UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE FÍSICA

MATHEUS DE OLIVEIRA BISPO

APRENDIZADO DE MÁQUINA APLICADO A SUPERFÍCIES DE ENERGIA POTENCIAL PARA INTERAÇÕES ENTRE PERÓXIDO DE HIDROGÊNIO E GASES NOBRES

BRASÍLIA

12 DE SETEMBRO DE 2022

Matheus de Oliveira Bispo

Aprendizado de Máquina Aplicado a Superfícies de Energia Potencial Para Interações Entre Peróxido de Hidrogênio e Gases Nobres

Dissertação de Mestrado apresentado ao Instituto de Física da Universidade de Brasília como parte dos requisitos necessários à obtenção do título de Mestre em Física.

Orientador: Demétrio Antônio da Silva Filho

Universidade de Brasília – UnB
Instituto de Física

Brasília
12 de Setembro de 2022

Matheus de Oliveira Bispo
Aprendizado de Máquina Aplicado a Superfícies de Energia Potencial Para Interações Entre Peróxido de Hidrogênio e Gases Nobres/ Matheus de Oliveira Bispo. – Brasília, 12 de Setembro de 2022-
73 p. : il. (algumas color.) ; 30 cm.

Orientador: Demétrio Antônio da Silva Filho

Dissertação de Mestrado – Universidade de Brasília – UnB
Instituto de Física, 12 de Setembro de 2022.

1. Superfície de Energia Potencial. 2. Aprendizado de Máquina. 3. Métodos de Kernel. I. Demétrio Antônio da Silva Filho. II. Universidade de Brasília. III. Instituto de Física. IV. Aprendizado de Máquina Aplicado a Superfícies de Energia Potencial Para Interações Entre Peróxido de Hidrogênio e Gases Nobres

CDU 02:141:005.7

Matheus de Oliveira Bispo

Aprendizado de Máquina Aplicado a Superfícies de Energia Potencial Para Interações Entre Peróxido de Hidrogênio e Gases Nobres

Dissertação de Mestrado apresentado ao Instituto de Física da Universidade de Brasília como parte dos requisitos necessários à obtenção do título de Mestre em Física.

Trabalho aprovado. Brasília, 12 de Setembro de 2022:

Demétrio Antônio da Silva Filho
Orientador

Ricardo Gargano
Membro da Banca 1 - IF/UnB

Bernardo de Assunção Mello
Membro da Banca 2 - IF/UnB

Brasília
12 de Setembro de 2022

Aos meus pais, que nunca me deixaram desistir.
Amo vocês.

Agradecimentos

Aos meus pais, José Roberto e Márcia Marcela por me terem, providenciando um lar aconchegante e me formado como indivíduo. Aos meus padrinhos por terem servido de exemplo e influência intelectual e cultural. Às minhas avós que nunca permitiram que eu entrasse em estado de subnutrição em seus lares. Aos meus tios, tias, primos e primas por terem me zoado construtivamente todo esse tempo. Sem o apoio dessa família, não teria chegado onde cheguei.

Aos professores que tive ao longo de minha educação básica, especialmente ao Alex ("Tartaruga Ninja", meu segundo pai) pelas aulas, mnemônicos, macetes, conceitos e, acima de tudo, por ter me auxiliado a escolher meu curso de graduação durante uma chuvosa visita à Experimentoteca de Física da UnB.

Aos colegas de classe que me deram a oportunidade de ensinar e aprender, compartilharam alegrias e tristezas e me acompanharam no transporte ou no RU. Tentei citar todos, mas falhei miseravelmente. Vocês sabem quem são.

Por fim, ao meu orientador de mestrado, Demétrio Antônio da Silva Filho, pela grande oportunidade de realizar pesquisas científicas durante o mestrado, e aos membros da banca Bernardo Mello e Ricardo Gargano, que irão certamente providenciar *insights* importantes durante a apresentação e sabatina deste trabalho. Agradecimento extra ao professor Pedro Henrique e sua equipe de alunos por me acolher no grupo dele durante os meses (semi)presenciais de aulas.

"Over-thinking, over-analyzing
Separates the body from the mind
Withering my intuition
Leaving opportunities behind."
(TOOL)

Resumo

O aprendizado de máquina (ML) se tornou uma ferramenta computacional importante no estudo de sistemas físicos, proporcionando avanços importantes no estudo e na descrição de superfícies de energia potencial (SEPs). No entanto, pouco se investigou sobre o comportamento destes algoritmos em sistemas fracamente ligados, tais como moléculas presentes na atmosfera terrestre e no meio interestelar. O presente estudo visa analisar o desempenho dos métodos de kernel - um tipo de algoritmo de ML - na descrição das SEPs do sistema $H_2O_2 - Kr$ por meio de um estudo topográfico desta. Analisaremos também as curvas de aprendizado e as capacidades preditivas desse modelo no que tange à interpolação de pontos na superfície.

Palavras-chaves: aprendizado de máquina. métodos de kernel. superfície de energia potencial.

Abstract

Machine learning (ML) has become an important computational tool in the study of physical systems, providing important advancements in the study and description of potential energy surfaces (PESs). However, little has been investigated about the behavior of these algorithms in weakly bound systems, such as molecules present in the terrestrial atmosphere and in the interstellar medium. The present study aims to analyze the performance of kernel methods - a type of ML algorithm - in the description of the SEPs of the $H_2O_2 - Kr$ system through a topographic study of it. We will also analyze the learning curves and predictive capabilities of this model in terms of interpolating points on the surface.

Keywords: machine learning, kernel methods, potential energy surface.

Résumé

L'apprentissage automatique (ML, d'anglais *Machine Learning*) est devenu un outil de calcul important dans l'étude des systèmes physiques, fournissant des avancées importantes dans l'étude et la description des surfaces d'énergie potentielle (SEP). Cependant, il y a peu d'études ont été menées sur le comportement de ces algorithmes dans des systèmes faiblement liés, comme les molécules présentes dans l'atmosphère terrestre et dans le milieu interstellaire. La présente étude vise à analyser les performances des méthodes de kernel - un type d'algorithme ML - dans la description des SEP du système $H_2O_2 - Kr$ à travers une étude topographique de celui-ci. Nous analyserons également les courbes d'apprentissage et les capacités prédictives de ce modèle en termes d'interpolation de points sur la surface.

Mots-clés : apprentissage automatique. surfaces d'énergie potentielle. méthodes de kernel.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ABO	Aproximação de Born Oppenheimer
SEP	Superfície de Energia Potencial
CEP	Curva de Energia Potencial
HF	Hartree-Fock
SCF	Campo Auto-consistente (*Self-Consistent Field*)
DS	Determinante de Slater
LCAO	Combinação Linear de Orbitais Atômicos (*Linear Combination of Atomic Orbitals*)
SCF	Campo Auto-consistente (*Self-Consistent Field*)
MPn	Møller-Plesset de n-ésima ordem
ML	Aprendizado de Máquina (*Machine Learning*)
MLP	Potenciais de Aprendizado de Máquina (*Machine Learning Potentials*)
KRR	Regressão Regularizada de Kernel (*Kernel Ridge Regression*)
BSSE	Erro de Sobreposição do Conjunto de Base (*Basis Set Superposition Error*)
MSE	Erro médio quadrático (*Mean Square Error*)
RMSE	Erro raiz-média-quadrática (*Root-Mean-Square Error*)
LC	Curva de Aprendizado (*Learning Curve*)
IF	Instituto de Física
UnB	Universidade de Brasilia

Lista de símbolos

ω	Parâmetro de longo alcance
V	Superfície de Energia Potencial
$\hat{H}$	Operador hamiltoniano do sistema molecular
$\hat{H}_i$	Operador hamiltoniano de um elétron
$\hat{F}$	Operador de Fock
Ψ	(auto)função de onda total
Φ	(auto)função de onda eletrônica
ϕ	(auto)função de onda de um elétron
χ	(auto)função de onda eletrônica
ρ	Densidade eletrônica
E_{xc}	Funcional de Troca-correlação
Δ	Discrepância entre as superfícies

Sumário

Introdução

Um dos maiores avanços na tecnologia moderna foi impulsionado pelos desenvolvimentos realizados na área de Aprendizado de Máquina (ML, do inglês *Machine Learning*) (1). Apesar de seus conceitos e fundamentos estarem presentes desde a década de 1950 (2), foi somente nas últimas décadas que essa área avançou significativamente.

Esse avanço se deve, em grande parte, pelo aumento das capacidades de processamento das CPUs e GPUs em anos recentes e pelos grandes volumes de dados disponíveis graças à evolução da era digital (3). Como consequência, grande parte dos algoritmos de ML desenvolvidos já fazem parte do cotidiano.

Essas técnicas obtiveram sucesso ao resolverem problemas nos quais instruções programadas eram insuficientes para compor uma solução, tais como: identificar números e letras escritos à mão (4), sugerir novos produtos a usuários de uma loja com base em outros produtos previamente visitados (5, 6) e desenvolver assistentes virtuais que analisam, interpretam e respondem às mensagens de um(a) usuário(a) (7).

Para isso, os algoritmos de ML são treinados e ajustados a um conjunto representativo e relativamente grande de dados - muitas vezes não estruturados - de acordo com a arquitetura de cada modelo. Exemplos desses algoritmos são as redes neurais (8) e as máquinas de vetores de suporte (9), entre outros.

Diversos problemas físicos têm sido solucionados utilizando ML. Com efeito, o emprego dessas técnicas na Física tem crescido aceleradamente, com resultados cada vez mais surpreendentes em áreas tais como dinâmica de fluidos (10), física médica (11), sistemas dinâmicos (12) e física atômica e molecular (13).

Esforços recentes na literatura conseguiram expandir as aplicações dos métodos de ML em diversas áreas da física atômica e molecular, seja na evolução temporal de uma simulação de dinâmica molecular (MD, do inglês *Molecular Dynamics*) (14), seja na construção de superfícies de energia potencial (SEPs) (15). Tais aplicações tem produzido resultados e publicações cada vez mais acelerados (16), o que indica a importância de se estudar essa área em nosso instituto enquanto ela ainda está sendo desenvolvida.

Objetivos

O objetivo principal desta dissertação é analisar se algoritmos de aprendizado de máquina (em especial, métodos de kernel) são de fato alternativas viáveis aos métodos *ab initio* na construção de diferentes SEPs.

Parte I

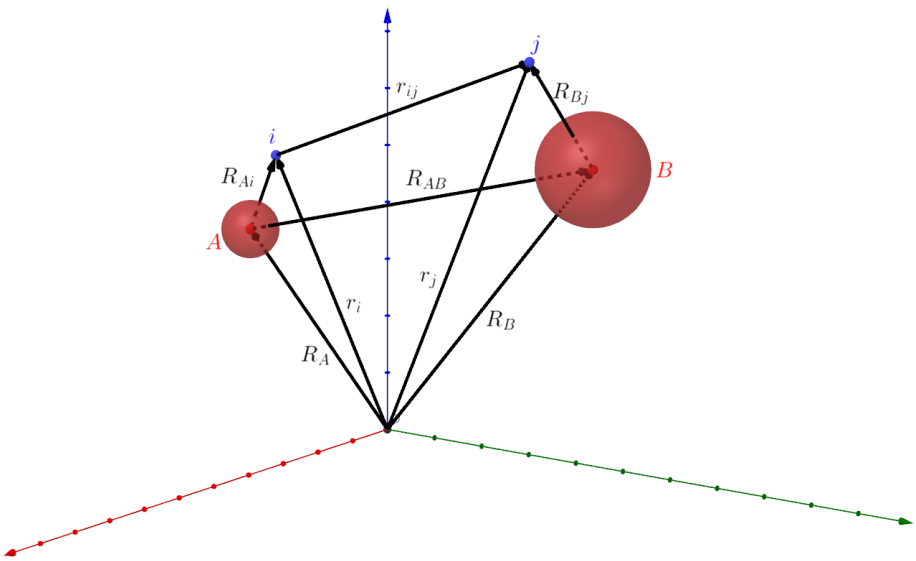
Fundamentação Físico-Química

1 O Sistema Molecular

1.1 Aproximação de Born-Oppenheimer

Considere um sistema molecular genérico composto de N núcleos (indexados por letras maiúsculas), com números atômicos $\{Z_A\}$ e massas $\{M_A\}$, e M elétrons (indexados por letras minúsculas). Em princípio, para descrever completamente o sistema, é necessário levar em consideração não somente os termos cinéticos atrelados aos movimentos dos núcleos e dos elétrons, mas também as interações entre elétrons, entre os núcleos e entre elétrons e núcleos.

Figura 1 – Representação gráfica do sistema molecular, com A e B representando os núcleos e i e j representando os elétrons.



Desprezando efeitos relativísticos, podemos escrever o operador hamiltoniano do sistema (em unidades atômicas) como:

$$\hat{H} = -\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2 - \frac{1}{2}\sum_{i=1}^{M}\nabla_i^2 - \sum_{i=1}^{M}\sum_{A=1}^{N}\frac{Z_A}{R_{Ai}} + \sum_{A=1}^{N}\sum_{B>A}^{N}\frac{Z_A Z_B}{R_{AB}} + \sum_{i=1}^{M}\sum_{j>i}^{M}\frac{1}{r_{ij}}. \tag{1.1}$$

A equação de Schrödinger associada ao hamiltoniano em (1.1) só pode ser resolvida se levarmos em conta que o movimento dos elétrons é muito maior que o movimento dos núcleos, dada a diferença de massa de aproximadamente 1836 vezes entre eles. Com isso, podemos realizar a **expansão adiabática** da autofunção $\Psi(\mathbf{r}, \mathbf{R})$ do operador $\hat{H}$ como o produto das autofunções dos elétrons e dos núcleos:

$$\Psi(\mathbf{r},\mathbf{R}) = \Phi(\mathbf{r};\mathbf{R})\chi(\mathbf{R}). \tag{1.2}$$

As funções $\Phi(\mathbf{r};\mathbf{R})$ e $\chi(\mathbf{R})$ correspondem respectivamente às funções de onda eletrônica (que depende da posição dos elétrons) e nuclear (que depende da posição dos núcleos) (17). Note que a função de onda eletrônica depende parametricamente da posição dos núcleos.

