

Roteiro para o Primeiro Trabalho de Programação - Parte 2

Solução de sistemas lineares - Triangular Superior - Diagonal

Augusto Cesar de Castro Barbosa e Luiz Mariano Carvalho
Departamento de Matemática Aplicada
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro



Este documento está organizado da seguinte forma:

- na Seção 1, apresentamos informações sobre o início do trabalho, sobre as notas de trabalhos plagiados, sobre os arquivos disponíveis e sobre regras de uso de funções do Octave;
- na Seção 2, explicamos os significados dos parâmetros de entrada e saída das funções a serem programadas;
- na Seção 3, sugerimos procedimentos de edição e testagem dos códigos no AVA;
- na Seção 4, mostramos como submeter o trabalho para correção automática no AVA e
- na Seção 5, informamos o prazo final de entrega.

1 Orientações iniciais

Esta atividade está disponível no AVA-UERJ (<https://ava.pr1.uerj.br/>), no tópico “Primeiro Trabalho de Programação”, no curso “SALA02IMECNUM”. Escolha a opção “Trabalho 1 - Parte 2: Códigos para sistemas lineares com matrizes diagonais ou triangulares superiores”, ver Figura 1.

Primeiro Trabalho de Programação

Oculto para estudantes

São quatro atividades, você pode escolher fazer apenas uma, duas, três ou as quatro. A nota será a maior dentre os programas que você entregar:

1. solução de sistemas lineares com matrizes diagonais ou triangulares inferiores;
2. solução de sistemas lineares com matrizes diagonais ou triangulares superiores;
3. método da bisseção;
4. método da falsa posição.

Cada uma delas tem um roteiro e um Laboratório Virtual de Programação específicos.

O prazo de entrega é 30/04/2021.

Aviso:

Os programas são individuais. O AVA tem como reconhecer facilmente trabalhos semelhantes. Caso alguns trabalhos sejam semelhantes, após a avaliação e confirmação pelos professores da acuidade da classificação de plágio feita pelo AVA, a nota será zero, para todos os trabalhos em questão. Trabalhos feitos individualmente são sempre diferentes!!!

	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input checked="" type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Figura 1: No Tópico “Primeiro Trabalho de Programação”, escolha a opção “Trabalho 1 - Parte 2: Códigos para sistemas lineares com matrizes diagonais ou triangulares superiores”

1. Caso você ainda não tenha feito algum treinamento com o Laboratório Virtual de Programação, assista pelo menos o primeiro vídeo (Treinamento LVP - (1/4)), onde está descrito como começar a trabalhar no LVP, ver Figura 2 . Depois, assista os vídeos e faça as atividades que estão no tópico “Segundo Treinamento com o Laboratório Virtual de Programação”, ver Figura 3 ;
2. Os programas são individuais. O AVA tem como reconhecer facilmente trabalhos semelhantes. **Caso alguns trabalhos sejam semelhantes, após a avaliação e confirmação pelos professores da acuidade da classificação de plágio feita pelo AVA, a nota será zero, para todos os trabalhos identificados.** Trabalhos feitos individualmente são sempre diferentes!
3. Entre na aba “Editar”;
4. Já vão estar disponíveis três códigos:
 - a) **principalsptd.m**: este código **não pode ser alterado**. Qualquer alteração nele impugnará o trabalho;
 - b) **subparatras.m**: este código resolve um sistema triangular superior, além disso faz três testes de sanidade: se a matriz é quadrada, se o lado direito tem dimensões consistentes com a matriz dos coeficientes e se a matriz tem algum elemento nulo na diagonal principal. Caso ocorra algum destes fatos, o código para a execução e devolve uma informação precisa do problema ocorrido;







Primeiro Treinamento com o Laboratório Virtual de Programação

Restrito Disponível a:

- É depois de **22 fevereiro 2021, 13:00**
- É antes do final de **30 de maio de 2021**

Estas atividades visam o treinamento para o uso do Laboratório de Programação Virtual. Trata-se de um ambiente para escrever e submeter códigos em Octave e para correção automática dos programas submetidos.

