

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

EDUARDO VINICIUS GALLE

**PROPOSTA DE UM ALGORITMO QUÂNTICO PARA A
RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS NÃO-
LINEARES BASEADO NO MÉTODO DE ADOMIAN**

Dissertação de Mestrado

Porto Alegre

2019

EDUARDO VINICIUS GALLE

**PROPOSTA DE UM ALGORITMO QUÂNTICO PARA A
RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS NÃO-
LINEARES BASEADO NO MÉTODO DE ADOMIAN**

Dissertação de mestrado apresentada ao
Programa de Pós-Graduação em Física, da
Universidade Federal do Rio Grande do Sul,
como parte dos requisitos para a obtenção do
título de Mestre em Física.

ORIENTADOR: SANDRA D. PRADO

COLABORADOR: BARDO E. J. BODMANN

Porto Alegre

2019

DEDICATÓRIA

Dedico este trabalho a Deus por fornecer a paz necessária para superar as dificuldades e aos meus pais por fornecer o aporte material e familiar necessários para enfrentar essa empreitada.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Física, PPGFIS, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Aos colegas do PPGFIS pelo seu auxílio nas tarefas desenvolvidas durante o curso e apoio na revisão deste trabalho.

À CNPq pela provisão da bolsa de mestrado.

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil

RESUMO

O método de Adomian é usado na solução de equações diferenciais não lineares com resíduo, em áreas das engenharias. Nesse trabalho, usaremos o método para transformar a Equação de Schrödinger não linear em um sistema de equações lineares para, então, estudar a construção de um algoritmo quântico de Adomian. Como **método**, realizamos uma revisão bibliográfica do método de Adomian e utilizamos Matlab para aplicar o método, no caso de um sistema clássico. Já quanto a versão quântica do método, fizemos uma revisão bibliográfica de algoritmos quânticos e identificamos possibilidades de usos de algoritmos quânticos para equações diferenciais lineares que podem ser utilizados, após a aplicação do método de Adomian. Nesse particular, discutimos abordagens que poderão ser úteis em trabalhos futuros. **Resultados** os resíduos do método de Adomian são muito variáveis em ordem de grandeza, contudo, em certas condições, o método apresenta soluções muito próximas da solução real. A ideia de algoritmo quântico proposto neste trabalho precisa ser aprofundada em estudos futuros para ser testada de forma apropriada, dependendo do sistema que se queira trabalhar.

PALAVRAS-CHAVE: Adomian. Método de Adomian. Método de decomposição de Adomian. Equações não lineares. Não linearidade. Algoritmos quânticos.

ABSTRACT

The Adomian method is often used to solve nonlinear differential equations mainly in engineering problems. In this work, we use the method to obtain a set of linear equations from the nonlinear Schrödinger equation in order to study a quantum version of the Adomian algorithm. Regarding the methodology we did a bibliographic review of the Adomian method and used Matlab to apply the method in a classical system. As for the quantum version, we did a bibliographical review of quantum algorithms and we have identified a certain number of quantum algorithms for linear differential equations that can be used after the application of the Adomian method. In this regard, we have developed an approach that may be useful in future work. Results The residues of the Adomian method can vary a lot in terms of the order of magnitude, but under certain conditions, the method presents solutions very close to the the real solution. The idea of a quantum Adomian algorithm proposed in this work needs to be deepened in future studies, so that it can be properly tested, depending on the system where it will be applied.

KEYWORDS: Adomian. Adomian method. Method of Adomian decomposition. Nonlinear equations. Nonlinearity. Quantum algorithms.

Sumário

1 INTRODUÇÃO	10
2 O MÉTODO DE DECOMPOSIÇÃO DE ADOMIAN	12
2.1 INTRODUÇÃO	12
2.2 O MÉTODO	12
2.3 IMPORTÂNCIA E APLICAÇÕES.....	15
2.4 CONVERGÊNCIA	15
3 APLICAÇÃO DE ADOMIAN	16
3.1 INTRODUÇÃO	16
3.2 A NÃO LINEARIDADE KERR	16
3.3 A EQUAÇÃO DE SCHROEDINGER NÃO LINEAR	16
3.3.1 DESCRIÇÃO DO POTENCIAL HARMÔNICO.....	16
3.3.2 SOLUÇÕES DE $\varphi_{0,n}$	18
3.3.2.1 Solução $\varphi_{0,0}$	18
3.3.2.2 Solução $\varphi_{0,1}$	19
3.3.2.3 Solução $\varphi_{0,2}$	20
3.3.2.4 Passo a passo para soluções	22
3.3.2.5 Alguns resultados.....	22
3.3.3 A EQUAÇÃO DE SCHROEDINGER NÃO LINEAR PARA $V=0$	27
3.3.3.1 Inicialização.....	27
3.3.3.2 primeira recursão	27
3.3.3.3 Segunda recursão.....	28
3.3.4 SOLUÇÕES DE $\varphi_{1,n}$	32
3.3.4.1 Solução $\varphi_{1,0}$	32
3.3.4.2 Solução $\varphi_{1,1}$	32
3.3.4.3 Solução $\varphi_{1,2}$	33
3.3.4.4 $\varphi_{1,3}$ e demais soluções.....	34
4 ALGORITMOS	36
4.1 CONCEITOS	36
4.1.1 ALGORITMO	36
4.1.2 COMPLEXIDADE DE UM ALGORITMO	36
4.1.3 NOTAÇÃO “o”	37
4.2 ALGORITMO QUÂNTICO	37
4.2.1 DIFERENÇA ENTRE ALGORITMO CLÁSSICO E ALGORITMO QUÂNTICO	38
4.2.2 PARADIGMA DOS PROBLEMAS ALGÉBRICOS E DA TEORIA DOS NÚMEROS	38
4.2.3 PARADIGMA ORACULAR	39
4.2.4 PARADIGMA DA SIMULAÇÃO	39
4.3 SUGESTÕES DE ALGORITMO QUÂNTICO.....	39
4.3.1 ESBOÇO: APLICAR FOURIER NA EQUAÇÃO DE RECURSÃO E APÓS ALGORITMO QUÂNTICO.	39
4.3.2 ESBOÇO: APLICAR ALGORITMO QUÂNTICO PARA ENCONTRAR AS RAÍZES DE POLINÔMIOS NA NORMALIZAÇÃO.	40
4.3.3 ESBOÇO: APLICAR O ALGORITMO QUÂNTICO PARA SOLUCIONAR UM SISTEMA DE EQUAÇÕES DIFERENCIAIS.....	41
4.3.3.1 Sistema de Equações.....	41
5 CONCLUSÃO	43
5.1 OBJETIVOS ALCANÇADOS.....	43
5.2 DIFICULDADES	43
5.2.1 APLICAÇÃO DO MÉTODO DE ADOMIAN	43
5.2.2 ALGORITMOS QUÂNTICOS.....	44
5.3 SUGESTÕES DE ESTUDOS FUTUROS.....	45
5.3.1 APLICAÇÃO DO MÉTODO	45
5.3.1 ALGORITMOS QUÂNTICOS.....	45

REFERÊNCIAS	46
APÊNDICE 1: MEMORIAL DESCRITIVO DO PRIMEIRO CÓDIGO DE MATLAB	48
APÊNDICE 2: MEMORIAL DESCRITIVO DO SEGUNDO CÓDIGO DE MATLAB	49
ANÁLISE DE RESÍDUO	49
APENDICE 3: CÓDIGOS	50
4.1 – CÓDIGO 1 – UNIDO	50
4.2 – CÓDIGO 2 – ANÁLISE DE RESÍDUO	54

LISTA DE TABELAS

Tabela 1	Varição do fator não linear $E=E_0=1$	23
Tabela 2	Varição das constantes $E=E_0$, com $C=0,1$	23
Tabela 3	Varição da constante E , com $E_0=1$ e com $C=0,1$	25
Tabela 4	Varição da constante E_0 , com $E=1$ e com $C=0,1$	26
Tabela 5	Varição de x , com $E=E_0 / x$, com $C=0,1$ e com $E_0=1$	26
Tabela 6	Contribuição do termo não-homogêneo $\psi_0\psi_0 * \psi_1$	30
Tabela 7	- Contribuição do termo não-homogêneo $\psi_0^2\psi_1 *$	30
Tabela 8	Contribuição do termo não-homogêneo $\psi_0 * \psi_{12}$	30
Tabela 9	Contribuição do termo não-homogêneo $\psi_0\psi_1 * \psi_1$	30
Tabela 10	Contribuição do termo não-homogêneo $\psi_{12}\psi_1 *$	30
Tabela 11	Constantes das exponenciais de ψ_2	31
Tabela 12	Possíveis vínculos para cada solução	32
Tabela 13	Equações de vínculo	34

1 INTRODUÇÃO

As equações diferenciais não lineares, as quais são geralmente de solução analítica difícil, ou mesmo impossível, fazem parte das pesquisas mais modernas da física, como o caso da equação de Schroedinger não linear [1], que, entre outras aplicações, a propagação da luz em meio não lineares e na dinâmica do condensado de Bose-Einstein. Mas será que existe algum método que transforme a equação não linear em linear? Isso seria bom, porque a linearidade de uma equação diferencial facilita o processo de resolução.

Este trabalho apresenta o método de decomposição de Adomian [2], que transforma o problema não linear em infinitas equações lineares recursivas, ou seja, as equações diferenciais posteriores dependem das soluções anteriores. Assim, pode-se escolher um grau de precisão e solucionar um número N de equações lineares, com essa solução, o teste de resíduo irá determinar se N é suficiente, ou se é necessário avançar na relação de recorrência.

À primeira vista, pode parecer que transformamos o problema não linear em muitas equações lineares totalmente diferentes. Contudo, conforme será apresentado no capítulo 3 de aplicações, os problemas estudados apresentaram um padrão em cada recursão, o que possibilitou um passo-a-passo nos problemas, alterando somente certos valores em cada solução.

O método traz uma possibilidade de resolução algorítmica de diversos problemas, o que por sua vez, facilita a implementação de um programa de computador para a resolução de equações não lineares. Quanto às aplicações, é comum nos estudos de dinâmicas nas áreas de engenharia, como no estudo de dinâmica de fluídos, embora o método seja pouco conhecido na física.

Após essa abordagem inicial de apresentação e aplicação da técnica, este trabalho ainda busca entender os algoritmos quânticos visando buscar uma maneira de integrá-los a técnica de Adomian, isto é, após apresentar um breve estudo sobre algoritmos clássicos e quânticos, o trabalho apresenta três esboços de algoritmos.

O primeiro esboço de uma proposta de algoritmo quântico apresentado como uma curiosidade dos entraves que podem surgir no desenvolvimento de um algoritmo quântico. O segundo não é aplicado numa etapa inicial do problema e sim na normalização da solução truncada. Já o terceiro é aplicado logo após a decomposição da

equação não linear, quer dizer, ele é utilizado para resolver as equações lineares. Basicamente, neste trabalho, usamos o método de Adomian para transformar uma equação diferencial não linear em uma série de equações diferenciais lineares, para as quais há algoritmos quânticos bem fundamentados na literatura, que resolvem o problema, e para ilustrar aplicamos na equação de Schroedinger não linear.

2 O MÉTODO DE DECOMPOSIÇÃO DE ADOMIAN

2.1 INTRODUÇÃO

O método de decomposição de Adomian [2] utiliza infinitas recursões, sendo que na inicialização buscamos a solução da equação original sem o termo não linear. Já na primeira recursão buscamos a solução para uma equação com um termo linear adicional; a segunda recursão, em geral, terá mais termos lineares do que a primeira recursão e as demais recursões terão mais termos lineares do que a segunda recursão. Já a utilidade do método é garantida, porque a analiticidade é preservada em termos de séries infinitas – somas das soluções de infinitas recursões – assim, a técnica fornece uma possibilidade de solução numérica para o problema não linear original, uma vez que a convergência ao truncarmos a série seja garantida.

2.2 O MÉTODO

Iniciamos a explicação do método com um caso particular para exemplificar a nomenclatura e, após a exemplificação, partiremos para o estudo do caso universal do método, isto é, em termos gerais.

Ao escolher uma equação para aplicar o método de Adomian, dividimos ela em quatro partes, por exemplo, na equação:

$$\left(\frac{\partial^2}{\partial x^2} + 1 + y^2\right)y = x^2 \quad (1)$$

há quatro termos, cada um representando uma parte da técnica.

- $\frac{\partial^2}{\partial x^2}$ Termo linear de mais alta ordem, representaremos por “L”;
- 1 Resto do termo linear, representaremos por “R”;
- y^2 Termo não linear, representaremos por “N”;
- x^2 Termo não-homogêneo, representaremos por “g”.

Agora que está exemplificado o que cada parte significa, introduzimos o método formalmente. Iniciando com $Ly = g - Ny - Ry$, aplicamos L^{-1} , isto é, o inverso de L.

$$L^{-1}Ly = L^{-1}g - L^{-1}Ny - L^{-1}Ry \quad (2)$$

$$y = (L^{-1}Ly)|_{y=0} + L^{-1}g - L^{-1}Ny - L^{-1}Ry \quad (3)$$

Em paralelo a isso, fazemos $y = \sum_{i=0}^{\infty} \phi_i$ e, assumindo “Ny” como analítico, uma expansão em séries de Taylor, $Ny = \sum_{j=1}^{\infty} A_j(\phi_0, \phi_1, \dots, \phi_n)$, aonde os “ A_j ” são seleções parcialmente arbitrárias dos termos da expansão, os polinômios de Adomian, cujos quatro primeiros termos são exemplificados a seguir:

$$A_1 = f(\phi_0) \quad (4)$$

$$A_2 = \phi_1 \left(\frac{d}{d\phi_0} \right) f(\phi_0) \quad (5)$$

$$A_3 = \phi_2 \left(\frac{d}{d\phi_0} \right) f(\phi_0) + \frac{\phi_1^2}{2!} \left(\frac{d^2}{d\phi_0^2} \right) f(\phi_0) \quad (6)$$

$$A_4 = \phi_3 \left(\frac{d}{d\phi_0} \right) f(\phi_0) + \phi_1 \phi_2 \left(\frac{d^2}{d\phi_0^2} \right) f(\phi_0) + \frac{\phi_1^3}{3!} \left(\frac{d^3}{d\phi_0^3} \right) f(\phi_0) \quad (7)$$

...

