



## Seriál: Kvanta molekul

V dnešním díle se konečně podíváme na to, jak se pomocí kvantové fyziky počítají molekuly.

### *Nic není tak jednoduché*

Jak jistě tušíte a také jak Vám jistě napověděl závěr druhého dílu seriálu, spočítat celou molekulu není jen tak jednoduché. V průběhu řešení musíme provést řadu zjednodušení. Začínáme už tím, jakou uvažujeme fyziku problému, tedy jaké fyzikální efekty uvažujeme. V první řadě většinou neuvažujeme efekty speciální teorie relativity, tedy například to, že pohybující se elektron je těžší. Druhá, zásadnější úvaha, je, že atomová jádra jsou mnohonásobně těžší než elektrony, a tím pádem se budou pohybovat na mnohem pomalejších časových škálách. Výsledkem toho je, že elektrony vidí jádra jako téměř nehybná, a tedy můžeme řešit jejich pohyb (vlnovou funkci) se zafixovanými jádry, a pohybem jader se případně zabývat až později. Ze stejného důvodu můžeme i zanedbat kvantovou povahu jader a považovat je za bodové náboje.

Ve výsledku tedy řešíme Schrödingerovu rovnici jen pro sadu elektronů v elektrostatickém poli zafixovaných jader. Samozřejmě nechceme zanedbávat elektrostatickou interakci mezi elektrony, takže hamiltonián bude obsahovat příslušné členy. Ale pokud si znovu vzpomenete na závěr druhého dílu, ani toto není vůbec jednoduché. Tam jsme si naznačili, že na přesné řešení Schrödingerovy rovnice můžeme v případě, kdy máme více než několik málo elektronů, rovnou zapomenout. A to si uvědomme, že často nás zajímají molekuly, které mají mnoho desítek atomů a pár set elektronů.

Řešení tohoto problému je snadné, stačí si vzpomenout na minulý díl. Zavedeme si báze funkce. V případě, že chceme počítat molekuly, máme k dispozici mnoho různých optimalizovaných sad. Většinou se jedná o funkce, které připomínají atomové orbitály, jak je známe, jen jsou definované tak, aby se s nimi lépe numericky počítalo. Při výpočtu pak použijeme pro každý atom jednu sadu funkcí s parametry specifickými pro konkrétní prvek, které jsou centrovány okolo polohy tohoto atomu. Většinou nám stačí pro každý atom jen několik desítek báze funkcí.

Mohlo by se zdát že máme vyhráno – Povedlo se nám zredukovat problém na několik set až pár tisíc báze funkcí, přestože předtím jsme měli více mřížových bodů, než je částic ve vesmíru! Opak je ale pravdou. Musíme si uvědomit, že nemáme jednočásticový problém. Tedy základní „konstrukční“ jednotkou nejsou samotné orbitály, ale stavy, kde máme v dané sadě orbitalů nějakým způsobem naskládané elektrony. Každý takovýto stav pak bude mít svůj vlastní neznámý koeficient. Kolik takových stavů je? To dokážeme jednoduše určit. Máme  $N$  báze funkcí, to znamená  $N$  různých orbitalů, do kterých můžeme obsazovat elektrony. V každém orbitalu může být elektron se spinem nahoru, dolů, oba, nebo ani jeden. To znamená, že máme vlastně  $2N$  chlívčků, do kterých umísťujeme po jednom elektrony. Počet možností pak je, pokud počet elektronů je  $k$ , dán kombinačním číslem  $\binom{2N}{k}$ . To je stále pro většinu systémů astronomické číslo, například pokud máme 40 elektronů a 100 orbitalů, tak počet možností je řádově  $10^{42}$ . A to odpovídá ještě docela malé molekule, například benzenu. Ale naštěstí existuje aproximace, která nás z této kaše umí dostat.

## Hartree-Fockova aproximace

Jak z toho ven? Problém vězí v tom, že se snažíme řešit problém pro všechny elektrony najednou. Musíme se pokusit problém rozdělit, tak, abychom měli nezávislou rovnici pro každý elektron zvlášť, ale zároveň jsme nezanedbali odpuzování mezi jednotlivými elektrony, jako jsme to dělali doteď. Prvním krokem je opustit myšlenku toho, že můžeme mít různé konfigurace ve vlnové funkci, tedy že do vlnové funkce přispívají různé způsoby, jak mohou být orbitály obsazeny. Místo toho využijeme to, že vlnovou funkci většiny molekul můžeme dobře popsat jedinou konfigurací, takovou, že postupně odspodu dvakrát obsazujeme orbitály s nejnižší energií. Celý problém se nám pak postupně smrskne jen na hledání správného tvaru těchto obsazených orbitalů. Jak ale tyto orbitály hledáme? Představme si, že by nám někdo prozradil, jak vypadají všechny obsazené orbitály kromě jednoho. Pak bychom mohli vzít vlnové funkce všech těchto orbitalů a vyrobit z nich hustotu pravděpodobnosti, kterou jsme si zavedli v druhém díle seriálu. Když ale tuto hustotu pravděpodobnosti vynásobíme nábojem elektronu, dostaneme nábojovou hustotu!