- todo o código deve ser digitado, a interface não aceita importação de texto através de cópias;
- a saída dos programas devem estar estritamente iguais às pedidas no trabalho, pois a correção será automática;
- coloquem as dúvidas no fórum relativo aos ambientes;
- tirem as dúvidas nos atendimentos.

 Orientações para o treinamento	<input checked="" type="checkbox"/>
Apresenta o programa e as funções a serem implementados.	
 Treinamento LVP - (1/4)	<input checked="" type="checkbox"/>
Inicializa o servidor através de um programa simples.	
 Treinamento LVP - (2/4)	<input checked="" type="checkbox"/>
Implementa um programa para ser testado com a correção automática.	
 Treinamento LVP - (3/4)	<input checked="" type="checkbox"/>
Implementa duas funções simples para serem chamadas pelo programa principal.	
 Treinamento LVP - (4/4)	<input checked="" type="checkbox"/>
Submete o programa e as funções para a correção automática e mostra alguns erros comuns de formatação.	
 Treinamento com o LVP	<input checked="" type="checkbox"/>

Restrito Disponível a partir de **31 de outubro de 2020**

Figura 2: Faça o “Primeiro Treinamento com o Laboratório Virtual de Programação”.

Segundo Treinamento com o Laboratório Virtual de Programação

Restrito Disponível a partir de **11 de novembro de 2020**







 Roteiro para o treinamento	<input checked="" type="checkbox"/>
 Slides com as operações a serem implementadas	<input checked="" type="checkbox"/>
Aqui se encontram as operações a serem implementadas, a matriz em questão e de ordem quatro. No entanto, a função implementada deve aceitar uma matriz de qualquer ordem.	
 Segundo Treinamento com VLP - parte (1/3)	<input checked="" type="checkbox"/>
 Segundo Treinamento com VLP - parte (2/3)	<input checked="" type="checkbox"/>
 Segundo Treinamento com VLP - parte (3/3)	<input checked="" type="checkbox"/>
 Treinamento com Fatoração LU	<input checked="" type="checkbox"/>

Figura 3: Faça o “Segundo Treinamento com o Laboratório Virtual de Programação”.

- c) **diagonal.m**: este código resolve um sistema diagonal, além disso faz três testes de sanidade: se a matriz é quadrada, se o lado direito tem dimensões consistentes com a matriz dos coeficientes e se a matriz tem algum elemento nulo na diagonal principal. Caso ocorra algum destes fatos, o código para a execução e devolve uma informação precisa do problema ocorrido.
5. **Não pode usar o comando barra invertida para resolver os sistemas**, caso seja usado, o trabalho será impugnado. Funções simples, como, por exemplo, cálculo de determinante ou norma de vetores e matrizes, podem ser utilizadas. Caso tenha alguma dúvida, coloque uma pergunta no Fórum de Programação para esclarecermos se uma função pode ou não ser utilizada no seu programa.

2 Programação

Os códigos subparatras.m e diagonal.m têm os mesmos parâmetros de entrada e saída:

Parâmetros de entrada

1. **a**: a matriz dos coeficientes;
2. **b**: o lado direito;
3. **exa**: a solução exata do problema.

Parâmetros de saída

1. **sol**: caso “info” seja igual a 1, devolve a solução calculada, caso “info” diferente de 1, devolve Inf;
2. **info**: veja abaixo os valores possíveis e seus significados;
3. **res**: caso “info” seja igual a 1, devolve o resíduo calculado como sendo a norma da diferença ($a * sol - b$), caso “info” diferente de 1, devolve Inf;
4. **err**: caso “info” seja igual a 1, devolve o erro, dado pela norma da diferença ($sol - exa$), caso “info” diferente de 1, devolve Inf.

Significado dos valores do parâmetro **info**

1. **info=1**: o programa calculou a solução exata com erro e resíduo muito próximos ou iguais a zero;
2. **info=2**: a matriz não é quadrada;
3. **info=3**: o lado direito tem número de linhas diferente do número de linhas da matriz dos coeficientes;
4. **info=4**: foi encontrado um elemento nulo na diagonal principal da matriz dos coeficientes.

