

Computação

Utilização do ChatGPT na aprendizagem da programação em linguagem Matlab

Xavier Romão

Setembro 2023

1. O que é o ChatGPT?

O ChatGPT é uma aplicação computacional de inteligência artificial, desenvolvida pela empresa OpenAI, que funciona como um *chatbot* (uma aplicação computacional com a qual um utilizador pode manter uma conversa em tempo real por texto ou por voz). O utilizador do ChatGPT pode fazer-lhe perguntas ou dar-lhe tarefas, e ele irá fornecer uma resposta (**que poderá ou não ser a resposta correta** como será discutido mais adiante). O ChatGPT foi especificamente concebido para processar linguagem e utiliza técnicas de aprendizagem automática (*machine learning*) para compreender e responder a pedidos feitos em linguagem natural. Embora este tipo de *chatbot* já exista há algum tempo, o ChatGPT foi o primeiro a ser disponibilizado para a generalidade dos utilizadores de forma gratuita.

O modelo de inteligência artificial do ChatGPT foi treinado com base numa enorme quantidade e variedade de textos disponíveis na Internet, os quais incluíram textos com linguagens de programação. Com base nas técnicas de aprendizagem automática utilizadas, o modelo de inteligência artificial do ChatGPT foi capaz de identificar padrões e associações de palavras dentro de vários idiomas, o que lhe permite fazer previsões sobre a sequência de palavras que tem maior probabilidade de ser relevante após uma qualquer sequência inicial de palavras, como por exemplo uma pergunta feita pelo utilizador. Este modelo de aprendizagem, que permite ao ChatGPT adquirir uma compreensão abrangente da estrutura e da sintaxe de uma língua, confere-lhe, igualmente, a possibilidade de interpretar e gerar instruções numa linguagem de programação com base na interação com o utilizador.

Desta forma, a utilização do ChatGPT pode trazer benefícios ao processo de aprendizagem da linguagem de programação MATLAB funcionando como um assistente em tempo real que poderá responder a perguntas sobre conceitos específicos da linguagem MATLAB, explicar a lógica de funcionamento de determinados comandos, dar exemplos de código para resolver certos problemas, ou ajudar na identificação de erros em programas que lhe sejam fornecidos. Apesar destas potencialidades, é fundamental não esquecer que o ChatGPT é uma ferramenta que **fornece respostas que não são 100% fiáveis**. De modo a ilustrar estas potencialidades, mas também alertar para os vários problemas que podem surgir durante a utilização do ChatGPT neste contexto, serão discutidos vários exemplos que mostram várias situações que poderão ocorrer.

2. Como utilizar o ChatGPT?

Uma das vantagens do ChatGPT é ser fácil de usar. A utilização do ChatGPT requer uma conta de utilizador, a qual é facilmente criada ao aceder ao *site* da OpenAI (<https://openai.com/>) e à opção *Sign up*. A conta de utilizador pode ser criada utilizando uma conta de email existente ou a partir de uma conta da Google, Microsoft ou Apple (Figura 1). Após criar a conta, o utilizador tem acesso aos vários serviços da OpenAI (ChatGPT, DALL-E e à interface de comunicação entre estas aplicações e outras aplicações externas disponibilizadas pela OpenAI). Escolhendo o ChatGPT, o utilizador tem acesso à interface representada na Figura 2 a partir da qual o utilizador comunica com o ChatGPT. Na versão atual, esta interface fornece alguma informação acerca das capacidades e limitações do ChatGPT que convém ter em mente de modo a maximizar a experiência de utilização.

O processo de utilização do ChatGPT pode ser descrito através dos passos seguintes:

- Inicialmente, o utilizador deve inserir uma mensagem de texto (conhecido como *prompt*) na interface do ChatGPT. Essa mensagem pode ser uma pergunta, um pedido de informação sobre um assunto ou mesmo uma afirmação. Embora o ChatGPT seja principalmente utilizado em inglês e forneça melhores resultados nesse idioma, ele também pode ser usado noutras línguas, nomeadamente em português.
- Seguidamente, o ChatGPT analisa o texto da mensagem inserida e utiliza o seu modelo de inteligência artificial para gerar uma resposta. Com base nos padrões e nas associações de palavras que o modelo de inteligência artificial desenvolveu na fase de treino, vai gerar uma resposta composta por uma sequência de palavras que tem maior probabilidade de ser relevante.
- Após gerar essa resposta, ela é apresentada ao utilizador no ecrã da interface.
- Posteriormente, o utilizador pode escrever uma nova mensagem de texto que pode ou não ser baseada na resposta anterior de modo a gerar um diálogo sobre um determinado tema que integre várias perguntas e respostas sequenciais. O ChatGPT irá posteriormente analisar e responder a esta nova mensagem, tendo em conta as mensagens anteriores e as respostas já fornecidas ao longo do diálogo.

Create your account

Note that phone verification may be required for signup. Your number will only be used to verify your identity for security purposes.

Continue

Already have an account? [Log in](#)

OR



Continue with Google



Continue with Microsoft Account



Continue with Apple

Figura 1 – Vista do ecrã de criação de conta de utilizador do ChatGPT.

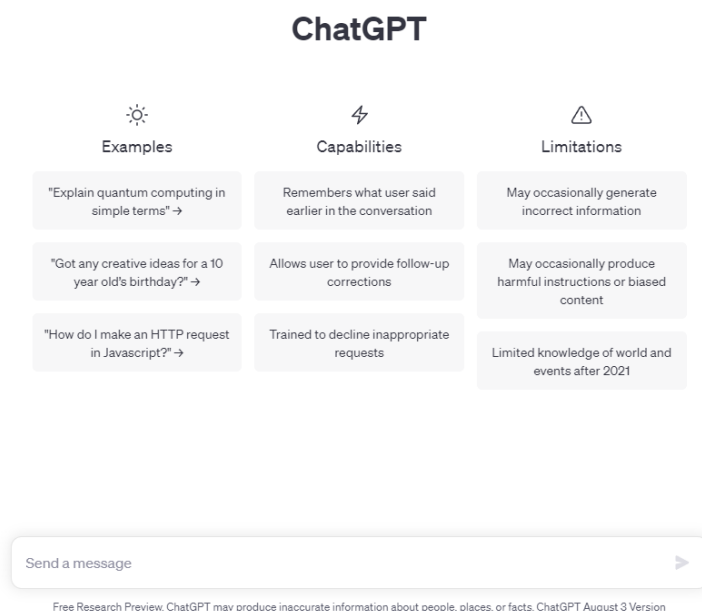


Figura 2 – Vista da interface de utilização do ChatGPT.

Atualmente, a OpenAI disponibiliza a versão 3.5 do ChatGPT de forma gratuita. Esta versão pode responder mais lentamente e estar inclusivamente indisponível em períodos de maior procura. Mediante um pagamento mensal, a OpenAI disponibiliza ainda a versão 4 do ChatGPT que possui um modelo de inteligência artificial mais avançado e mais fiável. Assim, salienta-se que as análises e conclusões apresentadas ao longo deste documento são limitadas a resultados obtidos com a versão 3.5 do ChatGPT, podendo, eventualmente, obter-se melhores resultados com a referida versão 4. Este documento não deve, portanto, ser encarado como um documento exaustivo sobre como utilizar o ChatGPT para programar em MATLAB, mas apenas como um guia introdutório ao tema.

3. Que aspetos a ter em conta, ao utilizar o ChatGPT para aprender a programar em MATLAB?

Como já referido, o ChatGPT pode ser útil em vários contextos relacionados com a aprendizagem da linguagem de programação MATLAB. No entanto, a utilização do ChatGPT neste contexto deve ter em conta os seguintes aspetos essenciais:

- A resposta fornecida pelo ChatGPT a uma determinada mensagem inserida pelo utilizador é a que o modelo de inteligência artificial considera como tendo maior probabilidade de ser relevante em função da referida mensagem, como já referido. No entanto, a resposta a uma determinada mensagem não é única. Se o utilizador inserir a mesma mensagem várias vezes, poderá obter respostas com conteúdos diferentes. Num contexto de aprendizagem da linguagem de programação MATLAB, isto pode querer dizer que uma mensagem que peça para explicar o funcionamento de um determinado programa fornecido pode resultar em respostas diferentes, ou que os programas gerados para resolver o mesmo problema pedido na mensagem podem também apresentar diferenças.
- A qualidade da resposta fornecida pelo ChatGPT é particularmente dependente do contexto e da especificidade da mensagem inserida pelo utilizador. De modo a potenciar a obtenção de respostas com maior qualidade e que vão ao encontro dos objetivos do utilizador, é

fundamental escrever a mensagem com o máximo de informação possível de modo a que o modelo de inteligência artificial do ChatGPT consiga extrair o contexto e a informação necessária para gerar a resposta. O processo de melhoria da mensagem para obtenção da resposta pretendida pode ser também, por vezes, conseguido através dum processo de diálogo em que uma sequência de perguntas e respostas vai convergindo iterativamente para o objetivo pretendido. Num contexto de aprendizagem da linguagem de programação MATLAB, isto pode querer dizer que a explicação sobre o funcionamento de um determinado programa fornecido pode não ter o detalhe pretendido, ou que o programa gerado para resolver um problema pedido na mensagem não está a responder ao que é pretendido porque faltam detalhes na mensagem.

- Dado que o **ChatGPT pode fornecer respostas erradas**, é fundamental ter um espírito crítico acerca do conteúdo dessas respostas para avaliar a sua fiabilidade. De modo a facilitar a interpretação das respostas num contexto de aprendizagem da linguagem de programação MATLAB, é importante acompanhar a interação com o ChatGPT com a utilização do Matlab para, por exemplo, verificar os programas que são gerados pelo ChatGPT ou confirmar a interpretação que fornece de programas inseridos na mensagem.
- Apesar do ChatGPT poder ser utilizado em português, as respostas que fornece neste caso são na maioria dos casos escritas em português do Brasil. Mesmo referindo na mensagem que a resposta deverá vir em português de Portugal, esta muitas vezes continua a ser escrita em português do Brasil.

4. Exemplos de utilização do ChatGPT para auxiliar a aprendizagem da programação em linguagem MATLAB

De modo a ilustrar a utilização do ChatGPT em vários cenários relacionados com a aprendizagem da linguagem de programação MATLAB, apresentam-se em seguida vários exemplos, incluindo situações em que o **ChatGPT não fornece a resposta correta**. Nestes casos, são discutidas algumas estratégias para identificar os erros e minimizar o seu impacto. Estes exemplos têm como principal objetivo não só salientar as vantagens da utilização do ChatGPT nos exemplos abordados, mas também alertar para os vários problemas que podem surgir durante essa utilização.

4.1 Utilizar o ChatGPT para entender o funcionamento de um programa em MATLAB

4.1.1 Primeiro exemplo – mensagens em inglês

O primeiro exemplo discutido corresponde ao programa apresentado na Figura 3 (Exemplo #1). De modo a ilustrar a variedade de respostas que podem ser obtidas através do ChatGPT para explicar o funcionamento deste programa, foram utilizadas várias mensagens em inglês e foi também repetida uma mesma mensagem várias vezes.

```
a = -2*ones(2,1);  
b = [-3:3:1]';  
z = (a + ~b) & (a < b) - 1
```

Figura 3 – Código MATLAB do Exemplo #1.

Na primeira tentativa de obter uma explicação para o funcionamento do Exemplo #1 utilizou-se a mensagem em inglês apresentada na Figura 4 e que resultou na explicação igualmente apresentada na mesma figura. Como se pode observar, a mensagem pede para descrever o que faz o programa, passo-a-passo. A resposta obtida descreve as várias operações efetuadas pelo programa de forma detalhada. No entanto, uma análise mais cuidada da resposta revela que os Passos 1 e 2 estão corretos, mas que o Passo 3 contém vários erros.

Verifica-se que o primeiro erro está na comparação $a < b$ quando o ChatGPT refere que o segundo elemento de a (-2) não é inferior ao segundo elemento de b (0). Apesar dos resultados seguintes estarem naturalmente errados devido a este erro, observa-se que existem outros erros que convém igualmente salientar. Por exemplo, quando apresenta o resultado errado da comparação $a < b$, o ChatGPT apresenta o resultado no formato **[false; false]** que não corresponde ao formato que o MATLAB utilizaria já que este converte os resultados lógicos em 0 e 1; ou seja o resultado com o formato correto seria **[0; 0]**. Em seguida, o ChatGPT apresenta ainda outro erro relacionado com a ordem com que executa as operações seguintes. A solução apresentada pelo ChatGPT assume que a comparação lógica $(a + \sim b) \& (a < b)$ é efetuada antes de subtrair o valor 1. No entanto, em linguagem MATLAB, após resolver as operações dentro dos parênteses, a operação aritmética (subtração) tem precedência sobre a comparação lógica (&). Assim, na sequência correta de operações em MATLAB, o valor 1 é subtraído ao resultado da comparação lógica $(a < b)$, sendo o consequente resultado numérico convertido em valores lógicos para ser depois comparado com o resultado de $(a + \sim b)$.