Substituindo (1.1) e (1.2) na equação de Schrödinger $\hat{H}\Psi(\mathbf{r},\mathbf{R}) = E\Psi(\mathbf{r},\mathbf{R})$, temos

$$\left[-\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2 - \frac{1}{2}\sum_{i=1}^{M}\nabla_i^2 - \sum_{i=1}^{M}\sum_{A=1}^{N}\frac{Z_A}{R_{Ai}} + \sum_{A=1}^{N}\sum_{B>A}^{N}\frac{Z_AZ_B}{R_{AB}} + \sum_{i=1}^{M}\sum_{j>i}^{M}\frac{1}{r_{ij}}\right]\Phi(\mathbf{r};\mathbf{R})\chi(\mathbf{R}) = E\Phi(\mathbf{r};\mathbf{R})\chi(\mathbf{R}). \tag{1.3}$$

Podemos escrever o termo cinético nuclear como um operador separado:

$$\hat{T}_n = -\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2 \tag{1.4}$$

$$\begin{aligned}\implies \hat{T}_n\Phi(\mathbf{r};\mathbf{R})\chi(\mathbf{R}) &= -\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2\Phi(\mathbf{r};\mathbf{R})\chi(\mathbf{R}) \\ &= -\sum_{A=1}^{N}\frac{1}{2M_A}\left[\nabla_A^2\chi(\mathbf{R}) + \nabla_A^2\Phi(\mathbf{r};\mathbf{R}) + 2\nabla_A\Phi(\mathbf{r};\mathbf{R})\cdot\nabla_A\chi(\mathbf{R})\right].\end{aligned} \tag{1.5}$$

Levando em conta os princípios da expansão adiabática, podemos desprezar os dois últimos termos da soma da equação (1.5), uma vez que a dependência de Φ com relação às coordenadas nucleares é paramétrica. Essa aproximação é conhecida como **Aproximação de Born-Oppenheimer** (ABO) (18). Por causa disso, a equação (1.3) pode ser desacoplada em duas: uma equação eletrônica

$$-\frac{1}{2}\sum_{i=1}^{M}\nabla_i^2\Phi(\mathbf{r};\mathbf{R}) - \sum_{i=1}^{M}\sum_{A=1}^{N}\frac{Z_A}{R_{Ai}}\Phi(\mathbf{r};\mathbf{R}) + \sum_{i=1}^{M}\sum_{j>i}^{M}\frac{1}{r_{ij}}\Phi(\mathbf{r};\mathbf{R}) = \varepsilon(R)\Phi(\mathbf{r};\mathbf{R}), \tag{1.6}$$

e outra nuclear

$$-\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2\chi(\mathbf{R}) + \sum_{A=1}^{N}\sum_{B>A}^{N}\frac{Z_AZ_B}{R_{AB}}\chi(\mathbf{R}) = -\varepsilon(R)\chi(\mathbf{R}) + E\chi(\mathbf{R}). \tag{1.7}$$

1.1.1 Superfície de Energia Potencial

Por outro lado, podemos escrever a equação nuclear como:

$$-\sum_{A=1}^{N}\frac{1}{2M_A}\nabla_A^2\chi(\mathbf{R})+V(R)\chi(\mathbf{R})=E\chi(\mathbf{R}), \tag{1.8}$$

onde $V(R)$ é um potencial efetivo que representa a Superfície de Energia Potencial (SEP) do sistema molecular, que é um dos focos desta dissertação:

$$V(R)=\sum_{A=1}^{N}\sum_{B>A}^{N}\frac{Z_AZ_B}{R_{AB}}+\varepsilon(R). \tag{1.9}$$

O conceito de SEP é central tanto para a Física Atômica e Molecular como para a Química Computacional. Ela governará a dinâmica dos núcleos de cada átomo da molécula, ditando como esta realizará os movimentos de vibração, rotação e translação. Com base nestes movimentos, pode-se ainda deduzir propriedades espectroscópicas e termodinâmicas desses sistemas.

A SEP mostra como a energia potencial varia de acordo com os graus de liberdade geométricos possíveis presentes no sistema. Por exemplo, se o sistema for uma molécula diatômica com apenas um grau de liberdade (o comprimento da ligação), temos uma Curva de Energia Potencial (CEP), que nada mais é que um caso unidimensional particular de SEP.

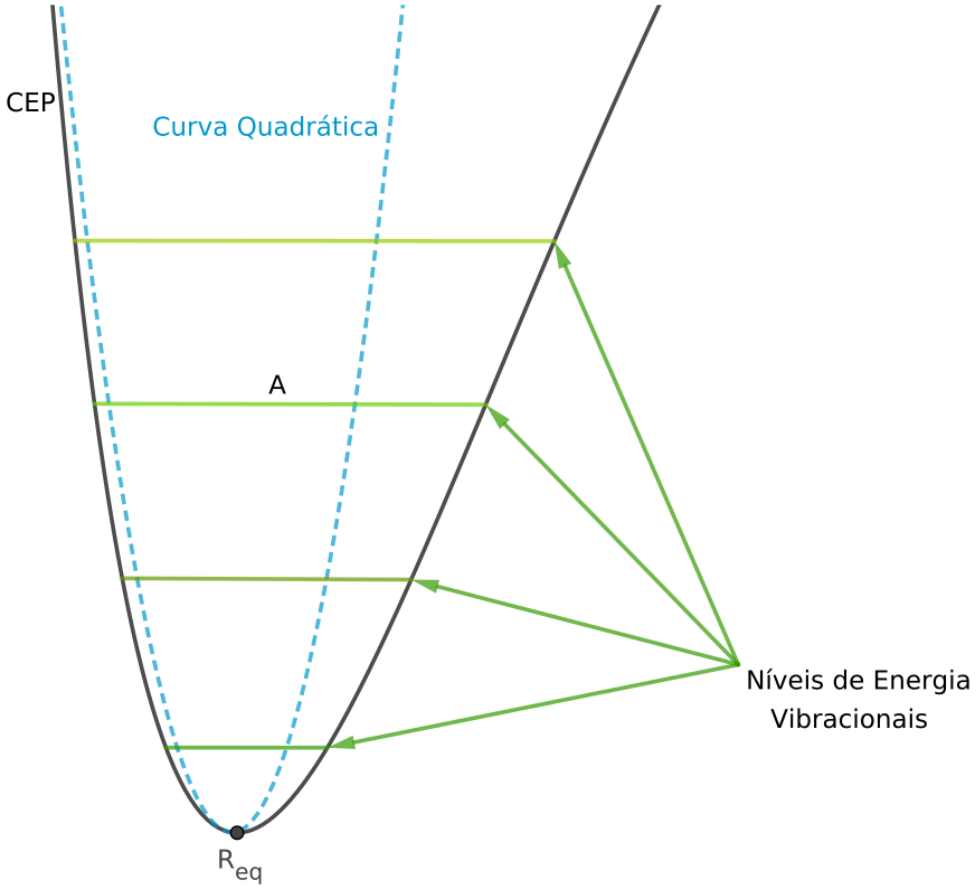
Dentre os pontos mais importantes das SEPs, destacam-se os pontos estacionários, obtidos quando a derivada parcial da energia com relação a cada parâmetro geométrico é nula:

$$\left.\frac{\partial V}{\partial\mathbf{R}}\right|_{\mathbf{R}_0}=0$$

Caso um destes pontos corresponda a um mínimo da energia potencial, então esse ponto representa uma configuração molecular estável ou metaestável, a depender da profundidade do mínimo e da temperatura. É em torno destes pontos de mínimo que podemos fazer uma aproximação harmônica para descrever os modos normais de vibração dos núcleos atômicos e suas frequências correspondentes com relação à posição de equilíbrio (19), conforme ilustra a Figura 2.

Por outro lado, caso este seja um ponto de sela, teremos uma estrutura de transição entre duas configurações estáveis. Em geral, reações químicas ocorrem por um caminho que liga um mínimo a outro passando pelo ponto de sela, de forma que este seja um ponto de máximo ao longo deste caminho.

Figura 2 – Exemplo de aproximação quadrática em torno do ponto de equilíbrio da CEP e os respectivos níveis de energia vibracionais.



Para determinar apropriadamente uma SEP é necessário resolver a equação de Schrödinger eletrônica (1.6) para várias conformações nucleares diferentes. Para uma SEP que descreve um processo colisional reativo (por exemplo, uma reação química), essas conformações vão desde a região dos reagentes até a região dos produtos, passando pela estrutura de transição (ou complexo ativado). Já para um sistema ligado, as conformações vão desde a região de forte interação até o regime assintótico, passando pela região de equilíbrio.

Contudo, mesmo com a aproximação de Born-Oppenheimer, a solução acurada da equação de Schrödinger eletrônica não é uma tarefa fácil e sua solução analítica até mesmo para sistemas moleculares simples é praticamente impossível. Para contornar este problema, vários métodos numéricos e computacionais de estrutura eletrônica foram desenvolvidos. A seguir, serão apresentados alguns dessas metodologias.

1.2 Método de Hartree-Fock

Um dos métodos de resolver sistemas quânticos de vários elétrons foi desenvolvido em 1927 por Douglas Hartree - e posteriormente corrigida por Vladmir Fock e John Slater em 1928 - partindo de princípios fundamentais da mecânica quântica (método *ab initio*). Este método foi denominado **Método de Hartree-Fock** (HF) (17).

Como ponto de partida, considere a função de onda eletrônica como o produto de Hartree, isto é, o produto de todas as funções de onda de 1-elétron:

$$\Phi(\mathbf{r}_1, ..., \mathbf{r}_M) = \prod_{i=1}^{M} \phi_i(\mathbf{r}_i). \tag{1.10}$$

Elétrons fazem parte de uma classe de partículas denominados férmions, os quais, por antissimetria, não podem ocupar todos o mesmo estado, seguindo o chamado Princípio de Exclusão de Pauli (17). Por exemplo, num sistema de dois elétrons, o produto de Hartree deve ser modificado da seguinte forma:

$$\Phi(\mathbf{r}_1, \mathbf{r}_2) = \phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2) - \phi_1(\mathbf{r}_2)\phi_2(\mathbf{r}_1). \tag{1.11}$$

Além disso, temos que levar em conta um grau de liberdade extra de cada elétron denominado *spin*. Cada elétron pode ter *spin up* (denotado por $\alpha(\omega)$) ou *spin down* (denotado por $\beta(\omega)$). Com os orbitais de spin $\tau_1(\mathbf{x}_i) = \alpha(\omega_i)\phi_1(\mathbf{r}_i)$ e $\tau_2(\mathbf{x}_i) = \beta(\omega_i)\phi_2(\mathbf{r}_i)$, onde $\mathbf{x}_i \equiv (\omega_i, \mathbf{r}_i)$, podemos reescrever a equação (1.11) como

$$\Phi(\mathbf{x}_1, \mathbf{x}_2) = \tau_1(\mathbf{x}_1)\tau_2(\mathbf{x}_2) - \tau_1(\mathbf{x}_2)\tau_2(\mathbf{x}_1). \tag{1.12}$$

No caso mais geral, para M elétrons, podemos escrever essa combinação como um determinante de Slater (DS):

$$\Phi(\mathbf{x}_1, ..., \mathbf{x}_M) = \frac{1}{\sqrt{M!}} \begin{vmatrix} \tau_1(\mathbf{x}_1) & \tau_2(\mathbf{x}_1) & \cdots & \tau_M(\mathbf{x}_1) \\ \tau_1(\mathbf{x}_2) & \tau_2(\mathbf{x}_2) & \cdots & \tau_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \tau_1(\mathbf{x}_M) & \tau_2(\mathbf{x}_M) & \cdots & \tau_M(\mathbf{x}_M) \end{vmatrix}, \tag{1.13}$$

onde $1/\sqrt{M!}$ é um fator de normalização.

Minimizando, com relação aos orbitais de spin τ_i, o funcional obtido pelo método variacional a partir das equações (1.6) e (1.13), chega-se à equação de Hartree-Fock na forma canônica e integrada na parte de spin:

$$\left[\hat{h}(i) + \sum_{k=1}^{M/2}(2\hat{J}_{ik} - \hat{\mathcal{P}}_{ik})\right]\phi_i(\mathbf{r}_i) \equiv \hat{f}(i)\phi_i(\mathbf{r}_i) = \varepsilon_i\phi_i(\mathbf{r}_i), \tag{1.14}$$

onde $\hat{h}_i$ é o hamiltoniano de 1-elétron do sistema, $\hat{J}_{ik}$ é o operador de Coulomb, que descreve a repulsão elétrica entre dois elétrons, $\hat{\mathcal{P}}_{ik}$ é o operador de permutação. A combinação destes constitui o operador de Fock $\hat{f}(i)$, e a equação de autovalor e autovetor associada a ele é dita **Equação de Hartree-Fock** (HF).

A equação HF deve ser resolvida para cada elétron do sistema. Podemos resolvê-la para todos os elétrons do sistema fazendo uso de uma expansão dos orbitais espaciais $\phi_i(r_i)$ em funções de base conhecidas:

$$\phi_i(\mathbf{r}_i) = \sum_{\mu=1}^{K} C_{\mu i}\psi_\mu(\mathbf{r}_i) \tag{1.15}$$

Aplicando (1.15) em (1.14), multiplicando à direita por $\psi^*_\nu(\mathbf{r}_i)$ e integrando nas coordenadas espaciais, obtemos a **equação de Hartree-Fock-Roothan**, que é a forma matricial da (1.14):

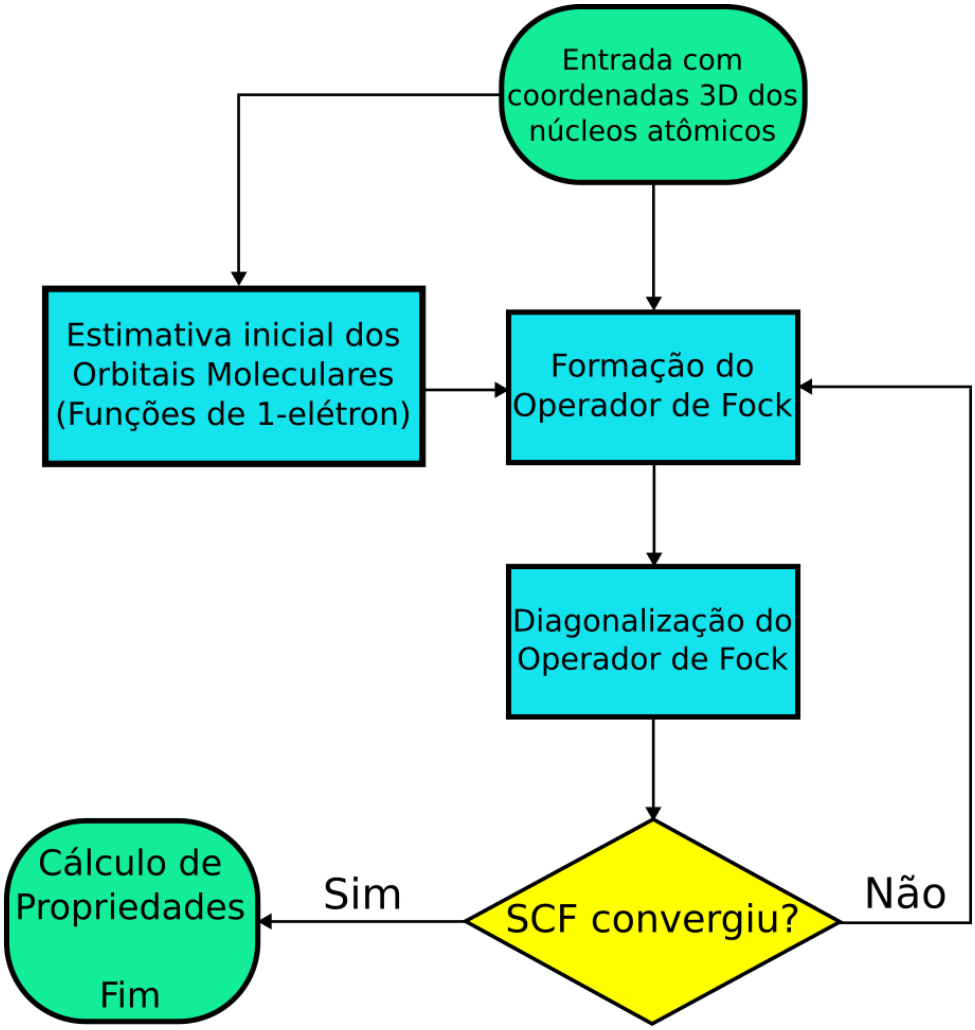
$$\begin{aligned} &\sum_{\mu=1}^{K} F_{\nu\mu}C_{\mu i} = \varepsilon_i \sum_{\mu=1}^{K} S_{\nu\mu}C_{\mu i} \\ &\implies \mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC}, \end{aligned} \tag{1.16}$$

onde $\mathbf{F} \doteq F_{\nu\mu}$ é a matriz de Fock, $\mathbf{S} \doteq S_{\nu\mu}$ é a matriz de sobreposição (*overlap*) e $\boldsymbol{\varepsilon} \doteq \varepsilon_i\delta_{\nu i}$ é a matriz diagonal com as energias de ionização ε_i do orbital i.