Tím pádem známe jak elektrostatický potenciál jader, tak rozložení náboje ostatních elektronů, nic nám nebrání vytvořit Schrödingerovu rovnici pro poslední orbital z potenciálem daným tímto rozložením náboje a tu snadno vyřešit, najednou máme jen několik set neznámých, což pro počítač je hračka. (Hledáme jen jeden neznámý koeficient pro každou básovou funkci.) Výsledkem pak je vlnová funkce posledního orbitalu a energie elektronu v tomto orbitalu.

A jak se vypořádáme s tím, že vlnové funkce jednotlivých obsazených orbitalů neznáme? Na začátku si je nějakým způsobem odhadneme. Tyto orbitály pak použijeme k sestavení hamiltoniánů pro jednotlivé elektrony. Vyřešíme odpovídající rovnice a získáme novou sadu orbitalů, která je přesnější než náš původní odhad. A pak celý postup opakujeme s novými orbitály, pokračujeme do té doby, než dosáhneme toho, že dostaneme stejné orbitály, jaké jsme použili k vyrobení hamiltoniánu, tedy orbitály zkonvergují.

Proč výsledek této aproximace není přesný? Háček spočívá v tom, že elektrostatický potenciál od ostatních elektronů v Schrödingerově rovnici pro orbital je nezávislý na poloze tohoto elektronu. V reálném světě si jistě dokážeme představit, že podle toho, kde se elektron nachází, bude v důsledku odpuzování ostatní orbitály různě deformovat. To bohužel ale neumíme popsat, pokud řešíme rovnici pro jediný elektron.

Celková chyba této aproximace je asi 1% ve výsledné energii. Bohužel pro popis chemických dějů většinou potřebujeme přesnost výrazně vyšší. Proto se velká část vývoje metod na kvantově chemické výpočty zaměřuje právě na to, jak obejít tento nedostatek, aniž by se neúměrně zvýšila výpočetní náročnost. Proč ale vlastně jedno procento je tak velká chyba?

Musíme si uvědomit, že celková energie molekuly, kterou vypočítáme, je energie, kterou bychom potřebovali k „rozebrání“ na jednotlivé elektrony a jádra a na jejich odtahení do nekonečna. Tato energie je ve srovnání s energií chemických reakcí ohromná. V chemických reakcích totiž dochází jen k přeuspořádání orbitalů ve valenční vrstvě. Když tedy počítáme energii reakce, musíme spočítat celkovou energii produktů a od ní odečteme celkovou energii reaktantů. Pokud máte zkušenosti s tím, jak se při matematických úpravách pracuje s odchylkami, víte, že odchylka reakční energie bude ohromná. Dokonce může vyjít i vyšší než energie samotné reakce. Proto se snažíme, aby se nám chyba u energie reaktantů a produktů co nejlépe vykompenzovala – Musíme použít pro oba výpočty stejnou metodu i bázi.

## A co s geometrií molekuly?

Jak jsme si již na začátku řekli, pozice jader jsou v celém výpočtu zafixované a musíme jejich souřadnice na začátku běhu programu zadat. Kde ale tyto souřadnice jader získáme? Sice je někdy můžeme získat z experimentu, pomocí rentgenové difrakce, ale ve většině případů je předem neznáme. Co s tím? Stačí nám vědět, jak chceme, aby molekula vypadala, a z toho určíme přibližnou geometrii. Pro tu spočítáme energii. Pak zkusíme jádru trochu pohnout nějakým směrem a sledujeme, jestli se nám energie sníží, nebo ne. Postupně takto umíme dojít až k takové geometrii, která má nejnižší energii. Výhodou je, že nemusíme tato posunutí jader hledat „naslepo“, ale přímo z vlnové funkce umíme spočítat gradient energie, tedy směr, ve kterém se nám energie snižuje nejrychleji.

Díky tomu umíme najít i struktury takových molekul a komplexů, které experimentálně pozorovat téměř nemůžeme, například krátce žijící meziproducty některých chemických reakcí.