Finalmente, o ChatGPT apresenta ainda outro erro na operação final que apresenta em que é subtraído o valor 1 ao resultado da comparação lógica $(a + \sim b) \& (a < b)$. Admitindo que o resultado dessa operação lógica está correto e corresponde a **[false; false]**, ao subtrair o valor 1, o resultado correto seria **[-1; -1]** dado que um resultado lógico **[false; false]** seria representado por **[0; 0]** no formato MATLAB e convertido num vetor com valores numéricos para se subtrair o valor 1 aos vários elementos vetor.

Apesar da forma detalhada como o ChatGPT apresenta a descrição deste programa, esta análise revela que é fundamental confirmar os vários elementos dessa descrição. Essa confirmação pode, por exemplo, ser obtida através de uma leitura cuidada da resposta, mas pode ainda ser facilmente efetuada utilizando o MATLAB, testando os vários passos descritos pelo ChatGPT. Neste caso, o resultado correto obtido seria **[1; 0]**.

De forma a verificar se o ChatGPT seria capaz de detetar os erros que apresentou, escreveu-se uma nova mensagem focando o Passo 3 (Figura 5). Como se pode observar, nesta nova versão do Passo 3 o primeiro erro da explicação anterior foi corrigido, mas a ordem das operações seguintes continua a não estar correta. Numa última tentativa de chegar ao resultado correto, escreveu-se uma nova mensagem que apontava diretamente para uma verificação da precedência de operações (Figura 6). Mesmo apontando para esta questão e mesmo após várias repetições da mensagem (usando a opção *Regenerate*), a resposta final nunca foi a resposta correta, tendo sido inclusivamente obtidas outras

respostas com vários erros. Ao fim da 4ª tentativa, o ChatGPT voltou a devolver a resposta que tinha apresentado com a segunda mensagem (Figura 5).

Este exemplo ilustra as potencialidades do ChatGPT para ajudar a compreender o funcionamento de um determinado programa, mas também revela os perigos de uma utilização do ChatGPT sem espírito crítico e sem uma validação adicional dos resultados que devolve. Apesar de não ter sido capaz de explicar corretamente o funcionamento deste programa, a utilização do ChatGPT com este objetivo conjugada com a utilização do Matlab traz vantagens para a aprendizagem da linguagem MATLAB. Ao executar passo-a-passo as operações do programa no Matlab utilizando um processo de tentativa-erro para perceber a razão pela qual o resultado (incorreto) fornecido pelo ChatGPT é diferente do resultado (correto) obtido com o Matlab, somos inevitavelmente levados concluir que a operação aritmética (subtração) tem de ser executada antes da comparação lógica (&). Após confirmar, desta forma, a precedência de operações, é, no entanto, essencial consolidar este conhecimento com a consulta dos apontamentos de Computação ou a Ajuda do Matlab, de modo a compreender melhor todas as regras que definem as precedências para a execução de operações em MATLAB.



explain what the following Matlab code does step by step: `a = -2*ones(2,1); b = [-3:3:1]'; z = (a + ~b) & (a < b) - 1`



Certainly, I'd be happy to explain the provided MATLAB code step by step:



Step 1: `a = -2*ones(2,1);`

- This line initializes a variable `a` as a column vector of size 2×1.
- The function `ones(2,1)` creates a column vector of ones with a size of 2×1.
- The value of `-2` is then multiplied element-wise with this vector of ones, resulting in a column vector with all elements being -2.
- So, after this line, the variable `a` holds the value `[-2; -2]`.

Step 2: `b = [-3:3:1]';`

- This line initializes a variable `b` as a column vector.
- The syntax `[-3:3:1]` generates a sequence starting from -3, incrementing by 3, and stopping before or at 1.
- The result of this sequence generation is `[-3, 0]`.
- The `'` at the end transposes the generated row vector into a column vector.
- After this line, the variable `b` holds the value `[-3; 0]`.

Step 3: `z = (a + ~b) & (a < b) - 1;`

- This line initializes a variable `z` based on some logical operations involving the vectors `a` and `b`.
 - `~b` inverts the logical values of vector `b`, so `~b` becomes `[0; 1]`.
 - `a + ~b` performs element-wise addition between vectors `a` and `~b`, resulting in `[-2; -1]`.
 - `a < b` performs element-wise comparison between vectors `a` and `b`. In this case, it's comparing `[-2; -2]` with `[-3; 0]`.
 - The first element of `a` is not less than the first element of `b`, so it's `false`.
 - The second element of `a` is also not less than the second element of `b`, so it's `false`.
 - Therefore, the result is `[false; false]`.
 - `(a + ~b) & (a < b)` performs element-wise logical AND operation between the results of `a + ~b` and `a < b`. The results are `[false; false]`.
 - `... - 1` subtracts 1 from each element of the logical array `[false; false]`, resulting in `[false; false]`.
- So, after this line, the variable `z` holds the value `[false; false]`.

In summary, the provided MATLAB code initializes vectors `a` and `b`, and then performs various element-wise arithmetic and logical operations to compute the vector `z`, which consists of `false` values for each element.

Regenerate

Figura 4 – Mensagem e resposta do ChatGPT para o Exemplo #1.



Can you check Step 3? It seems there are errors



Apologies for the confusion. Let's go through Step 3 again:



Step 3: `z = (a + ~b) & (a < b) - 1;`

- `~b` inverts the logical values of vector `b`, so `~b` becomes `[0; 1]`.
- `a + ~b` performs element-wise addition between vectors `a` and `~b`, resulting in `[-2; -1]`.
- `a < b` performs element-wise comparison between vectors `a` and `b`. In this case, it's comparing `[-2; -2]` with `[-3; 0]`.
 - The first element of `a` is not less than the first element of `b`, so it's `false`.
 - The second element of `a` is less than the second element of `b`, so it's `true`.
 - Therefore, the result is `[false; true]`.
- `(a + ~b) & (a < b)` performs element-wise logical AND operation between the results of `a + ~b` and `a < b`.
 - The first element is `(-2) & false`, which is `false`.
 - The second element is `(-1) & true`, which is `true`.
 - Therefore, the result is `[false; true]`.
- Finally, the result `[false; true]` is subtracted by 1, element-wise.
 - The first element is `false - 1`, which is `-1`.
 - The second element is `true - 1`, which is `0`.
 - Therefore, the final result is `[-1; 0]`.

So, after Step 3, the variable `z` holds the value `[-1; 0]`.

Figura 5 – Segunda mensagem e resposta do ChatGPT para o Exemplo #1.

X Can you check if you are using the correct operator precedence for Matlab?

< 4/4 >  Certainly, let's revisit the operator precedence in MATLAB and go through Step 3 again:  

Step 3: `z = (a + ~b) & (a < b) - 1;`

- `~b` inverts the logical values of vector `b`, so `~b` becomes `[0; 1]`.
- `a + ~b` performs element-wise addition between vectors `a` and `~b`, resulting in `[-2; -1]`.
- `a < b` performs element-wise comparison between vectors `a` and `b`. In this case, it's comparing `[-2; -2]` with `[-3; 0]`.
 - The first element of `a` is not less than the first element of `b`, so it's `false`.
 - The second element of `a` is less than the second element of `b`, so it's `true`.
 - Therefore, the result is `[false; true]`.
- Now, the logical AND operator `&` has higher precedence than the subtraction operator `-`.
- `(a + ~b) & (a < b)` is evaluated first:
 - The first element is `-2 & false`, which is `false`.
 - The second element is `-1 & true`, which is `true`.
 - Therefore, the intermediate result is `[false; true]`.
- `(a + ~b) & (a < b) - 1` performs element-wise subtraction between the intermediate result `[false; true]` and `1`.
 - The first element is `false - 1`, which is `-1`.
 - The second element is `true - 1`, which is `0`.
 - Therefore, the final result is `[-1; 0]`.

So, after Step 3, the variable `z` holds the value `[-1; 0]`.

Apologies for the confusion in my previous responses. The original interpretation was indeed correct.

 Regenerate

Figura 6 – Terceira mensagem e resposta do ChatGPT para o Exemplo #1.

4.1.2 Segundo exemplo – mensagens em inglês e em português

O segundo exemplo discutido corresponde ao programa apresentado na Figura 7 (Exemplo #2) que não corresponde a um programa completo pois a variável A não está definida. Neste caso, a mensagem vai pedir ao ChatGPT a sequência de operações que são efetuadas durante a execução do programa, bem como uma explicação daquilo que o programa faz em termos globais, admitindo que a variável A é de um certo tipo. Num primeiro caso, utilizou-se a mensagem em inglês apresentada na Figura 8 onde se admite que a variável A é uma matriz. A explicação obtida é igualmente apresentada na mesma figura. Como se pode observar, neste caso, quer a sequência de operações, quer a explicação do que

o programa faz estão corretas. No entanto, é importante perceber que isto não quer dizer que o ChatGPT irá sempre devolver uma resposta correta para este caso. Por exemplo, repetindo a mensagem (usando a opção *Regenerate*), obtém-se a resposta apresentada na Figura 9 que contém erros. O resultado do Passo 3 está incorreto (o Passo 3 devolve um resultado por cada coluna da matriz A e não para a matriz toda) bem como a explicação do que programa faz (o programa verifica se existe alguma coluna da matriz A em que todos os números sejam múltiplos de 3 e não se existe alguma linha da matriz A em que todos os números sejam múltiplos de 3).

```
v1 = any(all(mod(A,3) == 0))
```

Figura 7 – Código MATLAB do Exemplo #2.

 Consider that A is a matrix. List the sequence of operations that the following Matlab code performs: `v1=any(all(mod(A,3)==0))` and then explain what the code does. 

 The MATLAB code ``v1 = any(all(mod(A, 3) == 0))`` performs the following sequence of operations:   

1. ``mod(A, 3)``: This computes the element-wise remainder of the matrix ``A`` when divided by 3. In other words, it calculates the matrix where each element is the remainder when the corresponding element of ``A`` is divided by 3.
2. ``all(mod(A, 3) == 0)``: This applies the element-wise comparison ``mod(A, 3) == 0``, resulting in a logical matrix of the same size as ``A``. Each element in this logical matrix is ``true`` if the corresponding element in the ``mod(A, 3)`` matrix is equal to 0, and ``false`` otherwise. The ``all`` function is then used to check if all elements in this logical matrix are ``true`` along every column. This step checks if all elements in each column of ``A`` are divisible by 3.
3. ``any(all(mod(A, 3) == 0))``: This applies the ``all(mod(A, 3) == 0)`` condition across all columns of the matrix ``A`` and uses the ``any`` function to check if there is at least one column in which all elements are divisible by 3 (i.e., all elements are ``true`` in that column). The result is a single logical value assigned to the variable ``v1``.

In summary, the code checks whether there exists at least one column in the matrix ``A`` where all elements are divisible by 3. If such a column exists, ``v1`` will be assigned the value ``true``; otherwise, it will be assigned the value ``false``.

In other words, the code is determining whether there is a vertical line (column) in the matrix ``A`` where all the elements are multiples of 3.

Figura 8 – Primeira mensagem e resposta do ChatGPT para o Exemplo #2.