Note que, como $\mathbf{F}$ depende dos coeficiente de expansão dos orbitais usados, é necessário utilizar um método iterativo conhecido como método de campo autoconsistente (SCF, do inglês *self-consistent field*) para obter uma solução para a equação (1.16). A Figura 3 ilustra o algoritmo do método SCF.

A priori, sabendo das posições dos núcleos no sistema, é possível fornecer um *ansatz* sobre os orbitais moleculares na forma de combinações lineares de orbitais já conhecidos de átomos de 1-elétron. Tais orbitais atômicos formam um conjunto de base. Todavia, como o operador de Fock presente na equação (1.14) faz uso de um campo médio na descrição da interação eletrônica, estes orbitais moleculares são influenciados pela média da presença de outros elétrons ao invés de cada um separadamente (17).

Figura 3 – Fluxograma algorítmico que ilustra o método de campo autoconsistente.



Fonte: Wikimedia Commons (20) (Adaptado).

1.2.1 Conjuntos de Base

Observa-se a importância da escolha de um conjunto de base na determinação de uma solução para o sistema molecular. A primeira e mais intuitiva abordagem foi a já evidenciada combinação linear de orbitais atômicos (LCAO, do inglês *Linear Combination of Atomic Orbitals*). Temos, então, orbitais do tipo Slater (STO, do inglês *Slater-type Orbital*):

$$\psi_{STO}(\mathbf{r}) = A\mathbf{r}^{n-1}e^{-\zeta|\mathbf{r}|} \tag{1.17}$$

Onde A é uma constante de normalização, n é o número quântico principal e o parâmetro ζ está atrelada à carga efetiva do núcleo e leva em consideração o efeito de blindagem eletrônica. Note que tratamos o elétron de valência como pertencente a uma só camada.

Além disso, podemos ter uma base composta de n gaussianas contraídas cujos parâmetros são ajustadas de tal forma que a combinação linear delas aproxime um orbital do tipo Slater, daí a nomenclatura *STO-nG* dessas bases. No caso $n = 3$, temos:

$$\psi_{STO-3G}(r) = c_0\phi_0(r) + c_1[\phi_1(r) + \phi_2(r) + \phi_3(r)], \tag{1.18}$$

$$\text{com } \phi_i(r) = \left(\frac{2\alpha_i}{\pi}\right) e^{-\alpha_i r^2}. \tag{1.19}$$

Existem também os conjuntos de base que permitem o tratamento de valência dupla, isto é, permitem tratar cada camada de valência com um número diferente de funções. Bases desse tipo seguem a nomenclatura *X-YZg*, onde X é o número de gaussianas das camadas mais internas, e Y e Z representam os números de gaussianas das camadas mais externas. Esse tipo de base, denominadas bases de Pople, são as mais utilizadas na atualidade.

Contudo, quando se leva em conta métodos pós-HF, outros tipos de bases são preferíveis, uma vez que estes métodos tratam melhor da correlação eletrônica e efeitos de dispersão. Essas bases mais adequadas são as bases de Dunning, e seguem a nomenclatura *cc-pVXZ*, onde:

cc significa correlação consistente,

p indica polarização,

V significa valência,

X identifica o número de camadas de valência (D para duas camadas de valência, T para três, Q para quatro, 5 para cinco e assim por diante), e

Z significa zeta.

Se a base contiver o prefixo *aug-*, isso significa que a base foi aumentada com funções de dispersão e polarização adicionais. Por exemplo, a base aug-cc-pVTZ é dita base aumentada cc-polarizada com valência tripartida zeta. É evidente que, quanto maior a partição das camadas de valência, mais funções são necessárias para descrever os orbitais e, por conseguinte, mais preciso o cálculo.

1.3 Teoria de Perturbação de Møller-Plesset

Apesar de sua ampla utilização, o método HF possui algumas limitações associadas ao tratamento da correlação eletrônica no sistema molecular, as quais levaram a desvios

grandes de resultados experimentais. Para sanar esta deficiência, diversos métodos pós-HF foram criados.

Um destes métodos foi concebido em 1934 por Christian Møller e Milton Plesset (21). Eles utilizaram métodos perturbativos para corrigir os problemas associados ao método HF. Na atualmente denominada **Teoria de Perturbação de Møller-Plesset** (MP), o hamiltoniano do sistema é acrescido de uma pertubação externa:

$$\hat{H} = \hat{H}_0 + \lambda \hat{V}, \tag{1.20}$$

onde λ é um parâmetro perturbativo, $\hat{H}_0 = \sum_{i=1}^{n} h(i)$ é a soma dos hamiltonianos de 1-elétron e a perturbação é o potencial de correlação, dado por

$$\hat{V} = \sum_{i=1}^{M} \sum_{j>i}^{M} \frac{1}{r_{ij}} - \sum_{i=1}^{M} v^{HF}(i). \tag{1.21}$$

Assume-se que λ seja pequeno o suficiente para realizar uma expansão em série de suas potências dos autovalores e autovetores de energia do estado fundamental:

$$\begin{cases} E_0 = E_0^{(0)} + \lambda E_0^{(1)} + \lambda^2 E_0^{(2)} + \lambda^3 E_0^{(3)} + \lambda^4 E_0^{(4)} + \cdots \\ |\Phi_0\rangle = |\Phi_0^{(0)}\rangle + \lambda |\Phi_0^{(1)}\rangle + \lambda^2 |\Phi_0^{(2)}\rangle + \lambda^3 |\Phi_0^{(3)}\rangle + \lambda^4 |\Phi_0^{(4)}\rangle + \cdots \end{cases} \tag{1.22}$$

Utilizando a expansão (1.22) na equação de autovalor do operador (1.20), obtemos

$$\begin{aligned} &[\hat{H}_0 + \lambda \hat{V}][|\Phi_0^{(0)}\rangle + \lambda |\Phi_0^{(1)}\rangle + \lambda^2 |\Phi_0^{(2)}\rangle + \cdots] \\ &\qquad = [E_0^{(0)} + \lambda E_0^{(1)} + \lambda^2 E_0^{(2)} + \cdots][|\Phi_0^{(0)}\rangle + \lambda |\Phi_0^{(1)}\rangle + \lambda^2 |\Phi_0^{(2)}\rangle + \cdots] \end{aligned} \tag{1.23}$$

Agrupando os termos de mesma potência em λ e rearranjando-os, temos:

$$\lambda^0\text{: } (\hat{H}_0 - E_0^{(0)})|\Phi_0^{(0)}\rangle = 0$$

$$\lambda^1\text{: } (\hat{H}_0 - E_0^{(0)})|\Phi_0^{(1)}\rangle = (E_0^{(1)} - \hat{V})|\Phi_0^{(0)}\rangle$$

$$\lambda^2\text{: } (\hat{H}_0 - E_0^{(0)})|\Phi_0^{(2)}\rangle = (E_0^{(1)} - \hat{V})|\Phi_0^{(1)}\rangle + E_0^{(2)}|\Phi_0^{(0)}\rangle$$

$$\lambda^3\text{: } (\hat{H}_0 - E_0^{(0)})|\Phi_0^{(3)}\rangle = (E_0^{(1)} - \hat{V})|\Phi_0^{(2)}\rangle + E_0^{(2)}|\Phi_0^{(1)}\rangle + E_0^{(3)}|\Phi_0^{(0)}\rangle$$

$$\lambda^4\text{: } (\hat{H}_0 - E_0^{(0)})|\Phi_0^{(4)}\rangle = (E_0^{(1)} - \hat{V})|\Phi_0^{(3)}\rangle + E_0^{(2)}|\Phi_0^{(2)}\rangle + E_0^{(3)}|\Phi_0^{(1)}\rangle + E_0^{(4)}|\Phi_0^{(0)}\rangle$$

$$\vdots$$

E assim por diante. Pode-se escolher um ponto de parada arbitrário às potências de λ, assim definindo o grau de precisão do método perturbativo. Por exemplo, se deseja-se obter uma correção em segunda ordem, temos o MP2; em terceira ordem, o MP3; em quarta ordem (o qual será usado no trabalho), o MP4; etc.

Agora, sabendo que os $|\Phi_0^{(i)}\rangle$, $i > 0$, podem ser escritos como combinações lineares dos estados $|\Phi_n^{(0)}\rangle$ já conhecidos, podemos encontrar as correções do autovalor de energia (22). Note que, neste caso, as correções por correlação eletrônica só aparecem a partir da 2ª ordem.

$$\begin{aligned}
\text{MP2} : E_0^{(2)} &= \sum_{j\neq 0} \frac{V_{0j}V_{j0}}{E_{0j}} \\
\text{MP3} : E_0^{(3)} &= \sum_{j\neq 0}\sum_{k\neq 0} \frac{V_{0j}\tilde{V}_{jk}V_{k0}}{E_{0j}E_{0k}} \\
\text{MP4} : E_0^{(4)} &= \sum_{j\neq 0}\sum_{k\neq 0}\sum_{l\neq 0} \left[\frac{V_{0j}V_{jk}V_{kl}V_{l0}}{E_{0j}E_{0k}E_{0l}} - E_0^{(2)}\frac{|V_{0l}|^2}{E_{0l}^2}\right]
\end{aligned} \tag{1.24}$$

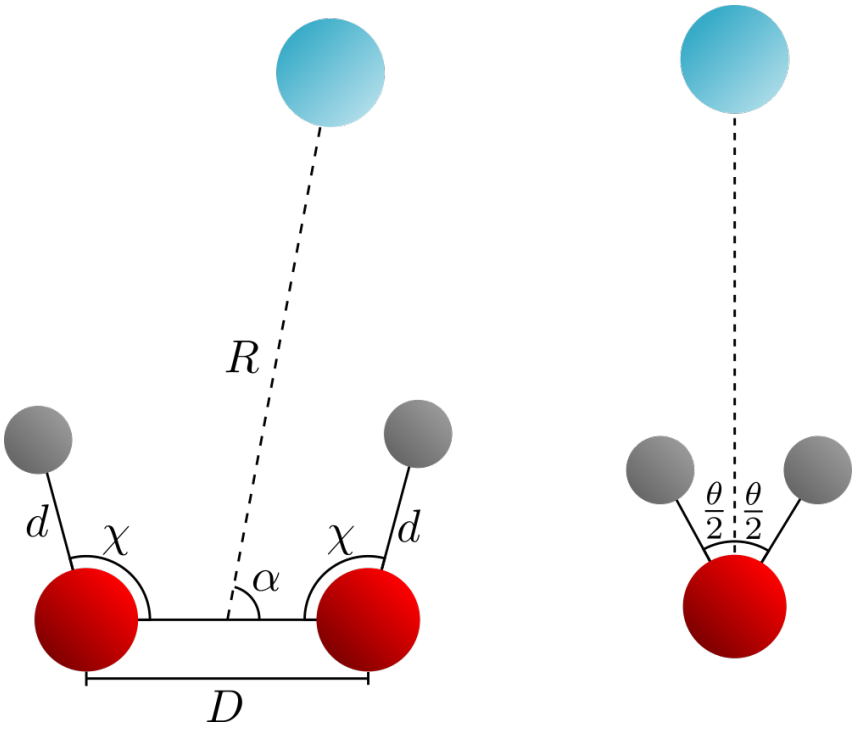
Foram usadas as notações:

$$\begin{aligned}
V_{ij} &\equiv \langle \Phi_0^{(i)} | \hat{V} | \Phi_0^{(j)} \rangle; \\
\tilde{V}_{ij} &= V_{ij} - V_{00}\delta_{ij}; \\
E_{ij} &\equiv E_i^{(0)} - E_j^{(0)}.
\end{aligned} \tag{1.25}$$

2 Geometria do Sistema $H_2O_2 - Kr$

O sistema estudado nesta dissertação é formado por uma molécula de peróxido de hidrogênio (H_2O_2) e um átomo de criptônio (Kr), e foi escolhido por possuir uma superfície de energia potencial bem caracterizada na literatura, inclusive pelos professores Ricardo Gargano e Luiz Roncaratti do nosso instituto (23, 24). Além disso, é um sistema relativamente simples, composto de apenas 5 átomos.

Figura 4 – Vista frontal (esquerda) e lateral (direita) do sistema estudado, composto de um átomo de Kr (acima, em azul) e uma molécula de H_2O_2 (abaixo), com os respectivos parâmetros geométricos.



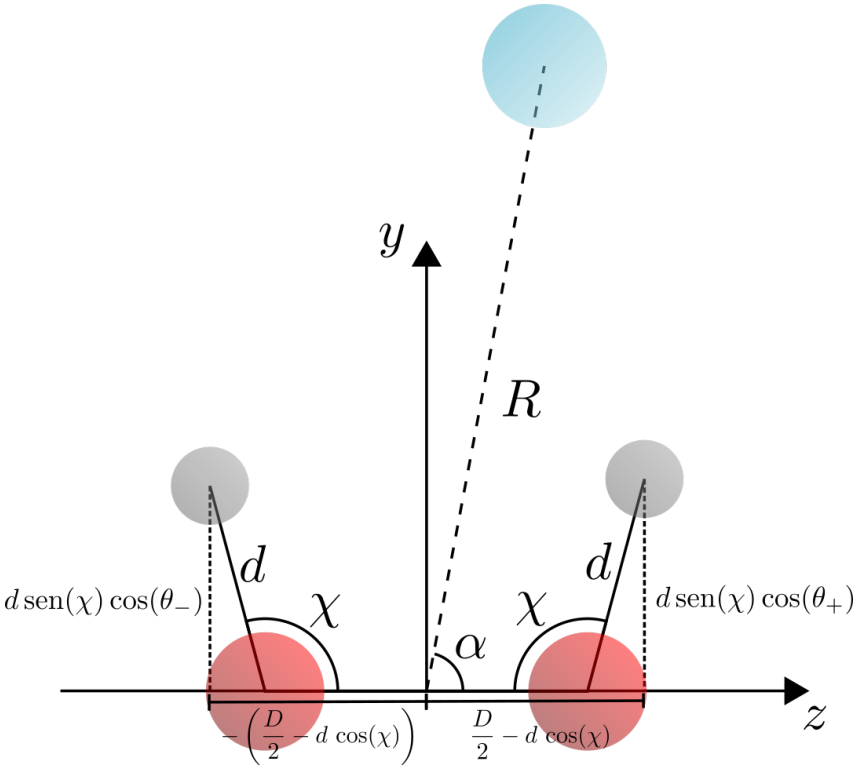
A geometria do sistema dar-se-á como na Figura 4. Foram fixados os seguintes parâmetros: o comprimento de ligação $O - O$ foi mantido a $D = 1,45$ Å, os comprimentos das ligações $O - H$ foram mantidos a $d = 0.966$ Å e os ângulos $O - O - H$ foram mantidos a $\chi = 108°$.