## A jak to vypadá v praxi?

Nyní si ukážeme, jak v praxi vypadají výpočty. My budeme používat program Psi4, který je volně dostupný. (<https://psicode.org/>) Zkusme si celý postup ukázat na molekule vody. Musíme si připravit vstupní soubor, kde tomuto programu popíšeme, co po něm vlastně chceme.

V první řadě je potřeba popsat molekulu. Na vstupu zadáme molekulu vody, to se udělá pomocí

```
molecule{
0 1
O      -1.818556   -0.320567    1.225399
H      -0.898992   -0.062457    1.771117
H      -2.278145   -1.054597    1.903623
}
```

Primárně by nás měla zajímat první řádka `0 1`. První číslo udává náboj molekuly. Číslo 0 tedy značí, že máme neutrální molekulu. V případě kationtu bychom měli kladné číslo, pro aniont bychom měli číslo záporné. Druhé číslo značí spinovou multiplicitu. To je trochu složitější koncept, který označuje celkový počet nespárovaných elektronů, tedy o kolik je víc elektronů se spinem nahoru než dolů. Z historických důvodů se označuje číslem o jedna vyšším, než je nespárovaných elektronů. Tedy pokud máme všechny elektrony spárovány, jedná se o singletní stav a značíme jej 1. Stav se dvěma nespárovanými elektrony je triplet (3). Radikály mohou mít i jeden nespárovaný elektron, pak máme dublet. Každopádně drtivá většina molekul jsou v základním stavu singlet, takže daná řádka je málokdy jiná než `0 1`. Notorickou výjimkou je molekula kyslíku  $O_2$ , která je triplet. Dále následují souřadnice jader udané v angstromech.

Bázi udáme řádkem `set basis cc-pvdz`. Zkratka `cc-pvdz` označuje jednu z běžně používanýchází. Další možnosti se vzrůstající velikostí (tedy vzrůstající přesností i výpočetní náročností) jsou: `sto-3g`, `cc-pvdz`, `cc-pvtz`, `cc-pvqz`.

Ve chvíli, kdy máme toto zadefinované, už můžeme na závěr přidat řádek `optimize("HF")`, který spustí geometrickou optimalizaci. "HF" říká, že chceme optimalizovat právě Hartree-Fockovou metodou. Pokud bychom měli již správné souřadnice jader a chtěli bychom jen spočítat energii, vystačili bychom si s příkazem `energy("HF")`.

Když tedy dáme všechny tyto řádky dohromady, dostaneme celý vstup pro program:

```
set basis cc-pvdz
molecule {
0 1
  O      -1.818556   -0.320567   1.225399
  H      -0.898992   -0.062457   1.771117
  H      -2.278145   -1.054597   1.903623
}
optimize("HF")
```

kteřý můžeme uložit jako soubor pojmenovaný například `vstup.inp`. Výpočet můžeme spustit v příkazové řádce pomocí `psi4 vstup.inp`.

Po proběhnutí programu ve složce najdeme soubor `vstup.inp.dat`, případně `vstup.out` nebo podobný, který obsahuje veškeré informace z běhu programu. Pokud vše proběhlo v pořádku, na konci tohoto souboru najdeme výsledné optimalizované souřadnice jednotlivých atomů a výslednou energii celé molekuly. V tomto případě získáme na konci souboru:

```
...
==> Convergence Check <==
```

Measures of convergence in internal coordinates in au.

Criteria marked as inactive (o), active & met (\*), and active & unmet ( ).

Step	Total Energy	Delta E	Max Force	RMS Force	Max Disp	RMS Disp
Convergence Criteria		1.00e-06 *	3.00e-04 *	o	1.20e-03 *	o
7	-76.02703278	-6.09e-06	3.51e-05 *	2.65e-05 o	1.79e-04 *	1.05e-04 o

Next Geometry in Ang  
Fragment 1 (Ang)

```
O  0.0057046143  -0.0645290211  0.0000000000
H -0.7911519198  0.4458305979  0.0000000000
H  0.7006716423  0.5777071999  0.0000000000
```

Final optimized geometry and variables:

Molecular point group: cs

Full point group: Cs

Geometry (in Angstrom), charge = 0, multiplicity = 1:

```
O      0.005700842385  -0.064489709337  0.000000000000
H      -0.791211319490  0.445805409758  0.000000000000
H      0.700734813858  0.577693076293  0.000000000000
```

Psi4 stopped on: Wednesday, 04 January 2023 05:36PM

Psi4 wall time for execution: 0:00:01.40

\*\*\* Psi4 exiting successfully. Buy a developer a beer!