O somatório $\sum_{j=1}^{\infty} A_j(\phi_0, \phi_1, \dots, \phi_n)$ é a expansão em série de Taylor da função “Ny” em torno de ϕ_0 . Quanto aos significados, especifiquemos agora:

$f(\phi_0)$: é a solução da inicialização aplicada na não linearidade;

ϕ_i : é a solução da recursão “i”, sendo o “0” a inicialização, ou seja, a solução da equação sem o termo não-linear;

A_i : é o termo que será utilizado na recursão “i”, este termo é adicionado na equação diferencial e esta, por sua vez, produzirá um “ ϕ_i ”. Repare, que na referência bibliográfica o “A” inicia em “0”, optamos por iniciar em “1”, para ficar evidenciado que é o termo utilizado na recursão “1”.

Há um destaque importante a ser feito, a escolha de “ A_i ” é arbitrária, desde que no infinito se some todos os termos da expansão de Taylor e que em uma recursão, somente utilize-se termos referentes a recursões anteriores.

Sendo assim, neste trabalho, a regra é que cada polinômio de Adomian seja composto por todos os termos ainda não utilizados que envolvam somente recursões anteriores. Então, até a terceira recursão os polinômios de Adomian, temos:

$$A_1 = f(\phi_0) \quad (8)$$

$$A_2 = \phi_1 \left(\frac{d}{d\phi_0} \right) f(\phi_0) + \frac{\phi_1^2}{2!} \left(\frac{d^2}{d\phi_0^2} \right) f(\phi_0) + \frac{\phi_1^3}{3!} \left(\frac{d^3}{d\phi_0^3} \right) f(\phi_0) + \dots \quad (9)$$

$$A_3 = \phi_2 \left(\frac{d}{d\phi_0} \right) f(\phi_0) + \phi_1 \phi_2 \left(\frac{d^2}{d\phi_0^2} \right) f(\phi_0) + \frac{1}{2!} \phi_2^2 \left(\frac{d^2}{d\phi_0^2} \right) f(\phi_0) \dots \quad (10)$$

Observe que essa escolha arbitrária de polinômios de Adomian gerariam um problema, se a não linearidade fosse de ordem muito grande, por exemplo, $Ny = y^{10}$, porque os polinômios de Adomian teriam muitos termos em cada recursão. Contudo, conforme veremos no próximo capítulo, a não linearidade utilizada será do tipo $Ny = y^3$, o que torna viável a utilização dessa regra e permite uma técnica geométrica de obtenção rápida dos polinômios.

Continuando a explicação do método de forma geral, escreveremos as equações de cada recursão, pelo menos até a segunda.

Inicialização:

$$\phi_0 = Y(0) + L^{-1}g \quad (11)$$

Primeira recursão:

$$\phi_1 = -L^{-1}R\phi_0 - L^{-1}A_1 \quad (12)$$

Segunda recursão:

$$\phi_2 = -L^{-1}R\phi_1 - L^{-1}A_2 \quad (13)$$

É importante ressaltar que podemos simplificar o processo, caso a inversão do operador “L+R” seja uma função simples - como trigonométrica, ou exponencial, ou seja, que no final vale a pena – porque, assim, simplificamos as recursões e inclusive é o que será feito no capítulo de aplicação. Observe como ocorre a simplificação.

Inicialização:

$$(L + R)\phi_0 = g \quad (14)$$

Primeira recursão:

$$(L + R)\phi_1 = A_1 \quad (15)$$

Segunda recursão:

$$(L + R)\phi_2 = A_2 \quad (16)$$

Agora que vimos sobre o método, discorramos um pouco sobre a importância e aplicação.

2.3 IMPORTÂNCIA E APLICAÇÕES

Para concluirmos o capítulo sobre o método de Adomian, devemos destacar a importância e as aplicações deste método, como resolver o oscilador harmônico linear quântico, ou como resolver as equações de Kundu-Eckhaus [3] [4]. Sendo assim, o método serve para resolver equações não lineares, ou lineares, de forma analítica, porém, com o inconveniente de haver uma série infinita. Contudo, nem tudo está perdido, pois, havendo uma tolerância quanto ao resíduo e garantindo a convergência, basta utilizarmos quantidade suficiente de recursões para obtermos uma solução satisfatória.

Já as equações não lineares, que são costumeiras na física e por nem sempre terem solução, atrapalham o desenvolvimento e a aplicação da física; poderão ter respostas e poderão ser utilizadas de maneira mais satisfatória, como no caso que estudaremos neste trabalho: equação de Schroedinger não linear.

2.4 CONVERGÊNCIA

Este trabalho não tratará de forma profunda a questão da convergência do método, contudo, citamos a tese [5] e o próprio livro do Adomian [2] como fontes para tratar sobre a convergência. Assim, limitamos o estudo sobre a convergência a simplesmente observar se há um comportamento convergente no resíduo após cada recursão.

3 APLICAÇÃO DE ADOMIAN

3.1 INTRODUÇÃO

Apesar do método estar explicado no capítulo anterior, como observamos ao aplicá-lo é crucial explicitar sua utilização, além de que, os mesmos casos serão utilizados no algoritmo que buscamos, isto é, algoritmo que integre Adomian a computação quântica.

Para isso, realizaremos uma aplicação na partícula livre, porém, sobre efeito da não linearidade. Bem como para o oscilador harmônico, ambos os casos em uma dimensão. Cujos resultados lineares são bem conhecidos de livros textos da disciplina de mecânica quântica.

3.2 A NÃO LINEARIDADE KERR

Um exemplo dessa não linearidade é o campo de polarização não linear que é induzida por um campo elétrico. [1]

$$3\chi|E|^2E e^{-i\omega_0 t} \quad (17)$$

Neste trabalho, a não linearidade utilizada será:

$$c|\psi|^2\psi \quad (18)$$

3.3 A EQUAÇÃO DE SCHROEDINGER NÃO LINEAR

As equações que obteremos a resposta serão do tipo.

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} + V\psi = E\psi + c|\psi|^2\psi \quad (19)$$

Para saber mais sobre a equação, consulte [1].

Para isso, primeiro obteremos algumas soluções para $V = \frac{1}{2}m\omega^2x^2$ e após a solução para $V = 0$.

Primeiro, escrevamos a equação de Schroedinger não linear para o oscilador harmônico no estado fundamental.

3.3.1 DESCRIÇÃO DO POTENCIAL HARMÔNICO

$$-\frac{\hbar^2}{2m} \frac{d^2 \psi_0}{dx^2} + \frac{1}{2} m \omega^2 x^2 \psi_0 = E \psi_0 + c \psi_0^3 \quad (20)$$

Como o método trabalha com recursões, teremos que decompor a função ψ_0 em $\psi_0 = \sum_{n=0}^{\infty} \varphi_{0,n}$ agora, em cada recursão escolheremos alguns termos para representar a não linearidade $c\psi_0^3$. O critério utilizado para escolher os termos de cada recursão, será retomado e explicitado a seguir, mas antes, precisamos explicitar como se realiza uma recursão nesta equação.

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{0,n}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{0,n} = E \varphi_{0,n} + \text{termo da recursão } n \quad (21)$$

O “termo da recursão n” é o polinômio de Adomian para $n=1$, calculado da seguinte maneira. Na recursão $n=0$, isto é, na inicialização, esse termo é nulo. Nas demais, seguimos a seguinte regra:

Utilizando a combinação dos somatórios:

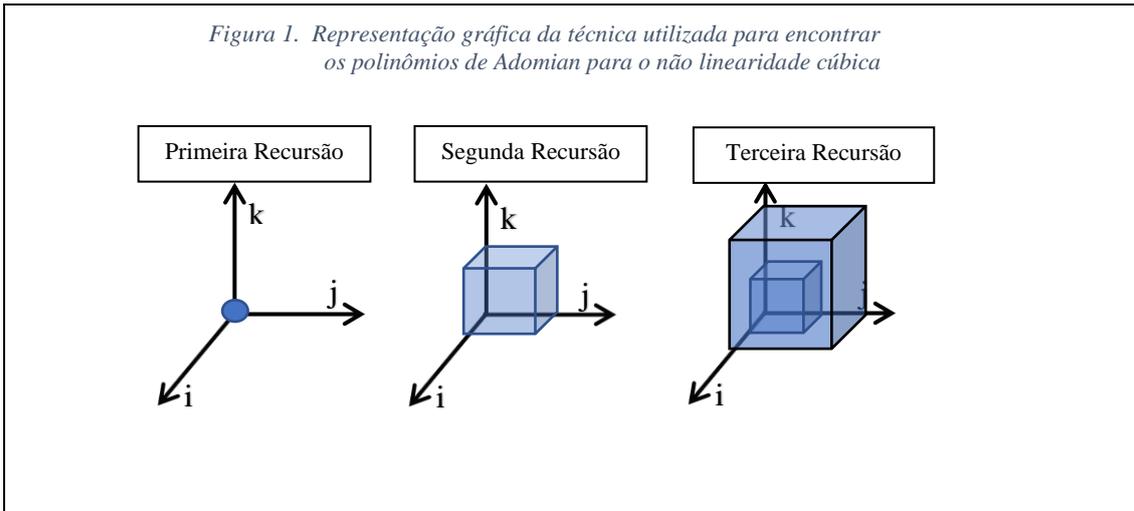
$$\left(\sum_{n=0}^{\infty} \varphi_{0,n} \right)^3 = \left(\sum_{i=0}^{\infty} \varphi_{0,i} \right) * \left(\sum_{j=0}^{\infty} \varphi_{0,j} \right) * \left(\sum_{k=0}^{\infty} \varphi_{0,k} \right) \quad (22)$$

Na primeira recursão – $n=1$ na equação (3) - utilizar somente $\varphi_{0,0}^3$, isto é, $i=j=k=0$. Depois, na segunda recursão, utilizar todas as combinações que contenham índices menores do que “2”, excluindo-se a combinação já utilizada $\varphi_{0,0}^3$. Então, ficamos com $3\varphi_{0,1}\varphi_{0,0}^2 + 3\varphi_{0,0}\varphi_{0,1}^2 + \varphi_{0,1}^3$. Seguindo a lógica, na terceira recursão, $n=3$, temos:

$$3\varphi_{0,2}\varphi_{0,0}^2 + 3\varphi_{0,2}\varphi_{0,1}^2 + 3\varphi_{0,2}^2\varphi_{0,0} + 3\varphi_{0,2}^2\varphi_{0,1} + \varphi_{0,2}^3 + 6\varphi_{0,0}\varphi_{0,1}\varphi_{0,2} \quad (23)$$

Observe que a técnica consiste em utilizar o resultado $\varphi_{0,0}$ obtido na inicialização e utilizar na recursão $n=1$, depois, utilizar os resultados obtidos nas próximas recursões e assim por diante.

Para fins didáticos, desenvolvemos técnicas de obtenção dos termos de cada recursão de forma prática. Como a regra neste trabalho é: “O polinômio de Adomian de cada recursão é composto de todos os termos da expansão em Taylor ainda não utilizados, porém, que envolvam somente recursões anteriores a que se busca solucionar”. Imagine que tenhamos um cubo que expande em cada recursão. Assim:



Então, utilizamos os índices naturais que estão na superfície de cada cubo. Observe que na primeira recursão é um cubo de volume zero e utilizamos $\{\varphi_{0,0}, \varphi_{0,0}, \varphi_{0,0}\} = \rightarrow \varphi_{0,0}^3$. Na segunda recursão, utilizamos os índices:

$$(i, j, k) = \{(1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), (1,1,1)\}$$

Resultando em: $3\varphi_{0,1}\varphi_{0,0}^2 + 3\varphi_{0,0}\varphi_{0,1}^2 + \varphi_{0,1}^3$, observe que nem todos os valores utilizados estão nos vértices.

3.3.2 SOLUÇÕES DE $\varphi_{0,n}$

3.3.2.1 Solução $\varphi_{0,0}$

Neste problema do estado fundamental, as soluções analíticas de cada recursão são padrões e seguem uma lógica, vejamos a inicialização, que é (3) sem nenhum termo de recursão. A solução é bem conhecida e é: $\varphi_{0,0} = A_{0,1}e^{-\frac{1}{2}x^2}$, na constante "A" o "0" indica a recursão e o "1" indica a constante no numerador da exponencial com que há o produto da constante.

Repare que ao substituir a solução na equação de inicialização, surgem as seguintes equações que devem ser respeitadas.

$$E = \frac{\hbar^2}{2m} \quad (24)$$

$$\frac{\hbar^2}{2} = m^2\omega^2 \quad (25)$$

O leitor deve reparar que as unidades estão diferentes nos dois lados das equações acima e sempre faltará alguma unidade em relação ao metro quadrado nas equações de vínculo que surgirem, isso ocorre, pois, nossa solução já utiliza a exponencial com o valor de “ $-\frac{1}{2}$ ”, então, a constante nas equações de vínculo não são adimensionais. Para ficar evidente, se ao invés de “ $-\frac{1}{2}$ ” utilizássemos “ α ” a primeira equação seria.

$$E = \frac{-\alpha\hbar^2}{m} \quad (26)$$

Com “ α ” com unidade de “ $\frac{1}{m^2}$ ”. Outra maneira de enxergar a situação é dizer que a posição “ x ” é adimensional, ou seja, foi adimensionalizado.

3.3.2.2 Solução $\varphi_{0,1}$

Utilizando o termo da primeira recursão, $c\varphi_{0,0}^3$, obtemos a seguinte equação:

$$-\frac{\hbar^2}{2m} \frac{d^2\varphi_{0,1}}{dx^2} + \frac{1}{2}m\omega^2x^2\varphi_{0,1} = E\varphi_{0,1} + c\varphi_{0,0}^3 \quad (27)$$

Como sabemos a solução $\varphi_{0,0}$, temos que a primeira recursão é:

$$-\frac{\hbar^2}{2m} \frac{d^2\varphi_{0,1}}{dx^2} + \frac{1}{2}m\omega^2x^2\varphi_{0,1} = E\varphi_{0,1} + cA_{0,1}^3(e^{-\frac{1}{2}x^2})^3 \quad (28)$$

Ou seja:

$$-\frac{\hbar^2}{2m} \frac{d^2\varphi_{0,1}}{dx^2} + \frac{1}{2}m\omega^2x^2\varphi_{0,1} = E\varphi_{0,1} + cA_{0,1}^3e^{-\frac{3}{2}x^2} \quad (29)$$

Acontece que essa é uma equação linear não homogênea cuja solução é uma equação homogênea mais uma solução particular, mas a forma da solução homogênea é igual a $\varphi_{0,0}$, mudando somente a constante multiplicativa. Então, utilizando a notação geral do capítulo 1, utilizaremos a equação (15).

