



Universidade Federal de Mato Grosso
Campus Universitário do Araguaia
Instituto de Ciências Exatas e da Terra



Utilização do Software FreeMat no Ensino de Funções, Matrizes e Sistemas Lineares no Ensino Médio

Fernando Henrique Cardoso

Mestrado Profissional em Matemática: PROFMAT/SBM

Orientador: **Prof. Dr. Juan Elmer Villanueva Zevallos**

Trabalho financiado pela Capes

Barra do Garças - MT

Setembro de 2015

Utilização do Software FreeMat no Ensino de Funções, Matrizes e Sistemas Lineares no Ensino Médio

Este exemplar corresponde à redação final da dissertação, devidamente corrigida e defendida por Fernando Henrique Cardoso e aprovada pela comissão julgadora.

Barra do Garças, 14 de Setembro de 2015.

Prof. Dr. Juan Elmer Villanueva Zevallos
Orientador

Banca examinadora:

Prof. Dr. Juan Elmer Villanueva Zevallos
Prof. Dr. Adilson Antônio Berlatto
Prof. Dr. Alonso Sepúlveda Castellanos

Dissertação apresentada ao curso de Mestrado Profissional em Matemática – PROFMAT, da Universidade Federal de Mato Grosso, como requisito parcial para obtenção do título **de Mestre em Matemática**.

Dados Internacionais de Catalogação na Fonte.

C268u Cardoso, Fernando Henrique.
Utilização do Software FreeMat no Ensino de : Funções, Matrizes e Sistemas Lineares / Fernando Henrique Cardoso. -- 2015
xiv, 204 f. : il. color. ; 30 cm.

Orientador: Juan Elmer Villanueva Zevallos.
Dissertação (mestrado profissional) - Universidade Federal de Mato Grosso, Instituto de Ciências Exatas e da Terra, Programa de Pós-Graduação em Matemática, Cuiabá, 2015.
Inclui bibliografia.

1. Ensino. 2. FreeMat. 3. Funções. 4. Matrizes. 5. Sistemas Lineares. I. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Permitida a reprodução parcial ou total, desde que citada a fonte.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE MATO GROSSO
PRÓ-REITORIA DE ENSINO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL - PROFMAT
Av. Fernando Corrêa da Costa, 2367 - Boa Esperança - Cep: 78.060-900 - Cuiabá/MT
Tel : (65) 3615-8713/8710/8576

FOLHA DE APROVAÇÃO

TÍTULO : "Utilização do software FreeMat no ensino de funções, matrizes e sistemas lineares no ensino médio"

AUTOR : Fernando Henrique Cardoso

defendida e aprovada em 04/09/2015.

Composição da Banca Examinadora:

Presidente Banca / Orientador Doutor Juan Elmer Villanueva Zevallos
Instituição : UNIVERSIDADE FEDERAL DE MATO GROSSO

Examinador Interno Doutor Adilson Antônio Berlatto
Instituição : UNIVERSIDADE FEDERAL DE MATO GROSSO

Examinador Externo Doutor Alonso Sepúlveda Castellanos
Instituição : Universidade Federal de Uberlândia

BARRA DO GARÇAS, 04/09/2015.

*Aos meus pais, pois sem eles nada
aconteceria na minha vida.*

Agradecimentos

À Deus pela minha vida e bênçãos recebidas desde o meu caminhar e a Jesus Cristo, meu Salvador, que na cruz entregou sua vida por mim.

Ao professor Dr. Juan Villanueva, pelas orientações e suporte dado no desenvolvimento deste trabalho.

À Vivia Lúcia, que nas dificuldades encontradas no período deste curso me incentivou e motivou a continuar.

Aos meus pais, que sempre estiveram dispostos a ajudar, torcendo pelo sucesso desta longa e difícil caminhada.

Aos colegas de trabalho, em especial, aos professores do núcleo do IFMT de Jaciara pelo apoio dado durante este período.

Aos amigos do mestrado, que ao longo destes dois anos se tornaram parte da minha família.

Um grande abraço e sonoro muito obrigado!!!

Resumo

Este trabalho apresenta algumas abordagens pedagógicas para o ensino da matemática utilizando o software FreeMat. As abordagens caracterizam-se como propostas de atividades metodológicas a serem desenvolvidas e aplicadas no ensino de funções, matrizes e sistemas lineares. Busca-se, facilitar o aprendizado destes conteúdos, pelos discentes do Ensino Médio, através da aplicação deste recurso computacional.

Palavras-chave: Ensino, FreeMat, Funções, Matrizes e Sistemas Lineares.

Abstract

This work presents some pedagogical approaches to teaching mathematics using the FreeMat software. The approaches are characterized as methodological proposals for activities to be developed and applied in teaching functions, matrices and linear systems. The aim is to facilitate the learning of these contents, by the students of high school, through the application of computational resource.

Key-words: Teaching, FreeMat, Function, Matrices and Linear Systems.

Sumário

Agradecimentos	v
Resumo	vi
Abstract	vii
Lista de figuras	xiii
Lista de tabelas	xiv
Introdução	1
1 Ensino de matemática: do giz ao uso do computador	4
1.1 O uso de computadores no ensino de matemática	5
2 FreeMat: dentre vários softwares pagos, uma escolha livre	11
2.1 FreeMat	13
2.2 Programação em FreeMat	18
2.3 Comandos de repetição e de condição	20
2.3.1 Comandos de condição	21
2.3.2 Comando de repetição	23
3 Tópicos de Matemática a serem aplicados no FreeMat	24
3.1 Função	24
3.2 Matrizes	29
3.2.1 Operações com matrizes	32
3.2.2 Aplicação das matrizes à criptografia	42
3.3 Determinantes	48

3.4	Sistemas lineares de m equações com n incógnitas	51
3.4.1	Operações elementares de matrizes	53
3.4.2	Forma escalonada de uma matriz	55
4	Aplicação do FreeMat no estudo de funções, matrizes e sistemas lineares	71
4.1	Proposta I: Uma abordagem para o ensino e aplicação das funções	71
4.1.1	Plotagem gráfica de uma função	72
4.1.2	Como calcular o custo por consumo em m^3 de água?	85
4.1.3	Relato da aplicação da proposta no Ensino Médio	95
4.2	Proposta II: Ensino de matrizes utilizando o FreeMat, das operações com matrizes à criptografia.	98
4.2.1	Ensino de matrizes	98
4.2.2	Criptografando mensagens através do FreeMat	109
4.3	Proposta III: Resolução de sistemas lineares de m equações com n incógnitas	146
	Considerações finais	179
	Referências Bibliográficas	181
	Apêndice: Material adicional	184
A.1	Programa para determinar o valor a pagar para um certo consumo de água das residências em Cuiabá/MT	184
A.2	Programa para criptografar mensagens pelo sistema poligráfico, baseado na multiplicação de matrizes com entradas reais	187
A.3	Programa para decifrar mensagens pelo sistema poligráfico, baseado na multiplicação de matrizes com entradas reais	189
A.4	Programa para criptografar mensagens pelo sistema poligráfico: 2-Cifras de Hill	192
A.5	Programa para decifrar mensagens pelo sistema poligráfico: 2-Cifras de Hill	194
A.6	Programa para resolução de sistemas pela Regra de Cramer	198
A.7	Programa para resolução de sistemas pelo método de Gauss-Jordan	201

Lista de Figuras

2.1	Janelas dispostas na Interface do FreeMat	14
2.2	Janela gráfica do FreeMat	15
2.3	Operações com números reais desenvolvidas no FreeMat	16
2.4	Janela de edição de texto	18
2.5	Local para salvar arquivos digitados na janela de edição	19
2.6	Diretórios do FreeMat	19
3.1	Gráfico da função f	28
3.2	Matriz reduzida por linha à forma escada	56
4.1	Menu da Janela gráfica	74
4.2	Interface do FreeMat após inserção do comando <code>x=linspace(-4,4)</code>	76
4.3	Inserção de comandos para plotar o gráfico da função f	77
4.4	Gráfico da função $f : [-4, 4] \rightarrow \mathbb{R}$, definida por $f(x) = x^3 + x^2 + 1$	77
4.5	Gráfico da função $f : [-4.5] \rightarrow \mathbb{R}$, definida por $f(x) = 10$	78
4.6	Gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$	80
4.7	Gráficos das funções $f, g : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$ e $g(x) = 10$	81
4.8	Ponto de interseção entre os gráficos das funções f e g	81
4.9	Comandos para obter os valores numéricos das funções das locadoras de automóveis A e B	83
4.10	Janela gráfica e os gráficos das funções das locadoras A e B	84
4.11	Gráficos e o ponto de interseção das funções das locadoras A e B	84
4.12	Tabela de Estrutura tarifária e dos serviços complementares da CAB	85
4.13	Faixas de consumo relativa a categoria 2 da conta de água da CAB	86
4.14	Inserção de comandos no FreeMat para desenvolvimento da Atividade 6	89
4.15	Gráfico da função f definida na Atividade 5	90

4.16	Editor do FreeMat e parte dos comandos relativos ao programa “VALORES” . . .	93
4.17	Compilando o programa “VALORES” no FreeMat: primeira etapa do programa	94
4.18	Compilando o programa “VALORES” no FreeMat: segunda etapa do programa	95
4.19	Compilando o programa “VALORES” no FreeMat: terceira etapa do programa	95
4.20	Desenvolvimento da Proposta I com os alunos do IFMT no Laboratório de Informática	97
4.21	Desenvolvimento da Atividade 9 no FreeMat	102
4.22	Desenvolvimento da Atividade 10 no FreeMat	103
4.23	Desenvolvimento da Atividade 11 no FreeMat	104
4.24	Desenvolvimento da Atividade 12 no FreeMat	106
4.25	Desenvolvimento da Atividade 13 no FreeMat	107
4.26	Desenvolvimento da Atividade 13 no FreeMat, utilizando o comando <code>format long</code>	109
4.27	Desenvolvimento da Atividade ?? no FreeMat	110
4.28	Variáveis armazenados e dispostos na janela <i>Variables</i>	110
4.29	Desenvolvimento da Atividade 15 no FreeMat	111
4.30	Desenvolvimento da Atividade 16 no FreeMat	113
4.31	Desenvolvimento da Atividade 17 no FreeMat	114
4.32	Desenvolvimento da Atividade 18 no FreeMat	116
4.33	Desenvolvimento da Atividade 19 no FreeMat	118
4.34	Editor do FreeMat e parte dos comandos relativos ao programa “CRIPTOGRAFAR”	122
4.35	Editor do FreeMat e parte dos comandos relativos ao programa “DECIFRAR” .	125
4.36	Desenvolvimento da Atividade 22, primeira etapa do programa	126
4.37	Desenvolvimento da Atividade 22, segunda etapa do programa.	127
4.38	Desenvolvimento da Atividade 22, terceira etapa do programa	127
4.39	Desenvolvimento da Atividade 22, quarta etapa do programa.	128
4.40	Desenvolvimento da Atividade 23, primeira etapa do programa.	129
4.41	Desenvolvimento da Atividade 23, segunda etapa do programa.	130
4.42	Desenvolvimento da Atividade 23, terceira etapa do programa.	130
4.43	Desenvolvimento da Atividade 23, quarta etapa do programa.	131
4.44	Desenvolvimento da Atividade 26, primeira etapa do programa	140

4.45	Desenvolvimento da Atividade 26, segunda etapa do programa	140
4.46	Desenvolvimento da Atividade 26, terceira etapa do programa	141
4.47	Desenvolvimento da Atividade 26, quarta etapa do programa	142
4.48	Desenvolvimento da Atividade 27, primeira etapa do programa	143
4.49	Desenvolvimento da Atividade 27, segunda etapa do programa	144
4.50	Desenvolvimento da Atividade 27, terceira etapa do programa	144
4.51	Desenvolvimento da Atividade 27, quarta etapa do programa	145
4.52	Desenvolvimento da Atividade 28 no FreeMat	148
4.53	Desenvolvimento da atividade 29 no FreeMat	151
4.54	Desenvolvimento da Atividade 30 no FreeMat	152
4.55	Desenvolvimento da Atividade 31 no FreeMat	154
4.56	Editor do FreeMat e parte dos comandos relativos ao programa “CRAMER” . .	158
4.57	Desenvolvimento da Atividade 33, primeira etapa do programa CRAMER . . .	159
4.58	Desenvolvimento da 33, segunda etapa do programa CRAMER	160
4.59	Desenvolvimento da Atividade 33, terceira etapa do programa CRAMER	160
4.60	Desenvolvimento da Atividade 33, quarta etapa do programa CRAMER	161
4.61	Desenvolvimento da Atividade 34, primeira etapa do programa CRAMER	162
4.62	Desenvolvimento da Atividade 34, segunda etapa do programa CRAMER	163
4.63	Desenvolvimento da Atividade 34, terceira etapa do programa CRAMER	163
4.64	Editor do FreeMat e parte dos comandos relativos ao programa “sistemas” . . .	168
4.65	Desenvolvimento da Atividade 36, primeira etapa do programa Gauss Jordan .	169
4.66	Desenvolvimento da Atividade 36, segunda etapa do programa Gauss Jordan .	170
4.67	Desenvolvimento da Atividade 36, terceira etapa do programa Gauss Jordan . .	171
4.68	Desenvolvimento da Atividade 36, quarta etapa do programa Gauss Jordan . . .	171
4.69	Desenvolvimento da Atividade 36, quinta etapa do programa Gauss Jordan	172
4.70	Desenvolvimento da Atividade 37, primeira etapa do programa Gauss Jordan .	173
4.71	Desenvolvimento da Atividade 37, segunda etapa do programa Gauss Jordan .	174
4.72	Desenvolvimento da Atividade 37, terceira etapa do programa Gauss Jordan . .	174
4.73	Desenvolvimento da Atividade 37, quarta etapa do programa Gauss Jordan . .	175
4.74	Desenvolvimento da Atividade 38, primeira etapa do programa Gauss Jordan .	176
4.75	Desenvolvimento da Atividade 38, segunda etapa do programa Gauss Jordan .	176
4.76	Desenvolvimento da Atividade 38, terceira etapa do programa Gauss Jordan . .	177

4.77	Desenvolvimento da Atividade 38, terceira etapa do programa Gauss Jordan . .	177
4.78	Desenvolvimento da Atividade 38, quarta etapa do programa Gauss Jordan . . .	178

Lista de Tabelas

2.1	Softwares matemáticos livres	12
2.2	Comandos utilizados nas operações com números reais e matrizes	15
2.3	Algumas funções e comandos utilizados no FreeMat	17
2.4	Operadores lógicos	20
2.5	Operadores relacionais	20
3.1	Inversos multiplicativos em \mathbb{Z}_{27}	41
3.2	Tabela de correspondência	43

Introdução

O ensino de matemática é apresentado ao aluno desde as primeiras séries do ensino básico. Já, no Ensino Médio, o discente depara-se com novos conteúdos matemáticos, caso das matrizes e dos sistemas lineares, ou aprofunda o estudo dos tópicos apresentados no nível fundamental, caso das funções. Os conceitos de *função*, *matriz* e *sistema linear* são trabalhados dentro da disciplina de matemática do primeiro e segundo ano do Ensino Médio e aplicados em outras áreas do conhecimento. Estes conteúdos podem ser trabalhados de forma interdisciplinar e contextualizada, todavia, para isto, o discente precisa compreender suas propriedades algébricas e construções gráficas, acarretando uma maior facilidade na aplicação destes tópicos em exercícios ligados ao cotidiano ou visando a interdisciplinaridade.

Algumas disciplinas necessárias para a interação da criança ou adolescente com meios tecnológicos, que facilitariam a inserção e aplicação da tecnologia nas áreas das ciências exatas, não são abordados na Matriz Curricular do Ensino Médio, como exemplo, o ensino de informática. Ainda, a matemática pode proporcionar esta relação com a tecnologia, porém, isto não é alcançado. Observa-se por exemplo, no ensino de funções a nível médio, professores habituados a esboçar gráficos somente no quadro, remetendo à um ensino de caráter tradicional. Também, evidencia-se o desestímulo do discente tanto no estudo algébrico das funções quanto na abordagem geométrica, uma vez que, os docentes encontram dificuldades ao trazer aplicações desse conteúdo, bem como, para torná-lo mais dinâmico.

No caso do ensino de matrizes, cujo conteúdo é extremamente aplicado na programação computacional, não é diferente. Presenciasse dificuldades dos docentes em apresentar e propor meios facilitadores para o ensino desse conteúdo nas escolas, também, além dos alunos não observarem aplicações no cotidiano, muitas vezes, devidos estas dificuldades, não são apresentadas nenhuma aplicação das matrizes nas soluções de sistemas

lineares, o que poderia ser facilmente revertido ao apresentar programas computacionais.

Neste contexto, esta dissertação tem por objetivo apresentar uma nova abordagem para o ensino das funções, matrizes e sistemas lineares, utilizando como recurso computacional, o FreeMat, bem como, delimitar passos a serem seguidos na utilização deste software para o ensino dos conteúdos citados.

Este trabalho está estruturado em quatro capítulos, sendo os três primeiros relativos a um estudo teórico para amparar o entendimento das propostas, que serão apresentadas no quarto capítulo.

No Capítulo 1, serão apresentados aportes teóricos, na área de ensino, para o entendimento das propostas dentro das concepções didáticas e centrados na concepção construtivista do uso do computador em sala de aula que, segundo FUGIMOTO (2014), amparam o uso de novos meios no ensino de matemática.

No Capítulo 2, dedica-se a informações sobre alguns softwares matemáticos, focando no estudo do FreeMat, onde são apresentados alguns dos seus comandos, necessários para aplicação nas propostas metodológicas.

No Capítulo 3, são abordados tópicos sobre funções, matrizes e sistemas lineares. Dentro das matrizes, é apresentada uma aplicação na criptografia, sendo tratados os sistemas poligráficos baseados na multiplicação de matrizes. Nos sistemas lineares, descreve-se os métodos de Cramer e de Gauss-Jordan. Os conceitos abordados, neste capítulo, são os tratados no Ensino Médio, porém, o estudo de matrizes inversas e dos sistemas lineares, aprofunda-se de modo a abranger propriedades não contempladas neste nível de ensino, ao passo que, nas propostas, são apresentados meios para serem desenvolvidos na educação básica.

No Capítulo 4, são apresentadas três propostas para aplicação do software FreeMat no estudo de funções, matrizes e sistemas lineares. A primeira proposta, aborda atividades para plotagem gráficas de funções e delimita atividades para construir uma função, e um programa computacional, que represente uma conta de água, onde seja reportado o valor cobrado para um determinado consumo em metros cúbicos (m^3). A segunda proposta, aborda o ensino de matrizes, delineando atividades que partem do estudo das operações com matrizes e aplicações na criptografia através dos sistemas poligráficos. Também, baseados nestes sistemas, propõe-se atividades para construção de programas no FreeMat, para criptografar ou decifrar uma mensagem, quando inserido uma matriz

que a corresponde. A terceira proposta, apresenta atividades para resolução de sistemas lineares através dos métodos de Cramer e de Gauss-Jordan, ainda, propõe-se a construção de programas através do FreeMat, para reportar as soluções de um sistema linear, quando inserido a matriz dos coeficientes e a matriz dos termos independentes. Dentre estas propostas, apenas a primeira foi aplicada, em sala de aula, sendo em uma turma da primeira série do Ensino Médio do Instituto Federal de Mato Grosso/Campus São Vicente.

Finalizamos este trabalho, apresentando, no Apêndice A, a sintaxe dos programas construídos na proposta, que acreditamos, poder ser inserido no ensino de matemática da primeira e segunda série do Ensino Médio.

Capítulo 1

Ensino de matemática: do giz ao uso do computador

O ensino de matemática tem necessitado da inserção de novos recursos e metodologias pedagógicas, visto que, a utilização de ferramentas tecnológicas na escola tornou-se essencial e que os alunos dominam e interagem bem com estes recursos. As práticas de ensino da matemática, nas escolas de nível básico, não apresentam grandes diferenças das utilizadas antes do avanço da informática, como exemplo, é frequente nas escolas o ensino de construções gráficas no quadro utilizando a régua enquanto os softwares computacionais de plotagem gráfica, com poderosos recursos de visualização, não são nem apresentados aos alunos.

Confirma-se a importância da utilização de softwares, no que diz respeito o ensino de funções através das construções gráficas, quando deparado com as informações do Programa Nacional do Livro Didático (PNLD) de 2015, neste, são dados relevantes elogios aos livros que aplicam tais recursos no desenvolvimento de suas atividades. Também, são apresentadas críticas quanto as dificuldades de alguns autores ao explicar o uso de softwares gráficos, aliando, de forma coesa, com os conceitos e definições matemáticas.

O uso de aplicativos computacionais, que permitem visualizar o gráfico de funções, ajuda tanto a perceber as propriedades dos seus vários tipos, quanto a fazer experimentos com maior riqueza de exemplos. Por isso, é elogiável a tendência, observada em alguns materiais didáticos destinados ao ensino médio, de empregar os referidos aplicativos como recursos para a aprendizagem da matemática [...] Além disso, softwares livres são sugeridos para auxiliar o traçado de gráficos, o que é elogiável. Contudo, nas conversões entre as formas de representação, nem sempre são levados em consideração, de modo preciso, o domínio, o contradomínio e a lei de formação das funções apresentadas (BRASIL, 2014).

Atualmente, existem diversas pesquisas sobre softwares matemáticos aplicados ao ensino, também, orientações de práticas voltadas para o ensino de matemática utilizando-os, dentre estas, no banco de trabalhos do Profmat, estão dispostas várias dissertações orientando práticas através do Geogebra, software matemático para o ensino de geometria.

Nesse capítulo, serão descritos alguns conceitos do ensino de matemática, através do uso de recursos computacionais; além disto, busca-se, orientar e facilitar a compreensão pedagógica da proposta para utilização do computador no ensino de matemática através do software FreeMat, o qual será apresentada adiante.

1.1 O uso de computadores no ensino de matemática

Com o crescimento do papel da informática na sociedade e os computadores tornando cada vez mais comuns nos lares da população brasileira, no final do século XX, passa a ser importante, e necessário, o uso de computadores e ensino de informática nas escolas. Por sua vez, depara-se com a necessidade de modificar o ensino para a inserção destas ferramentas em sala de aula.

O mundo está mudando e também deve mudar a formação que damos aos nossos educandos, a fim de que enfrentem e se desenvolvam plenamente neste mundo. Precisamos capacitá-los para que se apropriem dessas tecnologias, de forma que as façam suas e as utilizem como ferramentas de superação pessoal e de mudança social (LLANO, 2006).

Diante disto, mudanças no ensino foram esperadas: escolas com um ensino mais dinâmico e diversificado, professores que dominassem novas metodologias e trouxessem

para o aluno novos caminhos visando alcançar o entusiasmo do discente e a motivação pela busca do saber, programas computacionais que facilitassem o entendimento do aluno, enfim, um anseio pela mudança do ensino, uma mudança que fosse percebida e vivenciada nas escolas. Porém, ainda não vivencia-se isto, muitos docentes discriminam o uso da tecnologia por não dominarem ou conhecerem meios para aliar conteúdos trabalhados em sala de aula com os recursos tecnológicos. Assim, mesmo com a mudança da sociedade na utilização de recursos tecnológicos, percebe-se um lento avanço das escolas neste aspecto.

Recordam-se casos de centros educacionais onde se inseriram os computadores e que, depois da emoção inicial, tudo continuou do mesmo jeito. Quando acreditava-se que os aparelhos teriam em si o poder de gerar esta mudança, percebeu-se que tais equipamentos nos enganaram, não cumpriram a promessa de “modernizar” e “melhorar” a qualidade de nossa educação. O erro está em pensar que eles são “varinhas mágicas”, que a sua presença basta para transformar a realidade de uma instituição educativa (LLANO, 2006).

Desta forma, exigências para com os docentes, afim de que, lecionem disciplinas do núcleo comum nas escolas de ensino básico, aliando suas aulas às tecnologias da informação, tem tornado-se cada vez mais frequentes. Porém, observa-se o uso destas tecnologias de forma limitada, sem a exploração de mecanismos facilitadores do ensino aprendizagem. Atividades e exercícios, que poderiam ser mais aprofundados com a utilização destes recursos, interligando-os às disciplinas, são apenas trabalhadas de forma mecânica e repetitiva, ofuscando sua aplicação, enquanto recurso para um ensino dinâmico, e trazendo um ensino tradicional com uma roupagem diferente.

Muitas vezes, os professores, premidos pelas exigências oficiais e institucionais, levam seus alunos a laboratórios de informática sem planejar atividades que desenvolvam competências, tais como a de formular hipóteses ou de argumentar. Aprendem a acessar um determinado software e estabelecem tarefas em que os estudantes vão apenas digitar comandos e visualizar as respostas (CURY, 2004).

Quanto ao ensino de matemática na educação básica, este, pode-se tornar mais atrativo quando empregado o uso do computador e, a inserção deste recurso, proporciona ao discente, além do aprendizado matemático, a aquisição de novas habilidades sobre uso de ferramentas computacionais. Quando bem orientado, os discentes podem aplicar

estas ferramentas na solução de problemas do seu cotidiano, bem como, apresentarem mais facilidade quanto à aquisição e fixação de conteúdos matemáticos que lhe são apresentados.

Cabe à matemática do Ensino Médio apresentar ao aluno o conhecimento de novas informações e instrumentos necessários para que seja possível a ele continuar aprendendo. Saber aprender é a condição básica para prosseguir aperfeiçoando-se ao longo da vida. Sem dúvida, cabe a todas as áreas do ensino médio auxiliar no desenvolvimento da autonomia e da capacidade de pesquisa, para que cada aluno possa confiar em seu próprio conhecimento (BRASIL, 2000).

Ao aplicar o uso de computadores no ensino, além de um conhecimento prévio para lidar com alguns softwares, exige-se dos docentes, abordagens nas aulas que norteiem o discente na construção do conhecimento, utilizando esta ferramenta não apenas para conseguir resultados, mas, um elo entre discussões, interpretações, construções e os tópicos estudados. Porém, quando aplicados, atividades mais elaboradas e com tarefas que sobreponha atividades mecânicas são pouco evidenciadas.

Não é necessário que o professor seja um especialista em tecnologia, mas é interessante que ele a conheça e domine, de modo a utilizá-lo numa perspectiva crítica. Para fazer uso adequado dos recursos tecnológicos e para facilitar o desenvolvimento das seqüências didáticas, é importante que o professor conheça o modo de operação técnica (comandos, funções e linguagens) de forma a explorar suas possibilidades e identificar as limitações [...] o uso da tecnologia, e, em particular, do computador na educação, pode ser considerado uma inovação e, como toda inovação, ela só será integrada à prática profissional após um processo longo de apropriação e de utilização frequente em situações diversificadas. O processo de integração envolve aspectos tecnológicos, pedagógicos e, também, de gestão. Além de dominar tecnologia e de conhecer os softwares disponíveis, é necessário incorporá-los ao currículo (BELINE, 2010).

Por mais que existam uma gama de softwares educacionais com grandes potencialidades voltadas para o ensino de matemática, o despreparo ou desconhecimento dos docentes de como aliar suas aulas com estes softwares, são alguns dos fatores que acarretam na sua utilização de forma mecânica, semelhante ao utilizar uma calculadora para

obter resultados ou de resolver uma equação no caderno, deixando as perspectivas de uso dessas ferramentas em sala de lado.

Quanto aos recursos didáticos, o uso de calculadora e de outros recursos tecnológicos ainda é um terreno insuficientemente explorado no ensino médio atual. Por exemplo, nas obras didáticas, é sempre presente o emprego da calculadora, porém visa-se mais à realização e conferência de cálculos em detrimento de outras possibilidades de trabalho didático com essa ferramenta tecnológica. Entre os outros recursos tecnológicos, de forma geral, há boas sugestões de utilização de softwares livres. (BRASIL, 2014).

Logo, atividades orientadas e fundamentadas para utilização do computador são necessárias, assim como, o conhecimento pelo docente de recursos computacionais que sirvam como ponte para desmistificar o ensino de matemática nas escolas públicas e torná-lo mais atrativo, podendo ser através da utilização de softwares em atividades aplicadas que busquem desenvolver novos olhares sobre a matemática e suas aplicações.

Os professores necessitam, portanto, conhecer as tecnologias disponíveis e estudar possibilidades de uso dessa ferramenta como mais um recurso didático para o processo de aprendizagem. É importante ressaltar que não se trata de tornar a aprendizagem mais fácil aligeirando o ensino. Ao contrário, a aprendizagem deve ser favorecida com situações que a tornem mais significativa e que os alunos possam interagir entre si e com a máquina, construindo conhecimentos, vivenciando situações que, muitas vezes, não tinham sentido, ou tinham outro sentido, no ambiente papel e lápis (BELINE, 2010).

Ressalta-se que, professores de matemática ainda insistem em práticas docentes norteadas pela ementa, onde, práticas baseadas na realização de projetos podem atrasar o cumprimento da mesma. Desta forma, a incorporação de um ensino desmistificado através da utilização do computador, necessita de tempo e práticas diferenciadas, exigindo do professor flexibilidade quanto a matriz curricular, visto que, quando se trabalha com um software específico, o professor terá o desafio de ensinar o aluno a trabalhar com o programa, bem como, vincular atividades que tenha por finalidade o ensino ou fixação de certo conteúdo matemático.

Vale insistir que a atualização curricular não deve significar complementação de ementas, ao se acrescentarem tópicos a uma lista de assuntos. Ao contrário, é preciso superar a visão enciclopédica do currículo, que é um obstáculo à verdadeira atualização do ensino, porque estabelece uma ordem tão artificial quanto arbitrária, em que pré-requisitos fechados proíbem o aprendizado de aspectos modernos antes de se completar o aprendizado clássico e em que os aspectos “aplicados” ou tecnológicos só teriam lugar após a ciência pura ter sido extensivamente dominada. Tal visão dificulta tanto a organização dos conteúdos escolares quanto a formação dos professores (BRASIL, 2000).

Para a implementação efetiva das tecnologias da informação em sala, será preciso, além de um olhar especial para o currículo escolar, também, atualizações e novas metodologias de ensino. E, ao inserir o uso do computador em sala, há necessidade de desenvolver trabalhos orientados aos alunos em relação ao uso correto dessas tecnologias e suas aplicações junto aos conteúdos abordados, buscando interligar os conteúdos matemáticos presentes nos planejamentos com os softwares presentes no universo da internet.

Esse impacto da tecnologia, cujo instrumento mais relevante é hoje o computador, exigirá do ensino de matemática um redirecionamento sob uma perspectiva curricular que favoreça o desenvolvimento de habilidades e procedimentos com os quais o indivíduo possa se reconhecer e se orientar nesse mundo do conhecimento em constante movimento (BRASIL, 2000).

Quanto as práticas orientadas para o uso e ensino desses recursos, de modo a facilitar e orientar a utilização de meios tecnológicos no ensino de matemática, a execução de projetos dentro da concepção construtivista é um meio para alcançar e inserir tais práticas nas escolas, além de conceber espaços para atividades em grupos, podendo ser atividades dinâmicas ou interdisciplinares inseridas no meio tecnológico.

O desenvolvimento de projetos, conduzidos por grupos de alunos com a supervisão de professores, pode dar oportunidade de utilização dessas e de outras tecnologias, especialmente no Ensino Médio. Isso, é claro, não ocorre espontaneamente, mas sim como uma das iniciativas integrantes do projeto pedagógico de cada unidade escolar, projeto que pode mesmo ser estimulado pelas redes educacionais. Para a elaboração de tal projeto, pode-se conceber, com vantagem, uma nucleação prévia de disciplinas de uma área, como a matemática e ciências da natureza, articulando-se em seguida com as demais áreas (BRASIL, 2000).

Neste contexto, atividades voltadas para a introdução do computador em salas de aula são um anseio e, estas práticas, tornam o ensino mais atrativo e desafiador, dentre vários recursos do computador, cabe aos docentes atualizarem e dominarem a utilização de softwares voltados para a educação.

Quanto a quantidade de softwares voltados para o ensino, a matemática é mais beneficiada, sendo encontrados softwares para esboço gráfico, resolução de equações, estudo do cálculo e diversos programas para atividades lúdicas.

Cabe ressaltar que a matemática é uma área privilegiada tanto pela quantidade de materiais (softwares) existentes quanto de pesquisas relacionadas ao uso dos mesmos materiais. Entretanto, poucas pesquisas tratam do uso que o professor faz de um software em sala e, também, poucos são os textos destinados a difundir as pesquisas para os professores (BELINE, 2010).

Desta forma, a quantidade de softwares voltados para matemática e seu ensino, bem como, a falta de conhecimento e despreparo, por parte de uma parcela dos docentes do ensino básico, de como lidar com estes programas nas aulas de matemática ou no uso do computador em sala, reforça a importância de orientações voltadas para docentes quanto a utilização de softwares e a difusão destas práticas no ensino básico, visto a importância e crescente inserção das tecnologias da informação nas escolas.

Capítulo 2

FreeMat: dentre vários softwares pagos, uma escolha livre

Neste capítulo, serão apresentados alguns softwares aplicados na matemática, utilizados para o estudo de áreas como: álgebra, geometria, equações diferenciais, dentre outras; em especial, destes softwares, será dado ênfase ao *FreeMat*, sendo apresentado um breve histórico e comandos básicos para sua utilização no que tange, funções e matrizes. Os comandos apresentados também servirão de base para entendimento e aplicação das propostas, que serão descritas no Capítulo 4.

Diante de uma grande quantidade de softwares aplicados à matemática e ao ensino de matemática, destacam-se alguns pesquisados e comentados em artigos científicos ou trabalhos acadêmicos, tais como SANTOS (2010) e BATISTA (2004), destes, devido alguns softwares serem pagos e dificilmente utilizados pelas escolas públicas, existem alternativas similares e livres no mercado para o uso destas instituições ou pelos docentes das mesmas.

Na Tabela 2.1, são listados alguns softwares matemáticos gratuitos, incluindo os conteúdos abordados pelos mesmos e os sistemas operacionais compatíveis. Além dos softwares citados na Tabela 2.1, existem outros de natureza comercial, tais como: Cabri Géomètre, Derive, MS-Excel, Geospace, Geoplan, Mathematica, Maple, Matlab, entre outros.

Software	Aplicações	Sistema operacional
Geogebra	Geometria, álgebra e cálculo	Windows, Linux, MacOS X
Calc 3D	Álgebra linear, geometria plana e espacial	Windows e MacOS X
FreeMat	Gráficos de relações, cálculo numérico, álgebra	Windows, Linux e MacOS X
kmplot	Gráficos de relações	Linux e MacOS X
Kseg	Geometria euclidiana	Windows, Linux e MacOS X
Maxima	Cálculo, equações diferenciais e álgebra linear	Windows, Linux e MacOS X
WinGeo	Geometria espacial	Windows e MacOS X
Winplot	Geometria espacial e plana, gráficos de relações	Windows e MacOS X
Régua e Compasso	Geometria plana	Windows, Linux e MacOS X
Octave	Cálculo numérico	Windows, Linux e MacOS X
Scilab	Álgebra, cálculo numérico, gráficos de relações	Windows, Linux e MacOS X

Tabela 2.1: Softwares matemáticos livres

Devido a proximidade com objeto de estudo que será apresentado nos próximos capítulos, destaca-se o *Matlab*, uma ferramenta matemática aplicada no cálculo numérico, sendo referência entre os softwares matemáticos desta área. Este é um software pago segundo uma licença comercial para Windows, Linux e outras plataformas, e é produzido pela empresa Mathworks. Com o Matlab, é possível esboçar gráficos de funções, resolver equações diferenciais ordinárias, derivar e integrar funções.

O Matlab (acrônimo de MATrix LABoratory) é um software computacional conhecido mundialmente como uma excelente ferramenta para soluções de problemas matemáticos, científicos e tecnológicos, que possui comandos muito próximos da forma como escrevemos as expressões matemáticas, podem ser usado como prancheta de rascunhos para avaliar expressões digitadas nas linhas de comando, ou para executar programas grandes previamente escritos, pois ele possui um ambiente de desenvolvimento integrado embutido, um depurador de informações (NOBRE, 2014).

Ainda,

Convém observar que esse sistema não é desenhado especialmente para atingir objetivos pedagógicos, mas é projetado para atender às necessidades do profissional na resolução de problemas. É certo que a utilização adequada desse sistema pode contribuir muito para cientistas, professores e alunos a nível de graduação ou profissional (NOBRE, 2014).

Similar ao Matlab, existem alguns softwares matemáticos distribuídos gratuitamente, destes, diante dos citados na Tabela 2.1, cabe destacar o *FreeMat*, um software

que embora não tenha todas as suas potencialidades, tem grandes recursos e aproxima-se muito do mesmo.

2.1 FreeMat

O *FreeMat* é um pacote computacional numérico desenvolvido pela Samitbasu, tem síntese e 95% da compatibilidade dos recursos iguais ao do Matlab, tendo como plataformas suportadas: Windows, MAC OS X e Linux. Por ser um software livre disponível sob a Licença Pública Geral (GPL), é possível baixá-lo gratuitamente através da seguinte página da internet, <http://freemat.sourceforge.net>, sendo, a versão 4.2, a atual até a data deste trabalho. Informações gerais sobre o software e seus desenvolvedores são encontradas em FREEMAT (2014).

O FreeMat é um programa de cálculo numérico para Matemática e ciências afins, que funciona interativamente, baseado em matrizes, permitindo a visualização gráfica de vários dos seus objetos. O FreeMat é rápido e simples, resolvendo problemas numéricos em tempo menor que o exigido em uma linguagem de programação como C ou Fortran (SODRE, 2014).

Com o FreeMat é possível plotar gráficos 2D e 3D, realizar diversas operações com matrizes e polinômios, trabalhar com funções, podendo realizar um tratamento mais aprofundado como derivar, integrar funções e resolução de equações diferenciais. Em SHARMA (2014), é dado maior ênfase e apresentado comandos para estes tratamentos gráficos e algébricos no software.

Pela sua grande quantidade de recursos e similaridade com o Matlab, o FreeMat pode ser aplicado tanto no ensino básico quanto no ensino superior. Observa-se que, para ser aplicado no ensino básico, foco de estudo deste trabalho, muitos de seus comandos são simples e de fácil programação, podendo ser trabalhado sem um estudo minucioso de seu tutorial, bem como, é gratuito, leve e tem suporte para o sistema Linux, utilizado pelas escolas públicas.

Por ser muito fácil de usar, a linguagem é ideal para uso educacional e para desenvolver rapidamente protótipos de novos programas [...] sistema computacional flexível capaz desenvolver essencialmente qualquer problema técnico (NOBRE, 2014).

Ao iniciar o FreeMat, o usuário depara-se com a Interface do Programa (Figura 2.1), tal Interface está em língua inglesa e nela está disposta a janela de comandos, uma tela branca onde devem ser digitados comandos para realização de ações matemáticas. Tais comandos, são digitados no *prompt*, isto é, inseridos na janela de comandos após o seguinte sinal: “-- >”; em seguida, pressiona-se a tecla **Enter**, para que sejam reportadas as respostas, que serão apresentadas na própria janela precedidas pelo anagrama “ans” (*answer*, que em português significa *resposta*).

Observação 2.1. Durante a apresentação das propostas utilizando o FreeMat, ao serem descritos a inserção de n comandos referente à execução de uma ação matemática, estes, serão apresentados dentro de um retângulo, como segue

comando 1
comando 2
⋮
comando n

subentendendo-se que, após cada comando inserido, deve-se clicar na tecla **Enter**.

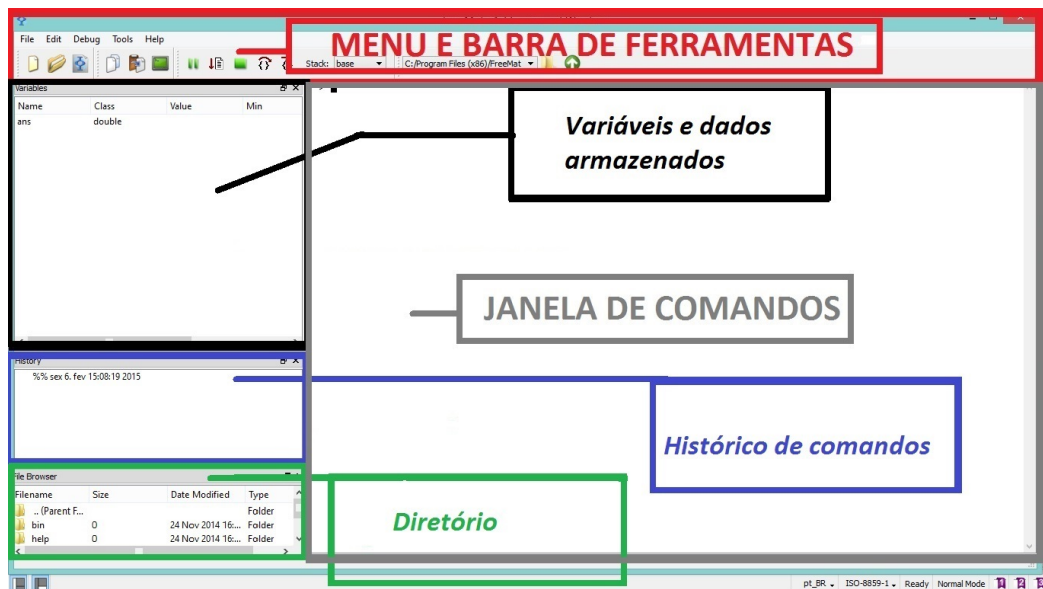


Figura 2.1: Janelas dispostas na Interface do FreeMat

Na parte superior da Interface, encontram-se as Barra de Menu e Ferramentas, no canto esquerdo da Interface, são dispostas três janelas: History, File Browser e Variables que, respectivamente, corresponde ao armazenamento do histórico dos últimos comandos

digitados, diretório corrente do programa e o espaço de trabalho onde se visualizam dados e variáveis, como mostra a Figura 2.1.

Ao plotar um gráfico de uma relação matemática, este, é apresentado em uma nova janela aberta automaticamente pelo software. Na parte superior desta janela gráfica, encontra-se o Menu de Ferramentas, e nele são dispostos sequencialmente as seguintes ferramentas: *save* (*salvar gráfico*), *close* (*fechar gráfico*), *copy* (*copiar gráfico*), *zoom* (*aumentar gráfico*), *rotate* (*rotacionar gráfico*), *camera rotate* (*rotacionar câmera*) e *sample* (*mostrar pontos do gráfico*), como observa-se na Figura 2.2.

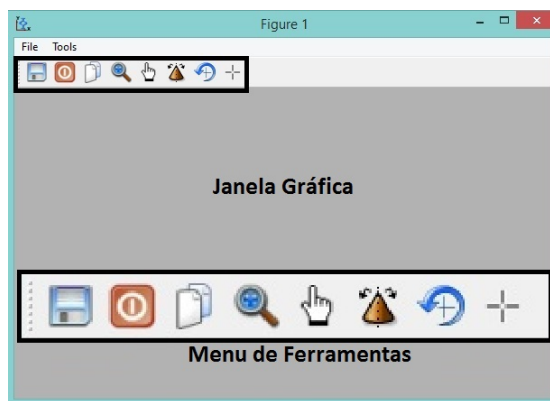


Figura 2.2: Janela gráfica do FreeMat

Segundo SODRE (2014), a realização de operações básicas no FreeMat assemelham-se com as de uma calculadora científica, ou seja, operações na forma usual. Na Tabela 2.2, são apresentados os comandos utilizados para estas operações entre números reais e, também, com matrizes. Exemplos destas operações básicas realizadas no FreeMat, com números reais, podem ser vistos na Figura 2.3 e, no caso das operações com matrizes, exemplos detalhados foram construídos no Capítulo 4 deste trabalho.

Operação	Operador aritmético	Números reais	Matrizes
Adição	+	$2+3$	$A+B$
Subtração	-	$2-3$	$A-B$
Multiplicação	*	$2*3$	$A*B$
Divisão	/	$2/3$	-
Potenciação	^	2^3	A^3

Tabela 2.2: Comandos utilizados nas operações com números reais e matrizes

Na resolução de expressões que envolvam chaves, colchetes e parênteses; como o FreeMat só possibilita utilizar parênteses, este ordenará as operações. SODRE (2014)

explica que, “Para realizar cálculos básicos no FreeMat, basta inserir as operações nas formas usuais, lembrando que parênteses alteram a precedência nas operações realizando primeiro as operações envolvidas por eles”.

Exemplo 2.2. Para determinar o valor da expressão

$$\frac{[(2 + 3) \cdot 2]^2 - 1}{3},$$

no FreeMat, o comando correspondente seria:

$$(((2+3)*2)^2-1)/3$$

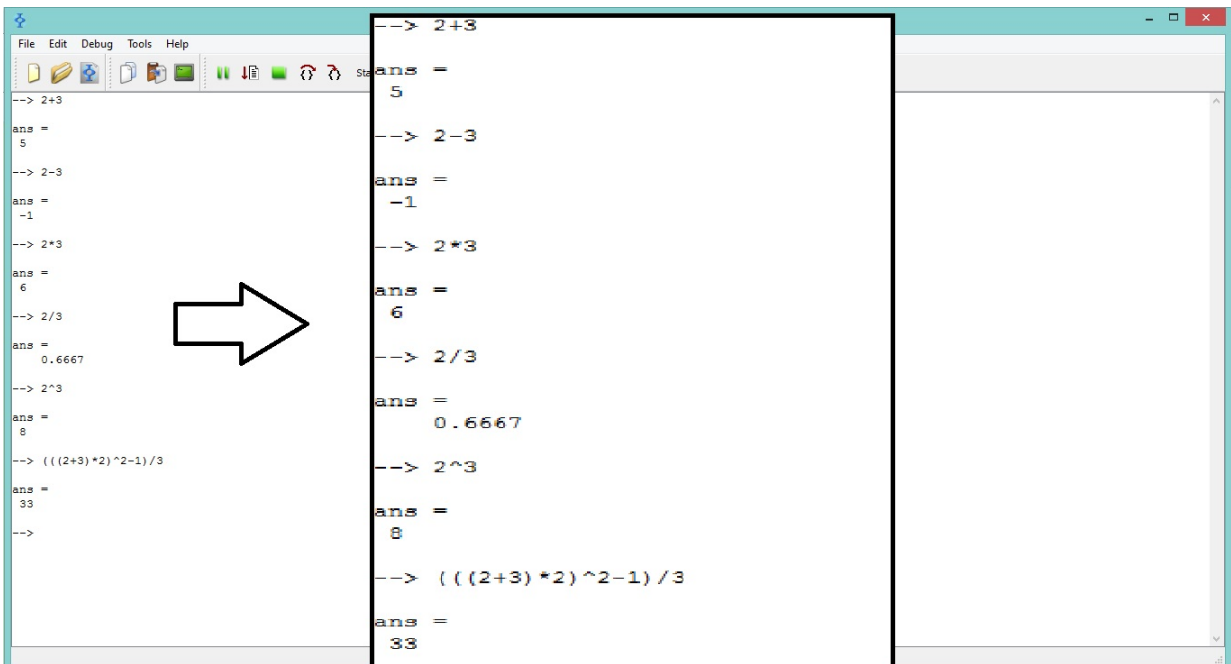


Figura 2.3: Operações com números reais desenvolvidas no FreeMat

Quanto as matrizes, estas constituem a estrutura básica do FreeMat, por isso, ele permite várias maneiras de manipulá-las. Para inserir uma matriz no FreeMat, entre colchetes, os elementos de uma mesma linha devem estar separados por um espaço e as linhas devem estar separadas por ponto e vírgula.