Programação de subparatras.m

1. Programe a substituição para frente de um sistema linear triangular superior;
2. Programe os três testes de sanidade: matriz quadrada, sem zeros na diagonal principal e com mesmo número de linhas do que o lado direito;
3. Para facilitar o seu trabalho, faça os programas em seu ambiente (Octave, CoCalc, ANOC, etc) e vá para o AVA somente quando você estiver com o programa funcionando corretamente. Pode copiar os códigos que estão no AVA para seu ambiente local;
4. Use as matrizes e os lados direitos que estão disponíveis no programa principal-sptd.m para testar as suas funções em seu ambiente;
5. Há várias formas de implementar essa função, na própria aula sobre sistemas triangulares há duas alternativas, mas há outras. Implemente a que você tiver mais confiança;
6. avalie o resíduo calculando a norma da diferença $a * sol - b$;
7. avalie o erro calculando a norma da diferença $sol - exa$;
8. A seguir, vamos colocar os comentários e alguns comandos para orientar a sua programação:

```
1 function [sol info res err] = subparatras(a,b,exa)
2
3 % eh necessario calcular as dimensoes da matriz para
4 %   realizar as proximas operacoes
5
6 % antes e fazer o restante do codigo realize os testes de
7 % sanidade dos dados de entrada,
8 % caso algum dado esteja ruim, pare a execucao
9 % e informe ao programa principalsptd.
10
11 % eh melhor guardar espaco para o vetor solucao
12
13 % voce pode começar a calcular a solucao fora do loop
14 % principal, mas pode deixar para fazer todas as operacoes
15 % dentro do loop tambem.
16
17
18 % aqui começa o loop principal. Ele terá uma quantidade
19 % de passos que depende se voce começou a calcular
20 % a solucao fora do loop principal ou vai calcular
21 % apenas dentro do loop.
```

```

22 %
23
24 for k = ? : ? % aqui os limites vao depender
25               % da observacao acima
26
27 % ha diversas maneiras de se fazer este calculo , faca um
28 % esboco em papel antes de comecar a programar.
29
30 % voce pode armazenar as operacoes entre as partes da
31 % solucao ja calculadas e da matriz , mas voce tambem
32 % pode fazer todas as operacoes de uma so vez. Escolha
33 % a que voce compreenda melhor ou crie a sua propria
34
35 % nao se esqueca de terminar o for
36 endfor
37
38 % nao esqueca de colocar o comando com o fim da funcao
39 endfunction

```

9. não se esqueça de salvar seu trabalho.

Programação de diagonal.m É muito semelhante à função `subsparatras.m`, apenas troque o loop principal para a solução de um sistema diagonal. Neste caso, também há muitas formas de fazer estas operações, crie a sua.

3 Testagem no ambiente AVA

Os testes de correção de seus códigos são feito acionando o programa “principal-sptd.m”. Este programa não pode ser modificado, ele gera matrizes que testam se o seu código está atendendo as orientações passadas neste trabalho. Como explicado abaixo, a cada entrada correspondem algumas saídas que mostram que o seu código está correto ou não. A seguir, vamos detalhar este processo:

1. após testar os códigos no seu ambiente, vá para o ambiente AVA e escolha no Tópico “Primeiro Trabalho de Programação” a opção “Trabalho 1 - Parte 1: Códigos para sistemas lineares com matrizes diagonais ou triangulares superiores”, ver Figura 4.
2. observe que já existem os arquivos `principalsptd.m`, `subsparatras.m` e `diagonal.m`, ver Figura 5;
3. digite os seus códigos nos arquivos `subsparatras.m` e `diagonal.m`, sem alterar os nomes das funções ou os parâmetros de entrada e de saída;

Primeiro Trabalho de Programação

Oculto para estudantes

São quatro atividades, você pode escolher fazer apenas uma, duas, três ou as quatro. A nota será a maior dentre os programas que você entregar:

1. solução de sistemas lineares com matrizes diagonais ou triangulares inferiores;
2. solução de sistemas lineares com matrizes diagonais ou triangulares superiores;
3. método da bisseção;
4. método da falsa posição.

Cada uma delas tem um roteiro e um Laboratório Virtual de Programação específicos.