Contudo, lembre-se que $\psi_0 = \sum_{n=0}^{\infty} \varphi_{0,n}$, logo, podemos considerar a contribuição de todas as soluções homogêneas somente na inicialização, isto é, as constantes das demais soluções homogêneas serão idênticas a zero.

Assim, todo o trabalho para solucionar as recursões será procurar soluções particulares do termo não homogêneo. Acontece, que quando temos um termo não-

homogêneo do tipo $C_1 e^{ax^2}$ a solução particular é do tipo $C_2 e^{ax^2}$. Concluimos que para a primeira recursão.

$$\varphi_{0,1} = A_{1,3} e^{-\frac{3}{2}x^2} \quad (30)$$

Agora, ao substituirmos $\varphi_{0,1}$ na equação, verificaremos que surgem algumas equações de vínculo:

$$\frac{m^2 w^2}{\hbar^2} = 9 \quad (31)$$

$$A_{1,3} = \frac{cA_{0,1}^3}{\left(\frac{3\hbar^2}{2m} - E\right)} \quad (32)$$

Essas equações de vínculo serão exploradas no código em Matlab para acharmos a função de onda.

3.3.2.3 Solução $\varphi_{0,2}$

Como podemos constatar, a partir da segunda recursão, o termo não homogêneo será combinações de exponenciais, como em:

$$3\varphi_{0,1}\varphi_{0,0}^2 + 3\varphi_{0,0}\varphi_{0,1}^2 + \varphi_{0,1}^3 \quad (33)$$

$$3A_{1,3}A_{0,1}^2 e^{-\frac{3}{2}x^2} (e^{-\frac{1}{2}x^2})^2 + 3A_{0,1}A_{1,3}^2 e^{-\frac{1}{2}x^2} (e^{-\frac{3}{2}x^2})^2 + A_{1,3}^3 (e^{-\frac{3}{2}x^2})^3 \quad (34)$$

$$3A_{1,3}A_{0,1}^2 e^{-\frac{5}{2}x^2} + 3A_{0,1}A_{1,3}^2 e^{-\frac{7}{2}x^2} + A_{1,3}^3 e^{-\frac{9}{2}x^2} \quad (35)$$

Isso poderia dificultar o processo de resolução, [6] se não fosse o teorema da superposição de soluções particulares, que diz que numa equação diferencial com soma de funções não homogêneas, a solução particular será uma soma de soluções particulares de cada equação diferencial contendo uma das funções não-homogêneas. Então, temos a equação:

$$\begin{aligned}
-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{0,2}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{0,2} & \quad (36) \\
= E \varphi_{0,2} + 3c A_{1,3} A_{0,1}^2 e^{-\frac{5}{2}x^2} + 3c A_{0,1} A_{1,3}^2 e^{-\frac{7}{2}x^2} + c A_{1,3}^3 e^{-\frac{9}{2}x^2}
\end{aligned}$$

Que terá “ $\varphi_{0,2}$ = soma da solução particular de cada termo não – homogêneo”, isto é, faremos:

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi'_{0,2}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi'_{0,2} = E \varphi'_{0,2} + 3c A_{1,3} A_{0,1}^2 e^{-\frac{5}{2}x^2} \quad (37)$$

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi''_{0,2}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi''_{0,2} = E \varphi''_{0,2} + 3c A_{0,1} A_{1,3}^2 e^{-\frac{7}{2}x^2} \quad (38)$$

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi'''_{0,2}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi'''_{0,2} = E \varphi'''_{0,2} + c A_{1,3}^3 e^{-\frac{9}{2}x^2} \quad (39)$$

$$\varphi_{0,2} = \varphi'_{0,2} + \varphi''_{0,2} + \varphi'''_{0,2} \quad (40)$$

Ora, já fora discutido a solução particular desse tipo de equação não homogênea.

Assim:

$$\varphi' = A_{2,5} e^{-\frac{5}{2}x^2} \quad (41)$$

$$\varphi'' = A_{2,7} e^{-\frac{7}{2}x^2} \quad (42)$$

$$\varphi''' = A_{2,9} e^{-\frac{9}{2}x^2} \quad (43)$$

Logo:

$$\varphi_{0,2} = A_{2,5} e^{-\frac{5}{2}x^2} + A_{2,7} e^{-\frac{7}{2}x^2} + A_{2,9} e^{-\frac{9}{2}x^2} \quad (44)$$

E as equações de vínculo são

$$A_{2,5} = \frac{3c A_{1,3} A_{0,1}^2}{\left(\frac{5\hbar^2}{2m} - E\right)} \quad (45)$$

$$A_{2,7} = \frac{3cA_{0,1}A_{1,3}^2}{\left(\frac{7\hbar^2}{2m} - E\right)} \quad (46)$$

$$A_{2,9} = \frac{cA_{1,3}^3}{\left(\frac{9\hbar^2}{2m} - E\right)} \quad (47)$$

Observe que ao somar os índices das constantes ao lado direito, obteremos o valor do índice da constante do lado esquerdo, vejamos, em $A_{2,5}$, $3+1+1=5$, isso é uma facilidade da notação.

3.3.2.4 Passo a passo para soluções

Observamos no desenvolvimento do trabalho que o processo para encontrar as soluções de cada recursão seguem um padrão. Porém, no final, devemos expressar todas as constantes em termo de $A_{0,1}$ para podermos normalizar e reduzir o problema a solução de uma equação polinomial. Resumindo o passo-a-passo a seguir:

- I. Achar o termo não-homogêneo da recursão “n” em questão.
- II. Fazer $\varphi_{0,n} = \sum_{i=a}^b A_{n,i} e^{-\frac{i}{2}x^2}$ sendo que: $a=(2n+1)$, “i” só pode ser ímpar e $b = 3^n$.
- III. As constantes da solução da n-ésima recursão $A_{n,i}$ seguem um padrão. Utilizamos o termo que acompanha o termo exponencial de numerador “ $-\frac{i}{2}$ ” e dividimos pelo termo $\left(\frac{i\hbar^2}{2m} - E\right)$. Então, se um termo não-homogêneo da terceira recursão for:

$$c(6A_{2,5}A_{1,3}A_{0,1} + 3A_{2,7}A_{0,1}^2)e^{-\frac{9}{2}x^2}$$

A equação de vínculo será:

$$A_{3,9} = \frac{c(6A_{2,5}A_{1,3}A_{0,1} + 3A_{2,7}A_{0,1}^2)}{\left(\frac{9\hbar^2}{2m} - E\right)}$$

Para continuar, substituiremos todas as equações em termos de $A_{0,1}$, então, normalizaremos e para realizar essa tarefa, foi criado um código de Matlab que está no apêndice 1. Na próxima seção, será apresentado alguns dados desse problema com algumas observações referentes aos dados.

3.3.2.5 Alguns resultados

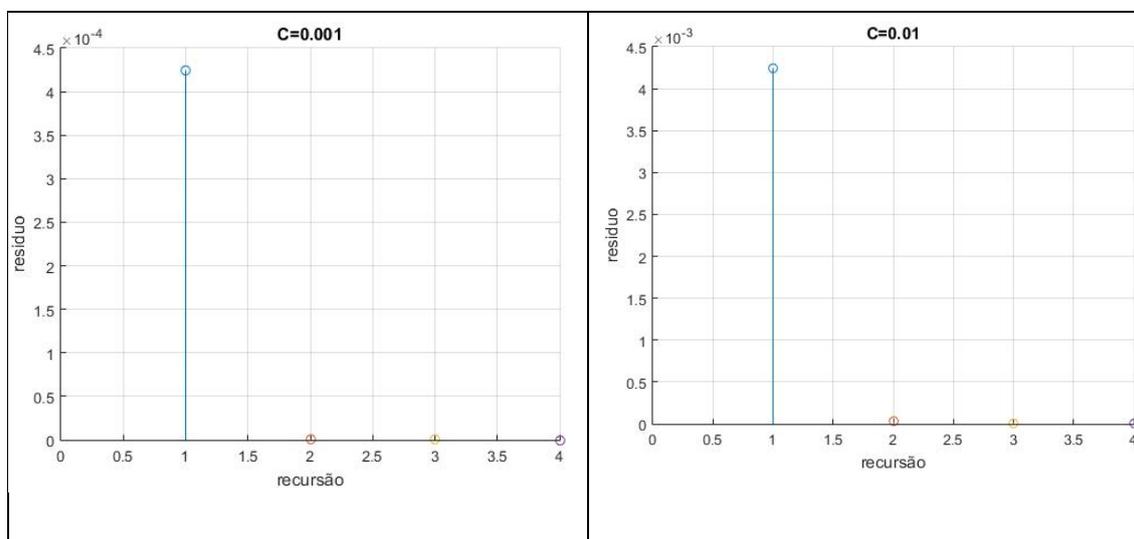
Antes de iniciarmos a apresentação dos dados, é interessante destacar que $E_0 \equiv E_0 = \frac{\hbar^2}{2m}$, e C é o fator do termo não linear da equação (19).

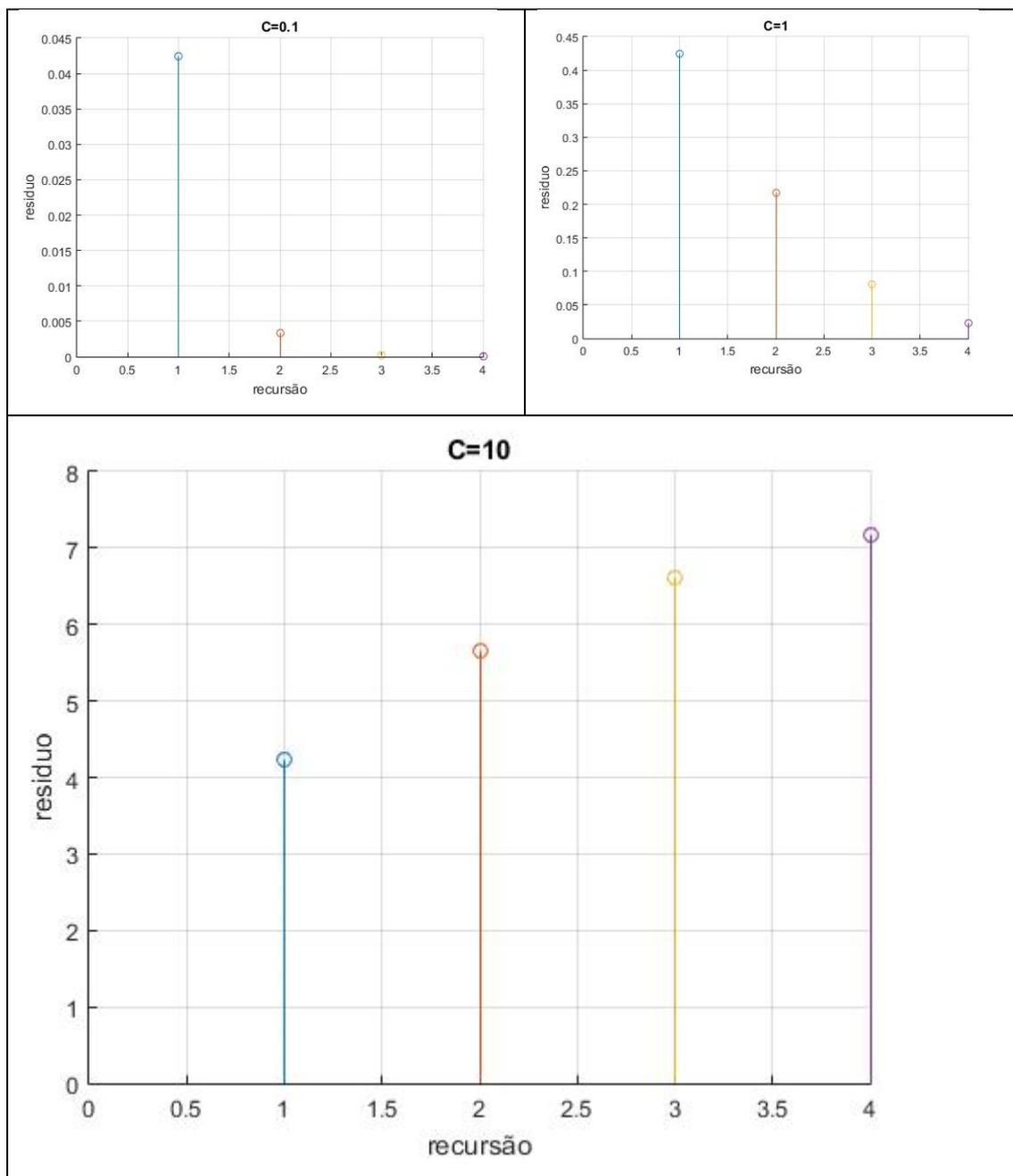
A seguir, seguem algumas tabelas com os maiores resíduos - isto é, ao aplicar a solução truncada na equação diferencial o maior erro para um ponto - de cada solução para diferentes parâmetros e até a terceira recursão.