Exemplo 2.3. Ao trabalhar com matrizes no FreeMat, para inserir a matriz quadrada

$$B = \begin{bmatrix} 1 & 4 \\ 2 & 1 \end{bmatrix},$$

deve-se digitar o seguinte comando:

$$B = \begin{bmatrix} 1 & 4 \\ 2 & 1 \end{bmatrix}.$$

Relativo ao suporte e acesso do tutorial do FreeMat, na janela de comandos, digita-se `helpwin`; isto ajudará quanto dúvidas e desenvolvimento de operações mais complexas, bem como, fornecerá todas as ferramentas que o FreeMat dispõe.

Para um estudo introdutório, a Tabela 2.3, apresenta algumas funções e comandos básicos utilizadas no FreeMat para manipulação de gráficos, matrizes e funções. Os comandos citados na Tabela 2.3, são referentes aos apresentados em FIPIAK (2012), SHARMA (2014) e SODRE (2014).

Funções	Descrição
<code>helpwin</code>	Abre um tutorial do programa em língua inglesa
<code>clear</code>	Limpa as variáveis armazenadas
<code>clc</code>	Limpa a janela de comandos
<code>plot(x,y)</code>	Plota um gráfico 2D para os pares ordenados (x, y)
<code>det(A)</code>	Calcula o determinante da matriz A
<code>format short</code>	Configura o FreeMat para reportar 4 casas decimais após a vírgula
<code>format long</code>	Configura o FreeMat para reportar 16 casas decimais após a vírgula
<code>inv(A)</code>	Determina a matriz inversa da matriz A
<code>rank(A)</code>	Determina o posto da matriz A
<code>A'</code>	Determina a matriz transposta da matriz A
<code>eye(n)</code>	Cria a matriz identidade de ordem n
<code>zeros(m,n)</code>	Cria a matriz nula de ordem $m \times n$
<code>size[A]</code>	Informa a quantidade de linhas e colunas da matriz A
<code>C=[A B]</code>	Determina a matriz em blocos $C = [A B]$
<code>;</code>	Quando inserido após um comando, não reportará os resultados
<code>mod(a,n)</code>	Determina o inteiro não negativo, menor do que n , $b = \text{mod}(a, n)$ congruente à a módulo n
<code>mod(A,n)</code>	Determina a matriz $B = \text{mod}(A, n)$ cujas entradas são congruentes, respectivamente, as entradas da matriz A módulo n
<code>disp</code>	Utilizado para apresentar mensagens de textos no programa
<code>x=input</code>	Entrada de dados para uma variável x via valor digitado

Tabela 2.3: Algumas funções e comandos utilizados no FreeMat

2.2 Programação em FreeMat

O FreeMat é uma ferramenta e uma linguagem de programação, por possuir uma biblioteca de funções predefinidas, torna as tarefas de programação técnica mais fáceis.

Os comandos do FreeMat são normalmente digitados na Janela de Comandos, onde uma única linha de comando é introduzida e processada imediatamente, porém, o FreeMat é capaz de executar sequências de comandos armazenadas em arquivos com extensão `.m`, para isto, deve-se acessar o Menu de Ferramentas e selecionar a opção *editor*. Após selecionado, abrirá uma nova janela para edição de textos, tal janela, apresentada na Figura 2.4, é específica para programação e os comandos deverão ser digitados na mesma. Portanto, por meio do editor de texto, é possível construir um programa em FreeMat através dos mesmos comandos e sintaxe dos utilizados na janela de comandos.

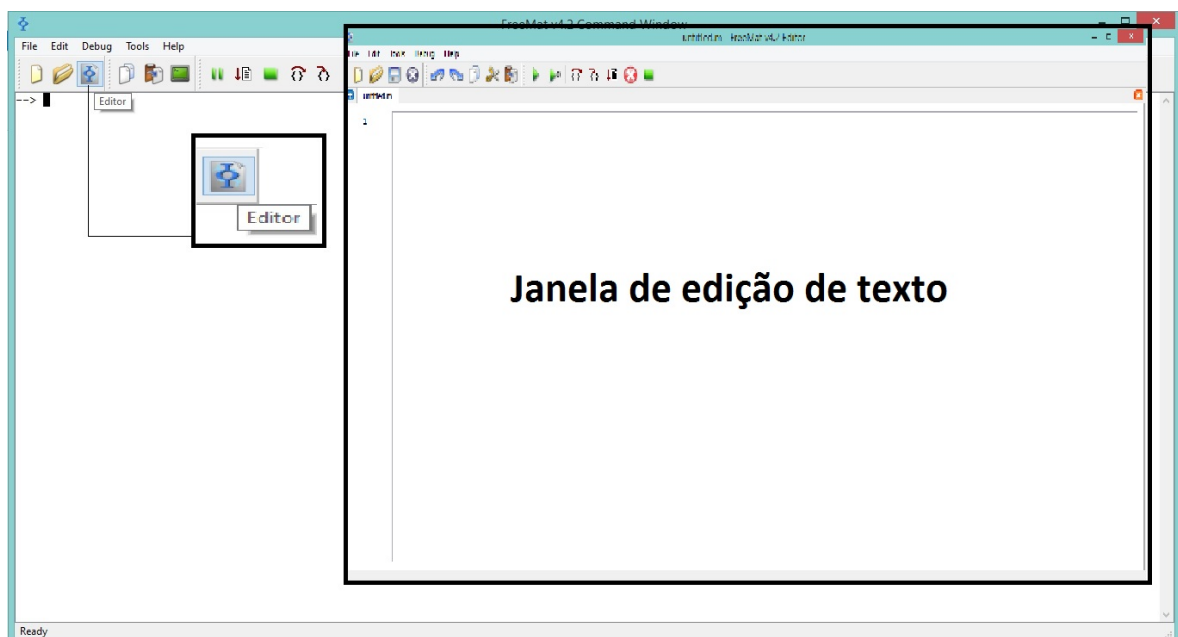


Figura 2.4: Janela de edição de texto

Para programação em FreeMat, existem dois tipos de arquivos: *script* e *function*. Os arquivos *script* são úteis quando se deseja efetuar uma sequência de comandos com muita frequência, pois, toda vez que for chamado, efetua a lista dos comandos como se eles fossem inseridos sequencialmente, via teclado. Os arquivos tipo *function*, admitem parâmetros de entrada, retornam valores e possuem variáveis locais. O último arquivo será o que utilizaremos, no Capítulo 4.

Para salvar um arquivo digitado no editor do FreeMat, na barra do Menu do

editor, seleciona-se a opção *File* e clica-se em *Save* ou *Save As*, como mostra a Figura 2.5. Após, deve-se escolher um diretório para salvar o arquivo, sugere-se escolher preferencialmente o diretório onde esteja instalado o FreeMat.

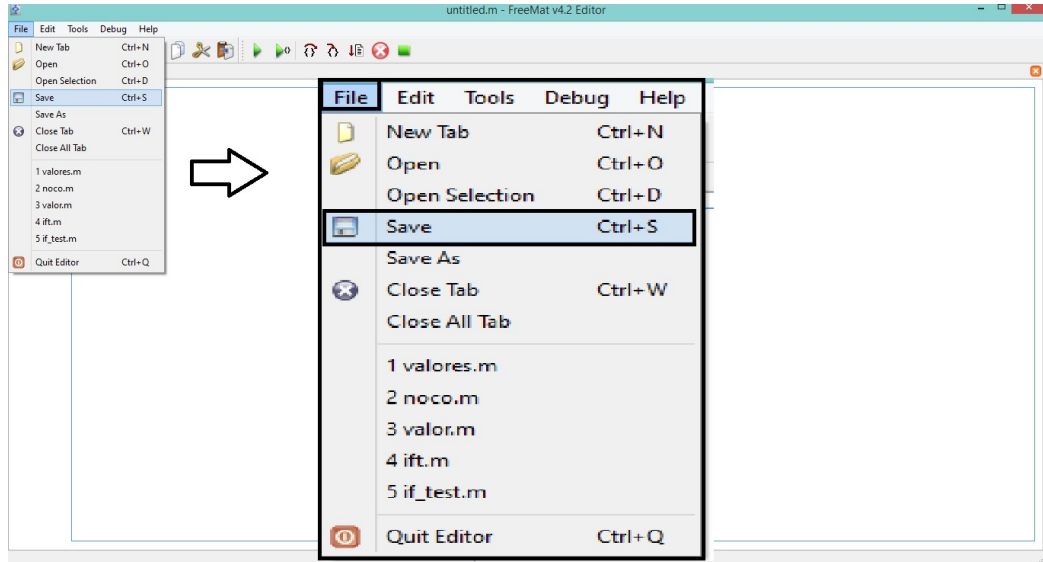


Figura 2.5: Local para salvar arquivos digitados na janela de edição

Em particular, para salvar um arquivo *script*, deve-se escolher um nome para o arquivo e salvá-lo na extensão *.m*. Para utilizá-lo, selecione o diretório onde foi salvo, por exemplo, escolher o diretório *Program Files/FreeMat*, como mostra a Figura 2.6, e digita-se na janela de comandos o nome com que foi salvo, assim, todas as sequências de comandos serão executadas.

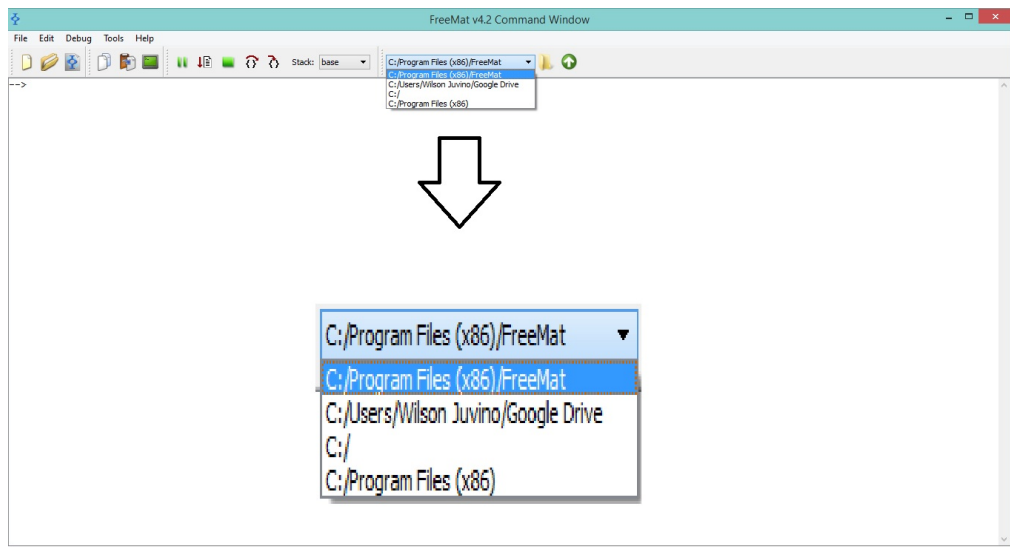


Figura 2.6: Diretórios do FreeMat

Já, para salvar programas em arquivos tipo *function*, o procedimento é similar e explicaremos com mais detalhes, a continuação: Na primeira linha do editor deve-se escrever:

```
function nome do arquivo
```

e o arquivo de ser salvo com o nome descrito. Por exemplo, ao digitar na primeira linha

```
function resultado
```

para salvar este arquivo, deve-se escolher o diretório corrente onde está instalado o FreeMat (como foi observado na Figura 2.6), salvando-o com o nome: `resultado.m`. Para executar a função, basta digitar na janela de comandos a palavra: `resultado`.

Quanto aos operadores utilizados na programação, a Tabela 2.4 apresenta alguns *operadores lógicos* e, a Tabela 2.5, *operados relacionais*.

Descrição	operador lógico
e	&
ou	

Tabela 2.4: Operadores lógicos

Descrição	operador relacional
igual a	==
maior do que	>
maior ou igual a	>=
menor do que	<
menor ou igual a	<=
diferente a	~=

Tabela 2.5: Operadores relacionais

2.3 Comandos de repetição e de condição

Na programação, existem comandos específicos que controlam o fluxo e especificam a ordem. Tais comandos são chamados como *comandos de condição* e *comandos de repetição*, os quais, no FreeMat, são os mesmos utilizados no Matlab e funcionam de forma semelhante aos usados na linguagem *C*, porém, com uma estrutura diferente.

2.3.1 Comandos de condição

Relativo aos comandos de condição, neste trabalho, serão abordados os laços `if`, `else` e `elseif` (*se, senão, senão se*). O comando `if`, avalia uma expressão condicional¹ e, executa uma sequência de comandos, se esta condição é verdadeira. A forma geral deste comando é a seguinte:

```
if expressão-condicional
sequência de comandos
end
```

Exemplo 2.4. Inserido a sequência de comandos

```
if x>10
y=10
end
```

tem-se o seguinte: se a condição $x > 10$ for verdadeira, então $y = 10$.

O comando `elseif`, avalia uma expressão condicional considerando a possibilidade de ela ser falsa. Se o comando `if` diz o que fazer quando a condição é verdadeira, o comando `elseif` permite executar uma sequência de comandos se a condição `if` é falsa. A forma geral deste comando é a seguinte:

```
if expressão-condicional
sequência de comandos
elseif expressão-condicional
sequência de comandos
end
```

¹Entende-se por expressão condicional, qualquer expressão que resulte em uma resposta do tipo verdadeiro ou falso e construída utilizando operadores matemáticos, relacionais ou lógicos.

Exemplo 2.5. Inserido a sequência de comandos:

```
if x>10
y=10
elseif x<=10
y=-10
end
```

tem-se o seguinte: se a condição $x > 10$ for falsa, então $y = -10$.

O comando **else**, é utilizado após as expressões condicionais definidas por **if** e **elseif**. O comando **else** executa uma sequência de comandos, quando as condições anteriores são falsas. A forma geral é a seguinte:

```
if expressão-condicional
sequência de comandos
elseif expressão-condicional
sequência de comandos
else expressão-condicional
sequência de comandos
end
```

Exemplo 2.6. Inserido a sequência de comandos

```
if x>10
y=10
elseif x<10
y=-10
else x=10
y=0
end
```

tem-se o seguinte: se a condição $x > 10$ e $x < 10$ forem falsas, então $y = 0$.

2.3.2 Comando de repetição

Relativo aos comandos de repetição, neste trabalho, serão abordados os laços `for` e `while`.

O comando `while`, avalia uma expressão condicional e executa uma sequência de comandos enquanto essa condição for verdadeira. Ou seja, ao final da sequência de comandos, o comando `while` testa a expressão condicional para saber se deve ou não executar novamente aquela sequência de comandos. A forma geral de um comando `while` é:

```
while (expressão condicional)
sequência de comandos
end
```

O comando `for` é um controlador de fluxo que serve para repetir uma sequência de comandos diversas vezes. A forma geral de um comando `for` é:

```
for variavel = valor inicial: valor final
sequência de comandos
end
```

Com estas informações é possível utilizar tal software para aplicação superficial em conteúdos do ensino básico, observando que são inúmeras as funções que o programa dispõe. Para uma aplicação e estudo mais aprofundado, é necessário, além dos conhecimentos destes comandos básicos, também, domínio da parte algébrica e geométrica do conteúdo a ser desenvolvido.

Capítulo 3

Tópicos de Matemática a serem aplicados no FreeMat

Neste capítulo, serão apresentados definições e propriedades elementares das funções, matrizes, determinantes e sistemas lineares, na qual, juntamente com os exemplos abordados no capítulo, serão utilizadas na proposta para utilização do FreeMat no ensino de matemática. As propriedades, definições e teoremas matemáticos apresentadas, possuem como referências: BOLDRINI (1980), CALLIOLI (2000), HEFEZ (2012), HOWARD (2001), IEZZI (2002), LIMA (2013) e SILVA (2007).

De agora em diante, utilizaremos as notações \mathbb{N} , \mathbb{Z} , \mathbb{R} e \mathbb{Z}_n ($n \geq 2$), para indicar o conjunto dos números naturais, inteiros, reais e dos inteiros módulo n , respectivamente.

3.1 Função

Data-se o primeiro aparecimento da representação de uma função há 2000 a.c, de forma intuitiva, segundo MARIANI (2015), foi encontrado nas tábuas babilônicas em seus cálculos com tabelas sexagesimais de quadrados e de raízes quadradas. Quanto a origem do conceito de função, atribui-se ao matemático Nicole Oresme (1323 - 1382), que descreveu graficamente a dependência entre velocidade e tempo usando linhas verticais e horizontais. Com o avanço da matemática, conseqüentemente, houve contribuições para o desenvolvimento das funções, até que se chegasse nos conceitos utilizados atualmente.

Definição 3.1. Sejam dois conjuntos não vazios, A e B , e $f \subset A \times B$. A terna (A, B, f) é uma *função de A em B* se, para todo $x \in A$, existe um único $y \in B$ tal que $(x, y) \in f$.

O conjunto A é denominado *domínio da função* e o conjunto B é denominado *contradomínio da função*. O conjunto dos elementos $y \in B$, para os quais existe $x \in A$, tal que $(x, y) \in f$, é chamado de conjunto imagem e será denotado por $\text{Im}(f)$.

A terna (A, B, f) é usualmente denotada por $f : A \longrightarrow B$ e lê-se: função f de A em B . Neste contexto, se $(x, y) \in f$, denomina-se y como imagem de x pela aplicação de f e escreve-se $f(x) = y$, onde, x é chamado de variável independente e y é chamado de variável dependente. Mais precisamente, representa-se

$$\begin{aligned} f : A &\longrightarrow B \\ x &\longmapsto f(x) \end{aligned}$$

como sendo a função f de A em B tal que a cada $x \in A$, faz corresponder um único elemento $f(x) \in B$.

O *gráfico* de f é o conjunto

$$\text{graf } f = \{(x, y) \in A \times B : y = f(x)\}.$$

Se A e B são subconjuntos de \mathbb{R} , a função $f : A \longrightarrow B$ chama-se *função real de variável real* e, o seu gráfico é representado no plano cartesiano.

Exemplo 3.2. A função real de variável real $f : \mathbb{R} \setminus \{0\} \longrightarrow \mathbb{R}$, definida por $f(x) = \frac{1}{|x|}$, possui por imagem o conjunto $(0, +\infty)$.

Exemplo 3.3. Para alugar um carro certa pessoa dispõe de duas Locadoras, A e B . A locadora de automóveis A , cobra uma taxa de 20 reais mais 2 reais por quilômetro rodado, após a retirada do carro da garagem. Já, a locadora B , cobra uma taxa de 50 reais mais 1 real por quilômetro rodado. Nota-se que, a locadora A tem uma taxa menor que a sua concorrente, diante disso, existirá um momento no qual a locadora B torna-se viável?

A princípio parece ser que a locadora A é favorável. No entanto, veremos que isto não é verdade, a partir de uma certa quilometragem. Para ambas locadoras os valores a serem pagos y estão em função dos quilômetros rodados x , sendo os quilômetros números reais não negativos. Desta forma, se a locadora A cobra R\$ 2,00 por quilômetro x mais uma taxa fixa de R\$ 20,00, então o valor a pagar y é dado por $y = 2x + 20$. De maneira análoga, a locadora B cobra R\$ 1,00 por quilômetro x mais uma taxa fixa de R\$ 50,00,

então o valor a pagar y é dado por $y = x + 50$. Assim, define-se uma função para cada locadora, de modo que, $f : [0, +\infty) \rightarrow \mathbb{R}$ dada por $f(x) = 2x + 20$ é a função da locadora A e $g : [0, +\infty) \rightarrow \mathbb{R}$ dada por $g(x) = x + 50$ é a função da locadora B .

Afim de determinar para quais valores do domínio, as imagens da função f são maiores que as imagens da função g , devemos achar os valores $x \in [0, +\infty)$ tais que $f(x) > g(x)$. De fato, se $x \in [0, +\infty)$, temos que

$$f(x) > g(x) \Leftrightarrow 2x + 20 > x + 50 \Leftrightarrow x > 30.$$

Concluimos que, $f(x) > g(x)$ se, e somente se, $x > 30$. Isto é, a locadora B é mais atrativa que a locadora A após percorridos 30 km.

Definição 3.4. Sejam n um número inteiro não negativo e a_0, \dots, a_n , números reais. A função $f : \mathbb{R} \rightarrow \mathbb{R}$, definida por

$$f(x) = \sum_{i=0}^n a_i x^i = a_n x^n + \dots + a_0,$$

denomina-se *função polinomial*.

Se $a_n \neq 0$, n é chamado *grau* da função polinomial f . Em particular, temos que:

- (1) Se $n = 0$, a função polinomial f é a função constante. Além disso, se $a_0 \neq 0$, f possui grau 0. A função nula não possui grau.
- (2) Se $n = 1$ e $a_1 \neq 0$, a função polinomial f denomina-se *função afim*.
- (3) Se $n = 2$ e $a_2 \neq 0$, a função polinomial f denomina-se *função quadrática*.
- (4) se $n = 3$ e $a_3 \neq 0$, a função polinomial f denomina-se *função cúbica*.

Definição 3.5. Seja n um número natural, com $n \geq 2$, e D_i conjuntos não vazios, tais que $D_i \cap D_j = \emptyset$, para todo $i \neq j$ e $i, j = 1, 2, \dots, n$. Sejam $f_i : D_i \rightarrow Y$ e $D = \bigcup_{i=1}^n D_i$. A função $f : D \rightarrow Y$, dada por $f(x) = f_i(x)$, se $x \in D_i$, denomina-se *função definida por partes*. Escreve-se:

$$f(x) = \begin{cases} f_1(x), & \text{se } x \in D_1; \\ f_2(x), & \text{se } x \in D_2; \\ \vdots & \\ f_n(x), & \text{se } x \in D_n. \end{cases}$$

Exemplo 3.6. A função $f : \mathbb{R} \rightarrow \mathbb{R}$, definida por

$$f(x) = \begin{cases} 0, & \text{se } x < 0; \\ x^2, & \text{se } 0 \leq x \leq 2; \\ x + 3, & \text{se } x > 2; \end{cases}$$

é uma função definida por partes. O gráfico de f é representado na Figura 3.1. A imagem de f é o conjunto $[0, 4] \cup (5, +\infty)$. Com efeito, dado $y \in [0, 4]$, consideramos $x = \sqrt{y}$. Então $x \in [0, 2]$ e $f(x) = f(\sqrt{y}) = (\sqrt{y})^2 = y$. Logo, $y \in \text{Im}(f)$. Assim,

$$[0, 4] \subset \text{Im}(f). \quad (3.1)$$

Agora, dado $y \in (5, +\infty)$, consideremos $x = y - 3$. Então, $x \in (2, +\infty)$ e $f(x) = f(y - 3) = (y - 3) + 3 = y$. Logo, $y \in \text{Im}(f)$. Assim,

$$(5, +\infty) \subset \text{Im}(f). \quad (3.2)$$

De (3.1) e (3.2), $[0, 4] \cup (5, +\infty) \subset \text{Im}(f)$.

Vejam agora que $\text{Im}(f) \subset [0, 4] \cup (5, +\infty)$. De fato, dado $y \in \text{Im}(f)$, existe um $x \in \mathbb{R}$ tal que $y = f(x)$. Se $x < 0$, então $y = f(x) = 0 \in [0, 4]$. Logo, $y \in [0, 4]$. Assim,

$$\text{Im}(f) \subset [0, 4] \cup (5, +\infty). \quad (3.3)$$

Se $x \in [0, 2]$, então $y = f(x) = x^2 \in [0, 4]$. Logo, $y \in [0, 4]$. Assim,

$$\text{Im}(f) \subset [0, 4] \cup (5, +\infty). \quad (3.4)$$

Se $x \in (2, +\infty)$, então $y = f(x) = x + 3 \in (5, +\infty)$. Logo, $y \in (5, +\infty)$. Assim,

$$\text{Im}(f) \subset [0, 4] \cup (5, +\infty). \quad (3.5)$$

De (3.3), (3.4) e (3.5), $\text{Im}(f) \subset [0, 4] \cup (5, +\infty)$.

Definição 3.7. Duas funções $f : A \rightarrow B$ e $g : C \rightarrow D$ são chamadas *funções iguais* se, $A = C$, $B = D$ e $f(x) = g(x)$, para todo $x \in A$.

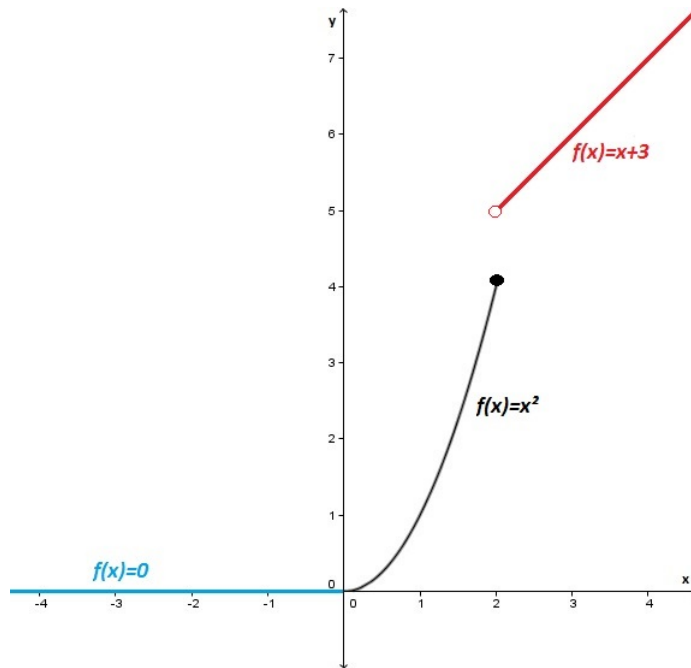


Figura 3.1: Gráfico da função f

Definição 3.8. Sejam $f : A \rightarrow B$ e $g : C \rightarrow B$. Diz-se que f é uma *restrição* de g se $A \subset C$ e $f(x) = g(x)$, para todo $x \in A$. Neste caso, denota-se $f = g|_A$.

3.1.0.1 Função bijetiva

Definição 3.9. Sejam A e B dois conjuntos e $f : A \rightarrow B$.

- (a) Diz-se que f é *injetiva* se, para quaisquer x_1, x_2 pertencentes a A , tais que $x_1 \neq x_2$, tenha-se $f(x_1) \neq f(x_2)$. Equivalentemente, f é injetiva se, para quaisquer x_1, x_2 pertencentes à A , tais que $f(x_1) = f(x_2)$, tenha-se $x_1 = x_2$.
- (b) Diz-se que f é *sobrejetiva* se, para quaisquer $y \in B$, existe um $x \in A$ tal que $f(x) = y$. Isto é, f é sobrejetiva, se $\text{Im} f = B$.
- (c) Diz-se que $f : A \rightarrow B$ é *bijetiva* se, f for injetiva e sobrejetiva.

Exemplo 3.10. Seja $f : \mathbb{R} \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$. Esta função não é injetiva, pois $f(-3) = 10 = f(3)$. Além disso, f também não é sobrejetiva. De fato, para $y = 0$, não existe $x \in \mathbb{R}$ tal que $f(x) = 0$, pois, caso contrário, $x^2 + 1 = 0$, para algum $x \in \mathbb{R}$, o qual contradiz o fato de $x^2 + 1 \geq 1 > 0$.

Consideremos agora uma restrição da função f . A função $g = f|_{[0, +\infty)} : [0, +\infty) \rightarrow \mathbb{R}$ é injetiva. Com efeito, dados $x_1, x_2 \in [0, +\infty)$ tais que $g(x_1) = g(x_2)$. Então

$x_1^2 + 1 = x_2^2 + 1$ e, assim, $x_1^2 = x_2^2$. Logo, $|x_1| = |x_2|$. Como $x_1, x_2 \in [0, +\infty)$, segue que $x_1 = x_2$.

Por outro lado, a função $h : \mathbb{R} \rightarrow [1, +\infty)$, definida por $h(x) = x^2 + 1$, é sobrejetiva. Com efeito, dado $y \in [1, +\infty)$, consideremos $x = \sqrt{y-1} \in \mathbb{R}$. Então, $h(x) = h(\sqrt{y-1}) = (\sqrt{y-1})^2 + 1 = (y-1) + 1 = y$. Isto mostra que $\text{Im}(h) = [1, +\infty)$.

Analogamente, as demonstrações anteriores, prova-se que a função $l : [0, +\infty) \rightarrow [1, +\infty)$, definida por $l(x) = x^2 + 1$, é bijetiva.

Sejam A e B subconjuntos de \mathbb{R} . Utilizando a representação gráfica de uma função $f : A \rightarrow B$ no plano cartesiano, podemos verificar se esta é injetiva, sobrejetiva ou bijetiva. Para isto, basta analisarmos a interseção das retas paralelas ao eixo x , que passam pelo ponto $(0, y)$ com $y \in B$.

- (1) Se cada uma dessas retas interseção o gráfico em um só ponto, então a função é injetiva.
- (2) Se cada uma dessas retas interseção o gráfico em um ou mais pontos, então a função é sobrejetiva.
- (3) Se cada uma dessas retas interseção o gráfico em um só ponto, então a função é bijetiva.

Exemplo 3.11. Seja f a função definida no Exemplo 3.6. Analisando o gráfico de f , apresentado na Figura 3.1, note que, a reta paralela ao eixo x , passando pelo ponto $(0, -1)$, não interseção o gráfico de f ; mostrando que f não é sobrejetiva.

3.2 Matrizes

Ao longo desta seção, o conjunto R será um anel comutativo com unidade¹ (que podemos considerar como sendo $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ ou \mathbb{Z}_n).

Definição 3.12. Sejam $m, n \in \mathbb{N}$ e $I_{mn} = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq n\}$. Define-se a *matriz A de ordem $m \times n$* (lê-se: m por n), como a função A que a cada par ordenado

¹Na literatura, existem muitos autores que tratam sobre teoria de anéis. O leitor interessado, pode ver, por exemplo, em (HYGINO, 2001, pág. 218).

(i, j) , no conjunto I_{mn} , associa um único elemento $A(i, j) = a_{ij}$ pertencente a R . Isto é,

$$\begin{aligned} A : I_{mn} &\longrightarrow R \\ (i, j) &\longmapsto A(i, j) = a_{ij} \end{aligned}$$

Cada elemento a_{ij} é chamado *entrada* da matriz. Uma matriz A de ordem $m \times n$ com entradas no anel R é usualmente representada por um arranjo retangular com m linhas e n colunas da forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

ou simplesmente por $A = [a_{ij}]_{m \times n}$.

O símbolo a_{ij} representa a entrada da matriz A que está na i -ésima linha e j -ésima coluna. O conjunto das matrizes de ordem $m \times n$ com entradas no anel R será denotado por $\mathcal{M}_{m \times n}(R)$.

As matrizes de ordem $1 \times n$

$$L_i = \left[a_{i1} \quad a_{i2} \quad \cdots \quad a_{in} \right],$$

$1 \leq i \leq m$, são chamadas *linhas* da matriz $A = [a_{ij}]_{m \times n}$. Escreve-se,

$$A = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix}.$$

As matrizes de ordem $m \times 1$

$$C_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix},$$

$1 \leq j \leq n$, são chamadas *colunas* da matriz $A = [a_{ij}]_{m \times n}$. Escreve-se,

$$A = \begin{bmatrix} C_1 & \cdots & C_n \end{bmatrix}.$$

Uma matriz A de ordem $n \times n$ denomina-se *matriz quadrada*. Neste caso, a matriz $A = [a_{ij}]_{n \times n}$ diz-se que é uma *matriz de ordem n* e, denota-se por $A = [a_{ij}]_n$. Denotaremos por $\mathcal{M}_n(R)$, o conjunto de matrizes quadradas de ordem n com entradas no anel R .

A matriz quadrada $A = [a_{ij}]_n$, onde $a_{ij} = 0$, para todo $i, j = 1, \dots, n$ com $i \neq j$, denomina-se *matriz diagonal*.

A matriz quadrada $I_n = [\delta_{ij}]_n$, onde

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j; \\ 0, & \text{se } i \neq j; \end{cases}$$

denomina-se *matriz identidade*.

A matriz de ordem $m \times n$ $O_{m \times n} = [e_{ij}]_{m \times n}$, onde $e_{ij} = 0$, para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$, denomina-se *matriz nula*.

Dada uma matriz $A = [a_{ij}]_{m \times n}$, define-se a *matriz oposta de A* , denotada por $-A$, como a matriz $-A = [-a_{ij}]_{m \times n}$.

Dada uma matriz $A = [a_{ij}]_{m \times n}$, define-se a *matriz transposta de A* , denotada por A^t , como a matriz $A^t = [b_{ij}]_{n \times m}$, onde $b_{ij} = a_{ji}$, para todo $1 \leq i \leq n$ e para todo $1 \leq j \leq m$.

Exemplo 3.13. A matriz transposta da matriz quadrada $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix}$ é a matriz

$$A^t = \begin{bmatrix} 1 & 4 & 0 \\ 2 & 1 & 2 \\ 3 & 6 & -3 \end{bmatrix}.$$

Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$ duas matrizes. Diz-se que a matriz A é igual a matriz B se, $a_{ij} = b_{ij}$, para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$.

3.2.1 Operações com matrizes

3.2.1.1 Adição de matrizes

Definição 3.14. Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$ duas matrizes em $\mathcal{M}_{m \times n}(R)$. Defina-se a soma de $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$, denotada por $A + B$, como a matriz

$$C = [c_{ij}]_{m \times n},$$

onde

$$c_{ij} = a_{ij} + b_{ij},$$

para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$. Isto define uma operação de *adição de matrizes* em $\mathcal{M}_{m \times n}(R)$.

Dadas as matrizes $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$, denota-se $A - B = A + (-B)$.

Exemplo 3.15. Dadas as matrizes $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix}$ e $B = \begin{bmatrix} 5 & 3 & 3 \\ 4 & -1 & 6 \\ 1 & 2 & -4 \end{bmatrix}$ em $\mathcal{M}_3(\mathbb{R})$, têm-se:

$$A + B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix} + \begin{bmatrix} 5 & 3 & 3 \\ 4 & -1 & 6 \\ 1 & 2 & -4 \end{bmatrix} = \begin{bmatrix} 1+5 & 2+3 & 3+3 \\ 4+4 & 1-1 & 6+6 \\ 0+1 & 2+2 & -3-4 \end{bmatrix} = \begin{bmatrix} 6 & 5 & 6 \\ 8 & 0 & 12 \\ 1 & 4 & -7 \end{bmatrix}$$

e

$$A - B = A + (-B) = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix} + \begin{bmatrix} -5 & -3 & -3 \\ -4 & 1 & -6 \\ -1 & -2 & 4 \end{bmatrix} = \begin{bmatrix} -4 & -1 & 0 \\ 0 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Proposição 3.16. Dadas três matrizes A , B e C pertencentes a $\mathcal{M}_{m \times n}(R)$, a operação de adição de matrizes, satisfaz as seguintes propriedades:

- (1) $A + B = B + A$ (comutativa);
- (2) $(A + B) + C = A + (B + C)$ (associativa);
- (3) $A + O_{m \times n} = A$ (existência do elemento neutro aditivo);

(4) $A + (-A) = O_{m \times n}$ (existência do elemento oposto).

Demonstração.

(1) Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$. Então

$$A + B = [a_{ij}]_{m \times n} + [b_{ij}]_{m \times n} = [a_{ij} + b_{ij}]_{m \times n} = [b_{ij} + a_{ij}]_{m \times n} = [b_{ij}]_{m \times n} + [a_{ij}]_{m \times n} = B + A.$$

(2) Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$. Então

$$\begin{aligned}(A + B) + C &= [a_{ij} + b_{ij}]_{m \times n} + [c_{ij}]_{m \times n} \\ &= [(a_{ij} + b_{ij}) + c_{ij}]_{m \times n} \\ &= [a_{ij} + (b_{ij} + c_{ij})]_{m \times n} \\ &= [a_{ij}]_{m \times n} + [b_{ij} + c_{ij}]_{m \times n} \\ &= A + (B + C).\end{aligned}$$

(3) Seja $A = [a_{ij}]_{m \times n}$. Como $O_{m \times n} = [e_{ij}]_{m \times n}$, onde $e_{ij} = 0$, para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$, temos

$$A + O_{m \times n} = [a_{ij} + e_{ij}]_{m \times n} = [a_{ij} + 0]_{m \times n} = [a_{ij}]_{m \times n} = A.$$

(4) Seja $A = [a_{ij}]_{m \times n}$. Como $-A = [-a_{ij}]_{m \times n}$, temos

$$A + (-A) = [a_{ij} + (-a_{ij})]_{m \times n} = [e_{ij}]_{m \times n} = O_{m \times n}.$$

□

Observação 3.17. A Proposição 3.16, estabelece que o conjunto $\mathcal{M}_{m \times n}(R)$ munido da operação de adição é um grupo abeliano ¹.

¹Na literatura, existem muitos autores que tratam sobre teoria de grupos. O leitor interessado, pode ver, por exemplo, em (HYGINO, 2001, pág. 137).

3.2.1.2 Multiplicação de escalares por matrizes

Definição 3.18. Seja $A = [a_{ij}]_{m \times n}$ uma matriz em $\mathcal{M}_{m \times n}(R)$. Dado $\lambda \in R$, define-se o produto do escalar λ pela matriz $A = [a_{ij}]_{m \times n}$, denotada por λA , como a matriz

$$C = [c_{ij}]_{m \times n},$$

onde

$$c_{ij} = \lambda a_{ij},$$

para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$. Isto define uma operação de *multiplicação de um escalar por uma matriz*.

Proposição 3.19. Dadas as matrizes A e B pertencentes a $\mathcal{M}_{m \times n}(R)$ e λ, μ escalares pertencentes a R , a operação de multiplicação de escalares por matrizes, satisfaz as seguintes propriedades:

- (1) $\lambda(\mu A) = (\lambda\mu)A = \mu(\lambda A)$;
- (2) $\lambda(A + B) = \lambda A + \lambda B$;
- (3) $(\lambda + \mu)A = \lambda A + \mu A$;
- (4) $1A = A$.

Demonstração.

(1) Sejam $A = [a_{ij}]_{m \times n}$ e $\lambda, \mu \in R$. Então

$$\lambda(\mu A) = \lambda(\mu[a_{ij}]_{m \times n}) = \lambda[\mu a_{ij}]_{m \times n} = [\lambda(\mu a_{ij})]_{m \times n} = [(\lambda\mu)a_{ij}]_{m \times n} = (\lambda\mu)[a_{ij}]_{m \times n} = (\lambda\mu)A.$$

Por outro lado, sendo λ e μ quaisquer, pelo visto anteriormente, segue que

$$\lambda(\mu A) = (\lambda\mu)A = (\mu\lambda)A = \mu(\lambda A).$$

Portanto, $\lambda(\mu A) = (\lambda\mu)A = \mu(\lambda A)$.

(2) Sejam $A = [a_{ij}]_{m \times n}$, $B = [b_{ij}]_{m \times n}$ e $\lambda \in R$. Então

$$\begin{aligned}\lambda(A + B) &= \lambda[a_{ij} + b_{ij}]_{m \times n} \\ &= [\lambda(a_{ij} + b_{ij})]_{m \times n} \\ &= [\lambda a_{ij} + \lambda b_{ij}]_{m \times n} \\ &= [\lambda a_{ij}]_{m \times n} + [\lambda b_{ij}]_{m \times n} \\ &= \lambda[a_{ij}]_{m \times n} + \lambda[b_{ij}]_{m \times n} \\ &= \lambda A + \lambda B.\end{aligned}$$

(3) Sejam $A = [a_{ij}]_{m \times n}$ e $\lambda, \mu \in R$. Então

$$\begin{aligned}(\lambda + \mu)A &= (\lambda + \mu)[a_{ij}]_{m \times n} \\ &= [(\lambda + \mu)a_{ij}]_{m \times n} \\ &= [\lambda a_{ij} + \mu a_{ij}]_{m \times n} \\ &= [\lambda a_{ij}]_{m \times n} + [\mu a_{ij}]_{m \times n} \\ &= \lambda[a_{ij}]_{m \times n} + \mu[a_{ij}]_{m \times n} \\ &= \lambda A + \mu A.\end{aligned}$$

(4) Seja $A = [a_{ij}]_{m \times n}$. Então

$$1A = 1[a_{ij}]_{m \times n} = [1 \cdot a_{ij}]_{m \times n} = [a_{ij}]_{m \times n} = A.$$

□

Observação 3.20. As proposições 3.16 e 3.19, estabelecem que o conjunto $\mathcal{M}_{m \times n}(R)$ munido das operações de adição e multiplicação por escalar é um R -módulo ¹.

¹Na literatura, existem muitos autores que tratam sobre teoria de módulos. O leitor interessado, pode ver, por exemplo, em POLCINO (1972).

3.2.1.3 Multiplicação de matrizes

Definição 3.21. Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{jk}]_{n \times p}$ duas matrizes com entradas no anel R . Define-se o *produto* de A e B , denotada por AB , como a matriz

$$C = [c_{ik}]_{m \times p},$$

onde

$$c_{ik} = \sum_{j=1}^n a_{ij}b_{jk} = a_{i1}b_{1k} + \cdots + a_{in}b_{nk},$$

para todo $1 \leq i \leq m$ e para todo $1 \leq k \leq p$. Isto define uma *multiplicação* no conjunto de matrizes de ordem $m \times n$ e $n \times p$ no conjunto de matrizes ordem $m \times p$. Em particular, isto define uma operação de *multiplicação* no conjunto de matrizes quadradas com entradas no anel R .

Em geral, dadas duas matriz quadradas A e B , da mesma ordem, e com entradas no anel R , o produto AB não é necessariamente igual a BA , ou seja, a multiplicação de matrizes não é comutativo.

Exemplo 3.22. Dada as matrizes $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix}$ e $B = \begin{bmatrix} 5 & 3 & 3 \\ 4 & -1 & 6 \\ 1 & 2 & -4 \end{bmatrix}$ em $\mathcal{M}_3(\mathbb{Z})$,

temos que

$$\begin{aligned} AB &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix} \begin{bmatrix} 5 & 3 & 3 \\ 4 & -1 & 6 \\ 1 & 2 & -4 \end{bmatrix} \\ &= \begin{bmatrix} 1 \cdot 5 + 2 \cdot 4 + 3 \cdot 1 & 1 \cdot 3 + 2 \cdot (-1) + 3 \cdot 2 & 1 \cdot 3 + 2 \cdot 6 + 3 \cdot (-4) \\ 4 \cdot 5 + 1 \cdot 4 + 6 \cdot 1 & 4 \cdot 3 + 1 \cdot (-1) + 6 \cdot 2 & 4 \cdot 3 + 1 \cdot 6 + 6 \cdot (-4) \\ 0 \cdot 5 + 2 \cdot 4 + (-3) \cdot 1 & 0 \cdot 3 + 2 \cdot (-1) + (-3) \cdot 2 & 0 \cdot 3 + 2 \cdot 6 + (-3) \cdot (-4) \end{bmatrix} \\ &= \begin{bmatrix} 16 & 7 & 3 \\ 30 & 23 & -6 \\ 5 & -8 & 24 \end{bmatrix} \end{aligned}$$

e

$$\begin{aligned}
BA &= \begin{bmatrix} 5 & 3 & 3 \\ 4 & -1 & 6 \\ 1 & 2 & -4 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix} \\
&= \begin{bmatrix} 5 \cdot 1 + 3 \cdot 4 + 3 \cdot 0 & 5 \cdot 2 + 3 \cdot 1 + 3 \cdot 2 & 5 \cdot 3 + 3 \cdot 6 + 3 \cdot (-3) \\ 4 \cdot 1 + (-1) \cdot 4 + 6 \cdot 0 & 4 \cdot 2 + (-1) \cdot 1 + 6 \cdot 2 & 4 \cdot 3 + (-1) \cdot 6 + 6 \cdot (-3) \\ 1 \cdot 1 + 2 \cdot 4 + (-4) \cdot 0 & 1 \cdot 2 + 2 \cdot 1 + (-4) \cdot 2 & 1 \cdot 3 + 2 \cdot 6 + (-4) \cdot (-3) \end{bmatrix} \\
&= \begin{bmatrix} 17 & 19 & 24 \\ 0 & 19 & -12 \\ 9 & -4 & 27 \end{bmatrix}.
\end{aligned}$$

Proposição 3.23. *A multiplicação de matrizes, satisfaz as seguintes propriedades:*

- (1) se $A \in \mathcal{M}_{m \times n}(R)$, $B \in \mathcal{M}_{n \times p}(R)$ e $C \in \mathcal{M}_{p \times q}(R)$ então $A(BC) = (AB)C$ (associatividade da multiplicação);
- (2) se $A \in \mathcal{M}_{m \times n}(R)$ então $AI_n = A$ e $I_m A = A$;
- (3) se $A \in \mathcal{M}_{m \times n}(R)$ e $B, C \in \mathcal{M}_{n \times p}(R)$ então $A(B+C) = AB + AC$ (distributividade à esquerda da multiplicação em relação à adição);
- (4) se $A, B \in \mathcal{M}_{m \times n}(R)$ e $C \in \mathcal{M}_{n \times p}(R)$ então $(A+B)C = AC + BC$ (distributividade à direita da multiplicação em relação à adição);
- (5) se $A \in \mathcal{M}_{m \times n}(R)$ então $AO_{n \times p} = O_{m \times p}$ e $O_{p \times m}A = O_{p \times n}$.

Demonstração.