O prazo de entrega é 30/04/2021.

Aviso:

Os programas são individuais. O AVA tem como reconhecer facilmente trabalhos semelhantes. Caso alguns trabalhos sejam semelhantes, após a avaliação e confirmação pelos professores da acuidade da classificação de plágio feita pelo AVA, a nota será zero, para todos os trabalhos em questão. Trabalhos feitos individualmente são sempre diferentes!!!









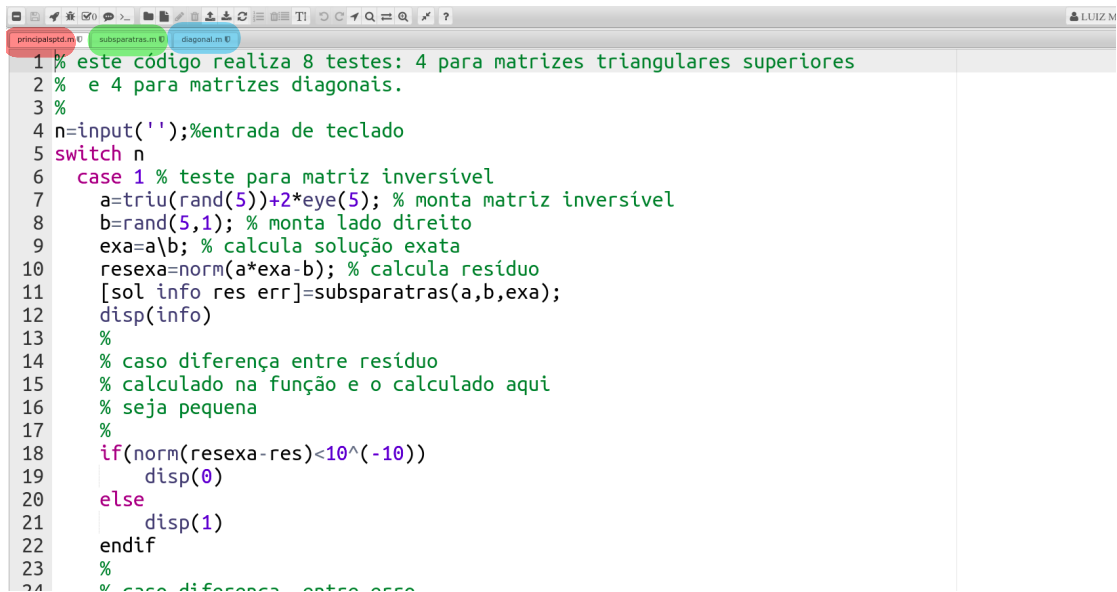
 Trabalho 1 - Parte 1: Roteiro para sistemas lineares com matrizes diagonais ou triangulares inferiores	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 1: Códigos para sistemas lineares com matrizes diagonais ou triangulares inferiores	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 2: Roteiro para sistemas lineares com matrizes diagonais ou triangulares superiores	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 2: Códigos para sistemas lineares com matrizes diagonais ou triangulares superiores	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 3: Roteiro para o método da bisseção	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 3: Códigos para o método da bisseção	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 4: Roteiro para o método da falsa posição	<input checked="" type="checkbox"/>
 Trabalho 1 - Parte 4 Códigos para o método da falsa posição	<input checked="" type="checkbox"/>