Tabela 1 – Variação do fator não linear $E=E0=1$

C	inicialização	1ª recursão	2ª recursão	3ª recursão
1E-03	4,238E-04	3,584E-07	1,517E-10	4,279E-14
1E-02	4,238E-03	3,561E-05	1,508E-07	4,264E-10
1E-01	4,238E-02	3,350E-03	1,422E-04	4,102E-06
1	4,238E-01	2,171E-01	8,099E-02	2,358E-02
1E+01	4,238E+00	5,655E+00	6,619E+00	7,163E+00
1E+02	4,238E+01	7,852E+01	1,244E+02	3,646E+15
1E+03	4,238E+02	8,835E+02	1,555E+03	1,638E+07
1E+04	4,238E+03	9,278E+03	1,735E+04	1,228E+75
1E+05	4,238E+04	9,482E+04	1,853E+05	5,433E-07
1E+06	4,238E+05	9,577E+05	1,937E+06	3,822E+120

Segue as figuras correspondentes a alguns valores do fator não linear da tabela 1





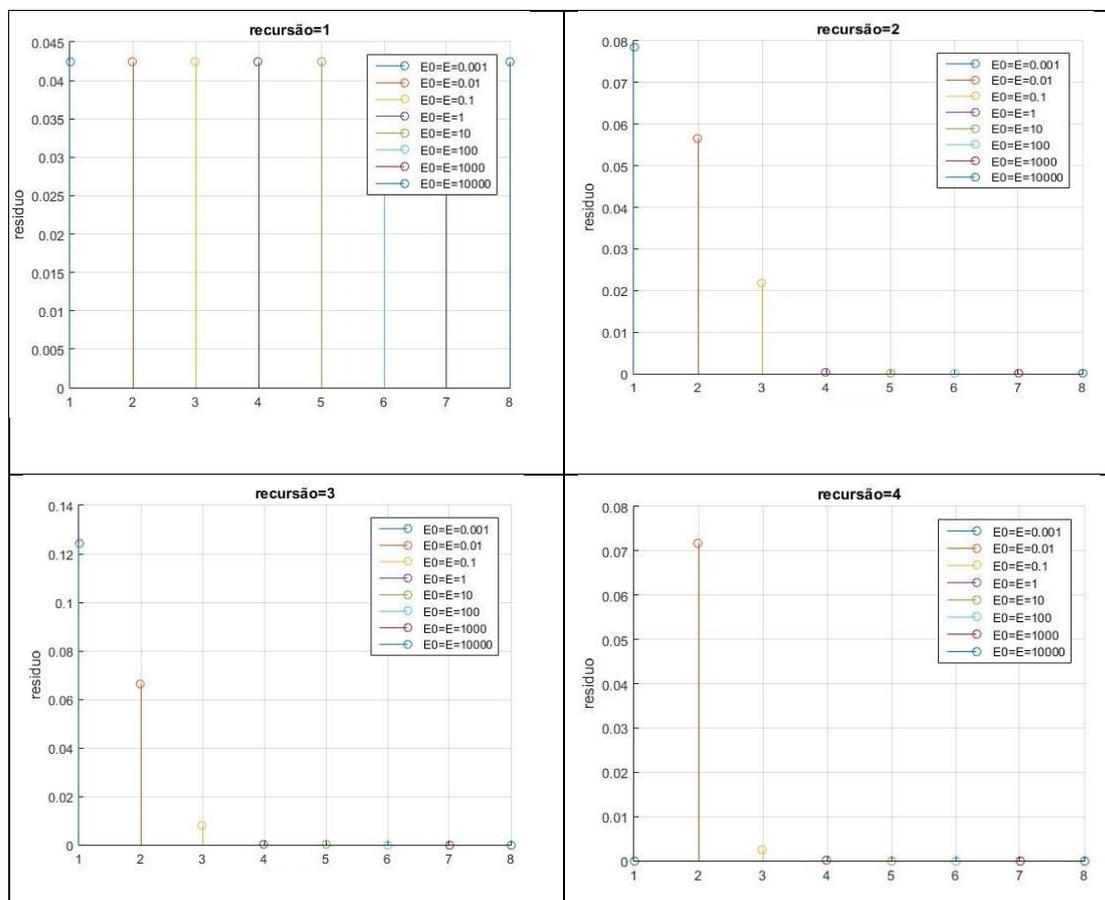
De acordo com a tabela 1, para um aumento do fator não linear, há um resíduo maior e uma necessidade maior de recursões para obter-se um resíduo menor. Observe que alguns números foram muito grandes, como $3,822E+120$, entendemos isso como um limite da máquina e dos recursos utilizados para demonstrar o método.

Tabela 2 – Variação das constantes $E=E_0$, com $C=0,1$

$E=E_0=$	inicialização	1ª recursão	2ª recursão	3ª recursão
1E-03	4,2378E-02	7,8518E-02	1,2437E-01	5,5141E-07
1E-02	4,2378E-02	5,6553E-02	6,6190E-02	7,1628E-02

1E-01	4,2378E-02	2,1714E-02	8,0991E-03	2,3583E-03
1	4,2378E-02	3,3499E-03	1,4221E-04	4,1016E-06
1E+01	4,2378E-02	3,5609E-04	1,5078E-06	4,2643E-09
1E+02	4,2378E-02	3,5838E-05	1,5166E-08	4,2793E-12
1E+03	4,2378E-02	3,5861E-06	1,5174E-10	4,2813E-15
1E+04	4,2378E-02	3,5863E-07	1,5175E-12	2,0817E-17
1E+05	4,2378E-02	3,5864E-08	1,5196E-14	2,0817E-17
1E+06	4,2378E-02	3,5864E-09	1,4572E-16	1,0408E-17

Segue os gráficos da tabela 2 para análise visual.



Na tabela 2, observamos algo mais interessante, isolando o efeito das diferenças das constantes, forçando que elas sejam iguais, destaca-se que aumentando a proporção delas em relação ao fator não linear, o resíduo diminuí abruptamente.

Tabela 3 – Variação da constante E, com E0=1 e com C=0,1

E=	inicialização	1ª recursão	2ª recursão	3ª recursão
1E-03	7,0800E-01	7,3848E-01	7,4043E-01	7,4050E-01
1E-02	7,0124E-01	7,3177E-01	7,3373E-01	7,3380E-01
1E-01	6,3364E-01	6,6471E-01	6,6675E-01	6,6682E-01
1	4,2378E-02	3,3499E-03	1,4221E-04	4,1016E-06
1E+01	6,8025E+00	6,7983E+00	6,7982E+00	6,7983E+00
1E+02	7,4404E+01	7,4392E+01	7,4392E+01	3,0145E+11
1E+03	7,5042E+02	7,5040E+02	7,5040E+02	1,4569E+13
1E+04	7,5105E+03	7,5105E+03	7,5105E+03	4,7641E+14
1E+05	7,5112E+04	7,5112E+04	7,5112E+04	1,5114E+16
1E+06	7,5112E+05	7,5112E+05	7,5112E+05	4,7822E+17

Tabela 4 – Variação da constante E0, com E=1 e com C=0,1

E0=	inicialização	1ª recursão	2ª recursão	3ª recursão
1E-03	7,928E-01	7,761E-01	7,805E-01	4,607E+08
1E-02	7,860E-01	7,699E-01	7,742E-01	3,014E+08
1E-01	7,184E-01	7,098E-01	7,120E-01	7,086E-01
1	4,238E-02	3,350E-03	1,422E-04	4,102E-06
1E+01	6,718E+00	6,751E+00	6,751E+00	6,751E+00
1E+02	7,432E+01	7,435E+01	7,435E+01	7,435E+01
1E+03	7,503E+02	7,504E+02	7,504E+02	7,504E+02
1E+04	7,510E+03	7,510E+03	7,510E+03	7,510E+03
1E+05	7,511E+04	7,511E+04	7,511E+04	7,511E+04
1E+06	7,511E+05	7,511E+05	7,511E+05	7,511E+05

Tabela 5 – Variação de x, com E=E0 / x , com C=0,1 e com E0=1

X =	inicialização	1ª recursão	2ª recursão	3ª recursão
1	4,2378E-02	3,3499E-03	1,4221E-04	4,1016E-06
2	3,3319E-01	3,6712E-01	3,6957E-01	3,6966E-01
3	4,5837E-01	4,9101E-01	4,9328E-01	4,9336E-01
4	5,2097E-01	5,5302E-01	5,5520E-01	5,5528E-01
5	5,5852E-01	5,9024E-01	5,9237E-01	5,9245E-01
6	5,8356E-01	6,1506E-01	6,1715E-01	6,1723E-01
7	6,0144E-01	6,3279E-01	6,3486E-01	6,3494E-01

8	6,1486E-01	6,4609E-01	6,4815E-01	6,4822E-01
9	6,2529E-01	6,5643E-01	6,5848E-01	6,5856E-01
10	6,3364E-01	6,6471E-01	6,6675E-01	6,6682E-01

Nas tabelas 3, 4 e 5, buscamos demonstrar a forte dependência entre a proporção de E e E0 nesse algoritmo. Observamos que o método depende fortemente da proporção entre as constantes da equação estudada.

3.3.3 A EQUAÇÃO DE SCHROEDINGER NÃO LINEAR PARA V=0

Procuraremos soluções da equação:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi}{dx^2} = E\psi + c\psi^2 \psi^* \quad (48)$$

Observe a adaptação ao método em relação ao potencial harmônico, pois, a inicialização resultará em exponenciais complexas, o que exige que utilizemos a não linearidade geral.

3.3.3.1 Inicialização

Começando pela inicialização, obtemos:

$$\psi_0 = Ae^{-i\gamma x} + Be^{i\gamma x} \quad (49)$$

Com $\gamma = \sqrt{\frac{2mE}{\hbar^2}}$, apesar de na física essa constate ser o módulo o vetor de onda (k), utilizaremos uma variável nova para não carregar sentido físico.

3.3.3.2 primeira recursão

Para a primeira recursão, temos:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi_1}{dx^2} = E\psi_1 + c\psi_0^2 \psi_0^* \quad (50)$$

Ou seja:

$$-\frac{\hbar^2}{2m} \frac{d^2\psi_1}{dx^2} = E\psi_1 + c(Ae^{-i\gamma x}(|A|^2 + 2|B|^2) + Be^{i\gamma x}(2|A|^2 + |B|^2) + A^2B^*e^{-3i\gamma x} + A^*B^2e^{3i\gamma x}) \quad (51)$$

Para obter ψ_1 observamos que parte do termo não-homogênea é proporcional a solução do estado homogêneo (inicialização). Isso requer que utilizemos dois tipos de soluções particulares.

Caso 1: Termo não-homogêneo do tipo $\lambda e^{\pm i\gamma x}$

Solução do caso 1: $\beta x e^{\pm i\gamma x}$ com $\beta = \pm i\lambda \sqrt{\frac{m}{2\hbar^2 E}}$

Caso 2: Termo não-homogêneo do tipo $\lambda e^{-\alpha i\gamma x}$ com $\alpha \neq \{-1, 1\}$

Solução do caso 2: $\beta e^{-\alpha i\gamma x}$ com $\beta = \frac{\lambda}{(\alpha^2 - 1)E}$

Então, aplicando as soluções de cada caso e substituindo os valores das constantes, obtemos:

$$\psi_1 = c \left(ix \sqrt{\frac{m}{2\hbar^2 E}} (A(|A|^2 + 2|B|^2)e^{-i\gamma x} + B(2|A|^2 + |B|^2)e^{i\gamma x}) + \frac{1}{8E} (A^2B^*e^{-3i\gamma x} + A^*B^2e^{3i\gamma x}) \right) \quad (52)$$

A pergunta que fica é, sendo as soluções funções complexas, não podemos seguir como fizemos no caso do potencial harmônico. Teremos de usar expansão em séries de Taylor de duas variáveis, uma ψ^2 e outra ψ^* , contudo, a regra de utilizar todos os termos que já conhecidos até a respectiva recursão ainda é válida. Assim, obtemos o polinômio de Adomian para a segunda recursão.

3.3.3.3 Segunda recursão

O polinômio de Adomian para a segunda recursão é:

$$2\psi_0\psi_0^*\psi_1 + \psi_0^2\psi_1^* + \psi_0^*\psi_1^2 + 2\psi_0\psi_1^*\psi_1 + \psi_1^2\psi_1^* \quad (53)$$

Apesar do aumento de dificuldade devido a inclusão das generalizações complexas, o aspecto que este trabalho deseja destacar é que é possível encontrar um passo-a-passo para encontrar as soluções, isso ocorre, porque este problema de Adomian utiliza soluções

exponenciais e o método de Adomian trabalha com recursões, então, as soluções são combinações que seguem uma certa regra. Assim, para explicar essa ideia, reescrevamos as soluções da inicialização e da primeira recursão da seguinte forma:

$$\psi_0 = A_{0,-1}e^{-1i\gamma x} + A_{0,1}e^{+1i\gamma x} \quad (54)$$

$$\psi_1 = A_{1,-3}e^{-3i\gamma x} + A_{1,-1}e^{-1i\gamma x} + A_{1,1}e^{1i\gamma x} + A_{1,3}e^{3i\gamma x} \quad (55)$$

Fizemos isso, porque facilita escrevermos o termo não-homogênea da segunda recursão; também se destaque que as “constantes” podem conter uma $f(x)$, por isso, definimos elas como o que multiplica a exponencial. E a justificativa é que para escrevermos quais constantes multiplicam uma dada exponencial precisamos de duas regras, a primeira regra é que se a exponencial contiver o expoente “-5”, as somas dos índices das constantes que multiplicam a exponencial devem resultar em “-5”. Logo, para o termo não-homogêneo “ $2\psi_0\psi_0^*\psi_1$ ” a exponencial $e^{-5i\gamma x}$ é multiplicada pelas constantes “ $A_{0,-1}; A_{0,1}; A_{1,-3}$ ” repare que o sinal indicativo de complexo “*” faz com que o índice subtraia, ao invés, de adicionar. Essa é a primeira regra, a segunda regra trata da quantidade de constantes que utilizaremos de cada recursão para compor o termo não-homogêneo.

A regra é: um termo não homogêneo da segunda recursão conterà a quantidade de constantes de uma recursão anterior - complexa, ou não - igual à quantidade de vezes que essa recursão é utilizada no termo.

Então, por exemplo, o termo “ $\psi_0^2\psi_1^*$ ” terá em cada exponencial uma soma de constantes, porém, essas constantes serão compostas por duas de ψ_0^2 e uma de ψ_1^* , e a soma dos índices de cada constante tem que ser igual ao número no expoente da exponencial. Agora que temos as regras e como utilizá-las vamos escrever os termos não-homogêneos em termos das constantes.

