という説（ボーアモデル）を提唱しました。このモデルは電子が特殊な軌道に入っているというのがどういうことなのか、について具体的には言及しないものの、当時話題になっていた原子スペクトル（電子がとびとびのエネルギー準位をもつこと）をよく説明し、かつ原子のエネルギーだけでなく、その内部構造も量子化されていることを指摘、アインシュタインによる光の量子化を引き継ぐ、大きな意義を持つものでした。



図 10　ボーアの原子模型[[20]](#footnote-20)

さて、電子がとびとびの（離散的な）エネルギーしか持ちえないとはどういうことなのかという点については、フランスの科学者であるルイ・ド・ブロイ[[21]](#footnote-21)が次のような説を提唱しました。「アインシュタインが喝破したように光は粒子性と波動性をもつのだから、原子を構成する電子にも粒子性とおなじく波動性があるのではないか？」　この説に基づくと、電子が離散的なエネルギー準位をとるのはそれが定在波条件であるからだ、という風に理解することができます。

折れ線グラフ

中程度の精度で自動的に生成された説明

図 11　ド・ブロイの物質波[[22]](#footnote-22)

続いてオーストリアのシュレディンガーは、物質波の考え方をさらに発展させました。彼は、物質波というものがあるならば、その形式を決める方程式（波動方程式）があるはずだ、という目論見でもって、電子が満たすべき波動方程式を探し出したのです[[23]](#footnote-23)。

この方程式はシュレディンガー方程式と呼ばれ、ψが波動関数、*E*が系のエネルギーに相当します。着目する系に対してシュレディンガー方程式を解けば、波動関数とエネルギーが求められる。つまり、原子シミュレーションにおいて必須となるエネルギー（と力）に関する情報を得ることができます。高校化学の教科書に登場するs軌道、p軌道というのは、この波動方程式から得られた波動関数（正確にはその2乗）を指します[[24]](#footnote-24)。

* 1. シュレディンガー方程式の問題点

シュレディンガー方程式の登場により、ようやく科学者たちは原子内部の描像を3次元空間内で具体的に議論できるようになったかと思われました。しかし、この期待は取り扱う電子の数が2個に増えた瞬間に裏切られます。シュレディンガー方程式を解けば確かに系のエネルギーが得られるのですが、波動関数の形は複数電子系になったとたんに可視空間をはるかに超えて多次元化し、そもそも現実的な計算時間で解けない[[25]](#footnote-25)という問題点がありました[[26]](#footnote-26)。以後多電子系、多原子系については何等かの形で多体問題を一体軌道形式に近似する、という手法が発展しました（例；ハートリー・フォック法）。ただし、こうした近似手法は多体問題ならではの性質を一部無視することにもつながるため、実際の現象とうまく合わないこともありうる、という点が悩みの種でした。そういった近似を行うことなく系の情報を3次元空間の一体軌道形式に落としこむ、というシュレディンガーの期待が荒唐無稽なのか、それとも事実として可能なのかが分かるためには、さらに長い年月を要しました。

* 1. 戦争の世紀

閑話休題、量子力学が発展した20世紀は、人類にとって未曾有の戦争と虐殺を経験する時代でもありました。とりわけ、第二次世界大戦以前に最もこの分野が発展していたのがオーストリアおよびドイツであったことは、その後の歴史に深い影を落としています。

1930年代以降、ナチスドイツが台頭するにつれて、ドイツおよびその周辺国における科学者たちもまた血生臭い争いに引きずり込まれていきました。アインシュタインをはじめとする多くのユダヤ人科学者は、運と才気と資金を駆使して、第三国を経由してアメリカに逃れました（さもなければ、ナチスドイツによる所の「最終的解決」が彼らを待つだけでした）。また一方で、非ユダヤ人ドイツ科学者達にとっても、世に倣って反ユダヤ主義に染まるか、それとも黙して身を引くか、あるいは身を賭して異を唱えるかという難しい選択が突き付けられました[[27]](#footnote-27)[[28]](#footnote-28)[[29]](#footnote-29)。

そのようにしてドイツを逃れた科学者の一人に、ウォルター・コーンが挙げられます。1923年に生まれた彼は、1939年にイギリスを経由してアメリカ大陸（カナダ）に逃れました（両親はアウシュビッツで死亡）。なぜ彼の名前をもちだしたかと言いますと、彼こそがこのテキストで取り扱う計算化学の礎となる、密度汎関数法を生み出し、3次元空間で電子を理解するというシュレディンガーの目論見に対して、50年越しの回答を与えた人物であったからです。

* 1. 密度汎関数法：シュレディンガーへの回答[[30]](#footnote-30)

先に述べた通り、多体問題となるシュレディンガー方程式を解くためには、多体空間で罹れていた高次元の記述形式をどう取り扱うか、を考えねばなりませんでした。これに対し、1964年に発見されたホーエンベルク・コーンの定理は、以下の回答を与えました。

「いかなる多体空間における記述形式に対しても、**それと等価な答えが与えられるような一体軌道形式**は存在する」

つまり、シュレディンガーが願った3次元の一体軌道形式への落とし込みは理論的に可能であり、現実的な計算時間で厳密に解くことができる、ということを保証したのが、ホーエンベルク・コーンの定理です[[31]](#footnote-31)[[32]](#footnote-32)。これをきっかけとして、汎関数を探し出そう[[33]](#footnote-33)という形で発展してきたのが、現代一般的な計算化学的手法の一つとなっている、密度汎関数法（Density Functional Theory, DFT）です[[34]](#footnote-34)。



図 12　ウォルター・コーン（1923-2016）

* 1. 分子（動）力学法

DFTの登場により、系のサイズがある程度大きくともエネルギーと力を計算することができるようになりました。とはいえ、DFT計算もまたコストが大きいことに変わりはありません（多体問題をそのまま解くのは「現実的に不可能」であるのに対し、DFTで解くのは「まあ大変」くらいの認識です）。特に20世紀はコンピュータの計算速度もまだまだ遅かったため、もっと別のやり方で原子シミュレーションを行いたいという機運が高まっていました。

そこで登場したのが分子力学法です。分子力学法は、様々な分子内における結合長や角度のデータベース（実験、もしくはDFTなどの計算化学手法により求めたもの）を上手く予測できるような、分子の結合に関するいわばバネと錘からなる古典的モデル（力場といいます）を「別に」作ってしまおう、という手法です。この手法を用いれば、シュレディンガー方程式を解かずとも、古典物理学的なシミュレーションと同様に原子シミュレーションを行うことができるという大変大きな利点があり、特にタンパク質や分子の分野で広く使われるようになりました。ただし、結合の種類や登場する元素によって力場のパラメータ群をいちいち変更する必要があることから、汎用性に欠けるというのが悩みの種でした。

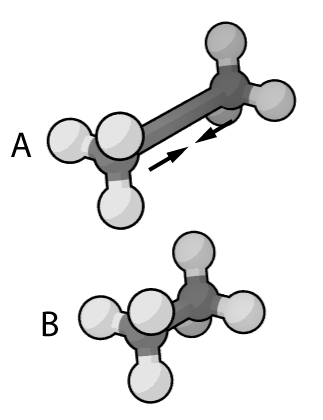


図 13　エタン分子の結合長と働く力の模式図（Wikipediaより引用）

* 1. ニューラルネットワークポテンシャルの登場：AIとビッグデータの時代

ここまでの歴史をまとめてみましょう。原子の内部構造は量子化されているため、原子シミュレーションにはエネルギーと力を計算するために、量子力学に従うシュレディンガー方程式を解く必要があります。この方程式は多電子系になると厳密解を得るのがとんでもなく難しいという弱点がありましたが、DFTを用いることで、ある程度複雑な系に対してもこの方程式を現実的な計算時間とある程度の厳密さで解くことが可能となりました。一方で、DFTの計算コストは依然として高いことから、実験値および計算化学による計算結果のデータベースをもとに、古典物理モデル（力場）に近似して簡易に予測するという分子力学法が発展しました。しかしながら、様々な元素を含む系に対して自由に使えるという汎用性には難がありました。

さて、2010年代に入ると、ディープラーニングを筆頭とするAIと、これが学習の元とするビッグデータという分野が一気に台頭するようになりました。ディープラーニングが初めて脚光を浴びたのはILSVRC 2012[[35]](#footnote-35)という画像認識精度を競うコンテストがきっかけであり、以後様々なAIモデルのコンペティションにおいて集中的に使用されるようになりました[[36]](#footnote-36)。

ディープラーニングが何をやっているかについて念のため概説します。例えば画像認識の問題を考えた場合、1枚ごとに何の画像であるか（犬、猫など）がラベリングされた膨大なデータベースを用意します。このデータベース（画像の場合はピクセルごとにRGB数値が入力されたもの）をディープラーニングに学習させると、画像とそのラベルとの相関関係を何とかして把握することができます。そうすると、後は画像さえ入力すれば、それが一体何の画像であるかというラベルを出力（予測）することができます。

机の上にあるコンピューターのスクリーンショット

中程度の精度で自動的に生成された説明

図 14　例えば食べごろのセミを見分けることもできる、という例[[37]](#footnote-37)

計算化学の分野においても、計算化学のビッグデータを作成するという機運が高まりました。そして、2011年にはMaterials Projectが発足しました。このWebページには、幅広い種類の化合物に関する結晶構造のDFT計算結果が保存、公開されています（2023.8月時点で15万種類以上）。

先ほど述べたように、分子力学法は様々な元素に幅広く適用できる汎用的な力場の開発に課題がありました。一方で、計算化学に関するビッグデータと、ディープラーニングに代表されるAIの手法があれば、様々な元素を含む系に対して、力とエネルギーを予測するモデル、すなわち汎用力場が作れるのではないでしょうか？　この発想に基づいて設計されたのが、いよいよ本テキストで皆さんに体験いただくニューラルネットワークポテンシャル（NNP）です[[38]](#footnote-38)。

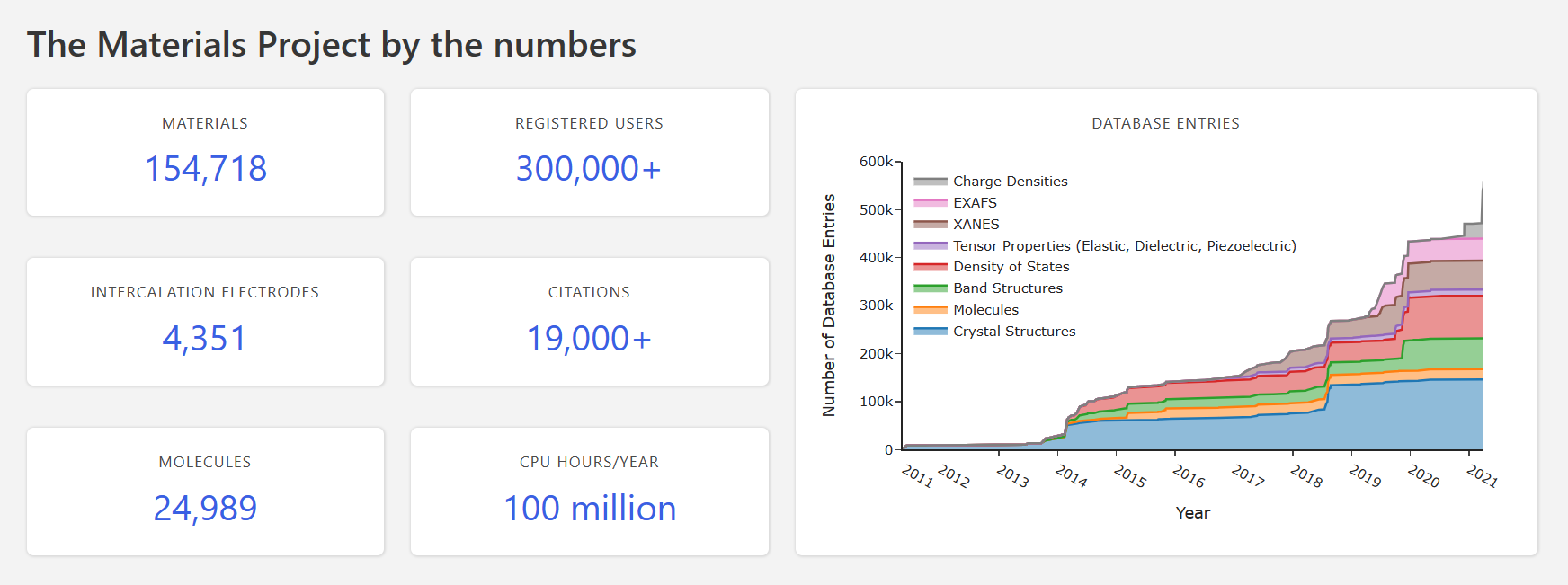


図 15　Materials Projectの概要[[39]](#footnote-39)