As variáveis sobre as quais a SEP será construída serão: a distância entre a Kr e a ligação $O - O$, denominada R, o ângulo diédrico formado entre os dois conjuntos de ligações $O - O - H$, denominado θ, e o ângulo entre a ligação $O - O$ e o vetor posição do átomo de Kr, denominado α (vide Figura 4). Para os cálculos desta pesquisa, R foi variado de 3.0 Å a 5.0 Å com um passo de 0.1 Å, θ foi variado de $0°$ a $360°$ com um passo de $10°$, e α foi variado de $0°$ a $90°$ com um passo de $30°$.

Para realizar estes cálculos, é necessário localizar precisamente cada átomo do sis-

tema no espaço. Introduz-se, então, um sistema de coordenadas cartesianas com origem no meio da ligação $O - O$ e eixo z colinear à ligação $O - O$. Como os dois oxigênios estão localizados no eixo z, podemos então dizer que estes se encontram respectivamente nos pontos $O_+ = (0, 0, D/2)$ e $O_- = (0, 0, -D/2)$. A convenção de sinais adotada indica se o átomo está no sentido positivo ou negativo de z. O átomo de criptônio e sua incidência vertical podem ser facilmente descritos por uma coordenada y variável: $\mathrm{Kr} = (0, R\,\mathrm{sen}\,\alpha, R\cos\alpha)$.

Figura 5 – Vista frontal da geometria da molécula H_2O_2 com as coordenadas y e z dos hidrogênios em destaque.



Já para as coordenadas dos hidrogênios, fez-se uso de trigonometria para expressar esse ponto em relação aos ângulos χ e $\theta_\pm = \pm\theta/2$ que cada ligação faz com o eixo y. A coordenada z será a soma da coordenada z do oxigênio respectivo e da projeção da ligação $O - H$ no eixo z: $z_\pm = \pm\left(\frac{D}{2} - d\cos(\chi)\right)$, como mostra a Figura 4. As coordenadas x e y serão dadas analisando as relações trigonométricas dos $\theta_\pm$, como na Figura 5. Com isso, obtém-se:

$$\begin{cases} x_\pm = d\,\mathrm{sen}\,(\chi)\,\mathrm{sen}\,(\theta_\pm) \\ y_\pm = d\,\mathrm{sen}\,(\chi)\cos(\theta_\pm) \\ z_\pm = \pm\left(\dfrac{D}{2} - d\cos(\chi)\right) \end{cases} \tag{2.1}$$

Parte II

Metodologia Computacional

3 Potenciais de Aprendizado de Máquina

Como demonstrado anteriormente, é de suma importância que as SEPs sejam construídas com a maior precisão possível para que sejam deduzidas propriedades físico-químicas em concordância com observações empíricas. No entanto, essa acurácia vem com elevados custos computacionais atrelados ao tempo de cálculo de cada ponto da SEP.

Para otimizar o uso desses recursos computacionais, introduziu-se na última década a utilização de Aprendizado de Máquina (ML, do inglês *Machine Learning*) na obtenção das SEPs, assim iniciando o estudo dos Potenciais de Aprendizados de Máquina (MLP, do inglês *Machine Learning Potentials*). ML é uma ferramenta de natureza estatística e computacional que visa solucionar problemas através do ajuste de um conjunto de dados a uma forma funcional não linear.

Temos duas formas de aprendizado de máquina:

- **Aprendizado Supervisionado:** Quando é fornecido ao programa um conjunto de dados de treino com entradas e saídas esperadas para que o algoritmo ajuste-se a eles.

- **Aprendizado Não Supervisionado:** Quando é fornecido ao programa somente um conjunto de dados de entrada para que o próprio algoritmo deduza as saídas.

No contexto do aprendizado supervisionado, o objetivo é: dado o conjunto de treino $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, determinar uma função $f(\mathbf{x}; \mathbf{p})$ com parâmetros $\mathbf{p}$ otimizados de forma a reproduzir com a maior fidedignidade possível a relação verdadeira e desconhecida $\mathbf{y} = F(\mathbf{x}; \mathbf{p})$. O procedimento de otimização dos parâmetros $\mathbf{p}$ é chamado de **treinamento**.

A ideia por trás da otimização desses parâmetros é minimizar uma **função custo ou erro** $\mathcal{L}(\mathbf{p})$ do modelo, que depende desses parâmetros. Essa função nos dá uma métrica do quão bem ajustado o modelo está ao conjunto fornecido ou, indiretamente, o quão longe $f(\mathbf{x}; \mathbf{p})$ está da função $F(\mathbf{x}; \mathbf{p})$. No contexto da regressão, comumente adota-se o **Erro Médio Quadrático** (MSE, do inglês *Mean Square Error*) como função custo:

$$\mathcal{L}(\mathbf{p}) = \sum_{i=1}^{N} (\mathbf{y}_i - f(\mathbf{x}_i; \mathbf{p}))^2 \tag{3.1}$$

A escolha dos parâmetros $\mathbf{p}$ deve ser tal que o modelo consiga generalizar bem para acomodar dados não vistos no processo de treinamento, seja na etapa de testes para avaliação de acurácia, seja na aplicação do modelo em situações práticas.

Para isso, o modelo não pode perder a sua capacidade de generalização em prol de um ajuste perfeito ao conjunto de treino. Esse fenômeno é denominado ***overfitting***, e geralmente acontece quando o modelo é complexo demais e/ou contém mais parâmetros que o necessário. Em outras palavras, dizemos que a **variância** do nosso modelo é alta e, consequentemente, temos mais parâmetros do que observações para ajustá-las.

Por outro lado, é possível que nosso modelo generalize demais e não consiga capturar bem a forma funcional que melhor compreende os dados. Nesse caso, temos o chamado ***underfitting***, que geralmente ocorre quando o modelo não contém os parâmetros necessárias para realizar o ajuste. Dessa forma, o **viés** do nosso modelo é alto, implicando no baixo desempenho do algoritmo mesmo com um número muito grande de observações disponíveis. Mais adiante, serão abordadas técnicas de mitigação do *overfitting* e do *underfitting*.

3.1 Métodos de Regressão

3.1.1 Regressão Linear

O exemplo mais básico de algoritmo de ML é a regressão linear. Apesar de simples, é uma ferramenta surpreendentemente poderosa na análise de um conjunto de dados, e seu estudo serve de base para melhor compreender conceitos fundamentais do ML. Além disso, como esse modelo é relativamente simples de se implementar e computacionalmente barato, ele pode ser utilizado como primeiro recurso para resolver um problema de regressão (25). Logicamente, problemas mais complexos vão requirir modelos mais complexos.

Considere um conjunto com N dados de treino $\{\mathbf{x}_i, y_i\}$, onde

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ik} \end{bmatrix}$$

é o vetor coluna correspondente às variáveis independentes (referidas como da i-ésima amostra do conjunto. Com k parâmetros $\{\beta_j\}$, é possível relacionar y_i a uma combinação linear de β_j's com x_{ij}'s, a menos de um erro ϵ_i:

$$y_i - \epsilon_i = f(\mathbf{x}_i; \{\beta_j\}) = \beta_1 x_{i1} + ... + \beta_k x_{ik} = \sum_{j=1}^{k} \beta_j x_{ij} \tag{3.2}$$

Geometricamente falando, estamos buscando o hiperplano $k-$dimensional que melhor se ajusta aos pontos $\{\mathbf{x}_i, y_i\}$. Dada a natureza do somatório em (3.2), convém agrupar todos os vetores $\mathbf{x}_i$ numa só matriz chamada **matriz de design**:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nk} \end{bmatrix} \tag{3.3}$$

Da mesma forma, podemos vetorizar as demais quantidades:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}, \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{3.4}$$

Assim, reescrevemos (3.2) como:

$$\mathbf{y} - \boldsymbol{\epsilon} = f(\mathbf{X}; \boldsymbol{\beta}) = \mathbf{X}\boldsymbol{\beta} \implies \mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.5}$$

$$\therefore \boldsymbol{\epsilon}^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^2 \tag{3.6}$$

De onde temos que $\epsilon_i^2 = (y_i - f(\mathbf{x}_i; \boldsymbol{\beta}))^2$ são os resíduos quadrados, ou seja, os erros de cada entrada separadamente. A soma desses resíduos quadrados nos dá o MSE:

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{N} \epsilon_i^2 = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i; \boldsymbol{\beta}))^2 \tag{3.7}$$

Ou, em notação matricial,

$$\mathcal{L}(\boldsymbol{\beta}) = \|\boldsymbol{\epsilon}\|^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{3.8}$$

Onde $\|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v} = \sum_{i=1}^{N} v_i^2$ é a norma euclidiana (L2). Podemos minimizar analiticamente esse erro derivando-o com relação aos parâmetros $\boldsymbol{\beta}$ e igualando a zero, obtendo os coeficientes treinados:

$$\boldsymbol{\beta}^{(o)} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.9}$$

3.1.2 Regressão Regularizada (RR)

Um problema que pode surgir na avaliação de modelos de regressão é a chamada *supercolinearidade*: quando duas ou mais dimensões das entradas do conjunto de dados apresentam forte interdependência. Um exemplo físico desse acontecimento ocorre na avaliação de moléculas diatômicas (25). Caso estejamos analisando a massa e o número atômico de cada átomo, espera-se que átomos mais pesados possuam maior número atômico. Note que, quanto maior o número de características, maior a probabilidade que duas delas apresentem supercolinearidade.

Uma consequência direta desse fenômeno é que o vetor de coeficientes $\boldsymbol{\beta}$ pode possuir componentes com valores muito grandes, afetando a complexidade do modelo. A matriz $\mathbf{X}$ também será afetada, uma vez que $\mathbf{X}^T\mathbf{X}$ pode possuir uma inversa computacionalmente inviável ou até impossível de se calcular. Isso é um exemplo claro de *overfitting*, uma vez que a variância do nosso modelo está elevada por ele ser complexo demais.

Assim, ao analisar diversas caraterísticas simultaneamente, convém introduzir um **parâmetro de regularização** λ na função custo (26):

$$\mathcal{L}_{RR}(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i; \boldsymbol{\beta})\right)^2 + \lambda \sum_{j=1}^{k} \beta_j^2 \tag{3.10}$$

Ou, em notação matricial,

$$\mathcal{L}_{RR}(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \tag{3.11}$$

Como λ é um parâmetro escolhido *a priori* e é ajustado fora da etapa de treino, dizemos que ele é um **hiperparâmetro** do modelo. Podemos, assim como os parâmetros internos, otimizá-lo de acordo com a complexidade de cada modelo, fazendo uma troca entre viés e variância: quanto maior o valor de λ, maior o viés do modelo. Portanto, para evitar *underfitting*, devemos escolher esse valor com cautela.

Ainda é possível resolver analiticamente o sistema, obtendo os parâmetros otimizados em função da regularização:

$$\boldsymbol{\beta}^{(r)} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbb{1})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.12}$$

Onde $\mathbb{1}$ é a matriz identidade de dimensão $k \times k$. Note que:

$$\lim_{\lambda \to 0} \boldsymbol{\beta}^{(r)} = \boldsymbol{\beta}^{(o)} \tag{3.13}$$

3.1.3 Regressão Não-Linear

Há casos em que a regressão linear pura não é suficiente para ajustar os dados. Por exemplo, vamos supor que seja necessário ajustar uma parábola que não passe pela origem ao invés de uma reta que passe pela origem. Nesse caso, é preciso ampliar nosso espaço de entrada aplicando cada entrada a um conjunto de $C \geq k$ **funções não-lineares** $\{\phi_j\}$, tal que:

$$\{\mathbf{x}_1, ..., \mathbf{x}_k\} \mapsto \{\phi_1(\mathbf{x}_i), ..., \phi_C(\mathbf{x}_i)\} \Rightarrow f(\mathbf{x}_i; \boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j \phi_j(\mathbf{x}_i), \tag{3.14}$$

onde $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_N \end{bmatrix}^T$ são o coeficientes do ajuste. No exemplo da parábola fora da origem, teríamos que

$$y = ax^2 + bx + c = a\phi_2(x) + b\phi_1(x) + c\phi_0(x), \text{ com :}$$

- $\phi_2(x) = x^2$,
- $\phi_1(x) = x$, e
- $\phi_0(x) = 1$.

Note que, nesse exemplo, não lidamos mais com uma só dimensão ($\{x\}$), mas com três ($\{\phi_0(x), \phi_1(x), \phi_2(x)\}$). Ou seja, aumenta-se a dimensionalidade do espaço de entrada em prol de melhorar o ajuste do modelo. Nesse contexto, implementar o parâmetro de regularização é crucial, pois fica mais fácil de ocorrer *overfitting* nesse espaço de maior dimensionalidade, que é chamado de **espaço de características**.

É possível seguir todos os passos da seção anterior, com as substituições $x_{ij} \rightarrow \phi_j(\mathbf{x}_i)$ e $\mathbf{X} \rightarrow \boldsymbol{\Phi}$, onde

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_C(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_C(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_C(\mathbf{x}_N) \end{bmatrix} \tag{3.15}$$

Contudo, há um sério problema relacionado à regressão não-linear: o cálculo da matriz $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ e sua inversa são computacionalmente caros, devido ao seu tamanho ser $C \times C$, i.e. maior que o da matriz $\mathbf{X}^T\mathbf{X}$, que é $k \times k$. Sobretudo, cada elemento é dado por

$$(\boldsymbol{\Phi}^T\boldsymbol{\Phi})_{ij} = \sum_{l=1}^{C} \phi_l(\mathbf{x}_i) \cdot \phi_l(\mathbf{x}_j), \tag{3.16}$$

ou seja, cada elemento de matriz requer o cálculo do produto de diferentes funções avaliados em diferentes pontos, os quais a depender da forma funcional de cada $\phi_l(\mathbf{x})$, podem ser bastante onerosos do ponto de vista computacional.