Devido ao termo não homogêneo “ $\psi_1^2\psi_1^*$ ”, o de maior ordem, sabemos que ψ_2 terá expoentes ímpares de -9 a 9. Assim, a solução da segunda recursão poderá ser expressa como:

$$\begin{aligned} \psi_2 = & A_{2,-9}e^{-9i\gamma x} + A_{2,-7}e^{-7i\gamma x} + A_{2,-5}e^{-5i\gamma x} + A_{2,-3}e^{-3i\gamma x} + A_{2,-1}e^{-1i\gamma x} \quad (56) \\ & + A_{2,1}e^{1i\gamma x} + A_{2,3}e^{3i\gamma x} + A_{2,5}e^{5i\gamma x} + A_{2,7}e^{7i\gamma x} + A_{2,9}e^{9i\gamma x} \end{aligned}$$

Veamos a tabela 1 a seguir que relaciona as constantes da primeira recursão e da inicialização as exponenciais da solução da segunda recursão, devido ao termo não-homogêneo $\psi_0\psi_0^*\psi_1$

Tabela 6 Contribuição do termo não-homogêneo $\psi_0\psi_0^*\psi_1$

Exponencial	Contribuição
$e^{-5i\gamma x}$	$A_{0,-1}A_{0,1}^*A_{1,-3}$
$e^{-3i\gamma x}$	$A_{0,-1}A_{0,-1}^*A_{1,-3} ; A_{0,1}A_{0,1}^*A_{1,-3} ; A_{0,-1}A_{0,1}^*A_{1,-1}$
$e^{-1i\gamma x}$	$A_{0,-1}A_{0,-1}^*A_{1,-1} ; A_{0,1}A_{0,1}^*A_{1,-1} ; A_{0,-1}A_{0,1}^*A_{1,1} ; A_{0,1}A_{0,-1}^*A_{1,-3}$
$e^{1i\gamma x}$	$A_{0,1}A_{0,1}^*A_{1,1} ; A_{0,-1}A_{0,-1}^*A_{1,1} ; A_{0,1}A_{0,-1}^*A_{1,-1} ; A_{0,-1}A_{0,1}^*A_{1,3}$
$e^{3i\gamma x}$	$A_{0,1}A_{0,1}^*A_{1,3} ; A_{0,-1}A_{0,-1}^*A_{1,3} ; A_{0,1}A_{0,-1}^*A_{1,1}$
$e^{5i\gamma x}$	$A_{0,1}A_{0,-1}^*A_{1,3}$

Repare que há uma simetria para obter os termos não-homogêneos das exponenciais positivas, logo, precisamos somente copiar as negativas e trocar os sinais. Portanto, agora somente escreveremos as exponenciais negativas. Segue abaixo a tabela 2, que relaciona as contribuições devido ao termo $\psi_0^2\psi_1^*$.

Tabela 7 Contribuição do termo não-homogêneo $\psi_0^2\psi_1^*$.

Exponencial	Contribuição
$e^{-5i\gamma x}$	$A_{0,-1}A_{0,-1}A_{1,3}^*$
$e^{-3i\gamma x}$	$A_{0,-1}A_{0,-1}A_{1,1}^* ; 2A_{0,1}A_{0,-1}A_{1,3}^* ;$
$e^{-1i\gamma x}$	$2A_{0,-1}A_{0,1}A_{1,1}^* ; A_{0,1}A_{0,1}A_{1,3}^* ; A_{0,-1}A_{0,-1}A_{1,-1}^*$

Abaixo, são apresentadas as tabelas para os demais termos.

Tabela 8 Contribuição do termo não-homogêneo $\psi_0^*\psi_1^2$

Exponencial	Contribuição
$e^{-7i\gamma x}$	$A_{0,1}^*A_{1,-3}A_{1,-3}$
$e^{-5i\gamma x}$	$2A_{0,1}^*A_{1,-1}A_{1,-3} ; A_{0,-1}^*A_{1,-3}A_{1,-3}$
$e^{-3i\gamma x}$	$2A_{0,1}^*A_{1,1}A_{1,-3} ; 2A_{0,-1}^*A_{1,-1}A_{1,-3} ; A_{0,1}^*A_{1,-1}A_{1,-1}$
$e^{-1i\gamma x}$	$2A_{0,1}^*A_{1,3}A_{1,-3} ; 2A_{0,1}^*A_{1,-1}A_{1,1} ; 2A_{0,-1}^*A_{1,1}A_{1,-3} ; 2A_{0,-1}^*A_{1,-1}A_{1,-1}$

Tabela 9 Contribuição do termo não-homogêneo $\psi_0\psi_1^*\psi_1$

Exponencial	Contribuição
$e^{-7i\gamma x}$	$A_{0,-1}A_{1,3}^*A_{1,-3}$
$e^{-5i\gamma x}$	$A_{0,-1}A_{1,1}^*A_{1,-3} ; A_{0,-1}A_{1,3}^*A_{1,-1} ; A_{0,1}A_{1,3}^*A_{1,-3}$
$e^{-3i\gamma x}$	$A_{0,-1}A_{1,-1}^*A_{1,-3} ; A_{0,-1}A_{1,1}^*A_{1,-1} ; A_{0,-1}A_{1,3}^*A_{1,1} ; A_{0,1}A_{1,3}^*A_{1,-1} ; A_{0,1}A_{1,1}^*A_{1,-3}$

$e^{-1i\gamma x}$	$A_{0,-1}A_{1,-3}^*A_{1,-3}; A_{0,-1}A_{1,-1}^*A_{1,-1}$; $A_{0,-1}A_{1,1}^*A_{1,1}; A_{0,-1}A_{1,3}^*A_{1,3}; A_{0,1}A_{1,-1}^*A_{1,-3}; A_{0,1}A_{1,1}^*A_{1,-1}; A_{0,1}A_{1,3}^*A_{1,1}$
-------------------	--

Tabela 10 Contribuição do termo não-homogêneo $\psi_1^2\psi_1^*$

Exponencial	Contribuição
$e^{-9i\gamma x}$	$A_{1,-3}A_{1,-3}A_{1,3}^*$
$e^{-7i\gamma x}$	$2A_{1,-1}A_{1,-3}A_{1,3}^*; A_{1,-3}A_{1,-3}A_{1,1}^*$
$e^{-5i\gamma x}$	$A_{1,-1}A_{1,-1}A_{1,3}^*; 2A_{1,-3}A_{1,1}A_{1,3}^*; 2A_{1,-3}A_{1,-1}A_{1,1}^*; A_{1,-3}A_{1,-3}A_{1,-1}^*$
$e^{-3i\gamma x}$	$2A_{1,-3}A_{1,3}A_{1,3}^*; 2A_{1,-1}A_{1,1}A_{1,3}^*;$ $2A_{1,-3}A_{1,1}A_{1,1}^*; A_{1,-1}A_{1,-1}A_{1,1}^*; 2A_{1,-3}A_{1,-1}A_{1,-1}^*; A_{1,-3}A_{1,-3}A_{1,-3}^*;$
$e^{-1i\gamma x}$	$2A_{1,-1}A_{1,3}A_{1,3}^*; A_{1,1}A_{1,1}A_{1,3}^*; 2A_{1,-3}A_{1,3}A_{1,1}^*; 2A_{1,-1}A_{1,1}A_{1,1}^*;$ $2A_{1,-3}A_{1,1}A_{1,-1}^*; A_{1,-1}A_{1,-1}A_{1,-1}^*; 2A_{1,-1}A_{1,-3}A_{1,-3}^*;$

Agora que temos todos os termos não homogêneos expressos em termos das recursões anteriores, podemos escrever a solução ψ_2 como em termos da equação (56).

Como há muitos termos em cada constante e há simetria entre índices positivos e negativos, apresentamos explicitamente os três primeiros a seguir:

Tabela 11 Constantes das exponenciais de ψ_2

Exponencial	Contribuição
$A_{2,-9}$	$\frac{1}{80E} \cdot \frac{c^4}{(8E)^3} (A^5 \bar{B}^4)$
$A_{2,-7}$	$\frac{3c^3}{48E(8E)^2} \cdot A^4 \bar{B}^3 \cdot \left[1 + icx \sqrt{\frac{m}{2\hbar^2 E}} (B ^2) \right]$
$A_{2,-5}$	$\frac{c^2}{24E \cdot (8E)} \cdot A^3 \bar{B}^2 \cdot \left[3 + \frac{c}{8E} \left[A ^2 + 2 B ^2 + \left(icx \sqrt{\frac{m}{2\hbar^2 E}} \right) \cdot (- A ^4 + \right. \right.$ $\left. \left. 2 A ^2 B ^2 + 2 B ^4 \right) \right] + c \left(ix \sqrt{\frac{m}{2\hbar^2 E}} \right) \cdot [6(B ^2)] +$ $c^2 \left(3x^2 \frac{m}{2\hbar^2 E} A ^2 (1 + 2 B ^2) \right) \right]$

Neste trabalho, truçaremos a explicitação de termos e constantes, pois, a intenção de exemplificar a aplicação do método já está satisfeita e o caso de análise de dificuldades e problemas será feita seção 5.2.1.

3.3.4 SOLUÇÕES DE $\varphi_{1,n}$

3.3.4.1 Solução $\varphi_{1,0}$

O estado de inicialização é conhecido da literatura [7] e adaptamos para a forma $\varphi_{1,0} = B_{0,1} x e^{-\frac{x^2}{2}}$, onde utilizamos a constante “B” para indicar que se trata do primeiro estado excitado.

Com equações de vínculo.

$$\hbar^2 = m^2 \omega^2 \quad (57)$$

$$E = \frac{3}{2} \frac{\hbar^2}{m} = \frac{3}{2} m \omega^2 \quad (58)$$

3.3.4.2 Solução $\varphi_{1,1}$

A equação da primeira recursão é:

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{1,1}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{1,1} = E \varphi_{1,1} + c \varphi_{1,0}^3 \quad (59)$$

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{1,1}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{1,1} = E \varphi_{1,1} + c B_{0,1}^3 x^3 e^{-\frac{3x^2}{2}} \quad (60)$$

Para essa recursão há, pelo menos, duas possíveis soluções particulares:

$$\varphi_{1,1} = B_{1,1} x e^{-\frac{3x^2}{2}} \quad (61)$$

$$\varphi_{1,1} = B_{1,1} x^5 e^{-\frac{3x^2}{2}} \quad (62)$$

Isso acontece, porque ao escolher uma solução do tipo $\varphi_{1,1} = B_{1,1} x^\beta e^{-\frac{3}{2}x^2}$, para que essa seja solução da equação, existe somente as possibilidades $\beta = 1$, ou 5 . Cada uma implica em situações diferentes. Para compreender as situações, vejamos as possíveis equações de vínculo na tabela 7 a seguir:

Tabela 12 Possíveis vínculos para cada solução

Exponencial	Vínculo para E	Vínculo para B	Vínculo extra
$\beta = 1$	$E = 3E_0 = 3 \cdot \frac{3}{2} \frac{\hbar^2}{m}$	$B_{1,3} = \frac{c B_{0,1}^3}{\frac{m \omega^2}{2} - E_0}$	-

$\beta = 5$	$E = 11E_0$	$B_{1,3} = -\frac{3cB_{0,1}^3}{5 \cdot 4 \cdot E}$	$w = \sqrt{2mE_0}$
-------------	-------------	--	--------------------

Repare que “E” não é energia, isso acontece, porque estamos trabalhando com recursões e não existe um estado de fato que represente essa energia, então, aqui o “E” perde seu significado físico.

Para o caso $\beta = 1$, surgirá uma equação de vínculo relacionando “ ω ” as constantes das exponenciais, o problema é o que é “ ω ”? Quer dizer, no oscilador harmônico quântico clássico há um significado físico e sabemos como trabalhar com essa quantidade, contudo, em termos de recursões, o que significa? Além disso, temos o benefício que, até o momento, representa uma facilidade nas soluções das recursões, pois, fixa-as em $\varphi_{1,d} = B_{d,a} x e^{-\frac{ax^2}{2}}$, com “d” representando a recursão e “a” o fator no expoente. Também, mantém as equações de vínculo padronizadas, o que facilita na criação de um algoritmo.

Para o caso $\beta = 5$, eliminamos o “ ω ” na equação que fornece a constante da exponencial - $B_{1,3}$ - e temos uma relação da constante E.

Em síntese, a discussão é matemática, ao substituir a sugestão de solução na equação de uma dada recursão e desenvolver o trabalho para a obtenção das constantes. Desejamos que o termo não-homogêneo seja contrabalanceado com qual outro termo?

Podemos escolher entre: $x^{\beta+2}$ para o caso $\beta = 1$, ou $x^{\beta-2}$, para o caso $\beta = 5$.

3.3.4.3 Solução $\varphi_{1,2}$

Optando por $\beta = 1$, isto é, todas as soluções serão com “x” elevado a potência “1” a equação para a segunda recursão vira:

$$\begin{aligned} \frac{\hbar^2}{2m} \frac{d^2 \varphi_{1,2}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{1,2} & \quad (63) \\ & = E \varphi_{1,2} + c(3B_{1,3} B_{0,1}^2 x^3 e^{-\frac{5}{2}x^2} + 3B_{0,1} B_{1,3}^2 x^3 e^{-\frac{7}{2}x^2} \\ & + B_{1,3}^3 x^3 e^{-\frac{9}{2}x^2}) \end{aligned}$$

Com solução:

$$\varphi_{1,2} = B_{2,5} x e^{-\frac{5}{2}x^2} + B_{2,7} x e^{-\frac{7}{2}x^2} + B_{2,9} x e^{-\frac{9}{2}x^2} \quad (64)$$

Destacamos que a solução desta recursão é parecida com a solução expressa na equação (44), o que demonstra certa simetria no método, mesmo tratando-se de outro estado.

Abaixo, é apresentado na tabela 8 as equações de vínculo para as constantes.

Tabela 13 Equações de vínculo

Vínculo para B	Vínculo para E
$B_{2,5} = -\frac{3cB_{1,3}B_{0,1}^2}{24\frac{E_0}{3}}$	$E = 5E_0$
$B_{2,7} = -\frac{3cB_{0,1}B_{1,3}^2}{48\frac{E_0}{3}}$	$E = 7E_0$
$B_{2,9} = -\frac{B_{1,3}^3}{\frac{80}{3}E_0}$	$E = 9E_0$

3.3.4.4 $\varphi_{1,3}$ e demais soluções

Com a escolha de $\beta = 1$, ou seja, soluções forçadas do estilo $\varphi = Kxe^{-\frac{\alpha x^2}{2}}$, basta seguirmos o passo-a-passo a seguir.

- I. Achar o termo não-homogêneo da recursão “n” em questão.
- II. Fazer $\varphi_{1,n} = \sum_{i=a}^b B_{n,i}xe^{-\frac{i}{2}x^2}$ onde: $a=(2n+1)$, “i” só pode ser ímpar e $b = 3^n$.
- III. As equações de vínculo das constantes da enésima solução - $B_{n,i}$ - seguem o seguinte padrão. Utilizamos o termo não-homogêneo, encontrado em 1, que acompanha o termo exponencial de numerador “ $-\frac{i}{2}$ ” e dividimos pelo termo $-\left(\frac{i^2-1}{3}E_0\right)$.

Então, por exemplo, se o termo não-homogêneo da terceira recursão, que acompanha a exponencial ($x^3e^{-\frac{27}{2}x^2}$) for $cB_{2,9}^3$. Teremos as duas equações de vínculo a seguir:

$$B_{3,27} = -\frac{B_{2,9}^3c}{\frac{27^2-1}{3} \cdot E_0} \quad (65)$$

$$E = 27E_0 \quad (66)$$

Tendo abordado o algoritmo de Adomian clássico, iremos no próximo capítulo revisar sobre algoritmos para então tratar de algoritmo quântico.