2020年、Metaおよびカーネギーメロン大学を中心として、Open Catalyst Projectが発足しました。当プロジェクトについて、overviewの冒頭を抜粋して紹介します。

*“再生可能なエネルギーを利活用することは、気候変動への対処として必要不可欠になりつつあります。一方で、太陽光や風力をはじめとするこれらの電力は気象条件により変動することをまぬかれません。従って、我々の需要に応じてこれらのエネルギーを供給するためには、大規模なエネルギーの貯蔵技術が必要です。1つの方法としては、これらの電力を一旦水素やメタンなどの燃料として、化学反応（電気化学的反応）を通じて変換し貯蔵することでしょう。ただし、これら反応を経済的に実現するためには、高性能な触媒の開発が不可欠です。ここで、DFT（密度汎関数理論）による原子シミュレーションは、触媒開発における強力な武器となり得るものの、原子シミュレーションの高い計算コストは、探索可能な構造の数を制限するという課題がありました。そこで本プロジェクトでは、不均一触媒反応に関する大規模な計算化学データセット（Open Catalyst Project OC20 dataset）を使用し、DFTを代替するAIモデルの開発を目指します。”[[40]](#footnote-40)*

気候変動への具体的な解決策として計算化学を紹介し、その課題であった計算速度を抜本的に向上するためにビッグデータとAIを利用しようという、これまで本テキストで説明してきた流れとほぼ合致するAgendaであることが分かります。このプロジェクトでは、先ほど紹介した計算化学ビッグデータのMaterials Projectから取得した構造に対して、様々な低分子の吸着構造をDFTにより生成し、構造とそれに紐づいたエネルギー（および力）をAIに学習させることにより、任意の結晶もしくはそこから切り出した表面（スラブと呼びます）における各種低分子の吸着構造について、DFTを大きく凌駕する速度での原子シミュレーションを可能にしました（AIとビッグデータによるOCP「力場」の獲得）[[41]](#footnote-41)。さらに、このOCP力場については、2022年にフリーで（無償で）公開されました[[42]](#footnote-42)。

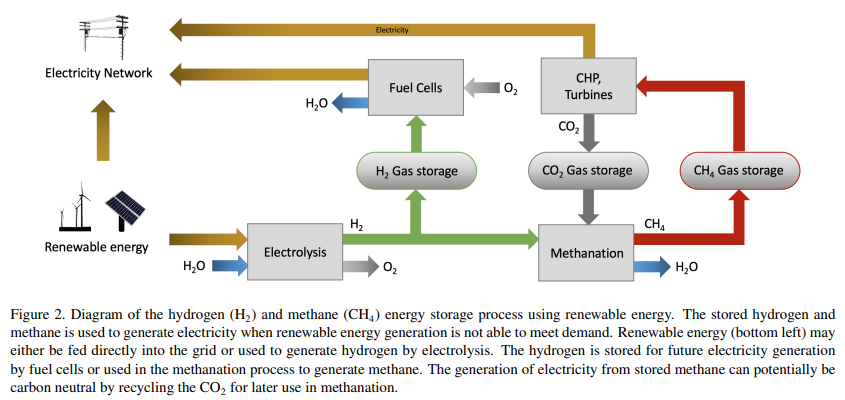


図 16　Open Catalyst Projectより引用、再生可能エネルギーの化学反応による変換と貯蔵

ダイアグラム

自動的に生成された説明

図 17　OC22データセットの概要[[43]](#footnote-43)

さて、ようやっと皆さんにOCP力場について説明することができました。次からはいよいよ、OCP力場を使った原子シミュレーション[[44]](#footnote-44)に向けて手を動かしていきたいと思います。

1. コンピュータを用意しよう

本章のまとめ

・ハイスペックでなくてもよいので、GPU付きのコンピュータを購入しよう

・中古品でよければ、5万円あればなんとかなる

・利便性を求めるとノート型が好ましいが、価格の面からはデスクトップ一択

突然ですが質問です。「ゲーミングPCをお持ちですか？」

Yesと答えたあなたは、おそらくこの章を読み飛ばしていただいて結構です。そうではないという方は、この章を参考にコンピュータを準備ください。

* 1. どんなコンピュータが必要か

コンピュータを購入するときには、利用目的はなんなのか、というのをはじめにきちんと定義するのが最重要です。なにしろ、コンピュータというのはまあほんとうにピンキリの世界でして、上を見ても下を見てもきりがありません。さらには、インターネット上でおすすめの情報を手に入れようと検索したとしても、「少なくとも●●万円以上じゃないと話にならない」とか「中古PCは手をだしちゃいけない」とか、まあ有象無象の断定的情報（その多くは、書き手が持つ暗黙の前提というのを暗黙のままにしています）があふれていますから、初学者の方はおそらく情報の取捨選択に困るだろうと思います。こういう時に一番助けになるのは、例えば大学の研究室であれば先輩や教授陣となるのですが、誰もが気軽に大学の研究室と連絡を取れるわけではありません[[45]](#footnote-45)。

ということで、本テキストでは以下のような要件定義に基づいて、コンピュータの選定購入方法についてまとめてまいります。

・OCP力場を使った原子シミュレーション（基礎的な研究）に耐えうること

・大学における基本的な作業（論文、プレゼン、会議）にも後々耐えうること[[46]](#footnote-46)

・初期投資を可能な限り抑えること

* 1. 推奨スペック

以上の点を踏まえると、推奨されるスペックは下記の通りとなります。参考サイトを見ていただければわかりますが、これらの要求値は昨今のコンピュータとしてはそれほど高いものではありません（CPUスコアが8000未満となると、正直オフィス作業に支障をきたすレベルです…だからと言って私企業のPCがおおむね8000以上を満たしているわけでもない、というのが日本の辛い所です[[47]](#footnote-47)）。本テキストを読んでいただいている方が、仮に機械学習を少しかじった経験があったミドルエイジであった場合「そんな装備で大丈夫か[[48]](#footnote-48)」とついつい突っ込みを入れたくなるような値なのです。どうしてこの程度のスペックでよいかというのを機械学習用語で申し上げると「学習と予測では必要な計算資源が桁で違うから」だ、というのが回答になります（が機械学習について議論するのは本テキストの主旨と異なりますので、脚注に追いやります）[[49]](#footnote-49)。なお、特にこれまでコンピュータに触れてこなかった読者の方からすると、この記載ではよく分からない！ということもあろうかと思いますので、次の項に具体的な例と購入方法について示します。

表 1　OCP力場を使った原子シミュレーションに必要なコンピュータのスペック35



* 1. 購入方法とスペック例

新品で購入する場合には、価格.comなどで最低価格を見極めることがまず第一手となるでしょう。例えば、ゲーミングノートというくくりで先ほど記載のスペックを満たすものを検索すると次のようになります。おおむね10万円前後というところでしょうか。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

図 18　必要なスペックを満たすゲーミングノートPCの例[[50]](#footnote-50)

デスクトップPCで探すと以下のようになります。ノートタイプに比べて、同一価格でもややスペックの高いGPUが選べるようです（使ってみればわかりますが、冷却性能の点からもデスクトップの方が有利です）。

グラフィカル ユーザー インターフェイス

中程度の精度で自動的に生成された説明

図 19　同、デスクトップPCの例50

さて、大学1年次に税込157500円も出してCore 2 duoのPC（メモリなんて2GBで、それでも多い方でした。ムーアの法則おそるべしです）を購入した筆者からすると、その6割の出費でこれほど潤沢な計算機資源を手に入れられるというのは何と言いますか、正直言って今の学生さんがうらやましくて仕方がないのですが、さりとて、さりとてです。例えば高校生にとって10万円の出費は大変ですから、これだけでは心もとないでしょう。何とかならないだろうかと思ったそこの貴方には、中古PCという選択肢を紹介いたします。

例として、下記はパソコン工房[[51]](#footnote-51)から引用した中古PCとなります。CPU、ストレージ、GPU、メモリ共に非常に贅沢な仕様ですが、中古であれば先に示した新品に比してとても安価に手に入れることができます。

テキスト

自動的に生成された説明

図 20　中古ゲーミングPCのスペックと価格例[[52]](#footnote-52)

さらにスペックを下げると、こちらのような品物も見つかります。スペックを下げたと言っても、当方が所有しているゲーミングノート（4年落ち）とほぼ同等の性能です。本テキストが目標とする、OCP力場を用いた原子シミュレーションには十分耐えるでしょう[[53]](#footnote-53)。

テキスト が含まれている画像

自動的に生成された説明

図 21　5万円未満で手に入るゲーミングPCの例52

1. 必要なソフトを整備しよう

本章のまとめ

・Anaconda / Visual Studio / Gitの3つをインストールする必要があります

（個人が利用する限り、無償です）

・その後、ocp-modelsをインストールしてください

* 1. Anaconda

本テキストで使用するocp-modelsは基本的にPythonで記述されています。従ってここからは、Pythonによる計算環境を皆さんに構築していただくこととなります。最近の高等学校における情報課程では、一部PythonやＲも内容に含まれているようですので、皆さんの中にはすでにAnacondaをインストール済みの方もいらっしゃるかもしれません。その場合はこの項をスキップして次項に進めてください。

Anaconda[[54]](#footnote-54)はPythonを使用した開発をやりやすくするために作られたソフトウェアです。脚注に記載のリンクから、Windows (64 bit)版をダウンロード、インストールしてください。

テーブル

自動的に生成された説明

図 22　Anaconda installers (Python 3.9版)54

* 1. Visual Studio

続いてVisual Studio[[55]](#footnote-55)をインストールします。脚注のリンクからインストーラをダウンロードしてください。その後、どのソフトをインストールするかを問われますので、C++を使用した開発に該当するものをインストールするようにしていただければOKです。