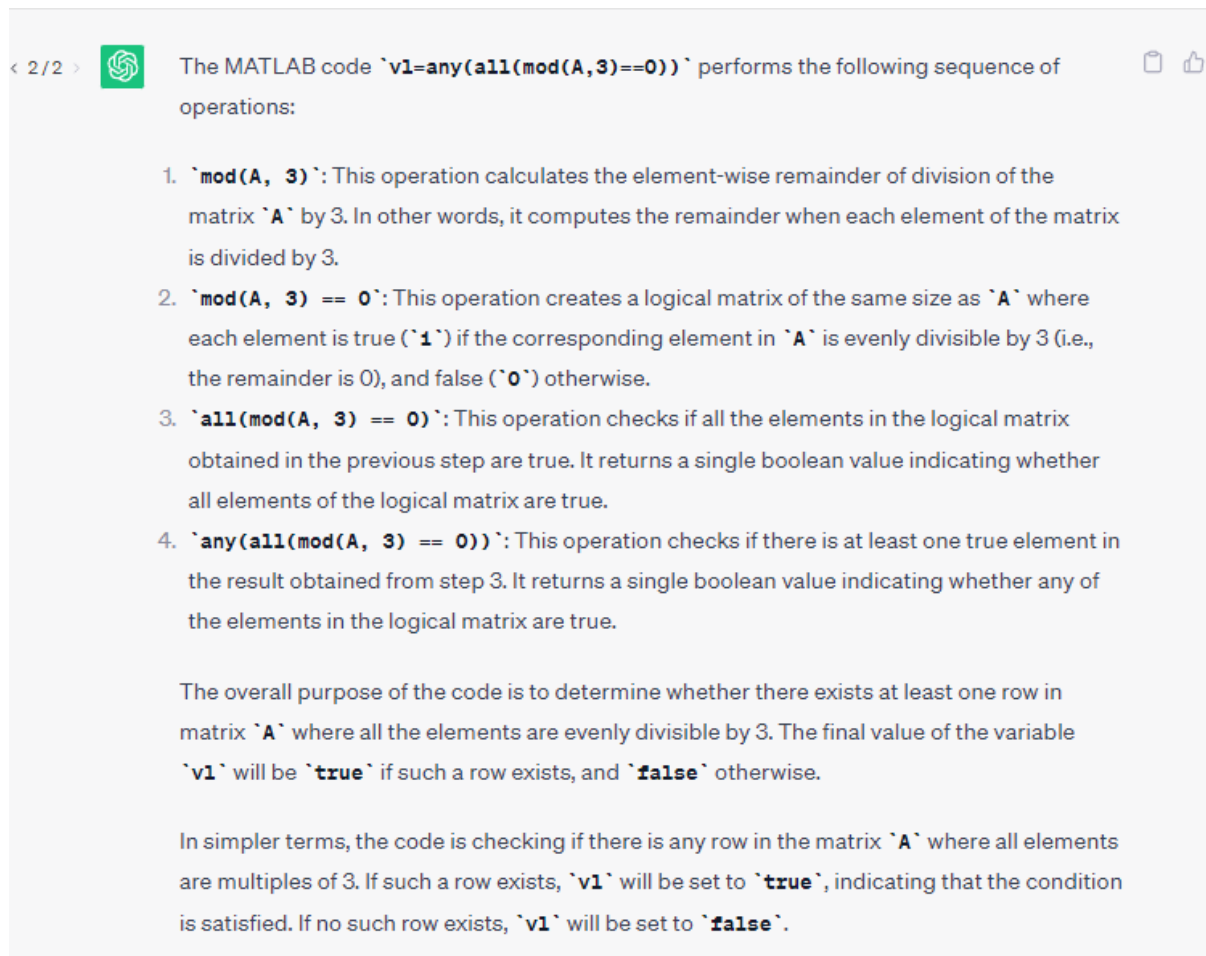


Figura 9 – Segunda resposta do ChatGPT para o Exemplo #2.

Relativamente às respostas obtidas do ChatGPT, observa-se ainda que as mesmas não referem nenhuma condição relacionada com o sinal dos elementos da matriz *A*. Como é sabido, os operadores *mod* e *rem* devolvem o mesmo resultado se o divisor e o dividendo tiverem o mesmo sinal (o que implica que, neste caso, os elementos da matriz *A* teriam de ser todos positivos). Dado que o código MATLAB do Exemplo #2 considera um divisor positivo, o operador *mod* devolve os valores do resto sempre com sinal positivo independentemente do sinal dos vários elementos da matriz *A*.

De modo a testar a qualidade das respostas geradas pelo ChatGPT em português, repetiu-se o mesmo exemplo usando a mesma mensagem traduzida. A mensagem em português e a respetiva resposta são apresentadas na Figura 10. Como se pode observar, a resposta obtida apresenta os mesmos erros da segunda resposta obtida em inglês (Figura 9). Repetindo a mensagem algumas vezes (usando a opção *Regenerate*), obteve-se a resposta apresentada na Figura 11 que não contém erros. Verifica-se, no entanto, que esta resposta é menos detalhada que a correspondente versão em inglês. Como já referido, é esperado que o ChatGPT forneça melhores resultados (mais corretos e mais detalhados) nesse idioma, dado que a informação utilizada para treinar o modelo de inteligência artificial é maioritariamente em inglês.

É importante salientar que esta mensagem em português foi repetida 20 vezes, das quais apenas uma resultou numa resposta totalmente correta (Figura 11). Das restantes respostas, salienta-se que algumas delas estavam parcialmente corretas como, por exemplo, os casos apresentados nas Figuras 12 e 13. A resposta da Figura 12 fornece uma explicação correta para o que o programa faz, mas a explicação do Passo 2 está incorreta (o Passo 2 devolve um resultado por cada coluna da matriz *A* e

não para a matriz toda). No caso da resposta da Figura 13, observa-se que a explicação quer dos passos, quer do que o programa faz é vaga, dado que não identifica se os resultados das operações são obtidos por colunas ou por linhas.

Tal como no exemplo anterior, os vários resultados obtidos neste caso ilustram os perigos de uma utilização do ChatGPT sem espírito crítico e sem uma validação adicional dos resultados que devolve feita com o Matlab.



Considere que A é uma matriz. Identifique a sequência de operações que o programa seguinte em Matlab vai executar: `v1=any(all(mod(A,3)==0))` e explique o que o programa faz.



O programa em MATLAB que você forneceu executa as seguintes operações:



1. `mod(A, 3)`: Calcula o módulo de cada elemento da matriz `A` em relação a 3. Isso significa que o programa verifica o restante da divisão de cada elemento de `A` por 3.
2. `all(...)`: O `all` é usado para verificar se uma condição é verdadeira para todos os elementos ao longo de uma dimensão específica. Neste caso, a condição é a comparação dos restos da divisão por 3 para cada elemento. Portanto, essa etapa verifica se todos os elementos de `A` têm um restante de divisão igual a 0 quando divididos por 3.
3. `any(...)`: O `any` é usado para verificar se pelo menos um elemento é verdadeiro ao longo de uma dimensão específica. Neste caso, está sendo usado para verificar se pelo menos um dos elementos da matriz resultante da etapa anterior (ou seja, a matriz de booleanos indicando se os restos da divisão eram todos 0) é verdadeiro.
4. `v1=...`: O resultado do `any(all(...))` é armazenado na variável `v1`. Portanto, `v1` será um valor booleano que indica se há pelo menos uma linha da matriz `A` em que todos os elementos são divisíveis por 3.

Em resumo, o programa verifica se há pelo menos uma linha (ou seja, um conjunto de elementos ao longo da mesma dimensão) na matriz `A` onde todos os elementos são divisíveis por 3. Se essa condição for verdadeira, a variável `v1` será atribuída o valor `true`; caso contrário, será atribuído o valor `false`.

No contexto de matrizes, isso pode ser útil para verificar se existe uma linha onde todos os elementos atendem a um determinado critério, nesse caso, sendo divisíveis por 3.

Figura 10 – Mensagem e resposta em português do ChatGPT para o Exemplo #2.

< 4 / 4 >  O programa em MATLAB executa as seguintes operações:  

1. `mod(A, 3)`: Calcula o elemento a elemento o resto da divisão de cada elemento da matriz `A` por 3.
2. `all(mod(A, 3) == 0)`: Para cada coluna da matriz resultante do passo anterior, verifica se todos os elementos são iguais a 0. Isso cria um vetor de booleanos indicando se todos os elementos em cada coluna são divisíveis por 3.
3. `any(all(mod(A, 3) == 0))`: Verifica se existe alguma coluna em que todos os seus elementos são divisíveis por 3 (ou seja, o resultado do passo 2 contém pelo menos um `true`).

A explicação do que o programa faz é a seguinte:

O programa verifica se existe pelo menos uma coluna na matriz `A` em que todos os elementos são divisíveis por 3. Isso é feito através da utilização das funções `mod` (que calcula o resto da divisão) e das funções `all` e `any` (que verificam se todos os elementos são verdadeiros ou se pelo menos um elemento é verdadeiro, respectivamente).

O valor final da variável `v1` será `true` se pelo menos uma coluna da matriz `A` possuir todos os seus elementos divisíveis por 3; caso contrário, o valor de `v1` será `false`. Em outras palavras, o programa está verificando se há alguma coluna onde todos os elementos são múltiplos de 3.

Figura 11 – Quarta resposta em português do ChatGPT para o Exemplo #2.

< 12 / 12 >  O programa em MATLAB realiza uma série de operações em uma matriz A e, em última instância, produz um resultado booleano (verdadeiro ou falso). Vamos analisar as operações uma a uma e entender o que o programa faz:  

1. `mod(A, 3)`: Esta operação calcula o resto da divisão de cada elemento da matriz A por 3. Isso resulta em uma matriz do mesmo tamanho de A, onde cada elemento é o resto da divisão correspondente.
2. `all(...)`: Essa função verifica se uma determinada condição é verdadeira para todos os elementos ao longo de uma dimensão específica. No caso do programa, está sendo aplicada ao resultado da operação `mod(A, 3)`. Isso verifica se todos os elementos da matriz resultante da operação anterior são iguais a zero, ou seja, se todos os elementos da matriz A são divisíveis por 3.
3. `any(...)`: Essa função verifica se pelo menos um elemento ao longo de uma dimensão específica satisfaz uma determinada condição. Neste caso, a função está sendo aplicada ao resultado da operação `all(mod(A, 3)==0)`. Isso verifica se pelo menos uma coluna da matriz resultante da operação anterior contém apenas elementos divisíveis por 3.

Portanto, o programa em MATLAB está verificando se existe pelo menos uma coluna na matriz A onde todos os elementos são divisíveis por 3. O resultado dessa operação é um valor booleano (verdadeiro ou falso), que será armazenado na variável `v1`.

Em resumo, o programa determina se existe pelo menos uma coluna na matriz A cujos elementos sejam todos divisíveis por 3 e armazena esse resultado na variável `v1`.

Figura 12 – Resposta parcialmente correta em português do ChatGPT para o Exemplo #2.

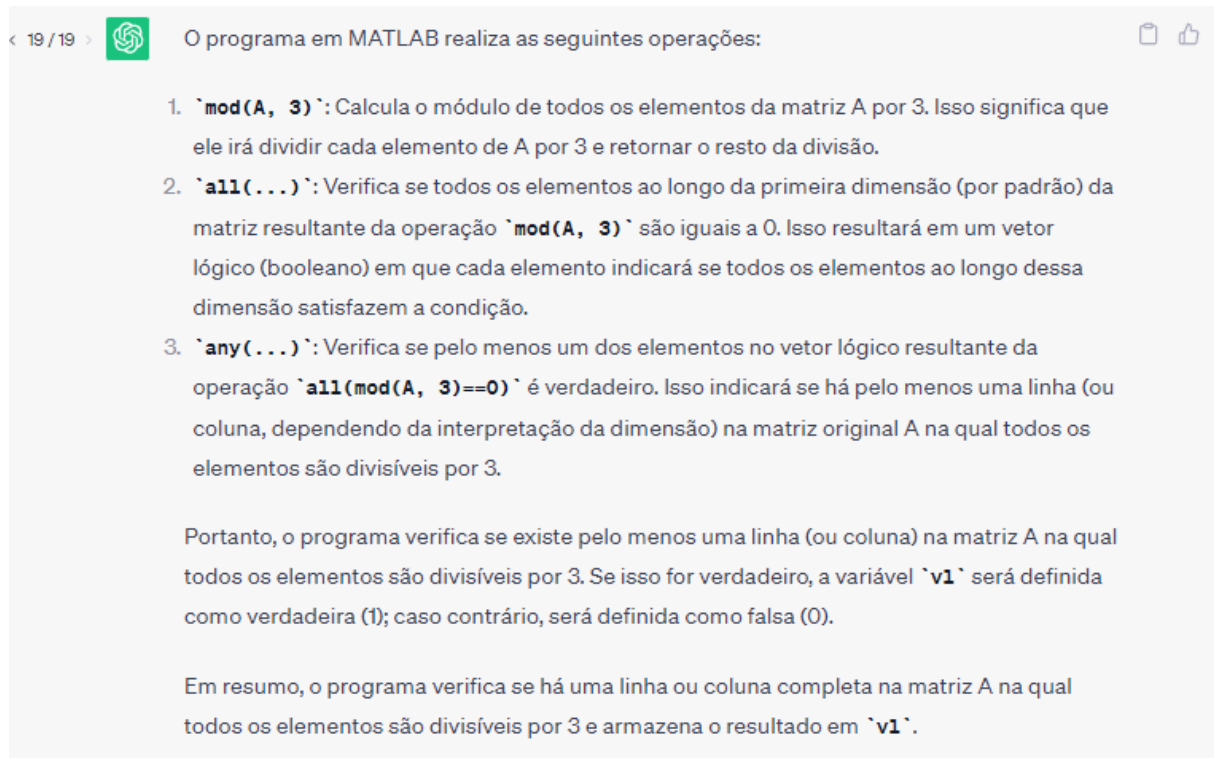


Figura 13 – Resposta parcialmente correta em português do ChatGPT para o Exemplo #2.