(1) Sejam $A = [a_{ij}]_{m \times n}$, $B = [b_{jk}]_{n \times p}$ e $C = [c_{kr}]_{p \times q}$. Então

$$(AB)C = ([a_{ij}]_{m \times n} [b_{jk}]_{n \times p}) [c_{kr}]_{p \times q} = \left[\sum_{j=1}^n a_{ij} b_{jk} \right]_{m \times p} [c_{kr}]_{p \times q} = [f_{ir}]_{m \times q},$$

onde

$$f_{ir} = \sum_{k=1}^p \left(\sum_{j=1}^n a_{ij} b_{jk} \right) c_{kr} = \sum_{k=1}^p \sum_{j=1}^n (a_{ij} b_{jk}) c_{kr} = \sum_{j=1}^n \sum_{k=1}^p (a_{ij} b_{jk}) c_{kr}. \quad (3.6)$$

Por outro lado,

$$A(BC) = [a_{ij}]_{m \times n} ([b_{jk}]_{n \times p} [c_{kr}]_{p \times q}) = [a_{ij}]_{m \times n} \left[\sum_{k=1}^p b_{jk} c_{kr} \right]_{n \times q} = [g_{ir}]_{m \times q},$$

onde

$$g_{ir} = \sum_{j=1}^n \left(a_{ij} \sum_{k=1}^p (b_{jk} c_{kr}) \right) = \sum_{j=1}^n \sum_{k=1}^p a_{ij} (b_{jk} c_{kr}) = \sum_{j=1}^n \sum_{k=1}^p (a_{ij} b_{jk}) c_{kr}. \quad (3.7)$$

Logo, de (3.6) e (3.7), segue que $f_{ir} = g_{ir}$, para todo $1 \leq i \leq m$ e para todo $1 \leq r \leq q$.

Portanto, $A(BC) = (AB)C$.

(2) Sejam $A = [a_{ij}]_{m \times n}$ e $I_n = [\delta_{jk}]_n$. Provemos que $AI_n = A$. Seja, $AI_n = [d_{ik}]_{m \times n}$, onde $d_{ik} = \sum_{j=1}^n a_{ij} \delta_{jk}$. Temos que

$$d_{ik} = a_{ik} \delta_{kk} + \sum_{\substack{j=1 \\ j \neq k}}^n a_{ij} \delta_{jk} = a_{ik} \cdot 1 + \sum_{\substack{j=1 \\ j \neq k}}^n a_{ij} \cdot 0 = a_{ik}.$$

Portanto, $AI_n = [a_{ik}]_{m \times n} = A$.

Provemos agora que $I_m A = A$, onde $I_m = [\delta_{ki}]_m$. Seja, $I_m A = [d_{kj}]_{m \times n}$, onde $d_{kj} = \sum_{i=1}^m \delta_{ki} a_{ij}$. Temos que

$$d_{kj} = \delta_{kk} a_{kj} + \sum_{\substack{i=1 \\ i \neq k}}^m \delta_{ki} a_{ij} = 1 \cdot a_{kj} + \sum_{\substack{i=1 \\ i \neq k}}^m 0 \cdot a_{ij} = a_{kj}.$$

Portanto, $I_m A = [a_{kj}]_{m \times n} = A$.

(3) Sejam $A = [a_{ij}]_{m \times n}$, $B = [b_{jk}]_{n \times p}$ e $C = [c_{jk}]_{n \times p}$. Então

$$A(B + C) = [a_{ij}]_{m \times n} [b_{jk} + c_{jk}]_{n \times p} = [d_{ik}]_{m \times p},$$

onde

$$d_{ik} = \sum_{j=1}^n a_{ij} (b_{jk} + c_{jk}) = \sum_{j=1}^n (a_{ij} b_{jk} + a_{ij} c_{jk}) = \sum_{j=1}^n a_{ij} b_{jk} + \sum_{j=1}^n a_{ij} c_{jk}.$$

Por outro lado,

$$AB + AC = \left[\sum_{j=1}^n a_{ij} b_{jk} \right]_{m \times p} + \left[\sum_{j=1}^n a_{ij} c_{jk} \right]_{m \times p} = \left[\sum_{j=1}^n a_{ij} b_{jk} + \sum_{j=1}^n a_{ij} c_{jk} \right]_{m \times p} = [d_{ik}]_{m \times p}.$$

Portanto, $AB + AC = A(B + C)$.

(4) Sejam $A = [a_{ij}]_{m \times n}$, $B = [b_{ij}]_{m \times n}$ e $C = [c_{jk}]_{n \times p}$. Então

$$(A + B)C = [a_{ij} + b_{ij}]_{m \times n} [c_{jk}]_{n \times p} = [d_{ik}]_{m \times p},$$

onde

$$d_{ik} = \sum_{j=1}^n (a_{ij} + b_{ij})c_{jk} = \sum_{j=1}^n a_{ij}(c_{jk} + b_{ij}c_{jk}) = \sum_{j=1}^n a_{ij}c_{jk} + \sum_{j=1}^n b_{ij}c_{jk}.$$

Por outro lado,

$$AC + BC = \left[\sum_{j=1}^n a_{ij}c_{jk} \right]_{m \times p} + \left[\sum_{j=1}^n b_{ij}c_{jk} \right]_{m \times p} = \left[\sum_{j=1}^n a_{ij}c_{jk} + \sum_{j=1}^n b_{ij}c_{jk} \right]_{m \times p} = [d_{ik}]_{m \times p}$$

Portanto, $AC + BC = (A + B)C$.

(5) Sejam $A = [a_{ij}]_{m \times n}$ e $O_{n \times p} = [e_{jk}]_{n \times p}$. Provemos que $AO_{n \times p} = O_{m \times p}$. Seja, $AO_{n \times p} = [d_{ik}]_{m \times p}$, onde $d_{ik} = \sum_{j=1}^n a_{ij}e_{jk}$. Temos que

$$d_{ik} = \sum_{j=1}^n a_{ij}e_{jk} = \sum_{j=1}^n a_{ij} \cdot 0 = 0.$$

Portanto, $AO_{n \times p} = O_{m \times p}$.

Provemos agora que $O_{p \times m}A = O_{p \times n}$, onde $O_{p \times m} = [e_{ki}]_{p \times m}$. Seja, $O_{p \times m}A = [d_{kj}]_{p \times n}$, onde $d_{kj} = \sum_{i=1}^m e_{ki}a_{ij}$. Temos que

$$d_{kj} = \sum_{i=1}^m e_{ki}a_{ij} = \sum_{i=1}^m 0 \cdot a_{ij} = 0.$$

Portanto, $O_{p \times m}A = O_{p \times n}$. □

Exemplo 3.24. Sejam

$$A = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 6 & 6 \\ 3 & 2 \end{bmatrix} \quad \text{e} \quad C = \begin{bmatrix} 0 & 6 \\ 1 & 4 \end{bmatrix}$$

pertencentes a $\mathcal{M}_2(\mathbb{Z}_7)$. Para determinar $2(AB + AC)$, calculemos $2A$ e $B + C$, pois

$2(AB + AC) = 2A(B + C)$. Temos que

$$B + C = \begin{bmatrix} 6 & 6 \\ 3 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 6 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 12 \\ 4 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 4 & 6 \end{bmatrix}$$

e

$$2A = 2 \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 10 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 3 & 6 \end{bmatrix}.$$

Logo,

$$2(AB+AC) = 2A(B+C) = \begin{bmatrix} 4 & 2 \\ 3 & 6 \end{bmatrix} \cdot \begin{bmatrix} 6 & 5 \\ 4 & 6 \end{bmatrix} = \begin{bmatrix} 4 \cdot 6 + 2 \cdot 4 & 4 \cdot 5 + 2 \cdot 6 \\ 3 \cdot 6 + 6 \cdot 4 & 3 \cdot 5 + 6 \cdot 6 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 0 & 2 \end{bmatrix}.$$

Definição 3.25. Diz-se que uma matriz A , pertencente a $\mathcal{M}_n(R)$, é *inversível*, se existe uma matriz B , pertencente a $\mathcal{M}_n(R)$, tal que

$$AB = BA = I_n.$$

A matriz B é chamada *matriz inversa de A* .

Observação 3.26. Se uma matriz A , pertencente a $\mathcal{M}_n(R)$, é inversível, então a matriz inversa de A é única e denota-se por A^{-1} . De fato, se $B, D \in \mathcal{M}_n(R)$ são tais que $AB = BA = I_n = AD = DA$, tem-se

$$B = BI_n = B(AD) = (BA)D = I_n D = D.$$

Portanto, $B = D$, provando que a matriz inversa de A é única.

Exemplo 3.27. A matriz inversa de $A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \in \mathcal{M}_2(\mathbb{R})$ é a matriz $B = \begin{bmatrix} 2 & -1 \\ -3/2 & 1 \end{bmatrix} \in \mathcal{M}_2(\mathbb{R})$. De fato,

$$AB = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 & -1 \\ -3/2 & 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 2 + 2 \cdot (-3/2) & 2 \cdot (-1) + 2 \cdot 1 \\ 3 \cdot 2 + 4 \cdot (-3/2) & 3 \cdot (-1) + 4 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

e

$$BA = \begin{bmatrix} 2 & -1 \\ -3/2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 \cdot 2 + (-1) \cdot 3 & 2 \cdot 2 + (-1) \cdot 4 \\ (-3/2) \cdot 2 + 1 \cdot 3 & (-3/2) \cdot 2 + 1 \cdot 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Exemplo 3.28. A matriz inversa de $A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$ é a matriz $B = \begin{bmatrix} 2 & 26 \\ 12 & 1 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$. De fato,

$$AB = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 & 26 \\ 12 & 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 2 + 2 \cdot 12 & 2 \cdot 26 + 2 \cdot 1 \\ 3 \cdot 2 + 4 \cdot 12 & 3 \cdot 26 + 4 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

e

$$BA = \begin{bmatrix} 2 & 26 \\ 12 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 \cdot 2 + 26 \cdot 3 & 2 \cdot 2 + 26 \cdot 4 \\ 12 \cdot 2 + 1 \cdot 3 & 12 \cdot 2 + 1 \cdot 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Enunciaremos a seguir uma proposição que será aplicada na Subseção 4.2.2.

Proposição 3.29. *Seja $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ pertencente a $\mathcal{M}_2(\mathbb{Z}_n)$. Então A é inversível se, e somente se, $ad - bc$ é inversível em \mathbb{Z}_n . Neste caso,*

$$A^{-1} = (ad - bc)^{-1} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Observação 3.30. Um elemento $k \in \mathbb{Z}_n$ é inversível se, e somente se, $\text{mdc}(k, n) = 1$. Se $n = 27$ isto é equivalente a dizer que 0 não é congruente à k módulo 3. Logo, os elementos inversíveis em \mathbb{Z}_{27} e seus respectivos inversos são dados na Tabela 3.1.

a	1	2	4	5	7	8	10	11	13	14	16	17	19	20	22	23	25	26
a^{-1}	1	14	7	11	4	17	19	5	25	2	22	8	10	23	16	20	13	26

Tabela 3.1: Inversos multiplicativos em \mathbb{Z}_{27}

Em particular, pela Proposição 3.29, segue que $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$ é inversível se, e somente se, $(ad - bc) \in \{1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26\}$.

Proposição 3.31. *Se uma matriz $A \in \mathcal{M}_n(R)$ possui uma linha nula, então A não é inversível.*

Demonstração. Seja $A = [a_{ij}]_n$ uma matriz quadrada de ordem n e $L_k = [a_{kj}]_{1 \times n}$ sua k -ésima linha nula. Suponhamos que A é inversível, então existe uma matriz $B = [b_{jr}]_n$ de ordem n tal que $AB = I_n$. Seja $AB = [c_{ir}]_n$, onde $c_{ir} = a_{ij}b_{jr}$. Em particular,

$$c_{kk} = \sum_{j=1}^n a_{kj}b_{jk} = \sum_{k=1}^n 0 \cdot b_{jk} = 0 \neq 1.$$

Logo, $AB \neq I_n$, uma contradição. Portanto, A não é inversível. \square

Proposição 3.32. *Sejam $A, B \in \mathcal{M}_n(R)$. Se A e B são inversíveis, então AB é inversível e $(AB)^{-1} = B^{-1}A^{-1}$.*

Demonstração. Sejam A e B matrizes invertíveis de ordem n . Como

$$(AB)(B^{-1}A^{-1}) = [A(BB^{-1})]A^{-1} = (AI_n)A^{-1} = AA^{-1} = I_n$$

e

$$(B^{-1}A^{-1})(AB) = [B^{-1}(A^{-1}A)]B = (B^{-1}I_n)B = B^{-1}B = I_n,$$

segue que, AB é inversível. Logo, pela unicidade da matriz inversa, $(AB)^{-1} = B^{-1}A^{-1}$.

\square

3.2.2 Aplicação das matrizes à criptografia

Nesta subseção, apresentamos dois métodos para codificar e decodificar mensagens. O estudo de codificação e decodificação de mensagens secretas é denominado *criptografia*. Na linguagem de criptografia, os códigos são denominados *cifras*, as mensagens não codificadas são denominados *textos comuns* e as mensagens criptografadas são denominados *textos cifrados* ou *criptogramas*. O processo de converter o texto comum em cifrado é conhecido como *cifrar* ou *criptografar* e o de recuperar texto comum a partir do texto cifrado é chamado de *decifrar*.

Um *sistema poligráfico*, é um sistema de criptografia no qual o texto comum é dividido em um conjunto de n letras, cada um dos quais é substituído por um conjunto de n letras cifradas. Apresentaremos uma classe de sistemas poligráficos que são baseados na multiplicação de matrizes.

¹ Consideramos o alfabeto português munido do caractere 0, o qual indicará um

¹O restante, desta seção, foi desenvolvida integralmente pelo meu orientador, a quem agradeço.

espaço em branco. Daqui em diante, admitiremos que cada letra de texto comum e de texto cifrado, tem um valor numérico que especifica sua posição no nosso alfabeto, isto será dado pelos elementos do anel R (onde $R = \mathbb{R}$ ou $R = \mathbb{Z}_{27}$), segundo a correspondência da Tabela 3.2.

a	b	c	d	e	f	g	h	i	j	k	l	m	
1	2	3	4	5	6	7	8	9	10	11	12	13	
n	o	p	q	r	s	t	u	v	w	x	y	z	0
14	15	16	17	18	19	20	21	22	23	24	25	26	0

Tabela 3.2: Tabela de correspondência

Seja A uma matriz quadrada de ordem n inversível com entradas em R , que será chamada *matriz chave* para o código. Consideremos o texto comum

$$a_{i_1} \cdots a_{i_t}, \quad (3.8)$$

onde $i_1, \dots, i_t \in \mathbb{N}$, $\{a_{i_1}, \dots, a_{i_t}\} \subset \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}, \mathbf{m}, \mathbf{n}, \mathbf{o}, \mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{0}\}$. Dividimos este texto em um conjunto de n caracteres, os quais serão chamados de *blocos*, completando com o caractere $\mathbf{0}$, se necessário. Seja m o número de blocos do texto comum e

$$a_{i_{n(k-1)+1}} \cdots a_{i_{nk}} \quad (3.9)$$

o k -ésimo bloco do texto comum, para todo $1 \leq k \leq m$. Se $\widehat{a_{i_j}}$ é o valor numérico da Tabela 3.2, correspondente ao caractere a_{i_j} , para todo $n(k-1)+1 \leq j \leq nk$, $1 \leq k \leq m$, a correspondência numérica do bloco de texto comum (3.9) é dado por

$$\widehat{a_{i_{n(k-1)+1}}} \cdots \widehat{a_{i_{nk}}}.$$

Sejam

$$L_k = \left[\widehat{a_{i_{n(k-1)+1}}} \cdots \widehat{a_{i_{nk}}} \right]_{1 \times n} \in \mathcal{M}_{1 \times n}(R)$$

e $MA = [x_{ij}]_{m \times n}$, onde

$$M = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix}_{m \times n}.$$

A matriz $MA \in \mathcal{M}_{m \times n}(R)$ será chamada *matriz cifrada*.

CASO 1: Se a matriz chave A possui entradas no corpo $R = \mathbb{R}$, o texto comum (3.8) é transformado no texto cifrado, utilizando as entradas da matriz cifrada MA , deixando um espaço em branco entre cada uma das entradas desta matriz, como segue:

$$x_{11} \quad \cdots \quad x_{1n} \quad x_{21} \quad \cdots \quad x_{2n} \quad \cdots \quad x_{m1} \quad \cdots \quad x_{mn}, \quad (3.10)$$

o qual será a mensagem transmitida.

CASO 2: Se a matriz chave A possui entradas no anel $R = \mathbb{Z}_{27}$, o texto comum (3.8) é transformado no texto cifrado, utilizando os caracteres do nosso alfabeto correspondentes as entradas da matriz cifrada AM , sem deixar espaços em branco entre cada uma das entradas desta matriz, como segue:

$$b_{11} \cdots b_{1n} b_{21} \cdots b_{2n} \cdots b_{m1} \cdots b_{mn}, \quad (3.11)$$

o qual será a mensagem transmitida, onde $x_{ij} = \widehat{b_{ij}}$, para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$. Este caso é um sistema poligráfico chamado *cifras de Hill* e o texto cifrado (3.11) é chamado *n-cifra de Hill*.

Para decodificar o texto cifrado (3.10) ou (3.11) (mensagem transmitida), devemos fazer um processo similar ao mostrado anteriormente, agrupando em n blocos de m caracteres, obtendo-se novamente a matriz MA . Finalmente, multiplicamos à direita pela inversa da matriz chave obtendo a matriz M (pois $M = (MA)A^{-1}$). Após, isto escrevemos a matriz M como sendo o texto comum.

Exemplo 3.33. Consideremos a matriz chave $A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \in \mathcal{M}_2(\mathbb{R})$. Utilizemos a matriz A para criptografar a mensagem: “Profmat”. Como a matriz A é de ordem 2, dividimos a mensagem em conjuntos de duas letras e acrescentando o caractere 0 para completar o espaço em branco, obtendo os seguintes blocos:

pr of ma t0,

o qual usando a correspondência numérica da Tabela 3.2 fica

16 18 15 6 13 1 20 0.

Forma-se, assim, a matriz

$$M = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix}.$$

A matriz cifrada é dada através do produto

$$MA = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 86 & 104 \\ 48 & 54 \\ 29 & 30 \\ 40 & 40 \end{bmatrix}.$$

Portanto, o texto cifrado é

$$86 \ 104 \ 48 \ 54 \ 29 \ 30 \ 40 \ 40,$$

o qual é a mensagem transmitida.

Exemplo 3.34. Decifremos o criptograma

$$86 \ 104 \ 48 \ 54 \ 29 \ 30 \ 40 \ 40,$$

dado no Exemplo 3.33, utilizando a mesma matriz chave A . Agrupando o criptograma, em blocos de dois elementos, obtém-se a matriz

$$C = \begin{bmatrix} 86 & 104 \\ 48 & 54 \\ 29 & 30 \\ 40 & 40 \end{bmatrix}.$$

Para obter os pares do texto comum, multiplicamos, à direita a matriz C pela inversa de

A , dada no Exemplo 3.27:

$$CA^{-1} = \begin{bmatrix} 86 & 104 \\ 48 & 54 \\ 29 & 30 \\ 40 & 40 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -3/2 & 1 \end{bmatrix} = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix}.$$

Agora, dividimos as entradas da matriz CA^{-1} em 4 blocos, contendo cada um 2 elementos de cada linha, obtendo-se

$$\begin{bmatrix} 16 & 18 & 15 & 6 & 13 & 1 & 20 & 0 \end{bmatrix},$$

o qual usando a correspondência numérica da Tabela 3.2 fica

$$\begin{bmatrix} \text{pr} & \text{of} & \text{ma} & \text{t0} \end{bmatrix}.$$

Portanto, a mensagem enviada era “Profmat”.

Exemplo 3.35. Consideremos a matriz chave $A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$. De forma análoga ao Exemplo 3.33, utilizemos a matriz A para criptografar a mensagem: “Profmat”, através das cifras de Hill. Como a matriz A é de ordem 2, dividimos a mensagem em conjuntos de duas letras e acrescentando o caractere 0 para completar o espaço em branco, obtendo os seguintes blocos:

$$\begin{bmatrix} \text{pr} & \text{of} & \text{ma} & \text{t0} \end{bmatrix},$$

o qual usando a correspondência numérica da Tabela 3.2 fica

$$\begin{bmatrix} 16 & 18 & 15 & 6 & 13 & 1 & 20 & 0 \end{bmatrix}.$$

Forma-se, assim, a matriz

$$M = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix}.$$

A matriz cifrada é dada através do produto

$$MA = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 86 & 104 \\ 48 & 54 \\ 29 & 30 \\ 40 & 40 \end{bmatrix} = \begin{bmatrix} 5 & 23 \\ 21 & 0 \\ 2 & 3 \\ 13 & 13 \end{bmatrix}.$$

Portanto, trocando cada uma das entradas da matriz AM , pela sua respectiva correspondência numérica da Tabela 3.2, o texto cifrado é

ewu0bcmmm,

o qual é a mensagem transmitida.

Exemplo 3.36. Decifremos o criptograma

ewu0bcmmm,

dado no Exemplo 3.35, utilizando a mesma chave A . Agrupando o criptograma, em blocos de dois elementos, obtém-se

ew u0 bc mm

o qual usando a correspondência numérica da Tabela 3.2 fica

5 23 21 0 2 3 13 13,

Forma-se, assim, a matriz

$$C = \begin{bmatrix} 5 & 23 \\ 21 & 0 \\ 2 & 3 \\ 13 & 13 \end{bmatrix}.$$

Para obter os pares do texto comum, multiplicamos à direita a matriz C pela inversa de

A , dada no Exemplo 3.28

$$CA^{-1} = \begin{bmatrix} 5 & 23 \\ 21 & 0 \\ 2 & 3 \\ 13 & 13 \end{bmatrix} \begin{bmatrix} 2 & 26 \\ 12 & 1 \end{bmatrix} = \begin{bmatrix} 286 & 153 \\ 42 & 546 \\ 40 & 55 \\ 182 & 351 \end{bmatrix} = \begin{bmatrix} 16 & 18 \\ 15 & 6 \\ 13 & 1 \\ 20 & 0 \end{bmatrix}.$$

Agora, dividimos as entradas da matriz CA^{-1} em 4 blocos, contendo cada um 2 elementos de cada linha, obtendo-se

$$\begin{matrix} 16 & 18 & 15 & 6 & 13 & 1 & 20 & 0, \end{matrix}$$

o qual usando a correspondência numérica da Tabela 3.2.2 fica

pr of ma t0.

Portanto, a mensagem enviada era “Profmat”.

3.3 Determinantes

Definição 3.37. Sejam $n \geq 2$ e $D : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ uma função, satisfazendo os seguintes axiomas:

(D1) D é linear como função de cada linha separadamente. Isto é, se $L_i = L'_i + tL''_i$ e $t \in \mathbb{R}$ então

$$D \begin{bmatrix} L_1 \\ \vdots \\ L'_i + tL''_i \\ \vdots \\ L_n \end{bmatrix} = D \begin{bmatrix} L_1 \\ \vdots \\ L'_i \\ \vdots \\ L_n \end{bmatrix} + tD \begin{bmatrix} L_1 \\ \vdots \\ L''_i \\ \vdots \\ L_n \end{bmatrix},$$

para todo $i = 1, \dots, n$.

(D2) Para todo $A \in \mathcal{M}_n(\mathbb{R})$, se duas linhas adjacentes L_i e L_{i+1} de A são iguais, então $D(A) = 0$.

(D3) $D(I_n) = 1$.

Se existe uma função, $D : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ satisfazendo os axiomas D1, D2 e D3, ela é única e denomina-se *função determinante* (ou simplesmente *determinante*) e, denota-se por \det . A demonstração é apresentada em (HEFEZ, 2012, Teorema 8.8, pág. 239).

Sejam $n \geq 2$ e $A = [a_{ij}] \in \mathcal{M}_n(\mathbb{R})$. Para cada $(i, j) \in \mathbb{N}^2$, com $1 \leq i, j \leq n$, define-se a matriz A_{ij} como a matriz de ordem $(n-1) \times (n-1)$ obtida de A suprimindo-se a i -ésima linha e a j -ésima coluna.

Seja $n \geq 2$. Definiremos indutivamente uma função $D^n : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ satisfazendo os axiomas D1, D2 e D3 da Definição 3.37. Para $n = 2$, definimos $D^2 : \mathcal{M}_2(\mathbb{R}) \rightarrow \mathbb{R}$, como sendo

$$D^2 \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) = a \cdot d - b \cdot c.$$

A função D^2 , satisfaz os axiomas D1, D2 e D3. De fato, se $a, a', b, b', c, c', d, d', t \in \mathbb{R}$, então

$$\begin{aligned} D^2 \left(\begin{bmatrix} a + ta' & b + tb' \\ c & d \end{bmatrix} \right) &= (a + ta') \cdot d - c \cdot (b + tb') \\ &= (ad - bc) + t(a' \cdot d - c \cdot b') \\ &= D^2 \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) + t \cdot D^2 \left(\begin{bmatrix} a' & b' \\ c & d \end{bmatrix} \right) \end{aligned}$$

e

$$\begin{aligned} D^2 \left(\begin{bmatrix} a & b \\ c + tc' & d + td' \end{bmatrix} \right) &= a \cdot (d + td') - b \cdot (c + tc') \\ &= (ad - bc) + t(a \cdot d' - b \cdot c') \\ &= D^2 \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) + t \cdot D^2 \left(\begin{bmatrix} a & b \\ c' & d' \end{bmatrix} \right); \end{aligned}$$

mostrando que a função D^2 satisfaz o axioma D1.

Além disto, se $\begin{bmatrix} a & b \\ a & b \end{bmatrix} \in \mathcal{M}_2(\mathbb{R})$, tem-se

$$D^2 \left(\begin{bmatrix} a & b \\ a & b \end{bmatrix} \right) = a \cdot b - b \cdot a = 0,$$

mostrando que a função D^2 satisfaz o axioma D2. Também, tem-se que

$$D^2 \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 1 \cdot 1 - 0 \cdot 0 = 1,$$

isto é, a função D^2 satisfaz o axioma D3.

Seja agora $n \geq 3$. Supondo definido a função $D^{n-1} : \mathcal{M}_{n-1}(\mathbb{R}) \rightarrow \mathbb{R}$, definimos a função $D_i : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$, $1 \leq i \leq n$, como sendo

$$D_i(A) = \sum_{j=1}^n (-1)^{(i+j)} \cdot a_{ij} D^{n-1}(A_{ij}).$$

Teorema 3.38. *Seja $n \geq 3$. Se $D^{n-1} : \mathcal{M}_{n-1}(\mathbb{R}) \rightarrow \mathbb{R}$ satisfaz os axiomas D1, D2 e D3, então a função $D_i : \mathcal{M}_n(\mathbb{R}) \rightarrow \mathbb{R}$ também satisfaz os axiomas D1, D2 e D3, para todo $1 \leq i \leq n$.*

Demonstração. Ver em (HEFEZ, 2012, Teorema 8.9, pág. 232). □

Exemplo 3.39. O determinante da matriz $A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$ é dado por

$$\begin{aligned} \det(A) &= 1 \cdot \det \begin{bmatrix} -1 & 3 \\ 1 & 8 \end{bmatrix} - 0 \cdot \det \begin{bmatrix} 2 & 3 \\ 4 & 8 \end{bmatrix} + 2 \cdot \det \begin{bmatrix} 2 & -1 \\ 4 & 1 \end{bmatrix} \\ &= 1 \cdot (-11) + 2 \cdot 6 \\ &= 1. \end{aligned}$$

Para calcular este determinante, foi utilizado a primeira linha da matriz A . Utilizando, a

terceira linha temos o seguinte:

$$\begin{aligned}\det(A) &= 4 \cdot \det \begin{bmatrix} 0 & 2 \\ -1 & 3 \end{bmatrix} - 1 \cdot \det \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} + 8 \cdot \det \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix} \\ &= 4 \cdot 2 - 1 \cdot (-1) + 8 \cdot (-1) \\ &= 1.\end{aligned}$$

O mesmo pode ser feito, também, com a segunda linha.

Proposição 3.40. *Seja $A \in \mathcal{M}_n(\mathbb{R})$. Se A possui uma linha nula então $\det(A) = 0$.*

Demonstração. Se $A \in \mathcal{M}_2(\mathbb{R})$ é imediato. Sejam $n \geq 3$ e $A \in \mathcal{M}_n(\mathbb{R})$. Suponha que $L_k = \begin{bmatrix} a_{k1} & a_{k12} & \cdots & a_{kn} \end{bmatrix}$ é uma linha nula de A . Então

$$\det(A) = D_k(A) = \sum_{j=1}^n (-1)^{(i+j)} \cdot a_{kj} \det(A_{k,j}) = \sum_{j=1}^n (-1)^{(i+j)} \cdot 0 \cdot \det(A_{k,j}) = 0.$$

□

3.4 Sistemas lineares de m equações com n incógnitas

Definição 3.41. Dados os números reais b e a_i , com $i = 1, \dots, n$, à equação

$$\sum_{i=1}^n a_i x_i = b, \tag{3.12}$$

onde x_i são variáveis em \mathbb{R} , denomina-se *equação linear sobre \mathbb{R} nas incógnitas x_i* .

Uma *solução* da Equação (3.12) é uma n -upla de números reais $(\alpha_1, \dots, \alpha_n)$ tal que

$$\sum_{i=1}^n a_i \alpha_i = b.$$

Definição 3.42. Um *sistema linear de m equações com n incógnitas* sobre \mathbb{R} , é um conjunto de m equações lineares, cada uma delas com n incógnitas, consideradas simul-

taneamente. Um sistema linear representa-se do seguinte modo:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m. \end{array} \right. \quad (3.13)$$

ou

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad 1 \leq i \leq m,$$

onde os números reais a_{ij} 's e b_i 's são chamados, respectivamente, de *coeficientes do sistema* e *termos independentes*. Os elementos x_j 's são as *variáveis* em \mathbb{R} do sistema.

Diz-se que uma n -upla de números reais $(\alpha_1, \dots, \alpha_n)$ é *solução* do sistema linear 3.13, quando é solução de cada uma das equações do sistema. O conjunto das soluções é chamado *conjunto solução do sistema*.

Quanto as suas soluções, um sistema linear classifica-se como: *compatível e determinado*, *compatível e indeterminado* ou *incompatível*. Um sistema linear é chamado *compatível e determinado*, quando tem uma única solução, *compatível e indeterminado*, quando tem mais de uma solução e *incompatível*, quando não tem solução.

Definição 3.43. Diz-se que dois sistemas de equações lineares são *equivalentes* se eles admitem as mesmas soluções.

O sistema de m equações e n incógnitas (3.13), pode ser expressado como a *equação matricial*

$$AX = B, \quad (3.14)$$

onde

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

As matrizes A , X e B são chamadas *matrizes dos coeficientes do sistema*, *matriz das incógnitas* e *matriz dos termos independentes*, respectivamente.

Diz-se que $X_0 = [\alpha_i]_{n \times 1}$ é *solução* da equação matricial (3.14), se $AX_0 = B$.

Observação 3.44. Dois sistemas de equações lineares são equivalentes se, e somente se, suas respectivas equações matriciais possuem as mesmas soluções.

A matriz

$$C = [A|B] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right]$$

é chamada de *matriz ampliada* do sistema linear (3.13).

Observação 3.45. Sejam $\alpha_1, \dots, \alpha_n \in \mathbb{R}$. Então $(\alpha_1, \dots, \alpha_n)$ é solução do sistema linear (3.13) se, e somente se, a matriz $X_0 = [\alpha_i]_{n \times 1}$ é solução da equação matricial (3.14).

3.4.1 Operações elementares de matrizes

Definição 3.46. Seja $A \in \mathcal{M}_{m \times n}(\mathbb{R})$. Definimos as *operações elementares nas linhas da matriz A*, como segue:

- (1) Permutar duas linhas de A . Se L_i e L_j são linhas de A , a permutação destas linhas é indicada por $L_i \leftrightarrow L_j$.
- (2) Multiplicar uma linha de A por um número real $k \neq 0$. Se L_i é uma linha de A , a multiplicação da linha L_i por um número real $k \neq 0$ é indicada por $L_i \leftrightarrow kL_i$.
- (3) Somar a uma linha de A uma outra linha de A . Se L_i e L_j são linhas de A , a adição da linha L_i pela linha L_j é indicada por $L_i \leftrightarrow L_i + L_j$.

Definição 3.47. Sejam $A, B \in \mathcal{M}_{m \times n}(\mathbb{R})$. A matriz A é dita *equivalente por linhas* à matriz B , se B pode ser obtida de A , pela aplicação sucessiva de um número finito de operações elementares sobre linhas.

Teorema 3.48. *Dois sistemas lineares com matrizes ampliadas equivalentes têm o mesmo conjunto solução.*

Demonstração. Basta provar que uma operação elementar, sobre a matriz ampliada, sempre produz um sistema equivalente. Para isto, seja

$$AX = B \tag{3.15}$$

uma equação matricial. A equação matricial 3.15 é equivalente ao sistema

$$L_1X = [b_1], \dots, L_iX = [b_i], \dots, L_mX = [b_m], \quad (3.16)$$

onde $A = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix}$ e $B = [b_i]_{m \times 1}$; isto é, X_0 é solução de (3.15) se, e somente se, X_0 é solução de (3.16).

Suponha que a operação elementar consiste na permutação da i -ésima e j -ésima linha de A , com $i < j$. Então, a equação matricial 3.15 é equivalente ao sistema

$$L_1X = [b_1], \dots, L_{i-1}X = [b_{i-1}], L_jX = [b_j], \dots, L_iX = [b_i], \dots, L_mX = [b_m]. \quad (3.17)$$

Agora, é claro que Y é solução do sistema 3.16 se, e somente se, Y é solução do sistema 3.17.

Suponha agora que a operação elementar consiste na substituição da i -ésima linha por k vezes a i -ésima linha ($k \neq 0$). Então, a equação matricial 3.15 é equivalente ao sistema

$$L_1X = [b_1], \dots, L_{i-1}X = [b_{i-1}], (kL_i)X = [kb_i], \dots, L_mX = [b_m]. \quad (3.18)$$

Agora, se Y é solução do sistema 3.16, então

$$L_iY = [b_i].$$

Então

$$(kL_i)Y = k(L_iY) = k[b_i] = [kb_i].$$

Portanto, Y é solução do sistema 3.18. Reciprocamente, se Y é solução do sistema 3.18, então, em particular,

$$(kL_i)Y = [kb_i].$$

Então

$$L_iY = (k^{-1}k)(L_iY) = k^{-1}[kb_i] = [b_i].$$

Portanto, Y é solução do sistema 3.16.

Por fim, suponha que a operação elementar consiste na substituição da i -ésima

linha pela i -ésima linha mais a j -ésima linha, com $i < j$. Então, a equação matricial 3.15 é equivalente ao sistema

$$L_1X = [b_1], \dots, L_iX + L_jX = [b_i + b_j], \dots, L_jX = [b_j], \dots, L_mX = [b_m]. \quad (3.19)$$

Agora, se Y é solução do sistema 3.16, então

$$L_iY = [b_i] \quad \text{e} \quad L_jY = [b_j].$$

Então

$$L_iY + L_jY = [b_i] + [b_j] = [b_i + b_j].$$

Portanto, Y é solução do sistema 3.19. Reciprocamente, seja Y uma solução do sistema 3.19, então, em particular,

$$L_iY + L_jY = [b_i + b_j].$$

Então

$$L_iY = [L_iY + L_jY] - (L_jY) = [b_i + b_j] - [b_j] = [b_i + (b_j - b_j)] = [b_i].$$

Portanto, Y é solução do sistema 3.16. □

Observe que, a noção de equivalência de matrizes por linhas, corresponde à noção de equivalência de sistemas lineares, quando se efetuam as respectivas operações elementares sobre as equações. De fato, a sistemas equivalentes correspondem matrizes ampliadas equivalentes.

3.4.2 Forma escalonada de uma matriz

Definição 3.49. Diz-se que, uma matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ é *reduzida por linha à forma escada* se:

- (1) o primeiro elemento não nulo de cada linha não nula de A for igual a 1;
- (2) cada coluna que contém o primeiro elemento não nulo de alguma linha tem todo os seus outros iguais a zero;
- (3) toda linha nula ocorre abaixo de todas as linhas não nulas;

- (4) se L_1, \dots, L_p são as linhas não nulas, e se o primeiro elemento não nulo da linha L_i ocorre na coluna k_i então $k_1 < \dots < k_p$.

A Figura 3.2, mostra uma matriz reduzida por linha à forma escada.

Figura 3.2: Matriz reduzida por linha à forma escada

Observação 3.50. Se A é uma matriz quadrada de ordem n , reduzida por linha à forma escada, que não possui linhas nulas, então $A = I_n$.

Teorema 3.51 (Existência e unicidade da forma escada). *Dada uma matriz A , existe uma única matriz \tilde{A} reduzida por linha à forma escada equivalente a esta.*

Demonstração. A demonstração da existência é apresentada em (HEFEZ, 2012, Teorema 2.2, pág.34) e, a demonstração da unicidade, em (HEFEZ, 2012, Teorema 2.13, pág.48).

□

Uma operação elementar na linha de uma matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ será representada por e ; denotar-se-á por $e(A)$ a matriz obtida de A aplicando-lhe a operação e .

Definição 3.52. Uma *matriz elementar de ordem n* é uma matriz obtida da matriz identidade I_n a partir da aplicação de uma operação elementar, isto é, trata-se de uma matriz da forma

$$E = e(I_n)$$

onde e é uma operação elementar.

Seja $A \in \mathcal{M}_{m \times n}(\mathbb{R})$. Nos dois próximos resultados, denotaremos por A^i , a i -ésima linha da matriz A . Assim, escreveremos,

$$A = \begin{bmatrix} A^1 \\ \vdots \\ A^m \end{bmatrix}.$$

Enunciaremos a seguir um lema que será aplicado no Teorema 3.54.

Lema 3.53. *Sejam $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ e $B \in \mathcal{M}_{n \times p}(\mathbb{R})$. Então $(AB)^i = A^i B$, para todo $i = 1, \dots, m$.*

Teorema 3.54. *Seja e uma operação elementar sobre as matrizes $\mathcal{M}_{m \times n}(\mathbb{R})$. Considere a matriz elementar $E = e(I_m)$. Então*

$$e(A) = EA, \text{ para todo } A \in \mathcal{M}_{m \times n}(\mathbb{R}).$$

Demonstração. Sejam $A \in \mathcal{M}_{m \times n}(\mathbb{R})$ e $E = e(I_m)$. Pelo Lema 3.53,

$$(EA)^i = E^i A \quad \text{e} \quad A^i = (I_m A)^i = I_m^i A, \quad (3.20)$$

para todo $i = 1, \dots, m$. Provaremos que, $e(A) = EA$, para cada operação elementar e . Consideraremos três casos:

CASO 1: e é a operação elementar de permutar a j -ésima linha com a k -ésima linha, onde $1 \leq j < k \leq m$. Temos que,

$$E^i = I_m^i, \quad \text{para todo } i = 1, \dots, m \text{ com } i \neq j, k,$$

$$E^j = I_m^k \quad \text{e} \quad E^k = I_m^j.$$

Logo, pela Equação 3.20, para todo $i = 1, \dots, m$ com $i \neq j, k$,

$$(EA)^i = E^i A = I_m^i A = A^i, \quad (3.21)$$

e, também,

$$(EA)^j = E^j A = I_m^k A = A^k \quad \text{e} \quad (EA)^k = E^k A = I_m^j A = A^j. \quad (3.22)$$

Portanto, de (3.21) e (3.22),

$$e(A) = EA.$$

CASO 2: e é a operação elementar de multiplicar a j -ésima linha pelo número real $\lambda \neq 0$,

onde $1 \leq j \leq m$. Temos que,

$$E^i = I_m^i, \quad \text{para todo } i = 1, \dots, m \text{ com } i \neq j$$

e

$$E^j = \lambda I_m^j.$$

Logo, pela Equação 3.20, para todo $i = 1, \dots, m$ com $i \neq j$,

$$(EA)^i = E^i A = I_m^i A = A^i, \quad (3.23)$$

e, também,

$$(EA)^j = E^j A = (\lambda I_m^j) A = \lambda (I_m^j A) = \lambda A^j. \quad (3.24)$$

Portanto, de (3.23) e (3.24),

$$e(A) = EA.$$

CASO 3: e é a operação elementar de somar a j -ésima linha com a k -ésima linha, onde $1 \leq j < k \leq m$. Temos que,

$$E^i = I_m^i, \quad \text{para todo } i = 1, \dots, m \text{ com } i \neq j$$

e

$$E^j = I_m^j + I_m^k.$$

Logo, pela Equação 3.20, para todo $i = 1, \dots, m$ com $i \neq j$,

$$(EA)^i = E^i A = I_m^i A = A^i, \quad (3.25)$$

e, também,

$$(EA)^j = E^j A = (I_m^j + I_m^k) A = (I_m^j A) + (I_m^k A) = A^j + A^k. \quad (3.26)$$

Portanto, de (3.25) e (3.26),

$$e(A) = EA.$$

□

Corolário 3.55. *Sejam A e B em $\mathcal{M}_{m \times n}(\mathbb{R})$. Então A é equivalente a B se, e somente se, existem matrizes elementares de ordem m tais que*

$$E_s \cdots E_2 \cdot E_1 \cdot A = B.$$

Demonstração. Por definição, A é equivalente a B quando existem operações elementares e_1, e_2, \dots, e_s tais que

$$e_s(\cdots(e_2(e_1A))\cdots) = B.$$

Pelo Teorema 3.54, a igualdade acima equivale a

$$E_s \cdots E_2 \cdot E_1 \cdot A = B,$$

onde $E_i = e_i(I_m)$, para cada $1 \leq i \leq s$. □

Corolário 3.56. *Toda matriz elementar é invertível e sua inversa também é uma matriz elementar.*

Demonstração. Seja E uma matriz elementar. Seja e a operação elementar tal que $E = e(I)$. Seja e' é a operação elementar inversa de e e se $E' = e'(I_n)$, pelo Teorema 3.54, temos

$$I_n = e'(e(I_n)) = e'(E) = e'(I_n)E = E'E$$

e

$$I_n = e(e'(I_n)) = e(E') = e(I_n)E' = EE'.$$

Logo, E é inversível e $E^{-1} = E'$. □

Teorema 3.57. *Seja $A \in \mathcal{M}_n(\mathbb{R})$. Então, as seguintes afirmações são equivalentes:*

- (1) A é inversível;
- (2) a matriz linha escada reduzida da matriz A é a matriz I_n ;
- (3) A é uma matriz elementar ou é um produto de matrizes elementares.

Demonstração. (1) \Rightarrow (2) Seja \tilde{A} a matriz reduzida por linha à forma escada e equivalente a matriz A . Pelo Corolário 3.55, existem matrizes elementares E_1, \dots, E_s de ordem n tais que

$$E_s \cdots E_1 \cdot A = \tilde{A}.$$

Por outro lado, pelo Corolário 3.56, cada E_i é inversível e sendo A , por hipótese, inversível, pela Proposição 3.32, temos que \tilde{A} é inversível. Então, pela Proposição 3.31, \tilde{A} não possui nenhuma linha nula. Logo, $\tilde{A} = I_n$.

(2) \Rightarrow (3) Suponhamos que A não é uma matriz elementar. Pelo Corolário 3.55, existem matrizes elementares E_1, \dots, E_s de ordem n tais que

$$E_s \cdots E_1 \cdot A = I_n.$$

Pelo Corolário 3.56, cada E_i é invertível, assim, multiplicando, a esquerda, ambos os lados da igualdade por $E_1^{-1} \cdots E_s^{-1}$, tem-se

$$A = E_1^{-1} \cdots E_s^{-1}.$$

Também, pelo Corolário 3.56, cada E_i^{-1} é uma matriz elementar. Portanto, A é um produto de matrizes elementares.

(3) \Rightarrow (1) Sejam E_1, \dots, E_s matrizes elementares tais que $A = E_1 \cdots E_s$. Pelo Corolário 3.56 cada E_i é inversível. Logo, pela Proposição 3.32, A é inversível. \square

Proposição 3.58. *Sejam $A \in \mathcal{M}_n(\mathbb{R})$ uma matriz inversível e e_1, \dots, e_s uma sequência de operações elementares tais que $e_s(\dots(e_2(e_1(A)))) = I_n$. Então essa mesma sequência de operações elementares aplicada a I_n produz A^{-1} .*

Demonstração. Para cada $1 \leq i \leq s$, seja E_i a matriz elementar correspondente à operação e_i . Então

$$E_s \cdots E_1 A = I_n$$

Assim,

$$E_s \cdots E_1 A A^{-1} = I_n A^{-1},$$

onde

$$E_s \cdots E_1 I_n = A^{-1}.$$

\square

Observação 3.59. Do Teorema 3.57, decorre uma forma de descobrir se uma matriz $A \in \mathcal{M}_n(\mathbb{R})$ admite inversa e, caso seja inversível, pela Proposição 3.58, decorre uma forma de determiná-la. Se aplicarmos uma sequência de operações elementares em A até

obtermos uma matriz \tilde{A} linha à forma escada reduzida, pelo Teorema 3.57, A é inversível se, e somente se, $\tilde{A} = I_n$. Se $\tilde{A} = I_n$, pela Proposição 3.58, a sucessão de operações na linha da matriz A , que levam na matriz I_n , é a mesma sucessão de operações que levará I_n em A^{-1} . Assim, para determinar se a matriz A é inversível, reunimos A e I_n na matriz em blocos $[A|I_n]$, e reduzimos por linha à forma escada $[\tilde{A}|B]$. Se $\tilde{A} = I_n$, então a matriz A é inversível e $B = A^{-1}$, ou seja,

$$[A|I_n] \longrightarrow [I_n|A^{-1}];$$

caso contrário, isto é, $\tilde{A} \neq I_n$, a matriz A não é inversível.

Exemplo 3.60. Dada a matriz $A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$, para determinar A^{-1} , caso

exista, forma-se a matriz em blocos $[A|I_3]_{3 \times 6}$ e a reduz por linha à forma escada $[\tilde{A}|B]$.