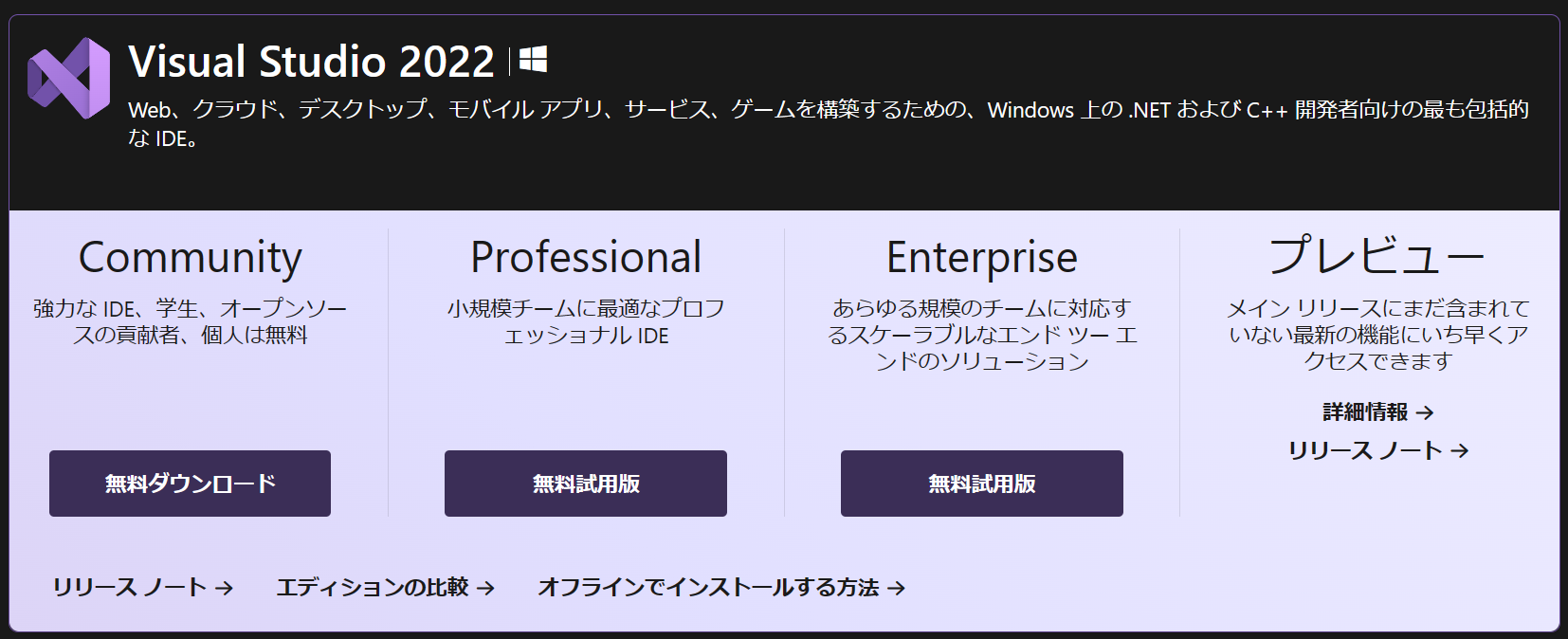


図 23　Visual Studio 2022のインストール（Community版を使用）55

* 1. Git

最後にGit[[56]](#footnote-56)をインストールします。同じく脚注に記載のリンクからWindows版をダウンロード、インストールください。

グラフィカル ユーザー インターフェイス, Web サイト

自動的に生成された説明

図 24　Gitのインストーラ56

* 1. ocp-modelsのインストール

脚注[[57]](#footnote-57)に記載の手順に従い、ocp-modelsをインストールします。まずはGPU計算に必要なライブラリであるCUDAをリンク先から手に入れましょう。

テキスト

自動的に生成された説明

図 25　CUDA 11.6のインストーラ[[58]](#footnote-58)

続いて手順の3以降に移ります（以下、NanoLaboの記載情報を引用いたします。手順2はどちらかというと会社のネットワークにて環境構築するときのお話ですので、本テキストにおいては割愛します）。Anaconda Promptを開き、下記のコマンドを順に入力してください（リンク先からコピペするのが最も早いです）。手順の3を行うと、ユーザフォルダの直下にocpフォルダが出来ますので、cd ocpと入力してocpフォルダに移動してください。その後、手順4に進んでください（GPUで計算する場合、の手順を実行）。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図 26　ocp-modelsインストール手順57

以下「ここで環境により…」の部分について説明しておきます。先ほどの手順4では、ocp-modelsのインストールに必要なパッケージ群がリストアップされたenv.ymlを作成しました。このenv.ymlはocpフォルダに入っていますので、エクスプローラ（📁マーク）から直接開いて修正することができます（テキストエディタ形式で開きます）。ymlファイルが開いたら、下記の修正を反映してください。

グラフィカル ユーザー インターフェイス

自動的に生成された説明

図 27　ocp-modelsインストール手順（ymlファイルの修正事項）57

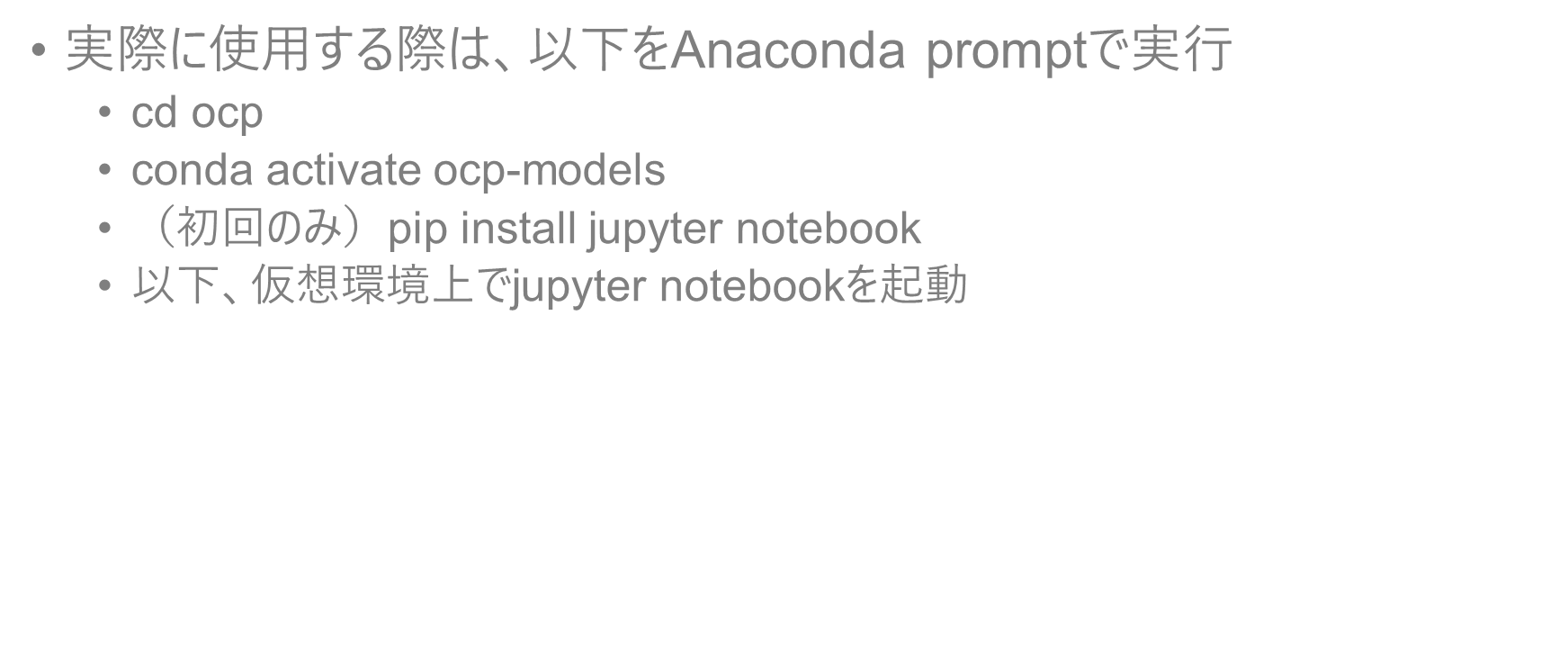
ymlファイルの修正が終わりましたら、保存してファイルを閉じたのちAnaconda Promptの画面に戻ります。下記のコマンドを入力することで、ocp-modelsが仮想環境としてインストールされます。完了しましたら、conda activate ocp-modelsで仮想環境を有効にし、pip install -e.でパッケージとしてインストールします。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

図 28　ocp-modelsインストール手順（仮想環境の作成と完了）57

実際に使用する際には以下の手順となります。Anaconda Promptを開いて、下記のコマンドを入力ください（Jupyter notebookが立ち上がります）。



* 1. 備考：エラーに出会ったら

コンピュータで何かをやろうとしたとき、いつの世も一番面倒なのは計算環境の構築におけるエラーとの闘いです。Pythonやそのライブラリ群はOSS（オープンソースソフトウェア）であり、ユーザ自身がその開発整備に携わり逐次更新されるという特徴から、テキストに記載されている事項と、本テキストの読者が実際に計算環境を整備しようとした時点における最新情報が異なる、ということが往々にしてあります。それどころか、ここで参照したNanoLaboに記載された事項ですら、最新の情報と相違することもあるのです（経験談）。従いまして、計算環境を構築するにあたっては常に何某かのエラーに遭遇することはほぼ間違いありませんから、より重要なのは「エラーが出た時にどうやって何とかするか」という所です。ここでは大まかな考え方を示しておきます。解決の方策は大きく、自分で調べるか、誰かに聞くかという2種類に分かれます。

自分で調べる場合には、エラーメッセージが何を意味しているかを理解することがポイントとなります。例えば下記NanoLaboに記載の情報では、エラーメッセージの例が示されています。このエラーが意味するのは「ymlに記載されたそのバージョンのパッケージはどこにもなかったよ」ということです。ですので、解決方法としては、じゃあ提供されているバージョンは幾つなのだろうというのを調べて、ymlファイルに記載されたパッケージのバージョン番号を実在するものに書き換える、という風になりますね。この例は極めて単純なものですが、より複雑に見えるエラーメッセージの場合はとりあえずChatGPTに丸ごと投げてしまう、ということも手です。少なくともChatGPTはエラーメッセージの内容について解説することができますから、これを参考にして次の手を考える、ということはしばしば役に立ちます（昨年秋まではググるというのが先に出たのですが、ChatGPTの出現以降ググる、という行為の優先順位は急速に下がりつつあります…とはいえ有効であることには変わりありません）。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

図 29　エラーへの対応に関するヒント57

もう一つの手段、誰かに聞くという手について最も直接的なのは「ocp-modelsを作った人に尋ねる」という方法です。以下は、ocp-modelsのgithubにおけるissues（相談）の一例となります[[59]](#footnote-59)。ここでは、H2O分子のエネルギーをOCP力場でもとめたところエラーが出た、という相談が持ち掛けられました。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

図 30　ocp-modelsにおけるissueの例

これに対する回答がこちらです。構成原子数が4つ以下の場合は、OCP力場での計算ができないという内容です（分子のエネルギーは単なる参照なので、これが求められなくとも吸着構造の最適化において特に問題はありません…後述）。

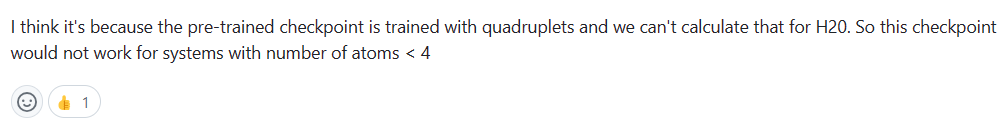
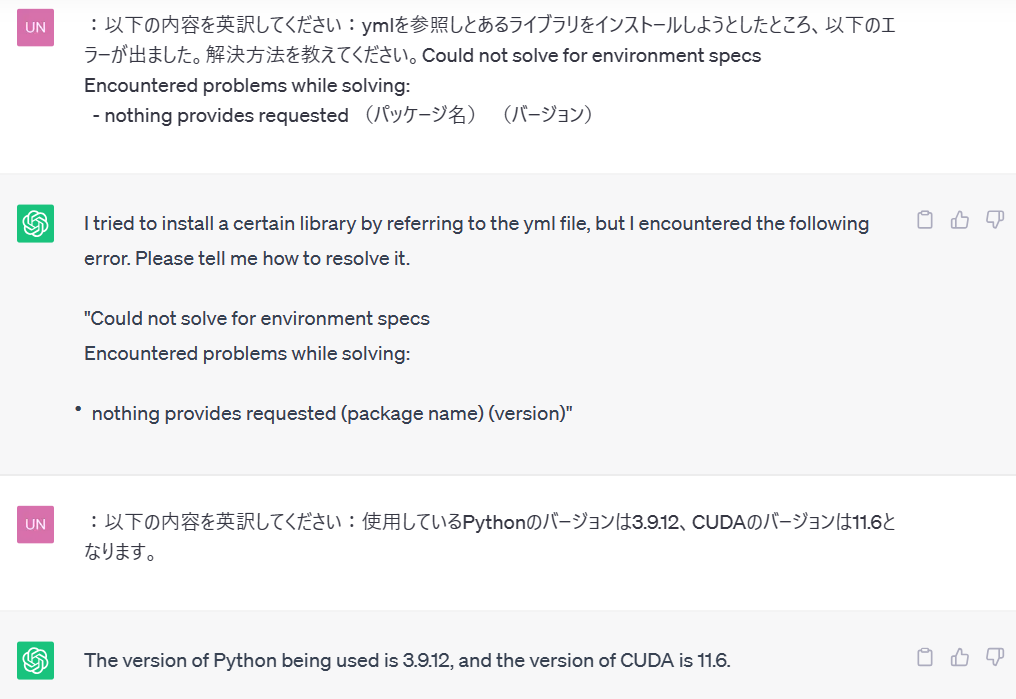