4.1.3 Terceiro exemplo – mensagens em inglês e português

O terceiro exemplo discutido corresponde ao programa apresentado na Figura 14 (Exemplo #3) que envolve um maior número de operações. Neste caso, a mensagem vai simplesmente pedir ao ChatGPT para explicar o que o programa faz. Num primeiro caso, utilizou-se a mensagem em inglês apresentada na Figura 15 que também apresenta a explicação obtida.

```
A = input(' matriz = ');
[nl,nc] = size(A);
if any(A(:) < 0)
    A(:) = A(:)-min(A(:));
end
k = 3-(A(1,:) <= 20) - (A(1,:) <= 10);
v = k.*A(1,:)
```

Figura 14 – Código MATLAB do Exemplo #3.

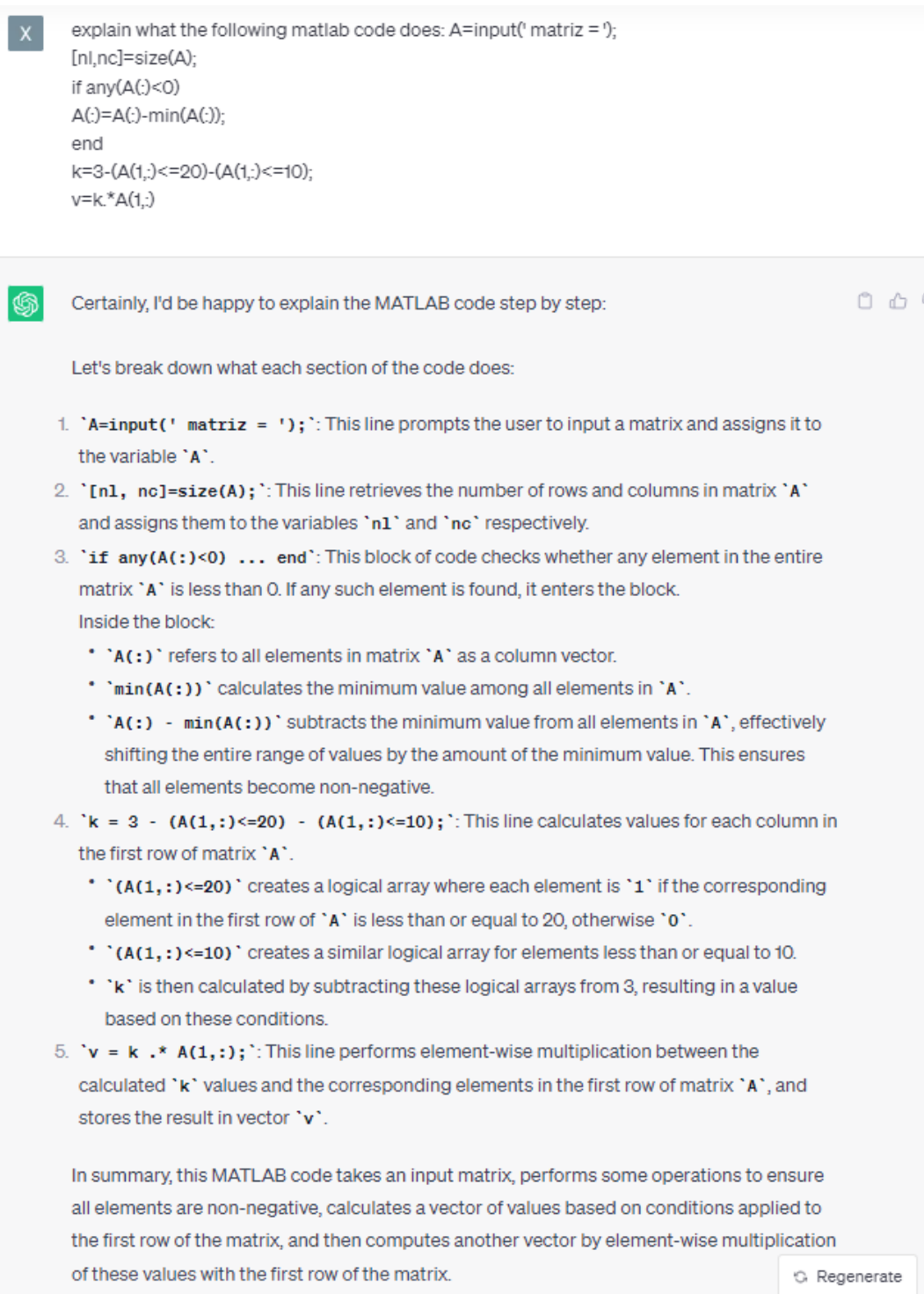


Figura 15 – Primeira mensagem e resposta do ChatGPT para o Exemplo #3.

Como se observa, o ChatGPT explica corretamente a sequência de operações do programa e interpreta de forma igualmente correta o objetivo da estrutura de decisão *if...end* (garantir que a matriz A final não tem elementos negativos). No entanto, isso não significa que ao repetir a mensagem (usando a opção *Regenerate*) se obtenha sempre uma resposta correta, como já referido. De modo a analisar o caso em que a mensagem é escrita em português, repetiu-se o mesmo exemplo em português. A mensagem usada e a respetiva resposta são apresentadas na Figura 16. Como se observa, a resposta obtida neste caso está também correta, mas repetir a mensagem poderá não gerar o mesmo resultado.



```
Explicue o que faz o programa seguinte em Matlab: A=input(' matriz = ');
[nl,nc]=size(A);
if any(A(:)<0)
A(:)=A(:)-min(A(:));
end
k=3-(A(1,:)<=20)-(A(1,:)<=10);
v=k.*A(1,:)
```



Claro, vou explicar o que cada parte do programa faz em MATLAB:



1. `A=input(' matriz = ');`: Este trecho de código solicita ao usuário que insira uma matriz A. O usuário pode inserir os valores da matriz separados por espaços ou vírgulas, e pressionar Enter quando terminar. A matriz é armazenada na variável `A`.
2. `[nl, nc] = size(A);`: Aqui, o programa calcula o número de linhas (`nl`) e o número de colunas (`nc`) da matriz `A` usando a função `size()`. Esses valores serão usados posteriormente.
3. `if any(A(:)<0)`: Este bloco condicional verifica se existe algum valor negativo na matriz `A` utilizando a função `any()`. Se houver algum valor negativo, o bloco de código dentro do `if` será executado.
4. `A(:) = A(:) - min(A(:));`: Dentro do bloco condicional, este código subtrai o valor mínimo de toda a matriz `A` de todos os elementos da matriz. Isso efetivamente desloca todos os valores da matriz para garantir que o valor mínimo seja zero.
5. `k = 3 - (A(1,:)<=20) - (A(1,:)<=10);`: Nesta linha, o programa cria um vetor de valores booleanos com base nas comparações dos elementos da primeira linha da matriz `A`. `A(1,:)<=20` cria um vetor de booleanos onde cada elemento será `true` se o valor correspondente na primeira linha da matriz `A` for menor ou igual a 20, e `false` caso contrário. O mesmo ocorre para `A(1,:)<=10`.
A expressão `k = 3 - (A(1,:)<=20) - (A(1,:)<=10);` subtrai esses vetores booleanos de 3, convertendo os `true` em 1 e os `false` em 0. Isso resulta em um vetor `k` que indica certas condições para cada elemento da primeira linha da matriz `A`.
6. `v = k .* A(1,:);`: Nesta linha, o programa multiplica o vetor `k` (calculado anteriormente) pelo vetor correspondente da primeira linha da matriz `A`. Isso resulta em um novo vetor `v`, onde cada elemento é o produto do valor correspondente de `k` pelo valor correspondente da primeira linha de `A`.

Resumindo, o programa realiza as seguintes ações:

1. Pede ao usuário para inserir uma matriz `A`.
2. Verifica se há valores negativos em `A` e, se houver, desloca todos os valores para garantir que o mínimo seja zero.
3. Calcula um vetor `k` com base nas comparações dos elementos da primeira linha de `A` em relação a 20 e 10.
4. Multiplica os valores da primeira linha de `A` pelos valores correspondentes de `k`, armazenando o resultado no vetor `v`.

Regenerate

Figura 16 – Primeira mensagem e resposta em português do ChatGPT para o Exemplo #3.

4.2 Utilizar o ChatGPT para corrigir erros de um programa em MATLAB

4.2.1 Quarto exemplo – mensagens em português

Para usar o ChatGPT na identificação e correção de erros de um programa existente, é importante que a mensagem faça um enquadramento detalhado do que o programa é suposto fazer. Caso contrário, haverá erros ou instruções que, não estando erradas, podem não gerar os resultados pretendidos. Sem o referido enquadramento, o ChatGPT terá dificuldade em fornecer uma versão corrigida do programa que vá de encontro aos objetivos do utilizador. De forma a ilustrar esta questão, vai-se utilizar o programa apresentado na Figura 17 (Exemplo #4) que contém vários erros. Em termos de objetivos, este programa deve:

- Pedir ao utilizador uma matriz quadrada M;
- Identificar o maior valor absoluto da diagonal principal da matriz;
- Trocar os valores da correspondente linha e coluna pelos valores que estão, respetivamente, na primeira linha e na primeira coluna da matriz. Caso exista mais do que um elemento na diagonal principal cujo valor absoluto corresponda ao maior valor absoluto da diagonal principal, o programa deve considerar o elemento que está na linha de maior índice;
- Manter a matriz M sem alterações caso o maior valor absoluto da diagonal principal esteja na primeira posição da matriz.

```
M = ('Matriz quadrada: ');  
valor = max(abs(diag(M(:))))  
kk = find abs(diag(M))==valor);  
k = kk[1];  
while k == 1  
    t = M(1,:);  
    M(k,:) = t;  
    M(1,:) = M(k,:);  
    M(:,1) = M(:,k);  
    M(:,k) = M(:,1);  
end
```

Figura 17 – Código MATLAB do Exemplo #4.

Na primeira tentativa de corrigir o programa, e de modo a ilustrar a importância de incluir na mensagem um enquadramento que explique os objetivos do programa, a mensagem vai ser escrita sem esse enquadramento. A mensagem usada em português e a respetiva resposta são apresentadas na Figura 18. Como se observa, o ChatGPT consegue corrigir erros de sintaxe. No entanto, não consegue deduzir que deve ser pedida uma matriz ao utilizador e não consegue perceber o objetivo do ciclo de repetição condicional *while...end*, introduzindo alterações que não vão de encontro aos objetivos. Repetir a mensagem para obter uma nova resposta também não traz benefícios significativos (Figura 19).

X

Considere o programa seguinte em Matlab que tem erros. Identifique os erros no programa seguintes e proponha as correções necessárias:

```
M = ('Matriz quadrada: ');
valor = max(abs(diag(M(:))))
kk = find abs(diag(M))==valor;
k = kk[1];
while k == 1
    t = M(1,:);
    M(k,:) = t;
    M(1,:) = M(k,:);
    M(:,1) = M(:,k);
    M(:,k) = M(:,1);
end
disp ("M")
```



Claro, vou identificar os erros no programa e fornecer as correções necessárias:

```
matlab Copy code

% Corrigido:
M = magic(4); % Exemplo de matriz quadrada
disp('Matriz quadrada: '); % Corrigido: Removido parênteses desnecessários
valor = max(abs(diag(M))); % Corrigido: Adicionado ponto e vírgula no final
kk = find(abs(diag(M)) == valor); % Corrigido: Adicionado ponto e vírgula no final
k = kk(1); % Corrigido: Substituído colchetes por parênteses
while k == 1
    t = M(1,:);
    M(1,:) = M(k,:);
    M(k,:) = t; % Corrigido: Trocadas as linhas k e 1
    M(:,1) = M(:,k);
    M(:,k) = t; % Corrigido: Trocadas as colunas k e 1
    k = find(abs(diag(M)) == valor, 1); % Corrigido: Atualizado o valor de k dentro do loop
end
disp(M); % Corrigido: Removido aspas duplas em torno de "M"
```

Principais correções realizadas:

1. Corrigido a inicialização da matriz `M` (coloquei um exemplo de matriz quadrada - matriz mágica de ordem 4 - como exemplo).
2. Removido parênteses desnecessários na linha de exibição do texto.
3. Adicionado ponto e vírgula no final das linhas de código.
4. Corrigida a sintaxe do comando `find`.
5. Corrigido o índice de acesso ao vetor `kk` (substituído colchetes por parênteses).
6. Corrigidas as trocas de linhas e colunas dentro do loop para corrigir a ordem das operações.
7. Atualizado o valor de `k` dentro do loop para que o loop possa avançar ou terminar corretamente.
8. Removidas as aspas duplas em torno do nome da variável `M` na linha de exibição.