4 ALGORITMOS

4.1 CONCEITOS

4.1.1 ALGORITMO

Iniciamos o estudo de algoritmos quânticos com uma definição do que é algoritmo:

Segundo Lipschutz e Lipson [8] um algoritmo é uma lista finita passo a passo de instruções bem definidas para resolver um problema em particular.

Isto é, o algoritmo é a receita passo-a-passo para obter uma resolução de um problema, por exemplo, nos capítulos anteriores descrevemos a obtenção das soluções das recursões no problema do oscilador harmônico não linear. Mas como definir se um algoritmo é melhor do que outro? Discorramos sobre complexidade.

4.1.2 COMPLEXIDADE DE UM ALGORITMO

Iniciamos a discussão sobre complexidade - baseados em [8]- via a definição do que é uma operação chave, que é a operação que toma o maior tempo em comparação as demais operações no algoritmo. Então, por exemplo, suponha um algoritmo com três operações que levam um tempo 100 vezes maior do que a média das demais operações, logo, haverá três operações chaves.

Assim, podemos classificar a complexidade de um algoritmo em duas eficiências: o tempo de máquina necessário para a realização do algoritmo, ou a quantidade de memória necessária, devido as operações chaves. A quantidade de memória necessária normalmente é um múltiplo da entrada de dados. Por isso, em geral, utiliza-se a eficiência do tempo necessário.

Dentro do assunto, é necessário falarmos do pior caso e o caso médio. Acontece que alguns algoritmos dependem das informações fornecidas no problema, por exemplo, um algoritmo que indique a posição do primeiro “o” que encontrar, em geral, será mais rápido do que procurar a posição do primeiro “w”. Isso se deve, pois, é fornecido um texto e neste a probabilidade de encontrar o “o” é muito maior do que a de achar o “w”, pois aquele é mais frequente do que este na língua portuguesa. Mas e se não quisermos medir a eficiência de forma dependente da informação fornecida? Então, teremos que

utilizar o pior caso, assim, tanto o “w” quanto o “o” levam o mesmo tempo. Afinal, posso submeter um texto para análise que não contenha “o”, nem “w”. Como em:

“A rainha é bela”

Já que o tempo necessário para executar o algoritmo de achar o primeiro “o” e o primeiro “w” do texto neste exemplo são iguais.

Contudo, como sabemos que o algoritmo para achar o “o” é mais eficiente do que o de “w” de forma geral? É que temos a intuição de que é mais provável encontrar um “o” num texto em português, assim, isso se deve ao fato do valor esperado de “o” ser maior. Então, na abordagem do caso médio, o algoritmo que sinaliza a posição do primeiro “o” será mais eficiente do que o algoritmo que sinaliza o primeiro “w”.

4.1.3 NOTAÇÃO “O”

Utilizamos essa notação $O(\dots)$, quando desejamos indicar o tempo de execução de um algoritmo depende do tamanho dos dados de entrada [8]. Assim, segue exemplos de como descrever a eficiência de um algoritmo com essa notação.

“O tempo necessário do pior caso é $O(n)$ ”, ou também, “o tempo necessário para o pior caso é $O(n^2)$ ”. Isto é, o primeiro é o caso da busca linear, como o algoritmo para achar o “o”, ou o “w” num texto. Sendo “n” a quantidade de letras do texto. Isto é, o tempo para executar no pior caso é proporcional a quantidade de letras do texto.

4.2 ALGORITMO QUÂNTICO

Iniciamos o estudo de algoritmo quântico, analisando os algoritmos de Grover [9], que é um algoritmo para achar um termo numa lista; e o de Shor [10], que é um algoritmo para fatorar um número inteiro muito grande. Parece bastante intuitivo que o algoritmo quântico é a descrição de como resolver um problema utilizando de algum efeito quântico com a intenção de potencializar algum parâmetro. Por exemplo, o problema solucionado por Grover, um algoritmo clássico teria o pior caso de $O(N)$, já o algoritmo quântico tem o pior caso $O(\sqrt{N})$.

O Grover, por exemplo, para encontrar um termo em uma lista, utilizou rotações sucessivas a fim de aumentar a probabilidade do estado procurado. Assim, se o termo estiver presente, a amplitude do estado correspondente será uma, já os demais estados terão outra amplitude.

4.2.1 DIFERENÇA ENTRE ALGORITMO CLÁSSICO E ALGORITMO QUÂNTICO

É possível construir um algoritmo clássico probabilístico, isto é, processando uma entrada de n bits, não necessariamente, ao fornecer o mesmo conjunto de “ n ” bits o resultado será o mesmo. Contudo, é possível controlar a probabilidade de cada passo de acordo com a necessidade.

Para uma melhor definição, citaremos: [11]

“Algoritmos randomizados são aqueles que utilizam experimentos randômicos para decidir, em um ou mais momentos durante sua execução, o que fazer ou para onde ir. Por motivo de clareza, algoritmos clássicos (não-randomizados) são também ditos determinísticos.

Na maioria dos casos, é conveniente imaginarmos um algoritmo randomizado como que lançando um dado ou uma moeda e, de acordo com o resultado obtido, decidindo entre a execução dessa ou daquela ação. Esta é a figura que estaremos constantemente evocando ao longo do texto quando nos referirmos as escolhas aleatórias que nossos algoritmos precisarão tomar: o algoritmo simplesmente lança um dado — com qualquer número de faces.

É evidente, no entanto, que computadores não podem valer-se de tais expedientes físicos ou mesmo lúdicos. Portanto, na prática, o que entra em cena nos papéis de dados e moedas são os famosos geradores de números aleatórios.”

Um algoritmo quântico é um algoritmo probabilístico generalizado, aonde os efeitos quânticos, como a superposição de estados e a interferência, serão os grandes diferenciais que gerarão efeitos de aumento de eficiência.

Segundo [12]:

“É interessante notar que um algoritmo quântico simula eficientemente um algoritmo probabilístico clássico, mas o contrário não é verdadeiro.”

Antes de partimos para uma discussão sobre o algoritmo quântico de Adomian, vamos apresentar as três grandes classes de problemas tratados pelos algoritmos quânticos atuais.

4.2.2 PARADIGMA DOS PROBLEMAS ALGÉBRICOS E DA TEORIA DOS NÚMEROS

Neste tipo de abordagem é fornecido um problema algébrico e busca-se calcular uma resposta. Como no caso do algoritmo de Shor [10], aonde deseja-se fatorar em termos de primos um dado número. Outro exemplo é o algoritmo para verificar se o produto de duas matrizes quadradas é igual a uma terceira. $AB = C$? [13]

4.2.3 PARADIGMA ORACULAR

Neste tipo de abordagem é dado uma entrada e deseja-se responder uma pergunta específica. Por exemplo, no Grover [9] deseja-se saber se um elemento está contido numa lista de elementos; também é exemplo o algoritmo Deutsch-Jozsa [14] onde dado uma função que pode ser balanceada, ou constante, busca-se descobrir qual das duas opções o caso da entrada.

4.2.4 PARADIGMA DA SIMULAÇÃO

Aqui os algoritmos têm o enfoque de utilizar um computador (sistema) quântico para simular outro problema quântico. Há diversos casos de simulação, por isso, citamos o site – quantum zoo [15]- para maiores estudos.

4.3 SUGESTÕES DE ALGORITMO QUÂNTICO

Desenvolver uma solução para a equação de Schroedinger não linear via algoritmos quânticos não se mostrou uma tarefa fácil, contudo, conseguimos contornar a situação aplicando, inicialmente, um algoritmo quântico a somente alguma parte do problema e posteriormente aplicando de forma mais abrangente. Por isso, a seguir seguem alguns esboços de algoritmos para discussão futura.

4.3.1 ESBOÇO: APLICAR FOURIER NA EQUAÇÃO DE RECURSÃO E APÓS ALGORITMO QUÂNTICO.

A ideia inicial era realizar a decomposição de Adomian da não linearidade $|\psi|^2\psi$ no caso do oscilador harmônico, então, em cada recursão, realizar uma expansão do exponencial quadrático em séries de potências para então aplicar a transformada de Fourier e eliminar a derivada segunda, o que resultaria numa equação polinomial, cujas

raízes encontraríamos com um algoritmo quântico. Como exemplo, segue uma descrição matemática:

Dada a inicialização do potencial harmônico

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{0,0}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{0,0} = E \varphi_{0,0} \quad (67)$$

Realiza-se normalmente, porém, na primeira recursão, equação (29), expandiremos a exponencial em série de potências

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{0,1}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{0,1} = E \varphi_{0,1} + c A_{0,1}^3 e^{-\frac{3}{2}x^2} \quad (29)$$

$$-\frac{\hbar^2}{2m} \frac{d^2 \varphi_{0,1}}{dx^2} + \frac{1}{2} m \omega^2 x^2 \varphi_{0,1} = E \varphi_{0,1} + c A_{0,1}^3 \sum_{n=0}^{\infty} \frac{\left(-\frac{3}{2}x^2\right)^n}{n!} \quad (68)$$

Para então aplicar a transformada de Fourier a fim de eliminar a derivada e transformar em uma equação algébrica, contudo, há um problema conforme vemos ao aplicar a transformada:

$$\frac{d\varphi_{0,1}(x)}{dx} \stackrel{FT}{\leftrightarrow} jk\Phi_{0,1}(jk) \quad (69)$$

O que eliminaria a derivada segunda na posição, porém, geraria uma derivada segunda em “k” devido ao “x²”, além de outras infinitas derivadas devido ao exponencial e assim não seria mais um polinômio, condição que desejávamos de início para aplicar o algoritmo quântico. No caso da partícula livre, a situação é diferente, porém, também não entendemos como resolver o problema aplicando Fourier em combinação ao método de Adomian.

4.3.2 ESBOÇO: APLICAR ALGORITMO QUÂNTICO PARA ENCONTRAR AS RAÍZES DE POLINÔMIOS NA NORMALIZAÇÃO.

O segundo esboço de algoritmo funciona, mas tem limitações. Para utilizar um algoritmo quântico aqui, aplicamos na etapa posterior a normalização, aonde desejamos encontrar as raízes de um polinômio.

O esboço de algoritmo quântico é o seguinte: Realiza-se o passo-a-passo para obter a solução normalmente, porém, no final, quando buscamos a constante de normalização, que é raiz de um polinômio, utilizamos um algoritmo quântico para obtê-lo. Por isso, nas referências [16] [17], temos dois algoritmos para obtenção de raízes de um polinômio. Em [16], é apresentado um algoritmo para encontrar raízes complexas de um polinômio, porém, a resposta fica dentro de uma margem de erro, isto é, é uma solução numérica; o que não é um problema, uma vez que o código implementado em Matlab também apresenta uma solução numérica. Já na [17], o algoritmo é para polinômios com raízes inteiras, o que dificilmente será nosso caso, porém, é interessante destacar a existência de tal algoritmo para estudos futuros.

4.3.3 ESBOÇO: APLICAR O ALGORITMO QUÂNTICO PARA SOLUCIONAR UM SISTEMA DE EQUAÇÕES DIFERENCIAIS.

O terceiro e última abordagem de esboço de algoritmo sugerido é atacar o problema da equação de Schroedinger em estágio anterior ao de procura de raízes de polinômio, então, ou como na [18], que utiliza um algoritmo quântico para solucionar uma equação diferencial não linear diretamente. Ou como na [19], que se utiliza na inicialização e em cada recursão para resolver a equação não-homogênea, porém, linear.

Para aprofundar um pouco o estudo sobre esse esboço, citaremos [20], neste artigo é apresentado um algoritmo quântico para resolver sistemas de equações lineares. Por isso, a seguir apresentaremos o problema do oscilador harmônico quântico como um sistema de equações.

4.3.3.1 Sistema de Equações

Utilizando a equação (21), isto é, a equação de Schroedinger aplicando o método de Adomian, porém, com os termos escritos de forma genérica.

$$Y = \frac{d\varphi_{0,n}}{dx} \quad (70)$$

$$\frac{dY}{dx} = \frac{d^2 \varphi_{0,n}}{dx^2} \quad (71)$$

Simplificando a notação, temos:

$$Y_x = \varphi_{xx} \quad (72)$$

Podemos escrever a equação (21) como um sistema linear da seguinte forma.

$$\begin{bmatrix} \varphi_x \\ Y_x \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{m\omega^2}{\hbar^2} \cdot (mx^2 - 2E) & 0 \end{bmatrix} \cdot \begin{bmatrix} \varphi \\ Y \end{bmatrix} - \frac{2m}{\hbar^2} \begin{bmatrix} 0 \\ TA \end{bmatrix} \quad (73)$$

Observe que TA significa o termo de Adomian da respectiva recursão.

Com a equação (21) escrita como sistema linear, podemos utilizar a técnica descrita em [20], que via processos quânticos realiza a seguinte operação $|x\rangle = (I - A)^{-1}|b\rangle$, ou seja, é aplicada a matriz quadrada 2x2 modificada somente ao termo de Adomian.

O artigo para resolver equações lineares [20] é complexo, pois, em muitas etapas ele delega uma sub-rotina a outro artigo, que muitas vezes é muito técnico. Por isso, este trabalho sugere para estudos futuros, talvez em uma tese de doutorado, um desenvolvimento do estudo sobre o algoritmo [20] e o [18] aprofundando em todos os aspectos do algoritmo.

Contudo, o interessante dessa abordagem é que ela utiliza um paralelismo quântico para atuar, porque ela não utiliza as variáveis isoladamente, mas sim em um estado de superposição.

5 CONCLUSÃO

5.1 OBJETIVOS ALCANÇADOS

O método de Adomian para resolução de equações não lineares foi apresentado e aplicado em três casos, cada qual apresentou suas próprias dificuldades, principalmente no aspecto de multiplicação entre os termos das recursões, contudo, se a solução da equação de inicialização for uma exponencial e o termo não linear for exponencial, ficou evidente que o método é útil, uma vez que os termos não homogêneos das equações lineares serão exponenciais.

Quanto aos algoritmos, apresentamos um rápido material teórico sobre o assunto para relacionar aos quânticos. Já neste tópico o que ficou claro é a complexidade e a necessidade de adaptação de um algoritmo, ou a criação de um totalmente novo para pequenas modificações do problema inicial estudado pelo algoritmo já existente.