図 31　同issueへの回答例

如何でしょう？　英語なので少し抵抗があるかもしれませんが、内容としてはかなり基礎的というか、ユーザ目線の素朴な質問であることが分かるかと思います。Issuesにはこのように、作られたライブラリについての様々な質問が挙げられ、開発者からの回答が格納されています。勿論、計算環境の構築におけるエラーについてもissuesに質問することは可能です（し、実際にそうした質問もしばしばあります）。ということで、どうしても解決できないエラーが出た際にはぜひ活用を検討いただければと思います。

なお、実際にissuesに質問を投稿する際には、質問内容を英語で記載する必要が出てきます。従って、一旦ChatGPTに質問内容を英訳させたのちに投稿するという手順をお勧めします。またエラーの内容に加えて背景情報が分かるよう、使用した計算環境（例：Pythonのバージョン…今回の場合では3.9）も可能な限り記載すると解決が早まります。それから、issuesを活用（GitHubへの登録が必要です）する際には極力「実名を使用しないように」注意ください[[60]](#footnote-60)。



-------------------その後「もう少しビジネスライクにまとめてくれ」と依頼------------------

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

図 32　ChatGPTにissuesへの投稿内容を英訳させた例

1. シミュレーション：基本

本章のまとめ

・不均一触媒上における反応は複数の「素反応」の組合せ

・このうち、最も活性化エネルギーの高い素反応を律速段階とよぶ

・律速段階をゼロから調べることは難しいが、多くの興味ある反応において律速段階は公知

・着目する触媒上における律速段階について、その活性化エネルギーが計算出来ればよい

・活性化エネルギーの計算には、反応物から生成物にいたる「アニメーション」が必要

・すなわち、構造計算をコマ数分だけ膨大な回数行う必要があるためかなり大変

・そこでよく使われるのが、BEP則

・活性化エネルギーは、反応物と生成物のエンタルピー変化に比例する

（つまり、はじめと終わりの2枚だけ絵があればよい）

・反応物-触媒と生成物-触媒の吸着構造についてエネルギー計算と最適化を行いたい

・そのために、Cu上におけるCOの吸着エネルギー計算について実習する

・OCP力場を使うと、吸着構造の最適化が十分に速く行える

いやはやお疲れ様でした。実は4章の環境構築が一番七面倒でして、これを乗り越えていただいたみなさんは、後は優雅に原子シミュレーションの世界を楽しむことが確約されたようなものです。おめでとうございます！　ということで、本テキストの最終目標である「不均一触媒上における反応のシミュレーション」に向けた実習を進めてまいります。まずは触媒シミュレーションについての（今回取り扱う）全体像と考え方を紹介します[[61]](#footnote-61)。

* 1. 不均一触媒上における反応シミュレーションの全体像

例えばハーバー・ボッシュ反応について、高校化学の教科書にはN2 + 3H2 → 2NH3という化学反応式が記載されています。これはoverallな反応式としては正しいのですが、不均一触媒（この反応系においては主としてFe）上においていきなり窒素と水素がアンモニアに変化するかというと、そうではありません[[62]](#footnote-62)。より具体的には、以下のようないくつかの化学反応を経てアンモニアの生成に至るということが知られています。このいくつかの化学反応を、素反応と呼びます。そして、素反応の中で最も活性化エネルギーが高いもの（つまり最も遅く、化学反応の速度を決定するもの）のことを、律速段階と呼びます。ハーバー・ボッシュ法においては特に、Fe表面上に吸着した窒素分子が乖離して2つの窒素原子になる反応が律速段階とされています。

テキスト, 手紙

自動的に生成された説明 コンピュータ, 異なる, 束, カウンター が含まれている画像

自動的に生成された説明

図 33　ハーバー・ボッシュ法の素反応62

すなわち、不均一触媒上における反応のシミュレーションにおいては、最も進みづらい律速段階の反応に着目すればよい（その他の素反応は律速段階と比べ物にならないくらいさっさと進むので、無視することができる）、とする考え方はしばしば使用されます。ここで、我々にとって興味のある化学反応（本テキストの主旨に沿って言えば、いわゆるグリーンケミストリーに関連した化学反応）においては、多くの場合素反応と律速段階については実験的もしくは基礎的な計算化学検討により、ある程度解明されております[[63]](#footnote-63)。

以上より、我々が不均一触媒上における反応をシミュレートしたいと考えた際には、当該反応の素反応および律速段階についての情報を収集し、しかる後に律速段階となる反応の活性化エネルギーを求めればよい（そうすれば反応速度論より、反応速度係数を*k =* exp*(-Ea/RT)*で求めることができるから、触媒としての良しあしが測れる）ということがなんとなく理解できたかと思います。さてそれでは、活性化エネルギーはどのように求めればよいのでしょうか？

反応活性化エネルギーを求める計算化学的手法の一つとして、NEB法（Nudged Elastic Band）が挙げられます。NEB法は、反応の始点と終点における構造（例：COの吸着構造→CおよびOの吸着構造）が分かっている場合に、始点と終点を結ぶ最も活性化エネルギーの低い経路を探索する手法です。当手法には、うまくいけば、反応始点から終点に至るエネルギー経路が得られるだけではなく、原子シミュレーション上でその変化を視覚的に（アニメーションとして）理解することもできるという利点があります。ただ一方で、反応パスのアニメーションを得られるということはとりもなおさず、その各コマとしての構造計算（力とエネルギーの算出）を膨大に繰り返さなければならないことをも意味します（漫画を描くよりも、アニメ化する方が、描く絵の枚数が増える分大変）。また、NEB法による計算自体、望ましい反応パスに収束しないこともしばしばあるという実用上の問題があります。エネルギーと力の計算をDFTにて実施する場合、一つの触媒、素反応に対してパスを生成するのに年単位の時間を要することも珍しくはありません[[64]](#footnote-64)。今回実習するNNPを用いれば、もちろん計算時間の大幅な短縮は見込めますが、それでも個人の保有するGPUで行うにはかなり重い計算であることには変わりありません（し、触媒開発担当者としては少なくとも実験より速く計算が進んでくれないと―探索の手立てとしてコンスタントに使用するには―困ってしまいます。探索ではなく機構解析の目的であればこの限りではありませんが）。

グラフ, バブル チャート

自動的に生成された説明グラフ, 折れ線グラフ

自動的に生成された説明

図 34 NEB法による金属表面上におけるNO乖離反応の計算例（OCP力場を使用）

こうした場合に用いられる大胆な近似手法の一つとして、BEP（Bell-Evans–Polanyi）則が挙げられます。BEP則とは、同じ種類の化学反応（例：様々な触媒表面上におけるNOの乖離反応）について、その活性化エネルギーは反応エンタルピーに正比例するというものです。

これにより、触媒シミュレーションは大幅に単純化されます。つまり、何某かの触媒候補材料が、参照となる触媒に対して優れた性能を持つかを調べるには、触媒候補材料および参照となる触媒における、反応物と生成物の吸着構造というたった２枚の絵を作成（構造最適化）し、そのエネルギー差を比較すればよい、ということになるからです（厳密にいうともう少し検討が要るのですが、それは比較的些末なことなので、応用編の実習で話しましょう）。

スイング, チェーン, ロケット, 消火栓 が含まれている画像

自動的に生成された説明 チェーン, スイング が含まれている画像

自動的に生成された説明

図 35　要するにこの2枚が出せればよい、という例（OCP力場にて計算）

* 1. 実習：Cu上におけるCO吸着エネルギーの計算
     1. 必要なライブラリとOCP力場の導入

ここからはGitHubに格納したノートブック[[65]](#footnote-65)に従って実習を進めていきます。はじめに、必要なライブラリ[[66]](#footnote-66)および、OCP力場をインポートしましょう（OCP力場については脚注[[67]](#footnote-67)よりダウンロードし、ocpフォルダの直下に格納ください。またjupyter notebookに記載のパスについては、各個人のユーザ名に変更のこと）。

このスクリプトについて説明しますと、OCP力場を使う際には、ASEというオープンソースのライブラリを使用します。そこで、ASEに関連したパッケージをインポートしたのち、ダウンロードしてきた学習済みの力場（パラメータの値がある程度最適化されたディープラーニングモデル）をcalcという変数として定義する、という操作を行いました。このcalcという変数がASEにおけるとても便利な点の一つでして、仮に我々が近い将来、OCP力場よりも優れた別のNNPに出会ったとしても（ほぼ確実に出会うでしょうが）、その力場がASEに対応している限りは、ここのcalcを別のcalculatorに置き換えるだけで、以下のスクリプトを特に変更せずに使用できるのです（計算化学においてはどういったcalculatorを使うか、つまりどうやってエネルギーと力を算出するかという部分さえ決まってしまえば、その他の手続はほぼ共用です[[68]](#footnote-68)）。

テキスト, アプリケーション

自動的に生成された説明

図 36　ASEにてサポートされている力場の一覧[[69]](#footnote-69)



* + 1. 吸着構造の作り方

ライブラリの導入が完了したところで、Cu(111)面のontop[[70]](#footnote-70)におけるCO吸着構造の計算における考え方を示します。我々が今やりたいのは、金属表面にCOが吸着した構造として「一番もっともらしい」ものを探し出すこと（構造最適化）です。ここで、一番もっともらしいというのはすなわち「エネルギー的に最安定である」と言い換えられます。

簡単な例として、水素分子の結合距離を変えた場合を考えてみましょう。原子間距離*r*とポテンシャルエネルギー*E*の関係は図のようにあらわされます。原子間距離が大きくなると、ポテンシャルエネルギーは0に漸近します（原子のポテンシャルエネルギーを基準点にとるため）。原子間距離が小さくなると、ポテンシャルエネルギーは緩やかに減少し、*r*fにて最小値をとります。さらに原子間距離を小さくとった場合は、原子核同士の反発が大きくなるため、ポテンシャルエネルギーが急激に上昇します。先に言った構造最適化とは、初期の構造（原子間距離*r*i）からはじめて、最もエネルギーの低い*r*fを探すことに他なりません。

