

ALTERANDO COORDENADAS DE MOLÉCULAS E NANOESTRUTURAS EM GIRO CONTROLADO POR SOFTWARE

André Flores¹; Mirkos Ortiz Martins²

RESUMO

Quando se configura um sistema atômico para se fazer simulação computacional, envolvendo moléculas e nanoestruturas, o posicionamento tridimensional dos átomos é de fundamental importância. Uma coordenada de átomo pode corromper todo um cálculo de interação que pode levar meses para ser concluído. Nesse cenário, a construção de arquivos que descrevem o posicionamento tridimensional de sistemas atômicos precisa ser feito de forma acurada quanto ao posicionamento dos átomos e moléculas. Esse trabalho se propôs a automatizar, através do uso de um algoritmo, a manipulação das coordenadas dos sistemas a nível atômico no sentido de controlar o giro de uma estrutura em relação às demais constantes em um arquivo de entrada para simulação computacional. O software desenvolvido foi escrito em linguagem de programação Python e as coordenadas lidas como entrada estavam em um arquivo no formato xyz. A saída da execução também é armazenada nesse formato. Os arquivos resultantes dos giros foram então analisados em software de visualização e confirmados com precisão do giro esperado.

Palavras-chave: Bioinformática, Matemática, Trigonometria, Python, Posicionamento tridimensional.

Eixo Temático: Tecnologia, Inovação e Desenvolvimento Sustentável (TIDS)

1. INTRODUÇÃO

A simulação computacional do ponto de vista físico pode ser definida como qualquer aplicação computacional que envolva a solução numérica de um sistema de equações diferenciais parciais (ROY, 2005). Existe um outro formalismo na descrição dos sistemas especificados para a simulação que são os modelos descritos em eventos discretos (ZEIGLER, MUZY e KOFMAN, 2018). Para a simulação computacional de elementos na escala atômica e molecular, utiliza-se a caracterização dos átomos formadores dessas estruturas em representações que os

¹ Autor/Apresentador – PPGNANO - UFN e-mail(s).

² Demais Autores – PPGNANO – UFN mirkos@prof.ufn.edu.br

definem tridimensionalmente. As simulações computacionais são executadas no sentido de compreender as propriedades estruturais das moléculas e as interações entre elas. Para representar essas estruturas foi criado o formato xyz para armazenar computacionalmente as informações de posicionamento atômico.

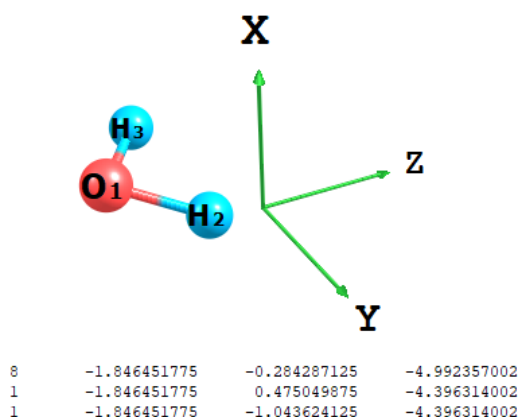


Figura 1 - Molécula de água (H₂O) representada no formato xyz. Fonte: autor

1.1 O FORMATO XYZ

O formato do arquivo xyz é composto basicamente de informações sobre cada átomo do sistema em cada linha e distribuído por colunas, cada coluna representando uma informação. No formato, a primeira linha é a exceção e constitui no número de átomos constantes no sistema, sendo a linha seguinte (segunda linha) uma reserva para observação extra e a partir da terceira linha então são seguidos os padrões de descrição.

A primeira coluna é preenchida pelo nome do átomo ou seu número atômico, conforme pode ser observado na Figura 1, onde uma molécula de água (H₂O) está representada pelas três linhas, uma para cada átomo. O átomo de oxigênio, de número atômico 8 é representado na primeira linha e as duas linhas seguintes são mostradas as coordenadas dos dois átomos de hidrogênio.

A segunda coluna representa o valor da coordenada no eixo x, a terceira coluna mostra o valor da coordenada no eixo y e a quarta coluna a coordenada no eixo z. Com esses três eixos é possível representar tridimensionalmente a estrutura de entrada para a simulação.

1.2 JUSTIFICATIVA

Esse trabalho se insere na importância da modelagem e simulação molecular, desenvolvidas no âmbito do programa de pós-graduação em Nanociências da Universidade Franciscana (UFN). A montagem da configuração inicial de sistemas no nível atômico, através da modelagem tridimensional dos átomos envolvidos no estudo, requer uma precisão e controle bastante apurado. Quando estudados esses sistemas, suas configurações são avaliadas conforme posição das estruturas atômicas, entre uma e outra, a distância e qual a natureza dos átomos que estão em maior interação.