Assim,

$$\begin{array}{l} \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 2 & -1 & 3 & | & 0 & 1 & 0 \\ 4 & 1 & 8 & | & 0 & 0 & 1 \end{bmatrix} \\ L_2 \leftrightarrow L_2 - 2L_1 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & -1 & -1 & | & -2 & 1 & 0 \\ 0 & 1 & 0 & | & -4 & 0 & 1 \end{bmatrix} \\ L_3 \leftrightarrow L_3 + L_2 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & -1 & -1 & | & -2 & 1 & 0 \\ 0 & 0 & -1 & | & -6 & 1 & 1 \end{bmatrix} \\ L_3 \leftrightarrow L_3 - 4L_1 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & 1 & 1 & | & 2 & -1 & 0 \\ 0 & 0 & -1 & | & -6 & 1 & 1 \end{bmatrix} \\ L_2 \leftrightarrow -1L_2 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & 1 & 1 & | & 2 & -1 & 0 \\ 0 & 0 & -1 & | & -6 & 1 & 1 \end{bmatrix} \\ L_3 \leftrightarrow -1L_3 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & 1 & 1 & | & 2 & -1 & 0 \\ 0 & 0 & 1 & | & 6 & -1 & -1 \end{bmatrix} \\ L_2 \leftrightarrow L_2 + L_3 \\ \begin{bmatrix} 1 & 0 & 2 & | & 1 & 0 & 0 \\ 0 & 1 & 0 & | & -4 & 0 & 1 \\ 0 & 0 & 1 & | & 6 & -1 & -1 \end{bmatrix} \\ L_1 \leftrightarrow L_1 - 2L_3 \\ \begin{bmatrix} 1 & 0 & 0 & | & -11 & 2 & 2 \\ 0 & 1 & 0 & | & -4 & 0 & 1 \\ 0 & 0 & 1 & | & 6 & -1 & -1 \end{bmatrix} \end{array}$$

$$\text{Como } \tilde{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ então } A^{-1} = \begin{bmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{bmatrix}.$$

Teorema 3.61 (Teorema de Binet). ¹ *Sejam $A, B \in \mathcal{M}_n(\mathbb{R})$, então*

$$\det(AB) = \det(A) \det(B).$$

Demonstração. Ver em (HEFEZ, 2012, Teorema 8.7, pág.228). □

Corolário 3.62. *Se $A \in \mathcal{M}_n(\mathbb{R})$ e $\det(A) \neq 0$, então A é inversível.*

Demonstração. Seja \tilde{A} a matriz linha escada reduzida equivalente a matriz A . Pelo Corolário 3.55, existem matrizes elementares E_1, \dots, E_s de ordem n tais que

$$E_s \cdots E_1 A = \tilde{A}.$$

Pelo Corolário 3.56, cada E_i é inversível, assim, multiplicando a esquerda, ambos os lados da igualdade por $E_1^{-1} \cdots E_s^{-1}$, tem-se

$$A = E_1^{-1} \cdots E_s^{-1} \tilde{A}.$$

Logo, pelo Teorema 3.61,

$$\det(A) = \det(E_1^{-1} \cdots E_s^{-1} \tilde{A}) = \det((E_1^{-1} \cdots E_s^{-1})(\tilde{A})).$$

Como $\det(A) \neq 0$ segue que $\det(\tilde{A}) \neq 0$. Então, pela Proposição 3.40, a matriz \tilde{A} não possui linha nula. Portanto, pela Observação 3.50, $\tilde{A} = I_n$. Assim, $A = E_1^{-1} \cdots E_s^{-1}$ e, pelo Teorema 3.57, A é inversível. □

Definição 3.63. Sejam A uma matriz de ordem $m \times n$ e a matriz \tilde{A} reduzida por linha à forma escada da matriz A . O *posto* de A , em símbolos $\text{posto}(A)$ ou p_A , é igual ao número de linhas não nulas de \tilde{A} . A *nulidade* de A , é igual a $n - p_A$.

Observação 3.64. As operações elementares e, a redução por linha de uma matriz à forma escada, foram definidas sobre as linhas da matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$. Porém, de modo análogo, podem ser definidas sobre as colunas da matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, de forma que, as proposições, teoremas e definições apresentadas nesta seção, continuam sendo válidas e são demonstradas de maneira análoga.

¹Jacques Philippe Marie Binet (França), 1786– 1856. Binet foi um dos precursores no estudo dos fundamentos da teoria matricial. Fonte:<http://www.ugr.es/> (Universidad de Granada).

Observação 3.65. As operações elementares foram definidas sobre as linhas de uma matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$. Porém, de modo análogo, podem ser definidas sobre as colunas desta matriz. Desta forma, obtém-se definições e resultados análogos aos apresentados nesta seção. Em particular, pode-se definir de forma similar o *posto de uma matriz* e, tem-se que o posto (linha) é igual ao posto (coluna). Como consequência, obtemos $n \geq p_A$ e, portanto, $n - p_A \geq 0$.

Teorema 3.66 (Teorema do Posto). *Seja $AX = B$ um sistema linear de m equações e n incógnitas. Sejam p_{AB} o posto da matriz ampliada do sistema e p_A o posto da matriz dos coeficientes do sistema. Então:*

- (1) *O sistema é compatível se, e somente se, $p_{AB} = p_A$;*
- (2) *O sistema é compatível e determinado se $p_{AB} = p_A = n$;*
- (3) *O sistema é compatível e indeterminado se $p_{AB} = p_A < n$. Neste caso, $n - p_A$ é o número de parâmetros livres do sistema, ou seja, parâmetros que podem assumir qualquer valor em \mathbb{R} .*

Demonstração. Seja $AX = B$ um sistema linear com m equações e n incógnitas. Seja $C = [A|B]_{m \times n}$ a matriz ampliada do sistema e $\tilde{C} = [\tilde{A}|\tilde{B}]_{m \times n}$ a forma linha escada reduzida de C . Denotaremos $\tilde{A} = [\tilde{a}_{ij}]_{m \times n}$ e $\tilde{B} = [\tilde{b}_{ij}]_{m \times n}$.

Como \tilde{A} é a forma escada reduzida de A e, sendo \tilde{A} um bloco de \tilde{C} , temos que $p_A = p_{\tilde{A}} \leq p_{(\tilde{C})} = p_{AB}$. Temos os seguintes casos a considerar:

$$p_A < p_{AB} \quad \text{ou} \quad p_A = p_{AB}.$$

CASO 1: $p_A < p_{AB}$. Neste caso, \tilde{C} tem uma linha do tipo

$$[0 \ 0 \ \dots \ 0 \ 1].$$

Portanto, o sistema $\tilde{A}X = \tilde{B}$ não possui solução e, como \tilde{C} é equivalente a C , pelo Teorema 3.48, o sistema $AX = B$ é incompatível.

CASO 2: $p_{AB} = p_A$. Neste caso, \tilde{C} e \tilde{A} têm o mesmo número de linhas não nulas. Como $p_A \leq n$, consideraremos os seguintes subcasos:

$$p_A = n \quad \text{ou} \quad p_A < n.$$

SUBCASO 2.1: $p_A = n$. Como $p_A \leq m$ segue que $n \leq m$. Se $n < m$ e, sendo \tilde{A} uma matriz com m linhas e n colunas na forma linha escada reduzida com $p_{\tilde{A}} = p_A = n$, tem-se que

$$\tilde{A} = \begin{bmatrix} I_n \\ O_{(m-n) \times n} \end{bmatrix}_{m \times n}$$

Como $p_A = p_{AB} = n$, segue que \tilde{B} é tal que $\tilde{b}_{n+1} = \dots = \tilde{b}_m = 0$. Portanto, $\tilde{A}X = \tilde{B}$ é compatível e determinado com a única solução $x_1 = \tilde{b}_1, \dots, x_n = \tilde{b}_n$. Se $n = m$ e $p_{\tilde{A}} = p_A = n$, então $\tilde{A} = I_n$. Logo, $\tilde{A}X = \tilde{B}$, também, é compatível e determinado com a única solução $x_1 = \tilde{b}_1, \dots, x_n = \tilde{b}_n$. Conseqüentemente, pelo Teorema 3.48, o sistema $AX = B$ é compatível e determinado.

SUBCASO 2.2: $p_A < n$. Coloquemos $p = p_A = p_{AB}$. Neste caso, \tilde{A} e \tilde{C} tem as primeiras p linhas não nulas: L_1, \dots, L_p , tais que o primeiro elemento não nulo de L_i está na coluna k_i e, assim, $k_1 < \dots < k_p$. Além disso, como $p = p_A < m$, temos que $p < m$ ou $p = m$. Se $p < m$ segue que $\tilde{b}_{p+1} = \dots = \tilde{b}_m = 0$. Então, a equação $\tilde{A}X = \tilde{B}$, se escreve como:

$$\begin{bmatrix} x_{k_1} + \tilde{a}_{1,k_1+1}x_{k_1+1} + \dots + \tilde{a}_{1n}x_n \\ x_{k_2} + \tilde{a}_{2,k_2+1}x_{k_2+1} + \dots + \tilde{a}_{2n}x_n \\ \vdots \\ x_{k_p} + \tilde{a}_{p,k_p+1}x_{k_p+1} + \dots + \tilde{a}_{pn}x_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

obtendo-se p equações lineares:

$$\begin{aligned} x_{k_1} + \tilde{a}_{1,k_1+1}x_{k_1+1} + \dots + \tilde{a}_{1n}x_n &= \tilde{b}_1, \\ x_{k_2} + \tilde{a}_{2,k_2+1}x_{k_2+1} + \dots + \tilde{a}_{2n}x_n &= \tilde{b}_2, \\ &\vdots \\ x_{k_p} + \tilde{a}_{p,k_p+1}x_{k_p+1} + \dots + \tilde{a}_{pn}x_n &= \tilde{b}_p. \end{aligned}$$

Como a matriz \tilde{A} esta na forma linha escada reduzida, temos que $\tilde{a}_{jk_i} = 0$, se $i > j$ para $j = 1, \dots, p-2$ e $\tilde{a}_{p-1,k_i} = 0$, se $i = k_p$. Estas igualdades, nos fornece o sistema de

equações

$$\begin{aligned}
 x_{k_1} &= - \sum_{j>k_1} \tilde{a}_{1j} x_j + \tilde{b}_1, & \text{onde } \tilde{a}_{1k_i} &= 0, \text{ se } i > 1; \\
 x_{k_2} &= - \sum_{j>k_2} \tilde{a}_{2j} x_j + \tilde{b}_2, & \text{onde } \tilde{a}_{2k_i} &= 0, \text{ se } i > 2; \\
 &\vdots \\
 x_{k_{p-1}} &= - \sum_{j>k_{p-1}} \tilde{a}_{p-1,j} x_j + \tilde{b}_{p-1}, & \text{onde } \tilde{a}_{p-1,k_i} &= 0, \text{ se } i = k_p; \\
 x_{k_p} &= - \sum_{j>k_p} \tilde{a}_{pj} x_j + \tilde{b}_p.
 \end{aligned}$$

Isto mostra que podemos escolher arbitrariamente valores para as incógnitas no conjunto

$$\{x_1, \dots, x_n\} \setminus \{x_{k_1}, \dots, x_{k_p}\} \quad (3.27)$$

e com esses determinar valores para x_{k_1}, \dots, x_{k_p} . Como o conjunto em 3.27 tem $n - p$ elementos, o sistema $\tilde{A}X = \tilde{B}$ tem $n - p$ incógnitas livres e, conseqüentemente, o mesmo ocorre para o sistema $AX = B$. Finalmente, o caso $p = m$ é similar ao visto anteriormente para o caso $p < m$. \square

3.4.2.1 Método de Gauss-Jordan

Ao reduzir por linha à forma escada a matriz ampliada de um dado sistema linear, pelo Teorema 3.51, existe uma única matriz reduzida por linha à forma escada equivalente a esta, ainda, pelo Teorema 3.48, encontramos um outro sistema linear com o mesmo conjunto solução do sistema dado, porém, com expressões mais simples. Este processo é chamado de *Método de Gauss Jordan*.

Exemplo 3.67. Seja o sistema

$$\begin{cases} x + 2z = 7 \\ 2x - y + 3z = 9 \\ 4x + y + 8z = 30 \end{cases}$$

tem-se a matriz ampliada C

$$C = [A|B] = \begin{bmatrix} 1 & 0 & 2 & 7 \\ 2 & -1 & 3 & 9 \\ 4 & 1 & 8 & 30 \end{bmatrix}.$$

Através de uma sequência de operações elementares sobre as linhas da matriz $C = [A|B]$, obtemos a matriz reduzida por linha à forma escada $\tilde{C} = [\tilde{A}|\tilde{B}]$.

$$\begin{array}{ccc} \begin{bmatrix} 1 & 0 & 2 & 7 \\ 2 & -1 & 3 & 9 \\ 4 & 1 & 8 & 30 \end{bmatrix} & L_2 \leftrightarrow \frac{-1}{2}L_2 + L_1 & \begin{bmatrix} 1 & 0 & 2 & 7 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{5}{2} \\ 4 & 1 & 8 & 30 \end{bmatrix} & L_3 \leftrightarrow \frac{-1}{4}L_3 + L_1 \\ \begin{bmatrix} 1 & 0 & 2 & 7 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{5}{2} \\ 0 & \frac{-1}{4} & 0 & \frac{-1}{2} \end{bmatrix} & L_3 \leftrightarrow 2L_3 + L_2 & \begin{bmatrix} 1 & 0 & 2 & 7 \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{5}{2} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{bmatrix} & L_2 \leftrightarrow L_2 - L_3 \\ \begin{bmatrix} 1 & 0 & 2 & 7 \\ 0 & \frac{1}{2} & 0 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{bmatrix} & L_1 \leftrightarrow L_1 - 4L_3 & \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{bmatrix} & L_2 \leftrightarrow 2L_2 \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{bmatrix} & L_3 \leftrightarrow 2L_3 & \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix} & \end{array}$$

Pelo método de Gauss-Jordan, tem-se $x = 1$, $y = 2$ e $z = 3$, portanto, o conjunto solução do sistema linear é $\{(1, 2, 3)\}$.

Quanto as soluções do sistema linear, pelo Teorema do Posto, sabemos se um dado sistema linear é compatível e determinado, compatível e indeterminado ou incompatível. Tomemos como exemplo, o sistema linear dado no Exemplo 3.67, onde a matriz ampliada reduzida por linha à forma escada é

$$\tilde{C} = [\tilde{A}|\tilde{B}] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}.$$

Segue que $p_A = p_{AB} = 3 = n$, onde n é o número de incógnitas, logo, pelo Teorema do Posto, o sistema linear é compatível e determinado. De fato, como apresentado no Exemplo 3.67, o conjunto solução do sistema linear é $\{(1, 2, 3)\}$.

Exemplo 3.68. Seja o sistema linear

$$\begin{cases} x + 2z + 3z = 15 \\ 3x + 3y = 6 \\ 6x + 6y = 12 \end{cases} \quad (3.28)$$

tem-se a matriz ampliada C

$$C = [A|B] = \begin{bmatrix} 1 & 2 & 3 & 15 \\ 3 & 3 & 0 & 6 \\ 6 & 6 & 0 & 12 \end{bmatrix}.$$

Através de uma sequência de operações elementares sobre as linhas da matriz $C = [A|B]$, obtemos a matriz reduzida por linha à forma escada $\tilde{C} = [\tilde{A}|\tilde{B}]$.

$$\begin{array}{ccc} \begin{bmatrix} 1 & 2 & 3 & 15 \\ 3 & 3 & 0 & 6 \\ 6 & 6 & 0 & 12 \end{bmatrix} & L_2 \leftrightarrow \frac{-1}{3}L_2 + L_1 & \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 6 & 6 & 0 & 12 \end{bmatrix} & L_3 \leftrightarrow \frac{-1}{6}L_3 + L_1 \\ \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 0 & 1 & 3 & 13 \end{bmatrix} & L_3 \leftrightarrow -L_3 + L_2 & \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & 0 \end{bmatrix} & L_1 \leftrightarrow L_1 - 2L_2 \\ \begin{bmatrix} 1 & 0 & -3 & -11 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & 0 \end{bmatrix} & & & \end{array}$$

Como $\tilde{C} = [\tilde{A}|\tilde{B}] = \begin{bmatrix} 1 & 0 & -3 & -11 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, então $p_A = p_{AB} = 2 < 3 = n$, onde n é o número de incógnitas, logo, pelo Teorema do Posto, o sistema linear é compatível e indeterminado e possui $n - p_A = 3 - 2 = 1$ parâmetro livre.

Para determinar o conjunto solução do sistema linear (3.28), pelo método de

Gauss-Jordan, tomemos o sistema linear equivalente

$$\begin{cases} x - 3z = -11 \\ y + 3z = 13. \end{cases} \quad (3.29)$$

Como existe um parâmetro livre, escolhamos $z = t$ com $t \in \mathbb{R}$. Substituindo no sistema linear (3.29), obtemos $y = -3t + 13$ e $x = 3t - 11$.

Portanto, o conjunto solução do sistema linear (3.28) é $\{(3t - 11, -3t + 13, t); t \in \mathbb{R}\}$.

Exemplo 3.69. Seja o sistema linear

$$\begin{cases} x + 2z + 3z = 15 \\ 3x + 3y = 6 \\ 6x + 6y = 13 \end{cases} \quad (3.30)$$

tem-se a matriz ampliada C

$$C = [A|B] = \begin{bmatrix} 1 & 2 & 3 & 15 \\ 3 & 3 & 0 & 6 \\ 6 & 6 & 0 & 13 \end{bmatrix}.$$

Através de uma sequência de operações elementares sobre as linhas da matriz $C = [A|B]$, obtemos a matriz reduzida por linha à forma escada $\tilde{C} = [\tilde{A}|\tilde{B}]$.

$$\begin{array}{ccc} \begin{bmatrix} 1 & 2 & 3 & 15 \\ 3 & 3 & 0 & 6 \\ 6 & 6 & 0 & 13 \end{bmatrix} & L_2 \leftrightarrow \frac{-1}{3}L_2 + L_1 & \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 6 & 6 & 0 & 13 \end{bmatrix} & L_3 \leftrightarrow \frac{-1}{6}L_3 + L_1 \\ \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 0 & 1 & 3 & \frac{77}{6} \end{bmatrix} & L_3 \leftrightarrow -L_3 + L_2 & \begin{bmatrix} 1 & 2 & 3 & 15 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} & L_1 \leftrightarrow L_1 - 2L_2 \end{array}$$

$$\begin{array}{ccc}
\begin{bmatrix} 1 & 0 & -3 & -11 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} & L_3 \leftrightarrow 6L_3 & \begin{bmatrix} 1 & 0 & -3 & -11 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 3 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix} & L_2 \leftrightarrow L_2 - 13L_3 & \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{array} \quad L_1 \leftrightarrow L_1 + 11L_3$$

Como $\tilde{C} = [\tilde{A}|\tilde{B}] = \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, então $p_A = 2 < 3 = p_{AB}$, logo, pelo Teorema do Posto, o sistema linear é incompatível, ou seja, não possui solução.

3.4.2.2 Regra de Cramer

Teorema 3.70. *Seja A uma matriz quadrada de ordem n . Se A é inversível então $X_0 = A^{-1}B$ é a única solução da equação matricial $AX = B$.*

Demonstração. Como

$$A(A^{-1}B) = (A^{-1}A)B = I_n B = B,$$

segue que $X_0 = A^{-1}B$ é solução da equação matricial $AX = B$. Provemos que esta solução é única, para isto, suponha que a equação matricial $AX = B$ tenha outra solução X_1 , isto é, $AX_1 = B$. Então,

$$X_1 = I_n X_1 = (A^{-1}A)X_1 = A^{-1}(AX_1) = A^{-1}B = X_0$$

Portanto, $X = A^{-1}B$ é a única solução do sistema $AX = B$. □

O resultado anterior é chamado de *Regra de Cramer*.

Exemplo 3.71. Consideremos o sistema

$$\begin{cases} x + 2z = 7 \\ 2x - y + 3z = 9 \\ 4x + y + 8z = 30. \end{cases} \quad (3.31)$$

O sistema linear é equivalente a equação matricial $AX = B$, onde

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 7 \\ 9 \\ 30 \end{bmatrix}.$$

No Exemplo 3.60, provamos que A é inversível e

$$A^{-1} = \begin{bmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{bmatrix}.$$

Então, pela Regra de Cramer,

$$X_0 = \begin{bmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \\ 30 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

é a única solução da equação matricial $AX = B$. Portanto, $\{(1, 2, 3)\}$ é o conjunto solução do sistema linear (3.31).

Caso $\det A = 0$, a Regra de Cramer é inconclusiva. De fato, nos exemplos 3.68 e 3.69, os sistemas lineares (3.28) e (3.30), possuem a mesma matriz dos coeficientes

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 3 & 0 \\ 6 & 6 & 0 \end{bmatrix},$$

cujo determinante é igual a zero. No entanto, o sistema (3.28) possui infinitas soluções e, o sistema (3.30), não possui solução.

Capítulo 4

Aplicação do FreeMat no estudo de funções, matrizes e sistemas lineares

Apoiado sob a concepção do uso dos computadores nas práticas docentes, são apresentadas três propostas para aplicação do FreeMat no ensino, sendo cada uma destas voltadas, respectivamente, para o ensino de funções, matrizes e sistemas lineares. Estas propostas, desdobram-se através de atividades aplicadas e que buscam levar ao discente novos conceitos não contemplados na grade curricular do ensino médio, tais como, posto de uma matriz, criptografia e programação computacional, conseqüentemente, ampliam o uso deste software para além da obtenção de resultados.

Quanto à aplicação das propostas em sala de aula, visto que, os conteúdos abordados na primeira proposta são apresentados ao aluno no primeiro ano do Ensino Médio e, os conteúdos da segunda e terceira propostas, são vistos pelo discente no segundo ano, foi aplicada apenas a primeira proposta, devido ao fato que os conteúdos trabalhados naquele período na instituição escolhida, não estarem em consonância com os abordados nas outras propostas.

4.1 Proposta I: Uma abordagem para o ensino e aplicação das funções

Esta proposta, divide-se em duas etapas: na primeira (Subseção 4.1.1), são descritas atividades sobre plotagens gráficas no FreeMat destinadas ao ensino e aplicações das funções. Na segunda etapa (Subseção 4.1.2), através da estrutura tarifária da CAB

(Concessionária dos Serviços Públicos de Abastecimento de Água e Esgotamento Sanitário de Cuiabá), busca-se delimitar uma função em que o valor a pagar esteja em função do consumo de água em m^3 e, construir um programa computacional no FreeMat, onde seja reportado o valor cobrado para um respectivo consumo.

4.1.1 Plotagem gráfica de uma função

A plotagem gráfica de função no FreeMat, pode ser feita somente se os seus domínios forem intervalos fechados. Assim, no caso de funções polinomiais, apenas plotará os gráficos das restrições destas, e as chamaremos como polinomiais.

Sejam $a, b \in \mathbb{R}$ com $a < b$. Para plotar o gráfico da função $f : [a, b] \rightarrow \mathbb{R}$, inicialmente, é necessário inserir o comando:

```
x=linspace (a,b)
```

Este comando refere-se aos valores de entrada para que o programa determine o domínio da função. Depois de inserido, o programa reportará alguns valores reais x pertencentes ao intervalo $[a, b]$.

Tendo o domínio da função, o próximo passo para que o programa plote o gráfico da função f é inserir a lei de correspondência $y = f(x)$. Por exemplo, ao inserir leis que determinem funções constantes e, funções polinomiais de grau 1, de grau 2 ou de grau 3, respectivamente, através da função pré-definida do FreeMat (função *anonymous*¹), deve-se escrever os comandos:

```
y= @(x) 0*x + s
```

função constante,

```
y= @(x) p*x + s
```

função polinomial de grau 1,

```
y= @(x) p*x.^2 + r*x + s
```

função polinomial de grau 2,

```
y= @(x) p*x.^3 + q*x.^2 + r*x + s
```

função polinomial de grau 3,

onde p, q, r e s são números reais dados com $p \neq 0$.

Finalmente, para fazer a plotagem, gráfica considera-se os seguintes casos:

¹A função *anonymous*, representada pelo comando $\@(x)$, permite declarar a variável independente x ao definir uma lei de correspondência.

Caso 1: Se f é dada por uma única sentença, deve-se inserir o comando:

```
plot(x,y(x))
```

Caso 2: Se f é uma função definida por partes, como segue:

$$f(x) = \begin{cases} f_1(x), & x \in [a, b_1], \\ f_2(x), & x \in (b_1, b_2], \\ \vdots & \\ f_n(x), & x \in (b_{n-1}, b_n]; \end{cases} \quad (4.1)$$

com $b_n = b$ e $n \geq 2$, deve-se fazer o mesmo procedimento que no Caso 1, com algumas restrições, pois, dentre algumas limitações do FreeMat, não é possível definir intervalos semiabertos ou abertos; desta maneira, ao definir cada intervalo do domínio $(b_{i-1}, b_i]$ com $2 \leq i \leq n$, sugere-se aproximar o intervalo esquerdo b_{i-1} por meio da soma de um número decimal suficientemente pequeno.

Por exemplo, neste caso, pode-se estabelecer intervalos fechados $[b_{i-1} + 0.1, b_i]$ e usar a variável x_i para indicar a variável $x \in [b_{i-1} + 0.1, b_i]$ e x_1 indicando a variável $x \in [a, b_1]$.

Assim, para plotar o gráfico da função definida por partes f , deve-se definir cada intervalo:

```
x1=linspace(a,b1)
```

 comando para definir o intervalo $[a, b_1]$.

```
xi=linspace(b_{i-1} + 0.1, bi)
```

 comando para definir os intervalos $[b_{i-1} + 0.1, b_i]$.

Após isso, para cada intervalo, definir as sentenças $y_1 = f_1(x)$ e $y_i = f_i(x)$, por fim, plota-se o gráfico inserindo o comando:

```
plot(x1, y1(x1), x2, y2(x2), ..., xn, yn(xn))
```

4.1.1.1 Plotagem gráfica de duas ou mais funções no mesmo plano cartesiano

Dadas as funções $f_i : [a, b] \rightarrow \mathbb{R}$ tal que $y_i = f_i(x)$, onde $1 \leq i \leq n$.

Para plotar os n gráficos no mesmo plano cartesiano, procede-se de modo análogo para a plotagem gráfica de uma função por partes. Inicialmente, inserir o domínio através do comando:

```
x=linspace(a,b)
```

Após, define-se a lei de correspondência $y_i = f_i(x)$ para cada função e plota-se o gráfico digitando o comando:

```
plot(x, y1(x), x, y2(x), ..., x, yn(x))
```

Interseção de funções: Ao plotar gráficos no mesmo plano cartesiano que apresentam interseções, para evidenciá-las, inicialmente, seleciona-se a opção *Sample*(Amostra) disposta no menu da janela gráfica, como mostra a Figura 4.1.

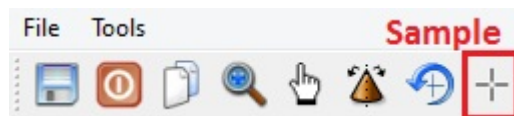


Figura 4.1: Menu da Janela gráfica

Selecionado a opção, clica-se sobre os pontos de interseções que se deseja evidenciar. Assim, os pares ordenados dos pontos serão apresentados.

4.1.1.2 Valor numérico da função

Dada a função $f : [a, b] \rightarrow \mathbb{R}$ com $a, b \in \mathbb{R}$ e $a < b$, seja $k \in [a, b]$, para determinar o valor numérico $f(k)$ da função, dividiremos em dois casos:

Caso 1: f é dada por uma única sentença.

Definir o domínio de f através do comando:

```
x= linspace(a,b)
```

e, após, definir a lei de correspondência $y = f(x)$. Com isto, para determinar o valor numérico $f(k)$, basta inserir o comando:

```
y(k)
```

Caso 2: f é uma função definida por partes, como na Equação 4.1.

Definir os intervalos do domínio de f , através dos comandos:

```
x1=linspace(a,b1)
x2=linspace(b1+0.1,b2)
xi=linspace(b2+0.1,b3)
⋮
xn=linspace(b(n-1)+0.1,bn)
```

Após, definir as respectivas sentenças:

$$y_1 = f(x_1)$$

$$y_2 = f(x_2)$$

⋮

$$y_n = f(x_n).$$

Com isto, para determinar o valor numérico $f(k)$, com $k \in (b_{i-1}, b_i]$, deve-se inserir o comando:

```
yi(k)
```

Ou, se $k \in [a, b_1]$, inserindo o comando:

```
y1(k)
```

Atividade 1. Dada a função $f : [-4, 4] \rightarrow \mathbb{R}$, definida por $f(x) = x^3 + x^2 + 1$, plotar seu gráfico no FreeMat.

Objetivo: Observar o comportamento de uma função.

Passo 1: Inserir o domínio da função.

O domínio da função é definido através do comando:

```
x=linspace (-4,4)
```

Após inserido o comando, é reportado alguns valores reais pertencentes ao domínio de f , como mostra a Figura 4.2.

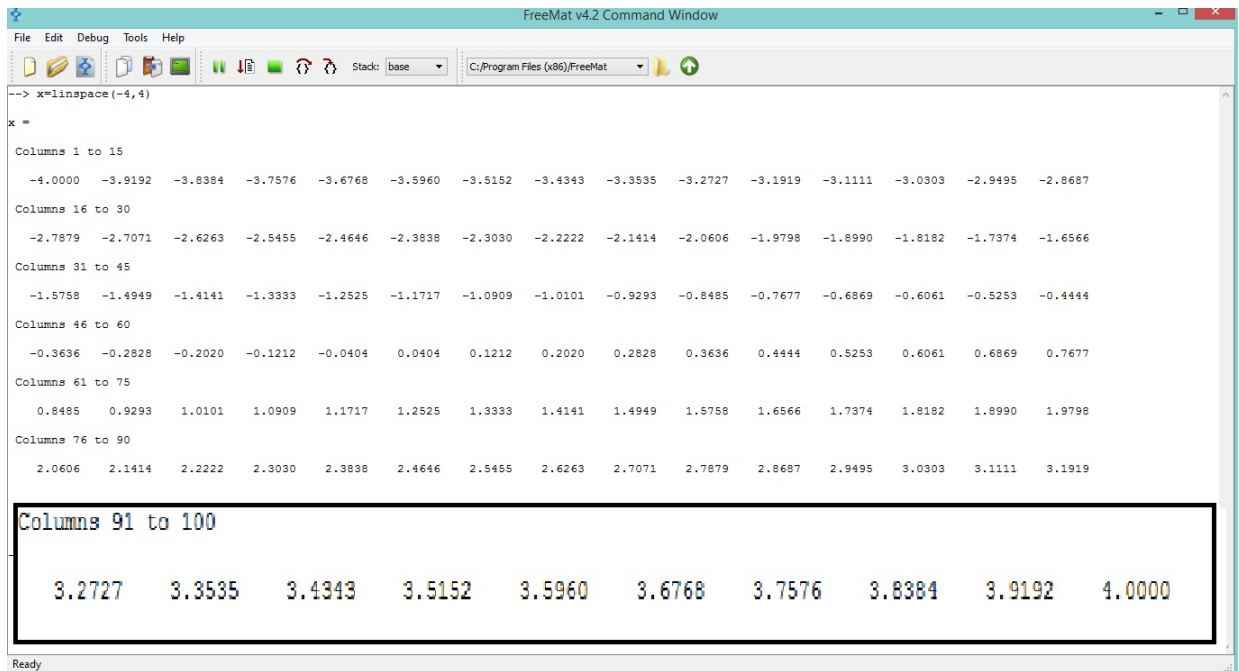


Figura 4.2: Interface do FreeMat após inserção do comando `x=linspace(-4,4)`

Passo 2: Inserir a lei de correspondência.

A lei que define a função é inserida através do comando:

$$y=@(x) \ x.^3 + x.^2 + 1$$

Passo 3: Plotar o gráfico da função.

O gráfico é plotado através da inserção do comando:

$$\text{plot}(x,y(x))$$

A Figura 4.3, mostra a execução dos passos 2 e 3 da Atividade 1 e, a Figura 4.4, apresenta o gráfico da função f plotado através do FreeMat.

```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Program Files (x86)/FreeMat
--> y=@(x) x.^3 + x.^2 + 1
y =
@(x) x.^3 + x.^2 + 1
--> plot(x,y(x))
```

Figura 4.3: Inserção de comandos para plotar o gráfico da função f

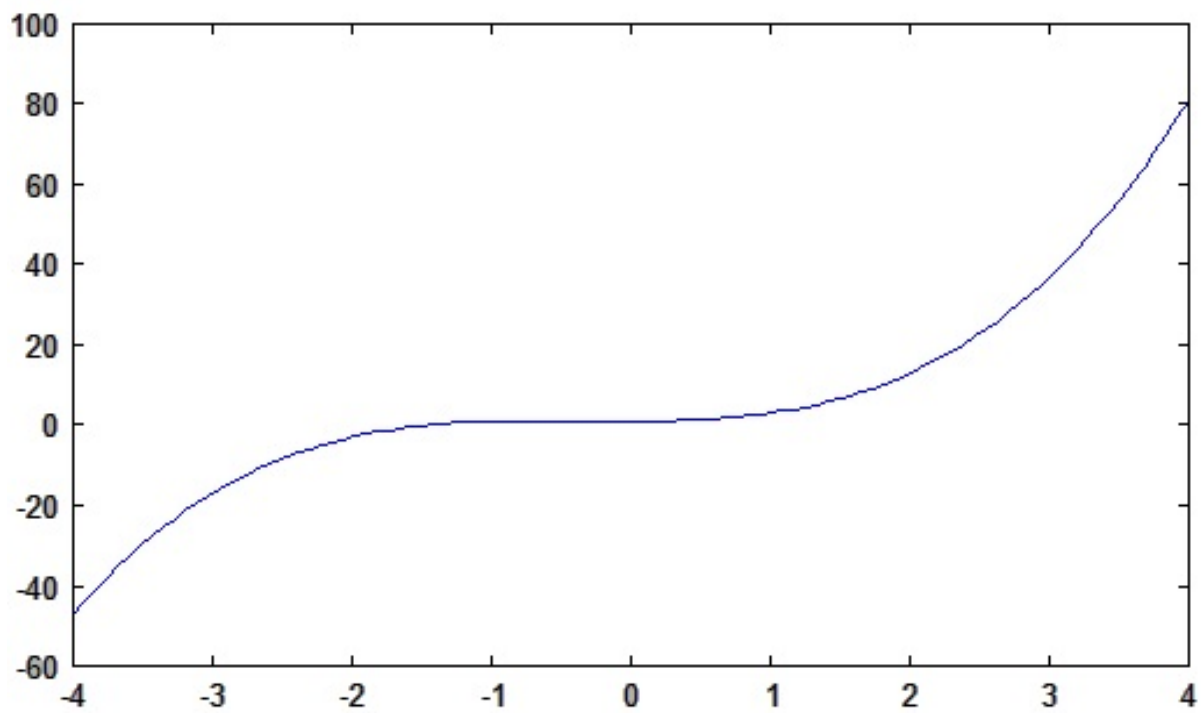


Figura 4.4: Gráfico da função $f : [-4, 4] \rightarrow \mathbb{R}$, definida por $f(x) = x^3 + x^2 + 1$

Atividade 2. Plote o gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = 10$.

Objetivo: Analisar o gráfico de uma função constante.

Passo 1: Inserir o domínio da função.

O domínio da função é definido através do comando:

```
x=linspace(-4,5)
```

Passo 2: Inserir a lei de correspondência

A lei que define a função é inserida através do comando:

```
y=@(x) 0*x + 10
```

Passo 3: Plotar o gráfico da função.

O gráfico é plotado através da inserção do comando:

```
plot(x,y(x))
```

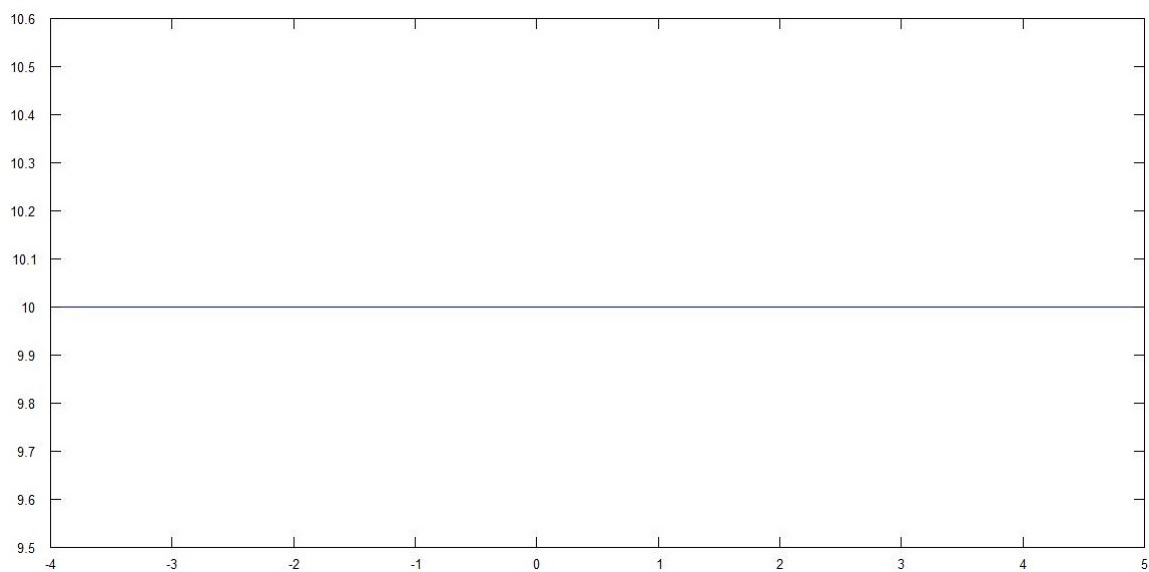


Figura 4.5: Gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = 10$

Atividade 3. Dada a função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$, faça o seguinte: (a) plote o gráfico de f ; (b) através do teste da reta horizontal, ilustre que a função não é injetiva; (c) evidencie os pontos de interseção da reta horizontal do item (b) com o gráfico de f .

Objetivo: Analisar o comportamento de uma função não injetiva através do gráfico.

(a) Plotagem gráfica de f .

Passo 1: Inserir o domínio.

O domínio da função é definido através do comando:

```
x=linspace(-4,5)
```

Passo 2: Inserir a lei de correspondência.

A lei que define a função é inserida através do comando:

```
f=@(x) x.^2+1
```

Passo 3: Plotar o gráfico da função f .

O gráfico é plotado através da inserção do comando:

```
plot(x,f(x))
```

O gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$ é apresentado na Figura 4.6.

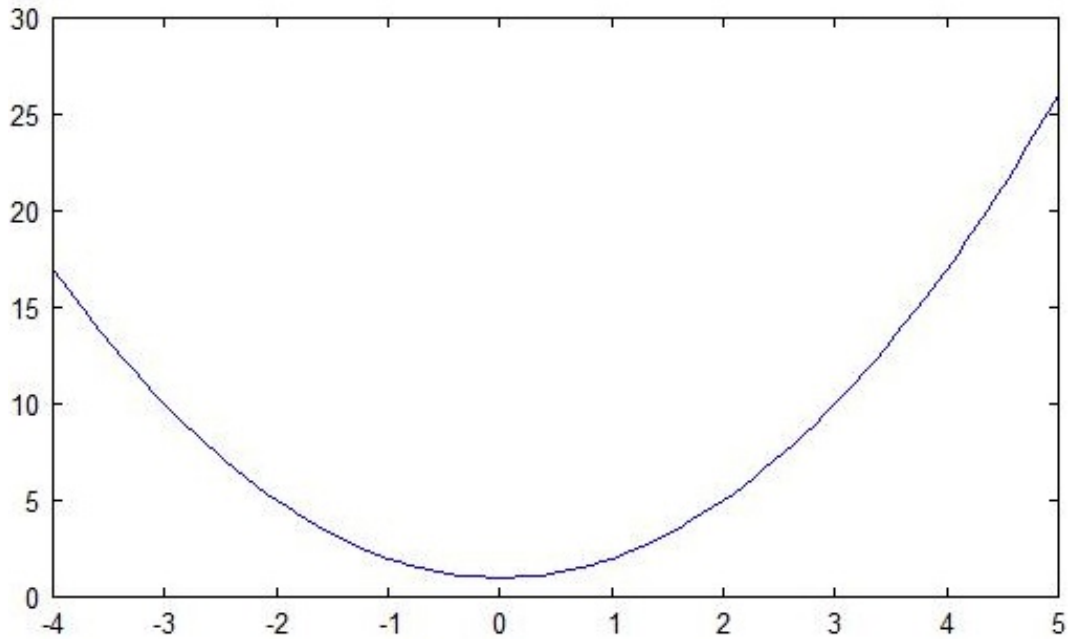


Figura 4.6: Gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$

(b) Ilustração que a função f não é injetiva.

Observação 4.1. Ao ilustrar que a função f não é injetiva, deve-se escolher uma reta horizontal que intersekte o gráfico de f em dois pontos. Escolhemos neste caso a reta $y = 10$ e, consideremos, o seu gráfico correspondente ao gráfico da função constante $g : [-4, 5] \rightarrow \mathbb{R}$, definida por $g(x) = 10$ (isto devido à limitação do domínio de f).

Passo 1: Inserir a lei de associação da função g .

A função g é definida através do comando:

```
g=@(x) 0*x + 10
```

Passo 2: Plotar o gráfico da função f e da função g no mesmo plano cartesiano.

Plota-se os gráficos inserindo o comando:

```
plot(x,f(x),x,g(x))
```

O gráfico da função $f : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$ e da função $g : [-4, 5] \rightarrow \mathbb{R}$, definida por $g(x) = 10$, plotados no mesmo plano cartesiano, são

apresentados na Figura 4.7.

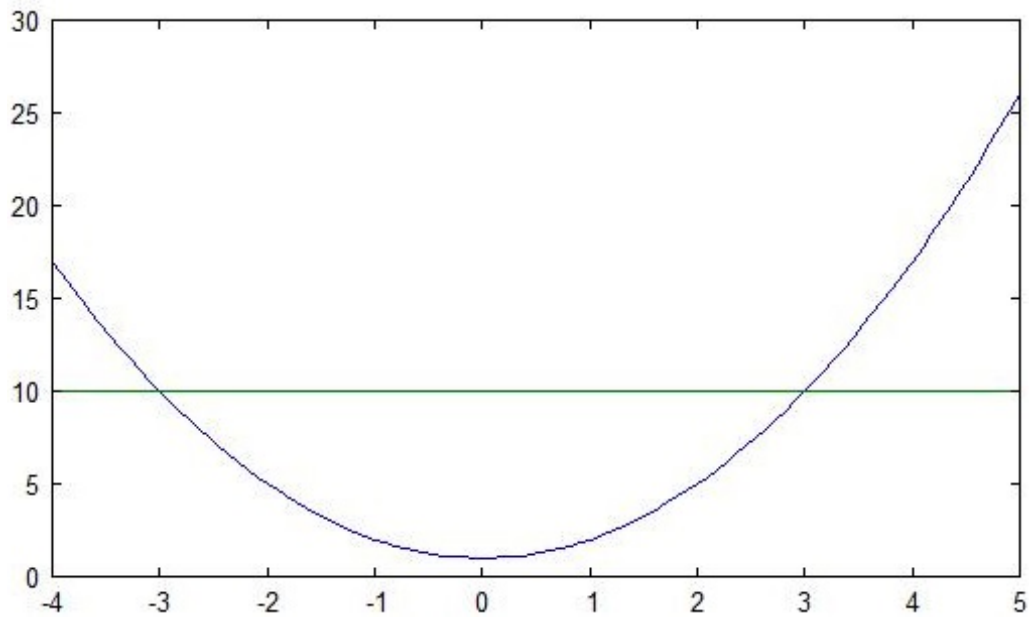


Figura 4.7: Gráficos das funções $f, g : [-4, 5] \rightarrow \mathbb{R}$, definida por $f(x) = x^2 + 1$ e $g(x) = 10$

(c) Evidenciando os pontos de interseção da reta horizontal com o gráfico de f .

Os pontos de interseção são evidenciados selecionando a opção *sample* e clicando sobre os pontos respectivos aos pares ordenados $(-3, 10)$ e $(3, 10)$, como mostra a Figura 4.8.

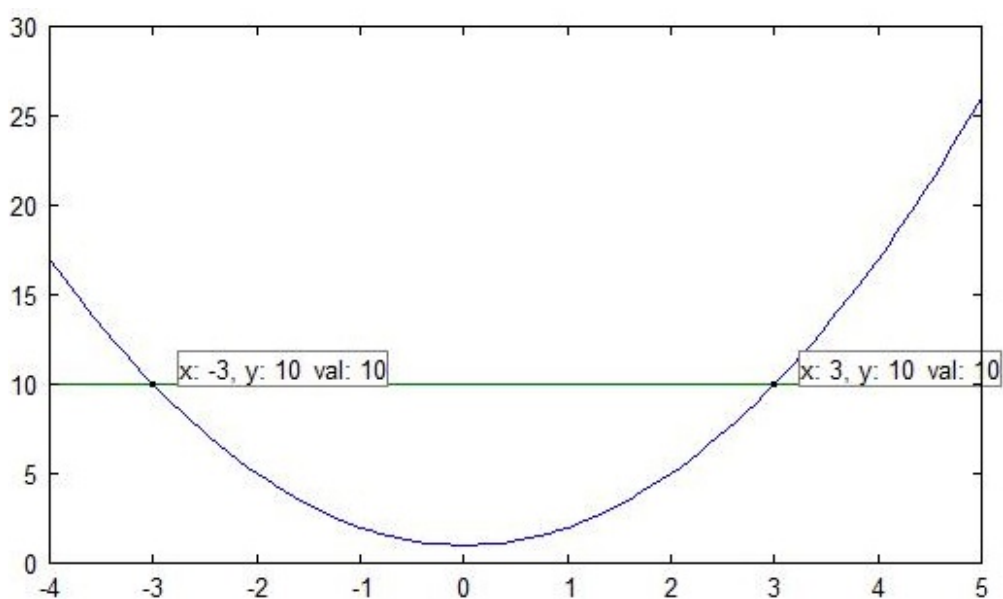


Figura 4.8: Ponto de interseção entre os gráficos das funções f e g

Atividade 4. Para alugar um carro certa pessoa dispõe de duas Locadoras, A e B . A locadora de automóveis A , cobra uma taxa de 20 reais mais 2 reais por quilômetro rodado, após a retirada do carro da garagem. Já, a locadora B , cobra uma taxa de 50 reais mais 1 real por quilômetro rodado. Utilizando o FreeMat, determine o valor cobrado, por cada locadora, para veículos que percorreram uma distância de 10 km e 40 km. Compare as locadoras, através da plotagem dos gráficos referentes as suas funções e, de acordo com os gráficos, onde o eixo das abcissas representa a distância a percorrer e o eixo das ordenadas o valor a pagar, determine em que momento a locadora B se torna mais viável que a locadora A ?