Z kolonky **Final energy** v tabulce, kde je uvedeno  $-76.02703278$ , zjistíme, že celková energie molekuly je  $-76,027$  Ha. Jednotky Ha v tomto případě značí hartree a jeden hartree odpovídá  $27,21$  eV, tedy asi  $2625,5 \text{ kJ}\cdot\text{mol}^{-1}$  nebo  $4,36 \cdot 10^{-18}$  J. Pokud bychom měli optimalizovanou geometrii a počítali jen energii, stačilo by nám hledat řádek **Total Energy** =  $-76.02703278$ . Těsně před koncem souboru je již výsledná optimalizovaná geometrie molekuly.

### *Pár poznámek na závěr*

Jak samotné hledání správných orbitalů, tak geometrická optimalizace jsou iterativní procedury. Tím pádem se může stát, a docela často se i stává, že iterace nekonvergují, tedy i kdybychom program nechali běžet nekonečně dlouho, nezískáme správný výsledek. Proto je program nastavený, aby vyhodil chybu a skončil, pokud nedosáhne konvergence po určitém množství iterací. V takovém případě je třeba jej zkusit spustit znovu s odlišnou vstupní geometrií, případně i jiným nastavením programu, ale to již žádá zkušeného kvantového chemika.

Zároveň Hartree-Fockova metoda v původní formulaci předpokládá molekulu, která má všechny orbitály molekuly buď obsazené dvěma elektrony, nebo prázdné. To automaticky znamená, že molekula je v singletním stavu. Pokud bychom chtěli počítat jiný spin (například triplet u zmíněného kyslíku), musíme tento požadavek uvolnit pomocí řádku **set reference rohf**.

Při přípravě geometrie komplikovanějších molekul se často hodí používat grafické programy, kde si molekulu můžeme „nakreslit“. Dobrý je v tomto ohledu například program Avogadro.

### *A jak nainstalovat a použít Psi4 na Windows?*

Instalační program na Windows si můžete stáhnout přímo z [http://vergil.chemistry.gatech.edu/psicode-download/Psi4conda-1.7-py38-Windows-x86\\_64.exe](http://vergil.chemistry.gatech.edu/psicode-download/Psi4conda-1.7-py38-Windows-x86_64.exe). Při instalaci můžete zakliknout volbu „Add Psi4conda to my PATH environment variable“, pak je možné ke spuštění možné použít přímo Windows příkazovou řádku. V každém případě vám po instalaci v menu Start přibude program „Anaconda prompt“, který můžete použít úplně stejně. Pokud otevřeme Anaconda prompt, můžeme napsat „psi4“, abychom si ověřili, že všechno funguje. Program by se měl spustit a vynadat nám, že nemá vstupní soubor.

Dalším krokem je tedy vzít vstupní soubor (můžete použít například ten ze seriálu), a dát jej programu jako parametr. Pouze je třeba dát si pozor na to, v jaké složce jsme v rámci příkazové řádky a kde je uložený vstupní soubor. Buď je třeba dát za příkaz celou cestu k souboru (např: **psi4 Dokumenty\psi4\vstup.inp**, pokud si na vstupy vyrobíme složku psi4 v Dokumentech) Druhou z možností je použít příkaz **cd**, který nás přesunuje mezi složkami. (Tedy bychom použili příkaz **cd Dokumenty**, pak **cd psi4**, a nakonec spustili program pomocí **psi4 vstup.inp**) Pokud bychom se chtěli přesunout o složku výše, je na to příkaz **cd ..**. A příkazem **dir** vypíšeme obsah aktuální složky. Výsledkem běhu programu je v tomto případě soubor **vstup.inp.out**, který si můžeme otevřít pomocí poznámkového bloku.

Tímto končí díl seriálu, a příště na něj navážeme tím, že si ukážeme, co se stane, když jádrům dovolíme se přeci jen hýbat.

---

Fyzikální korespondenční seminář je organizován studenty MFF UK. Je zastřešen Oddělením propagace a mediální komunikace MFF UK a podporován Ústavem teoretické fyziky MFF UK, jeho zaměstnanci a Jednotou českých matematiků a fyziků. Realizace projektu byla podpořena Ministerstvem školství, mládeže a tělovýchovy.

Toto dílo je šířeno pod licencí Creative Commons Attribution-Share Alike 3.0 Unported.  
Pro zobrazení kopie této licence navštivte <https://creativecommons.org/licenses/by-sa/3.0/>.