Lembre-se de que o exemplo utiliza uma matriz mágica de ordem 4 apenas como um exemplo de matriz quadrada. Certifique-se de substituir isso pela matriz real com a qual você está trabalhando.

Figura 18 – Primeira mensagem e resposta do ChatGPT para o Exemplo #4.

Claro, vou ajudar a identificar e corrigir os erros no programa MATLAB:

```
matlab Copy code

% Corrigido:
M = magic(4); % Exemplo de uma matriz quadrada
valor = max(abs(diag(M)));
kk = find(abs(diag(M)) == valor);
k = kk(1);
while k == 1
    t = M(1,:);
    M(1,:) = M(k,:);
    M(k,:) = t;
    M(:,1) = M(:,k);
    M(:,k) = t; % Correção: deve ser "M(:,k) = M(:,1);"
end
disp('M'); % Correção: deve ser "disp(M);"
```

Principais erros e correções realizadas:

1. `M = ('Matriz quadrada: ');`: Aqui você está tentando atribuir uma string a uma variável chamada `M`, o que não faz sentido. Se você deseja criar uma matriz, use a função `magic` (ou qualquer outra função apropriada para criar uma matriz quadrada) para criar uma matriz de exemplo.
2. `find abs(diag(M))==valor;`: Aqui, você está tentando usar a função `find` sem parênteses em torno do argumento completo. Além disso, a função `find` precisa ser aplicada a um resultado lógico (booleano) para encontrar os índices correspondentes.
3. `disp ("M")`: O uso de aspas curvas ("smart quotes") pode causar problemas na linguagem MATLAB. Use aspas simples (") para delimitar a string que você deseja exibir. Além disso, para exibir a matriz `M`, você deve usar `disp(M)` em vez de `disp("M")`.
4. O bloco `while` parece tentar realizar alguma manipulação de permutação de linhas e colunas da matriz `M`, mas há erros na lógica e na atribuição das variáveis. A lógica correta para permutações não foi implementada. A correção acima faz uma tentativa de corrigir isso, mas a lógica completa depende do que você está tentando alcançar com esse loop.

Lembre-se de que a correção acima é baseada na suposição de que você deseja realizar permutações nas linhas e colunas da matriz `M`. Se o objetivo for diferente, os ajustes podem ser necessários. Certifique-se de entender o que o programa deve fazer para ajustar as correções adequadamente.

Figura 19 – Segunda resposta do ChatGPT para o Exemplo #4.

Com base nas respostas obtidas, verifica-se que, sem um enquadramento que estabeleça os objetivos do programa, o ChatGPT apenas conseguirá corrigir corretamente questões relacionadas com a sintaxe dos comandos envolvidos. Salienta-se, no entanto, que ambas as respostas incluem uma advertência sobre a forma como a matriz `M` está a ser definida na versão corrigida do programa. As respostas do ChatGPT poderão não incluir avisos semelhantes relativamente ao facto das restantes modificações introduzidas, nomeadamente no ciclo de repetição condicional `while...end`, não irem ao encontro dos objetivos pretendidos. No entanto, no caso da resposta da Figura 19, o ChatGPT inclui um aviso final que indica que o programa corrigido pode não ir de encontro aos objetivos.

De modo a obter uma resposta consistente com os objetivos do programa, escreveu-se uma nova mensagem que explica esses objetivos. A mensagem usada é apresentada na Figura 20 e a resposta obtida é apresentada na Figura 21. Analisando a resposta, observa-se que as correções introduzidas pelo ChatGPT vão totalmente ao encontro dos objetivos pretendidos e resolvem os vários erros existentes. Apesar da versão programa gerada pelo ChatGPT estar correta, salienta-se que a explicação passo-a-passo que fornece não descreve todas as alterações que foram feitas. Por exemplo, o texto não indica a alteração de *diag(M(:))* para *diag(M)* de modo a que a diagonal principal da matriz M seja corretamente obtida e a introdução do parêntese entre os comandos *find* e *abs*.

Salienta-se novamente que repetir a mensagem poderá não conduzir à mesma solução e poderá inclusivamente gerar novas versões do programa que continuam a ter erros. A Figura 22 apresenta um exemplo de uma outra versão correta obtida com o ChatGPT, de modo a ilustrar esta questão. Neste caso, verifica-se que o ChatGPT adicionou uma outra condição ao ciclo de repetição condicional *while...end* de modo a mandar uma mensagem ao utilizador caso o maior valor absoluto da diagonal esteja na primeira posição. Esta questão não estava formalmente pedida na mensagem de enquadramento do programa, mas não altera o seu funcionamento global. Além disso, observa-se que a explicação passo-a-passo também não descreve todas as alterações que foram feitas neste caso.

Não obstante o facto de, neste exemplo, o ChatGPT ter gerado duas versões de um mesmo programa que estão corretas, é fundamental, validar os vários passos intermédios dos programas gerados com a sua execução em Matlab de modo a verificar que todas as suas partes são necessárias e estão a gerar resultados consistentes.

X

Considere o programa seguinte em Matlab que tem erros. O programa deve pedir ao utilizador uma matriz quadrada M, identificar o maior valor absoluto da diagonal principal e trocar os valores da correspondente linha e coluna pelos valores que estão, respetivamente, na primeira linha e na primeira coluna. Considere ainda que caso exista mais do que um elemento na diagonal principal cujo valor absoluto corresponda ao maior valor absoluto da diagonal principal, o programa deve considerar o elemento que está na linha de maior índice. Caso o maior valor absoluto da diagonal principal estiver na primeira posição, o programa não deve alterar a matriz M. Identifique os erros no programa seguintes e proponha as correções necessárias:

```

M = ('Matriz quadrada: ');
valor = max(abs(diag(M(:))))
kk = find abs(diag(M))==valor;
k = kk[1];
while k == 1
    t = M(1,:);
    M(k,:) = t;
    M(1,:) = M(k,:);
    M(:,1) = M(:,k);
    M(:,k) = M(:,1);
end
disp ("M")

```

✍

Figura 20 – Primeira mensagem com enquadramento para o Exemplo #4.



Parece que o código contém vários erros. Vou ajudá-lo a corrigi-los passo a passo:



matlab

Copy code

```
% Pedir ao utilizador uma matriz quadrada M
M = input('Matriz quadrada: ');

% Identificar o maior valor absoluto da diagonal principal
valor = max(abs(diag(M)));

% Encontrar índices dos elementos da diagonal que têm o maior valor absoluto
kk = find(abs(diag(M)) == valor);
k = kk(end); % Use o último índice caso haja mais de um elemento com o maior valor

% Verificar se k é igual a 1, e se não for, efetuar as trocas
if k ~= 1
    t = M(1, :);
    M(1, :) = M(k, :);
    M(k, :) = t;

    t = M(:, 1);
    M(:, 1) = M(:, k);
    M(:, k) = t;
end

% Mostrar a matriz M
disp('M');
disp(M);
```

Principais correções realizadas:

1. Substituí o uso de aspas simples (``) por `input` para solicitar a entrada do usuário para a matriz.
2. Usei a função `abs` corretamente para calcular o valor absoluto.
3. Corrigi a linha que encontra o índice `k` dos elementos da diagonal com o maior valor absoluto, escolhendo o último índice em caso de empate.
4. Usei a estrutura `if` para verificar se `k` é diferente de 1, em vez de um loop `while`.
5. Corrigi a troca de linhas e colunas para refletir as trocas descritas no problema.
6. Substituí as aspas duplas por aspas simples na função `disp`.

Lembre-se de que o código acima ainda pode ser otimizado e melhorado, mas as correções fornecidas devem resolver os principais problemas do código original.

Figura 21 – Resposta do ChatGPT à mensagem com enquadramento para o Exemplo #4.



Parece que há vários erros no código que você forneceu. Vou identificar esses erros e propor as correções necessárias:



matlab

Copy code

```
% Solicitar a matriz ao usuário
M = input('Matriz quadrada: ');

% Encontrar o maior valor absoluto da diagonal principal
valor = max(abs(diag(M)));

% Encontrar os índices dos elementos da diagonal que têm o maior valor absoluto
kk = find(abs(diag(M)) == valor);
k = kk(end); % Selecionar o índice de maior valor se houver mais de um

% Verificar se o maior valor absoluto está na primeira posição
if k == 1
    disp('O maior valor absoluto da diagonal está na primeira posição. A matriz não será alterada.');
else
    % Trocar valores da linha e coluna correspondentes
    t = M(1,:);
    M(1,:) = M(k,:);
    M(k,:) = t;
    t = M(:,1);
    M(:,1) = M(:,k);
    M(:,k) = t;

    % Exibir a matriz resultante
    disp('Matriz resultante:');
    disp(M);
end
```

Correções realizadas:

1. Usamos `input` para solicitar a entrada da matriz ao usuário.
2. Corrigimos a sintaxe para encontrar os índices dos elementos da diagonal com maior valor absoluto.
3. Utilizamos `kk(end)` para selecionar o índice de maior valor se houver mais de um.
4. Substituímos o `while` por um `if` para verificar se o maior valor absoluto está na primeira posição.
5. Corrigimos as trocas de valores entre linhas e colunas para que seja feita a troca correta.
6. Adicionamos uma mensagem de exibição da matriz resultante.

Lembre-se de que o programa agora trata o caso em que o maior valor absoluto já está na primeira posição e não faz alterações nesse caso.

Figura 22 – Resposta alternativa do ChatGPT à mensagem com enquadramento para o Exemplo #4.

4.2.2 Quinto exemplo – mensagens em português

Neste novo exemplo de utilização do ChatGPT para identificar e corrigir erros de um programa existente, considerou-se um problema um pouco mais complexo que envolve a verificação de mais condições. O programa considerado é apresentado na Figura 23 (Exemplo #5) e contém vários erros de sintaxe e de estrutura. Em termos de objetivos, este programa deve:

- Pedir ao utilizador uma matriz A;
- Analisar se todos os elementos da primeira linha são pares;
- No caso de todos os elementos da primeira linha serem pares, deve substituir a última linha pela metade dos valores da primeira linha, e substituir os valores da primeira linha pelos da última linha;
- Caso contrário, e se existir pelo menos um elemento par na primeira linha, o programa deve substituir a última coluna da matriz pela soma dos valores da primeira coluna com os valores das colunas cujo elemento da primeira linha é par;
- No caso de não haver nenhum elemento par na primeira linha, o programa deve substituir o elemento da primeira coluna e da segunda linha da matriz pelo máximo dos valores absolutos dos elementos da matriz A.

```
A = input('Matriz A (nxm)=');  
if all(rem(A(1,:),2) == 0)  
    A(1,:) == A(1,:)/2;  
    A = A([end 2:(end-1)]);  
else  
    posi=find(rem(A(1,:),2)~=0);  
    if isempty(posi) == 0  
        A(2,1) = max(abs(A));  
    else  
        A(:,end) = sum(A(:,[1 posi]))';  
    end  
disp'A nova matriz é igual a:'  
disp('A')
```

Figura 23 – Código MATLAB do Exemplo #5.