Nessas configurações, a orientação das estruturas moleculares sofre alterações para que a simulação avalie diferentes posicionamentos e possibilite uma discussão a respeito dos resultados.

Na Figura 2 pode ser observada uma sequência de investigação onde (da esquerda para a direita) inicia-se construindo o sistema de interesse em um arquivo contendo as coordenadas de todos os átomos.

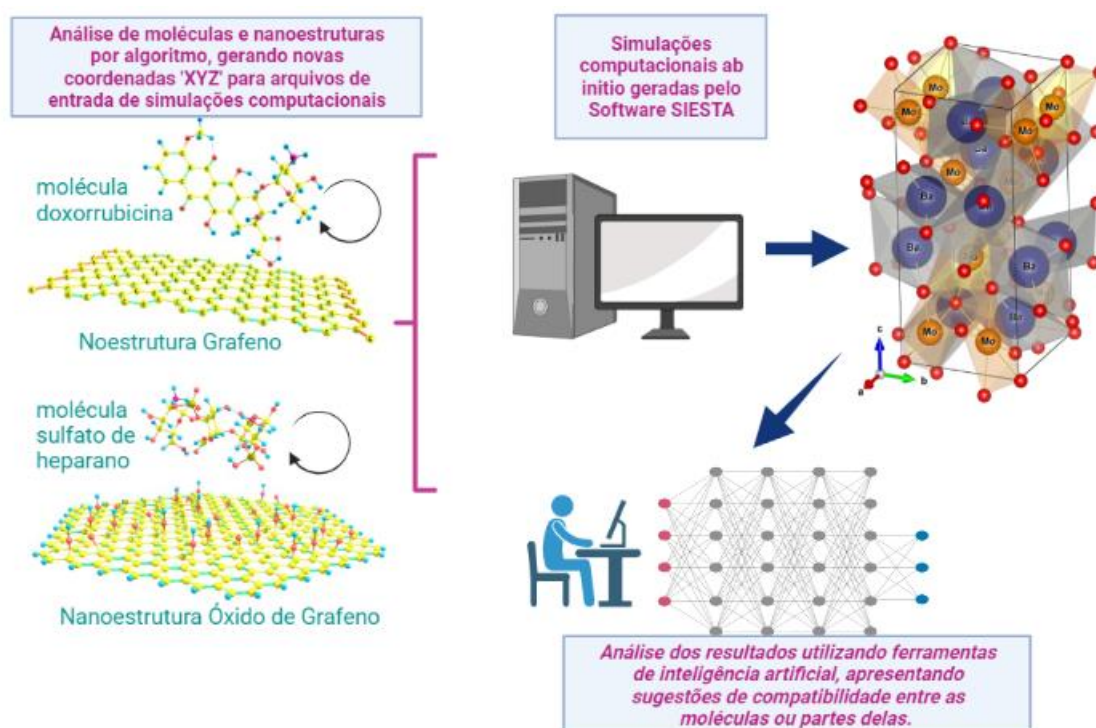


Figura 2 - Sequência dos processos no uso das coordenadas em uma simulação computacional. Fonte: autor

É mostrado que as moléculas de dexorubicina e sulfato de heparano são giradas em um eixo, enquanto a nanoestrutura de grafeno e óxido de grafeno continuam imóveis tridimensionalmente.

Para cada posicionamento das moléculas que giram, é criado um arquivo diferente (em extensão xyz).

Na sequência, cada arquivo desses arquivos xyz – chamados de *input* – são fornecidos à um software de simulação (no caso da figura, eles são inseridos em um software de simulação ab initio chamado SIESTA (SOLER, ARTACHO, *et al.*, 2002)) e os resultados analisados por um conjunto de ferramentas de inteligência artificial para extração de características importantes na interação do sistema.

Para a movimentação de giro da molécula, foi escrito um código computacional para calcular os próximos passos das coordenadas dos átomos nos arquivos de *input*.

2. METODOLOGIA

O código computacional – programa – para efetuar o giro das estruturas em arquivos xyz foi escrito em linguagem Python, em ambiente notebook aplicando técnicas de trigonometria com o uso da biblioteca Pandas.

Todo o trabalho se resume a resolver equações de rotação e as coordenadas dos átomos se limitam em 2 dimensões, levando em consideração que uma das dimensões é fixa baseada na posição da superfície da segunda molécula em interação no sistema.

A rotação portanto obedece um ângulo α , conforme mostrado na Figura 3. O sentido da rotação é anti-horário de forma que um átomo (Atomo na figura) seja deslocado para uma segunda posição (Atomo').