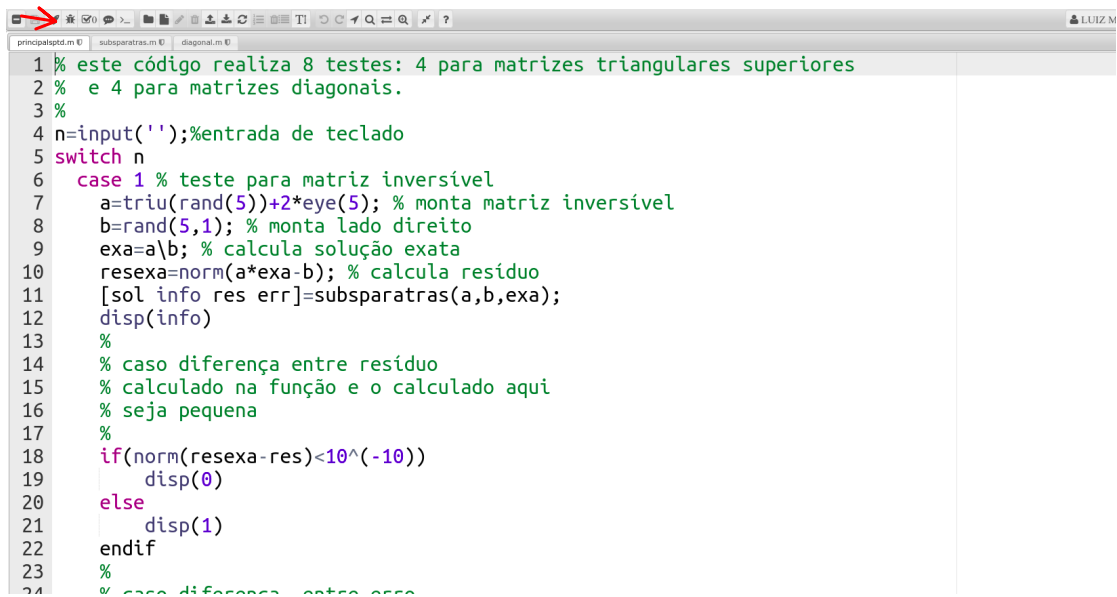
Figura 4: No Tópico “Primeiro Trabalho de Programação”, escolha a opção “Trabalho 1 - Parte 2: Códigos para sistemas lineares com matrizes diagonais ou triangulares superiores”



```
1 % este código realiza 8 testes: 4 para matrizes triangulares superiores
2 % e 4 para matrizes diagonais.
3 %
4 n=input('');%entrada de teclado
5 switch n
6 case 1 % teste para matriz inversível
7     a=triu(rand(5))+2*eye(5); % monta matriz inversível
8     b=rand(5,1); % monta lado direito
9     exa=a\b; % calcula solução exata
10    resex=norm(a*exa-b); % calcula resíduo
11    [sol info res err]=subsparatras(a,b,exa);
12    disp(info)
13    %
14    % caso diferença entre resíduo
15    % calculado na função e o calculado aqui
16    % seja pequena
17    %
18    if(norm(resex-res)<10^(-10))
19        disp(0)
20    else
21        disp(1)
22    endif
23    %
24    % caso diferença entre erro
```

Figura 5: Já existem os arquivos *principalsptd.m*, *subsparatras.m* e *diagonal.m*. O código *principalsptd.m* não pode ser alterado.

4. há 8 testes disponíveis, quatro, de 1 a 4, para a função *subsparatras* e outros quatro, de 5 a 8, para a função *diagonal*, ver Tabela 1 para as entradas e respectivas saídas corretas de cada teste;
5. para testar se você fez tudo certo, acione a tecla para execução (“foguetinho”) do programa *principalsptd*, ver Figura 6;
6. com isso, deverá abrir uma tela com o Octave, e vai ficar aguardando uma entrada sua pelo teclado. Entre com um número de 1 a 8, ver Figura 7 ;
7. na Figura 8, você vai ver a saída caso tenha teclado o número 1;
8. observem que o número que você digitou, de 1 a 8, também aparece, ou seja, na tela aparecem a sua entrada e as saídas correspondentes, caso as saídas sejam diferentes, há algum erro na sua programação;
9. para testar todas as possibilidades, quando o cursor do Octave ficar disponível, tecle *principalsptd*, depois *enter* e digite um número de 1 a 8, ver Figura 9. Confira na Tabela 1 quais são as entradas disponíveis e as saídas de tela corretas;
10. depois de ter conferido todas as entradas e saídas você poderá submeter o seu trabalho para correção automática pelo AVA, ver na Seção 4.



```
1 % este código realiza 8 testes: 4 para matrizes triangulares superiores
2 % e 4 para matrizes diagonais.
3 %
4 n=input('');%entrada de teclado
5 switch n
6 case 1 % teste para matriz inversível
7     a=triu(rand(5))+2*eye(5); % monta matriz inversível
8     b=rand(5,1); % monta lado direito
9     exa=a\b; % calcula solução exata
10    resex=norm(a*exa-b); % calcula residuo
11    [sol info res err]=subspatras(a,b,exa);
12    disp(info)
13    %
14    % caso diferença entre residuo
15    % calculado na função e o calculado aqui
16    % seja pequena
17    %
18    if(norm(resex-res)<10^(-10))
19        disp(0)
20    else
21        disp(1)
22    endif
23    %
24    % caso diferença entre erro
```