3.1.4 Regressão Regularizada de Kernel (KRR)

Para contornar os problemas computacionais elucidados no método anterior, vamos substituir a matriz em 3.16 por outra, dada por:

$$\mathbf{K} = \begin{bmatrix} \mathcal{K}(\mathbf{x}_1, \mathbf{x}_1) & \mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \mathcal{K}(\mathbf{x}_1, \mathbf{x}_N) \\ \mathcal{K}(\mathbf{x}_2, \mathbf{x}_1) & \mathcal{K}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \mathcal{K}(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{K}(\mathbf{x}_N, \mathbf{x}_1) & \mathcal{K}(\mathbf{x}_N, \mathbf{x}_2) & \cdots & \mathcal{K}(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}. \tag{3.17}$$

$\mathbf{K}$ é dita matriz de Gram, e $\mathcal{K} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ é dita **função de kernel** (9, 27). A substituição das C funções ϕ_j's por uma só $\mathcal{K}$ é chamado *kernel trick*, e é possível devido ao **Teorema de Mercer**, o qual associa produtos internos no espaço de características $\phi_l(\mathbf{x}_i) \cdot \phi_l(\mathbf{x}_j)$ com um produto interno no espaço de entradas:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi_l(\mathbf{x}_i) \cdot \phi_l(\mathbf{x}_j) \tag{3.18}$$

Em particular, a substituição só tem validade se o kernel e, por consequência, a matriz de Gram forem positivos definidos, ou seja,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \tag{3.19}$$

para quaisquer entradas $\mathbf{x}_i, \mathbf{x}_j$ e coeficientes $c_i, c_j \in \mathbb{R}$. Assim, nosso modelo de **Regressão Regularizada de Kernel** (KRR, do inglês *Kernel Ridge Regression*) tem forma funcional dada por:

$$f(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j \mathcal{K}(\mathbf{x}, \mathbf{x}_j) \tag{3.20}$$

onde α_j são os coeficientes da regressão (28). Nesse sentido, o kernel é uma medida de similaridade entre valores de entrada, e qualquer valor que não esteja presente no conjunto de treino pode ser determinado com base nessa medida.

A eficiência do modelo KRR é determinada durante o treino pelo ajuste dos coeficientes α_i's e antes do treino com a escolha da função de kernel e do parâmetro de regularização $\lambda \in \mathbb{R}$. Denomina-se, portanto, a forma funcional de $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ e seus parâmetros internos como **hiperparâmetros** do modelo, assim como λ. Além da determinação

dos α_i's, é possível otimizar os hiperparâmetros de cada forma funcional de maneira que o ajuste feito utilizando $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ seja o melhor possível.

3.1.4.1 Exemplos de Funções de Kernel

Consideremos a família de funções de kernel de Matérn:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}\right) \sum_{k=0}^{n} \frac{(n+k)!}{(2n)!} \binom{n}{k} \left(2\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}\right)^{n-k}, \tag{3.21}$$

onde $\|\cdot\|_2$ é a norma L2 (euclidiana) e $n, \sigma \in \mathbb{R}$. Note que, no limite $n \to \infty$, produzimos o *kernel gaussiano*:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right). \tag{3.22}$$

Por outro lado, quando $n = 0$, produzimos o *kernel exponencial*:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = -\exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}\right). \tag{3.23}$$

Caso substitua-se a norma em 3.23 pela norma L1 (também conhecida como norma de Manhattan), dada por

$$\|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{l=1}^{k} |x_{il} - x_{jl}|, \tag{3.24}$$

geramos o *kernel de Laplace*:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = -\exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_1}{\sigma}\right). \tag{3.25}$$

3.2 Representações de Sistemas Moleculares

Como os algoritmos de ML em geral são extremamente sensíveis à qualidade e à quantidade de dados que lhes são fornecidos, é importante determinar como as configurações internucleares serão alimentadas a tais algoritmos. Pode-se pensar que cada configuração internuclear corresponde a um conjunto de M pontos num espaço 4-dimensional $\mathcal{X} = \{(Z_i, \mathbf{R}_i)\}_{i=1}^{M}$, onde M é o número de núcleos.

Para que o algoritmo possa tirar conclusões fisicamente aceitáveis e úteis, é necessário fornecer o máximo de conhecimento físico *a priori* na entrada. Dados como simetrias, invariâncias e propriedades físico-químicas relevantes o auxiliam a detectar padrões e o

tornam mais generalizável. Via de regra, uma MLP nunca é treinada com base em coordenadas cartesianas puras.

Nessa ótica, um **descritor molecular** é uma função que mapeia as coordenadas cartesianas nucleares numa matriz de tamanho arbitrário, a qual servirá de entrada para o algoritmo. Geralmente, escolhem-se coordenadas internas ao sistema, tais como distâncias internucleares e ângulos de ligação, uma vez que essas quantidades são invariantes por rotações e translações do sistema no espaço.

Um exemplo de descritor molecular é a matriz de Coulomb (CM, do inglês *Coulomb Matrix*), com elementos definidos como

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4}, \; i = j \\ \dfrac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|}, \; i \neq j \end{cases} \tag{3.26}$$

Por se tratar de uma matriz simétrica, essa descrição respeita a invariância por permutações entre átomos de mesmo número atômico. Ao considerar apenas as distâncias internucleares, ela também respeita invariâncias por rotações e translações. Além disso, a CM é um descritor global, justamente por não considerar somente as interações entre átomos próximos, mas sim todos aqueles presentes no sistema.

Outra descritor molecular é a RE, que leva em consideração apenas as distâncias internucleares normalizadas e recíprocas:

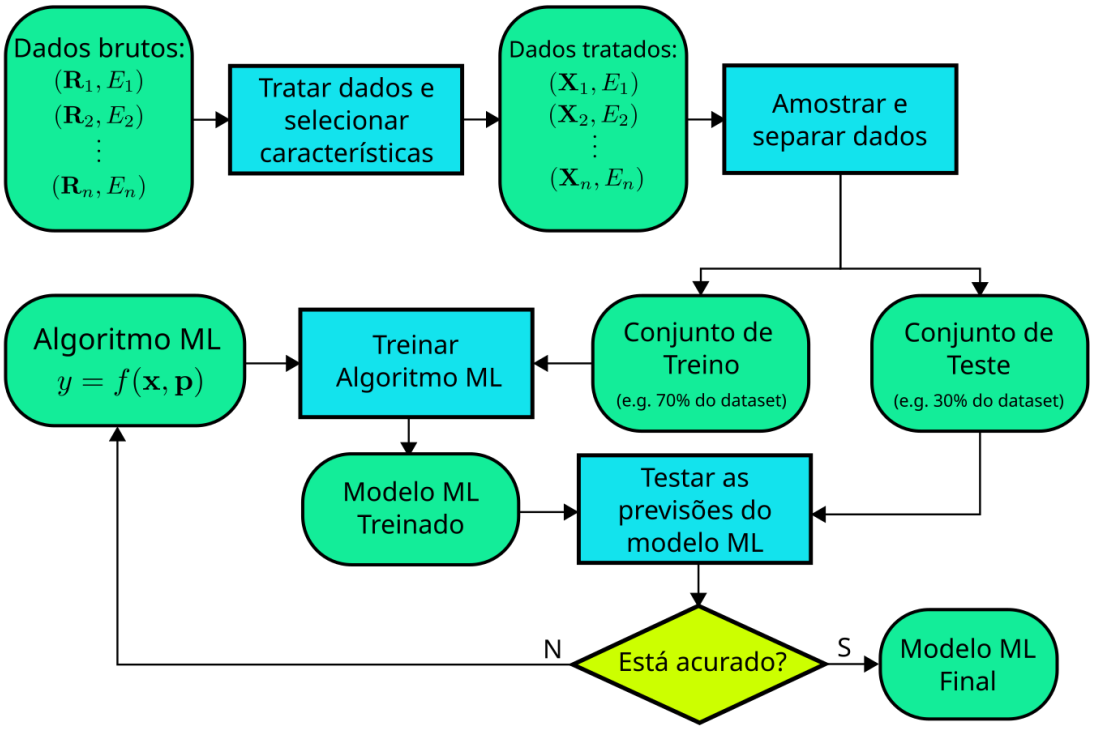
$$X_{ij}^{RE} = \frac{R_{ij}^{eq}}{\|\mathbf{R}_i - \mathbf{R}_j\|}(1 - \delta_{ij}) \tag{3.27}$$

Esse descritor também é invariante por rotações e translações. Para garantir a invariância por translações, pode-se ordenar esse descritor com base na soma das repulsões nucleares com relação aos demais átomos. Ao contrário da CM, a RE é um descritor local, por levar em consideração as distância de equilíbrio entre pares de átomos, ao invés de todos eles juntos.

3.3 Como Máquinas Aprendem

A Figura 6 ilustra o fluxo de treinamento de um algoritmo de aprendizado de máquina. Primeiramente, elenca-se um conjunto de dados brutos, coletados a partir de experimentos e/ou simulações teóricas por métodos *ab initio*, tais como os estudados no Capítulo 1. No caso de uma tarefa de aprendizado supervisionado, reúnem-se as variáveis independentes (e.g. configurações nucleares) e as variáveis dependentes (e.g. energias eletrônicas).

Figura 6 – Fluxograma algorítmico que ilustra o processo de Aprendizado de Máquina em resumo.



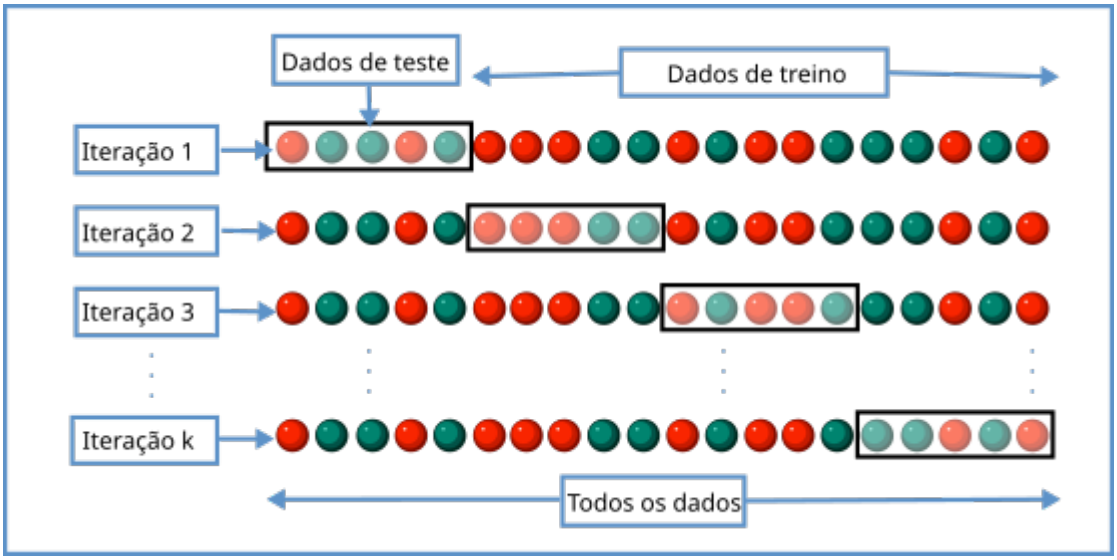
Em seguida, é realizado um tratamento dos dados brutos de forma a selecionar as características físico-químicas de maior relevância para o estudo que está sendo realizado. No caso das SEPs, aplica-se um descritor molecular nas configurações nucleares de forma a deixá-las invariantes por rotações e translações no espaço, bem como introduzir simetrias de permutação de átomos com mesmo número atômico.

Logo após, é feita a amostragem dos dados e a separação destes em dois grupos: um conjunto de treino, que compõe uma grande porcentagem dos dados, e um conjunto de teste com o restante dos dados. Com um algoritmo de ML selecionado (e.g. KRR), podemos ajustar seus parâmetros ao conjunto de treino, minimizando a função erro de forma analítica ou numérica (por meio de uma descida de gradiente, por exemplo).

No conjunto de treino, ainda é possível realizar uma **validação cruzada** iterada k-vezes (29), que é ilustrada na Figura 7. A ideia desse procedimento é comparar o desempenho em amostras diferentes de dados de treino antes de prosseguir para a etapa de testes. Se a performance do modelo for igualmente satisfatório entre as amostras, sabemos que não houve *overfitting* nem viés na seleção dos dados.

Após treinado, podemos avaliar a acurácia de suas previsões no conjunto de teste, o qual ele não viu no processo de treino. Se as previsões estiverem acuradas, i.e. o erro do conjunto de teste for pequeno o suficiente para as aplicações propostas para o algoritmo e, além disso, comparável ao erro do conjunto de treino, saberemos que o algoritmo soube generalizar bem o tratamento de dados desconhecidos e está apto a ser utilizado em

Figura 7 – Ilustração do processo de validação cruzada iterada k-vezes.



Fonte: Wikimedia Commons (30) (Adaptado).

cálculos de outras propriedades físicas e químicas do sistema molecular.

Caso contrário, teremos que fazer ajustes no nosso modelo. Inclui-se nesses ajustes: otimização de hiperparâmetros, troca de um algoritmo por outro, re-amostragem e retratamento dos dados, isto é, trocando um descritor molecular por outro. No geral, esta é a metodologia empregada para diversos algoritmos de ML, e que tem obtido sucesso recorrente na literatura.

4 Softwares e Procedimentos

4.1 Construção do Conjunto de Dados

Para treinar um algoritmo de ML, são necessários uma quantidade satisfatória de dados de referência. Para conseguir esses dados, fizemos cálculos de uma SEP no software *Gaussian 16* (31), utilizando o método MP4, já descrito na Seção 1.3. A SEP foi varrida nos intervalos especificados no Capítulo 2. Todos os cálculos no *Gaussian* utilizaram de 8 a 16 threads de processadores com frequência igual ou superior a 3,6 GHz, cada um a 100% de sua capacidade. Foram reservados, ainda, de 8 a 16 GB de memória RAM para os cálculos.

Foi usada a base aug-cc-pVTZ, pois esta obteve resultados satisfatórios na literatura (23). Contudo, esta base pode apresentar um erro durante a contração das gaussianas conhecida como Erro de Sobreposição do Conjunto de Base (BSSE, do inglês *Basis Set Superposition Error*), que pode levar a desvios de energia (32). Para evitar isso, utiliza-se a correção *counterpoise*, que utiliza a base dimérica (união das bases de cada monômero) no cálculo de todas as energias de interação ao invés de cada base separadamente.

4.2 *MLTools*: Uma suíte de scripts para *MLatom*

Na pesquisa desta dissertação, foi utilizada o software *MLatom* (33) para treino e teste do algoritmo KRR com kernel Gaussiano da equação (3.22) e descritor molecular RE da equação (3.27), implementados no executável *MLatomF*. Convencionou-se na literatura designar essa combinação de algoritmo KREG, que provou ser a melhor em sistemas poliatômicos (28, 34, 16).

MLatom trabalha com base em arquivos de configurações, os quais contém palavras-chave e variáveis que serão dissecadas pelo software e descrevem como serão realizados os cálculos, de forma similar ao que o *Gaussian* faz.

Os procedimentos de treino e teste do modelo seguiram de acordo com os procedimentos elucidados na Seção 3.3. Para automatizar a execução destes, bem como a análise de resultados e a geração das figuras, foi criado um conjunto de *scripts* de automação intitulado *MLTools*. Esses *scripts* foram escritos na linguagem *Python* versão 3.9, e fizeram uso das seguintes bibliotecas:

- `os`, para integrar o *MLTools* ao sistema operacional Linux e ao *MLatom*.
- `numpy` (35), para cálculos numéricos envolvendo arrays.