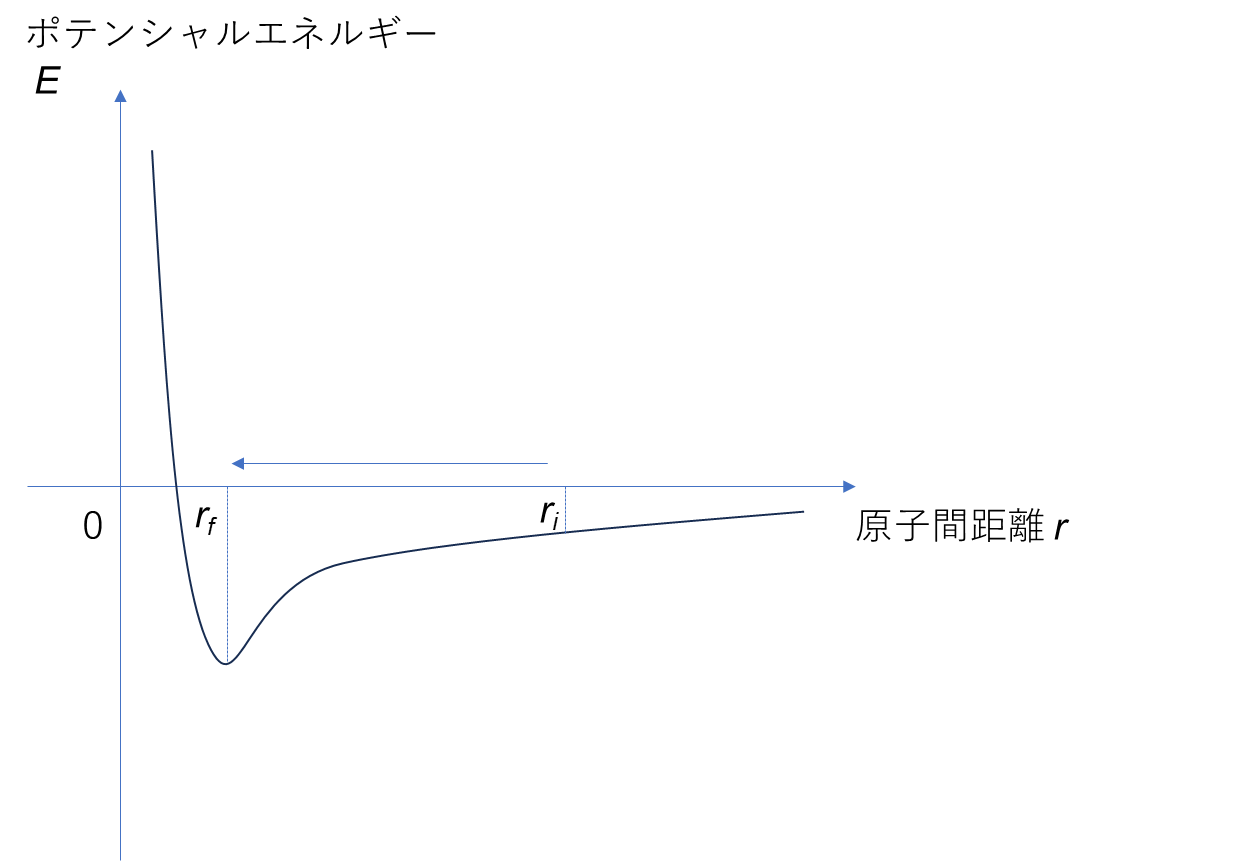


図 37　水素分子のポテンシャルエネルギーと原子間距離（イメージ）

より複雑な系であるCu（111）面のontopにおけるCOの吸着構造においても、原理的にやっていることは同じです。すなわち、吸着構造の初期値を作成し、そこからエネルギー的に最も安定な構造を目指して少しずつ構造を変化させていくのです（頭でイメージできる範囲を超えているので、この構造を少しずつ変える、という所には計算機を用います）。具体的な構造最適化の手順を以下に示します。

・Cuの結晶構造を生成し、格子定数を最適化

・Cuの表面を切り出し（スラブ）、構造最適化

・スラブ表面の所定位置にCOを吸着させ、構造最適化

（・吸着前後の構造から、吸着エネルギーを算出）

* + 1. 格子定数の最適化

それでは、着目する化合物について（ここではCuを例とします）格子定数をOCP力場のもとで最適化します。格子定数というのは、高校化学で学ぶ単位格子（例：面心立方格子）のサイズを指します。ASEには様々な機能を持つ関数が入っており、例えばCu結晶を作るのであればbulk(“Cu”)とするだけで初期構造を生成することができます。またその他にも、先ほどお話したMaterials Projectから結晶構造ファイルをダウンロードして、これを初期構造に用いるという手もあります。ただこれらの方法によって生成した初期構造というのは、OCP力場のもとで最安定の構造とは限りません（例：OCP力場にて計算した結果、デフォルトの格子定数は過剰に小さい（超高圧下条件）こともあります）。そこで、作成した構造については、格子定数が最安定となるように最適化計算を行う必要があります。

格子定数を最適化する手法として、ここでは下記の関数を定義しました。この関数は、指定した範囲で格子定数を変化させ、最もエネルギーが低いときの格子定数を求めるものです（先の水素分子で言えば、原子間距離を変化させながらポテンシャルエネルギーを測定し、最もその値が小さい所を探すことに相当します）。



この関数を用いて、Cuの格子定数を最適化した結果を示します（皆さんも適宜スクリプトを走らせてみて下さい）。ほぼほぼデフォルトの格子定数に最適値がある、ということが分かったかと思います。



ここで示したのは最も単純な構造最適化手法であり、想定しうる構造に対してひたすらエネルギーを算出するという代物でした。ASEではさらに高度な構造最適化のアルゴリズムが複数実装されています（ただOCP力場はstress parameterを出力する機能がないため、格子定数の最適化については今のところ本手法を用いる他ありません）。以後、結晶構造から表面を切り出したスラブモデルの構造最適化において、これらアルゴリズムの強力さを体験いただくことになります。ここではその大まかな概念について、先に説明しておきます。

先に示したように、構造最適化というのは初期構造を少しずつ変化させながら、最も安定な構造を探索するというものです。先ほどの「全部の構造についてエネルギーを確認する」というやり方以外に、よく使用されるものとしてFIREおよびLBFGSというアルゴリズムが挙げられます。FIREは分子動力学を応用した手法であり、先に挙げた水素分子の例で言えば「分子の片方を初期位置に置いて、後は場の力（エネルギーの傾き）に任せて転がそう」という考え方を発展させたものです。

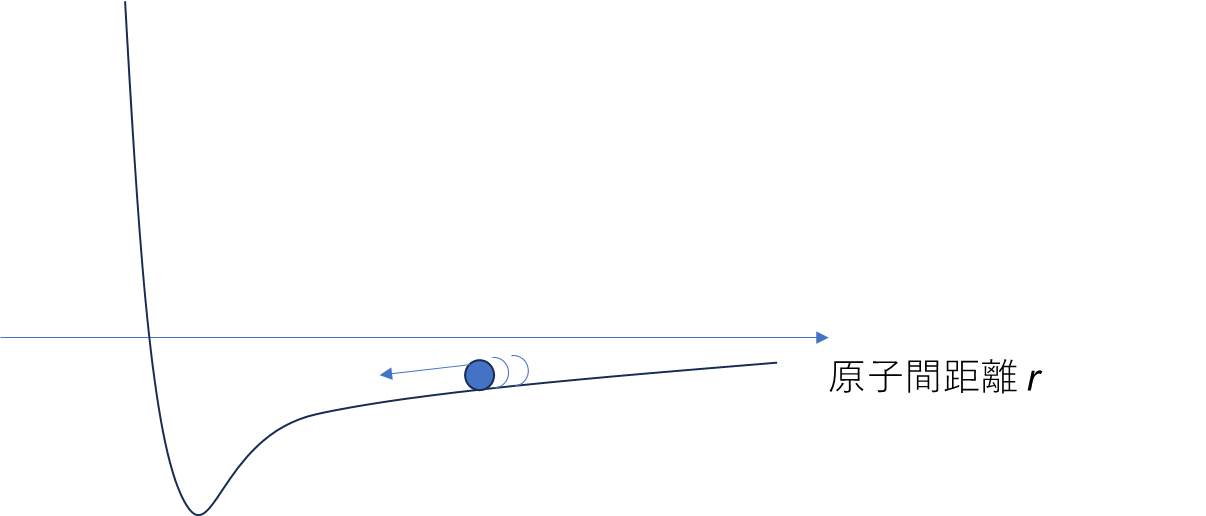


図 38　FIREによる構造最適化のイメージ

この方法による構造最適化は比較的安定して収束するのですが、場の力に任せるという性質上、初期構造における力（エネルギーの傾き）が小さい場合に収束が遅くなること、また最安定構造の周辺で力が振動しがちであり、収束を遅くすることという2つの弱点があります。

これに対し、LBFGS法はニュートン法に準ずる最適化手法（ただし、様々な工夫により計算量を削減したもの）です。先ほどの水素分子に類似したケースを想定して説明しますと、ニュートン法においては初期構造*r*0における力の接線を引き、その*r*切片を*r*1、再度接線を引いて…という手順の繰り返しにより、力が0となる（エネルギーが最小となる）構造を探索します。この手法はFIREに比べて一般的に収束が早いことから、かなりの頻度で使用されます。ただしこちらは、初期構造の選び方が好ましくない、あるいは構造変化に対して力の変化が非常に大きいようなケースについては、計算が収束せず発散してしまうという点に注意が必要です（FIREと強み、弱みが逆なので、どちらかがうまくいかない場合にもう片方を試す、という使い方がしばしば取られます）。

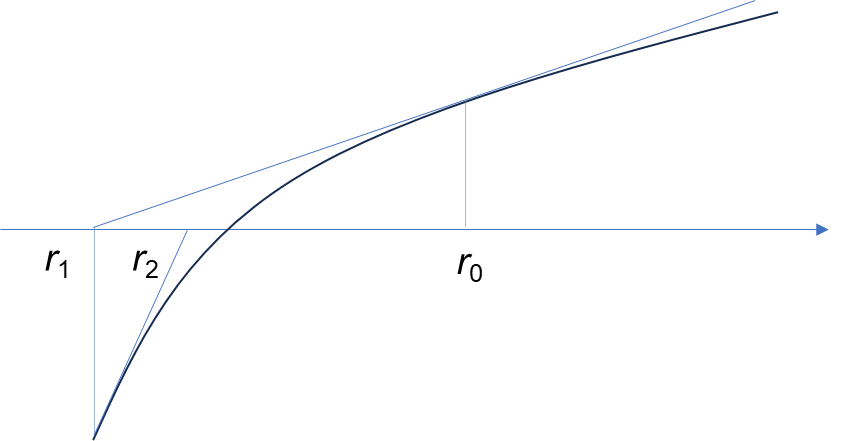


図 39　ニュートン法による最適化のイメージ

* + 1. 表面の切り出し（スラブモデルの構築）と構造最適化

続いて、最適化した格子定数を使用し、Cuの結晶構造に対して111面を作成、これを構造最適化します。「あれれ、格子定数はすでに最適化してあるのに、どうしてもう一度表面を構造最適化する必要があるの？」と思われた方もいらっしゃるかもしれません。これは、表面の原子が一般的に結晶内部（バルクといいます）に比して不安定であるためです。

青色で示した原子からなる結晶を例にとって説明します。この結晶構造に対し、真ん中の面をスパッと切りだして新たな表面を作成することを考えます。すると、オレンジ色の原子が表面に出てくるわけです。ここで、もともとこの表面原子は直上の原子と結合していたのですから、表面に出るというのはすなわち結合の手が余る状態に相当し、不安定な構造となってしまいます[[71]](#footnote-71)。従って、表面にいる原子にはなるべく自分達が露出する部分、つまり表面積を小さくしようとする力が働くため、その力を反映して再度構造最適化を行う必要があるのです。