Na primeira tentativa de corrigir o programa, utilizou-se a mensagem em português apresentada na Figura 24 que contém o enquadramento que estabelece os objetivos do programa. A resposta obtida é apresentada na Figura 25. Analisando a resposta, observa-se que as correções introduzidas pelo ChatGPT tornam o programa funcional (o programa consegue ser executado no Matlab sem dar erro). Verifica-se, assim, que o ChatGPT conseguiu corrigir os erros de sintaxe, mas não corrigiu os erros da

estrutura do programa. Repetindo a mensagem várias vezes (usando a opção *Regenerate*), observou-se que os erros da estrutura não foram alterados, obtendo-se várias soluções que são praticamente iguais à solução apresentada na Figura 25. No entanto, numa das repetições, o ChatGPT fez uma alteração na estrutura (Figura 26). Apesar disso, o programa continua a não resolver o problema pretendido. Uma análise destas duas soluções executando vários exemplos em Matlab mostra, por exemplo, que:

- Na primeira versão (Figura 25), se a matriz A apenas tiver elementos pares e ímpares na primeira linha, a condição `~isempty(posi)` será verdadeira, mas o que será executado não é a substituição da última coluna da matriz pela soma dos valores da primeira coluna com os valores das colunas cujo elemento da primeira linha é par;
- Na segunda versão (Figura 26), se a matriz A apenas tiver elementos ímpares na primeira linha, a condição `~isempty(posi)` será verdadeira, mas o que será executado não é a substituição do elemento da primeira coluna e da segunda linha da matriz pelo máximo dos valores absolutos dos elementos da matriz A.

Estes erros de funcionamento do programa apontam para a possibilidade de ser necessário modificar também a forma como a variável *posi* é definida. Fazendo essa sugestão ao ChatGPT nas duas versões obtiveram-se as novas respostas apresentadas nas Figuras 27 e 28. Nestas novas soluções, o ChatGPT apenas modificou a forma como a variável *posi* é definida (tal como sugerido na mensagem). No entanto, a modificação efetuada não tem o mesmo efeito nas duas versões. Na primeira versão, o programa continua a não executar o que é pretendido quando a matriz A apenas tiver elementos pares e ímpares na primeira linha. Na segunda versão, a alteração da variável *posi* faz com que o programa já funcione corretamente.

X User

Considere o programa seguinte em Matlab que tem erros. O programa deve pedir ao utilizador uma matriz A e analisar se todos os elementos da primeira linha são pares. No caso de todos os elementos da primeira linha serem pares, deve substituir a última linha pela metade dos valores da primeira linha, e substituir os valores da primeira linha pelos da última linha. Caso contrário, e se existir pelo menos um elemento par na primeira linha, o programa deve substituir a última coluna da matriz pela soma dos valores da primeira coluna com os valores das colunas cujo elemento da primeira linha é par. No caso de não haver nenhum elemento par na primeira linha, o programa deve substituir o elemento da primeira coluna da segunda linha pelo máximo dos valores absolutos dos elementos da matriz A. Identifique os erros no programa seguinte e proponha as correções necessárias:

```
A=input('Matriz A (nxm)=');
if all(rem(A(1,:),2)=0)
    A(1,:)=A(1,:)/2;
    A=A([end 2:(end-1)]);
else
    posi=find(rem(A(1,:),2)~=0);
    if isempty(posi)==0
        A(2,1)=max(abs(A));
    else
        A(:,end)=sum(A(:,[1 posi]))';
    end
disp('A nova matriz é igual a:');
disp('A')
```

Figura 24 – Primeira mensagem para o Exemplo #5.



Há vários erros no programa fornecido. Aqui estão as correções necessárias:



matlab

Copy code

```
A = input('Matriz A (nxm)=');

if all(rem(A(1,:), 2) == 0)
    A(1, :) = A(1, :) / 2;
    A = [A(end, :); A(2:end-1, :); A(1, :)];
else
    posi = find(rem(A(1,:), 2) ~= 0);
    if ~isempty(posi)
        A(2, 1) = max(abs(A(:)));
    else
        A(:, end) = sum(A(:, [1, posi]), 2);
    end
end

disp('A nova matriz é igual a:');
disp(A);
```

Correções realizadas:

1. Linha 1: Não é necessário o ponto e vírgula após a matriz `A`.
2. Linha 3: Substituído `==` por `=` para realizar a atribuição correta dos valores da primeira linha.
3. Linha 4: Corrigido o rearranjo das linhas da matriz `A` para trocar a primeira linha com a última linha.
4. Linha 6: Substituído `isempty(posi) == 0` por `~isempty(posi)` para verificar se `posi` não está vazio.
5. Linha 8: Corrigida a atualização dos valores na última coluna da matriz.
6. Linha 10: Corrigido o uso de `disp` para mostrar o valor da matriz `A`.

Lembre-se de que, ao executar o programa, você precisa inserir a matriz desejada no formato `[row1; row2; ...]`, onde cada linha é representada por um vetor separado por espaços. Certifique-se também de que a lógica do programa atenda aos requisitos que você mencionou.

Figura 25 – Resposta do ChatGPT para o Exemplo #5.



Parece que há vários erros no programa apresentado. Vou destacar os problemas e fornecer as correções necessárias:

matlab

Copy code

```
A = input('Matriz A (nxm)=');

% Verifica se todos os elementos da primeira linha são pares
if all(rem(A(1,:), 2) == 0)
    % Substitui a última linha pelos valores da metade da primeira linha
    A(end, :) = A(1, :) / 2;

    % Substitui a primeira linha pelos valores da última linha
    A(1, :) = A(end, :);
else
    % Encontra as posições dos elementos ímpares na primeira linha
    posi = find(rem(A(1,:), 2) ~= 0);

    if ~isempty(posi) % Verifica se há elementos ímpares na primeira linha
        % Substitui a última coluna pelos valores da soma da primeira coluna
        A(:, end) = A(:, end) + sum(A(:, posi), 2);
    else
        % Substitui o elemento da primeira coluna da segunda linha pelo máximo
        A(2, 1) = max(max(abs(A)));
    end
end

disp('A nova matriz é igual a:');
disp(A);
```

Correções realizadas:

1. Troquei `==` por `~= ` para realizar a substituição de linhas na primeira condição.
2. Corrigi a maneira como se selecionam as colunas na segunda substituição.
3. Adicionei a verificação de elementos ímpares na primeira linha corretamente.
4. Corrigi a substituição do elemento da primeira coluna da segunda linha pelo máximo dos valores absolutos da matriz.

Agora o programa deve funcionar corretamente de acordo com a lógica que você descreveu.

Figura 26 – Resposta alternativa do ChatGPT para o Exemplo #5.



Peço desculpas pela confusão. Você está correto, a definição da variável `posi` precisa ser ajustada. Aqui está o programa corrigido:

matlab

Copy code

```
A = input('Matriz A (nxm)=');

if all(rem(A(1,:), 2) == 0)
    A(1, :) = A(1, :) / 2;
    A = [A(end, :); A(2:end-1, :); A(1, :)];
else
    posi = find(rem(A(1,:), 2) == 0);
    if ~isempty(posi)
        A(2, 1) = max(abs(A(:)));
    else
        A(:, end) = sum(A(:, [1, posi]), 2);
    end
end

disp('A nova matriz é igual a:');
disp(A);
```

Correção realizada:

1. Linha 9: Corrigido o critério de seleção de elementos pares na primeira linha usando `rem(A(1,:), 2) == 0` em vez de `rem(A(1,:), 2) ~= 0`.

Agora o programa deve funcionar corretamente de acordo com a lógica que você descreveu. Desculpe novamente pela confusão anterior.

Figura 27 – Segunda resposta para a primeira versão da resposta do ChatGPT para o Exemplo #5.



Peço desculpas pela confusão anterior. Você está certo, a definição da variável `posi` precisa ser corrigida. Aqui está o programa com a correção necessária na definição da variável `posi`:

matlab

Copy code

```
A = input('Matriz A (nxm)=');

% Verifica se todos os elementos da primeira linha são pares
if all(rem(A(1,:), 2) == 0)
    % Substitui a última linha pelos valores da metade da primeira linha
    A(end, :) = A(1, :) / 2;

    % Substitui a primeira linha pelos valores da última linha
    A(1, :) = A(end, :);
else
    % Encontra as posições dos elementos pares na primeira linha
    posi = find(rem(A(1,:), 2) == 0);

    if ~isempty(posi) % Verifica se há elementos pares na primeira linha
        % Substitui a última coluna pelos valores da soma da primeira coluna
        A(:, end) = A(:, end) + sum(A(:, posi), 2);
    else
        % Substitui o elemento da primeira coluna da segunda linha pelo máximo
        A(2, 1) = max(max(abs(A)));
    end
end

disp('A nova matriz é igual a:');
disp(A);
```

A correção principal foi na linha que define `posi`, onde troquei `~= 0` por `== 0` para encontrar os elementos pares na primeira linha. Agora o programa deve funcionar conforme o esperado.

Figura 28 – Segunda resposta para a segunda versão da resposta do ChatGPT para o Exemplo #5.

A partir da análise das duas versões propostas e da forma como se chegou a uma versão correta do programa para uma delas, pode, de certa forma, concluir-se que o ChatGPT não soube interpretar o funcionamento do programa. Ele foi capaz de corrigir a sintaxe, mas não a estrutura. Em alguns casos, o ChatGPT só consegue corrigir a estrutura se for guiado pelo utilizador, mas poderá chegar à solução correta apenas por acaso. Assim, tal como já referido, é importante acompanhar a interação com o ChatGPT com a execução das várias tentativas no Matlab para perceber como guiá-lo na obtenção da solução, particularmente em problemas mais longos e complexos. Essa execução acompanhada com o Matlab é ainda mais importante quando se observa que as descrições fornecidas pelo ChatGPT acerca das alterações que introduziu não referem, por vezes, todas essas alterações. Portanto, é fundamental analisar cuidadosamente os programas gerados pelo ChatGPT utilizando o Matlab.

4.3 Utilizar o ChatGPT para criar um programa em MATLAB que resolva um determinado problema

4.3.1 Sexto exemplo – mensagens em português

De modo a obter melhores resultados quando se pede ao ChatGPT para criar um programa que resolva um determinado problema, é importante que a mensagem seja clara e detalhada. De forma a ilustrar este tipo de utilização, pediu-se ao ChatGPT para criar um programa em MATLAB que resolvesse as várias questões colocadas na mensagem apresentada na Figura 29 (Exemplo #6).



Considere que um centro de saúde fez o registo das consultas de todos os seus utentes ao longo dos anos de 2021 e 2022. A informação está organizada sob a forma de um array, designado por SAUDE, onde cada linha contém o registo de uma consulta efetuada a um determinado utente e cada coluna corresponde, respetivamente, a:

Coluna 1: número de identificação da consulta ao utente (número inteiro com 9 algarismos em que os 4 primeiros algarismos correspondem ao ano em que foi efetuada a consulta).
Coluna 2: idade do utente.
Coluna 3: número inteiro que indica o sexo do utente:
1: o utente é do sexo feminino;
2: o utente é do sexo masculino.
Coluna 4: número inteiro que indica a nacionalidade do utente:
1: nacionalidade portuguesa;
2: nacionalidade brasileira;
3: outra nacionalidade.
Coluna 5: data em que foi efetuada a consulta, na forma AAAAMMDD, em que AAAA é o ano, MM é o mês e DD é o dia. Por exemplo, 20220209 representa o dia 9 de fevereiro de 2022.
Coluna 6: número inteiro que indica o tipo de consulta efetuada pelo utente:
0: sem marcação;
1: clínica geral;
2: pediatria;
3: enfermagem.
Coluna 7: número inteiro que indica se foram receitados medicamentos ao utente, pelo seu médico:
0: não houve lugar a receita;
1: foram receitados medicamentos ao utente.
Coluna 8: quantidade de medicamentos receitados ao utente.
Coluna 9: preço a pagar pela consulta.