- `pandas`, para análises estatísticas dos dados obtidos.
- `matplotlib` (36), para gerar os gráficos.

Grande parte do diferencial do *MLTools* está na parte de gerar arquivos de geometrias tanto para o *MLatom* como para o *Gaussian*, assim como extrair energias dos logs do *Gaussian* e escrevê-los em arquivos nos formatos aceitos pelo *MLatom*. Além disso, foi possível rodar todas as entradas do *MLatom* em sequência de forma automática, minimizando assim o tempo gasto no processo de treino e teste do modelo.

Além disso, o *MLTools* é capaz de automaticamente gerar várias figuras de interesse da SEP, como suas curvas de nível e seções transversais. Também é possível gerar as curvas de aprendizado do nosso modelo, isto é, como o desempenho do modelo varia com o número de pontos de treino dele. Todos os gráficos da seção 5 foram feitos por meio deste programa, cujo código-fonte encontra-se disponibilizada em um repositório online (37).

Em primeiro lugar, precisamos converter as geometrias cartesianas puras em descritores moleculares que o *MLatom* consiga ler e interpretar. Para isso, a primeira entrada do *MLatom* foi usando o comando `XYZ2X`, que realiza essa conversão para o descritor RE e determina que os pares de átomos iguais dentro da molécula de H_2O_2 podem ser permutados. Em seguida, fazemos o mesmo para as geometrias de equilíbrio, as quais serão utilizadas no processo de amostragem.

A técnica de amostragem utilizada chama-se **amostragem baseada em estrutura** (34), e se baseia nas geometrias de equilíbrio para elencar quais pontos são melhores para treinar o algoritmo de forma mais eficiente. Utilizamos 75% dos dados para treinar a SEP, ou seja, 2268 dos 3024 pontos. Ainda definimos que 75% dos dados de treino serão utilizados como subtreino no processo de validação cruzada, deixando o restante para a validação.

Para finalizar, fazemos o treino e teste do nosso algoritmo KRR em cima dos dados tratados e amostrados. Chamamos os dados de treino, teste, subtreino e validação por meio de seus respectivos índices no arquivo de entrada original. Também especificamos se vamos otimizar os hiperparâmetros σ e λ, bem como implementar um kernel gaussiano invariante por permutações. O *MLTools* foi desenvolvido justamente para executar as entradas acima sequencialmente, agilizando o processo de treino.

Parte III

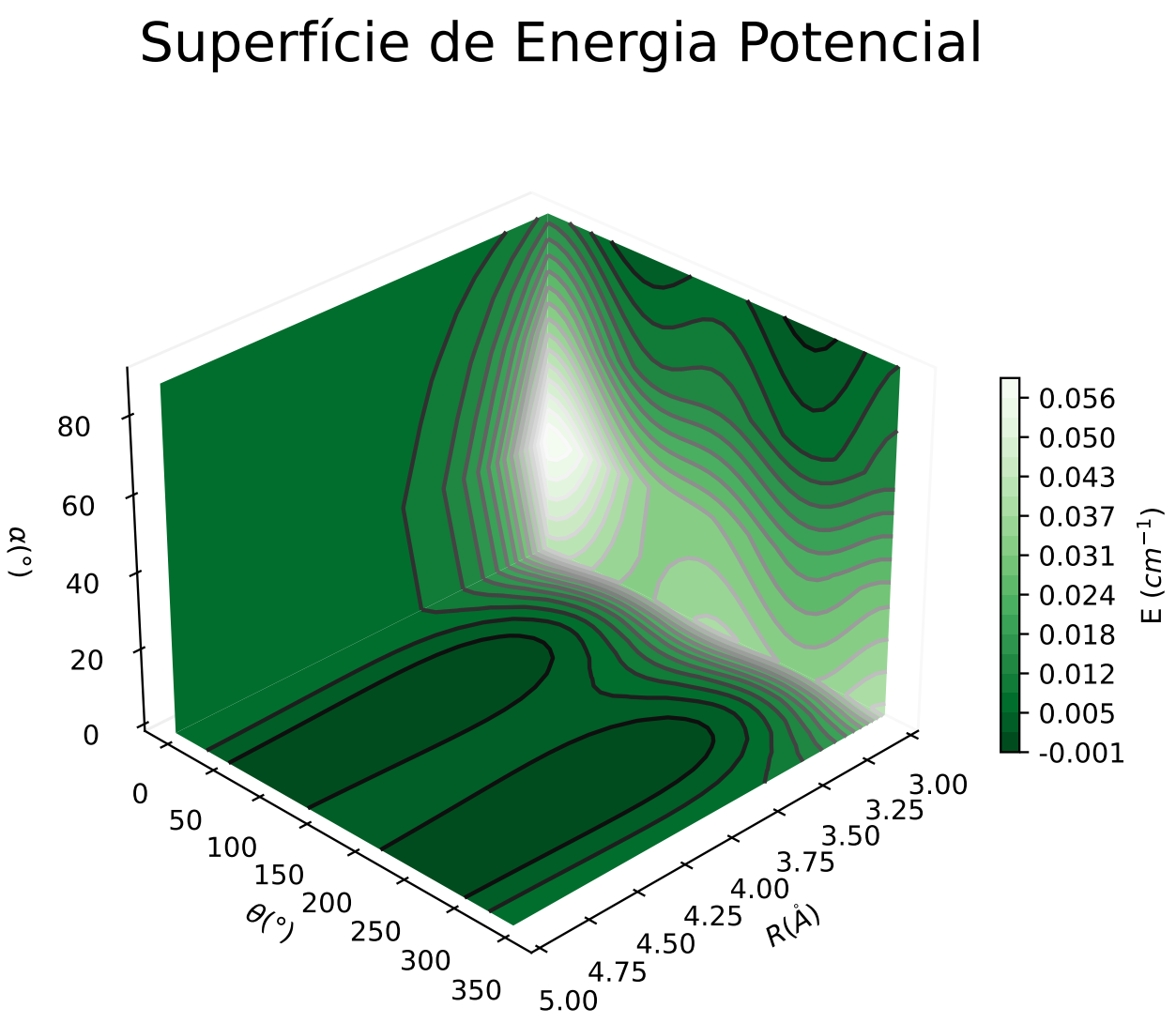
Resultados

5 Análise dos Resultados

5.1 Topografia das Superfícies

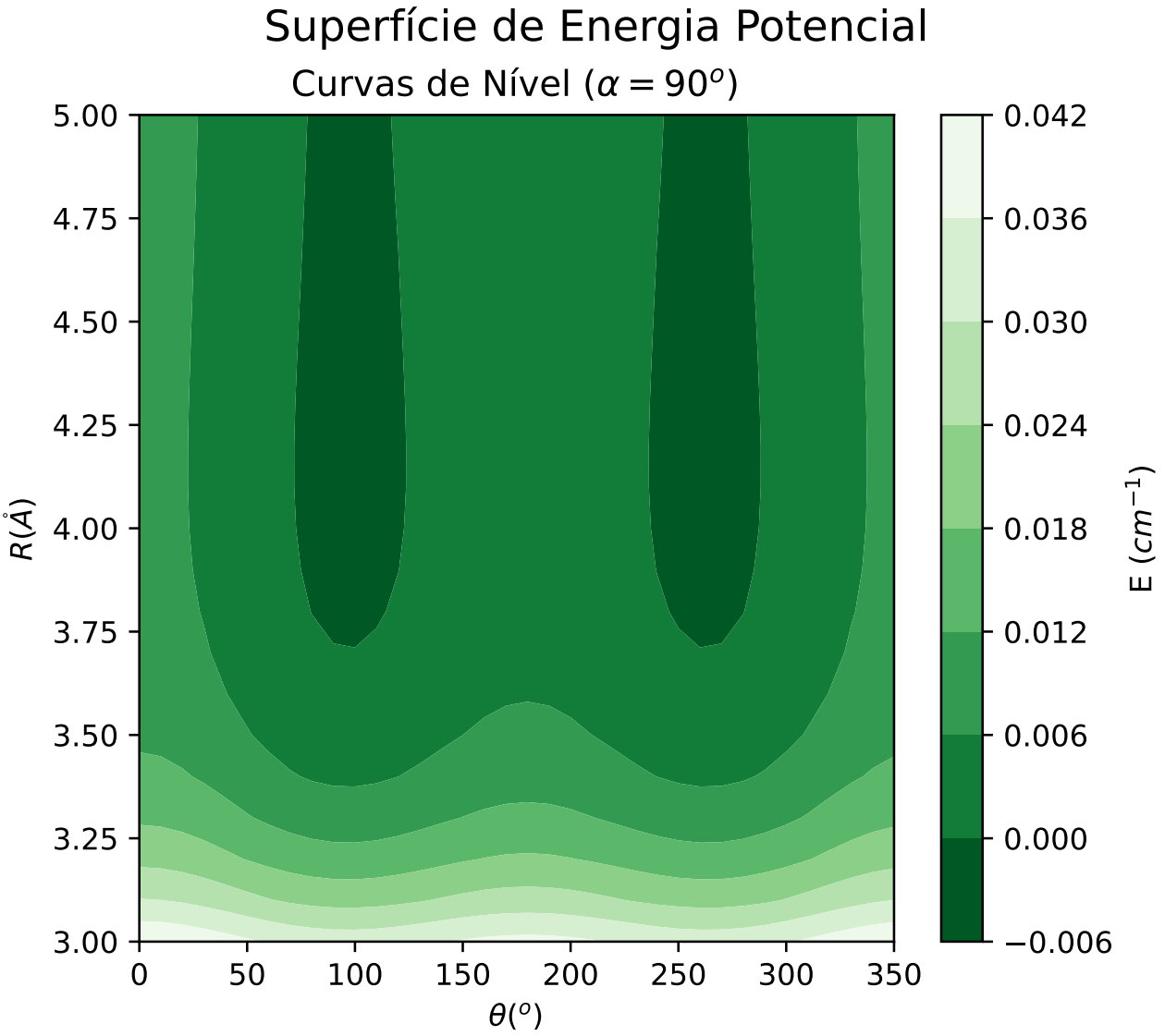
Ao final do treino do algoritmo KREG, obteve-se os pontos da SEP do sistema $H_2O_2 - Kr$. Como trata-se de uma superfície em um espaço quadridimensional, foram feitas projeções dessa superfície em curvas de nível em três planos diferentes: $R \times \theta$, $R \times \alpha$ e $\theta \times \alpha$. Com esses três planos apropriadamente orientados e posicionados no espaço 3D dos parâmetros geométricos analisados, obtém-se a Figura 8.

Figura 8 – Curvas de Nível da SEP do sistema $H_2O_2 - Kr$ gerados pelo algoritmo KREG treinado com dados do nível de cálculo MP4/aug-cc-pVTZ.



Para melhor visualização e comparação dos resultados obtidos com outros disponíveis na literatura, faremos curvas de nível dessa SEP em pontos de mínimo, os quais são de extrema importância nas aplicações usuais de uma SEP, como discutido na subseção 1.1.1. Analisando os dados obtidos após o treino, vemos que o ponto de menor energia se localizam em $R_e = 3.68\,\text{Å}$, $\theta_e = 100°$e $\alpha_e = 90°$. Tais valores são compatíveis com aqueles encontrados na literatura (23).

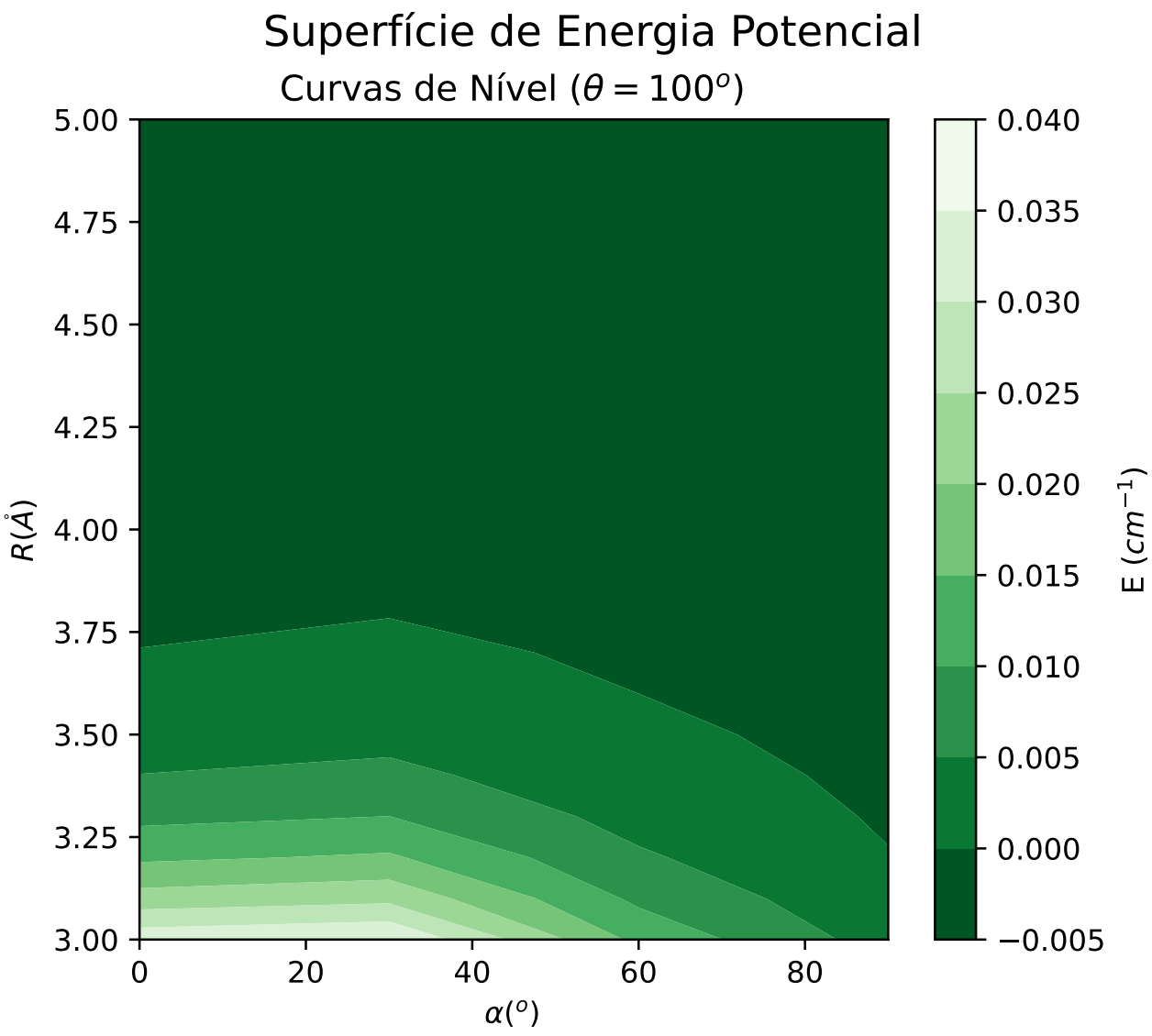
Figura 9 – Curvas de Nível da SEP do sistema $H_2O_2 - Kr$ no plano $R \times \theta$ gerados pelo algoritmo KREG treinado com dados do nível de cálculo MP4/aug-cc-pVTZ.