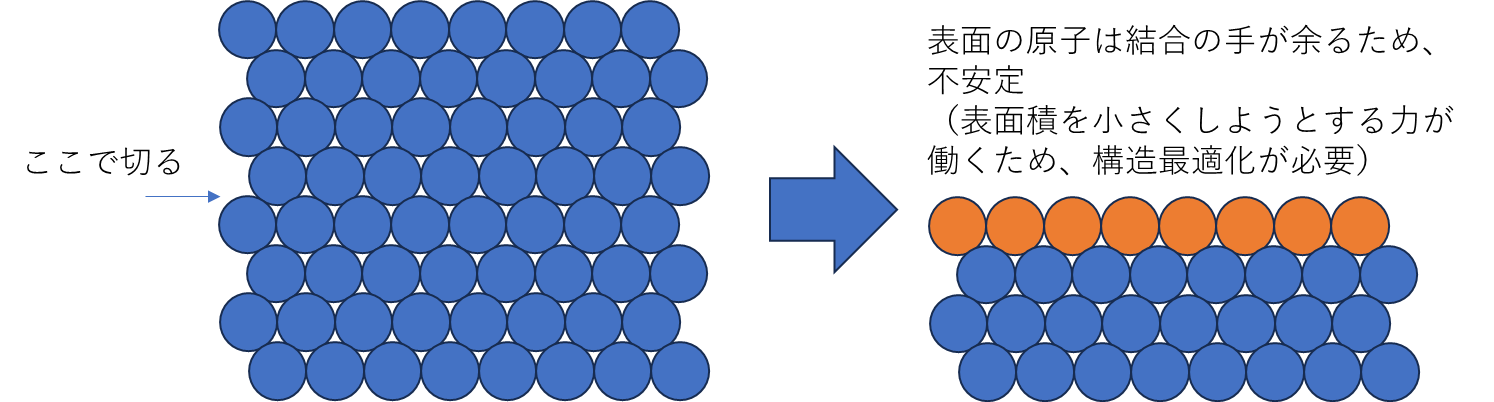


図 40　表面の切り出しによる不安定化のイメージ

それでは、以下のスクリプトに従ってCu(111)面の構造最適化を行ってみましょう。この関数は、先ほど定義したopt\_bulk関数を用いてCuの格子定数を算出したのち、これを反映したCu(111)面のスラブモデルを作成します。続いて、スラブモデルの構造最適化を行います。ここで、スラブ3層のうち、最下層の原子をFixAtoms関数にて固定します（ここでは直接固定する原子を指定しましたが、規則的に自動で固定する方法ももちろんあります）。この束縛条件は、OpenCatalyst力場の学習データにおいて、表面の2層以外は固定するという束縛条件が適用されていたことを反映したものです（この束縛条件を外すと、学習データに入っていない条件で最適化を行うこととなるため、不安定な挙動を示す可能性があります。個人的にはOCP力場にてスラブモデルの構造最適化を行う際には、表面2層以外を固定することをお勧めします）。束縛条件の指定後、LBFGS法にてスラブモデルの構造最適化を行います。引数のfmaxは、力の値がこれよりも小さくなったら収束したとみなしてよい、という条件、stepsはこの最適化計算を何回繰り返してもよいか、という最大値を著します。



実際に計算を開始すると、以下のように構造最適化が行われます。Timeの値をみればわかる通り、かなり高速に計算が行われることが分かります。この速度感が、計算化学で検討を進める上では非常に重要となります[[72]](#footnote-72)。



* + 1. COの吸着構造生成と最適化

Cuの表面を最適化し、そのエネルギーが算出出来たら、今度は以下のスクリプトに従い、Cu(111)面上にCOを載せ、構造最適化を行います。とはいえ、一口にCu(111)面上の吸着、と言っても、Cuとの相対位置によって吸着エネルギーは様々な値をとります。従って、実際に触媒反応を議論する場合には、何処に置くかで話が変わる、という点をどう料理するかが非常に重要となります（第6章で取り扱います）。今回はとりあえず、(x, y) = (1, 1)という点をCOの初期位置としました（add\_adsorbate関数のoffset = (1, 1)という引数です。直前の3というのは、表面からどれだけ離した位置にCOを配置するかを意味します）。



この関数を走らせると下記のようになります。単なるスラブモデルの最適化よりは時間がかかりますが、それでも文字通り、あっという間に構造最適化が完了しました[[73]](#footnote-73)。



最後に、COの吸着エネルギーを算出します。吸着エネルギーは一般に以下の式で定義されます。

（分子の吸着エネルギー）＝（吸着構造のエネルギー）- （スラブ単体のエネルギー + 分子単体のエネルギー）

要するに、分子が吸着したことによってどれだけエネルギー利得があったか、というのが吸着エネルギーです。ここで、先ほど申し上げた通り、OCP力場においては分子単体のエネルギーをその力場の中で求めることを想定していないため、分子のエネルギーについては各原子のエネルギーを仮定し、その総和で表します（繰り返しになるかも知れませんが、吸着エネルギーを議論するときに重要なのはその相対値である、と考えた場合、分子単体のエネルギーはあくまで参照すべき定数でしかないためです[[74]](#footnote-74)）。



実際にスクリプトを走らせると、Cu上におけるCOの吸着エネルギーが算出されます。



* 1. 参考：計算および実験値との比較

参考として、OCP力場の精度を確認するため、種々の金属表面上におけるCOの吸着エネルギーを計算し、複数の汎関数[[75]](#footnote-75)を用いたDFTによる計算値、および実験値[[76]](#footnote-76)と比較した結果を示します。ばらつきはあるものの、典型的な構造についてはおおむね傾向を再現することが確認されました。

グラフ, 折れ線グラフ

自動的に生成された説明

図 41　種々の金属表面におけるCO吸着エネルギー（黄色がOCP力場による計算値）

1. シミュレーション：応用
   1. Coを主成分とする触媒上におけるFT反応の解析と好適な添加元素の探索
      1. 方針とポイント

ここからはいよいよ、Coを主成分とする触媒上におけるフィッシャー・トロプシュ（FT）反応の好適な添加元素探索に取り掛かっていきます。簡単のために、本実習で示すべき事項を一度整理しておきましょう。5(ア)にて議論したように、触媒としての良しあしは律速段階となる素反応の活性化エネルギーで評価され（当然ながら活性化エネルギーは小さいほどよい）、またBEP則によれば、同種の反応に関する活性化エネルギーは反応前後のエンタルピー変化に正比例することが知られています。さらに、FT反応の律速段階となる素反応は、触媒表面上におけるCOの乖離反応です。従って、我々がすべきは、触媒表面上におけるCO、およびCとOの乖離吸着構造を最適化し、そのエネルギー差を触媒種ごとに比較することだ、と言い換えられます。

ここで-1を掛けたのは、ここで議論されるエネルギー（エンタルピー）は原子を0に取ったものであり、基本的に全て負の値をとる（負の値であるほど安定）ためです。もう少し言い換えれば、C-O乖離吸着エネルギーがCO吸着エネルギーに比してより安定であれば安定であるほど、活性化エネルギーは小さく、触媒性能は高くなると考えられます。

ということで、本反応に関する触媒性能の良しあしを判断するには、C-Ｏの乖離吸着構造およびCOの吸着構造を、触媒候補物質について最適化すればよいことが分かりました。そして、今我々はすでにCOの吸着構造最適化について学んでいますから、この計算はそれほど難しくないのではないか、と予想することができます（実際、難しくないのです[[77]](#footnote-77)）。ただし、吸着構造の最適化においては前章で学んだ基本的な計算手法に加えて、もう一つ考慮に入れておくべき事項があります。それは「どこに吸着することが最適か」という点です。

前章の実習においては、吸着分子となるCOをスラブ表面の(x, y) = (1, 1)なる地点に配置し、その後構造最適化を行いました。しかし実際には、たとえ均一な表面でも吸着サイトにはいくつかの種類があります。例えば典型的なfcc(111)面の場合、ontop、bridge、fcc、hcpの4種類が挙げられます[[78]](#footnote-78)。従って、とある表面における最安定な吸着構造を得たい、と考えるのであれば、吸着分子の初期配置を複数作成、構造最適化し、その中で最も安定なものを採用することが必要になります（後ほど実習にて行います）。

グラフ, バブル チャート

自動的に生成された説明

図 42 吸着サイトの例

さらに、C-O乖離吸着になるとより話は複雑になります。というのもこの場合は、Cがどこに吸着し、Oがどこに吸着する、という組合せが最適であるか、という組合せ問題を解くことが必要になるためです。本テキストでは簡単のため、まず片一方の原子についてどこに吸着することが最適かを求めたあと、この原子が吸着した状態に対してもう一方の原子を追加で吸着させ、最も安定な吸着位置（配座、conformation）を求める、という手順をとります。また、このconformation最適化においては、計算の高速化を図るためにoptuna[[79]](#footnote-79)というアルゴリズムを使用します[[80]](#footnote-80)。

さて、文献4に示されているように、Coを主成分とする触媒上におけるFT反応は、バナジウム（V）を添加することで促進することが知られています。そこで今回は、Coのみで構成される触媒と、Coの一部をバナジウムに置換した構造について、上記の吸着エネルギー差を計算し、Coの一部をバナジウムに置換することで、C-O乖離吸着構造がより安定化することを確認してみます。

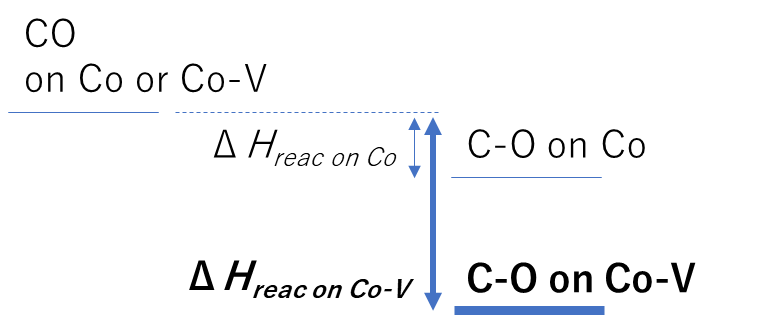


図 43　計算結果のイメージ[[81]](#footnote-81)

* 1. その他

1. おわりに
   1. 自分で研究をやってみよう
   2. なぜ学ぶのだろうか？

1. 大気を変える錬金術 ハーバー、ボッシュと化学の世紀（トマス・ヘイガー、みすず書房） [↑](#footnote-ref-1)
2. [【高校化学】「ハーバー・ボッシュ法」 | 映像授業のTry IT (トライイット) (try-it.jp)](https://www.try-it.jp/chapters-9578/sections-9579/lessons-9626/point-2/)

