https://eventos.utfpr.edu.br//sicite/sicite2019

O uso da linguagem de programação Python para o estudo de armadilhas de elétrons

Python programming language used to study trapped-electrons

Vitor Sangali Strabelli
vitorstrabelli@hotmail.com
Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Luciara Indrusiak Weiss
luciara@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

Vanderlei Aparecido de Lima
valima@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Pato Branco, Paraná, Brasil

RESUMO

O estudo aprofundado da matéria tem revelado interessantes comportamentos da natureza que, entretanto, não são de fácil compreensão. Para facilitar o aprendizado e estimular a busca por conhecimento, utilizou-se a linguagem de programação *Python* e a ferramenta *PyCharm (JetBrains)*para desenvolver um código capaz de calcular as energias de um elétron em um poço de potencial infinito, com a finalidade de criar um jogo. Obteve-se uma configuração que calculava as energias de um elétron em seus diferentes níveis, para armadilhas em uma, duas ou três dimensões. Também foi programado o cálculo do estado fundamental do poço unidimensional, dada uma largura e um número de elétrons confinados. Os códigos foram desenvolvidos de forma a gerar um jogo de adivinhação que teste o conhecimento na área de Física.

PALAVRAS-CHAVE: Elétron. Quântico. Python.

Recebido: 19 ago. 2019.
Aprovado: 01 out. 2019.

ABSTRACT

In-depth study of the matter has revealed interesting behaviors in nature, however, they are not easy to understand. To facilitate learning and stimulate the search for knowledge, the Python programming language and the PyCharm (JetBrains) tool were used to develop code that can calculate the energy of an electron in an infinite potential well for the purpose of creating a game. A configuration was obtained that calculated the energies of an electron at its different levels for traps in one, two or three dimensions. It was also programmed to calculate the ground state of the one-dimensional well, given a width and a number of confined electrons. The codes were developed to generate a guessing game that tests knowledge in the physics area.

KEYWORDS: Electron. Quantic. Python.

INTRODUÇÃO

O estudo das armadilhas de elétrons está presente em diversos cursos relacionados à área da Física. Assim sendo, esse estudo faz-se importante, pois o curso de Física é frequentado por um vasto número de acadêmicos de diversos cursos como: Curso de química, de Física e de Engenharias. O projeto de iniciação científica aqui descrito visa desenvolver uma maneira mais simples e didática a fim de promover o aprendizado sobre níveis energéticos de elétrons.

Em meio a tantas linguagens de programação possíveis, escolheu-se a *Python* devido a sua grande versatilidade e facilidade de uso, uma vez que apresenta funções bastante variadas com uma escrita compacta.

MATERIAL E MÉTODOS

Ao longo de todo o desenvolvimento deste projeto, utilizou-se o programa *Python* 3.7 e o aplicativo *PyCharm*, a fim de facilitar a escrita do código e aumentar a eficiência das atividades.

No primeiro momento, realizou-se o estudo da linguagem de programação *Python* através de plataformas de cursos online (plataforma Udemy), a fim de promover uma capacitação para a criação de um código que atendesse às necessidades do projeto.

Nesta etapa foram aprendidas na plataforma online estruturas de configuração capazes de simular jogos de formato simples, tais como o jogo de adivinhação e o jogo da forca, os quais serviriam de base de maneira conjunta com o conhecimento teórico sobre armadilhas, para a construção de um jogo mais elaborado envolvendo tal área.

A partir de então estudaram-se as características das armadilhas de elétrons utilizando-se, como principal referência, o livro Fundamentos da Física (HALLIDAY, RESNICK, WALKER, 2009).

Em seguida foram construídos códigos com a linguagem Python para armadilhas de elétrons unidimensionais, bidimensionais e tridimensionais de potencial infinito.

RESULTADOS E DISCUSSÃO

O código referente a um poço infinito de uma dimensão apresenta funções capazes de fornecer a energia do elétron em determinado nível, de acordo com o número quântico, como mostrado na Figura 1.

Nesta configuração o trecho “self._energia_nível = (self.__constante)*(nx ** 2)/(self._largx)**2” é análogo a equação $E_n = (h^2 / 8.m_{elétron}).(n^2/L)$.

Sendo que a segunda seção utiliza da mesma escrita, apenas apresentando artifícios que permitem o cálculo simultâneo de valores de energias para diferentes níveis.

Figura 1- Código para poço unidimensional

```
# dado um nx, calcula a energia do nivel nx
    def energia_nivel(self, nx):
        self.__energia_nivel = (self.__constante) * ((nx ** 2)/((self.__largx)**2))
        return self.__energia_nivel

# dado um nx máximo, calcula e armazena as energias de todos os níveis até esse valor
    def energias_niveis(self, nxmax):
        self.__energias_niveis = []
        for nx in range(1, nxmax + 1):
            valor = (self.__constante) * ((nx /(self.__largx)) ** 2)
            self.__energias_niveis.append((nx, valor))
        return self.__energias_niveis
```

Fonte: Autoria própria (2019)

Atribuindo os valores às variáveis nx, largx e aplicando as funções é possível obter os resultados referentes a constante. Para o número quântico 1 tem-se o valor da constante ($1.527369927651773.10^{-39}$) e para o 2 nota-se um resultado referente ao quádruplo da constante, o que faz sentido, uma vez que o fator n é elevado ao quadrado segundo a função.