Figura 6: Acione a tecla de execução (“foguetinho”) para rodar os testes.

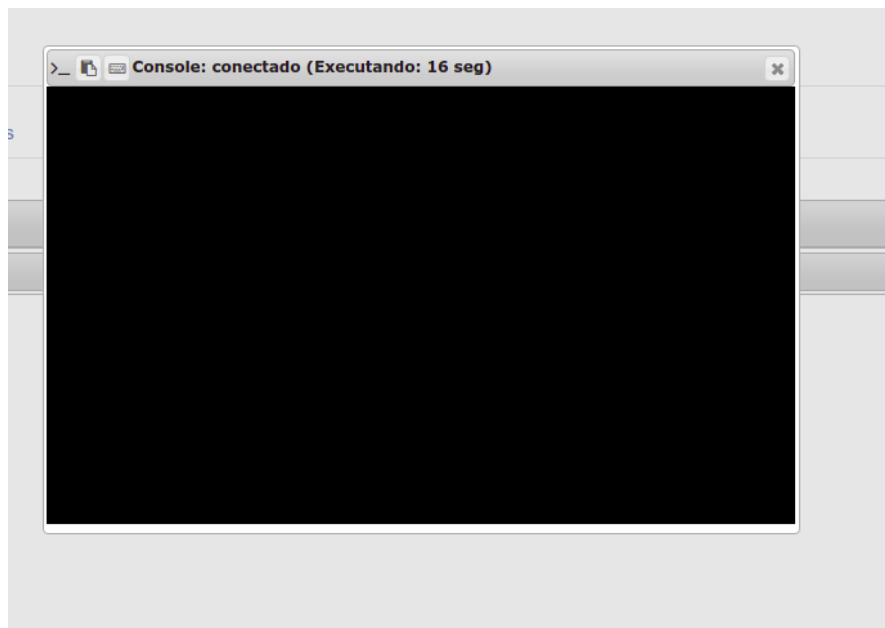


Figura 7: Após acionar o “foguetinho”, o programa aguarda uma entrada sua pelo teclado, digite um número de 1 a 8.

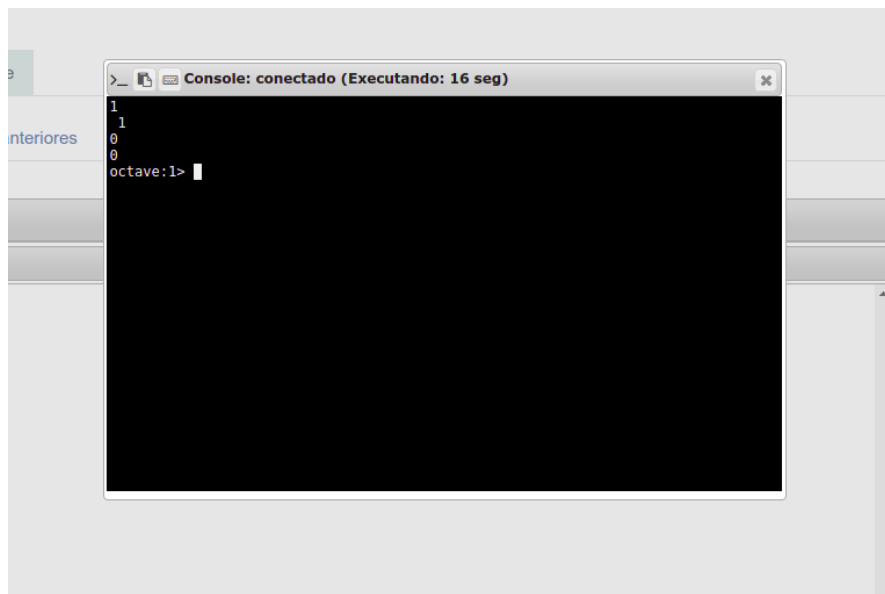


Figura 8: Depois de ter digitado o número 1, aparece esta tela, observe que tem o número que você digitou e a saída esperada do programa: 1 0 0.