Na Figura 9, pode-se ver as curvas de nível da SEP no plano $R \times \theta$, construídas com $\alpha = \alpha_e$. Nota-se um padrão similar ao da SEP de referência, no qual temos vales ao redor de $\theta = 100°$e $\theta = 260°$, que são as configurações *cis* e *trans*, respectivamente. Esses vales prolongam-se até a distância máxima analisada, que é de $5,0\,\text{Å}$, mas em tese esses vales continuariam até eventualmente alcançarem energia zero no infinito.

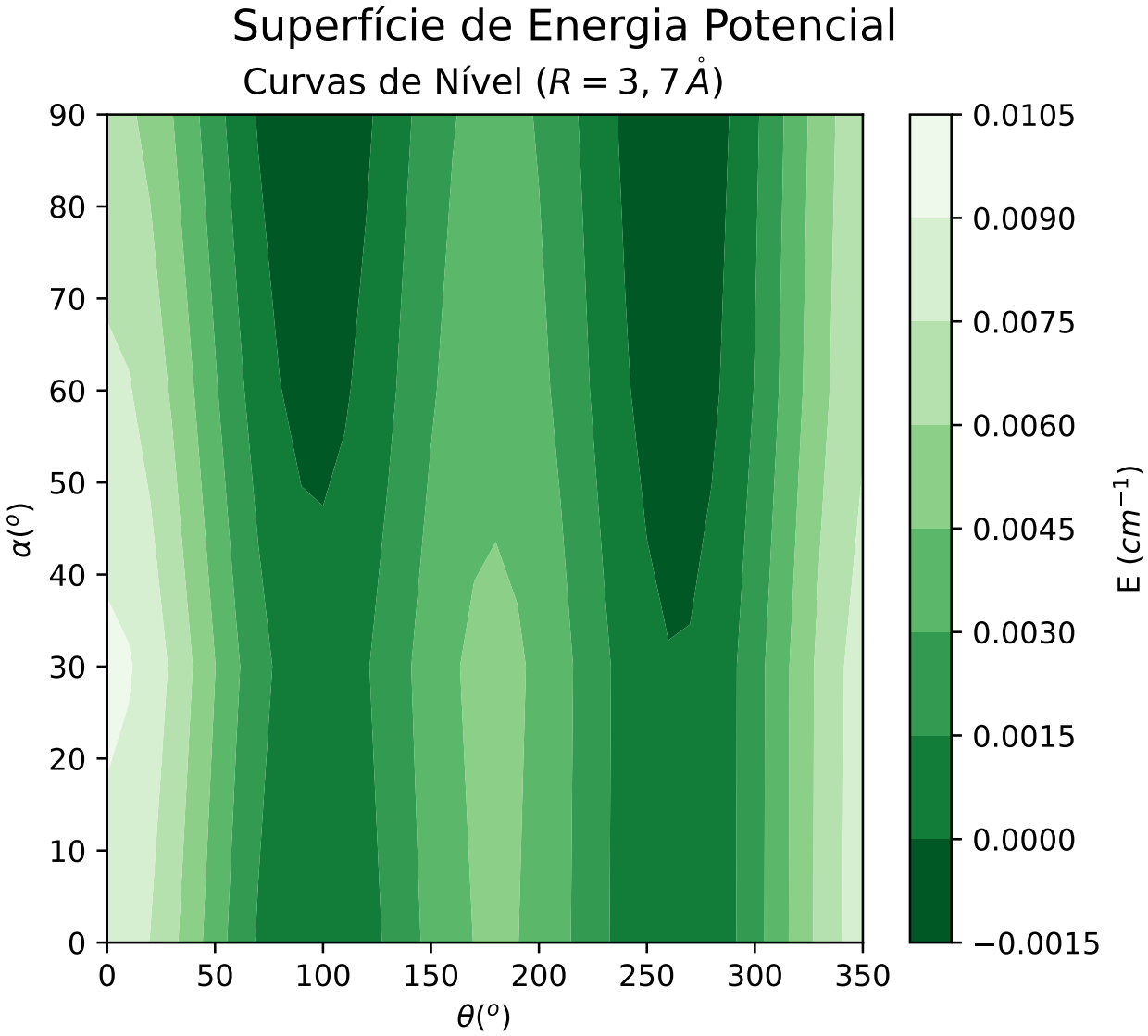
Radialmente, a SEP tem energias elevadas quando a distância entre o Kr e a molécula de H_2O_2 é curta, e rapidamente decai até a distância de equilíbrio R_e. A partir daí, a energia aumenta gradativamente até que, no infinito, ela alcançaria a energia zero. Tal energia foi tomada fazendo o cálculo de energia de interação entre o H_2O_2 na configuração de equilíbrio e o Kr a uma distância muito maior que os comprimentos de ligação internos do peróxido de hidrogênio ($R >> D, d$).

Figura 10 – Curvas de Nível da SEP do sistema $H_2O_2 - Kr$ no plano $R \times \alpha$ gerados pelo algoritmo KREG treinado com dados do nível de cálculo MP4/aug-cc-pVTZ.



Na Figura 10, podemos ver as curvas de nível da SEP no plano $R \times \alpha$. Nota-se que a energia decai conforme aumentamos a distância internuclear, como já era esperado. Contudo, vemos que a energia também diminui conforme o ângulo de incidência chega a 90°.

Figura 11 – Curvas de Nível da SEP do sistema $H_2O_2 - Kr$ gerados pelo algoritmo KREG treinado com dados do nível de cálculo MP4/aug-cc-pVTZ.

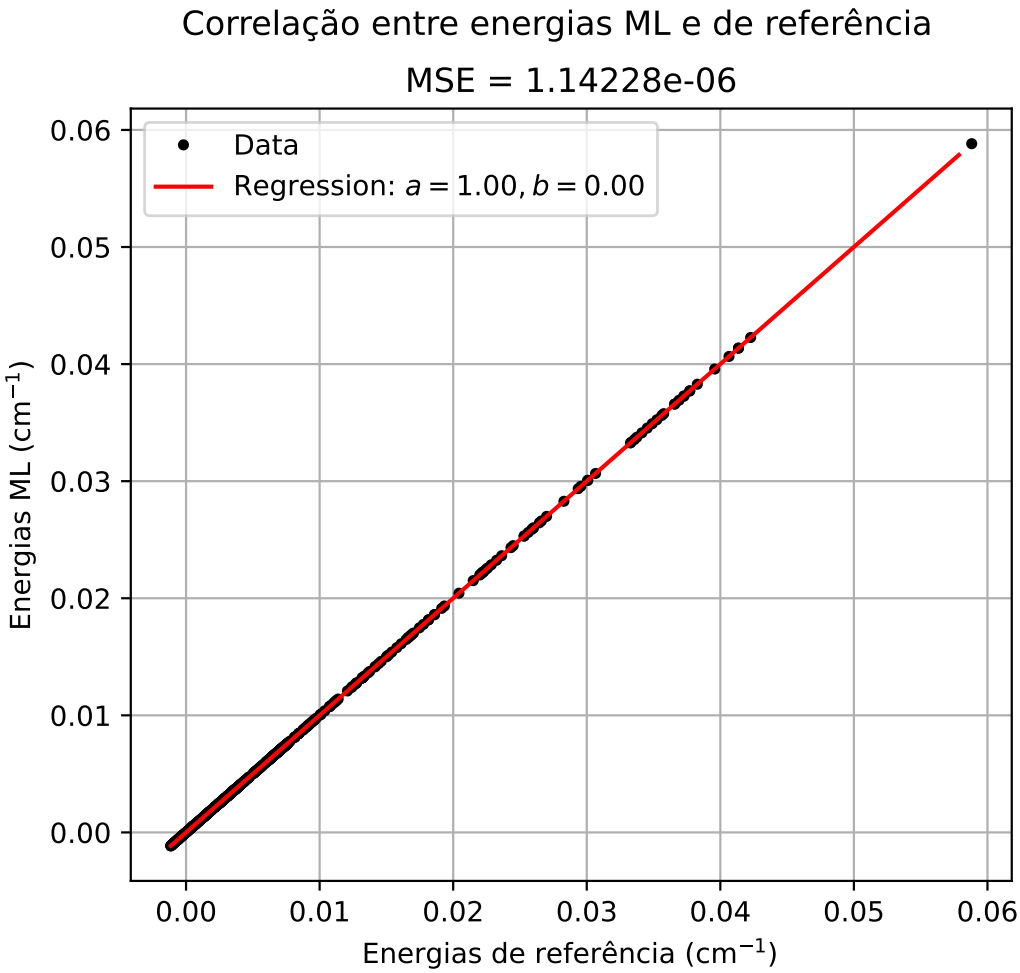


Quanto ao gráfico $\alpha \times \theta$, percebe-se uma distorção do gráfico de duplo poço esperado para a CEP com relação ao ângulo diédrico. Essa distorção é causada pela proximidade do Kr com de um dos hidrogênios do $H_2O_2 - Kr$, o que acarreta no aumento de energia especialmente quando $\alpha \approx 30°$, isto é, quando os átomos estão alinhados. Por outro lado, ao afastarmos o Kr do $H_2O_2 - Kr$, verifica-se que a distorção do gráfico é eliminada.

5.2 Desempenho do Algoritmo

Uma vez analisadas as SEPs do ponto de vista topográfico, vamos analisar o desempenho do algoritmo em produzir resultados comparáveis aos do método *ab initio* (MP4). Analisaremos, também, quantos pontos de treino (i.e. quantas configurações nucleares e

Figura 12 – Gráfico de correlação entre as energias de referência (nível MP4/aug-cc-pVTZ) e aquelas produzidas pelo algoritmo KREG para o sistema $H_2O_2 - Kr$.



energias correspondentes) são necessárias para gerar tais resultados.

Em primeiro lugar, vamos analisar o gráfico de correlação entre as energias de referência e as produzidas pelo algoritmo KREG na Figura 12. Este gráfico nos diz o quão próximos os valores de energia geradas pelo algoritmo estão dos valores de referência. Ao fazer um ajuste linear no gráfico, vemos que quanto mais próximo da função identidade ($f(x) = x$), mais corretas são as previsões de energia do nosso modelo.

Tabela 1 – Erros Raiz-Média-Quadrática (RMSE) e coeficientes de correlação nas etapas de treino e teste do algoritmo KREG para o sistema $H_2O_2 - Kr$.

Métrica de aprendizado	Treino	Teste
RMSE (cm^{-1})	0.405	0.455
Correlation coefficient	0.999	0.999

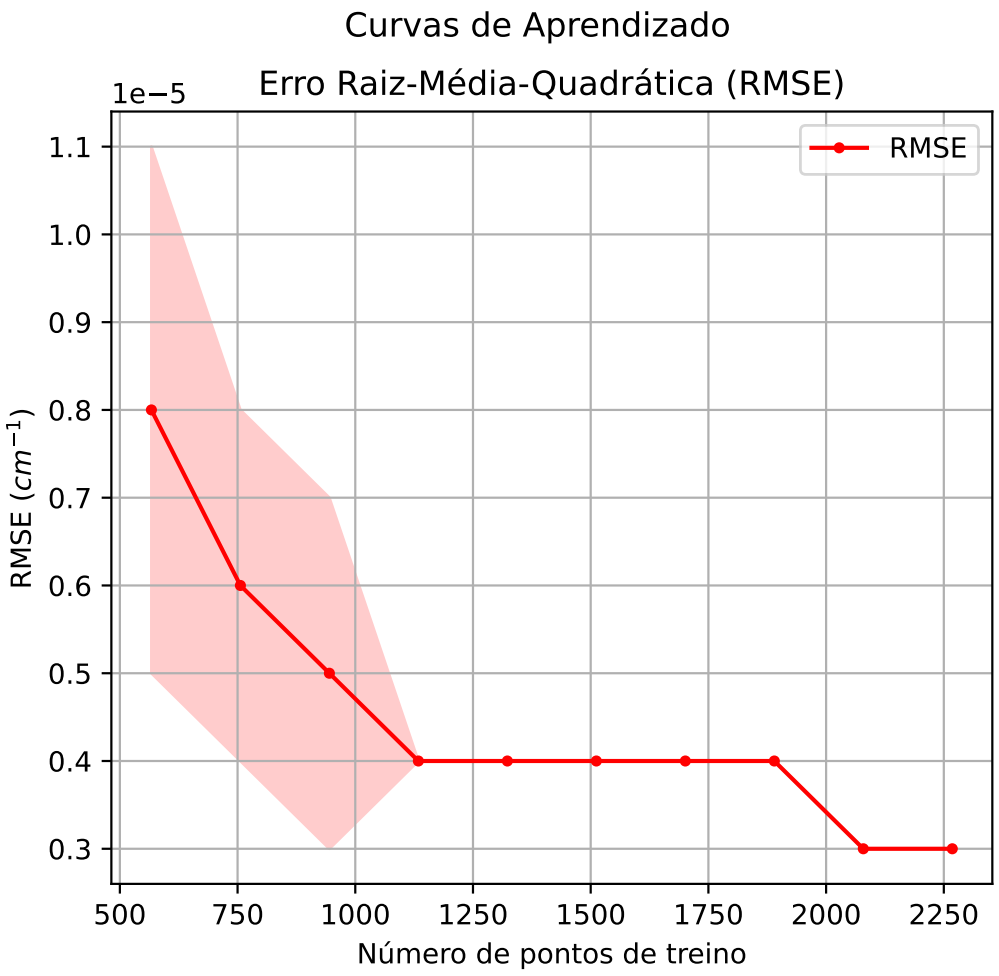
Na Tabela 1, podemos ver o erro RMSE e os coeficientes de correlação do algoritmo tanto na etapa de treino quanto na etapa de teste. Com efeito, os dados produzidos pelo

algoritmo possuem um fator de correlação muito próximo de 1. Além disso, possuem um erro médio quadrático muito pequeno, o que já é considerado acurado o suficiente para cálculos espectroscópicos.

Além do gráfico de correlação, podemos analisar as curvas de aprendizado (LC, do inglês *learning curve*) de um modelo ML. Essas curvas indicam como métricas de erro e tempo de cálculo variam em função do número de dados de treino. A ideia por trás das LCs é ter uma representação visual de quantos pontos são necessários para obter um modelo com acurácia e velocidades certos para diferentes aplicações.

As LCs foram construídas com base em um *ensemble* de algoritmos treinados em conjuntos diversificados de treino e teste. As acurácias de cada algoritmo foram testadas separadamente e uma média foi tomada sobre todo o *ensemble*.