Objetivo: Identificar os efeitos dos coeficientes angular e linear de uma função afim na representação gráfica.

Observação 4.2. Como visto no Exemplo 3.3, as funções das locadoras A e B são, respectivamente, $f : [0, +\infty) \rightarrow \mathbb{R}$ tal que $f(x) = 2x + 20$ e $g : [0, +\infty) \rightarrow \mathbb{R}$ tal que $g(x) = x + 50$. Devido o comando `linspace` aceitar só intervalos fechados, será restringido o domínio dos gráficos das funções f e g . De acordo com o Exemplo 3.3, quando $x = 30$ km as funções igualam seus os valores, desta forma, buscando observar o ponto de interseção das funções, escolhe-se o domínio $[0, 50]$. Assim, considera-se $f : [0, 50] \rightarrow \mathbb{R}$, definida por $f(x) = 2x + 20$, para locadora A e $g : [0, 50] \rightarrow \mathbb{R}$, definida por $g(x) = x + 50$, para locadora B .

Passo 1: Definir o domínio de cada função.

O domínio das funções da locadora A e B é definido através do comando:

```
x= linspace(0,50)
```

Passo 2: Inserir a lei de correspondência para cada função.

Para definir a lei de correspondência da locadora A e B no FreeMat, digita-se, respectivamente, os comandos:

```
y1= @(x) 20 + 2*x
```

e

```
y2=@(x) 50 + 1*x
```

Passo 3: Calcular o valor numérico de cada função para $x = 10$ e $x = 40$.

Para a locadora A , os respectivos valores numéricos são calculados inserindo os comandos:

`y1(10)`

e

`y1(40)`

Para a locadora B , de maneira análoga, os respectivos valores numéricos são calculados inserindo os comandos:

`y2(10)`

e

`y2(40)`

A Figura 4.9, apresenta a execução dos passos 1,2 e 3 no FreeMat.

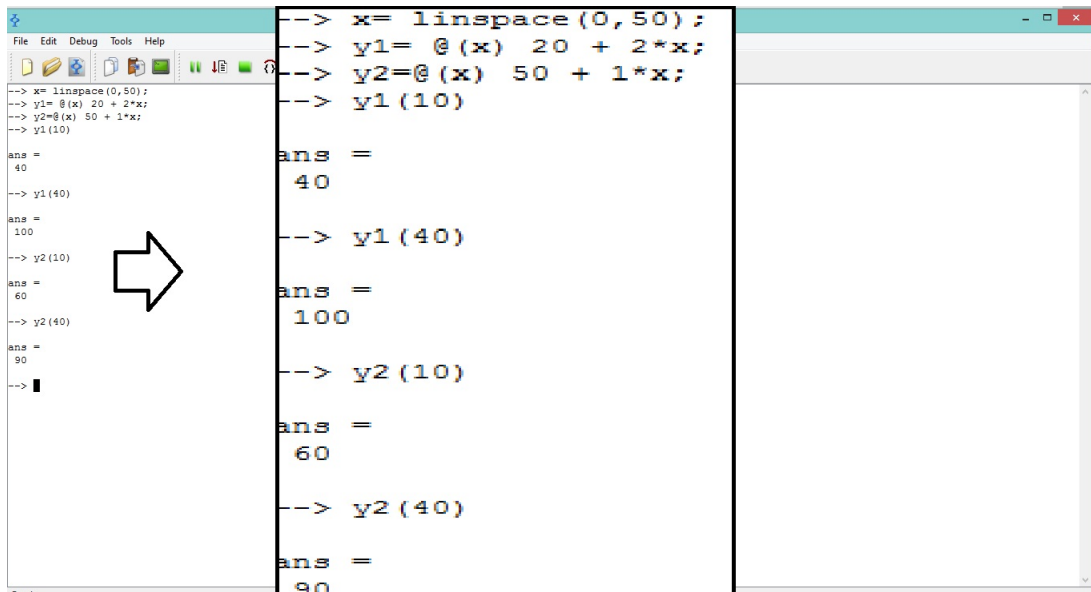


Figura 4.9: Comandos para obter os valores numéricos das funções das locadoras de automóveis A e B

55

Passo 4: Plotar os gráficos das funções das locadoras no mesmo plano cartesiano.

Os gráficos são plotados através do comando:

`plot (x,y1(x),x,y2(x))`

83

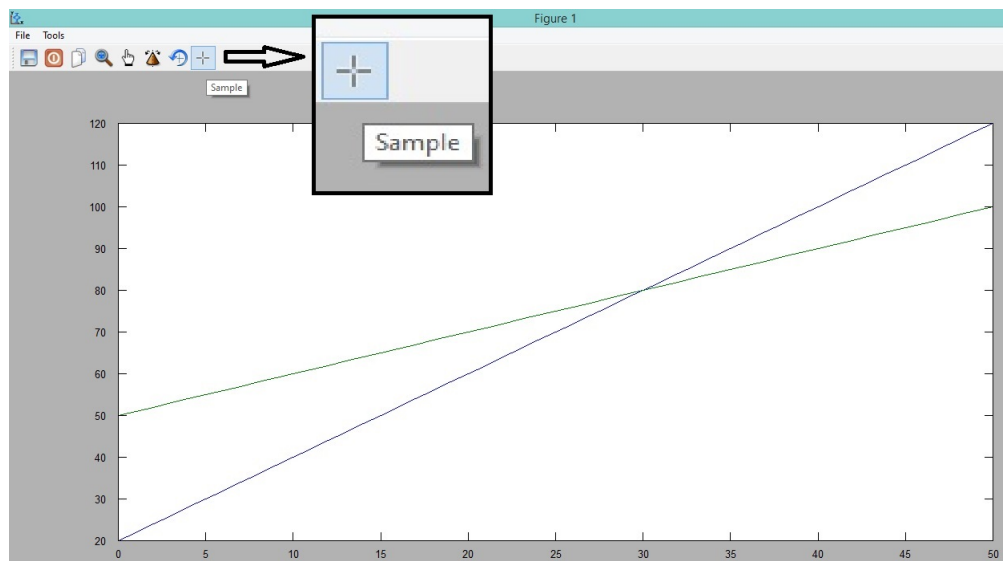


Figura 4.10: Janela gráfica e os gráficos das funções das locadoras A e B

Na Figura 4.10, são apresentados os gráficos das locadoras e, na Figura 4.11, apresenta-se o ponto de interseção entre os gráficos.

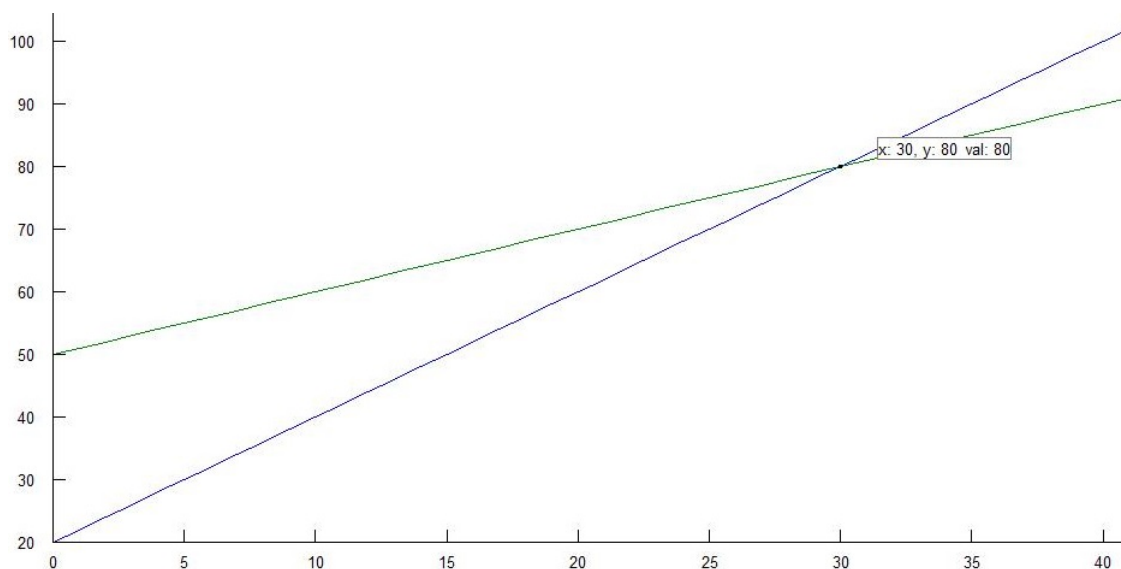


Figura 4.11: Gráficos e o ponto de interseção das funções das locadoras A e B

Passo 5: Evidenciar o ponto de interseção entre os gráficos das funções.

Selecionando a opção *Sample* (Amostra) no Menu de Ferramentas da janela gráfica e clicando no ponto de interseção do gráfico, evidencia-se o par ordenado $(30, 80)$.

Diante disto, conclui-se que a locadora B , em que o coeficiente angular é maior do que o da função da locadora A , passa a ser viável após 30 km.

4.1.2 Como calcular o custo por consumo em m³ de água?

Nesta etapa, são apresentadas atividades no qual propõe-se construir uma função definida por partes em que, o valor a pagar de uma conta de água esteja em função do consumo em m³ e, por meio de uma série de algoritmos, inserir comandos no FreeMat que reportem este valor quando inserido o consumo do mês.

A conta de água a ser utilizada é do município de Cuiabá/MT, referente ao ano de 2014. As faixas de consumo e os respectivos valores cobrados por metro cúbico da conta de água em questão são apresentadas na Figura 4.12, fornecida pela página virtual da CAB (Concessionária dos Serviços Públicos de Abastecimento de Água e Esgotamento Sanitário de Cuiabá) e, pode ser acessada pelo sítio virtual <http://www.cabcuiaba.com.br/InfGeral.aspx>:

A conta de água a ser utilizada é do município de Cuiabá/MT, referente ao ano de 2014. As faixas de consumo e os respectivos valores cobrados por metro cúbico da conta de água em questão são apresentadas na Figura 4.12, fornecida pela página virtual da CAB (Concessionária dos Serviços Públicos de Abastecimento de Água e Esgotamento Sanitário de Cuiabá) e, pode ser acessada pelo sítio virtual <http://www.cabcuiaba.com.br/InfGeral.aspx>:

ESTRUTURA TARIFÁRIA 2014				
Categoria	Tipo	Consumo (m ³ /mês)	Tarifa	
			Água (R\$/m ³)	Esgoto
1	Residencial Social	00 a 10	1,13	90%
2	Residencial	00 a 10	2,27	90%
		10,1 a 20	2,78	90%
		20,1 a 30	4,64	90%
		30,1 a 50	5,68	90%
		> 50	7,51	90%
3	Comercial	00 a 10	3,52	90%
		> 10	5,32	90%
4	Industrial	00 a 10	4,14	90%
		> 10	6,14	90%
5	Pública	00 a 10	4,43	90%
		> 10	7,26	90%

Nota explicativa: 1) Tarifa de esgoto = % da tarifa de água
2) Tarifa de água: consumo mínimo de 10m³/mês

Figura 4.12: Tabela de Estrutura tarifária e dos serviços complementares da CAB

Para o desenvolvimento das atividades, será utilizada a categoria 2 da conta de água, como é apresentado na Figura 4.13, sendo esta, referente aos valores cobrados pelo consumo residencial.

A cobrança para um certo consumo de água, de acordo com a estrutura tarifária apresentada na Figura 4.13, é feita da seguinte forma: para os primeiros 10 m³ de água

2	Residencial	00 a 10	2,27	90%
		10,1 a 20	2,78	90%
		20,1 a 30	4,64	90%
		30,1 a 50	5,68	90%
		> 50	7,51	90%

Figura 4.13: Faixas de consumo relativa a categoria 2 da conta de água da CAB

consumido, cobra-se R\$ 2,27 por m^3 mais um acréscimo de 90% sobre o valor, para os próximos $10 m^3$, cobra-se R\$ 2,78 por m^3 mais 90 % sobre o valor, assim sucessivamente, respeitando os intervalos da estrutura tarifária apresentada na Figura 4.13.

Exemplo 4.3. O valor a pagar pelo consumo de $15 m^3$ de água, é dado da seguinte maneira:

- Tarifa-se $10 m^3$ de acordo com o primeiro intervalo da estrutura tarifária, isto é,

$$2,27 \cdot (10) + 0,9 \cdot [2,27 \cdot (10)] = 43,13$$

- Tarifa-se $5 m^3$ de acordo com o segundo intervalo da estrutura tarifária, isto é,

$$2,78 \cdot (5) + 0,9 \cdot [2,78 \cdot (5)] = 26,41$$

- Soma-se os valores obtidos para determinar o valor a pagar pelo consumo de $15 m^3$. Logo, R\$ 69,54 é o valor a pagar por este consumo.

A continuação são apresentadas as atividades.

Atividade 5. Determinar uma lei de correspondência para cada faixa de consumo da categoria 2 da conta de água, onde, o valor a pagar y esteja em função do consumo x em m^3 . Após, expresse uma função f tal que $f(x) = y$.

Objetivo: Apresentar uma aplicação ao consumo de água através de uma função definida por partes.

Passo 1: Determine uma lei de correspondência para cada faixa de consumo.

Seja x_i o consumo em m^3 e y_i o valor a pagar em reais para uma faixa de consumo i , com $i \in \{1, 2, 3, 4, 5\}$.

Para a primeira faixa de consumo, considere $D_1 = \{x_1 \in \mathbb{R} : 0 \leq x_1 \leq 10\}$.

Como cada m^3 de água consumida custa R\$ 2,27 e ainda é acrescido 90% sobre o valor consumido, tem-se o valor a ser pago dado pela expressão:

$$y_1 = 2,27x_1 + 0,9.(2,27x_1) = 4,313x_1.$$

Para a segunda faixa de consumo, considere $D_2 = \{x_2 \in \mathbb{R} : 10 < x_2 \leq 20\}$. Como cada m^3 de água consumida custa R\$ 2,78 e ainda é acrescido 90% sobre o valor consumido, tem-se o valor a ser pago dado pela expressão:

$$y = 2,78x_2 + 0,9.(2,78x_2) = 5,282x_2.$$

Ao expressar a segunda faixa desta maneira, para os 10m^3 relativos ao primeiro intervalo, é cobrado R\$ 52,82 ao invés de R\$ 43,13, ou seja, é cobrado um valor a mais de $52,82 - 43,13 = \text{R\$ } 9,69$. Assim,

$$y_2 = 5,282x_2 - 9,69.$$

Determina-se as próximas expressões procedendo de maneira análoga ao método apresentado na segunda faixa de consumo.

Para a terceira faixa de consumo, considere $D_3 = \{x_3 \in \mathbb{R} : 20 < x_3 \leq 30\}$. Tem-se o valor a ser pago dado pela expressão:

$$y_3 = 8,816x_3 - 80,379.$$

Para a quarta faixa de consumo, considere $D_4 = \{x_4 \in \mathbb{R} : 30 < x_4 \leq 50\}$. Tem-se o valor a ser pago dado pela expressão:

$$y_4 = 10,792x_4 - 139,65.$$

Para a quinta faixa de consumo, considere $D_5 = \{x \in \mathbb{R} : x > 50\}$. Tem-se o valor a ser pago dado pela expressão:

$$y_5 = 14,269x_5 - 313,65.$$

Passo 2: Expressar uma função f tal que $f(x) = y$.

Observa-se que, $D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 = [0, +\infty)$. Portanto, pode-se expressar a função $f : [0, +\infty) \rightarrow \mathbb{R}$, como sendo

$$f(x) = \begin{cases} 4,313x, & \text{se } 0 \leq x \leq 10; \\ 5,282x - 9,69, & \text{se } 10 < x \leq 20; \\ 8,816x - 80,379, & \text{se } 20 < x \leq 30; \\ 10,792x - 139,65, & \text{se } 30 < x \leq 50; \\ 14,269x - 313,65, & \text{se } x > 50. \end{cases}$$

Atividade 6. Plote o gráfico da função f , obtida na Atividade 5.

Objetivo: Analisar o aumento de valores cobrados de acordo com cada faixa de consumo relativo a categoria 2 da conta de água.

Observação 4.4. Como citado na Seção 4.1.1, o FreeMat não permite definir intervalos semiabertos, com isto, o domínio da função f será restringido. Com o objetivo de analisar todas as faixas de consumo, visto que, consumos de água maiores que 100 m^3 nas residências é algo incomum. O domínio será dado por $[0, 100]$ e os subintervalos serão definidos como $[0, 10]$, $[10.1, 20]$, $[20.1, 30]$, $[30.1, 50]$ e $[50.1, 100]$.

Passo 1: Definir os subintervalos do domínio da função.

Os subintervalos do domínio serão definidos, respectivamente, através da inserção

dos seguintes comandos:

```
x1= linspace(0,10);  
x2= linspace(10.1,20);  
x3= linspace(20.1,30);  
x4= linspace(30.1,50);  
x5= linspace(50.1,100);
```

Passo 2: Inserir a lei de correspondência para cada intervalo do domínio.

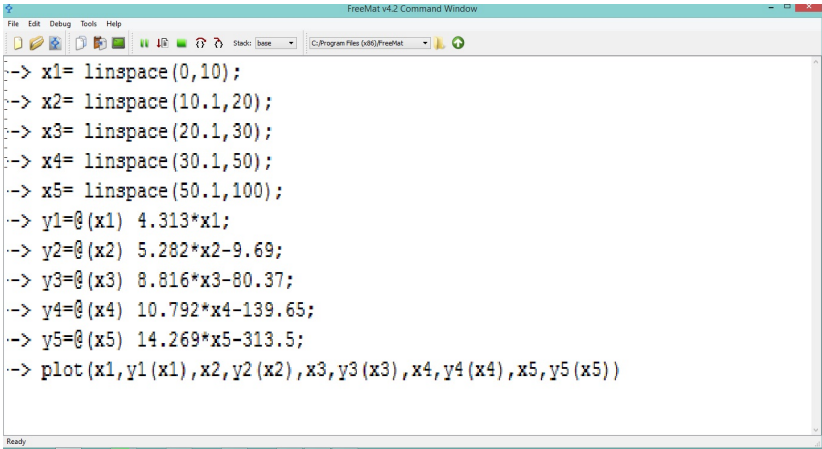
As expressões são inseridas no FreeMat através dos respectivos comandos:

```
y1=@(x1) 4.313*x1;  
y2=@(x2) 5.282*x2-9.69;  
y3=@(x3) 8.816*x3-80.37;  
y4=@(x4) 10.792*x4-139.65;  
y5=@(x5) 14.269*x5-313.5;
```

Passo 3: Plotar o gráfico da função f .

O gráfico é plotado com a inserção do comando:

```
plot(x1,y1(x1),x2,y2(x2),x3,y3(x3),x4,y4(x4),x5,y5(x5))
```



```
FreeMat v4.2 Command Window  
File Edit Debug Tools Help  
C:\Program Files (x86)\FreeMat  
--> x1= linspace(0,10);  
--> x2= linspace(10.1,20);  
--> x3= linspace(20.1,30);  
--> x4= linspace(30.1,50);  
--> x5= linspace(50.1,100);  
--> y1=@(x1) 4.313*x1;  
--> y2=@(x2) 5.282*x2-9.69;  
--> y3=@(x3) 8.816*x3-80.37;  
--> y4=@(x4) 10.792*x4-139.65;  
--> y5=@(x5) 14.269*x5-313.5;  
--> plot(x1,y1(x1),x2,y2(x2),x3,y3(x3),x4,y4(x4),x5,y5(x5))  
Ready
```

Figura 4.14: Inserção de comandos no FreeMat para desenvolvimento da Atividade 6

Na Figura 4.14, mostra-se a execução da Atividade 6 no FreeMat e, na Figura 4.15, é apresentado o gráfico da função definida por partes f , referente a categoria 2 da conta de água em questão.

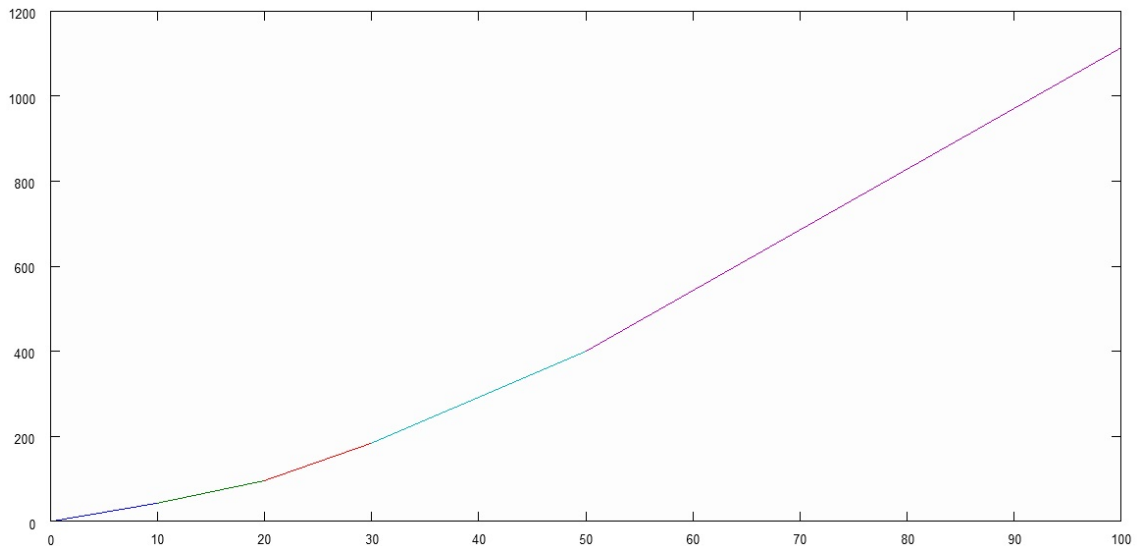


Figura 4.15: Gráfico da função f definida na Atividade 5

Atividade 7. Utilizando a estrutura tarifária da categoria 2, fornecido pela CAB (Figura 4.13) e a função definida na Atividade 5, construa um programa no FreeMat e o intitule de “VALORES”, o qual deve fazer o seguinte: ao inserir um valor correspondente ao consumo de água em m^3 de uma residência, deve ser reportado o valor a pagar em reais. Objetivo: Evidenciar a matemática envolvida na programação computacional.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘valores’ como o nome para o programa.

No menu do FreeMat, selecione a opção *editor*. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function VALORES
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda *não*, na linha

abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Criar um display com informações referente a aplicação do programa.

Cria-se o display, inserindo o comando:

```
disp('Programa para determinar o valor decorrente do consumo
de água das residências de Cuiabá')
```

Passo 4: Definir a entrada das variáveis independentes da função f .

Para seleccionar a entrada de valores correspondentes ao consumo em m^3 , digita-se o comando:

```
x=input('Insira o consumo em  $m^3$ ==>')
```

Passo 5: Definir as condições e parâmetros de decisão, de acordo com o consumo inserindo, para determinar o valor a pagar.

Passo 5.1: Inserir comando executando ações se o consumo é referente ao primeiro intervalo da função f .

```
if x>=0 & x<=10
consumo=x
y= 4.313*x;
disp('Valor a pagar em reais:')
valor=y
```

Passo 5.2: Inserir comando executando ações se o consumo é referente ao segundo intervalo da função f .

```
elseif x>10 & x<=20
consumo=x
y= 5.282*x-9.69;
disp('Valor a pagar em reais')
valor=y
```

Passo 5.3: Inserir comando executando ações se o consumo é referente ao terceiro intervalo da função f .

```
elseif x>20 & x<=30
consumo=x
y= 8.816*x-80.37;
disp('Valor a pagar em reais')
valor=y
```

Passo 5.4: Inserir comando executando ações se o consumo é referente ao quarto intervalo da função f .

```
elseif x>30 & x<=50
consumo=x
y= 10.792*x-139.65;
disp('Valor a pagar em reais')
valor=y
```

Passo 5.5: Inserir comando executando ações se o consumo é referente ao quinto intervalo da função f .

```
elseif x>50
consumo=x
y=14.269*x-313.5;
disp('Valor a pagar em reais')
valor=y
end
```

Passo 6: Definir uma pergunta, para que o usuário escolha se o programa deve ser re-inicializado, após, encerrar o programa.

```
continuar=input('Deseja obter o valor cobrado
para outro consumo? sim/nao');

disp ('* Fim do programa *')

end
```

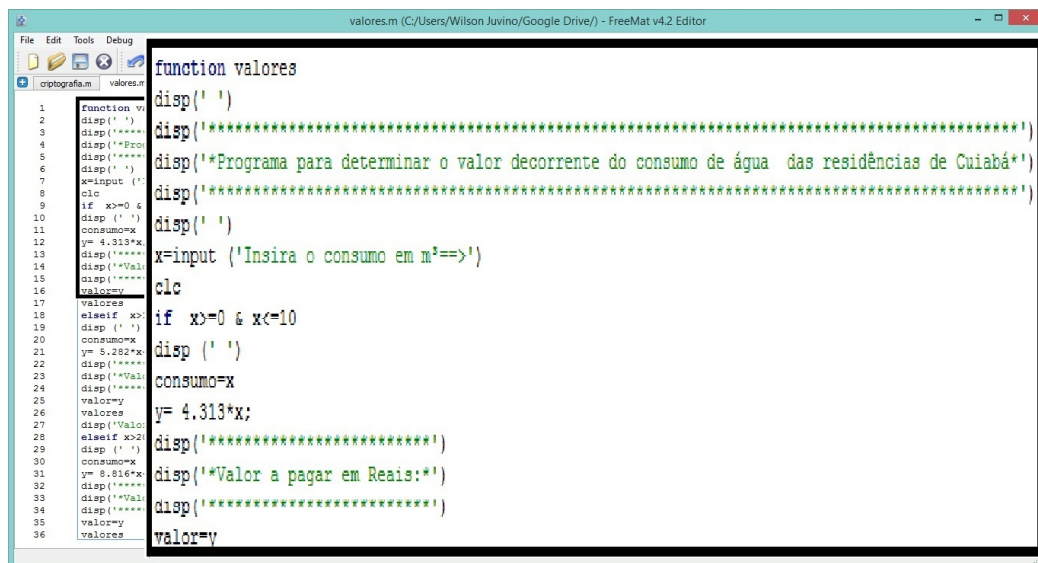


Figura 4.16: Editor do FreeMat e parte dos comandos relativos ao programa “VALORES”

Passo7: Salvar o programa.

Para salvá-lo, na barra do Menu do editor, selecione a opção *File* e clique em *Save*. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “VALORES” e salvo na extensão *.m*, assim o arquivo final será: VALORES.m.

Observação 4.5. O programa é apresentado integralmente no Apêndice A.1, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

Atividade 8. Utilize o programa, desenvolvido na Atividade 7, para determinar o valor cobrado para o consumo de 50 m^3 .

Objetivo: Apresentar a aplicabilidade e facilidade trazida pela programação computacional para desenvolvimento de problemas e tarefas matemáticas.

Passo 1: Abrir o programa ‘VALORES’.

Para abrir o programa, na janela de comando digita-se:

VALORES

Após, pressione **Enter** e o programa será compilado (ver Figura 4.17).

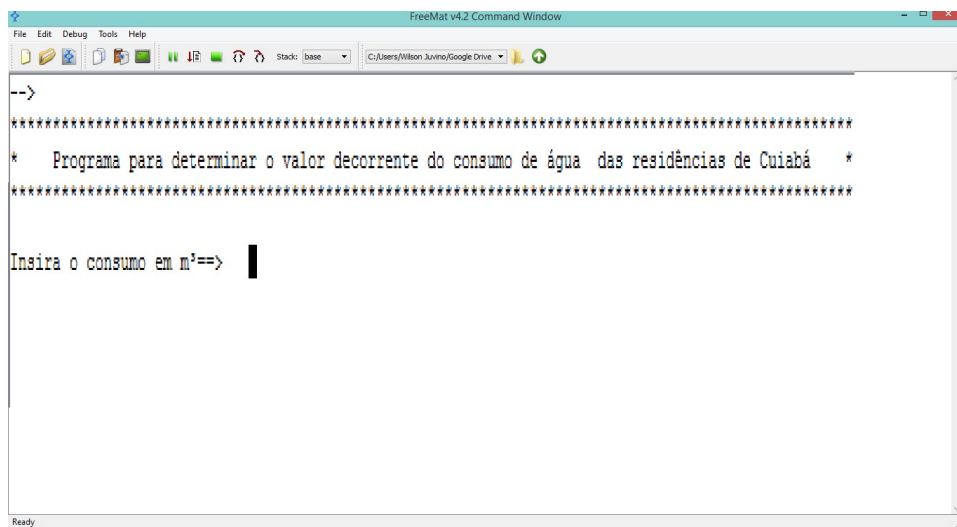


Figura 4.17: Compilando o programa “VALORES” no FreeMat: primeira etapa do programa

Passo 2: Inserir o consumo de 50 m³.

O programa solicitará a inserção de um consumo em m³, para inserir o consumo de 50 m³, digita-se:

50

e, ao pressionar **Enter**, o programa reportará o valor cobrado, como mostra a Figura 4.18.

Passo 3: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.17, caso contrário, se inserir , o programa é finalizado, sendo apresentado o display da Figura 4.19.

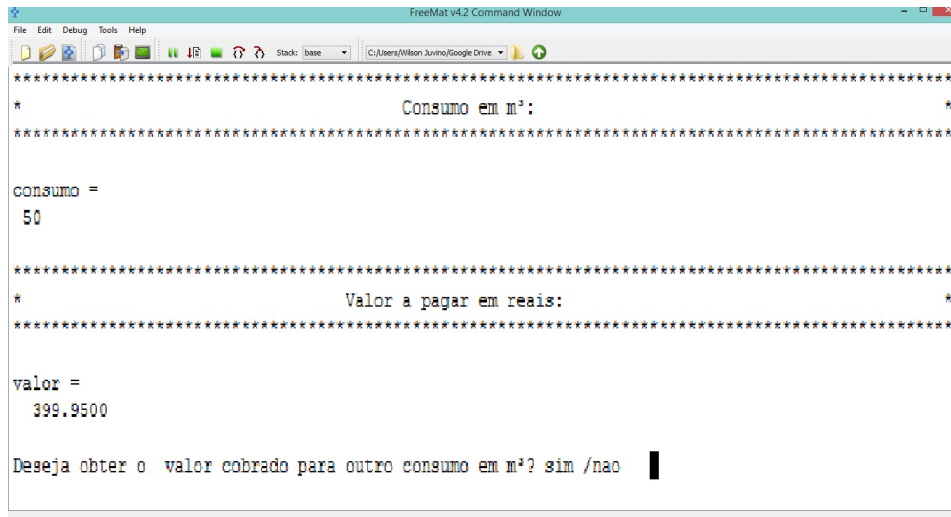


Figura 4.18: Compilando o programa “VALORES” no FreeMat: segunda etapa do programa

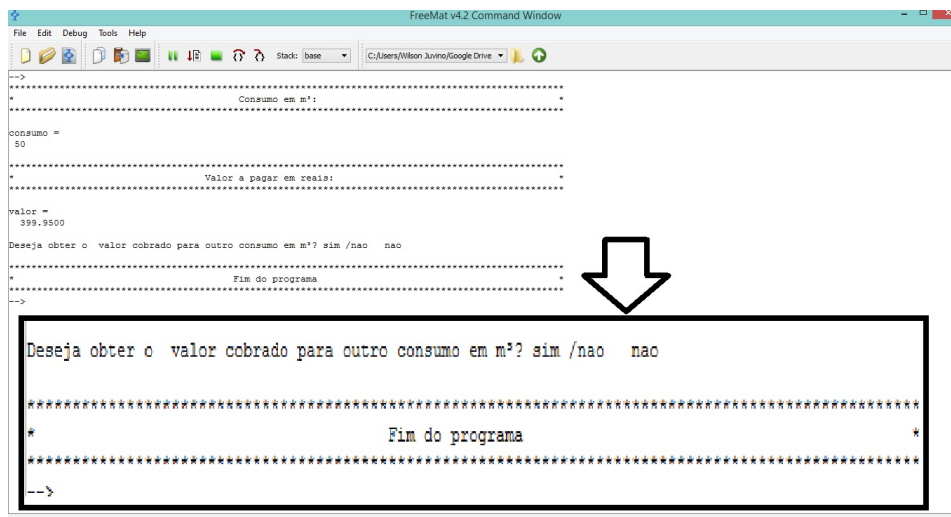


Figura 4.19: Compilando o programa “VALORES” no FreeMat: terceira etapa do programa

4.1.3 Relato da aplicação da proposta no Ensino Médio

As abordagens para o ensino e aplicação das funções através do FreeMat foram desenvolvidas no Instituto Federal de Mato Grosso, Campus São Vicente, localizado na BR-163/364, zona rural do município de Santo Antônio do Leverger/MT. A instituição oferta cursos técnicos e superiores, sendo estes, Ensino Médio Técnico em Agropecuária, Bacharelado em Agronomia, Bacharelado em Zootecnia, Tecnólogo em Análise e Desenvolvimento de Sistemas, e, Licenciatura em Ciências da Natureza. O curso Técnico em Agropecuária é integrado ao Ensino Médio, sendo, ofertado em modalidade seriado

de período integral e organizado em três anos. A escola dispõe de diversos laboratórios climatizados (informática, processamento de carne, matemática, biologia, química, psicologia, etc), 2 auditórios, biblioteca, quadra de esporte coberta, alojamentos, restaurante e salas de aulas climatizadas. Os estudantes matriculados no ensino médio moram nos alojamentos da escola e apresentam idade escolar entre 14 e 16 anos.

Das 5 turmas de primeiro ano do Ensino Médio Técnico em Agropecuária, foi escolhida a “Turma B”, para aplicar as abordagens. Estas, foram aplicadas durante os dias: 21 de outubro, 4 e 11 de novembro, 2 e 14 de dezembro do ano de 2014. Durante o desenvolvimento da proposta, a turma apresentava trinta e três discentes matriculados em período integral. Ressalta-se que, aulas sobre funções já haviam sido ministradas para estes discentes no período anterior à aplicação da proposta e não havia sido utilizado nenhum recurso computacional. As aulas foram aplicadas no laboratório de informática da instituição que dispunha de 22 computadores, porém, apenas 16 estavam funcionando corretamente. Devido a quantidade de computadores os discentes se organizaram em pares e, para iniciar as aplicações, cada par fez download do programa na internet e o instalou. A Figura 4.20, é uma foto tirada durante a aula ministrada no dia 4 de novembro de 2014.

Durante o desenvolvimento da Subseção 4.1.1, percebeu-se que, os passos tomados para a plotagem gráfica de uma função através do FreeMat, trouxeram ao discente facilidade em compreender e fixar a definição de uma função, onde, visualizaram-na como a terna: domínio, contradomínio e lei de correspondência. Os discentes também demonstraram facilidades ao plotarem os gráficos das funções apresentadas na proposta, em decorrência disto, plotaram outros gráficos de funções polinomiais com grau maior que 3. No final da aplicação destas atividades, quando comparado com aulas de esboço gráfico utilizando régua e quadro, destacou-se a grande quantidade de funções estudadas pelos alunos e o estudo de funções polinomiais de grau maior ou igual a 3, que não são apresentadas aos discentes do ensino básico.

Quanto a aplicação da Atividade 4 , onde se propõe delimitar a melhor oferta ao alugar um automóvel, destaca-se que, ao analisar a melhor oferta através do ponto de interseção dos gráficos, os discentes apresentaram dificuldades em identificar o gráfico de cada função das locadoras. Diante disto, para análises gráficas no mesmo plano cartesiano, observou-se ser necessário ensinar o comando que define a cor do gráfico de acordo com



Figura 4.20: Desenvolvimento da Proposta I com os alunos do IFMT no Laboratório de Informática

a lei estipulada, facilitando a identificação dos gráficos pertencentes ao mesmo plano cartesiano. Como não foi definido as cores do gráfico para cada função, alguns alunos não conseguiram identificar o respectivo gráfico plotado. Com a ajuda do professor, esta identificação foi feita depois de orientada a analogia entre o coeficiente linear da função e a interseção deste com o eixo das ordenadas. Construído e identificado cada gráfico, os discentes visualizaram facilmente o comportamento de cada função e explanaram qual seria a melhor escolha de acordo com o ponto de interseção do gráfico, sendo assim, alcançado o objetivo da atividade.

No desenvolvimento da Subseção 4.1.2, na Atividade 5, relata-se que, muitos alunos apresentaram dificuldades em construir uma lei de associação que caracteriza-se o valor a pagar em função da quantidade em m^3 de água consumida. Dentre estas dificuldades, a mais relevante, foi ao determinar as porcentagens e, apresentar a lei de correspondência para cada faixa de consumo, bem como, na interpretação das faixas de consumo e, das mudanças nos valores cobrados de acordo com os intervalos da categoria 2. Definidas as leis de correspondência, na Atividade 6, durante a inserção dos comandos das funções no FreeMat, os discentes não apresentaram dificuldades, compreendendo e executando perfeitamente a sequência de comandos inseridos para plotagem gráfica. O entusiasmo e facilidade na construção gráfica com o auxílio do FreeMat, implicou na facilidade em interpretar graficamente o comportamento da função de acordo com o consumo do mês. Durante o desenvolvimento destas atividades, ressalta-se, o bom rendimento dos discentes.

tes, quanto a utilização do FreeMat na inserção de comandos, para determinar o valor numérico e plotar o gráfico da função definida por partes. Pondera-se, as dificuldades dos discentes na interpretação e construção de problemas na linguagem matemática formal.

Na aplicação da Atividade 7, onde se propõe programar a conta de água, alguns alunos não conseguiram definir os comandos de condição e repetição, onde, foi necessário desenvolver no quadro os passos 4.1 e 4.2. Ao observarem o desenvolvimento destes passos no quadro e a função f definida na Atividade 6, o programa “VALORES” foi construído e aplicado facilmente. Ao final desta atividade, houve a percepção e comentários dos discentes quanto a matemática presente nas máquinas e nos programas utilizados para determinar valores no comércio, bem como, observaram a importância da matemática na vida social e no desenvolvimento da tecnologia, assim como almejado pela proposta.

Ao finalizar o desenvolvimento da Proposta I, observou-se, a aceitação e entusiasmo dos discentes quanto as aulas dentro da metodologia do uso do computador, também, poucas dificuldades encontradas no que diz respeito ao uso do software e na fixação dos conteúdos.

4.2 Proposta II: Ensino de matrizes utilizando o FreeMat, das operações com matrizes à criptografia.

Nesta seção, apresentaremos uma a proposta que visa reforçar o ensino das operações com matrizes no Ensino Médio. Também, mostra-se aplicações decorrentes do estudo de determinantes e das matrizes inversas. Na Subseção 4.2.1 são apresentadas atividades voltadas para o estudo das operações com matrizes, determinantes e matriz inversa. E, na Subseção 4.2.2, são descritas atividades voltadas para aplicação das matrizes na criptografia e desenvolvimento de um programa para criptografar e decifrar mensagens.

4.2.1 Ensino de matrizes

Para inserir um comando que defina uma matriz de ordem $m \times n$ no FreeMat, este deve ser disposto entre colchetes onde, os elementos de uma mesma linha estejam separados por um espaço, neste caso o comando terá $n - 1$ espaços, e as linhas devem estar separadas por ponto e vírgula, de forma que o comando tenha $m - 1$ pontos e vírgulas.

Assim, dada um matriz real

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

o comando para defini-la seria:

```
A=[a11 a12 ... a1n ; a21 a22 ... a2n; ... ;am1 am2 ... amn]
```

No Capítulo 2, foram apresentados comandos para determinar operações com matrizes e comandos para inserção de alguns tipos de matrizes, a qual serão utilizados, corriqueiramente, nesta seção.

A continuação, mostra-se alguns comandos no FreeMat, para calcular algumas operações com matrizes.

Determinante: Inserida a matriz A , para calcular o seu determinante, digita-se o comando:

```
det(A)
```

Matriz ampliada: Inseridas duas matrizes A e B , para determinar a matriz ampliada $C = [A|B]$, digita-se o comando:

```
C=[A B]
```

Matriz na forma escada: Para determinar a matriz escada reduzida por linha da matriz C , após inserida, digita-se o comando:

```
rref(C)
```

Posto da matriz: Para determinar o posto de uma matriz A no FreeMat, considera-se duas maneiras:

- (i) Inserir a matriz A e determinar o valor do posto através do comando:

```
rank(A)
```

(ii) Inserir a matriz A e obter a forma escada da matriz através do comando:

`rref(A)`

Matriz inversa: Para verificar se uma matriz A de ordem n admite inversa e determiná-la, também será considerada duas formas:

(i) *Passo 1* : Inserir a matriz A e verificar se a matriz admite inversa, para isto, calcula-se o determinante através do comando:

`det(A)`

Passo 2 : Se a matriz A admite inversa, determina-se a inversa através do comando:

`inv(A)`

(ii) *Passo 1* : Inserir a matriz A e verificar se a matriz admite inversa, para isto, calcula-se o posto através do comando:

`rank(A)`

Passo 2 : Se a matriz A admite inversa, para determiná-la, inserir a matriz ampliada $C = [A|I_n]$ através o comando:

`C=[A eye(n)]`

Passo 3 : Por fim, obter a matriz inversa através do comando:

`rref(C)`

Atividade 9. Dadas as matrizes:

$$A = \begin{bmatrix} 1 & 2 & 3 & 15 \\ 4 & 1 & 6 & -13 \\ 0 & 2 & -3 & 5 \\ 23 & 3 & 5 & -7 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 5 & 3 & 0 \\ 4 & -1 & 6 & 50 \\ 1 & 2 & -4 & 15 \\ -12 & 52 & -4 & 0 \end{bmatrix} \quad \text{e} \quad C = \begin{bmatrix} 0 & 5 & 2 & 0 \\ -4 & -1 & 6 & 5 \\ 10 & 12 & -3 & 1 \\ -2 & 2 & -4 & 1 \end{bmatrix},$$

pertencentes a $\mathcal{M}_4(\mathbb{R})$, calcule $(3A + C)B$.

Objetivo: Desenvolver as operações de adição, subtração, produto de matrizes e produto de um escalar pela matriz através do FreeMat.

Passo 1: Inserir as matrizes A, B e C no FreeMat.

Para inserir as matrizes A, B e C no FreeMat, deve-se digitar os respectivos comandos:

```
A=[1 2 3 15; 4 1 6 -13; 0 2 -3 5; 23 3 5 -7]
B=[5 5 3 0; 4 -1 6 50; 1 2 -4 15; -12 52 -4 0]
C=[0 5 2 0; -4 -1 6 5; 10 12 -3 1; -2 2 -4 1]
```

Passo 2: Calcular $(3A + C)B$.

Determina-se a matriz decorrente da operação $(3A + C)B$, inserindo o seguinte comando:

```
(3*A+C)*B
```

obtendo-se a matriz:

$$\begin{bmatrix} -470 & 2366 & -149 & 715 \\ -480 & -1682 & 76 & 460 \\ -82 & 840 & 122 & 720 \\ 630 & -694 & 303 & 715 \end{bmatrix}.$$

Na Figura 4.21, é apresentado o desenvolvimento da Atividade 9 no FreeMat.

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat
--> A=[1 2 3 15; 4 1 6 -13; 0 2 -3 5; 23 3 5 -7];
--> B=[5 5 3 0; 4 -1 6 50; 1 2 -4 15; -12 52 -4 0];
--> C=[0 5 2 0; -4 -1 6 5; 10 12 -3 1; -2 2 -4 1];
--> (3*A+C)*B

ans =
-470 2366 -149 715
 480 -1682 76 460
 -82 840 122 720
 630 -694 303 715

-->
Ready

```

Figura 4.21: Desenvolvimento da Atividade 9 no FreeMat

Atividade 10. Dada a matriz $C = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 4 & 1 & 6 & 5 & 3 \\ 0 & 2 & -3 & 5 & 1 \\ 4 & 1 & 8 & 5 & 1 \\ 7 & 5 & -3 & 3 & 1 \end{bmatrix}$, pertencente a $\mathcal{M}_5(\mathbb{R})$, calcule $\det(C)C^t$, onde C^t é a matriz transposta de C .

Objetivo: Assimilar os passos para calcular o determinante de uma matriz e a matriz transposta, bem como, operá-los através da utilização do FreeMat.

Passo 1: Definir a matriz C .

A matriz C é inserida através do comando:

$C=[1\ 2\ 3\ 0\ 0; 4\ 1\ 6\ 5\ 3; 0\ 2\ -3\ 5\ 1; 4\ 1\ 8\ 5\ 1; 7\ 5\ -3\ 3\ 1]$

Passo 2: Calcular $\det(C)C^t$.

Para calcular $\det(C)C^t$, deve-se digitar o comando:

$\det(C)*C'$

obtendo-se a matriz:

$$\begin{bmatrix} -1860 & -7440 & 0 & -7440 & -13020 \\ -3720 & -1860 & -3720 & -1860 & -9300 \\ -5580 & -11160 & 5580 & -14880 & 5580 \\ 0 & -9300 & -9300 & -9300 & -5580 \\ 0 & -5580 & -1860 & -1860 & -1860 \end{bmatrix}.$$

Na Figura 4.22, é apresentado o desenvolvimento da Atividade 10 no FreeMat.