Também fora realizado com sucesso a apresentação de um algoritmo quântico para atuar em conjunto com o método de Adomian, assim, o segundo objetivo foi satisfeito. Isto é, encontramos a possibilidade de utilizar um algoritmo quântico para a obtenção das soluções de equações não lineares. Contudo, o aspecto inicial de desenvolver um algoritmo que utilizasse as soluções das recursões como um todo em interações de superposição para buscar um resultado, isto é, sem precisar de uma solução em particular, ficou em aberto para um desenvolvimento futuro.

Destacamos que a técnica de utilização de constantes com índices para relacioná-las com as exponenciais, fora desenvolvido neste trabalho, e que apesar de ser limitada, ela foi essencial para desenvolver um algoritmo capaz de obter as soluções das recursões.

5.2 DIFICULDADES

5.2.1 APLICAÇÃO DO MÉTODO DE ADOMIAN

Apesar da proposta do método – resolver equações não lineares – ser interessante, ela apresenta alguns problemas a serem mais bem estudados. O primeiro encontrado na seção 3.3.2 foi a respeito do significado das constantes das equações de vínculo que surgiram nas recursões, principalmente o “ ω ”. Conseguimos nos afastar dessa situação ignorando-a, a fim de focar na parte matemática do método. Assim, o método acaba

virando uma maneira de simular um sistema físico para diversos casos, contudo com pouco, ou nenhum significado além do quantitativo.

O segundo aspecto problemático encontrado ainda em 3.3.2, foi a respeito da normalização, uma vez que reduzimos o problema a obter funções de onda para dados valores de constante. O problema se deve que ao normalizar, isto é, $\int_{-\infty}^{\infty} \psi \cdot \psi^* dx = 1$, surge um polinômio em função de $A_{0,1}$ e $A_{0,1}^*$, assim para encontrar a raiz teríamos um polinômio de duas variáveis complexas conjugadas. Por isso, para resolver a situação, optamos por escolher somente raízes reais, ou seja, $A_{0,1} = A_{0,1}^*$. O que resulta na pergunta: quem garante que sempre haverá tais constantes de normalização?

Como não estávamos interessados no estudo mais aprofundado neste aspecto, não insistimos na busca de uma solução para o problema, uma vez que o resultado real já nos é suficiente para um futuro estudo de comparação de eficiências entre o algoritmo clássico e o quântico de Adomian.

Na aplicação da seção 3.3.3, surge mais uma complicação para o método, quando a inicialização é resultado de uma soma de dois termos, porque ao desenvolver as próximas recursões é rápido o crescimento de termos e de complexidade. Além de que junto ao problema das constantes complexas, encontraremos problemas de quatro variáveis, a saber: A, B, A^*, B^* .

Apesar da dificuldade não solucionada, a seção foi útil para apresentar a aplicação da técnica com funções de onda complexas e demonstrar como simplificar o problema utilizando-se de simetrias.

Já na seção 3.3.4 é estudado o aspecto de haver mais de uma solução para a equação diferencial e isso poderá contribuir num estudo futuro, uma vez que seguir por uma abordagem levará para certas dificuldades e outra abordagem a outras. Este trabalho, limitou-se a observar o desenrolar de uma abordagem somente. Contudo, ao pensar sobre qual abordagem demonstrar, ficou claro que a outra teria um aumento no expoente do fator “x” que acompanha a exponencial em cada recursão, sendo assim, pressupondo que seria uma dificuldade a mais, optamos seguir diretamente pela outra abordagem.

5.2.2 ALGORITMOS QUÂNTICOS

Ao estudar algoritmos quânticos, observamos uma certa incongruência entre as alegações da capacidade dos computadores quânticos que inclusive alguns artigos fizeram e a realidade. Quer dizer, alguns artigos confundiram suposições com comprovações,

outros alegaram diferenças entre algoritmos quânticos e clássicos que não existem, como dizer que “n” bits não têm 2^n estados, enquanto “n” qubits têm – na verdade, os dois têm, é eletrônica digital básica [21], também há a situação que o artigo estuda um problema específico e utiliza certas informações de outro artigo, contudo, utiliza de forma imprecisa, como no caso do 2^n estados citado, pois, isso é uma discussão bem mais profunda ao se estudar o computador de Alan Turing aonde há diferença entre fita registradora e de processamento.

Além da dificuldade na precisão terminológica, observamos dificuldades como encontrar um algoritmo para uma tarefa específica; como limitação de um algoritmo a um espectro muito pequeno do problema maior; como algoritmos com passo-a-passo sem explicações; como falta de explicação do algoritmo e dificuldades menores.

5.3 SUGESTÕES DE ESTUDOS FUTUROS

5.3.1 APLICAÇÃO DO MÉTODO

Como a discussão sobre as dificuldades já fora feita na seção 5.2, sugeriremos em uma lista com uma pequena argumentação cada aspecto de estudos futuros do tópico.

- Aspecto físico das constantes dos problemas.
- Abordagem quanto a convergência e qualidade quantitativa do método.
- Técnica para encontrar as raízes de um polinômio multivariável, sendo um, ou mais pares de variáveis complexas conjugadas.
- Como melhorar o algoritmo do Matlab para testar a função de onda obtida e obter a normalização.

5.3.1 ALGORITMOS QUÂNTICOS

Há somente duas sugestões de estudos futuros, desenvolver a explicação de forma detalhada do algoritmo quântico [20] para melhor análise e testá-lo num simulador, ou nos “supostos” computadores quânticos das empresas privadas.

REFERÊNCIAS

- [1] G. Fibich, *The Nonlinear Schrödinger*, Switzerland: Springer, 2015.
- [2] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Netherlands : Kluwer Academic Publishers , 1994.
- [3] O. Gonzalez-Gaxiola, “The Laplace-Adomian Decomposition Method Applied to the Kundu-Eckhaus Equation,” 25 abril 2017.
- [4] A. Jaradat, M. Gharaibeh, A. Obeidat e M.-K. Qaseer, “Adomian Decomposition Approach to Solve the Simple Harmonic Quantum Oscillator,” *International Journal of Applied Engineering Research*, vol. 13, nº 2, pp. 1056 - 1059, 2018.
- [5] C. A. Ladeia, *Sobre Adomian e a convergência*, Porto Alegre: TESE NA UFRGS, 2016, pp. 33-40.
- [6] W. F. Trench, “O principio da superposição,” em *ELEMENTARY DIFFERENTIAL EQUATIONS*, San Antonio, Free Edition 1.01, 2013, pp. 225-227.
- [7] D. J. Griffiths, “Exemplo 2.4,” em *Mecânica Quântica*, São Paulo, Pearson, 2011, p. 35.
- [8] S. Lipschutz e M. L. Lipson, “Capítulo 3,” em *Matemática Discreta*, Porto Alegre, Bookman, 2013, pp. 42-68.
- [9] R. Portugal, “Grover em Português,” em *Uma Introdução à Computação Quântica*, São Carlos, SBMAC, 2012, pp. 37-62.
- [10] R. Portugal, “Shor em português,” em *Uma Introdução à Computação Quântica*, São Carlos, SBMAC, 2012, pp. 63-85.
- [11] C. d. Figueiredo, G. d. Fonseca, M. Lemos e V. d. Sá, “Introdução aos Algoritmos Randomizados,” em *26º Colóquio Brasileiro de Matemática*, Rio de Janeiro, 2007.
- [12] P. Kaye, R. Laflamme e M. Mosca, *An Introduction to Quantum Computing*, New York: OXFORD university press, 2007.
- [13] H. & S. R. Buhrman, “Quantum Verification of Matrix Products,” 2004.
- [14] D. Deutsch e R. Jozsa, “Rapid Solution of Problems by Quantum Computation,” *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, nº 1907, pp. 553-558, 1992.
- [15] S. Jordan, “Quantum Algorithm Zoo,” [Online]. Available: <http://quantumalgorithmzoo.org/>. [Acesso em 25 abril 2019].
- [16] T. Tansuwannont, S. Limkumnerd, S. Suwanna e P. Kalasuwan, “Quantum Phase Estimation Algorithm for Finding Polynomial Roots,” arXiv, 15 outubro 2015. [Online]. Available: arXiv:1510.04452. [Acesso em 25 abril 2019].
- [17] K. N. T. G. H. e. a. Nagata, “Efficient Quantum Algorithms of Finding the Roots of a Polynomial Function,” *International Journal of Theoretical Physics*, pp. 2546-2555, 18 Maio 2018.
- [18] S. K. LEYTON e T. J. OSBORNE, “A quantum algorithm to solve nonlinear differential equations,” arXiv.org, 2008.
- [19] D. W. Berry, “High-order quantum algorithm for solving linear differential equations,” *Journal of Physics A: Mathematical and Theoretical*, vol. 47, 19 fevereiro 2014.

- [20] A. W. Harrow, A. Hassidim e S. Lloyd, “Quantum algorithm for solving linear systems of equations,” *Phys. Rev. Lett*, vol. 103, n° 15, p. 150502, 7 outubro 2009.
- [21] S. T. Karris, *Digital Circuit Analysis and Design with Simulink Modeling and Introduction to CPLDs and FPGAs*, Orchard Publications, 2007.

APÊNDICE 1: MEMORIAL DESCRITIVO DO PRIMEIRO

CÓDIGO DE MATLAB.

Iniciamos dizendo qual recursão desejamos, e fornecemos os parâmetros $(E, \frac{\hbar^2}{2m}, c)$.

O limite do código é até a terceira recursão.

Então, o código cria as constantes $A_{i,j}$ em termos de $A_{0,1}$, fazendo um valor numérico vezes uma potência de $A_{0,1}$. Cria-se uma matriz contendo as constantes para trabalhar-se com elas.

Tendo as constantes podemos fazer $\int \psi_0 \cdot \psi_0^* dx = 1$ sendo o ψ_0 a função de onda truncada até a recursão desejada. Então, cria-se um vetor “f” que é o polinômio de potências de “ $A_{0,1}$ ”, isto é, “ $A_{0,1} = y$ ” resultante da integral de normalização em toda posição menos “1”.

Por exemplo, para a primeira recursão temos:

$$0,5221y^6 + 1,7904y^4 + 1,7725y^2 - 1 = f$$

É importante destacar que por esse método as raízes procuradas são somente as reais $A_{0,1} = A_{0,1}^*$

Agora achamos as raízes de “f” via comando “root” e salvamos num vetor. Depois achamos o erro do processo em procurar as raízes por ter usado o comando “root”. Para fazer isso, elevamos a raiz aos expoentes necessários e multiplicamos pela constante respectiva do expoente. Assim, se o resultado for zero é exato, sendo o erro a divergência desse valor. Nos testes, o erro chegou à ordem de 10^{-4} para a terceira recursão e 10^{-15} para a primeira recursão. Por fim, esse primeiro código salva diversos valores para um segundo código de análises e o primeiro código ainda gera um gráfico com os erros e as raízes.

Observamos que os dois códigos são unidos num só, porém há uma quebra clara no meio do código explicitando o início do segundo. Isso aconteceu assim, pois, eles foram criados de forma independentes inicialmente.

APÊNDICE 2: MEMORIAL DESCRITIVO DO SEGUNDO CÓDIGO DE MATLAB.

Este código tem o intuito de analisar a função de onda e apresentar resultados. Ele inicia lendo os parâmetros fornecidos pelo primeiro código. Então, seleciona somente as raízes reais, uma vez que no primeiro código assumimos que as raízes eram reais

Assim, as raízes $A_{0,1}$ para os valores $E=1$ $E_0=1$ e $C=1$ são $A_{0,1} \begin{cases} = -0,6521 \\ = 0,6521 \end{cases}$

Após selecionar somente as raízes reais, o código cria um vetor para as exponenciais num dado valor de x – escolhemos de -2 a 2 com o passo de $0,1$. Então, ele utiliza um bloco para associar os valores a cada constante, uma vez que uma exponencial pode ter uma soma de constantes de diferentes recursões. Com esses valores, o código passa para criar a função de onda associando a constante ao exponencial. Também cria um vetor módulo da função de onda.

Após, o código inicia o bloco de plotagem, cria-se um gráfico da amplitude da função de onda para os parâmetros fornecidos. Também um gráfico das funções de onda. Por fim, deseja-se saber o peso de cada exponencial, por isso, cria-se uma matriz “constantes” que determina para cada linha a raiz que se trata e para cada coluna a contribuição de cada exponencial. Após, plotasse esse vetor.

ANÁLISE DE RESÍDUO

Para realizar a análise de resíduo, no segundo código fora criado um adendo, partimos do pressuposto que a segunda derivada de exponenciais do tipo e^{ax^2} resultam em: $2ae^{ax^2}(2ax^2 + 1)$

Então, os termos contendo x quadrático anularam-se com os termos do potencial harmônico, devido as equações de vínculo das “frequências”. A partir daí, ficamos somente com $-E_0\psi_m - E\psi = c\psi^3$ sendo o índice “m” indicando uma modificação, aonde toda exponencial terá uma constante multiplicativa de “2a”, sendo “a” a constante que acompanha o “ x^2 ”. Por exemplo, na primeira recursão: $\psi_m = E_0A_{0,1}e^{-\frac{x^2}{2}} + 3E_0A_{1,3}e^{-\frac{3x^2}{2}}$.