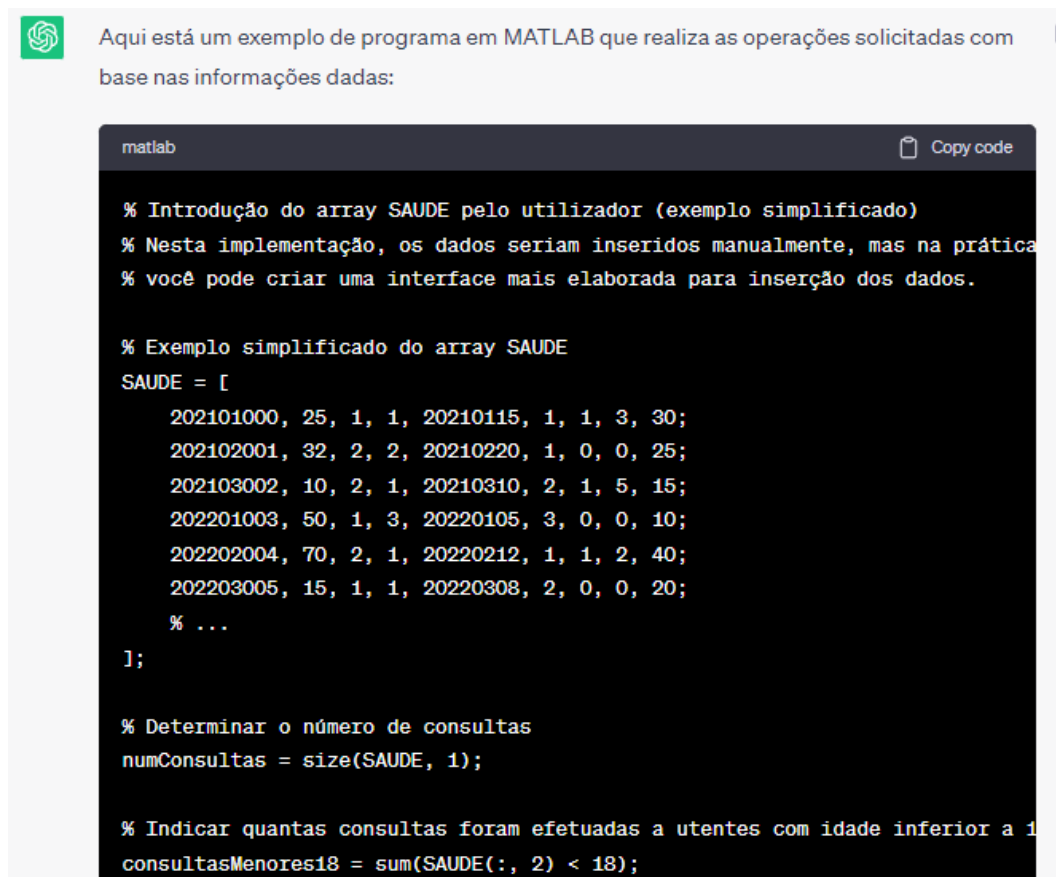
Considerando que o array SAUDE é introduzido pelo utilizador, escreva um programa em Matlab que:

- determine o número de consultas que constam na base de dados;
- indique quantas consultas foram efetuadas a utentes com idade inferior a 18 anos;
- peça ao utilizador o número de identificação da consulta a um utente e;
- escreva uma mensagem de erro, se esse número não constar em SAUDE;
- indique o sexo do utente;
- escreva uma mensagem, caso o médico tenha receitado medicamentos ao utente associado a essa consulta.
- indique a percentagem de consultas de enfermagem efetuadas a utentes com idade superior a 65 anos;
- determine a quantidade mais elevada de medicamentos receitados pelo médico a um utente do sexo masculino numa consulta de clínica geral;



Figura 29 – Mensagem para o Exemplo #6.

Verifica-se que o enquadramento do problema que foi efetuado é longo e detalhado, e que o programa que é pedido deverá responder a várias questões distintas. A resposta obtida do ChatGPT é apresentada nas Figuras 30 e 31. Desta resposta, observa-se que o ChatGPT criou um exemplo para a matriz SAUDE de modo a que o programa desenvolvido pudesse ser executado de forma autónoma. Adicionalmente, verifica-se que o programa desenvolvido contém comentários para os vários passos que executa, os quais indicam qual das questões está a ser resolvida por uma determinada operação ou por um grupo de operações. Uma análise mais detalhada do código produzido pelo ChatGPT mostra que ele responde corretamente às várias perguntas. Da resolução obtida, salienta-se que a resposta à 5ª pergunta (indicar a percentagem de consultas de enfermagem efetuadas a utentes com idade superior a 65 anos) é resolvida com três operações nas quais o programa cria várias submatrizes da matriz SAUDE inicial antes de calcular a percentagem: uma primeira chamada *enfermagemConsultas* apenas com as linhas correspondentes a consultas de enfermagem e uma segunda chamada *enfermagem65Mais* que, a partir de *enfermagem65Mais*, apenas contém as linhas correspondentes a utentes com mais de 65 anos. De igual forma, a resposta à 6ª pergunta (determinar a quantidade mais elevada de medicamentos receitados pelo médico a um utente do sexo masculino numa consulta de clínica geral) é também resolvida com recurso a uma submatriz da matriz SAUDE inicial, *clinicaGeralConsultas*, a qual contém apenas as linhas correspondentes a consultas de clínica geral que envolveram utentes do sexo masculino.



Aqui está um exemplo de programa em MATLAB que realiza as operações solicitadas com base nas informações dadas:

```
matlab Copy code

% Introdução do array SAUDE pelo utilizador (exemplo simplificado)
% Nesta implementação, os dados seriam inseridos manualmente, mas na prática
% você pode criar uma interface mais elaborada para inserção dos dados.

% Exemplo simplificado do array SAUDE
SAUDE = [
    202101000, 25, 1, 1, 20210115, 1, 1, 3, 30;
    202102001, 32, 2, 2, 20210220, 1, 0, 0, 25;
    202103002, 10, 2, 1, 20210310, 2, 1, 5, 15;
    202201003, 50, 1, 3, 20220105, 3, 0, 0, 10;
    202202004, 70, 2, 1, 20220212, 1, 1, 2, 40;
    202203005, 15, 1, 1, 20220308, 2, 0, 0, 20;
    % ...
];

% Determinar o número de consultas
numConsultas = size(SAUDE, 1);

% Indicar quantas consultas foram efetuadas a utentes com idade inferior a 18
consultasMenores18 = sum(SAUDE(:, 2) < 18);
```

Figura 30 – Primeira parte da resposta do ChatGPT para o Exemplo #6.

```

% Pedir ao utilizador o número de identificação da consulta
numeroConsulta = input('Insira o número de identificação da consulta: ');

% Procurar a linha correspondente ao número de identificação da consulta
indiceConsulta = find(SAUDE(:, 1) == numeroConsulta);

% Verificar se o número de identificação da consulta foi encontrado
if isempty(indiceConsulta)
    disp('Número de consulta não encontrado na base de dados.');
```

```

else
    % Indicar o sexo do utente
    sexo = SAUDE(indiceConsulta, 3);
    if sexo == 1
        sexoStr = 'feminino';
    else
        sexoStr = 'masculino';
    end
    disp(['O utente é do sexo ' sexoStr '.']);

    % Verificar se foram receitados medicamentos ao utente
    receitado = SAUDE(indiceConsulta, 7);
    if receitado == 1
        disp('Foram receitados medicamentos ao utente.');
```

```

    else
        disp('Não houve receita de medicamentos ao utente.');
```

```

    end
end

% Calcular a percentagem de consultas de enfermagem a utentes com idade superior a 65 anos
enfermagemConsultas = SAUDE(SAUDE(:, 6) == 3, :);
enfermagem65Mais = enfermagemConsultas(enfermagemConsultas(:, 2) > 65, :);
percentagemEnfermagem65Mais = (size(enfermagem65Mais, 1) / numConsultas) * 100;

% Determinar a quantidade mais elevada de medicamentos receitados pelo médico
clinicaGeralConsultas = SAUDE(SAUDE(:, 6) == 1 & SAUDE(:, 3) == 2, :);
maxMedicamentos = max(clinicaGeralConsultas(:, 8));

disp(['Número total de consultas: ' num2str(numConsultas)]);
disp(['Consultas a utentes com idade inferior a 18 anos: ' num2str(consultasIdadeInf18)]);
disp(['Percentagem de consultas de enfermagem a utentes com idade superior a 65 anos: ' num2str(percentagemEnfermagem65Mais)]);
disp(['Quantidade mais elevada de medicamentos receitados a um utente masculino: ' num2str(maxMedicamentos)]);

```

Lembre-se de que este é um exemplo simplificado e que na prática você precisaria de uma maneira de inserir os dados de forma mais eficiente e robusta, possivelmente usando um arquivo CSV ou interagindo com uma interface gráfica.

Figura 31 – Segunda parte da resposta do ChatGPT para o Exemplo #6.

Tal como acontece em vários problemas deste tipo, a sua resolução não é única sendo possível desenvolver vários códigos com um maior ou menor grau de otimização para atingir os resultados pretendidos. Apesar de não ser a versão mais otimizada possível, verifica-se que o programa proposto pelo ChatGPT apresenta, no entanto, um compromisso adequado desse ponto de vista. Salienta-se,

ainda que, caso o funcionamento do programa não fosse totalmente claro para o utilizador, é sempre possível pedir ao ChatGPT para o explicar passo-a-passo. De modo a ilustrar essa situação, as Figuras 32 a 34 apresentam a mensagem introduzida para esse efeito e a respetiva explicação. Como se pode observar, a explicação tem um nível de detalhe adequado que permite compreender de forma clara o que cada operação faz ao longo do programa.

X

explique o código que foi gerado, passo-a-passo

Claro, vou explicar o código passo a passo para você:

1. Definição do Array de Dados (exemplo simplificado):

matlab

Copy code

```
SAUDE = [  
    202101000, 25, 1, 1, 20210115, 1, 1, 3, 30;  
    202102001, 32, 2, 2, 20210220, 1, 0, 0, 25;  
    % ... (outras linhas de dados)  
];
```

Este é um exemplo simplificado do array SAUDE que contém informações sobre as consultas. Cada linha representa uma consulta com várias colunas correspondendo aos diferentes atributos (número de identificação da consulta, idade, sexo, nacionalidade, data, tipo de consulta, etc.).

1. Número Total de Consultas:

matlab

Copy code

```
numConsultas = size(SAUDE, 1);
```

A função `size(SAUDE, 1)` retorna o número de linhas no array SAUDE, que corresponde ao número total de consultas.

1. Consultas a Utentes Menores de 18 Anos:

matlab

Copy code

```
consultasMenores18 = sum(SAUDE(:, 2) < 18);
```

Aqui, `SAUDE(:, 2)` seleciona todos os valores da segunda coluna (idade) do array SAUDE. A expressão `SAUDE(:, 2) < 18` retorna um vetor lógico onde cada elemento indica se o utente tem menos de 18 anos. A função `sum()` então conta quantos desses elementos são verdadeiros, ou seja, quantas consultas foram feitas a utentes com menos de 18 anos.

Figura 32 – Explicação do programa gerado pelo ChatGPT para o Exemplo #6.

1. Pedindo o Número de Identificação da Consulta ao Utilizador:

```
matlab Copy code  
  
numeroConsulta = input('Insira o número de identificação da consulta: ');
```

A função `input()` solicita ao usuário que insira um valor e o atribui à variável `numeroConsulta`.

1. Procurando a Consulta Correspondente ao Número de Identificação:

```
matlab Copy code  
  
indiceConsulta = find(SAUDE(:, 1) == numeroConsulta);
```

A função `find()` procura no array SAUDE o índice das linhas onde o número de identificação da consulta (primeira coluna) é igual ao número inserido pelo usuário.

1. Verificando se a Consulta foi Encontrada:

```
matlab Copy code  
  
if isempty(indiceConsulta)  
    disp('Número de consulta não encontrado na base de dados.');else  
    % ... (código a seguir é executado se a consulta foi encontrada)  
end
```

A função `isempty()` verifica se o vetor `indiceConsulta` está vazio. Se estiver vazio, significa que o número de identificação da consulta não foi encontrado na base de dados e uma mensagem de erro é exibida. Caso contrário, o código dentro do bloco "else" é executado.

1. Determinando o Sexo do Utente:

```
matlab Copy code  
  
sexo = SAUDE(indiceConsulta, 3);  
if sexo == 1  
    sexoStr = 'feminino';  
else  
    sexoStr = 'masculino';  
end  
disp(['O utente é do sexo ' sexoStr '.']);
```

Figura 33 – Explicação do programa gerado pelo ChatGPT para o Exemplo #6 (continuação).

Aqui, `SAUDE(indiceConsulta, 3)` seleciona o valor da terceira coluna (sexo) na linha correspondente à consulta encontrada. Dependendo do valor, uma mensagem é exibida indicando o sexo do utente.

1. Verificando se Foram Receitados Medicamentos:

```
matlab Copy code

receitado = SAUDE(indiceConsulta, 7);
if receitado == 1
    disp('Foram receitados medicamentos ao utente.');
```

```
else
    disp('Não houve receita de medicamentos ao utente.');
```

```
end
```

Semelhante ao passo anterior, `SAUDE(indiceConsulta, 7)` seleciona o valor da sétima coluna (receita de medicamentos) na linha correspondente à consulta encontrada. Dependendo do valor, uma mensagem é exibida indicando se foram ou não receitados medicamentos.