   以下本テキストにおいては、引用した図については著作権の都合上、枠で囲って明示いたします。 [↑](#footnote-ref-2)
3. [JPEC\_report\_No.220302.pdf (pecj.or.jp)](https://www.pecj.or.jp/wp-content/uploads/2022/03/JPEC_report_No.220302.pdf) [↑](#footnote-ref-3)
4. <https://arxiv.org/abs/2106.14583> [↑](#footnote-ref-4)
5. [Open Catalyst Project](https://opencatalystproject.org/) [↑](#footnote-ref-5)
6. 果てはPC購入法についての雑談（経験談）まで正式な章に組み込んだテキストですから、どうぞ気楽に。 [↑](#footnote-ref-6)
7. <https://twitter.com/niewals/status/869526811020939264?s=20>　 [↑](#footnote-ref-7)
8. というのも歴史的経緯が量子力学の筋だったマニュアルになるかと言えばそうでもない、というくらいにはこの分野色々と人間ドラマに満ち溢れていたからであって、筋だったマニュアルを整備することを目標にすると教科書のスタイルが大分変ってしまうのです。 [↑](#footnote-ref-8)
9. 新版　量子論の基礎　その本質のやさしい理解のために（清水　明、サイエンス社） [↑](#footnote-ref-9)
10. 量子論が当時の科学者たちに与えた衝撃は、多くの名言として世に残っています。 [↑](#footnote-ref-10)
11. 本テキストにおいては、ニューラルネットワークポテンシャルを用いた原子シミュレーションに連なる科学史という構成をとりましたが、原子シミュレーションについてはこれにとどまらず様々な分野が開拓され続けている、ということを注記しておきます（初学者の方は、ニューラルネットワークポテンシャルというのは魅力的な原子シミュレーションの「一つ」だという認識をとっておいていただくのが適当かと思います）。 [↑](#footnote-ref-11)
12. 非現実的なゲームであれば当然、運動方程式にもまた非現実的な改変が加えられることになります。またゲームで取り扱うのは質点ではなく剛体ですので、運動方程式の複雑さは数段階上がる事は言うまでもありません。リアルさを損なわずにどうやって数学的に単純化するかというのは、ゲーム開発における腕の見せ所の一つなのかもしれません（少なくとも私はそういった目でグランツーリスモをプレイしています）。 [↑](#footnote-ref-12)
13. [グランツーリスモ７ - 製品情報 - グランツーリスモ・ドットコム (gran-turismo.com)](https://www.gran-turismo.com/jp/products/gt7/) [↑](#footnote-ref-13)
14. 高校化学で暗記する事項というのは、実のところゆとり教育が批判された平成時代の方が、昭和時代の教科書よりもはるかに膨大であることは実はあまり知られていません。ちなみに当方はゆとり教育のさなかに学生時代を送りましたが、その当時混成軌道についてはあくまで備考欄という取り扱い、現代ではこれも本文に格上げされたようです。のちに述べる通りこれらは大学課程で学ぶ量子論の範疇であり、高校生を果てしない暗記祭りに導かざるを得ない一因となっています（高校課程を批判しているのではなく、これは科学の進歩がもたらす一種の必然であると筆者は考えています）。 [↑](#footnote-ref-14)
15. 量子革命（マンジット・クマール、新潮文庫） [↑](#footnote-ref-15)
16. [RWP-comparison - ヴィーンの放射法則 - Wikipedia](https://ja.wikipedia.org/wiki/%E3%83%B4%E3%82%A3%E3%83%BC%E3%83%B3%E3%81%AE%E6%94%BE%E5%B0%84%E6%B3%95%E5%89%87#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:RWP-comparison.svg) [↑](#footnote-ref-16)
17. [https://web.archive.org/web/20111006162543/http://theochem.kuchem.kyoto-u.ac.jp/Ando/planck1901.pdf](https://web.archive.org/web/20111006162543/http:/theochem.kuchem.kyoto-u.ac.jp/Ando/planck1901.pdf) [↑](#footnote-ref-17)
18. 脚注が必要だと思っていたのですが、最近調べたところ新課程の高校化学ではエントロピーも含まれたとのこと。いやはや若い方は大変です。 [↑](#footnote-ref-18)
19. [ガイガー＝マースデンの実験 - Wikipedia](https://ja.wikipedia.org/wiki/%E3%82%AC%E3%82%A4%E3%82%AC%E3%83%BC%EF%BC%9D%E3%83%9E%E3%83%BC%E3%82%B9%E3%83%87%E3%83%B3%E3%81%AE%E5%AE%9F%E9%A8%93) [↑](#footnote-ref-19)
20. [Brighterorange](https://commons.wikimedia.org/wiki/User:Brighterorange) - Created by [Brighterorange](https://commons.wikimedia.org/wiki/User:Brighterorange" \o "User:Brighterorange), based on GFDL/cc image: [Bohratommodel.png](https://commons.wikimedia.org/wiki/File:Bohratommodel.png). [↑](#footnote-ref-20)
21. ついでながら彼は公爵です。 [↑](#footnote-ref-21)
22. <http://www.sf-fantasy.com/magazine/column/quantum/200311.shtml>　 [↑](#footnote-ref-22)
23. 量子力学　基礎からのアプローチ（真船　文隆、化学同人）…高校生の方は必ずしも読む必要はありませんが、高度な数学をそこまで使用せずにシュレディンガー方程式を導出する過程が分かりやすく述べられています。また、多電子系、多原子系に持ち込む際に使える近似手法についても詳しく述べられています。 [↑](#footnote-ref-23)
24. というわけで、高校化学の教科書に載っている原子構造を物理学の発展と照合して説明するとこのくらいのページ数を食ってしまうのです…読んでいただいたみなさん、お疲れ様でした。引き続きお楽しみください。 [↑](#footnote-ref-24)
25. [できるだけ簡単に密度汎関数理論（Density Functional Theory, DFT）を説明してみる（前編） - Qiita](https://qiita.com/dc1394/items/1bb597b27d7511ce31ba) [↑](#footnote-ref-25)
26. シュレディンガー方程式によって原子内部の3次元的な描像が得られる、という目論見はそのまま、シュレディンガー自身が量子性という「直感的に理解しがたいもの」に対して強い違和感を持っていたことの表れでもありました。20世紀初頭の偉大な科学者をしても、量子力学という分野は心理的に受け入れがたいものだったのです。同時代を生きたハイゼンベルクという科学者は、シュレディンガー方程式と数学的に等価な行列力学という形式を発見しましたが、数学的に等価であるにもかかわらず両者陣営の間にはしばしば学問上の―そして世界をどう見るかという哲学上の―戦いがありました。1927年に行われたソルヴェイ会議では、多次元化する波動関数を3次元空間に落とし込むことへのシュレディンガーの期待に対し、それを裏付けるものはないというコメントをハイゼンベルクが残しています。詳細は「量子革命（マンジット・クマール、新潮文庫）」を参照ください。 [↑](#footnote-ref-26)
27. エルサレムのアイヒマン（アーレント、みすず書房） [↑](#footnote-ref-27)
28. 全体主義の起源（アーレント、みすず書房） [↑](#footnote-ref-28)
29. 部分と全体（ハイゼンベルク、みすず書房） [↑](#footnote-ref-29)
30. ホーエンベルク・コーンの定理をシュレディンガーへの回答と考える人はあまり見たことがありませんが、量子論発展の歴史と計算化学による材料設計を深く愛する筆者にとって、この歴史観はきわめて魅力的です。 [↑](#footnote-ref-30)
31. より詳細な話はいくらでもありますが、今回は計算化学の大まかな歴史を初学者の方に追体験いただくことを主眼とするため、このくらいの記載にとどめさせていただきました。 [↑](#footnote-ref-31)
32. 「動かして理解する 第一原理電子状態計算（前園　涼、森北出版）」の、特に第5章を強くお勧めします。 [↑](#footnote-ref-32)
33. 本文に記載したように、汎関数の存在は証明されたものの、それがなんであるかについてはまた別問題です。 [↑](#footnote-ref-33)
34. 一般的とは言え、コーン氏がノーベル賞を受賞したのは僅々1998年のことでした。30代の筆者からするとつい最近の出来事に思えます（初代ポケモンが大人気のころでした）。 [↑](#footnote-ref-34)
35. 2012年のコンテストにおいて優勝したのはジェフリー・ヒントンらのチームでした。当時、大規模なニューラルネットワーク（ディープラーニング）の有用性については未知の領域であり、今のようにだれでも使えるライブラリや計算機資源が用意されたわけでもありませんでしたので、メンバーの一人であるクリジェフスキーは自宅のベッド下に置いたPCのGPUでこの計算を実験したそうです。ここでGPUはGraphic Processing Unitの略であり、もともとはその名のとおりゲームやアニメーションなどの映像処理に特化したチップでした。映像というのは各ピクセルのRGB値からなる数字の塊です。これを動かしたり回転させたり、といった処理をするにあたっては、各ピクセルのRGB値に対し「一括で」平行移動やら、回転やらといった計算（主として掛け算）を仕掛ける必要があります。この、同じような処理を一括でたくさんこなす、という計算を並列処理と呼び、GPUはまさにこの並列処理に特化しています。さて、ディープラーニングというのは極めて多くのパラメータ（重み）を仕込んだ予測モデルであり、これに予測対象となるデータを入力して（重みと掛け算して）答え（予測結果）を得ます。この、重みとデータを掛け算するという操作は先ほどお話した、同じような種類の計算を一括で同時に行うという種類のものであり、実はGPUにもってこいの計算処理だったのです。現代ではAI用の計算機資源と言えば、こうしたGPU系のものを指すことが多くなりました。 [↑](#footnote-ref-35)
36. 大規模言語モデルは新たな知能か ChatGPTが変えた世界（岡野原　大輔、岩波書店） [↑](#footnote-ref-36)
37. 食用ゼミの栽培をAIで自動化してみた（ホモサピ、YouTube） [↑](#footnote-ref-37)
38. NNPの発展についても複数のモデルが挙げられるのですが、本テキストでは特に化学反応に用いられやすい不均一触媒の開発に着目し、この面で比較的汎用性に富んだものと考えられるOCP（Open Catalyst Project）にフォーカスして説明します。 [↑](#footnote-ref-38)
39. [Materials Project - Home](https://next-gen.materialsproject.org/)　 [↑](#footnote-ref-39)
40. [[2010.09435] An Introduction to Electrocatalyst Design using Machine Learning for Renewable Energy Storage (arxiv.org)](https://arxiv.org/abs/2010.09435) [↑](#footnote-ref-40)
41. OC20データセットは金属のみ、OC22データセットでは金属酸化物を含む構造が対象となります。 [↑](#footnote-ref-41)
42. [Use S2EF model as a calcualtor in ASE to run the structure relaxation job · Issue #313 · Open-Catalyst-Project/ocp (github.com)](https://github.com/Open-Catalyst-Project/ocp/issues/313)　このイシューを見てわたしがどれほど喜んだか。 [↑](#footnote-ref-42)
43. [[2206.08917] The Open Catalyst 2022 (OC22) Dataset and Challenges for Oxide Electrocatalysts (arxiv.org)](https://arxiv.org/abs/2206.08917) [↑](#footnote-ref-43)
44. おそらく本テキストを手に取る方は読まなくてもよい点ですが、OCP力場にも当然弱点はございます。1つは、Materials Projectに存在するような構造から外れたデータに対しては、シミュレーションの精度が低下すること（これはOCPの論文にも記載されています）。また、不均一触媒の表面における各種分子の吸着構造、という分野を外した使い方をすると（実習いただくと分かる通り、非常に汎用的な計算環境なので、ついつい外した使い方をしたくなってしまうのです）同じく精度が低下することです。ただし、これらの欠点についてはおおむね次のように構えておくことで一応の備えができます。1つは、学習データの範囲外となるような構造を取り扱わない（で済むような作業仮説を人間が工夫して建てる）こと。もう1つは、場合によっては別のNNPに乗り換えることです。後者について、OCPの非常に優れた点は、そのユーザインターフェースが（のちのち取り扱いますが）計算化学の分野では比較的人口に膾炙した”ASE”である所です。このインターフェースを使っている限り、OCP力場が気に入らなければ別のcalculatorに変更する、という操作はスクリプトたった1行で出来てしまいます。私が本テキストの想定読者である初学者の方にOCP力場をお勧めする大きなポイントは、この乗り換え易さという所であるということを強調しておきます。 [↑](#footnote-ref-44)
45. コンピュータの導入コスト以上に、こうした前提情報が手に入れられるかどうかで、特に若い方々の学習環境には極めて大きな差が付きます（経験談）。筆者も大学に入るまで北国の片田舎で暮らしていたものですから、およそ教科書に載っている範囲外の出来事については、文化的にかなり自然側に寄った知識しか（例：氷点下10度の環境下、雪原に塹壕を掘る方法）有しておりませんでした。本テキストがやたらと参考文献を推しているように見えるのは、いわば前提知識としての教養に触れる機会、というのを可能な限りひろく用意したい、という個人的な思いもあります。 [↑](#footnote-ref-45)
46. これはmustではないのですが、特に高校生や高専生の方は近い将来大学に行くことも考えておいた方が、総合的な投資額を削減できる（つまり保護者の許諾を得やすくなる）、という目論見です。Microsoft Officeの有無が予算をかなり左右するという難点がありますが、今回の用途であればMicrosoft Officeもいりませんので、大学に入学した際に必要に応じてインストールする、という漸次的な投資のやり方もあります。なお同様の理由により、ひとまずOSはWindowsを前提としました（Linuxを強硬に推す向きもあろうと思いますが、当方の経験上「そういう人」はこのテキストを読まずとも自力で原子シミュレーションは実装できますので対象外とします）。 [↑](#footnote-ref-46)
47. 別にすべての社員が高度な作業をするわけでもないのだから、共用のPCはなるべくスペックを下げて必要な人だけいいものを別に予算申請しなさい（これがまたとてつもなく面倒でして、かつ担当するのは大抵若手になるので非常に心苦しい）、という思想が透けて見えたりします。がこれは完全なる雑談ですね。 [↑](#footnote-ref-47)
48. [freedeta (elshaddai.jp)](http://elshaddai.jp/elshaddai_crim/freedeta.html) [↑](#footnote-ref-48)
49. 計算化学に関するビッグデータを用いて、構造とエネルギーと力との関係をAIに学習させる（OCP力場を「作る」）には、AIが用いる予測モデルに含まれる膨大な数のパラメータを、膨大な数の試行錯誤によって最適化することが求められます。従って、この学習なるものをある程度腰を据えてやろう、という場合にはかなりの計算機資源を要します（しその多寡が力場のよしあしをかなりの程度で左右します）。一方で、我々が本テキストでやりたいのはOCP力場を「使う」ことですから、膨大な数のパラメータはすでに与えられたものとして使えばよく、したがって求められる計算機のスペックもそれなりで済みます。同じことは昨今話題の画像生成モデル（Stable Diffusion）を取り扱うときにも言えまして、ありものの画像生成モデルを使って画像を生成することが目的であれば（もちろんどの程度の規模で使うかによります…数十分のアニメーションを作りたいというのであれば膨大な枚数の画像生成を繰り返す必要がありますから）、ここで記した程度のコンピュータでも実は十分に事足りてしまいます。なお、GPUなんてGoogle cloudでイイじゃないかという向きもあるかと思いますが、クラウドでのGPU使用は制限時間が設けられている（かつStable Diffusionのような人気ツールは使用自体に制限がかかる）ため、好きなように研究を進めていただきたいという観点からここではローカルにGPUを導入するという流れで話を進めます。 [↑](#footnote-ref-49)
50. [価格.com - 「買ってよかった」をすべてのひとに。 (kakaku.com)](https://kakaku.com/)