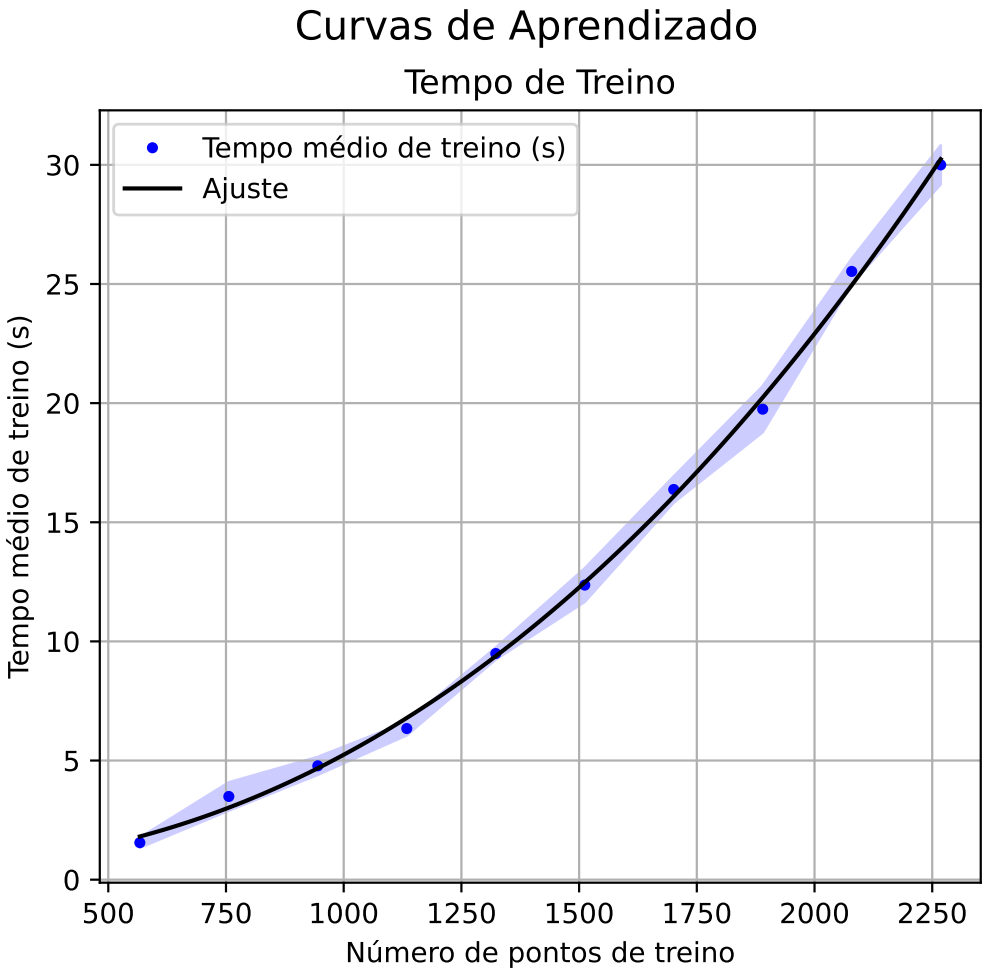
Figura 13 – Erro raiz-médio-quadrático do algoritmo KREG para o sistema $H_2O_2 - Kr$ em função do número de dados de treino.



Na Figura 13, vemos a LC do RMSE do algoritmo KREG em função do número de dados de treino. A área hachurada das curvas nos dá uma margem de incerteza sobre diversos treinos realizados para obter os resultados das LCs. Vemos que a partir de 1125

pontos de treino, temos que o RMSE varia pouco com respeito ao número de pontos de treino, indicando que essa quantidade de pontos já nos dá uma acurácia satisfatória.

Figura 14 – Tempo de treino do algoritmo KREG para o sistema $H_2O_2 - Kr$ em função do número de dados de treino.



Na Figura 14, vemos a LC do tempo médio de treino (em segundos) em função do número de dados de treino. claramente, pode-se notar um crescimento monótono do tempo de treino. Isso é esperado, uma vez que quanto mais pontos utilizamos na etapa de treino, mais tempo é gasto analisando-os.

Tabela 2 – Coeficientes do ajuste quadrático realizado na curva de aprendizado do tempo de treino e seu respectivo erro.

Ajuste:		$y = ax^2 + bx + c$
Coeficientes	Valor	Erro
a	$7,133 \times 10^{-6}$	$5,188 \times 10^{-7}$
b	$-3,609 \times 10^{-3}$	$1,492 \times 10^{-3}$
c	$1,648$	$0,966$

Investigando este comportamento quantitativamente, foi feito um ajuste quadrático do tempo de treino, uma vez que o custo computacional de métodos de kernel cresce com $\mathcal{O}(N^2)$, onde N é o número de pontos de treino (16). O resultado encontra-se na Figura 14 e, na Tabela 2, vemos os dados do ajuste.

Em suma, ao final do processo, obteve-se um modelo treinado para reproduzir qualquer ponto no intervalo no qual ele foi treinado com grau de precisão igual ao do método utilizado em seu treinamento. Ou seja, um modelo de ML serve como uma forma analítica implícita e mais geral, construída com menos tempo computacional necessário.

Aliás, essa diferença de tempo é bastante notável. Cálculos pós-HF para construir a SEP analisada levam cerca de 4 dias para serem computados, enquanto um modelo de ML treinado consegue gerar um número igual ou até maior de pontos em poucos minutos.

Conclusões e Perspectivas

A fim de estudar o desempenho de algoritmos de aprendizado de máquina em sistemas moleculares fracamente ligados, treinamos um algoritmo KREG com base em energias obtidas a nível de cálculo MP4/aug-cc-pVTZ. Com efeito, utilizando o ML na construção de SEPs, foi possível obter resultados comparáveis aos de métodos *ab initio* em menos tempo computacional.

Como visto no capítulo anterior, verificou-se que o modelo KREG é bastante eficiente na construção de SEPs. Os dados por ele produzidos apresentaram a mesma acurácia que aqueles produzidos pelo método MP4, no qual o algoritmo se baseou. Além disso, as SEPs possuem topologia similar o suficiente às SEPs de referência, indicando que o modelo conseguiu reconstruí-la com êxito a partir de um conjunto restrito de dados.

Observando as curvas de aprendizado do modelo, nota-se que ele tem um desempenho aceitável com cerca de 1000 pontos, quantidade certamente menor do que aquela necessária para construir a mesma SEP com o método MP4, que geralmente necessita do triplo desse valor. Contudo, esse número de pontos depende do tipo de sistema, dos graus de liberdade analisados, entre outros fatores.

Verifica-se, ainda, que como o tempo de treino cresce como $\mathcal{O}(N^2)$, é crucial utilizar o menor número possível de pontos de treino. Isso se justifica pelo fato de considerarmos não só o custo computacional do processo de treino do modelo, mas também o custo de cada ponto adicional no nosso conjunto de dados, uma vez que para obter o valor de referência, é necessário realizar um cálculo *ab initio* que pode ser 10^6 vezes mais demorado.

Com relação à outro trabalho similar realizado (24), pode-se afirmar que os métodos de kernel superam a otimização de funcionais da densidade em questão de acurácia e tempo computacional. Porém, um método não invalida automaticamente o outro, uma vez que estudos já estão sendo feitos para utilizar técnicas de ML na parametrização de funcionais novos na literatura (38).

Por fim, é seguro afirmar que o modelo de ML proposto cumpriu seu objetivos gerais. Todavia, há ainda algumas propriedades que podem ser futuramente estudadas a partir das SEPs (e.g. constantes espectroscópicas) e comparados com valores experimentais correspondentes. Além disso, o treinamento de outros algoritmos de ML e a avaliação de outros gases nobres além do Kr em sistemas do tipo $H_2O_2 - Ng$ ($Ng = He, Ne, Ar, Xe$) por meio da metodologia apresentada neste trabalho pode contribuir para o desenvolvimento do ML na Física Molecular.

Referências

1 SEJNOWSKI, T. J. *The deep learning revolution*. London, England: MIT Press, 2018. (The MIT Press). Citado na página 27.

2 COPELAND, B. *Alan Turing and the beginning of AI*. 2020. <https://tinyurl.com/a42uy5j4>. Citado na página 27.

3 MANYIKA, J. et al. *Big Data: The next frontier for innovation, competition, and productivity*. 2011. <https://tinyurl.com/4xk4k8e3>. Acessada em 14/08/2021. Citado na página 27.

4 LECUN, Y.; CORTES, C.; BURGES, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, v. 2, 2010. Citado na página 27.

5 MEINL, R. *Recommender Systems: The Most Valuable Application of Machine Learning (Part 1)*. 2020. <https://tinyurl.com/za24emrm>. Acessada em 14/08/2021. Citado na página 27.

6 MEINL, R. *Recommender Systems: The Most Valuable Application of Machine Learning (Part 2)*. 2020. <https://tinyurl.com/3eup7w62>. Acessada em 14/08/2021. Citado na página 27.

7 ALEXANDRU, G. *Conversational AI: Intelligent Virtual Assistants and the road ahead*. 2020. <https://tinyurl.com/3szctyc5>. Acessada em 14/08/2021. Citado na página 27.

8 SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015. Citado na página 27.

9 CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, p. 273–297, 1995. Citado 2 vezes nas páginas 27 e 52.

10 PATIL, A. et al. *Robust deep learning for emulating turbulent viscosities*. 2021. Citado na página 27.

11 PAREKH, V. S. et al. Multiparametric deep learning tissue signatures for a radiological biomarker of breast cancer: Preliminary results. *Medical Physics*, Wiley, v. 47, n. 1, p. 75–88, Nov 2019. ISSN 2473-4209. Disponível em: <http://dx.doi.org/10.1002/mp.13849>. Citado na página 27.

12 KRAJNáK, V.; NAIK, S.; WIGGINS, S. *Predicting trajectory behaviour via machine-learned invariant manifolds*. 2021. Citado na página 27.

13 ELLIS, J. A. et al. Accelerating finite-temperature kohn-sham density functional theory with deep neural networks. *Physical Review B*, American Physical Society (APS), v. 104, n. 3, Jul 2021. ISSN 2469-9969. Disponível em: <http://dx.doi.org/10.1103/PhysRevB.104.035120>. Citado na página 27.

14 CHMIELA, S. et al. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, v. 9, n. 3887, 2018. Citado na página 27.

15 WILLE, S. et al. An experimentally validated neural-network potential energy surface for h-atom on free-standing graphene in full dimensionality. *Physical Chemistry Chemical Physics*, Royal Society of Chemistry (RSC), v. 22, n. 45, p. 26113–26120, 2020. ISSN 1463-9084. Citado na página 27.

16 PINHEIRO, M. et al. Choosing the right molecular machine learning potential. *Chem. Sci.*, The Royal Society of Chemistry, v. 12, p. 14396–14413, 2021. Disponível em: <http://dx.doi.org/10.1039/D1SC03564A>. Citado 3 vezes nas páginas 27, 57 e 68.

17 SZABO, A.; OSTLUND, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. [S.l.]: Dover Publications Inc., 1996. Citado 3 vezes nas páginas 34, 37 e 38.

18 BORN, M.; OPPENHEIMER, R. Zur quantentheorie der molekeln. *Annalen der Physik*, v. 389, n. 20, p. 457–484, 1927. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19273892002>. Citado na página 34.

19 LEWARS, E. G. *Computational Chemistry - Introduction to the Theory and Applications of Molecular and Quantum Mechanics*. [S.l.]: Springer, 2016. Citado na página 35.

20 EDSANVILLE. 2005. Simplified Hartree–Fock procedural flowchart Created by me. Disponível em: <https://commons.wikimedia.org/wiki/File:Hartree-Fock.png>. Citado na página 39.

21 Møller, C.; Plesset, M. S. Note on an Approximation Treatment for Many-Electron Systems. *Physical Review*, v. 46, n. 7, p. 618–622, out. 1934. Citado na página 41.

22 KRISHNAN, R.; POPLE, J. A. Approximate fourth-order perturbation theory of the electron correlation energy. *International Journal of Quantum Chemistry*, 1978. Citado na página 42.

23 RONCARATTI, L. et al. Chirality of weakly bound complexes: The potential energy surfaces for the hydrogen-peroxide-noble-gas interactions. *The Journal of chemical physics*, v. 141, p. 134309, 2014. Citado 3 vezes nas páginas 43, 57 e 62.

24 BISPO, M. de O.; FILHO, D. A. da S. Long-range parameter optimization for a better description of potential energy surfaces using density functional theory. *Journal of Molecular Modeling*, Springer Science and Business Media LLC, v. 28, n. 5, abr. 2022. Disponível em: <https://doi.org/10.1007/s00894-022-05083-1>. Citado 2 vezes nas páginas 43 e 69.

25 LIU, X.; MEIJER, G.; PéREZ-RíOS, J. A data-driven approach to determine dipole moments of diatomic molecules. *Phys. Chem. Chem. Phys.*, The Royal Society of Chemistry, v. 22, p. 24191–24200, 2020. Disponível em: <http://dx.doi.org/10.1039/D0CP03810E>. Citado 2 vezes nas páginas 48 e 50.

26 HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, [Taylor Francis, Ltd., American Statistical Association, American Society for Quality], v. 42, n. 1, p. 80–86, 2000. ISSN 00401706. Citado na página 50.

27 HOFMANN, T.; SCHöLKOPF, B.; SMOLA, A. J. Kernel methods in machine learning. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 36, n. 3, jun 2008. Citado na página 52.

28 DRAL, P. O. Quantum chemistry assisted by machine learning. In: *Chemical Physics and Quantum Chemistry*. Elsevier, 2020. p. 291–324. Disponível em: <https://doi.org/10.1016/bs.aiq.2020.05.002>. Citado 2 vezes nas páginas 52 e 57.

29 HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. Springer New York, 2009. Disponível em: <https://doi.org/10.1007/978-0-387-84858-7>. Citado na página 55.

30 GUFOSOWA. *Diagram of k-fold cross-validation.* 2019. <https://wikipedia.org/wiki/File:K-fold_cross_validation_EN.svg?lang=en>. Acessada em 27/08/2022. Citado na página 56.

31 FRISCH, M. J. et al. *Gaussian~16 Revision B.01*. 2016. Gaussian Inc. Wallingford CT. Citado na página 57.

32 BOYS, S.; BERNARDI, F. The calculation of small molecular interactions by the differences of separate total energies. some procedures with reduced errors. *Molecular Physics*, Taylor Francis, v. 19, n. 4, p. 553–566, 1970. Citado na página 57.

33 DRAL, P. O. et al. Mlatom 2: An integrative platform for atomistic machine learning. *Topics in Current Chemistry*, v. 379, 2021. Citado na página 57.

34 DRAL, P. O. et al. Structure-based sampling and self-correcting machine learning for accurate calculations of potential energy surfaces and vibrational levels. *The Journal of Chemical Physics*, AIP Publishing, v. 146, n. 24, p. 244108, jun. 2017. Disponível em: <https://doi.org/10.1063/1.4989536>. Citado 2 vezes nas páginas 57 e 58.

35 van der Walt, S.; Colbert, S. C.; Varoquaux, G. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, v. 13, n. 2, p. 22–30, 2011. Citado na página 57.

36 HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. Citado na página 58.

37 BISPO, M. O. 2022. M-obispo/MLTools: Tools for MLatom, a package for atomistic simulations with machine learning. Disponível em: <https://github.com/m-obispo/MLTools>. Citado na página 58.

38 PALOS, E. et al. Density functional theory of water with the machine-learned DM21 functional. *The Journal of Chemical Physics*, AIP Publishing, v. 156, n. 16, p. 161103, abr. 2022. Disponível em: <https://doi.org/10.1063/5.0090862>. Citado na página 69.