```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat
--> C=[1 2 3 0 0; 4 1 6 5 3; 0 2 -3 5 1; 4 1 8 5 1; 7 5 -3 3 1];
--> det(C)*C'
ans =
-1860 -7440 0 -7440 -13020
-3720 -1860 -3720 -1860 -9300
-5580 -11160 5580 -14880 5580
0 -9300 -9300 -9300 -5580
0 -5580 -1860 -1860 -1860
--> █
Ready
```

Figura 4.22: Desenvolvimento da Atividade 10 no FreeMat

Atividade 11. Verifique se a matriz $A = \begin{bmatrix} 2 & 3 & 5 \\ 6 & 7 & 5 \\ 1 & 10 & 11 \end{bmatrix}$, pertencente a $\mathcal{M}_3(\mathbb{R})$, é inversa da matriz $B = \begin{bmatrix} 27/136 & 1/8 & -5/34 \\ -61/136 & 1/8 & 5/34 \\ 53/136 & -1/8 & -1/34 \end{bmatrix}$.

Objetivo: Fixar a definição de matriz inversa e desenvolvimento da multiplicação de matrizes no FreeMat.

Passo 1: Inserir a matriz A e B .

As matrizes A e B são definidas através dos seguintes comandos:

```
A=[2 3 5; 6 7 5; 1 10 11]
B=[27/136 1/8 -5/34; -61/136 1/8 5/34; 53/136 -1/8 -1/34]
```

Passo 2: Determinar os produtos AB e BA .

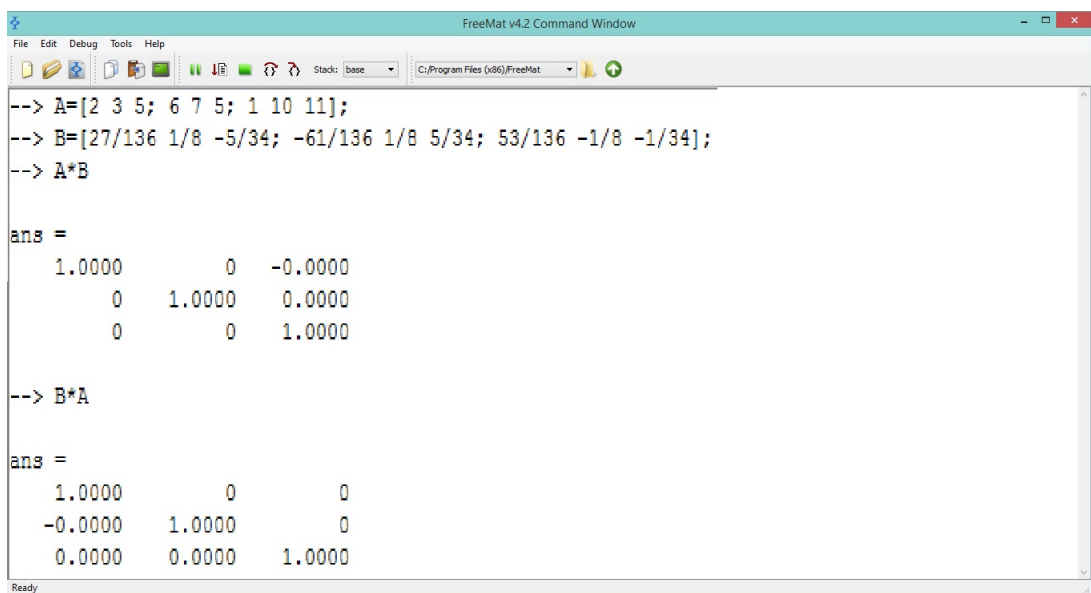
Determina-se os produtos através dos comandos:

```
A*B
B*A
```

Passo 3: Verificar os resultados reportados.

Os resultados reportados pelo FreeMat são iguais à matriz identidade. Logo, B é inversa de A .

Na Figura 4.23, é apresentado os resultados dos produtos AB e BA .



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
--> A=[2 3 5; 6 7 5; 1 10 11];
--> B=[27/136 1/8 -5/34; -61/136 1/8 5/34; 53/136 -1/8 -1/34];
--> A*B

ans =
  1.0000    0 -0.0000
    0  1.0000  0.0000
    0    0  1.0000

--> B*A

ans =
  1.0000    0    0
 -0.0000  1.0000  0
  0.0000  0.0000  1.0000
Ready
```

Figura 4.23: Desenvolvimento da Atividade 11 no FreeMat

Atividade 12. Utilize a primeira forma, dada na Subseção 4.2.1, para verificar se a matriz $A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix}$, pertencente a $\mathcal{M}_3(\mathbb{R})$, admite inversa e, se admitir, determine a matriz inversa B de A .

Objetivo: Apresentar uma aplicação dos determinantes ao verificar se uma matriz admite inversa e o comando para obtê-la.

Passo 1: Inserir a matriz A .

Insira a matriz A no FreeMat, através do comando:

```
A=[1 0 2 ; 2 -1 3; 4 1 8]
```

Passo 2: Verificar se a matriz A admite inversa. Para verificar se a matriz A admite inversa, calcule o determinante da matriz inserindo o comando:

```
det(A)
```

Como $\det A = 1 \neq 0$ a matriz admite inversa.

Passo 3: Determinar a matriz inversa.

Determina-se a matriz inversa B , inserindo o comando:

```
B=inv(A)
```

obtendo-se a matriz $B = \begin{bmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{bmatrix}$.

Na Figura 4.24, é apresentado o desenvolvimento da Atividade 12 no FreeMat.

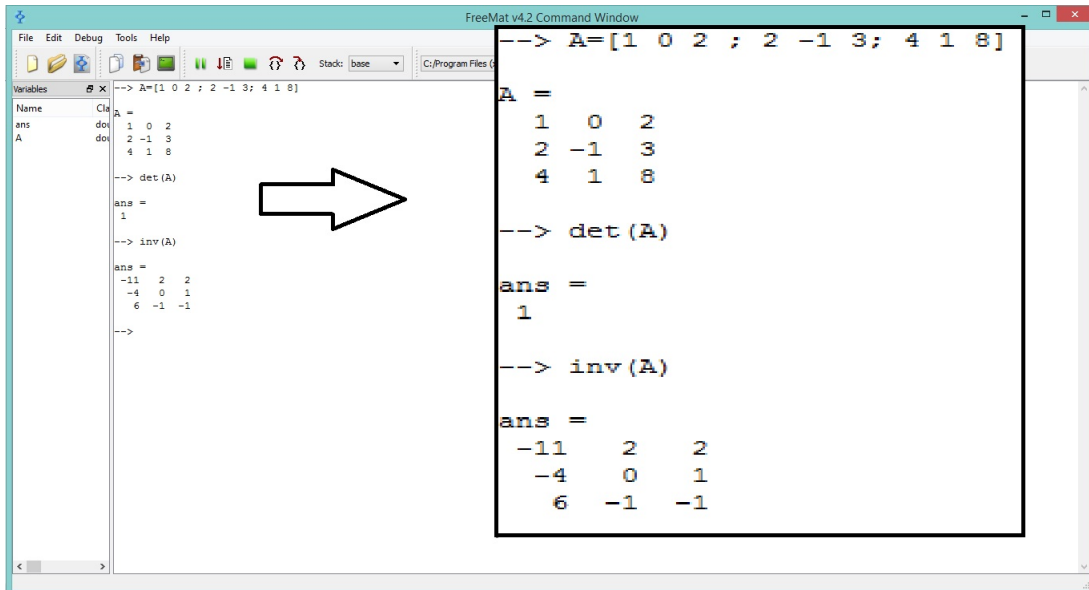


Figura 4.24: Desenvolvimento da Atividade 12 no FreeMat

Atividade 13. Utilize a segunda forma, dada na Subseção 4.2.1, para verificar se a matriz

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 6 & 7 & 5 \\ 1 & 10 & 11 \end{bmatrix},$$

pertencente a $\mathcal{M}_3(\mathbb{R})$, admite inversa e, se admitir, determine a matriz inversa B de A .

Objetivo: Obter a matriz inversa através da forma escada da matriz ampliada.

Passo 1: Inserir a matriz A .

Insira a matriz A no FreeMat, através do comando:

$$A = [2 \ 3 \ 5; 6 \ 7 \ 5; 1 \ 10 \ 11]$$

Passo 2: Verificar se a matriz A admite inversa.

Para verificar se a matriz A admite inversa, calcule o posto da matriz A inserindo o comando:

$$\text{rank}(A)$$

Como o posto(A) = 3, a matriz admite inversa.

Passo 3: Definir a matriz ampliada $C = [A|I_n]$.

A matriz é inserida no FreeMat através do comando:

$$C=[A \text{ eye}(3)]$$

Passo 4: Obter a forma escada reduzida da matriz C .

A matriz é obtida através da inserção do comando:

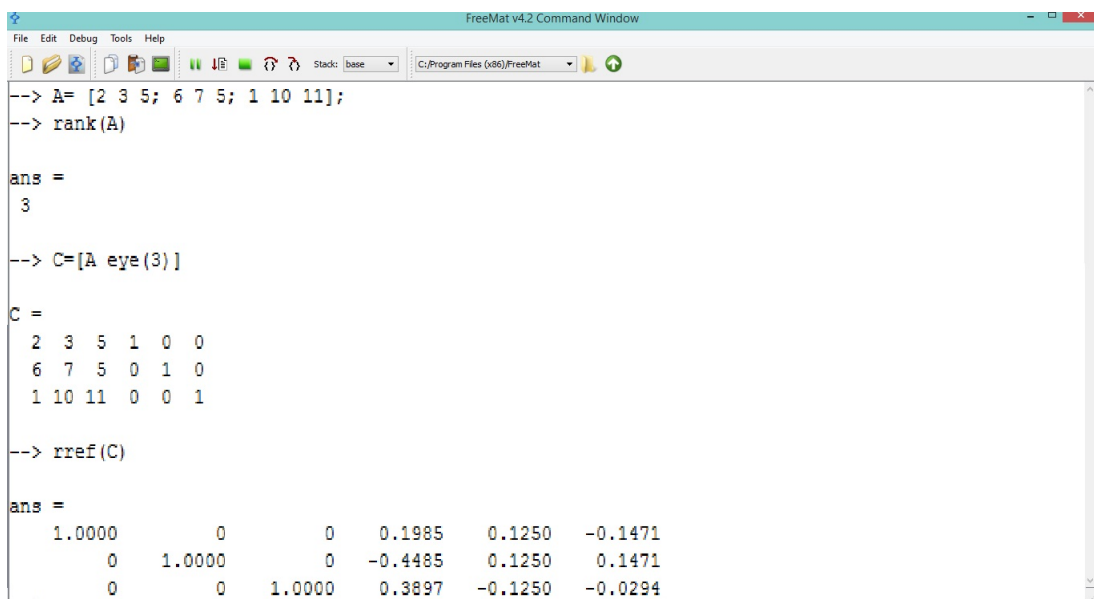
$$\text{rref}(C)$$

obtendo-se a matriz

$$C' = \begin{bmatrix} 1 & 0 & 0 & 0,1985 & 0,1250 & -0,1471 \\ 0 & 1 & 0 & -0,4485 & 0,1250 & 0,1471 \\ 0 & 0 & 1 & 0,3897 & -0,1250 & -0,0294 \end{bmatrix}$$

e, portanto a matriz inversa de A é a matriz

$$B = \begin{bmatrix} 0,1985 & 0,1250 & -0,1471 \\ -0,4485 & 0,1250 & 0,1471 \\ 0,3897 & -0,1250 & -0,0294 \end{bmatrix}.$$



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat
--> A=[2 3 5; 6 7 5; 1 10 11];
--> rank(A)
ans =
3
--> C=[A eye(3)]
C =
2 3 5 1 0 0
6 7 5 0 1 0
1 10 11 0 0 1
--> rref(C)
ans =
1.0000 0 0 0.1985 0.1250 -0.1471
0 1.0000 0 -0.4485 0.1250 0.1471
0 0 1.0000 0.3897 -0.1250 -0.0294
```

Figura 4.25: Desenvolvimento da Atividade 13 no FreeMat

Na Figura 4.25, é apresentado o desenvolvimento da Atividade 13 no FreeMat.

Observação 4.6. Na Atividade 11, foi verificado que a inversa da matriz A é a matriz:

$$\begin{bmatrix} 27/136 & 1/8 & -5/34 \\ -61/136 & 1/8 & 5/34 \\ 53/136 & -1/8 & -1/34 \end{bmatrix}.$$

Já, na Atividade 13, a inversa foi dada pela matriz:

$$\begin{bmatrix} 0,1985 & 0,1250 & -0,1471 \\ -0,4485 & 0,1250 & 0,1471 \\ 0,3897 & -0,1250 & -0,0294 \end{bmatrix}.$$

Embora sejam matrizes distintas, cada elemento da matriz possui seus valores aproximados por 4 casas decimais. Isto, deve-se ao fato, do FreeMat estar configurado para apresentar até 4 casas decimais, após a vírgula (`format short`). No entanto, é possível configurá-lo para reportar até 14 casas decimais, após a vírgula, digitando o comando (`format long`). Neste último formato, o programa irá apresentar um valor aproximado para este elemento com exatamente 14 casas decimais, após a vírgula.

Na Figura 4.26, é apresentado o desenvolvimento da Atividade 13 utilizando o comando `format long`.

Em particular, a matriz inversa da matriz A , apresentada na Atividade 13, será reportada da seguinte maneira:

$$\begin{bmatrix} 0,19852941176471 & 0,12500000000000 & -0,147105882352941 \\ -0,44852941176471 & 0,12500000000000 & 0,147105882352941 \\ 0,38970588235294 & -0,12500000000000 & -0,02941176470588 \end{bmatrix}.$$

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat

--> format long
--> A= [2 3 5; 6 7 5; 1 10 11];
--> C=[A eye(3)];
--> rref(C)

ans =
  1.0000000000000000      0      0  0.19852941176471  0.1250000000000000 -0.14705882352941
      0  1.0000000000000000      0 -0.44852941176471  0.1250000000000000  0.14705882352941
      0      0  1.0000000000000000  0.38970588235294 -0.1250000000000000 -0.02941176470588

-->
Ready

```

Figura 4.26: Desenvolvimento da Atividade 13 no FreeMat, utilizando o comando `format long`

4.2.2 Criptografando mensagens através do FreeMat

Todas as atividades apresentadas nesta subseção terão por objetivo criptografar mensagens no FreeMat, utilizando o sistema poligráfico apresentado na Subseção 3.3.2.

Atividade 14. Utilizando o FreeMat, insira comandos para corresponder cada letra do alfabeto dado na Subseção 3.2.2, de acordo com a Tabela 3.2.

As letra do alfabeto dado na Subseção 3.2.2, são correspondidas a um número inteiro no FreeMat, ao digitar, simultaneamente, os seguintes comandos entre ponto e vírgula:

```
a=1; b=2; c=3; d=4; e=5; f=6; g=7; h=8; i=9; j=10; k=11; l=12;m=13;n=14;
o=15; p=16; q=17; r=18; s=19; t=20; u=21; v=22; w=23; x=24; y=25; z=26;
```

Não há necessidade de declarar o caractere 0.

Na Figura 4.27, mostra-se a execução da Atividade 14 no FreeMat e, na Figura 4.28, apresenta as variáveis salvas após a inserção dos comandos.

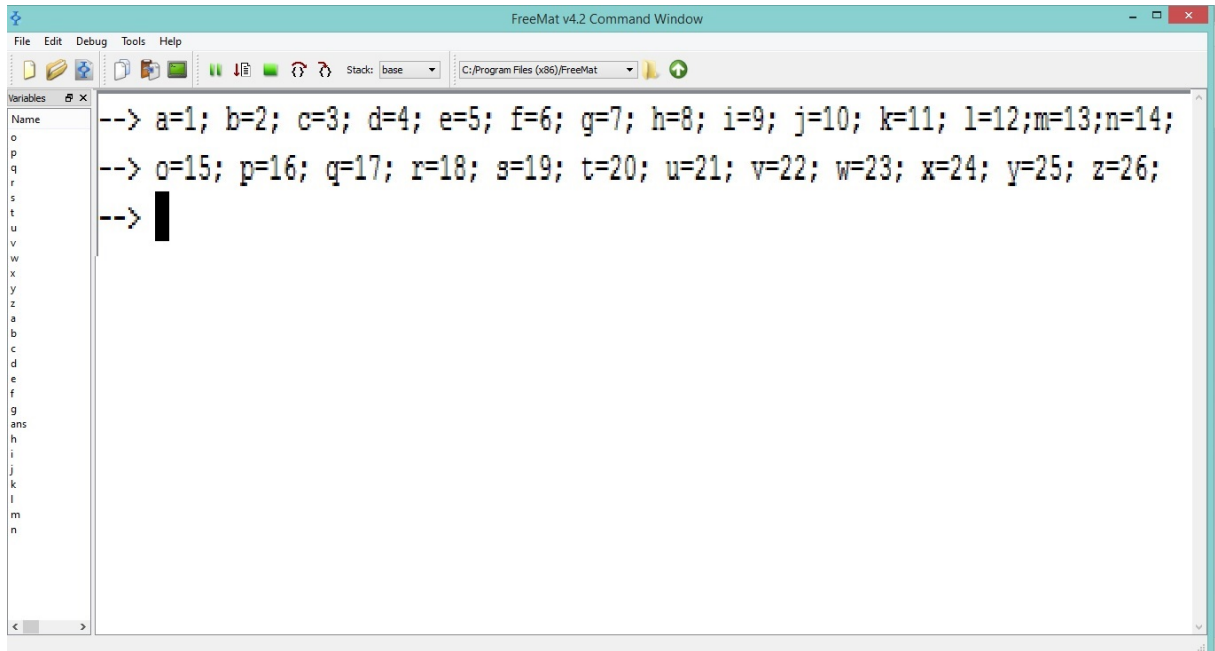


Figura 4.27: Desenvolvimento da Atividade ?? no FreeMat

Name	Value
o	15
p	16
q	17
r	18
s	19
t	20
u	21
v	22
w	23
x	24
y	25
z	26
a	1
b	2
c	3
d	4
e	5
f	6
g	7
ans	
h	8
i	9
j	10
k	11
l	12
m	13
n	14

Figura 4.28: Variáveis armazenados e dispostos na janela *Variables*

Atividade 15. Dado o texto comum “Alan Turing”¹, divida este texto em blocos de 3 caracteres e, utilizando a correspondência numérica da Tabela 3.2 (inserida na Atividade 14), forme uma matriz M de ordem 4×3 .

Passo 1: Dividir o texto comum em blocos de 3 caracteres.

Acrescentando o caractere 0 para completar espaços em branco e, agrupando em 4 blocos, contendo cada um 3 elementos, obtém-se os seguintes blocos:

a l a n 0 t u r i n g 0

Passo 2: Determinar a matriz M , cujas entradas são dadas pela correspondência numérica da Tabela 3.2.

A matriz é descrita, inserindo o seguinte comando:

M=[a l a; n 0 t; u r i; n g 0]

Na Figura 4.29, é apresentada a matriz reportada pelo programa.

```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Users\Wilson Junior\Google Drive
--> M=[a l a; n 0 t; u r i; n g 0]
M =
  1 12  1
 14  0 20
 21 18  9
 14  7  0
--> █
```

Figura 4.29: Desenvolvimento da Atividade 15 no FreeMat

¹Alan Turing, 1912-1954: Matemático britânico que lançou bases para ciências da computação formalizando o conceito de algoritmo. Durante a Segunda Guerra Mundial teve papel instrumental no desenvolvimento de técnicas capazes de decifrar mensagens codificadas usadas pelos alemães. Fonte: http://www.ugr.es/~aprieto/Ciclo_conferencias_vf3.pdf

Atividade 16. Considere a matriz chave $A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$. Por meio do FreeMat, utilize a matriz chave A , para criptografar a mensagem: “Alan Turing”.

Passo 1: Inserir a matriz chave A .

A matriz é inserida através do comando:

```
A=[1 0 2; 2 -1 3; 4 1 8]
```

Passo 2: Dividir o texto comum em blocos de 3 caracteres.

De modo idêntico ao que foi feito na Atividade 15, acrescenta-se o caractere 0 para completar espaços em branco e, agrupando em 4 blocos, contendo cada um 3 elementos, obtém-se os seguintes blocos:

```
ala  n0t  uri  ng0
```

Passo 3: Inserir a matriz M , cujas entradas são dadas pela correspondência numérica da Tabela 3.2.

A matriz é inserida através do seguinte comando:

```
M=[a l a; n 0 t; u r i; n g 0]
```

Passo 4: Obter a matriz cifrada MA .

A matriz cifrada é dada, inserindo o comando:

```
MA=M*A
```

Na Figura 4.30, é apresentada a matriz criptografada MA .

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat
--> A=[1 0 2; 2 -1 3; 4 1 8];
--> M=[a l a; n o t; u r i; n g o]

M =
  1 12  1
 14  0 20
 21 18  9
 14  7  0

--> MA=M*A

MA =
 29 -11  46
 94  20 188
 93  -9 168
 28  -7  49

```

Figura 4.30: Desenvolvimento da Atividade 16 no FreeMat

Passo 5: Apresentar a mensagem a ser transmitida.

Agrupando as linhas da matriz MA , em sequência, obtém-se a mensagem a ser transmitida:

29 -11 46 94 20 188 93 -9 168 28 -7 49

Atividade 17. Considere a matriz chave $A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & -1 & 3 \\ 4 & 1 & 8 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$. Por meio do FreeMat, decifre o criptograma:

29 -11 46 94 20 188 93 -9 168 28 -7 49

Passo 1: Inserir a matriz chave A .

A matriz é inserida através do comando:

$A=[1 \ 0 \ 2; \ 2 \ -1 \ 3; \ 4 \ 1 \ 8]$

Passo 2: Agrupar os blocos do criptograma e obter a matriz C formada por estes blocos.

Agrupar-se em 4 blocos, contendo cada um 3 elementos.

29 - 11 46 94 20 168 93 - 9 168 28 - 7 49

Então

$$C = \begin{bmatrix} 29 & -11 & 46 \\ 94 & 20 & 168 \\ 93 & -9 & 168 \\ 28 & -7 & 49 \end{bmatrix}.$$

Passo 3: Inserir a matriz C de ordem 3×4 .

Como cada bloco corresponde a uma linha da matriz C , a matriz é inserida através do comando:

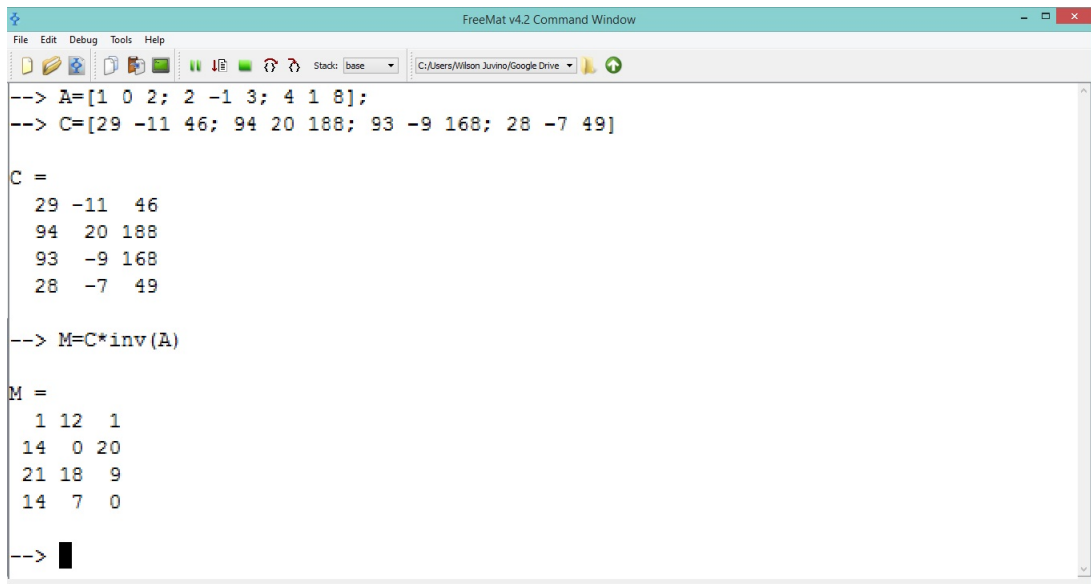
```
C=[29 -11 46; 94 20 188; 93 -9 168; 28 -7 49]
```

Passo 4: Obter os blocos do texto comum.

Para obter os blocos do texto comum, deve-se inserir o comando:

```
M=C*inv(A)
```

Na Figura 4.31, é apresentada a matriz M .



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Users\Wilson Junior\Google Drive
--> A=[1 0 2; 2 -1 3; 4 1 8];
--> C=[29 -11 46; 94 20 188; 93 -9 168; 28 -7 49]
C =
 29 -11 46
 94 20 188
 93 -9 168
 28 -7 49
--> M=C*inv(A)
M =
 1 12 1
 14 0 20
 21 18 9
 14 7 0
-->
```

Figura 4.31: Desenvolvimento da Atividade 17 no FreeMat

Passo 5: Apresentar a mensagem transmitida.

Agrupando as linhas da matriz M obtida no Passo 4, em sequência, obtém-se:

1 12 1 14 0 20 21 18 9 14 7 0

Logo, utilizando a correspondência numérica da Tabela 3.2, obtém-se o texto comum:

alan0turing0

Portanto, a mensagem transmitida era “Alan Turing”.

Atividade 18. Considere a matriz chave $A = \begin{bmatrix} 3 & 1 \\ 4 & 5 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$. Por meio do FreeMat, utilize a matriz A , para criptografar a mensagem: “Alan Turing”.

Passo 1: Inserir a matriz chave A .

A matriz é inserida através do comando:

```
A=[3 1; 4 5]
```

Passo 2: Dividir o texto comum em blocos de 2 caracteres.

De modo idêntico ao que foi feito na Atividade 15, acrescenta-se o caractere 0 para completar espaços em branco e, agrupando em 6 blocos, contendo cada um 2 elementos, obtém-se os seguintes blocos:

al an 0t ur in g0

Passo 3: Inserir a matriz M , cujas entradas são dadas pela correspondência numérica da Tabela 3.2.

A matriz é inserida através do seguinte comando:

```
M=[a l; a n; 0 t; u r; i n; g 0]
```

Passo 4: Obter a matriz cifrada MA .

A matriz cifrada com entradas em $\mathcal{M}_2(\mathbb{Z}_{27})$ é obtida inserindo o comando:

$MA = \text{mod}(M \cdot A, 27)$

NA Figura 4.32, é apresentada a matriz criptografada MA .

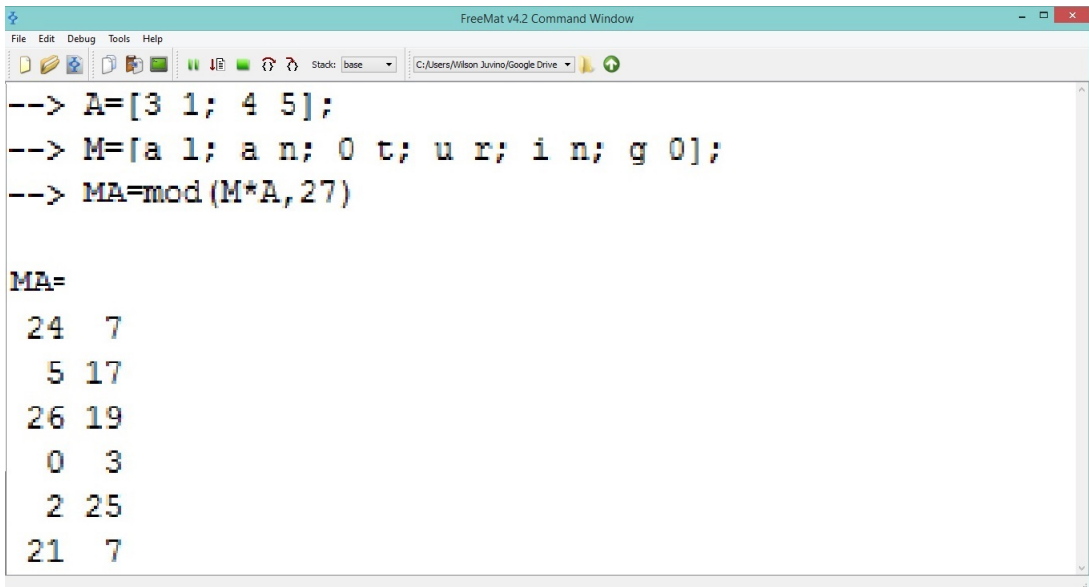


Figura 4.32: Desenvolvimento da Atividade 18 no FreeMat

Passo 5: Apresentar a mensagem a ser transmitida.

Agrupando as linhas da matriz MA obtida no Passo 4, em sequência, obtém-se:

24 7 5 17 26 19 0 3 2 25 21 7

Logo, utilizando a correspondência numérica da Tabela 3.2, obtém-se a mensagem a ser transmitida:

xgeqzs0cbyug

Atividade 19. Considere a matriz chave $A = \begin{bmatrix} 3 & 1 \\ 4 & 5 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$. Por meio do FreeMat, decifre o criptograma:

xgeqzs0cbyug

Passo 1: Inserir a matriz chave A .

A matriz é inserida através do comando:

$$A=[3 \ 1; \ 4 \ 5]$$

Passo 2: Agrupar os blocos do criptograma e obter a matriz C formada por estes blocos.

Agrupar-se em 6 blocos, contendo cada um 2 elementos.

xg eq zs 0c by ug

Então,

$$C = \begin{bmatrix} x & g \\ e & q \\ z & s \\ 0 & c \\ b & y \\ u & g \end{bmatrix}.$$

Passo 3: Inserir a matriz C de ordem 2×6 .

Como cada bloco corresponde a uma linha da matriz C , a matriz é inserida através do comando:

$$C=[x \ g; \ e \ q; \ z \ s; \ 0 \ c; \ b \ y; \ u \ g]$$

Passo 4: Obter a matriz chave inversa.

Como $3 \cdot 5 - 4 \cdot 1 = 11$ e, pela Tabela 3.1, o inverso multiplicativo de 11 em \mathbb{Z}_{27} é 5, então,

$$A^{-1} = 5 \cdot \begin{bmatrix} 5 & -1 \\ -4 & 3 \end{bmatrix}.$$

Logo a matriz $\text{inv}A = A^{-1}$ é obtida inserindo o comando:

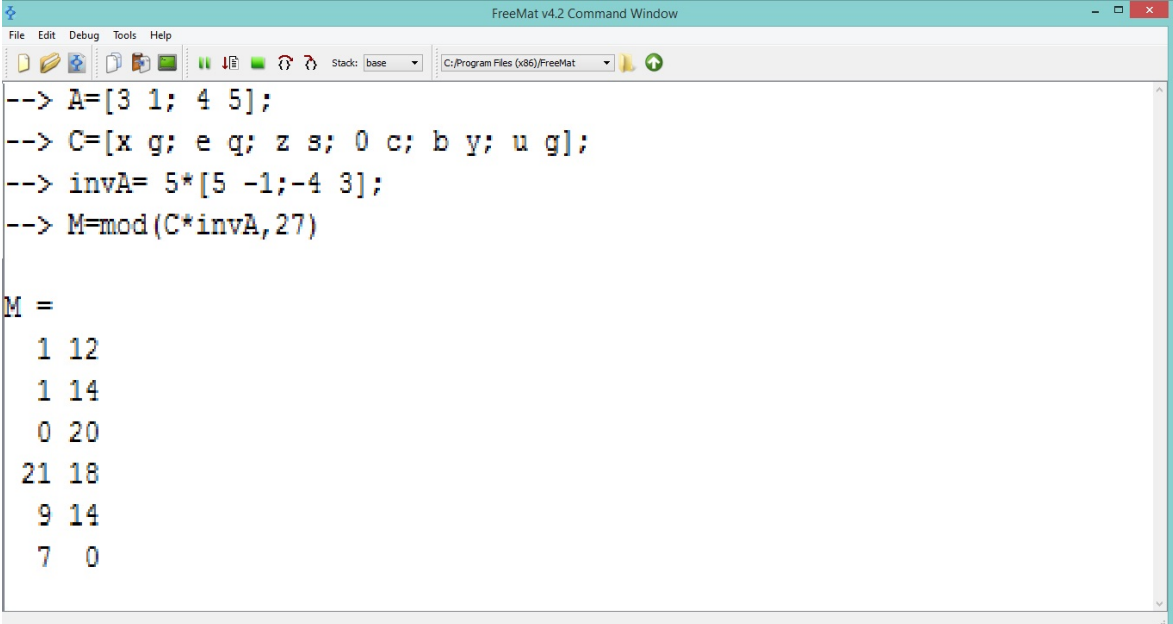
$$\text{inv}A= 5*[5 \ -1; \ -4 \ 3]$$

Passo 5: Obter os blocos do texto comum.

Para obter os blocos do texto comum, deve-se inserir os comandos:

```
M=mod(C*invA,27)
```

Na Figura 4.33, é apresentada a matriz M .



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:\Program Files (x86)\FreeMat
--> A=[3 1; 4 5];
--> C=[x g; e q; z s; 0 c; b y; u g];
--> invA= 5*[5 -1;-4 3];
--> M=mod(C*invA,27)

M =
 1 12
 1 14
 0 20
21 18
 9 14
 7  0
```

Figura 4.33: Desenvolvimento da Atividade 19 no FreeMat

Passo 6: Apresentar a mensagem transmitida.

Agrupando as linhas da matriz M obtida no Passo 4, obtém-se:

```
1 12 1 14 0 20 21 18 9 14 7 0
```

Logo, utilizando a correspondência numérica da Tabela 3.2, obtém-se o texto comum:

```
alan0turing0
```

Portanto, a mensagem transmitida era “Alan Turing”.

Atividade 20. Utilizando o sistema poligráfico baseado na multiplicação de matrizes com entradas em \mathbb{R} , construa um programa no FreeMat, para criptografar mensagens de texto comum, e o intitule de ‘CRIPTOGRAFAR’. O programa deve fazer o seguinte: a) solicitar a inserção da matriz chave $A \in \mathcal{M}_n(\mathbb{R})$; b) se $\det A \neq 0$, o programa deve solicitar o ingresso da matriz M de ordem $m \times n$, onde cada linha da matriz M é formada por m blocos de n letras do texto comum; c) se $\det A = 0$, o programa deve reportar a seguinte mensagem para o usuário: “A matriz chave não é válida, pois a matriz não é inversível”.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘CRIPTOGRAFAR’ como nome para o programa.

No menu do FreeMat selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function  CRIPTOGRAFAR
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda `nao`, na linha abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Corresponder cada letra do alfabeto, dado na Subseção 3.2.2, com um número inteiro, de acordo com a Tabela 3.2.

A correspondência é definida inserindo os comandos:

```
a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;
o=15;p=16;q=17;r=18;s=19;t=20; u=21; v=22; w=23; x=24; y=25; z=26;
```

Não há necessidade de declarar o caractere 0.

Passo 4: Criar um display com informações referente ao programa e que apresente a correspondência de cifras, definida no Passo 3.

Cria-se o display, com os caracteres e valores da tabela definidos no Passo 2, inserindo os comandos:

```
clc
disp('Programa para criptografar mensagens pelo sistema poligráfico')
disp(' baseado no método de multiplicação de matrizes com entradas reais')
disp(' a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;')
disp(' o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26;')
disp('0 caractere 0 corresponde a um espaço vazio')
```

Passo 5: Definir comandos para o usuário inserir a matriz chave $A \in \mathcal{M}_n(\mathbb{R})$ e para o programa armazenar a ordem da matriz inserida.

Para definir a entrada da matriz chave A e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
A=input('Entre com a matriz chave de ordem n : A =')
```

Para o programa armazenar a ordem da matriz inserida, define-se o comando

```
[I,J]= size(A);
```

Passo 6: Definir as condições e parâmetros de decisão, de acordo com a chave criptográfica inserida, para determinar a matriz criptografada ou, caso contrário, apresentar mensagens sobre a validade da chave criptográfica.

As chaves válidas deverão ser de ordem n e invertíveis, se estas condições são válidas deverá ser apresentada a matriz decifrada.

Passo 6.1: Inserir comando executando ações se a chave inserida não é inversível.

```
elseif det(A)==0
disp('A matriz chave não é válida, pois a matriz não é inversível')
. reiniciar=input('Deseja encerrar o programa? sim / nao' )
disp('Fim do Programa.')
```

Passo 6.2: Inserir comando executando ações se a chave criptográfica é válida, ou seja, se a condição anterior não é satisfeita.

```
else
clc
disp ('A matriz de M deve possuir J elementos em cada linha')
J=J
M=input('Entre com a matriz cujas entradas são caracteres
do texto comum. M=')
disp ('Cada linha da matriz MA é um bloco do criptograma')
MA=M*A
continuar=input('Deseja encerrar o programa? sim /nao')
disp ('Fim do Programa.')
```

Passo 7: Salvar o programa.

Para salvá-lo, selecione a opção *Save* ou *Save As* do menu. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “CRIPTOGRAFAR” e salvo na extensão *.m*, assim o arquivo final será: “CRIPTOGRAFAR.m”. Na Figura 4.34, apresenta-se parte dos comandos relativos a este programa.

Observação 4.7. O programa é apresentado integralmente no Apêndice A.2, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

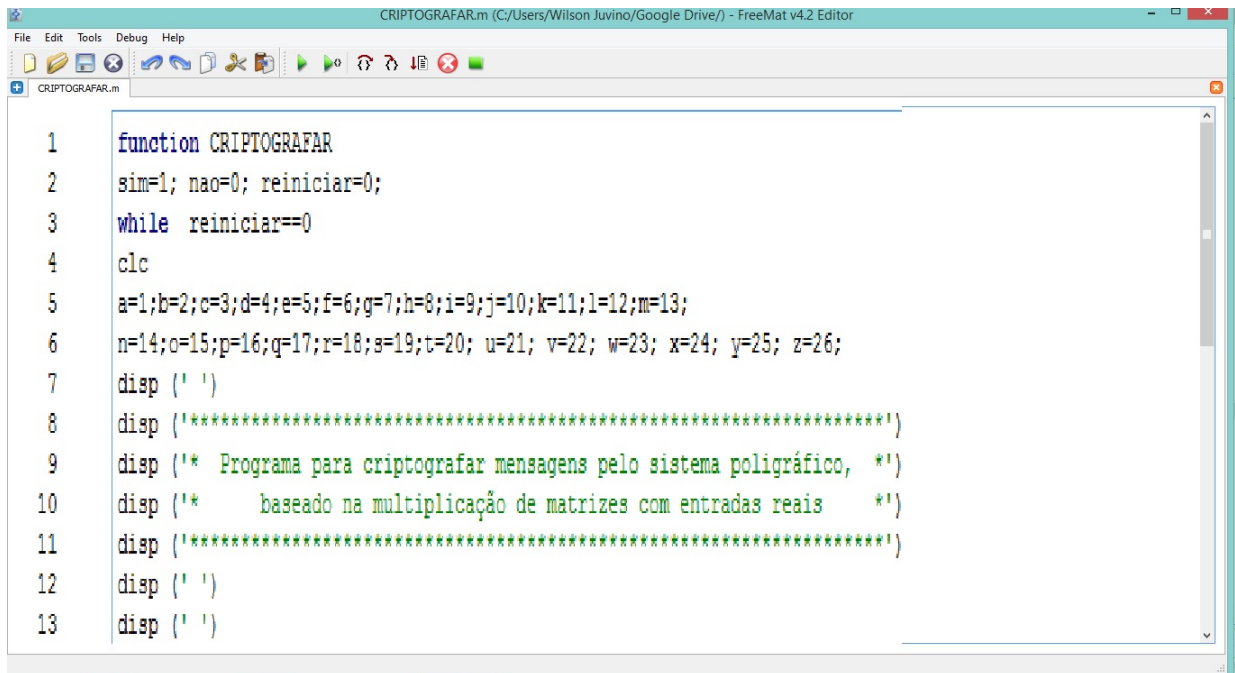


Figura 4.34: Editor do FreeMat e parte dos comandos relativos ao programa “CRIPTOGRAFAR”

Atividade 21. Utilizando o método poligráfico baseado na multiplicação de matrizes com entradas em \mathbb{R} , construa um programa no FreeMat, para decifrar textos cifrados, e o intitule de “DECIFRAR”. O programa deve fazer o seguinte: a) solicitar a inserção da matriz chave $A \in \mathcal{M}_n(\mathbb{R})$; b) se $\det A \neq 0$, o programa deve solicitar o ingresso da matriz C de ordem $m \times n$, onde cada linha da matriz C é formada por blocos de n caracteres do texto cifrado; c) se $\det A = 0$, o programa deve reportar a seguinte mensagem para o usuário: ‘A matriz chave não é válida, pois a matriz não é inversível’.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘DECIFRAR’ como nome para o programa.

No menu do FreeMat, selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function DECIFRAR
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda `nao`, na linha abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Criar um display com informações referente ao programa e que apresente a correspondência de cifras definida no Passo 2.

Cria-se o display, com os caracteres e valores da tabela definidos no Passo 2, inserindo os comandos:

```
clc
disp('Programa para decifrar mensagens pelo sistema poligráfico')
disp('baseado no método de multiplicação de matrizes com entradas reais.')
```

```
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;')
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26;')
```

```
disp('0 caractere 0 corresponde a um espaço vazio')
```

Passo 4: Definir comandos para que o usuário insira a a matriz chave $A \in \mathcal{M}_n(\mathbb{R})$ e para o programa armazenar a ordem da matriz inserida.

Para definir a entrada da matriz chave A e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
A=input('Entre com a matriz chave de ordem n: A = ')
```

Para o programa armazenar a ordem da matriz inserida, define-se o comando

```
[I,J]=size(A)
```

Passo 5: Definir as condições e parâmetros de decisão, de acordo com a chave criptográfica inserida, para determinar a matriz decifrada ou, caso contrário, apresentar mensagens sobre a validade da matriz chave.

A matriz chave válida deverá ser inversível, se esta condição é válida deve ser reportada a matriz decifrada.

Passo 5.1: Inserir comando executando ações se a chave inserida não é inversível.

```
if det(A)==0
disp('A matriz chave não é válida, pois a matriz não é inversível.')
```

reiniciar=input('Deseja encerrar o programa? sim / nao')

```
disp('Fim do Programa.')
```

Passo 5.2 : Inserir comando executando ações se a chave criptográfica é válida, ou seja, se a condição anterior não é satisfeita.

```
else
disp ('A matriz de C deve possuir J elementos em cada linha')
J=J
C=input('Entre com a matriz cujos elementos são caracteres)
do criptograma. C=')
D=inv(A);
disp ('*Cada linha da matriz M é um bloco do texto comum.*')
```

M=C*inv(A)

```
disp ('*Utilize a tabela abaixo para escrever o texto comum.*')
```

```
disp ('*Tabela de correspondência numérica*')
```

```
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11; l=12;m=13;n=14;')
```

```
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26')
```

```
disp('0 caractere 0 corresponde a um espaço vazio')
```

```
continuar=input('Deseja encerrar o programa? sim / nao=')
```

```
end
end
```

Passo 6: Salvar o programa.

Para salvá-lo, selecione a opção *Save* ou *Save As* do menu. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “DECIFRAR” e salvo na extensão *.m*, assim

o arquivo final será: “DECIFRAR.m”. Na Figura 4.35, apresenta-se parte dos comandos relativos a este programa.