Figura 9: Quando o cursor do Octave estiver disponível, digite *principalsptd* e depois *enter*, escolha um número de 1 a 8, neste caso o escolhido foi o 4.

Tabela 1: Entradas de teclado e saídas na tela para o teste do programa `principalsptd`. Na tela preta do Octave, no cursor disponível, digite *principalsptd*, *enter* e um número de 1 a 8 para rodar cada um dos testes.

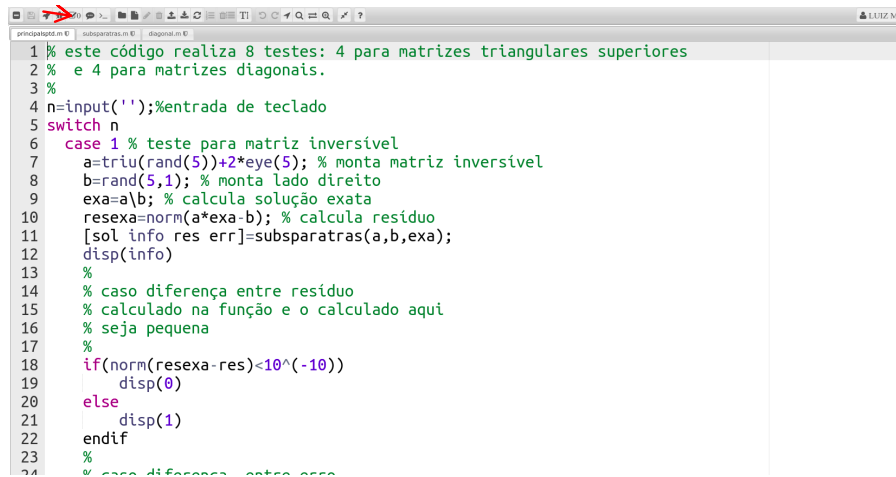
entrada	saída
1	1 0 0
2	2 0 0
3	3 0 0
4	4 0 0
5	1 0 0
6	2 0 0
7	3 0 0
8	4 0 0
outras	erro

4 Correção

1. quando todas as saídas estiverem corretas, ou quando você assim o desejar, acione o botão de correção, ver Figura 10, para rodar a correção automática dos programas;
2. os testes demoram cerca de 30s, em nossas máquinas, podendo variar bastante, aguarde até terminar;
3. como são 8 testes, as notas serão proporcionais à quantidade de testes corretos, variando de 12,5 até 100,0. Na Figura 11, há um exemplo em que os testes foram bem sucedidos. Na Figura 12, dois testes falharam, inclusive há a informação de quais foram as falhas;