APENDICE 3: CÓDIGOS

4.1 – CÓDIGO 1 – UNIDO

Escolhe-se a recursão e os valores iniciais do problema para então mostrar as possíveis constantes de normalização, o erro de cada uma, a função de onda e o peso de cada exponencial na recursão.

```
clear all
close all

%bloco de constantes iniciais colocadas.
Co4=1; %da equação
E=1; %energia
E0=1;%|h^2 /2m
re=3; %re=1 inicialização

Cp=1.77245385090551588191942755657; %pi
total=0; %total de bases.
sum=zeros(re-1,1); %numero de termos na f.

for k=0:1:re-2
    sum(k+1)=3^k;
    total=sum(k+1)+total;
end

%bloco para gerar as constantes de cada base.
Ca=zeros(1,(2*total+1));
for i=1:1:(2*total+1)
    Ca(i)=vpa(sqrt(i),20);
end

C=Cp./Ca; %constantes num vetor Cp=sqrt(pi) C2=sqrt(2)

%constantes de cada base que multiplicam uma potência de A01.
CA13=Co4/(3*E0-E); %A01^3
CA25=3*Co4*CA13/(5*E0-E); %A01^2
CA27=3*Co4*CA13^2/(7*E0-E); % A01
CA29=Co4*CA13^3/(9*E0-E);
CA37=3*Co4*CA25/(7*E0-E); %A01^2
CA39=(6*CA25*CA13 +3*CA27)*Co4/(9*E0-E); %A01^1 e A01^2
CA311=(3*CA29+3*CA25^2 +6*CA13*CA27)*Co4/(11*E0-E); %A01^2, A01^1 e
A01^1
CA313=(3*(CA25^2)*CA13 +6*CA13*CA29+6*CA25*CA27)*Co4/(13*E0-E); %
A01^0 A01^1 e A01^1
CA315=((CA25^3) +6*CA25*CA29+3*CA27^2+6*CA25*CA27*CA13)*Co4/(15*E0-E);
% A01^0 A01^1 A01^1 A01^0
CA317=(3*CA27*(CA25^2)+6*CA27*CA29+(3*CA27^2 *
CA13)+6*CA25*CA29*CA13)*Co4/(17*E0-E);
CA319=(6*CA27*CA29*CA13+3*CA29^2+3*CA25^2 *CA29 +
3*CA25*CA27^2)*Co4/(19*E0-E);
CA321=(3*CA29^2 * CA13 +CA27^3 +6*CA25*CA27*CA29)*Co4/(21*E0-E);
CA323=(3*CA27^2 *CA29 +3*CA25 *CA29^2)*Co4/(23*E0-E);
```

```

CA325=(3*CA27*CA29^2)*Co4/(25*E0-E);
CA327=CA29^3*Co4/(27*E0-E);

CA=zeros(re,total+1);
CA(1,1)=1;
CA(2,2)=CA13;
CA(3,3)=CA25;
CA(3,4)=CA27;
CA(3,5)=CA29;
CA(4,4)=CA37;
CA(4,5)=CA39;
CA(4,6)=CA311;
CA(4,7)=CA313;
CA(4,8)=CA315;
CA(4,9)=CA317;
CA(4,10)=CA319;
CA(4,11)=CA321;
CA(4,12)=CA323;
CA(4,13)=CA325;
CA(4,14)=CA327;

index=(2*(2*total+1)+1); %ordem do polinômio +1
f1=zeros(1,index);
f2=zeros(1,index);
f=zeros(1,index);

%bloco para montar a função que representa a constante de
normalização,
%onde a raiz será uma possível constante A0
f1(1,index)=-1;
for i=re:-1:1
    for j=(total+1):-1:1
        f1(4*(total+2-j)-3)=(CA(i,j).^2).*C((2*j)-1)+f1(4*(total+2-j)-
3);
            for w=i:-1:1
                for k=j-1:-1:1;
                    f2((4*(total+1)-
2*(j+k))+1)=2.*CA(i,j).*CA(w,k).*C(j+k-1)+ f2((4*(total+1)-
2*(j+k))+1);
                end
            end
        end

    end
    f=f1+f2;

r=roots(f);
ep=zeros(1,length(r-1));
for i=1:1:(length(r-1))/2 %Loop para criar expoentes
    ep(2*i)=2*i;
end

for j=1:1:length(f)-1 %calcular erro em modulo da raiz

    x(j)=vpa(r(j),20);
    erro(j)=abs([1,vpa((x(j).^ep(2:2:(length(f)-
1))))],20])*transpose(vpa(f((length(f):-2:1),10)));

end

```

```

Tf=length(f);

%bloco de plotagem raizes
plot(r(1:(length(f)-1)), 'x')
title(['E0=' num2str(E0) ' ', 'E=' num2str(E)], ['re=' num2str(re) ' '
', 'Cons da não line=' num2str(Co4)])
xlabel('parte real')
ylabel('parte imaginária')

figure
plot(vpa(erro(1:(length(f)-1)),20), 'x')
ylabel('amplitude de erro para uma dada raiz')
xlabel('raiz')
title(['E0=' num2str(E0) ' ', 'E=' num2str(E)], ['re=' num2str(re) ' '
', 'Cons da não line=' num2str(Co4)])

%O que seria o segundo código inicia aqui

cont=0;
n=total+1; %número de bases somando todas as recursões

%bloco para pegar somente as raizes reais.
for i=1:1:(Tf-1)
    if ((imag(r(i)))==0);
        cont=cont+1;
        pos(cont)=i;
        real(cont)=r(i);
        reshape(repmat(real,1,1),1,[]);
        reshape(repmat(pos,1,1),1,[]);
    end
end

% bloco da posição e exponenciais.
x=-2:0.1:2;

for i=1:1:n %Loop para criar vetor com exponenciais para cruzar depois
e(i,1:1:length(x))=exp(-(i-0.5).*x.^2);
end

%bloco das constantes que acompanham cada exponencial
Tr=length(real);
A=zeros(re,n,Tr);
A(1,1,(1:1:Tr))=vpa(real(1:1:Tr),20);

for i=1:1:Tr %página=raiz
    for j=2:1:re %nível de recursão
        for k=2:1:n %energia
            A(j,k,i)=vpa(A(1,1,i).^(2.*k-1) *CA(j,k),20);
        end
    end
end

K=zeros(n+1,length(x),Tr); %vetor da soma de exponenciais * constantes
Y=zeros((Tr),length(x)); %função de onda

```

```

for i=1:1:Tr %cada raiz
    TRA=zeros(re+1,length(x));
    for j=1:1:re %cada recursão
        for k=1:1:n; %numero de energias

K(k+1,1:1:length(x),i)=vpa(A(j,k,i).*e(k,1:1:length(x))+K(k,1:1:length
(x),i),20);
        end

TRA(j+1,1:1:length(x))=K(n+1,1:1:length(x),i)+TRA(j,1:1:length(x));
        end
Y(i,1:1:length(x))=vpa(TRA(j+1,1:1:length(x)),30);
Y2=vpa(Y.*conj(Y),30);
end

%bloco de plotagem

figure
for j=1:1:Tr
    hold on
    plot(x,Y2(j,1:1:length(x)))
    Legend{j}={['A01=' num2str(double(real(j)))]};
end

title({'E0=' num2str(E0) ' ', 'E=' num2str(E) }, ['re=' num2str(re) '
', 'Cons da não line=' num2str(Co4) ]})
xlabel('X')
ylabel('| | ^2')
legend([Legend])
hold off

figure
plot(x,Y)
title({'E0=' num2str(E0) ' ', 'E=' num2str(E) }, ['re=' num2str(re) '
', 'Cons da não line=' num2str(Co4) ]})
legend(Legend)
xlabel('X')

Constantes=zeros(Tr,n);
for i=1:1:Tr
    for j=1:1:re
        for k=1:1:n
            Constantes(i,k)=A(j,k,i)+Constantes(i,k);
        end
    end
end
end
figure

for i=1:1:Tr
    hold on
    stem(linspace(1,2*n-1,n),Constantes(i,1:1:n))
end
grid on
legend(Legend)
title('constantes por expoente')
xlim([0 2*n])
hold off

```

```

restotal=zeros(Tr,length(x)); %residuo (para raiz, para x)

for i=1:1:Tr %colocar Tr
    residuo=zeros(length(x),n+1); %residuo (para x, para exponencial)
    for p=1:1:length(x)
        for k=1:1:n

            residuo(p,k+1) = Constantes(i,k).*e(k,p).(E0.*(2*k-1)-
E)+residuo(p,k);

            end
            restotal(i,p)=residuo(p,n+1)-Co4.*Y(i,p).^3;
        end

        %Constantes(i,k).*((E0.*k.^2 +consteste).*(x.^2 -1) -E0.*k -
E).*e(k,1:1:length(x))...
        % -Co4.*Constantes(i,k).^3 .* (e(k,1:1:length(x))).^3;

    end
end

```

4.2 – CÓDIGO 2 – ANÁLISE DE RESÍDUO

Este código serve para comparar os resíduos para diferentes constantes iniciais.

```

clear all
U=zeros(5,4,2);
Cp=1.77245385090551588191942755657; %pi
x=-2:0.1:2;
for q=1:1:4
    re=q; %re=1 inicialização

    total=0; %total de bases.
    sum=zeros(re-1,1); %numero de termos na f.

    for k=0:1:re-2
        sum(k+1)=3^k;
        total=sum(k+1)+total;
    end

    %bloco para gerar as constantes de cada base.
    Ca=zeros(1,(2*total+1));
    for i=1:1:(2*total+1)
        Ca(i)=vpa(sqrt(i),20);
    end

    C=Cp./Ca; %constantes num vetor Cp=sqrt(pi) C2=sqrt(2)
    index=(2*(2*total+1)+1); %ordem do polinômio +1

    t_amostras=10;

    for ee=1:1:t_amostras %aqui faz o controle do E e E0

        %bloco de constantes iniciais colocadas.
        Co4=0.1;
        E0=1; %energia
    end
end

```

```

E=ee.*E0%|h^2 /2m
% até aqui vai o controle

f1=zeros(1,index);
f2=zeros(1,index);
f=zeros(1,index);

%constantes de cada base que multiplicam uma potência de A01.
CA13=Co4/(3*E0-E); %A01^3
CA25=3*Co4*CA13/(5*E0-E); %A01^2
CA27=3*Co4*CA13^2/(7*E0-E); % A01
CA29=Co4*CA13^3/(9*E0-E);
CA37=3*Co4*CA25/(7*E0-E); %A01^2
CA39=(6*CA25*CA13 +3*CA27)*Co4/(9*E0-E); %A01^1 e A01^2
CA311=(3*CA29+3*CA25^2 +6*CA13*CA27)*Co4/(11*E0-E); %A01^2, A01^1 e
A01^1
CA313=(3*(CA25^2)*CA13 +6*CA13*CA29+6*CA25*CA27)*Co4/(13*E0-E); %
A01^0 A01^1 e A01^1
CA315=((CA25^3) +6*CA25*CA29+3*CA27^2+6*CA25*CA27*CA13)*Co4/(15*E0-E);
% A01^0 A01^1 A01^1 A01^0
CA317=(3*CA27*(CA25^2)+6*CA27*CA29+(3*CA27^2 *
CA13)+6*CA25*CA29*CA13)*Co4/(17*E0-E);
CA319=(6*CA27*CA29*CA13+3*CA29^2+3*CA25^2 *CA29 +
3*CA25*CA27^2)*Co4/(19*E0-E);
CA321=(3*CA29^2 * CA13 +CA27^3 +6*CA25*CA27*CA29)*Co4/(21*E0-E);
CA323=(3*CA27^2 *CA29 +3*CA25 *CA29^2)*Co4/(23*E0-E);
CA325=(3*CA27*CA29^2)*Co4/(25*E0-E);
CA327=CA29^3*Co4/(27*E0-E);

CA=zeros(re,total+1);
CA(1,1)=1;
CA(2,2)=CA13;
CA(3,3)=CA25;
CA(3,4)=CA27;
CA(3,5)=CA29;
CA(4,4)=CA37;
CA(4,5)=CA39;
CA(4,6)=CA311;
CA(4,7)=CA313;
CA(4,8)=CA315;
CA(4,9)=CA317;
CA(4,10)=CA319;
CA(4,11)=CA321;
CA(4,12)=CA323;
CA(4,13)=CA325;
CA(4,14)=CA327;

f1(1,index)=-1;
for i=re:-1:1
    for j=(total+1):-1:1
        f1(4*(total+2-j)-3)=(CA(i,j).^2).*C((2*j)-1)+f1(4*(total+2-j)-
3);
        for w=i:-1:1
            for k=j-1:-1:1;
                f2((4*(total+1)-
2*(j+k))+1)=2.*CA(i,j).*CA(w,k).*C(j+k-1)+ f2((4*(total+1)-
2*(j+k))+1);
            end
        end
    end
end
end

```

```

end
f=f1+f2;
clear r Tf
r=roots(f);
ep=zeros(1,length(r-1));
for i=1:(length(r-1))/2 %Loop para criar expoentes
ep(2*i)=2*i;
end

Tf=length(f);

cont=0;
n=total+1; %número de bases somando todas as recursões

%bloco para pegar somente as raízes reais.

for i=1:(Tf-1)
    if ((imag(r(i)))==0);
        cont=cont+1;
        pos(cont)=i;
        real(cont)=r(i);
        reshape(repmat(real,1,1),1,[]);
        reshape(repmat(pos,1,1),1,[]);
    end
end

% bloco da posição e exponenciais.

for i=1:1:n %Loop para criar vetor com exponenciais para cruzar depois
e(i,1:length(x))=exp(-(i-0.5).*x.^2);
end

clear i

%bloco das constantes que acompanham cada exponencial
Tr=length(real);
A=zeros(re,n,Tr);

A(1,1,(1:1:Tr))=vpa(real(1:1:Tr),20);

for i=1:1:Tr %página=raiz
    for j=2:1:re %nível de recursão
        for k=2:1:n %energia
            A(j,k,i)=vpa(A(1,1,i).^(2.*k-1) *CA(j,k),20);
        end
    end
end

clear i j k

K=zeros(n+1,length(x),Tr); %vetor da soma de exponenciais * constantes
Y=zeros((Tr),length(x)); %função de onda

for i=1:1:Tr %cada raiz
    TRA=zeros(re+1,length(x));
    for j=1:1:re %cada recursão
        for k=1:1:n; %numero de energias

```

```

K(k+1,1:1:length(x),i)=vpa(A(j,k,i).*e(k,1:1:length(x))+K(k,1:1:length
(x),i),20);
    end

TRA(j+1,1:1:length(x))=K(n+1,1:1:length(x),i)+TRA(j,1:1:length(x));
    end
Y(i,1:1:length(x))=vpa(TRA(j+1,1:1:length(x)),30);
Y2=vpa(Y.*conj(Y),30);
end
clear i j k

%bloco de plotagem

Constantes=zeros(Tr,n);
for i=1:1:Tr
    for j=1:1:re
        for k=1:1:n
            Constantes(i,k)=A(j,k,i)+Constantes(i,k);
        end
    end
end

restotal=zeros(Tr,length(x)); %residuo (para raiz, para x)

for i=1:1:Tr %colocar Tr
    residuo=zeros(length(x),n+1); %residuo (para x, para exponencial)
    for p=1:1:length(x)
        for k=1:1:n

            residuo(p,k+1) = Constantes(i,k).*e(k,p).(E0.*(2*k-1)-
E)+residuo(p,k);

            end
            restotal(i,p)=residuo(p,n+1)-Co4.*(Y(i,p)).^3;

        end

    end

end

[maiorvalor,posicaomaiorvalor]=max(abs(restotal),[],2);
maiorvalor;
'x=';
x(posicaomaiorvalor);

U(ee,q,1)=maiorvalor(1);
U(ee,q,2)=maiorvalor(2);
U2(ee,q)=U(ee,q,2);

    end
end

%variável com os valores para as tabelas 1 a 5
U2

```