    ※2023/8月時点の情報です。 [↑](#footnote-ref-50)
51. あまり“安かろう、悪かろう”に寄ってもよくないので、なるべく大きな店を紹介しました。ちなみに当方は中野ブロードウェイを散歩していた折、冷やかしで立ち寄ったジャンクショップに並んでいたGPU機を衝動買いしました。値段で言えばよくあるゲーミングノートの半額以下でしたが、GPUの性能と冷却効率はこちらの方が遥かに優れており、快適に作業をこなせるということに感動したことを覚えています。 [↑](#footnote-ref-51)
52. <https://www.pc-koubou.jp/> [↑](#footnote-ref-52)
53. とはいえ中古PCにも欠点はありまして、最大の懸念点は耐久性です。中古PCというのは前ユーザがどのような利用形態だったかが読めないので（というかPCパーツというのは基本的に、オシャカになるその時まで見た目にはそれがまったく現れないので）、安物買いの銭失いになりはしないかという不安は常に一定程度付きまといます。中古PCの販売店として大手を紹介したのもそのあたりに理由があり、せめて何か月かは保証してもらえるところを選ぶというリテラシーは必要だと考えています（し、その不安を受け入れるくらいなら少し高くても新品を買う、という選択は十分に合理的です）。どのような選択をするか、絶対的な正解はありませんので、個々人で少し悩んでみて下さい（特に若い方にとっては、その悩む時間すら良い経験だと思います）。なお、ノートかデスクトップかというのも非常に悩ましい所です…スペックで言えばデスクトップなのですが、ノートは電源さえあれば「どこでも」研究が出来るというとても魅力的な利点がありますから（だからと言って空港のラウンジでシミュレータを回すのは周囲に何と言いますか、その、「異様な」雰囲気を与えてしまうので、ご利用は自身の責任で、ということを個人的な経験として申し添えておきます）。 [↑](#footnote-ref-53)
54. <https://repo.anaconda.com/archive/>　 [↑](#footnote-ref-54)
55. <https://visualstudio.microsoft.com/ja/downloads/>　 [↑](#footnote-ref-55)
56. <https://git-scm.com/downloads>　 [↑](#footnote-ref-56)
57. <https://nanolabo-doc.readthedocs.io/ja/latest/usage/ocp.html>　 [↑](#footnote-ref-57)
58. <https://developer.nvidia.com/cuda-toolkit-archive>　 [↑](#footnote-ref-58)
59. <https://github.com/Open-Catalyst-Project/ocp/issues/426>　 [↑](#footnote-ref-59)
60. これは良し悪しなのですが、特に日本人の場合issuesに出てくると名前ですぐにばれてしまいます。かつ、実名で登録してしまうと素性も比較的簡単に割れてしまうため、場合によっては知られたくない情報まで推し量られてしまう、ということへのケアとなります（筆者の経験談として、issuesに掲載された人名をたどった結果、特定のアカデミアにおいて、publishはされていないものの現在どのような研究がなされているかについておよその所が推測できてしまったという事例もございます。こうした情報のリスクはとくに私企業において問題となりうるでしょう）。 [↑](#footnote-ref-60)
61. 背景情報が多くて誠に恐縮ですが、これまた個人的な経験談として「出口に何があるのかも示されずに1000本ノック打たされるような勉学」にはあまりいい思い出がないので、本テキストはその真逆の体裁をとることにしました。 [↑](#footnote-ref-61)
62. <http://cat.chem.chiba-u.jp/PDF/2010ChemEducJpn.pdf>　 [↑](#footnote-ref-62)
63. もちろんそうではないケースというのも世の中にはたくさんあります。律速段階は勿論のこと、素反応もよく分からない（つまり複数の活性化障壁をもつ）反応について、その反応パスをシミュレートするという研究も、非常に興味深い分野の一つと言えます。 [↑](#footnote-ref-63)
64. <https://youtu.be/5HuJt6xPNvs>　 [↑](#footnote-ref-64)
65. [OCPmodels\_seminar/20230114\_OCPmodels\_seminar\_CO\_adsorption\_energy.ipynb at main · M-asaki-K/OCPmodels\_seminar (github.com)](https://github.com/M-asaki-K/OCPmodels_seminar/blob/main/20230114_OCPmodels_seminar_CO_adsorption_energy.ipynb) [↑](#footnote-ref-65)
66. 一部私が色々と遊んでいた際に使ったライブラリまでインポートしている点、ご容赦ください（逐次更新予定）。 [↑](#footnote-ref-66)
67. [ocp/MODELS.md at main · Open-Catalyst-Project/ocp (github.com)](https://github.com/Open-Catalyst-Project/ocp/blob/main/MODELS.md) [↑](#footnote-ref-67)
68. ただしここで重要なのは、そのライブラリ自体に人気があるかどうかという点です。計算化学によらず、マイナーなライブラリを用いた場合、せっかく構成としては様々な拡張ができるものであったとしても肝心の開発者（つまり新しい機能をそのライブラリに対応する形で書いてくれる人）がいないために陳腐化する、というリスクがあります。ASEはこの意味合いにおいて、比較的人口に膾炙したライブラリの一つです。 [↑](#footnote-ref-68)
69. [Calculators — ASE documentation (dtu.dk)](https://wiki.fysik.dtu.dk/ase/ase/calculators/calculators.html) [↑](#footnote-ref-69)
70. この(hkl)という書き方をミラー指標といいます。どこをどう切るかで一つの結晶からは様々な面を生じます。この種類を一意に（誤解を生じずに）名づけるための規則が、ミラー指標です。「ってことは、どの面で反応するかも考慮すると計算事項が膨大になるのでは？」と思ったあなた、その通りですorz　とはいえ、たいていの反応については「どの面がキーである」という点も含めて様々な文献が出ていますので、まずはそれを参考に面を指定することをお勧めします。そういうのが全然ない、でもなるべく計算点数は少なくしたい、という場合には別の工夫（というか思い切り）が必要になります。よく用いられるのは「最も安定な（エネルギーの低い）表面を使う」という考え方ですね。 [↑](#footnote-ref-70)
71. このとき、バルク結晶に比べてどれくらい不安定になったかというのを表面エネルギーと言います（バルクと表面でエネルギーの差分をとるだけで得られます）。今回は取り扱いませんが、表面エネルギーは実際の実験系においては「こんな結晶表面は現実にありうるだろうか」とか「この表面をより安定化するには、どんな元素を添加した方がよいだろうか」とか、より物理化学的な側面が寄与するようなときにとても良いヒントを与えることが期待されます（当然、OCP力場でも算出することができます）。この「表面は結合の手が余っていて不安定」というのは、純粋な金属表面が直ちに酸化されてしまうことや、一方で表面が触媒として反応に寄与するといった現象の生みの親でもあるのです。もっと卑近な例としては水の表面張力なんかも、これに関連した現象の一つです。 [↑](#footnote-ref-71)
72. もちろん、研究を進める上ではある程度の忍耐力が求められると言う事自体は否定しません。ですが、一つの構造に対して基礎的な最適化をかけるだけでも一日仕事になるような計算速度では、たいていの初学者はそっと画面を閉じて別のことを始めてしまうでしょう（人生は有限ですから、その判断は間違いではないと思います）。OCPについてはそれが（適用範囲はさておき）DFTに近い精度を目指している、ということよりも、計算速度の破壊的な向上によって初学者に対する心理的ハードルを著しく（ええもうそりゃ本当に著しく）下げてしまうという点にこそ、真の破壊力があるというのが筆者の個人的な所感です。 [↑](#footnote-ref-72)
73. これらのチュートリアルは筆者がOCPに初めて触れた頃に作成したものなので、今見返すとかなり無駄が多いです（必要もないのに最適化の各ステップをtrajファイルに保存していたり、格子定数なんて一度求めたら使いまわせばいいのに毎度算出していたり…）。応用編のスクリプトは多少マシになっていますので、ご勘弁ください。 [↑](#footnote-ref-73)
74. 実はOC20力場と22力場で参照すべきエネルギーの絶対値が異なります。ただし演習用のスクリプトでは吸着エネルギーの相対値（あるいは差分）でしか議論しないことから、OC20力場における値を共通で用いました（ものぐさですみません。。）。 [↑](#footnote-ref-74)
75. 計算化学の歴史という章で、「汎関数を見つけ出す」という表現をしていたかと思いますが、これがその実例です。コーン氏の定理は汎関数の存在を証明したものの、それが何であるかについては言及しないため、科学者たちは何とかしてその正解を探し出そうとしています（ヤコビの梯子、で検索してみて下さい）。 [↑](#footnote-ref-75)
76. [PhysRevB.100.035442 (aps.org)](https://link.aps.org/accepted/10.1103/PhysRevB.100.035442) [↑](#footnote-ref-76)
77. 難しくない、と言えるようになったのは、ここ数年でPython周りの最適化アルゴリズムが爆発的に発展したことが奏功したと言っても過言ではないでしょう。実際、2019年ごろまでは深層学習における各種パラメータの最適化はある種の職人芸を伴うものでした（KaggleやSignateといった機械学習関連のcompetitionが流行した時期です）が、特にベイズ最適化やOptunaといったアルゴリズムが無償で利用できるようになったことで、これらの作業は著しい勢いで自動化されていきました。 [↑](#footnote-ref-77)
78. <https://docs.matlantis.com/atomistic-simulation-tutorial/ja/3_3_slab_adsorption_energy.html#%E5%90%B8%E7%9D%80%E6%A7%8B%E9%80%A0%E3%81%AE%E4%BD%9C%E6%88%90> [↑](#footnote-ref-78)
79. <https://www.preferred.jp/ja/projects/optuna/> このアルゴリズムがタダで使えるというのは、筆者にとっては衝撃的でした（冗談抜きで、飛行機のタダ券を無限回数分もらったような印象）。 [↑](#footnote-ref-79)
80. 本テキストは数理最適化に深く立ち入るものではありませんが、optunaはきわめて汎用的な最適化アルゴリズムですので、ここで一度その威力を体験しておくことは皆さんの人生にいつか役に立つと思います（例えばいつか先生にになってクラス分けを自動化したいな、と思った時とか…）。 [↑](#footnote-ref-80)
81. ある種当たり前のことですが、ここでのイメージというのは我々の思い込みであり、それが事実であるとは限りません。このように「多分こうなっているだろう」という人間の思い込みのことを、作業仮説と呼びます。研究は分野を問わず、この作業仮説があっているかどうか、という検証の繰り返しです。今回は実習ですので作業仮説があっているケースを取り扱いましたが、実務上好ましいのは作業仮説が否定されることかもしれません。しかしながら、好ましい作業仮説の否定をもたらすためには、できうる限り精密な作業仮説を構築せねばなりません。研究の悩ましい所です。 [↑](#footnote-ref-81)