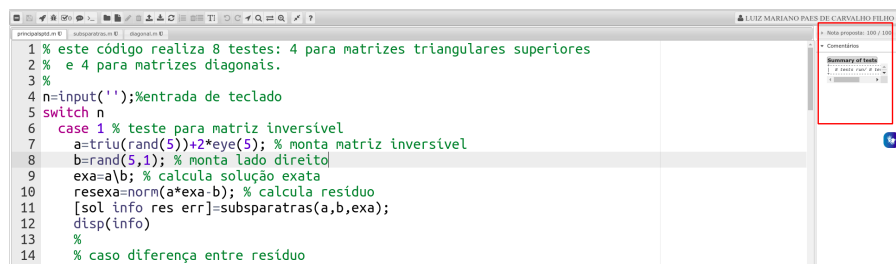
5 Limite para entrega

1. a data limite de entrega é 30/04/2021.



```
1 % este código realiza 8 testes: 4 para matrizes triangulares superiores
2 % e 4 para matrizes diagonais.
3 %
4 n=input('');%entrada de teclado
5 switch n
6     case 1 % teste para matriz inversivel
7         a=triu(rand(5))+2*eye(5); % monta matriz inversivel
8         b=rand(5,1); % monta lado direito
9         exa=a\b; % calcula solução exata
10        resex=norm(a*exa-b); % calcula residuo
11        [sol info res err]=subspatratras(a,b,exa);
12        disp(info)
13        %
14        % caso diferença entre residuo
15        % calculado na função e o calculado aqui
16        % seja pequena
17        %
18        if(norm(resex-res)<10^(-10))
19            disp(0)
20        else
21            disp(1)
22        endif
23    %
24    % caso diferença entre erro
```

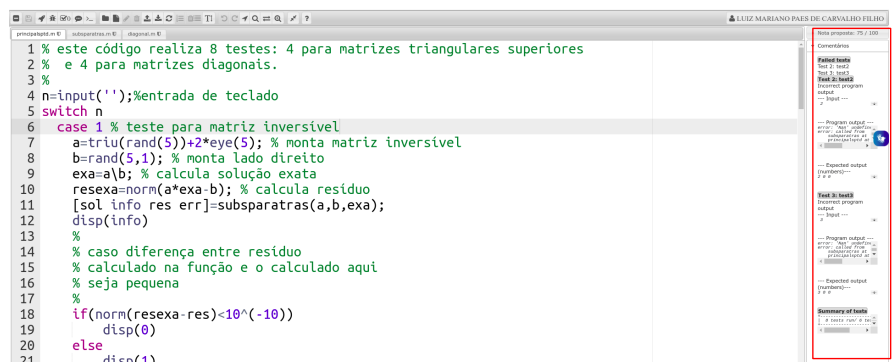
Figura 10: Acione o botão marcado para submeter o seu trabalho para correção automática.



```
1 % este código realiza 8 testes: 4 para matrizes triangulares superiores
2 % e 4 para matrizes diagonais.
3 %
4 n=input('');%entrada de teclado
5 switch n
6     case 1 % teste para matriz inversivel
7         a=triu(rand(5))+2*eye(5); % monta matriz inversivel
8         b=rand(5,1); % monta lado direito
9         exa=a\b; % calcula solução exata
10        resex=norm(a*exa-b); % calcula residuo
11        [sol info res err]=subspatratras(a,b,exa);
12        disp(info)
13        %
14        % caso diferença entre residuo
15        % calculado na função e o calculado aqui
16        % seja pequena
17        %
18        if(norm(resex-res)<10^(-10))
19            disp(0)
20        else
21            disp(1)
22        endif
23    %
24    % caso diferença entre erro
```

Summary of Tests
1. 8 tests out of 8 (100%)
2. 100/100

Figura 11: Todos os testes passaram, nota 100, veja no lado direito da imagem.



```
1 % este código realiza 8 testes: 4 para matrizes triangulares superiores
2 % e 4 para matrizes diagonais.
3 %
4 n=input('');%entrada de teclado
5 switch n
6     case 1 % teste para matriz inversivel
7         a=triu(rand(5))+2*eye(5); % monta matriz inversivel
8         b=rand(5,1); % monta lado direito
9         exa=a\b; % calcula solução exata
10        resex=norm(a*exa-b); % calcula residuo
11        [sol info res err]=subspatratras(a,b,exa);
12        disp(info)
13        %
14        % caso diferença entre residuo
15        % calculado na função e o calculado aqui
16        % seja pequena
17        %
18        if(norm(resex-res)<10^(-10))
19            disp(0)
20        else
21            disp(1)
22        endif
23    %
24    % caso diferença entre erro
```

Summary of Tests
1. 8 tests out of 8 (100%)
2. 75/100

Figura 12: Dois testes falharam, nota 75, veja no lado direito da imagem.

A.C.C. Barbosa e L.M. Carvalho. “Notas de Aula de Cálculo Numérico: Roteiro para o Primeiro Trabalho de Programação - Solução de sistemas lineares - Triangular Superior - Diagonal”, Departamento de Matemática Aplicada, Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Brasil, 2021.

Estas notas podem ser compartilhadas de acordo com a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