Figura 2- Resultados das funções de energia

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
>>> from poco_infinito_uni import poco_infinito
>>> poco = poco_infinito(1)
Construindo objeto ...<poco_infinito_uni.poco_infinito object at 0x043DFFF0>
>>> poco.energia_nivel(1)
1.527369927651773e-39
>>> poco.energias_niveis(2)
[(1, 1.527369927651773e-39), (2, 6.109479710607092e-39)]
```

Fonte: Autoria própria (2019)

Pode-se, então, calcular a energia do estado fundamental da armadilha para um número arbitrário de partículas, com o código mostrado na Figura-3.

Figura 3- Código para estado fundamental

```
def estado_fundamental(self, numpart):

    if numpart % 2 == 0:
        numniv = numpart/2
    else:
        numniv = (numpart + 1)/2

    numniv = int(numniv)
    print(numniv)

    energia_ef = 0
    numpartrest = numpart

    for nivel in range(1, numniv+1):
        for ocupaniv in range(1, 3):
            if numpartrest >= 1:
                valor = (self.__constante) * (nivel/self._largx) ** 2
                energia_ef += valor
                numpartrest = numpartrest-1
#                print(numpartrest)
            else:
                break
    print("A energia do estado fundamental para {} partículas é {}".format(numpart, energia_ef))
```

Fonte: Autoria própria (2019)

Em “def estado_fundamental(self, numpart):” é definido o número de partículas que se deseja analisar, a função seguinte determina o número de níveis (numniv) da armadilha. Em seguida, define-se a energia inicial (energia_ef = 0) e o elemento “numpartrest” como sendo igual ao número de partículas escolhido inicialmente.

Por fim, a última parte consiste em calcular o valor da energia para cada partícula utilizando uma função similar a apresentada anteriormente “valor = (self.__constante)*(nivel/self._largx)**2”, assim que o valor da energia de uma partícula é calculado, ele é somado à energia inicial e o processo é repetido até que todas as partículas tenham sido analisadas, ou seja, quando “numpartrest” = 0, neste momento a função é encerrada e há um *print* dos resultados na forma de uma frase “A energia do estado fundamental para ... partículas é ...”.

Como calculado anteriormente, para a largura (“self._largx”) com o valor 1, a energia de uma única partícula com n igual a 1 tem o mesmo valor da constante ($1.527369927651773.10^{-39}$). Já em um estado fundamental, considerando tal largura e o número de partículas (numpart) com o valor 2, encontra-se um resultado de $3.05473985503546.10^{-39}$ como mostrado pela figura 4, neste caso temos duas partículas no nível 1, dessa maneira faz sentido que o valor total da energia do estado fundamental seja o dobro do valor da constante, que é exatamente o que foi encontrado.

Figura 4- Resultado de estado fundamental

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
>>> from poco_infinito_uni import poco_infinito
>>> poco = poco_infinito(1)
Construindo objeto ...<poco_infinito_uni.poco_infinito object at 0x03EEBFF0>
>>> poco.estado_fundamental(2)
A energia do estado fundamental para 2 partículas é 3.054739855303546e-39
```

Fonte: Autoria própria (2019)

Além dos resultados parciais apresentados acima, foram desenvolvidos: os códigos para calcular a energia de um elétron em poços de potencial infinito de duas e três dimensões, e o código para calcular a energia de estados excitados (ainda não finalizados); a sequência principal de passos para um jogo envolvendo esse tipo de armadilha.

CONCLUSÃO

Conclui-se que os resultados do estudo foram produtivos, gerando um código que facilita a determinação de energias de estados quânticos de elétrons em poços de potencial infinito. Código esse que é plenamente capaz de ser integrados a outras funções, as quais poderiam realizar cálculos de situações mais diversas, como um poço de potencial finito.

Também é possível notar que, graças à versatilidade do programa, pode-se desenvolver um código que simule um jogo de adivinhação, de modo a facilitar o aprendizado e torná-lo mais atrativo.

AGRADECIMENTOS

Gostaria de agradecer à instituição Universidade Tecnológica Federal do Paraná, campus Pato Branco pela oportunidade. Agradeço também à Orientadora Luciara Indrusiak Weiss e ao Professor Vanderlei Aparecido de Lima.

REFERÊNCIAS

HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentos da Física**. 6. ed. V.4. Rio de Janeiro, RJ, 2008.