1. Percentagem de Consultas de Enfermagem a Utentes com Mais de 65 Anos:

```
matlab Copy code

enfermagemConsultas = SAUDE(SAUDE(:, 6) == 3, :);
enfermagem65Mais = enfermagemConsultas(enfermagemConsultas(:, 2) > 65, :);
percentagemEnfermagem65Mais = (size(enfermagem65Mais, 1) / numConsultas) * 100;
```

Aqui, `SAUDE(:, 6) == 3` cria um vetor lógico indicando as consultas de enfermagem. `enfermagemConsultas` é uma matriz contendo apenas as linhas correspondentes às consultas de enfermagem. O mesmo princípio é aplicado para identificar consultas de utentes com mais de 65 anos (`enfermagem65Mais`).

1. Quantidade Mais Elevada de Medicamentos Receitados a Utente Masculino em Consulta de Clínica Geral:

```
matlab Copy code

clinicaGeralConsultas = SAUDE(SAUDE(:, 6) == 1 & SAUDE(:, 3) == 2, :);
maxMedicamentos = max(clinicaGeralConsultas(:, 8));
```

`clinicaGeralConsultas` contém as linhas que correspondem a consultas de clínica geral para utentes masculinos. A função `max()` é usada para encontrar a quantidade máxima de medicamentos receitados nessas consultas.

Figura 34 – Explicação do programa gerado pelo ChatGPT para o Exemplo #6 (continuação).

1. Exibição dos Resultados:

matlab

Copy code

```
disp(['Número total de consultas: ' num2str(numConsultas)]);  
disp(['Consultas a utentes com idade inferior a 18 anos: ' num2str(consultas  
disp(['Percentagem de consultas de enfermagem a utentes com idade superior a  
disp(['Quantidade mais elevada de medicamentos receitados a um utente mascul
```

Finalmente, os resultados são exibidos na janela de comando.

Esse código exemplifica como você pode analisar os dados do array SAUDE e realizar diversas operações utilizando MATLAB. Lembre-se de que esse código é uma simplificação e não considera situações como entrada de dados incorretos ou ausência de dados.

Figura 35 – Explicação do programa gerado pelo ChatGPT para o Exemplo #6 (continuação).

4.3.2 Sétimo exemplo – mensagens em português

Neste último exemplo, considera-se o caso em que se pretende criar um programa que simule um problema de física. Assim, pretende-se que o programa simule e represente graficamente o movimento uniformemente variado de uma partícula sujeita à aceleração da gravidade com base em determinadas condições iniciais em termos de velocidade e posição. Ao contrário do exemplo anterior, a mensagem utilizada é bastante curta, não sendo sequer indicadas as equações da cinemática que regem o referido movimento da partícula. A mensagem utilizada é apresentada na Figura 36 (Exemplo #7) enquanto que o programa desenvolvido pelo ChatGPT é apresentado nas Figuras 37 e 38.

A análise do programa apresentado indica que, globalmente, o ChatGPT foi capaz de interpretar o que se pretendia, tendo utilizado as equações corretas para definir o movimento uniformemente variado de uma partícula, considerando um referencial direto usual e o Sistema Internacional de Unidades. Dado que a mensagem não estabelece informação específica relativamente às condições iniciais do problema (velocidade e posição), o ChatGPT considerou que essa informação deveria ser pedida ao utilizador sob a forma de vetores. Pelo contrário, relativamente à variável tempo a considerar na simulação do problema, o ChatGPT fez uma interpretação incorreta do que se pediu. A mensagem pretendia que a análise do movimento fosse efetuada e representada graficamente até um instante de tempo fornecido pelo utilizador. No entanto, o ChatGPT não considerou que o instante de tempo pedido seria a duração total para a qual o movimento deveria ser calculado e pediu separadamente dois dados ao utilizador: um instante de tempo t_{max} que corresponde ao tempo máximo para o qual se vai calcular o movimento (Figura 37) e um instante de tempo $t_{visualize}$ cujo ponto na trajetória da partícula seria marcado especificamente no gráfico global da trajetória (Figura 38). Assim, apesar do programa proposto fornecer a informação pretendida, é importante salientar que o ChatGPT não foi capaz de fazer uma interpretação totalmente correta da mensagem.

Observa-se que o facto do ChatGPT ter feito esta interpretação e pedido ao utilizador os dois instantes de tempo distintos t_{max} e $t_{visualize}$ tem uma implicação prática importante que o programa proposto não considera: se o valor de $t_{visualize}$ inserido for superior a t_{max} , o programa vai dar um erro. De forma a contornar esta situação, o programa deveria dar uma mensagem de aviso caso $t_{visualize}$ fosse superior a t_{max} , pedindo um novo valor de $t_{visualize}$ nesse caso.

 Escreva um programa em Matlab que simule a equação do movimento a duas dimensões de uma partícula ao longo do tempo sujeita a um vetor velocidade inicial e à aceleração da gravidade. O programa deve pedir ao utilizador o vetor velocidade inicial, o vetor posição inicial da partícula e deve calcular as posições x e y da partícula ao longo do tempo. Além disso o programa deve criar um gráfico que representa essas posições ao longo do tempo até um instante de tempo pedido ao utilizador. 

Figura 36 – Mensagem para o Exemplo #7.

Em termos do algoritmo, o programa também estabelece que o movimento é calculado para 1000 intervalos de tempo entre zero e o tempo máximo t_{max} pedido ao utilizador. Relativamente ao cálculo do movimento propriamente dito, observa-se que o programa opta por uma formulação incremental, a qual obriga a atualizar a velocidade inicial para cada novo instante de tempo (essa atualização é efetuada de forma correta no programa). Esta formulação não estando incorreta, não é a mais eficiente, sendo possível calcular o movimento com base numa formulação total sempre a partir das condições iniciais fornecidas pelo utilizador e substituindo a variável dt por uma posição de um vetor tempo discretizado nos vários instantes de tempo entre zero e t_{max} . Utilizando uma formulação total seria ainda possível otimizar o programa de modo a efetuar os vários cálculos sem recorrer ao ciclo *for...end* proposto pelo ChatGPT.

Ainda no que respeita à implementação proposta, salienta-se que o ChatGPT considera outra opção que não foi referida na mensagem. Durante o ciclo de repetição *for...end*, verifica-se que se a variável y for negativa, a execução do ciclo é interrompida (com a instrução *break*). Com esta opção, o ChatGPT está a assumir que $y = 0$ corresponde à posição do solo, não sendo por isso possível a partícula ter movimento abaixo desse nível. Dado que a mensagem nada refere em relação a esta questão, esta interpretação do ChatGPT pode ou não estar correta, dependendo do problema que o utilizador queira representar. Caberá ao utilizador estar atento a esta questão e adaptar o programa proposto caso seja necessário. Independente desta questão, salienta-se que, caso o ciclo seja interrompido, a questão anteriormente referida relacionada com o valor de $t_{visualize}$ inserido pelo utilizador é um pouco mais complexa pois neste caso $t_{visualize}$ deveria ser comparado com o instante de tempo em que $y = 0$.

Relativamente ao gráfico criado pelo programa (ver o exemplo da Figura 39), observa-se que o ChatGPT tomou uma série de opções adequadas relativamente ao seu formato, particularmente dado que a mensagem não contém detalhes sobre o tipo de gráfico ou o seu formato. Uma análise mais cuidada do gráfico revela, no entanto, que a trajetória do movimento da partícula está representada para valores negativos da posição y , ao contrário do que poderia ser esperado dada a referida interrupção do ciclo de repetição *for...end*. O que acontece, e que pode ser confirmado pela análise dos valores da variável y , é que o ciclo é apenas interrompido quando for encontrado o primeiro valor de y negativo e que esse valor não é eliminado da variável y . Assim, o gráfico irá sempre representar a trajetória do movimento até ao primeiro ponto que tem uma posição y negativa, ou até ao final do tempo máximo introduzido caso não se atinja uma posição y negativa até esse valor de tempo.

Globalmente, conclui-se que, mesmo não sendo particularmente eficiente, o programa proposto pelo ChatGPT fornece uma boa base para a compreensão do problema físico em causa. Neste contexto, salienta-se que o próprio ChatGPT refere que este programa é um exemplo simples que pode ser melhorado consoante as necessidades do utilizador. Observa-se, ainda, que o programa toma algumas opções que não são totalmente consistentes com a mensagem. No entanto, deve salientar-se que o enquadramento feito na mensagem é pouco detalhado, sendo possível obter um programa mais ajustado e completo se a mensagem for mais detalhada. Mesmo dessa forma, o utilizador deverá sempre confirmar e prever as várias contingências que o programa deve contemplar.



Claro! Aqui está um exemplo de programa em MATLAB que faz o que você descreveu:



matlab

Copy code

```
% Pedir ao usuário o vetor velocidade inicial
v0 = input('Digite o vetor velocidade inicial [vx, vy]: ');

% Pedir ao usuário o vetor posição inicial
p0 = input('Digite o vetor posição inicial [x0, y0]: ');

% Pedir ao usuário o instante de tempo máximo
t_max = input('Digite o instante de tempo máximo: ');

% Aceleração da gravidade
g = [0, -9.81]; % Aceleração na direção y

% Criar vetores para armazenar as posições x e y ao longo do tempo
num_steps = 1000; % Número de passos de tempo
dt = t_max / num_steps; % Intervalo de tempo
x = zeros(1, num_steps);
y = zeros(1, num_steps);

% Inicializar posições iniciais
x(1) = p0(1);
y(1) = p0(2);

% Simulação do movimento
for step = 2:num_steps
    % Atualizar posições usando a equação do movimento
    x(step) = x(step-1) + v0(1) * dt;
    y(step) = y(step-1) + v0(2) * dt + 0.5 * g(2) * dt^2;

    % Atualizar a velocidade na direção y devido à aceleração da gravidade
    v0(2) = v0(2) + g(2) * dt;

    % Parar a simulação se a partícula atingir o solo (y < 0)
    if y(step) < 0
        num_steps = step;
        break;
    end
end

% Plot das posições ao longo do tempo
figure;
plot(x(1:num_steps), y(1:num_steps), 'b');
xlabel('Posição x');
ylabel('Posição y');
title('Movimento de uma Partícula');
grid on;
```

Figura 37 – Primeira parte da resposta do ChatGPT para o Exemplo #7.

```

% Perguntar ao usuário o instante de tempo para o qual deseja visualizar o g
t_visualize = input('Digite o instante de tempo para visualizar o gráfico: ');

% Encontrar o índice correspondente ao instante de tempo fornecido
idx_visualize = min(round(t_visualize / t_max * num_steps), num_steps);

% Marcar o ponto correspondente no gráfico
hold on;
plot(x(idx_visualize), y(idx_visualize), 'ro');
text(x(idx_visualize), y(idx_visualize), [' t = ' num2str(t_visualize)], 'Ve

% Mostrar o gráfico
legend('Trajetória', 'Instante de tempo');
hold off;

```

Copie e cole o código acima em um arquivo `.m` no MATLAB. Certifique-se de que o arquivo esteja no mesmo diretório em que você está executando o MATLAB. O programa pedirá os vetores de velocidade e posição iniciais, bem como o instante de tempo para visualização do gráfico. Lembre-se de que este é apenas um exemplo básico e pode ser aprimorado para se adequar às suas necessidades específicas.

Figura 38 – Segunda parte da resposta do ChatGPT para o Exemplo #7.

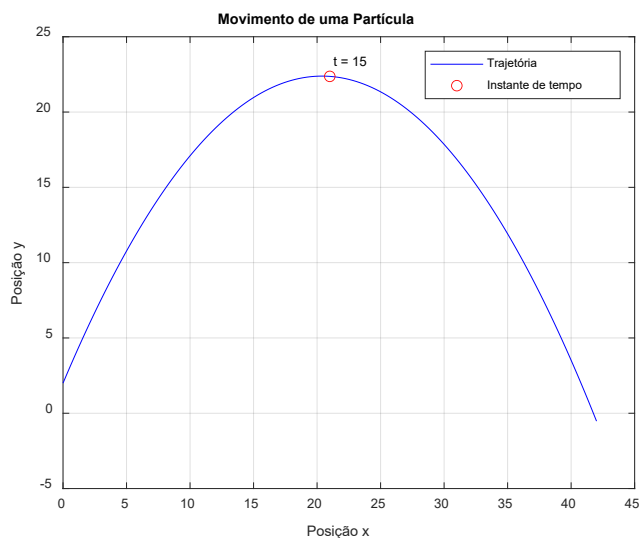


Figura 39 – Gráfico criado pelo programa do Exemplo #7 obtido com um vetor velocidade inicial [10 20], um vetor posição inicial [0 2], tempo máximo de análise de 40s e marcando o ponto para um tempo de 15s.