Para esse deslocamento, utiliza-se a forma de rotação angular, conforme (WATARI, 2003), visto nas Equações para x' e y' :

$$\begin{cases} x' = x \cos \theta - y \sin \theta \\ y' = x \sin \theta + y \cos \theta \end{cases} \quad (1)$$

Quando calculados, então são fornecidas as novas coordenadas para os átomos do sistema e então armazenados em um arquivo formato xyz com a rotação efetuada.

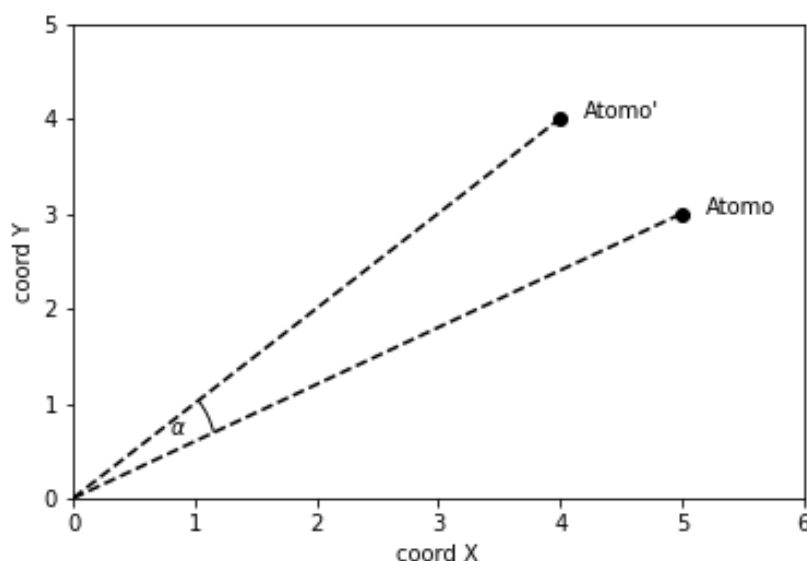


Figura 3 - Posicionamento de um átomo em rotação conforme um ângulo pré-definido. Fonte: autor

3. RESULTADOS E DISCUSSÕES

A construção do código computacional, implementada em linguagem Python, consistiu em ler o arquivo de entrada (formato xyz), armazenar as coordenadas e calcular a rotação, escrevendo como saída outro arquivo em formato xyz com a rotação já efetuada.

Na Figura 4 é possível observar a função de rotação, implementando a Equação 1, onde as coordenadas x e y resultantes são calculadas nas linhas 13 e 14 do código. Um detalhe importante de ser salientado é que o ângulo θ da rotação é informado em graus e convertido para radianos (linha 8 do código) devido aos métodos de seno e cosseno serem implementados dessa forma no módulo *math* da linguagem Python.

O resultado dessa execução é a construção de uma lista de coordenadas que serão escritas em um arquivo de saída, coordenadas essas servindo de base para futuras simulações em diferentes softwares de cálculo, seja de visualização, de mecânica clássica ou mecânica quântica.

```
4 #-.....
5 #funcao para rotacionar os atomos em torno de um atomo central
6 def rotacao(atomoCentral, atomoRecalcular,angulo):
7     atomo=[]
8     angulo = math.radians(angulo)
9     x = float(atomoRecalcular[1])-float(atomoCentral[1])
10    y = float(atomoRecalcular[2])-float(atomoCentral[2])
11    z = float(atomoRecalcular[3])
12
13    x_resultante = x*math.cos(angulo)-y*math.sin(angulo)
14    y_resultante = x*math.sin(angulo)+y*math.cos(angulo)
15
16    atomo.append(atomoRecalcular[0])#tipo atômico
17    atomo.append(x_resultante+float(atomoCentral[1]))
18    atomo.append(y_resultante+float(atomoCentral[2]))
19    atomo.append(z)
20
21    return atomo
22 #-.....
```

Figura 4 - codificação em Python da função de rotação bidimensional.

4. CONCLUSÃO

Esse trabalho foi de suma importância para a automatização na construção de diferentes arquivos de entrada, para simulação no laboratório de simulação de moléculas em nanoestruturas (LASIMON-UFN) de modo preciso e rápido. O uso de uma lógica computacional diminui os riscos de manipulação átomo a átomo na construção de arquivos de configuração utilizando o formato xyz.

REFERÊNCIAS

- ROY, C. J. Review of code and solution verification procedures for computational simulation. **Journal of Computational Physics**, v. 205, n. 1, p. 131-156, 2005.
- SOLER, J. M. et al. The SIESTA method for ab initio order-N materials simulation. **Journal of Physics: Condensed Matter**, p. 2745, 2002.
- ZEIGLER, B. P.; MUZY, A.; KOFMAN,. **Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations**. 3. ed. London: Academic Press, v. 1, 2018.