```

1 function DECIFRAR
2 sim=1; nao=0; reiniciar=0;
3 while reiniciar==0
4 clc
5 disp(' ')
6 disp('*****')
7 disp('* Programa para decifrar mensagens pelo sistema poligráfico, *')
8 disp('* baseado na multiplicação de matrizes com entradas reais *')
9 disp('*****')
10 disp(' ')
11 disp(' ')

```

Figura 4.35: Editor do FreeMat e parte dos comandos relativos ao programa “DECIFRAR”

Observação 4.8. O programa é apresentado integralmente no Apêndice A.3, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

Atividade 22. Considere a matriz chave: $A = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 5 & 1 \\ 2 & 2 & 0 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$. Por meio do programa “CRIPTOGRAFAR” desenvolvido na Atividade 20, utilize a matriz chave A , para criptografar a mensagem: “Mestrado Profissional em Matemática em Rede Nacional”.

Passo 1: Abrir o programa.

Para abrir o programa, na janela de comandos, digita-se:

CRIPTOGRAFIA

Após, pressione **Enter** e o programa será compilado (ver Figura 4.36).

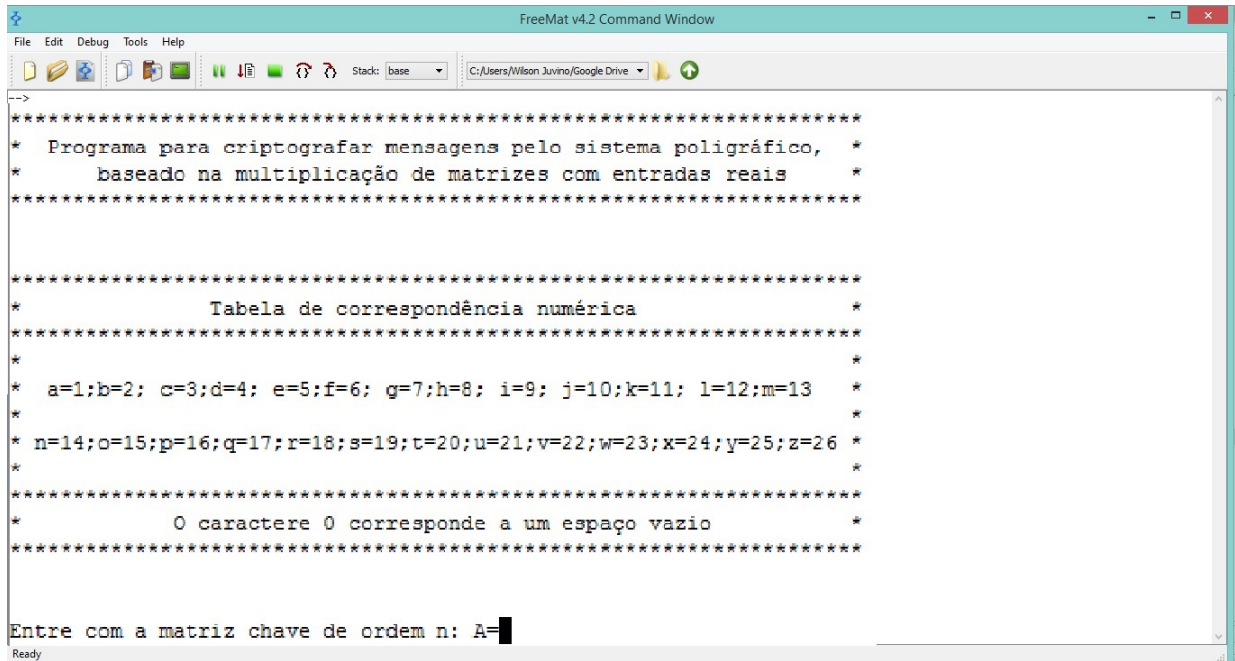


Figura 4.36: Desenvolvimento da Atividade 22, primeira etapa do programa

Passo 2: Inserir a matriz chave $A = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 5 & 1 \\ 2 & 2 & 0 \end{bmatrix}$.

Para inserir a matriz chave, digita-se o comando:

`[1 0 2; 3 5 1; 2 2 0]`

e, ao pressionar **Enter**, é solicitado para o usuário inserir a matriz M , cujas entradas são os caracteres do texto comum, conforme é apresentado na Figura 4.37.

Passo 3: Inserir a matriz onde as entradas são caracteres do texto comum.

A matriz é inserida através do comando:

`[m e s;t r a;d o 0;p r o;f i s;s i o;n a l;0 e m;0 m a;t e m;
a t i;c a 0;e m 0;r e d;e 0 n; a c i; o n a; 1 0 0]`

Ao pressionar **Enter**, o programa reportará a matriz criptografada, apresentada na Figura 4.38.

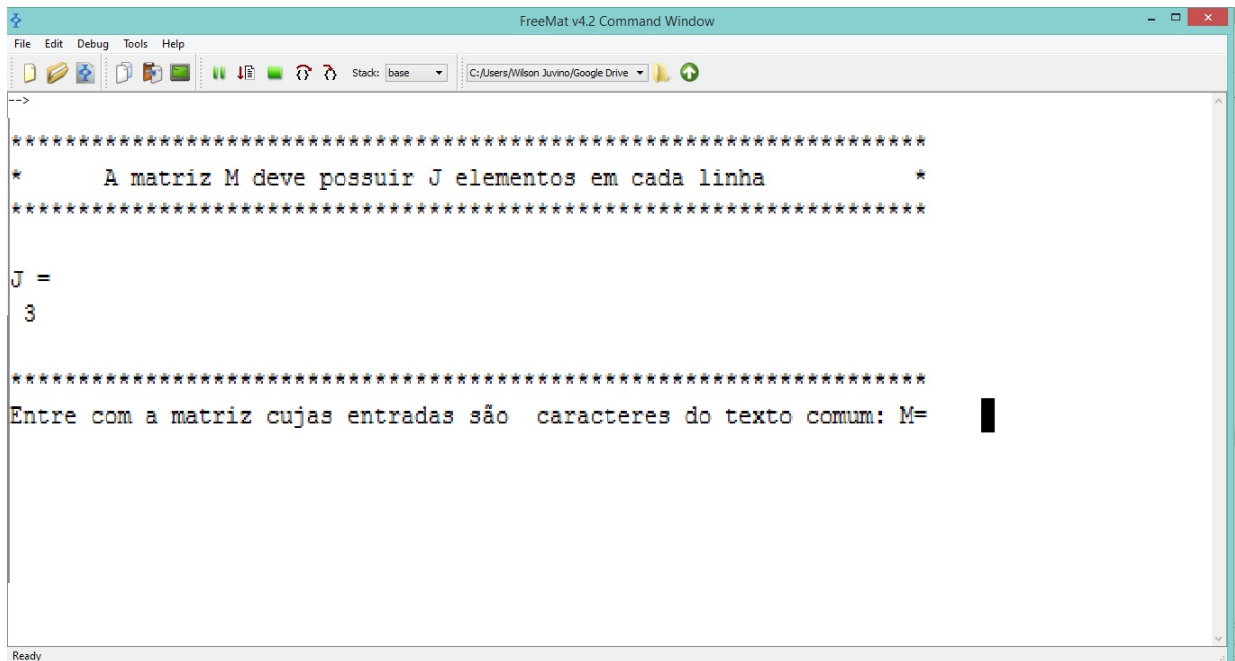


Figura 4.37: Desenvolvimento da Atividade 22, segunda etapa do programa.

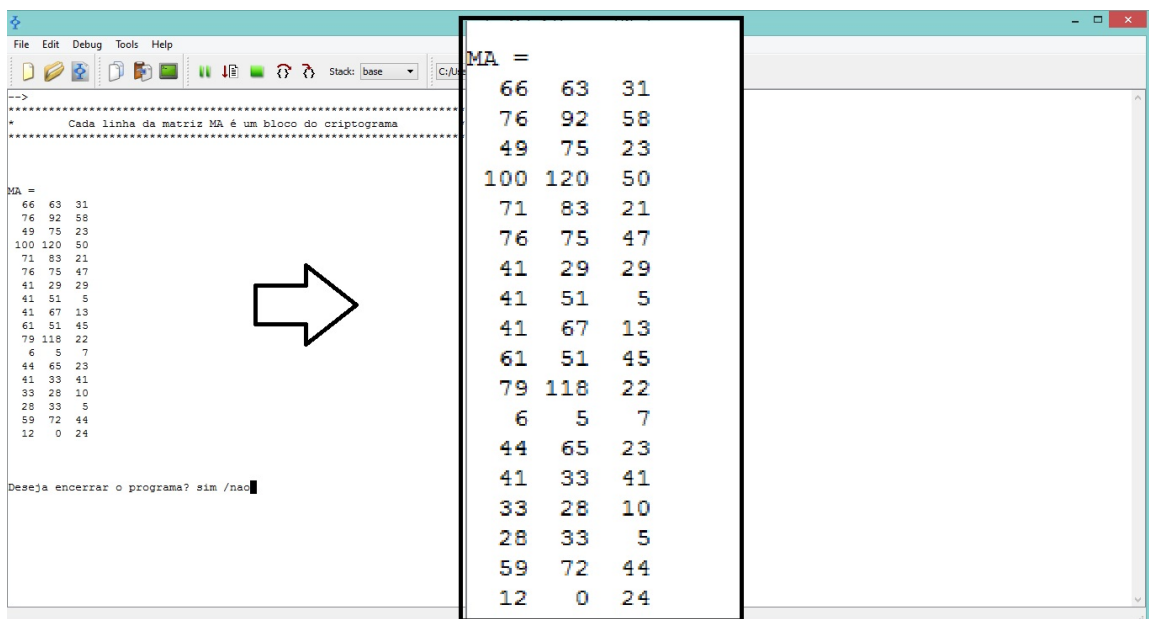


Figura 4.38: Desenvolvimento da Atividade 22, terceira etapa do programa

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir `nao` e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.36, caso contrário, se inserir `sim`, o programa é finalizado, sendo apresentado o display da Figura 4.39.

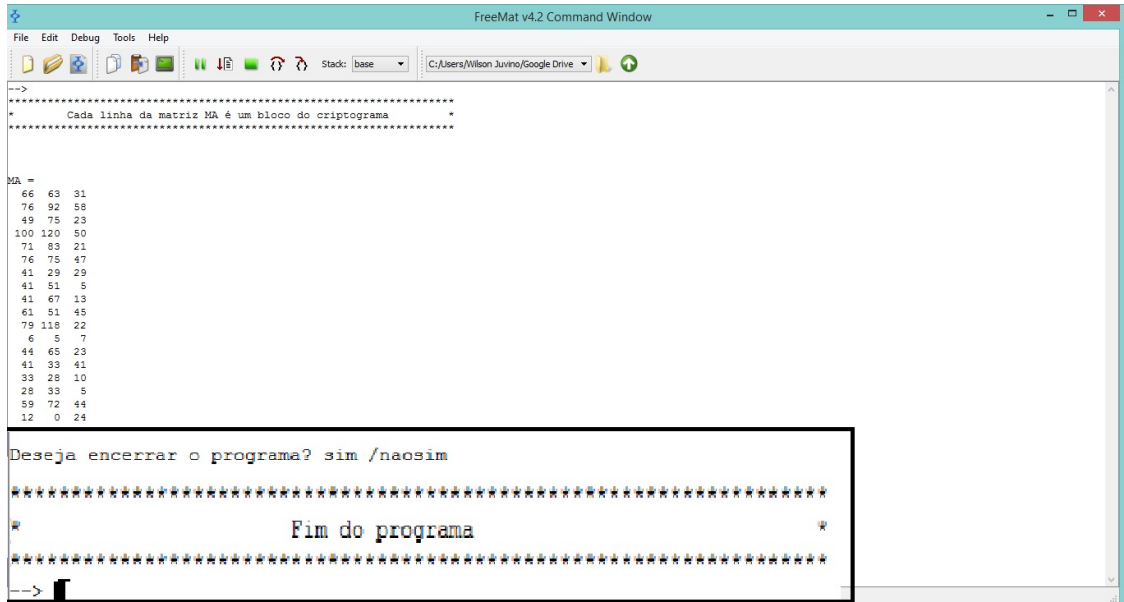


Figura 4.39: Desenvolvimento da Atividade 22, quarta etapa do programa.

Passo 5: Apresentar a mensagem a ser transmitida:

Agrupando as linhas da matriz MA obtida no Passo 3, em sequência, obtém-se a mensagem a ser transmitida:

66 63 31 76 92 58 49 75 23 100 120 50 71 83 21 76 75 47 41 29 29 41 51 5 41 67
13 61 51 45 79 118 22 6 5 7 44 65 23 41 33 41 33 28 10 28 33 5 59 72 44 12 0 24

Atividade 23. Considere a matriz chave: $A = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 5 & 1 \\ 2 & 2 & 0 \end{bmatrix} \in \mathcal{M}_3(\mathbb{R})$. Por meio do programa “DECIFRAR”, desenvolvido na Atividade 21, decifre o criptograma:

66 63 31 76 92 58 49 75 23 100 120 50 71 83 21 76 75 47 41 29 29 41 51 5 41 67
13 61 51 45 79 118 22 6 5 7 44 65 23 41 33 41 33 28 10 28 33 5 59 72 44 12 0 24

Passo 1: Abrir o programa.

Para abrir o programa, na janela de comandos, digita-se:

DECIFRAR

Após, pressione **Enter** e o programa será compilado (ver Figura 4.40).

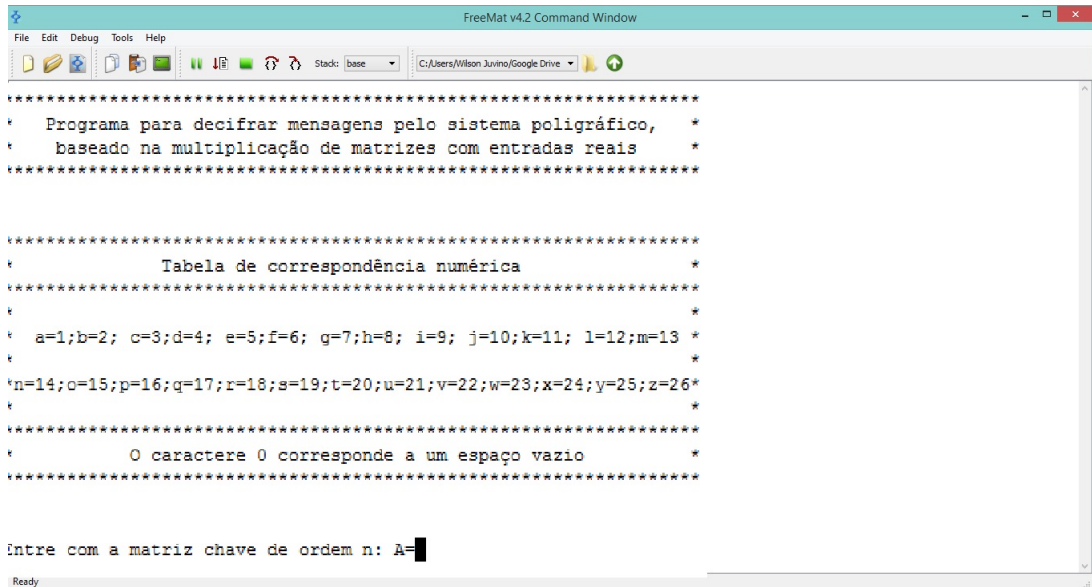


Figura 4.40: Desenvolvimento da Atividade 23, primeira etapa do programa.

Passo 2: Inserir a matriz chave $A = \begin{bmatrix} 1 & 0 & 2 \\ 3 & 5 & 1 \\ 2 & 2 & 0 \end{bmatrix}$.

Para inserir a matriz chave, digita-se o comando:

[1 0 2; 3 5 1; 2 2 0]

e, ao pressionar **Enter**, é solicitado para o usuário inserir a matriz C , cuja entradas são caracteres do criptograma, conforme mostra a Figura 4.41.

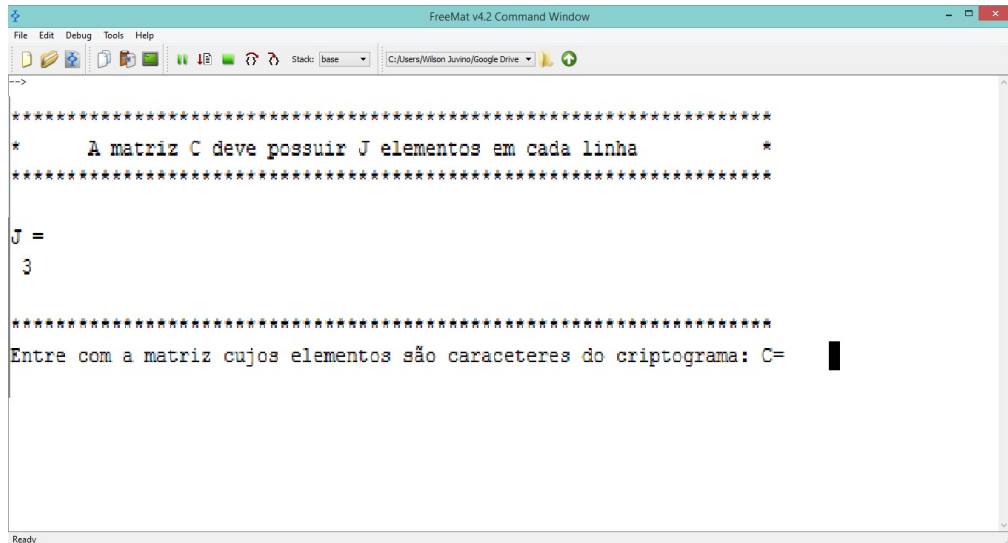


Figura 4.41: Desenvolvimento da Atividade 23, segunda etapa do programa.

Passo 3: Inserir a matriz onde as entradas são caracteres do criptograma.

A matriz é inserida através do comando:

```
[66 63 31;76 92 58;49 75 23;100 120 50;71 83 21;76 75 47;41 29 29;41 51 5;41 67 13;
 61 51 45; 79 118 22;6 5 7;44 65 23;41 33 41;33 28 10;28 33 5; 59 72 44; 12 0 24]
```

Ao pressionar **Enter**, o programa reportará a matriz decifrada apresentada na Figura 4.42.

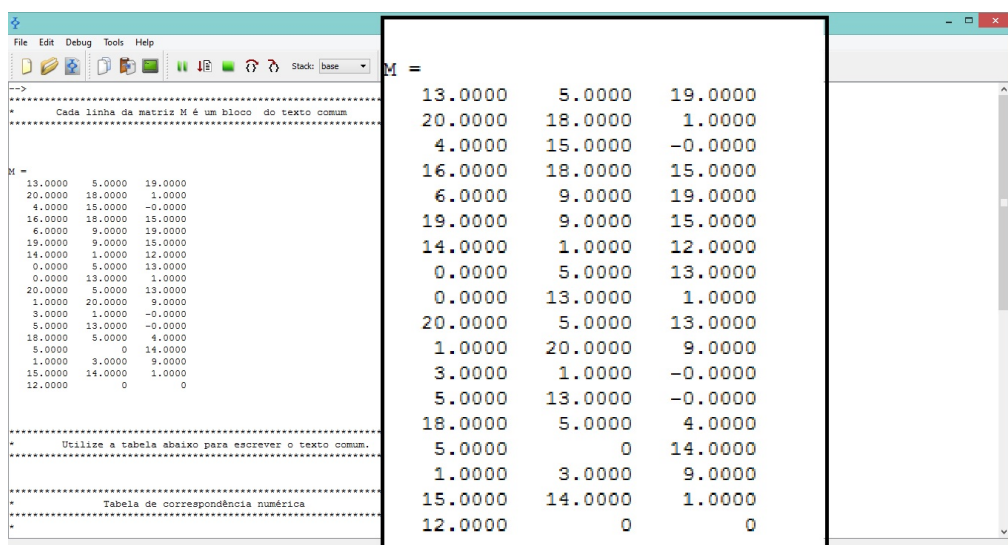


Figura 4.42: Desenvolvimento da Atividade 23, terceira etapa do programa.

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.40, caso contrário, se inserir , o programa é finalizado, sendo apresentado o display da Figura 4.43.

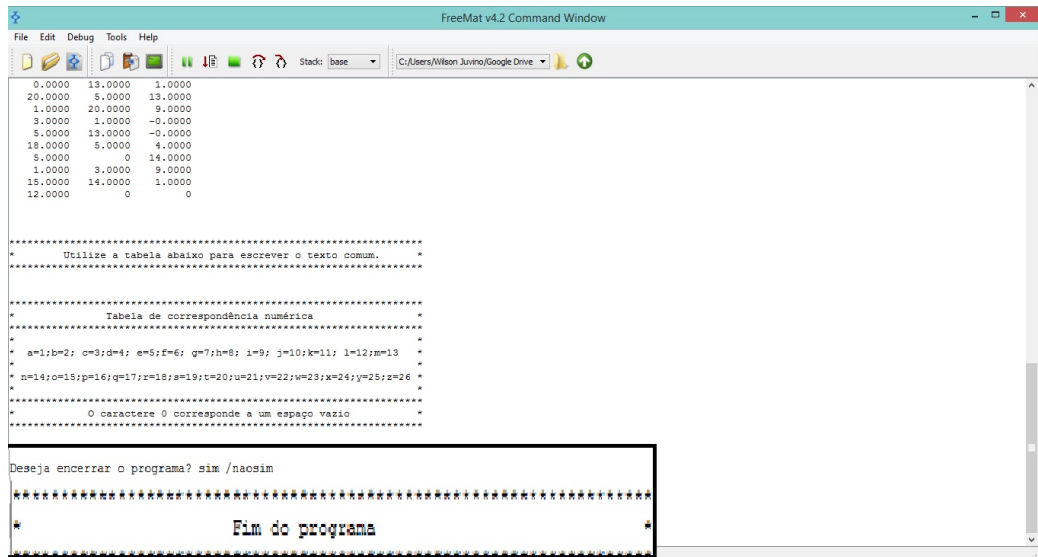


Figura 4.43: Desenvolvimento da Atividade 23, quarta etapa do programa.

Passo 5: Revelar a mensagem.

Agrupando as linhas da matriz M obtida no Passo 3, obtém-se:

```

13 5 19 20 18 1 4 15 0 16 18 15 6 9 19 19 9 15 14 1 12 0 5 13 0 13
1 20 5 13 1 20 9 3 1 0 5 13 0 18 5 4 5 0 14 1 3 9 15 14 1 12 0 0

```

Logo, utilizando a correspondência numérica da tabela, apresentada no display, obtém-se o texto comum:

```
mestrado0profissional0em0matematica0em0rede0nacional00
```

Portanto, a mensagem transmitida era: “Mestrado Profissional em Matemática em Rede Nacional”.

Atividade 24. Utilizando o sistema poligráfico baseado na multiplicação de matrizes com entradas em \mathbb{Z}_{27} , construa um programa no FreeMat, para criptografar mensagens de texto comum, e o intitule de ‘CRIPTOGRAFAR2HILL’. O programa deve fazer o seguinte: a) solicitar a inserção da matriz chave $A \in \mathcal{M}_2(\mathbb{Z}_{27})$; b) se A é inversível o programa deve solicitar o ingresso da matriz M de ordem $m \times 2$, onde cada linha da matriz M é formada por m blocos de 2 letras do texto comum; c) se A não é inversível, o programa deve reportar a seguinte mensagem para o usuário: “A matriz chave não é válida, pois a matriz não é inversível”.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘CRIPTOGRAFAR2HILL’ como nome para o programa.

No menu do FreeMat, selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function  CRIPTOGRAFAR2HILL
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda `nao`, na linha abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Corresponder cada letra do alfabeto, dado na Subseção 3.2.2, com um número inteiro, de acordo com a Tabela 3.2.

A correspondência é definida inserindo os comandos:

```
a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;
o=15;p=16;q=17;r=18;s=19;t=20; u=21; v=22; w=23; x=24; y=25; z=26;
```

Não há necessidade de declarar o caractere 0.

Passo 4: Criar um display com informações referente ao programa e que apresente a correspondência de cifras definida no Passo 2.

Cria-se o display, com os caracteres e valores da tabela definidos no Passo 2, inserindo o comando:

```
clc
disp('Programa para criptografar mensagens pelo sistema poligráfico')
disp(' 2-Cifras de Hill. ')
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;')
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26;')
disp('0 caractere 0 corresponde a um espaço vazio')
```

Passo 5: Definir comandos para o usuário inserir a matriz chave $A \in \mathcal{M}_n(\mathbb{Z}_{27})$.

Para definir a entrada da matriz chave A e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
A=input('Entre com a matriz chave de ordem 2 : A =')
```

Passo 6: Definir as condições e parâmetros de decisão, de acordo com a matriz chave inserida, para determinar a matriz criptografada ou, caso contrário, apresentar mensagens sobre a validade da matriz chave.

As chaves válidas deverão ser de ordem 2 e inversíveis, se estas condições são válidas deverá ser apresentada a matriz cifrada.

Observação 4.9. Para construção do próximo passo, é fundamental observar a Proposição 3.29 e a Tabela 3.1.

Passo 6.1: Inserir comando executando ações se a chave inserida não é de ordem 2 ou não é inversível.

```
if size(A) ~ =[2,2] || mod(A(1)*A(4)-A(2)*A(3),3)==0
disp('A matriz chave não é válida, pois a matriz não é inversível.')
reiniciar=input('Deseja encerrar o programa? sim / nao' )
disp('Fim do Programa.')
```

Passo 6.2: Inserir comando executando ações se a chave criptográfica é válida, ou seja, se as condições anteriores não são satisfeitas.

```
else
disp ('A matriz M deve possuir 2 elementos em cada linha')
M=input('Entre com a matriz cujas entradas são caracteres
do texto comum. M=')
disp (' Cada linha da matriz MA é um bloco do criptograma')
MA=mod(M*A,27)
disp ('Utilize a tabela abaixo para escrever o criptograma.')
```

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

```
disp ('Tabela de correspondência numérica')
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11; l=12;m=13;n=14;')
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26')
disp('0 caractere 0 corresponde a um espaço vazio')
continuar=input('Deseja encerrar o programa? sim /nao')
disp('Fim do Programa')
end
end
```

Passo 7: Salvar o programa.

Para salvá-lo, selecione a opção *Save* ou *Save As* do menu. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “CRIPTOGRAFARHILL2” e salvo na extensão *.m*, assim o arquivo final será: “CRIPTOGRAFARHILL2.m”.

Observação 4.10. O programa é apresentado integralmente no Apêndice A.4, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

Atividade 25. Utilizando o método poligráfico baseado na multiplicação de matrizes com entradas em \mathbb{Z}_{27} , construa um programa no FreeMat para decifrar textos cifrados, e o intitule de “DECIFRARHILL2”. O programa deve fazer o seguinte: a) solicitar a inserção da matriz chave $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$; b) se A é inversível, o programa deve solicitar o ingresso da matriz C de ordem $m \times 2$, onde cada linha da matriz C é formada por m blocos de 2 caracteres do texto cifrado; c) se A não é inversível, o programa deve reportar a seguinte mensagem para o usuário: “A matriz chave não é válida, pois a matriz não é inversível”.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘DECIFRARHILL2’ como nome para o programa.

No menu do FreeMat, selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function DECIFRARHILL2
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda *nao*, na linha abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Corresponder cada letra do alfabeto, dado na Subseção 3.2.2, com um número inteiro, de acordo com a Tabela 3.2.

A correspondência é definida inserindo os comandos:

```
a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;
o=15;p=16;q=17;r=18;s=19;t=20; u=21; v=22; w=23; x=24; y=25; z=26;
```

Não há necessidade de declarar o caractere 0.

Passo 4: Criar um display com informações referente ao programa e que apresente a correspondência de cifras definida no Passo 2.

Cria-se o display, com os caracteres e valores da tabela definidos no Passo 2, inserindo os comandos:

```
clc
disp('Programa para decifrar mensagens pelo sistema poligráfico')
disp(' 2-Cifras de Hill.  ')
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14;')
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26;')
disp('0 caractere 0 corresponde a um espaço vazio')
```

Passo 5: Definir comandos para que o usuário insira a matriz chave $A \in \mathcal{M}_2(\mathbb{Z}_{27})$ e para

o programa armazenar a matriz $D = \begin{bmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$.

Para definir a entrada da matriz chave A e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
A=input('Entre com a matriz chave de ordem 2: A = ')
```

Para o programa armazenar a matriz D , define-se o comando:

```
D[A(4) -A(3); -A(2) A(1)]
```

Passo 6: Definir as condições e parâmetros de decisão, de acordo com a matriz chave inserida, para determinar a matriz criptografada ou, caso contrário, apresentar mensagens sobre a validade da matriz chave .

As chaves válidas são de ordem 2 e inversíveis, se estas condições são válidas deverá ser apresentada a matriz decifrada.

Observação 4.11. Para construção do próximo passo, é fundamental observar a Proposição 3.29 e a Tabela 3.1.

Passo 6.1: Inserir comando executando ações se a chave inserida não é de ordem 2 ou não é inversível.

```
if size(A) ~ =[2,2] || mod(A(1)*A(4)-A(2)*A(3),3)==0
disp('A matriz chave não é válida, pois a matriz não é inversível.')
```

reiniciar=input('Deseja encerrar o programa? sim / nao')

```
disp('Fim do Programa.')
```

Passo 6.2: Inserir comandos executando ações se a chave criptográfica é válida, ou seja, se as condições anteriores não são satisfeitas.

```
elseif mod(A(1)*A(4)-A(2)*A(3),27)==1
inv=1*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==2
inv=14*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==4
inv=7*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==5
inv=11*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==7
inv=4*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==8
inv=17*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==10
inv=19*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==11
inv=5*D
elseif mod(A(1)*A(4)-A(2)*A(3),27)==13
inv=25*D
```



```

elseif mod(A(1)*A(4)-A(2)*A(3)),27)==14
inv=2*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==16
inv=22*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==17
inv=8*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==19
inv=10*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==20
inv=23*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==22
inv=16*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==23
inv=20*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==25
inv=13*D
elseif mod(A(1)*A(4)-A(2)*A(3)),27)==26
inv=26*D
end
disp ('A matriz de C deve possuir 2 elementos em cada linha')
C=input('Entre com a matriz cujos elementos são caracteres
do criptograma. C=')
disp ('*Cada linha da matriz M é um bloco do texto comum.*')
M=mod(C*inv,27);
disp ('*Utilize a tabela abaixo para escrever o texto comum.*')
disp ('*Tabela de correspondência numéria*')
disp('a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11; l=12;m=13;n=14;')
disp('o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26')
disp('0 caractere 0 corresponde a um espaço vazio')

```

```
continuar=input('Deseja encerrar o programa? sim=1 / não=0')
end
```

Passo 6: Salvar o programa.

Para salvá-lo, selecione a opção *Save* ou *Save As* do menu. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “DECIFRARHILL2” e salvo na extensão `.m`, assim o arquivo final será: “DECIFRARHILL2.m”.

Observação 4.12. O programa é apresentado integralmente no Apêndice A.5, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

Atividade 26. Considere a matriz chave: $A = \begin{bmatrix} 3 & 1 \\ -4 & 5 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$.

Por meio do programa “CRIPTOGRAFARHILL2” desenvolvido na Atividade 24, utilize a matriz chave A , para criptografar a mensagem: “Mestrado Profissional em Matemática em Rede Nacional”.

Passo 1: Abrir o programa.

Para abrir o programa, na janela de comandos, digita-se:

```
CRIPTOGRAFIAHILL2
```

Após, pressione `Enter` e o programa será compilado (ver Figura 4.44).

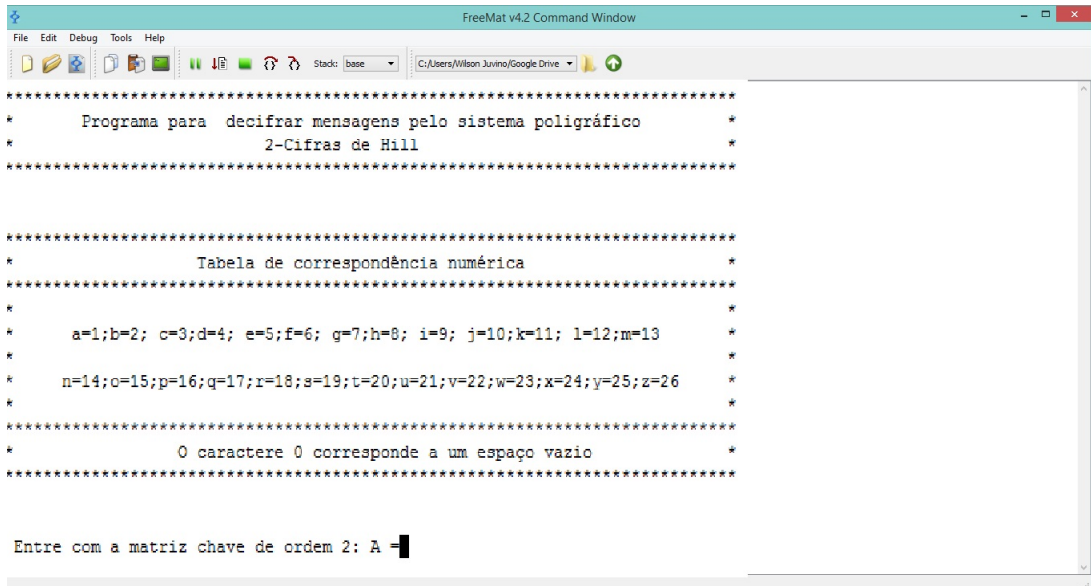


Figura 4.44: Desenvolvimento da Atividade 26, primeira etapa do programa

Passo 2: Inserir a matriz chave $A = \begin{bmatrix} 3 & 1 \\ -4 & 5 \end{bmatrix}$.

Para inserir a matriz chave, digita-se o comando:

[3 1; -4 5]

e, ao pressionar **Enter**, é solicitado para inserir a matriz M , cuja entradas são caracteres do texto comum, como mostra a Figura 4.45.

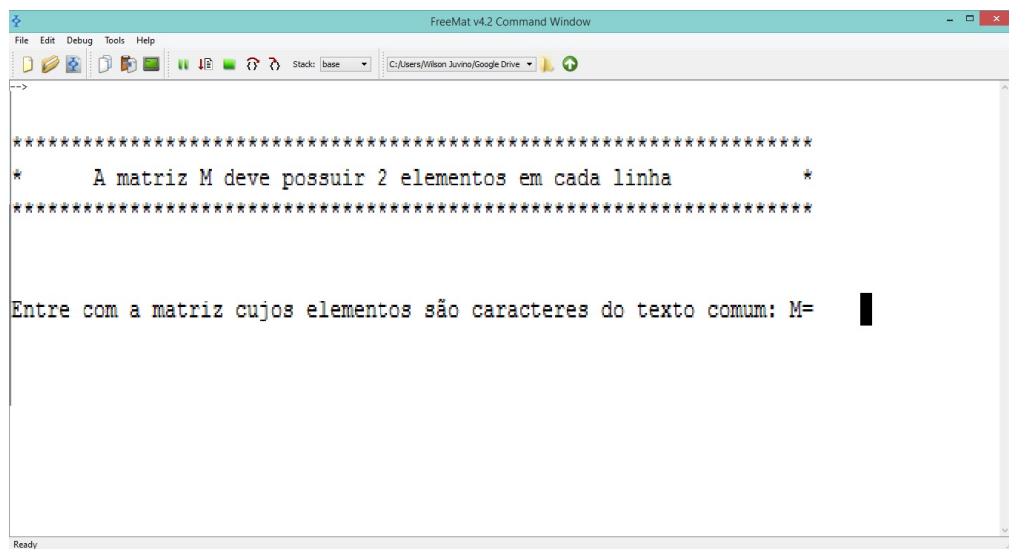


Figura 4.45: Desenvolvimento da Atividade 26, segunda etapa do programa

Passo 3: Inserir a matriz onde as entradas são caracteres da mensagem texto.

A matriz é inserida através do comando:

```
[m e; s t; r a;d o; 0 p; r o;f i; s s; i o;n a; l 0; e m; 0 m;
a t;e m;a t; i c;a 0;e m; 0 r; e d;e 0; n a;c i;o n; a l]
```

e, ao pressionar **Enter**, o programa reportará a matriz criptografada, apresentada na Figura 4.46.

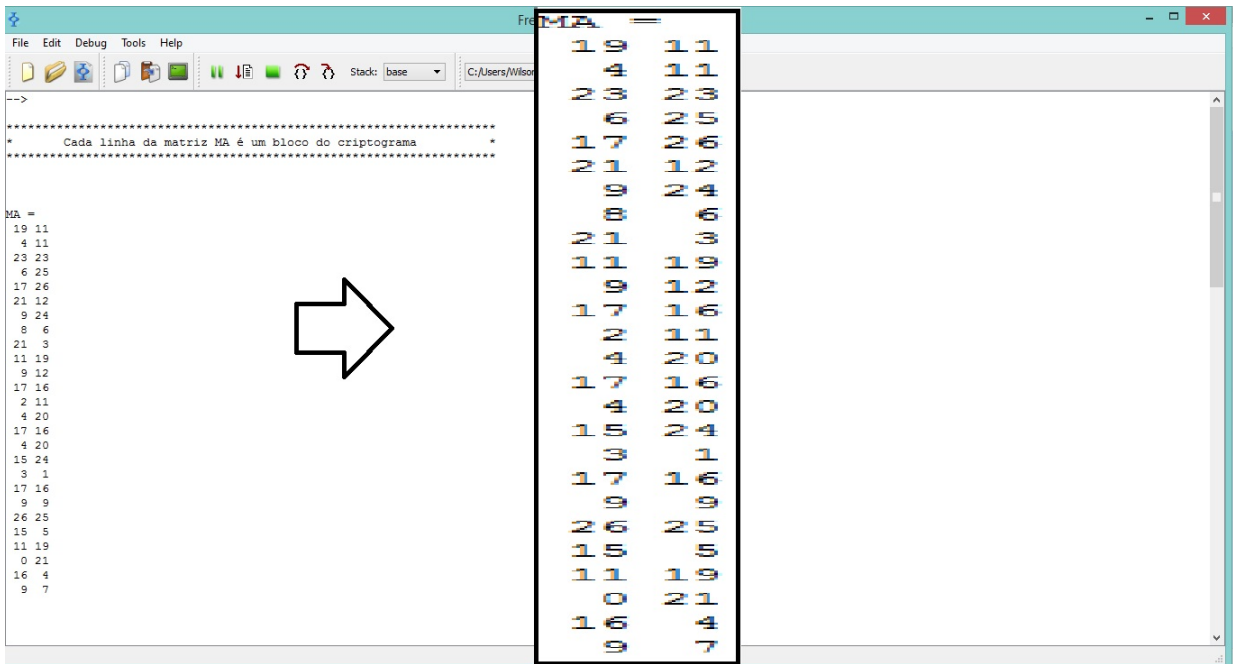


Figura 4.46: Desenvolvimento da Atividade 26, terceira etapa do programa

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir **nao** e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.44, caso contrário, se inserir **sim**, o programa é finalizado, sendo apresentado o display da Figura 4.47.

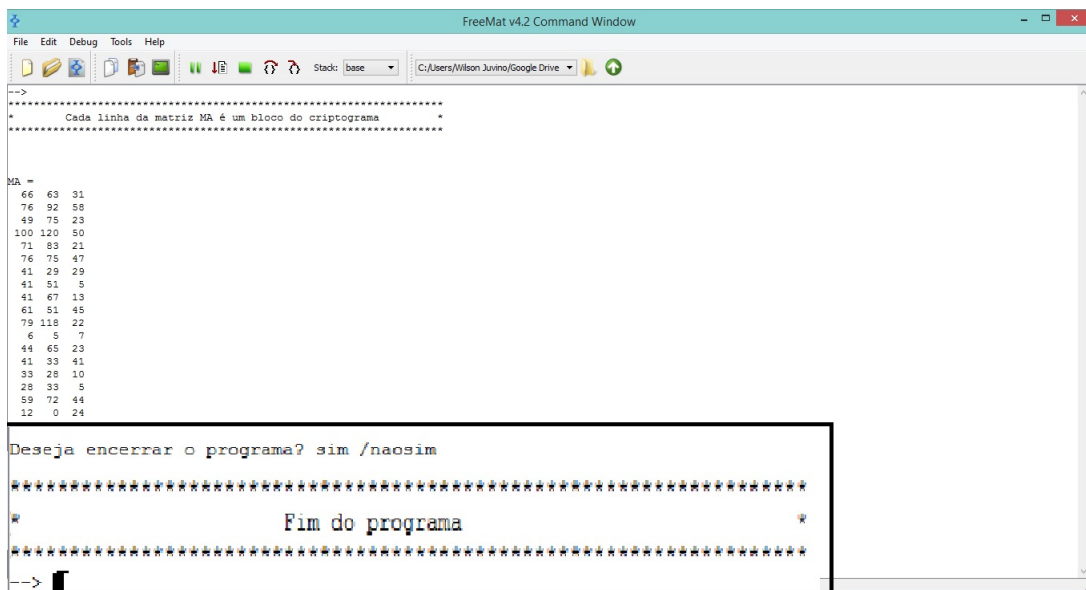


Figura 4.47: Desenvolvimento da Atividade 26, quarta etapa do programa

Passo 5: Apresentar a mensagem a ser transmitida

Agrupando as linhas da matriz MA obtida no Passo 3, em sequência, obtém-se:

```
19 11 4 11 23 23 6 25 17 26 21 12 9 24 8 6 21 3 11 19 9 12 17 16 2 11
4 20 17 16 4 20 15 24 3 1 17 16 9 9 26 25 15 5 11 19 0 21 16 4 9 7
```

Logo, utilizando a correspondência numérica da tabela, apresentada no display, obtém-se a mensagem a ser transmitida:

```
skdkwwfyqzulixhfucksilqpbkdtqpdttoxcaqpiizyoeks0updig
```

Atividade 27. Considere a matriz chave: $A = \begin{bmatrix} 3 & 1 \\ -4 & 5 \end{bmatrix} \in \mathcal{M}_2(\mathbb{Z}_{27})$. Por meio do programa “DECIFRARHILL2”, desenvolvido na Atividade 25, decifre o criptograma:

```
skdkwwfyqzulixhfucksilqpbkdtqpdttoxcaqpiizyoeks0updig
```

Passo 1: Abrir o programa.

Para abrir o programa, na janela de comandos, digita-se:

```
DECIFRARHILL2
```

Após, pressione **Enter** e o programa será compilado (ver Figura 4.48).

Passo 2: Inserir a matriz chave $A = \begin{bmatrix} 3 & 1 \\ -4 & 5 \end{bmatrix}$.

Para inserir a matriz chave, digita-se

```
[3 1; -4 5]
```

e, ao pressionar **Enter**, é solicitado inserir a matriz C onde as entradas são caracteres do criptograma, como mostra a Figura 4.49.

```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Users/Wilson Juvino/Google Drive
-->
*****
*      Programa para decifrar mensagens pelo sistema poligráfico      *
*      2-Cifras de Hill                                              *
*****

*****
*      Tabela de correspondência numérica                             *
*****
*
*      a=1;b=2; c=3;d=4; e=5;f=6; g=7;h=8; i=9; j=10;k=11; l=12;m=13  *
*
*      n=14;o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 *
*
*****
*      O caractere 0 corresponde a um espaço vazio                    *
*****

Entre com a matriz chave de ordem 2: A =
```

Figura 4.48: Desenvolvimento da Atividade 27, primeira etapa do programa

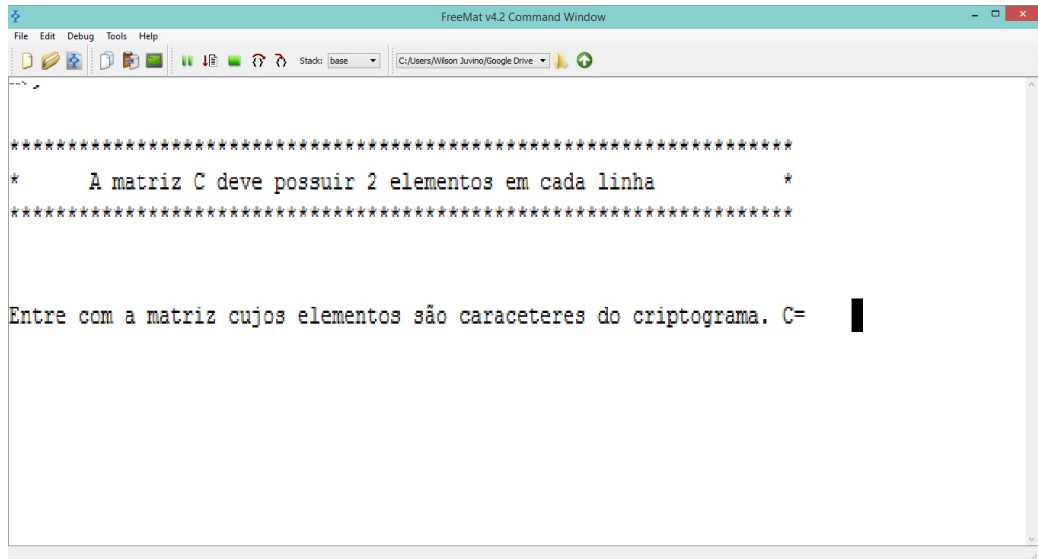


Figura 4.49: Desenvolvimento da Atividade 27, segunda etapa do programa

Passo 3: Inserir a matriz onde as entradas são caracteres do criptograma.

A matriz é inserida através do comando:

```
[s k;d k;w w;f y;q z;u l;i x;h f;u c;k s;i l;q p;b k;d t;q p;d t;o x;c a;q p;i i;z y;o e;
k s;0 u;p d;i g]
```

e, ao pressionar **Enter**, o programa reportará a matriz decifrada, apresentada na Figura 4.50.

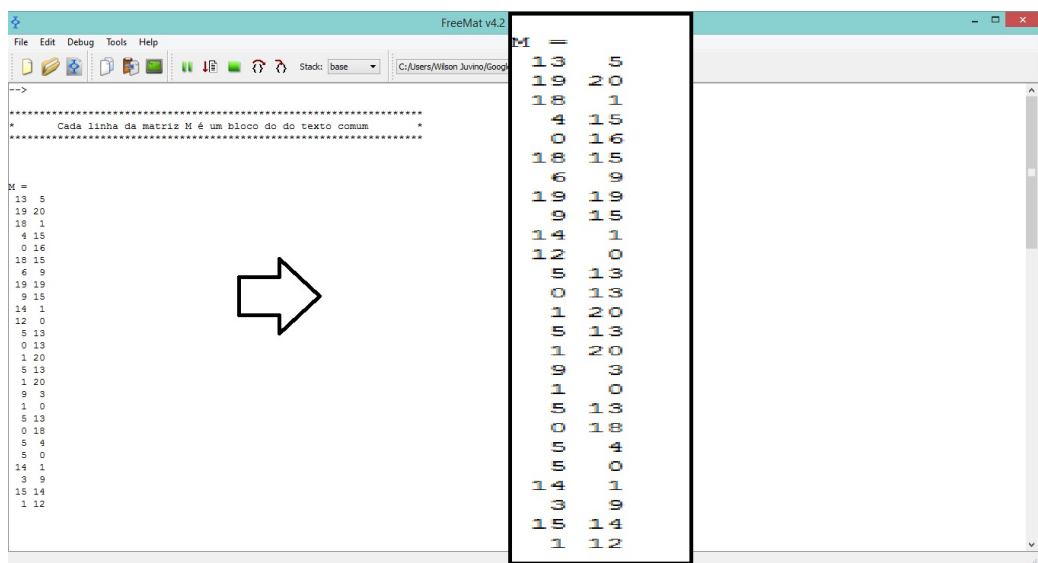


Figura 4.50: Desenvolvimento da Atividade 27, terceira etapa do programa

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.48, caso contrário, se inserir , o programa é finalizado, sendo apresentado o display da Figura 4.51.

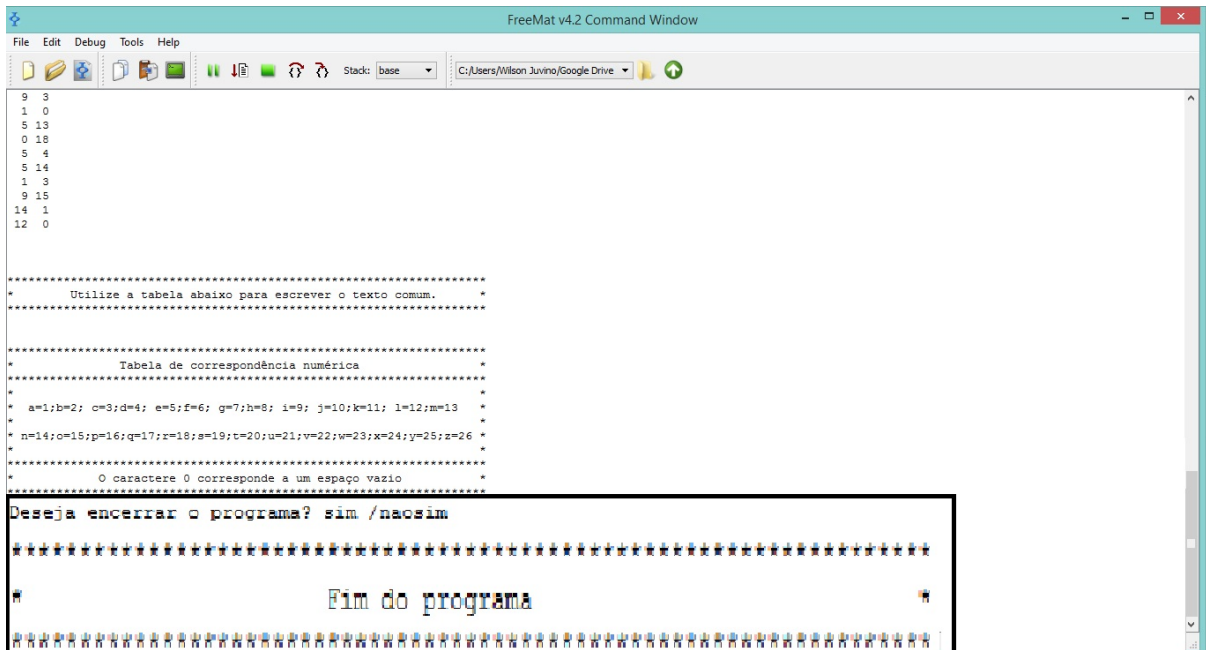


Figura 4.51: Desenvolvimento da Atividade 27, quarta etapa do programa

Passo 5: Apresentar a mensagem transmitida.

Agrupando as linhas da matriz M obtida no passo 3, em sequência, obtém-se:

```

13 5 19 20 18 1 4 15 0 16 18 15 6 9 19 19 9 15 14 1 12 0 5 13 0
13 1 20 5 13 1 20 9 3 1 0 5 13 0 18 5 4 5 0 14 1 3 9 15 14 1 12

```

Logo, utilizando a correspondência numérica da tabela, apresentada no display do programa, obtém-se:

```
mestrado0profissional0em0matematica0em0rede0nacional
```

Portanto, a mensagem transmitida era: “Mestrado Profissional em Matemática em Rede Nacional”.

4.3 Proposta III: Resolução de sistemas lineares de m equações com n incógnitas

Nesta seção, são descritas atividades para resolver sistemas lineares, onde, os métodos algébricos, utilizados para resolver estes sistemas, serão aplicados através do FreeMat. Ao final da seção, apresenta-se atividades para construir programas no FreeMat que reportem as soluções de sistemas lineares, expressos pela equação matricial $AX = B$.

A utilização do FreeMat, nesta proposta, busca facilitar o ensino de sistemas lineares no Ensino Médio, visto que, a resolução destes não é muito aprofundada no ensino básico.

Nas abordagens das atividades seguintes, para resolução de um sistema linear no FreeMat, será considerado a Regra Cramer e o método de Gauss-Jordan.

Regra de Cramer: Para um sistema de m equações e m incógnitas, a resolução do sistema, caso o determinante da matriz dos coeficientes seja diferente de zero ¹, se dará através dos seguintes passos:

Passo 1: Inserir a matriz dos coeficientes A e dos termos independentes B .

Passo 2: Verificar se a matriz A admite inversa, inserindo o comando:

$\det(A)$

Passo 3: Se o resultado reportado após a inserção do comando

$\det(A)$

 for diferente de zero, determina-se a solução (única) do sistema inserindo o comando:

$X=\text{inv}(A)*B$

Método de Gauss- Jordan: Para um sistema de m equações e n incógnitas, a resolução do sistema se dará através dos seguintes passos:

Passo 1: Inserir a matriz dos coeficientes A e dos termos independentes B .

¹Lembre-se que, se o determinante da matriz dos coeficientes for igual a zero, a Regra de Cramer é inconclusiva.

Passo 2: Determinar a matriz ampliada $C = [A|B]$ inserindo o comando:

```
C=[A B]
```

Passo 3: Determinar os postos da matriz A e B , respectivamente, através da inserção dos comandos:

```
rank(A)  
rank(C)
```

Passo 4: Se o sistema é compatível, determina-se a solução através do comando:

```
rref(C)
```

Atividade 28. Através do FreeMat, utilizando a Regra de Cramer, determine, se for possível, a solução do seguinte sistema linear:

$$\begin{cases} x + 2z = 7 \\ 2x - y + 3z = 9 \\ 4x + y + 8z = 30. \end{cases}$$

Objetivo: Aplicar a Regra de Cramer na resolução de sistemas lineares.

Passo 1: Definir a matriz dos coeficientes A e a matriz dos termos independentes B .

As matrizes A e B são inseridas no FreeMat, através dos respectivos comandos:

```
A=[1 0 2; 2 -1 3; 4 1 8]  
B=[7; 9; 30]
```

Passo 2: Calcular o determinante da matriz A .

Calcula-se o $\det(A)$ inserindo o comando:

```
det(A)
```

Como $\det(A) = 1 \neq 0$ a matriz A é inversível e o sistema admite única solução.

Passo 3: Determinar a solução do sistema.

Através da Regra de Cramer, a solução do sistema é dada pelos elementos da matriz X , que é determinada através do comando:

$$X = \text{inv}(A) * B$$

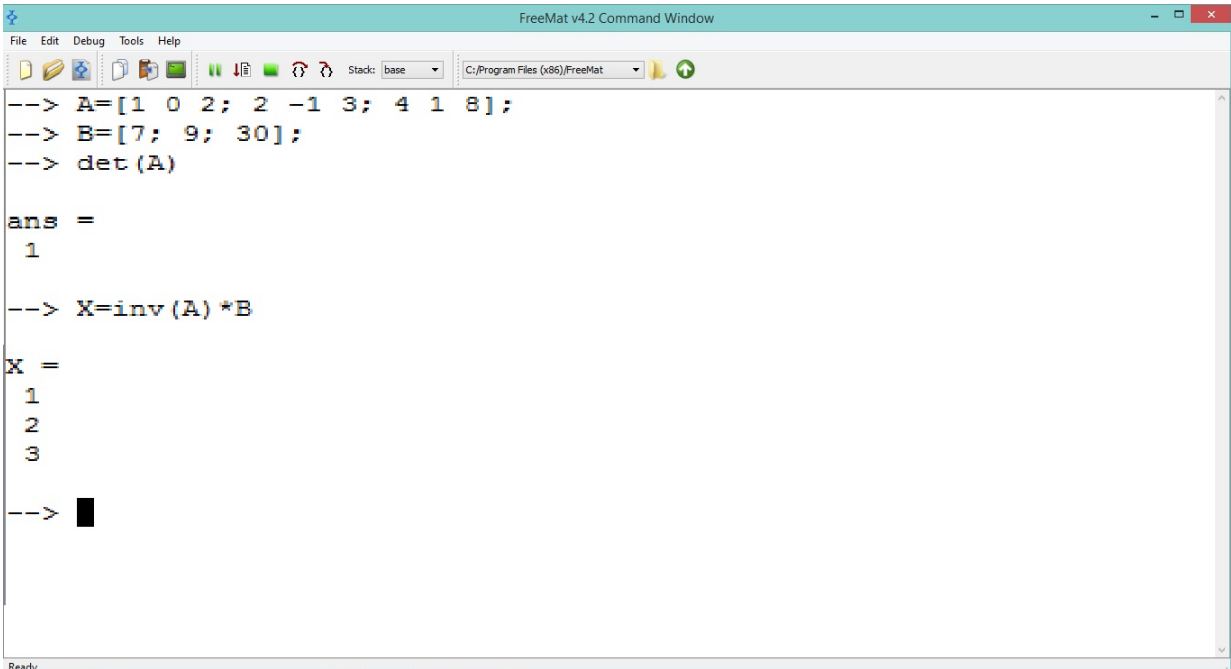
Na Figura 4.52, é apresentada a matriz X .

Passo 4: Explicitar a solução do sistema.

O programa nos fornece os valores:

$$x = 1, \quad y = 2, \quad z = 3.$$

Portanto, o conjunto solução do sistema linear dado é $\{(1, 2, 3)\}$.



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Program Files (x86)/FreeMat
--> A=[1 0 2; 2 -1 3; 4 1 8];
--> B=[7; 9; 30];
--> det(A)
ans =
  1
--> X=inv(A)*B
X =
  1
  2
  3
--> █
Ready
```

Figura 4.52: Desenvolvimento da Atividade 28 no FreeMat

Atividade 29. Foram estudados três tipos de alimentos. Fixada a mesma quantidade de alimento, determinou-se que:

- (i) o alimento I tem 1 unidade de vitamina A, 3 unidades de vitamina B e 4 unidades de vitamina C;
- (ii) o alimento II tem 2 unidades de vitamina A, 3 unidades de vitamina B e 5 unidades de vitamina C;
- (iii) o alimento III tem 3 unidades de vitaminas A, 1 unidade de vitamina C e não contém vitamina B.

Se são necessários 15 unidades de vitamina A, 6 de vitamina B e 13 de vitamina C, determine, se for possível, a quantidade dos alimentos I, II e III, que fornecem a quantidade de vitaminas desejadas.

Objetivo: Apresentar uma aplicação dos sistemas lineares e do método de Gauss-Jordan.

Passo 1: Caracterizar o problema.

Sejam x, y e z , a quantidade dos alimentos I, II e III, respectivamente. Então o problema, em questão, é representado pelo sistema linear:

$$\begin{cases} 1x + 2y + 3z = 15 \\ 3x + 3y + 0z = 6 \\ 4x + 5y + 1z = 13. \end{cases}$$

Passo 2: Inserir no FreeMat a matriz dos coeficientes A , a matriz dos termos independentes B , e a matriz ampliada $C = [A|B]$ do sistema linear obtido no Passo 1.

As matrizes A , B e C são inseridas no FreeMat, através dos seguintes comandos:

```
A=[1 2 3;3 3 0; 4 5 1]
B=[15; 6; 13]
C=[ A  B]
```

Passo 3: Verificar se o sistema admite solução.

Determina-se o posto das matrizes A e C para analisar se o sistema possui solução, para isto, digita-se os comandos:

```
rank(A)
rank(C)
```

Neste caso, obtém-se $\text{posto}(A) = \text{posto}(C) = 3$, como o número de incógnitas também é 3, o sistema é compatível e determinado.

Passo 4: Determinar a solução do sistema, exibindo a matriz reduzida por linha à forma escada.

Determina-se a matriz reduzida por linha à forma escada da matriz $C = [A|B]$, digitando o comando:

```
D=rref(C)
```

obtendo-se a matriz $D = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 4 \end{bmatrix}$, apresentada na Figura 4.53.

Passo 5: Explicitar a solução do sistema.

A solução do sistema é dada pelos elementos da última coluna da matriz D , sendo estes:

$$x = 1, \quad y = 1, \quad z = 4.$$

Portanto, o conjunto solução do sistema linear é $\{(1, 1, 4)\}$.

Na Figura 4.53, é apresentado o desenvolvimento da Atividade 29 no FreeMat.

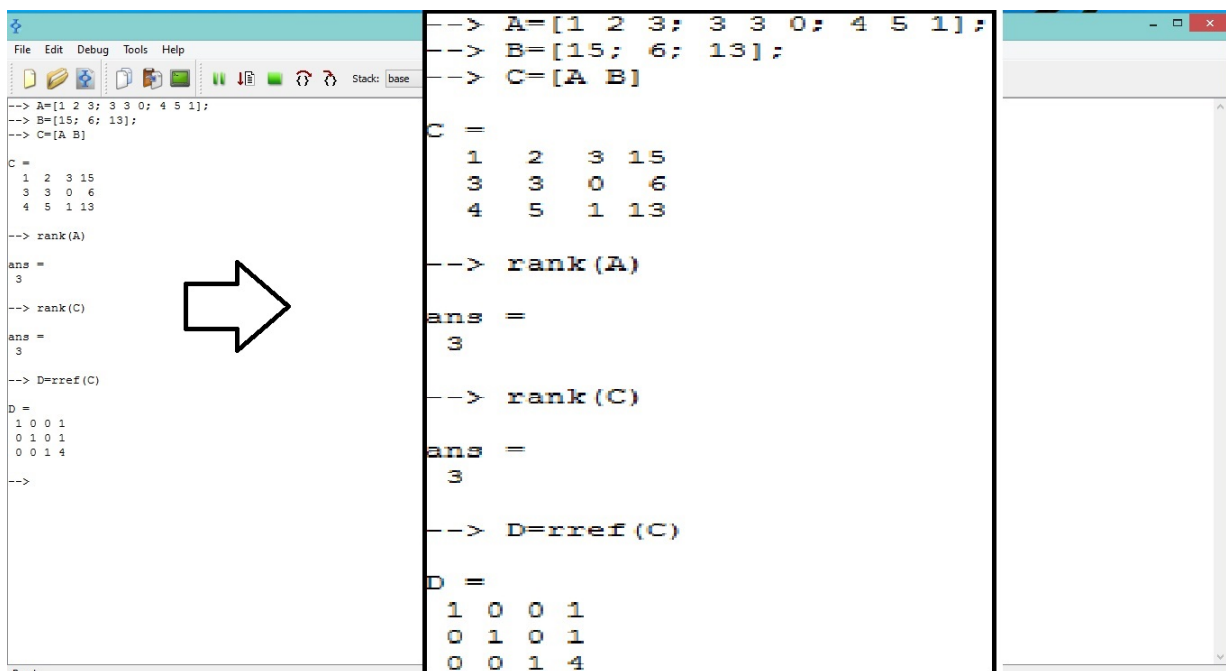


Figura 4.53: Desenvolvimento da atividade 29 no FreeMat

Passo 6: Determinar a quantidade de alimentos.

Pelo Passo 5, concluímos que, necessita-se de 1 quantidade do alimento I, 1 quantidade do alimento II e 4 quantidades do alimento III.

Atividade 30. Se o sistema linear que caracterizasse o problema da Atividade 29 fosse da seguinte forma:

$$\begin{cases} 1x + 2y + 3z = 15 \\ 3x + 3y + 0z = 6 \\ 4x + 5y + 3z = 13, \end{cases}$$

onde x , y e z representam a quantidade de elementos I, II e III, respectivamente. Existiria uma quantidade de alimentos I, II e III, que forneceriam as mesmas quantidades de vitaminas desejadas?

Passo 1: Definir a matriz dos coeficientes A , a matriz dos termos independentes B , e a matriz ampliada $C = [A|B]$.

As matrizes A , B e C são inseridas no FreeMat, através dos seguintes comandos:

```
A=[1 2 3; 3 3 0; 4 5 3]
B=[15;6; 13]
C=[ A B]
```

Passo 2: Verificar se o sistema admite solução.

Determina-se o posto das matrizes A e C para analisar se o sistema possui solução.

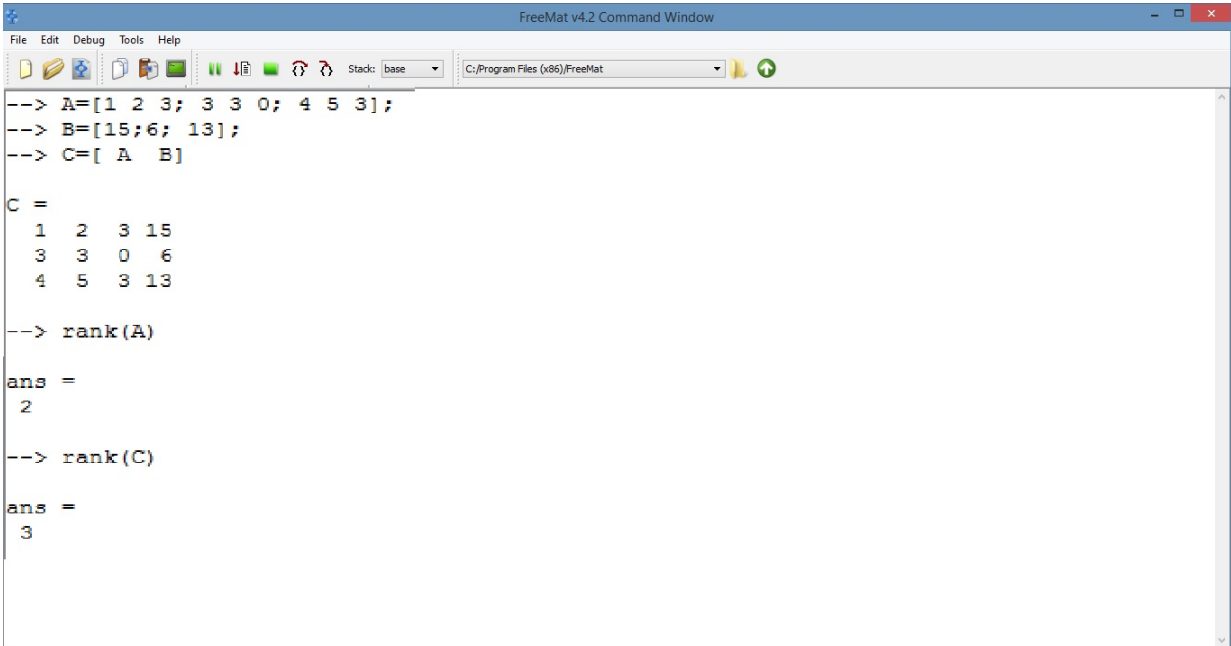
Para isto, digita-se os seguintes comandos:

```
rank(A)
rank(C)
```

Neste caso, obtém-se $\text{posto}(A) = 2$ e $\text{posto}(C) = 3$. Como $\text{posto}(A) < \text{posto}(C)$, então o sistema é incompatível.

Portanto, não existe uma quantidade de alimentos x, y e z que forneceria a quantidade de vitaminas desejada.

Na Figura 4.54, é apresentado o desenvolvimento da Atividade 30 no FreeMat.



```
FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Program Files (x86)/FreeMat
--> A=[1 2 3; 3 3 0; 4 5 3];
--> B=[15;6; 13];
--> C=[ A B]

C =
  1  2  3 15
  3  3  0  6
  4  5  3 13

--> rank(A)

ans =
  2

--> rank(C)

ans =
  3
```

Figura 4.54: Desenvolvimento da Atividade 30 no FreeMat

Atividade 31. Resolva o sistema linear:

$$\begin{cases} 1x + 2y - 2z + 3w = 2 \\ 2x + 4y - 3z + 4w = 5 \\ 5x + 10y - 8z + 11w = 12. \end{cases} \quad (4.2)$$

Passo 1: Definir a matriz dos coeficientes A , a matriz dos termos independentes B , e a matriz ampliada $C = [A|B]$.

As matrizes A , B e C são inseridas no FreeMat, através dos seguintes comandos:

```
A= [1 2 -2 3; 2 4 -3 4; 5 10 -8 11]
B= [2; 5; 12]
C= [ A B]
```

Passo 2: Verificar se o sistema admite solução.

Determina-se o posto das matrizes A e C para analisar se o sistema possui solução, para isto, digita-se os comandos:

```
rank(A)
rank(C)
```

Neste caso, obtém-se $\text{posto}(A) = 2$ e $\text{posto}(C) = 2$. Como $\text{posto}(A) = \text{posto}(C)$ e o número de incógnitas é $4 > 2$, o sistema é compatível e indeterminado e possui $4 - 2 = 2$ parâmetros livres.

Passo 3: Determinar a solução do sistema, exibindo a matriz reduzida por linha à forma escada.

Determina-se a matriz reduzida à forma escada da matriz $C = [A|B]$, digitando o comando:

```
D=rref(C)
```


obtendo-se a matriz $D = \begin{bmatrix} 1 & 2 & 0 & -1 & 4 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

Passo 4: Explicitar a solução do sistema.

Pelo Passo 3, o sistema (4.2) é equivalente ao sistema:

$$\begin{cases} x + 2y - w = 4 \\ z - 2w = 1 \end{cases} \quad (4.3)$$

Como existem dois parâmetros livres, escolhemos $y = t$ e $w = u$ com $t, u \in \mathbb{R}$. Substituindo no sistema (4.3), obtemos $z = 1 + 2u$ e $x = 4 + u - 2t$.

Portanto, o conjunto solução do sistema linear dado é $\{(4 + u - 2t, t, 1 + 2u, u); t, u \in \mathbb{R}\}$.

Na Figura 4.55, é apresentado o desenvolvimento da Atividade 31 no FreeMat.

```

--> A= [1 2 -2 3; 2 4 -3 4; 5 10 -8 11];
--> B= [2; 5; 12];
--> C=[ A B]
C =
  1  2 -2  3  2
  2  4 -3  4  5
  5 10 -8 11 12

--> rank(A)
ans =
  2

--> rank(C)
ans =
  2

--> rref(C)
ans =
  1  2  0 -1  4
  0  0  1 -2  1
  0  0  0  0  0

--> rank(A)
ans =
  2

--> rank(C)
ans =
  2

--> rref(C)
ans =
  1  2  0 -1  4
  0  0  1 -2  1
  0  0  0  0  0
  
```

Figura 4.55: Desenvolvimento da Atividade 31 no FreeMat

Atividade 32. Utilizando a Regra de Cramer, construa um programa no FreeMat e o intitule de ‘CRAMER’, para determinar, se for possível, a solução de um sistema linear de m equações com m incógnitas, dada por $AX = B$, onde A é a matriz dos coeficientes e B é a matriz dos termos independentes. O programa deve fazer o seguinte: a) se $\det(A) \neq 0$ for diferente de zero, o programa deve determinar explicitamente a solução do sistema linear; b) se $\det(A) = 0$, o programa deve reportar a seguinte mensagem para o usuário: “O determinante da matriz A é zero, informações sobre as soluções são inconclusivas”.
Objetivo: Desenvolver um programa, através da Regra de Cramer, que apresente as soluções de um sistema linear compatível e determinado.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘CRAMER’ como nome para o programa.

No menu do FreeMat, selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function CRAMER
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda `não`, na linha abaixo dos comandos, definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
while reiniciar==0
```

Passo 3: Criar um display com informações referente a aplicação do programa.

Cria-se o display inserindo o comando:

```
clc
disp ( ' ' )
disp ( '* Programa para obter solução de sistemas lineares de *' )
disp ( '* m equações com m incógnitas *' )
disp ( '* Regra de Cramer *' )
```

Passo 4: Definir comandos para o usuário inserir a matriz dos coeficientes

Para selecionar a entrada da matriz A e informar ao usuário quais matrizes devem ser inseridas digita-se o comando:

```
A=input('Entre com a matriz dos coeficientes: A =')
```

Passo 5: Definir comandos para que seja reportada a matriz A e o seu determinante.

Para que o programa reporte a matriz A e seu determinante, digita-se os comandos:

```
clc
disp ('* Matriz A*')
A=A
disp ('* Determinante da matriz A*')
detA=det(A)
```

Passo 6: Definir condições e parâmetros de decisão, de acordo com o valor do determinante da matriz inserida pelo usuário, para determinar a solução do sistema ou reportar a mensagem dada no enunciado.

Passo 6.1: Inserir comandos para executar ações se o determinante da matriz A é diferente de zero:

```
if det(A) ~ =0
B=input(' Para determinar a solução, entre com a matriz dos termos
independentes: B =')
clc
disp ('* Solução do sistema linear*')
X=inv(A)*B
continuar=input('Deseja revelar cada incógnita e seu valor?sim/nao')
```

Passo 6.2: Inserir comandos para executar ações se o determinante da matriz A é zero:

```
else det(A)==0
disp ('* Como det(A)=0, informações sobre as soluções *')
disp ('* do sistema linear são inconclusivas *')
reiniciar=input(' Deseja encerrar o programa?  sim / nao ');
disp ('* Fim do Programa *')
end
end
```

Passo 7: Utilizar o laço de repetição `for` para apresentar as incógnitas e seus respectivos valores.

Para reportar as incógnitas, caso o usuário deseje, na linha abaixo dos comandos inserido no Passo 6.2, digita-se os seguintes comandos:

```
clc
disp ( ' ')
disp ('*Conjunto solução do sistema linear {(x1, x2,...,xi,...,xm)}:*)')
[m n]= size(X);
for i=1:m
disp ('*Valor da incógnita xi*')
i=i
xi= X(i)
end
disp ('*Fim do programa*')
```

Passo 8: Salvar o programa.

Para salvá-lo, na barra do Menu do editor, selecione a opção *File* e clique em *Save*. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat).

O arquivo deve ser nomeado como “CRAMMER” e salvo na extensão `.m`, assim o

arquivo final será: CRAMMER.m. Na Figura 4.56, apresenta-se parte dos comandos relativos a este programa.

Observação 4.13. O programa é apresentado integralmente no Apêndice A.6, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

```

function CRAMER
continuar=0;
clc
while continuar==0
disp(' ')
disp('*****')
disp(' Programa para obter solução de sistemas lineares com m equações e m incógnitas *')
disp('**                               Método de Cramer                               **')
disp('*****')
A=input('Entre com a matriz dos coeficientes: A =')
if det(A)==0
continuar=1;
else
disp(' ')
disp('*****')
disp(' O determinante da matriz A é zero, informações sobre as soluções são inconclusivas,')
disp('**                               Fim do programa                               **')
disp('*****')
elseif det(A)~=0
clc
disp(' ')
disp('*****')
disp('**                               Solução do sistema                               **')
disp('*****')
X=inv(A)*B
[m n]=size(X);
continuar=input('Descontinuar? (1 para sim, 0 para não) ');
if continuar==1
disp(' ')
disp('*****')
disp('**                               Fim do programa                               **')
disp('*****')
end
end

```

Figura 4.56: Editor do FreeMat e parte dos comandos relativos ao programa “CRAMER”

Atividade 33. Utilize o programa ‘CRAMER’, desenvolvido na Atividade 32, para determinar, caso exista, a solução única do sistema:

$$\begin{cases} 3x + 2y + 3z = 32 \\ 3x - y = 2 \\ x - 2y + z = 0. \end{cases}$$

Passo 1: Abrir o programa ‘CRAMER’.

Para abrir o programa, na janela de comandos digita-se:

CRAMER

Após, pressione **Enter** e o programa será compilado (ver Figura 4.57).

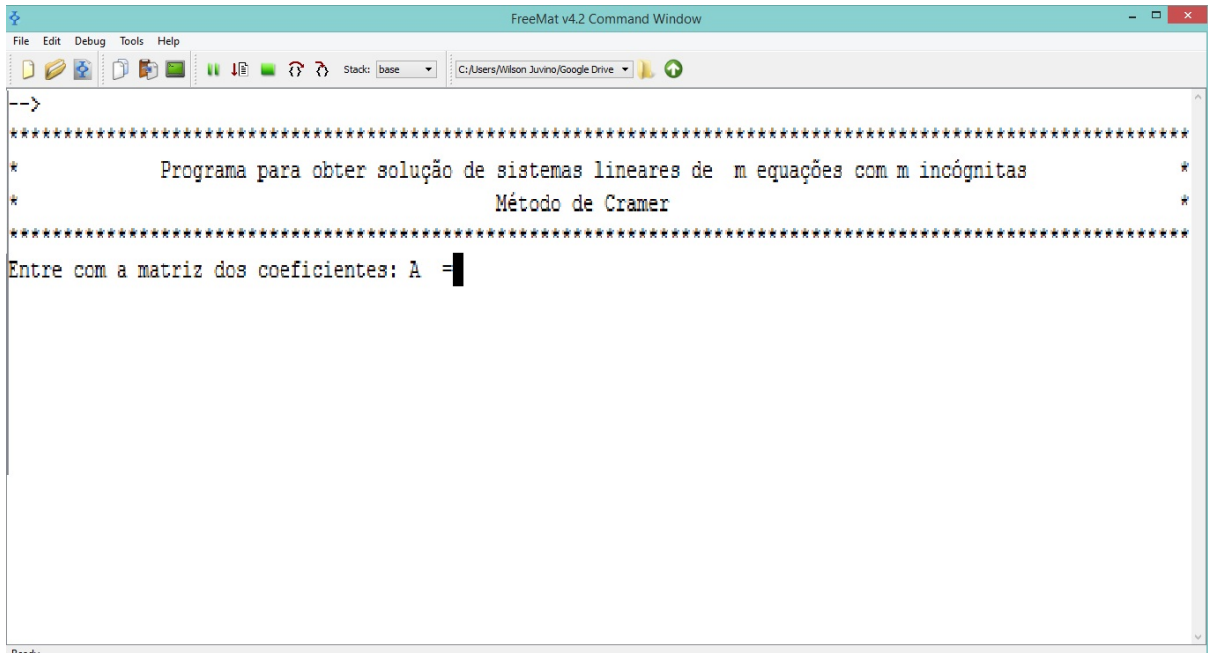


Figura 4.57: Desenvolvimento da Atividade 33, primeira etapa do programa CRAMER

Passo 2: Inserir a matriz dos coeficientes $A = \begin{bmatrix} 3 & 2 & 3 \\ 3 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}$.

Para inserir a matriz dos coeficientes, digita-se:

[3 2 3; 3 -1 0; 1 -2 1]

Após pressionar **Enter**, o programa irá reportar a matriz A e seu determinante, também, irá solicitar para inserir a matriz B , como mostra a Figura 4.58.

Como $\det A \neq 0$, o programa solicita a inserção da matriz dos termos independentes.

Passo 3: Inserir a matriz dos termos independentes.

Ao solicitar a inserção da matriz dos termos independentes, digita-se o comando:

[32; 2; 0]

Ao pressionar **Enter**, o programa reportará a solução do sistema, conforme apresentado na Figura 4.59.

```

*****
*                               Matriz A:                               *
*****
A =
 3  2  3
 3 -1  0
 1 -2  1

*****
*                               Determinante da matriz A:              *
*****
detA =
-24

*****
*                               Como det(A) é diferente de zero, o sistema possui solução única
*****
Para determinar a solução, entre com a matriz dos termos independentes: B =

```

Figura 4.58: Desenvolvimento da 33, segunda etapa do programa CRAMER

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Users/Wilson Juvino/Google Drive
-->
*****
*                               Solução do sistema linear:              *
*****
X =
 2.0000
 4.0000
 6.0000

Deseja revelar cada incógnita e o seu respectivo valor? sim /nao

```

Figura 4.59: Desenvolvimento da Atividade 33, terceira etapa do programa CRAMER

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.57, caso contrário, se inserir , o programa reportará cada incógnita com seu respectivo valor, sendo apresentado o display da Figura 4.60.

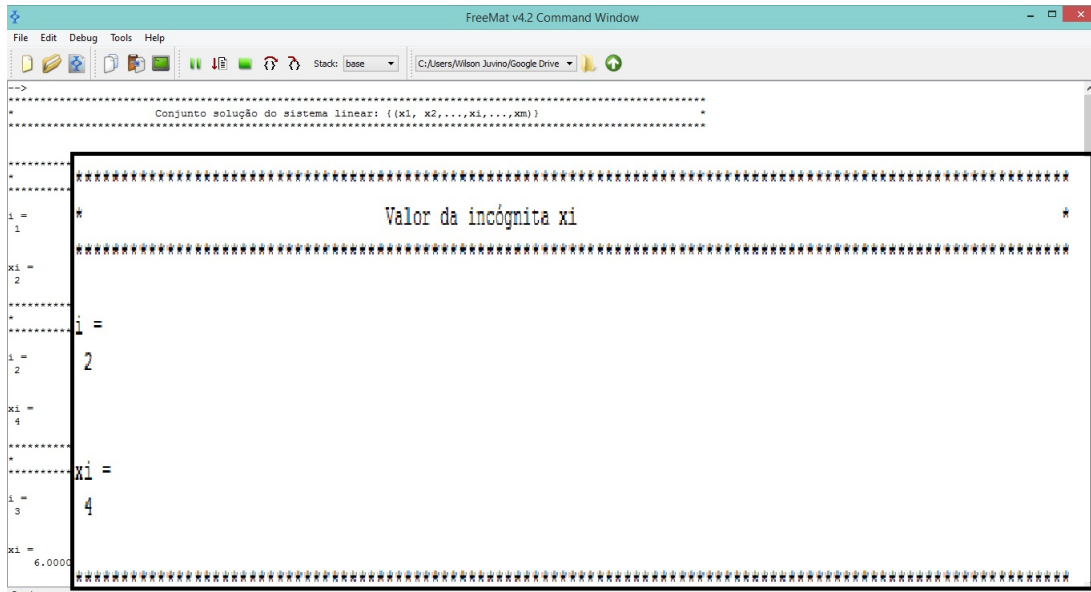


Figura 4.60: Desenvolvimento da Atividade 33, quarta etapa do programa CRAMER

Passo 5: Explicitar a solução do sistema.

Pelo Passo 3, a solução do sistema é dada pelos elementos da matriz X , sendo estes:

$$x = 2, \quad y = 4, \quad z = 6.$$

Portanto, o conjunto solução do sistema linear dado é $\{(2, 4, 6)\}$.

Atividade 34. Utilize o programa ‘CRAMER’, desenvolvido na Atividade 32, para determinar, caso exista, a solução única do sistema:

$$\begin{cases} 3x + 2y + 3z = 32 \\ 6x - 4y + 6z = 2 \\ x - 2y + z = 0. \end{cases}$$

Passo 1: Abrir o programa ‘CRAMER’.

Para abrir o programa, na janela de comandos digita-se:

CRAMER

Após, pressione **Enter** e o programa será compilado (ver Figura 4.61).

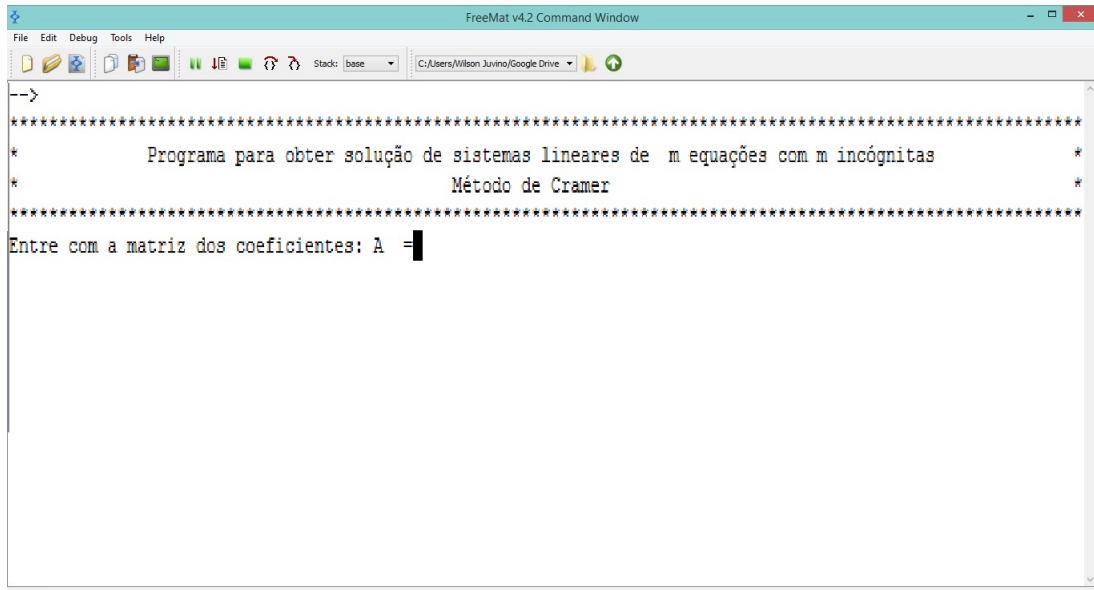


Figura 4.61: Desenvolvimento da Atividade 34, primeira etapa do programa CRAMER

Passo 2: Inserir a matriz dos coeficientes $A = \begin{bmatrix} 3 & 2 & 3 \\ 6 & -4 & 6 \\ 1 & -2 & 1 \end{bmatrix}$.

Para inserir a matriz dos coeficientes, digita-se:

[3 2 3; 6 -4 6; 1 -2 1]

Após pressionar **Enter**, o programa reportará algumas informações sobre o determinante da matriz A e da sua solução do sistema, conforme apresentado na Figura 4.62.

Como $\det A = 0$, informações sobre as soluções do sistema linear são inconclusivas.

Passo 3: Reinicializar ou encerrar o programa.

Ao inserir nao e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.61, caso contrário, se inserir sim, o programa será encerrado, sendo apresentado o display da Figura 4.63.

```

*****
*****                               Matriz A:                               *****
*****
A =
3  2  3
6 -4  6
1 -2  1

*****
*****                               Determinante da matriz A:                               *****
*****
detA =
0

*****
*****                               Como det(A)=0, informações sobre as soluções do sistema linear são inconclusivas                               *****
*****

Deseja encerrar o programa? sim / nao

```

Figura 4.62: Desenvolvimento da Atividade 34, segunda etapa do programa CRAMER

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
C:/Users/Wilson Juvino/Google Drive

*****
*****                               Matriz A:                               *****
*****
A =
3  2  3
6 -4  6
1 -2  1

*****
*****                               Determinante da matriz A:                               *****
*****
detA =
0

*****
*****                               Como det(A)=0, informações sobre as soluções do sistema linear são inconclusivas                               *****
*****

Deseja encerrar o programa? sim / nao sim

*****
*****                               Fim do Programa                               *****
*****
-->

```

Figura 4.63: Desenvolvimento da Atividade 34, terceira etapa do programa CRAMER

Atividade 35. Utilizando o método de Gauss-Jordan, construa um programa no FreeMat e o intitule de ‘GAUSSJORDAN’, para determinar, se for possível, a solução de um sistema linear de m equações com n incógnitas, dada por $AX = B$, onde A é a matriz dos coeficientes e B é a matriz dos termos independentes. O programa deve fazer o seguinte: a) se o sistema for compatível e determinado, o programa deve determinar explicitamente a solução do sistema linear; b) se o sistema for compatível e indeterminado, o programa deve determinar a matriz linha escada reduzida, da matriz ampliada $C = (A|B)$, e o número de parâmetros livres; c) se o sistema for incompatível, o programa deve reportar a seguinte mensagem para o usuário: “Como $\text{posto}(A) < \text{posto}(C)$ o sistema linear $AX = B$ é incompatível”.

Objetivo: Desenvolver um programa, através do método de Gauss-Jordan, que apresenta soluções de um sistema linear.

Passo 1: Abrir o editor de texto do FreeMat e definir ‘GAUSSJORDAN’ como nome para o programa.

No menu do FreeMat, selecione a opção `editor`. Aberta a janela de edição, na linha 1, digita-se o comando:

```
function GAUSSJORDAN
```

Passo 2: Utilizar o laço de repetição *while* para que, ao serem apresentadas perguntas que permitam a entrada de dois tipos de respostas (sim/não), de acordo com a resposta inserida, o programa seja reinicializado ou, execute ações e encerre-se.

Para que o programa seja reinicializado, caso o usuário responda `não`, na linha abaixo dos comandos definidos no Passo 1, digita-se:

```
sim=1; nao=0; reiniciar=0;
reiniciar=0
while reiniciar==0
```

Passo 3: Criar um display com informações referente a aplicação do programa.

Cria-se o display, inserindo o comando:

```
clc
disp(' ')
disp ('*Programa para obter solução de sistemas lineares de*')
disp ('*m equações com n incógnitas*')
disp ('*Método de Gauss-Jordan*')
```

Passo 4: Definir comandos para o usuário inserir a matriz dos coeficientes A .

Para definir a entrada da matriz A e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
A=input('Entre com a matriz dos coeficientes: A=')
```

Passo 5: Definir comandos para que, a partir da matriz A , seja reportado o número de equações e incógnitas do sistema linear.

Para que o programa reporte o número de equações e incógnitas do sistema linear, digita-se os comandos:

```
[m, n]= size(A);
clc
disp('* m: Número de equações do sistema *')
m=m
disp('* n: Número de incógnitas xi *')
n=n
```

Passo 6: Definir comandos para o usuário inserir a matriz dos termos independentes B .

Para definir a entrada da matriz B e informar ao usuário quais matrizes devem ser inseridas, digita-se o comando:

```
B=input('Entre com a matriz dos termos independentes: B=')
clc
```

Passo 7: Inserir comandos para que seja reportado ao usuário a matriz ampliada $C = [A|B]$, o posto da matriz A e o posto da matriz C .

Para reportar a matriz C , o posto(A) e o posto(C), digita-se os comandos:

```
disp ('* Matriz ampliada C: *')
C= [A B]
disp('* Posto da matriz A: *')
postoA=rank(A)
disp('* Posto da matriz C: *')
postoC=rank(C)
```

Passo 8: Definir as condições e parâmetros de decisão, de acordo com o posto da matriz A e da matriz C , para determinar o tipo de sistema e a solução do sistema ou, caso o sistema seja incompatível, reportar a mensagem dada no enunciado.

Passo 8.1: Inserir comandos para executar ações se o sistema é compatível e determinado

```
if rank(A)==rank(C) & rank(A)==n
disp('* Como posto(A)=posto(C) e posto (A)= n
o sistema é compatível e determinado *')
disp('* Matriz C na forma linha escada reduzida *')
Escada=rref (C)
continuar=input('Deseja revelar cada incógnita e
seu respectivo valor?  sim / nao *')
```

Passo 8.2: Inserir comandos para executar ações se o sistema é incompatível.

```
if rank(A)< rank(C)
disp('* Como posto(A) < posto (C) o sistema é incompatível *')
continuar=input('*Deseja encerrar o programa?  sim / nao*')
disp('* Fim do Programa *')
```

Passo 8.3: Inserir comandos para executar ações se o sistema é compatível e

indeterminado.

```
elseif rank(A)==rank(C) & rank(A)<n
disp(' ')
disp('* Como posto(A)=posto(C) e posto (A)< n
o sistema é compatível e indeterminado *')
disp('* Número de parâmetros livres do sistema: *')
n-rank(A)
disp('* Matriz C na forma linha escada reduzida *')
Escada=rref (C)
disp('* Atribua quaisquer valores reais para cada
parâmetro livre e substitua no sistema *')
disp('* escalonado, após, faça substituições regressivas
obtendo o conjunto solução. *')
continuar=input('*Deseja encerrar o programa? sim / nao*')
disp('* Fim do programa *')
end
end
```

Passo 9: Caso o sistema seja compatível e determinado, utilizar o laço de repetição *for* para reportar as incógnitas e seus respectivos valores.

Para reportar as incógnitas caso o usuário deseje, na linha abaixo aos comandos declarados no Passo 7.3, digita-se os seguintes comandos:

```
clc
disp (' ')
disp ('* Conjunto solução do sistema linear:(x1, x2,...,xi,...,xm)*')
[m n]=size(Escada);
for i=1:m
disp ('* Valor da incógnita xi*')
i=i
Xi=Escada(m*(n-1)+1*i)
end
disp('*Fim do programa*')
```

Passo 10: Salvar o programa.

Para salvá-lo, na barra do Menu do editor, selecione a opção *File* e clique em *Save*. Após selecionado, abrirá uma nova janela onde deve-se escolher um diretório para salvar o arquivo (escolha preferencialmente o diretório onde esteja salvo o FreeMat). O arquivo deve ser nomeado como “GAUSSJORDAN” e salvo na extensão *.m*, assim o arquivo final será: GAUSSJORDAN.m. Na Figura 4.64, apresenta-se parte dos comandos relativos a este programa.

```

function GAUSSJORDAN
    clc
    continuar=0;
    while continuar==0
        disp(' ')
        disp('*****')
        disp('*           Programa para obter solução de sistemas lineares com')
        disp('*****')
        disp('*                               Método de Gauss-Jordan')
        disp('*****')
        A=input(' Entre com a matriz dos coefic');
        [m, n]=size(A);
        clc
        disp(' ')
        disp(' ')
        A=input(' Entre com a matriz dos coeficientes: A = ');
        [m, n]=size(A);
        clc
        disp(' ')
        disp(' ')
        disp('*****')
        disp('*           m: Número de equações do sistema           *')
        disp('*****')
        B=input('Entre com a matriz dos termos');
        clc
        disp(' ')
        disp('*           Matriz amplia')
        C=[A B];
        disp(' ')
        disp('*           Posto da matriz')
        rank(A)
        disp(' ')
        disp('*           n: Número de incógnitas xi')
        disp('*****')
    end

```

Figura 4.64: Editor do FreeMat e parte dos comandos relativos ao programa “sistemas”

Observação 4.14. O programa é apresentado integralmente no Apêndice A.6, para utilizá-lo, basta copiar os comandos apresentados, neste apêndice, e digitá-los em cada linha do editor do FreeMat, salvando-o no diretório corrente do computador.

Atividade 36. Utilize o programa ‘GAUSSJORDAN’, desenvolvido na Atividade 35, para determinar a solução do sistema:

$$\begin{cases} 1x + 2y + 3z = 15 \\ 3x + 3y = 6 \\ 6x + 6y = 12. \end{cases}$$

Passo 1: Abrir o programa ‘GAUSSJORDAN’.

Para abrir o programa, na janela de comandos digita-se:

GAUSSJORDAN

Após, pressione **Enter** e o programa será compilado (ver Figura 4.65).

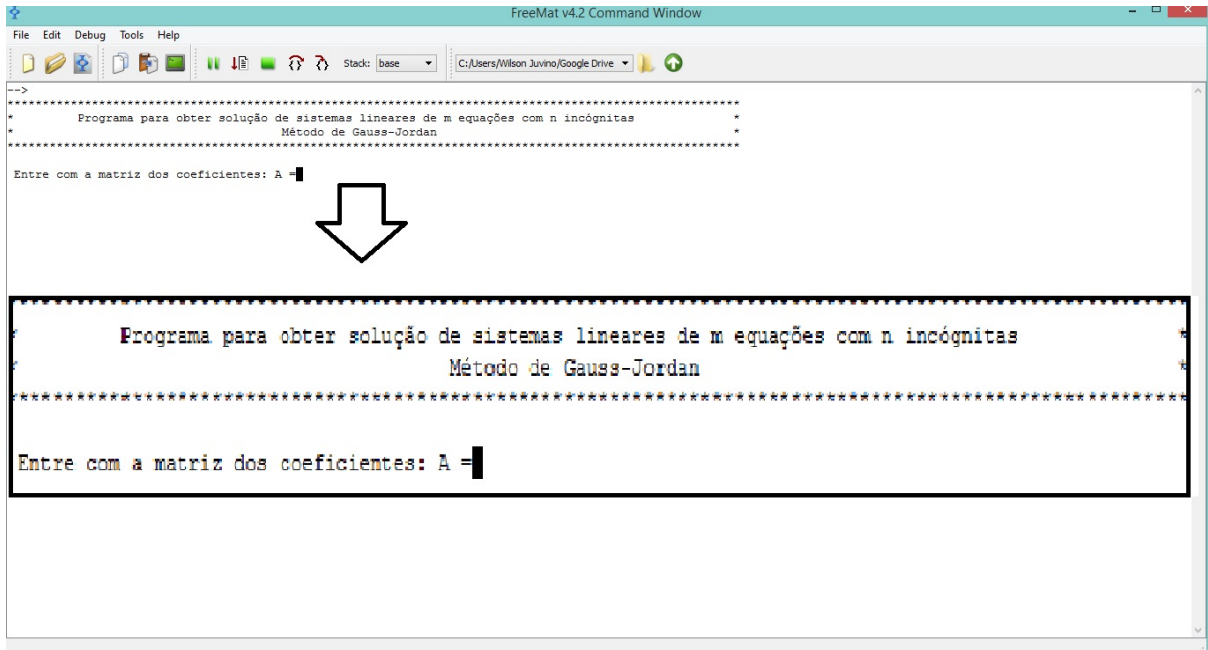


Figura 4.65: Desenvolvimento da Atividade 36, primeira etapa do programa Gauss Jordan

Passo 2: Inserir a matriz dos coeficientes $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 3 & 0 \\ 6 & 6 & 0 \end{bmatrix}$.

Para inserir a matriz dos coeficientes, digita-se:

[1 2 3; 3 3 0; 6 6 0]

e, ao pressionar **Enter**, o programa reportará o número de incógnitas e equações do sistema, conforme mostra a Figura 4.66.

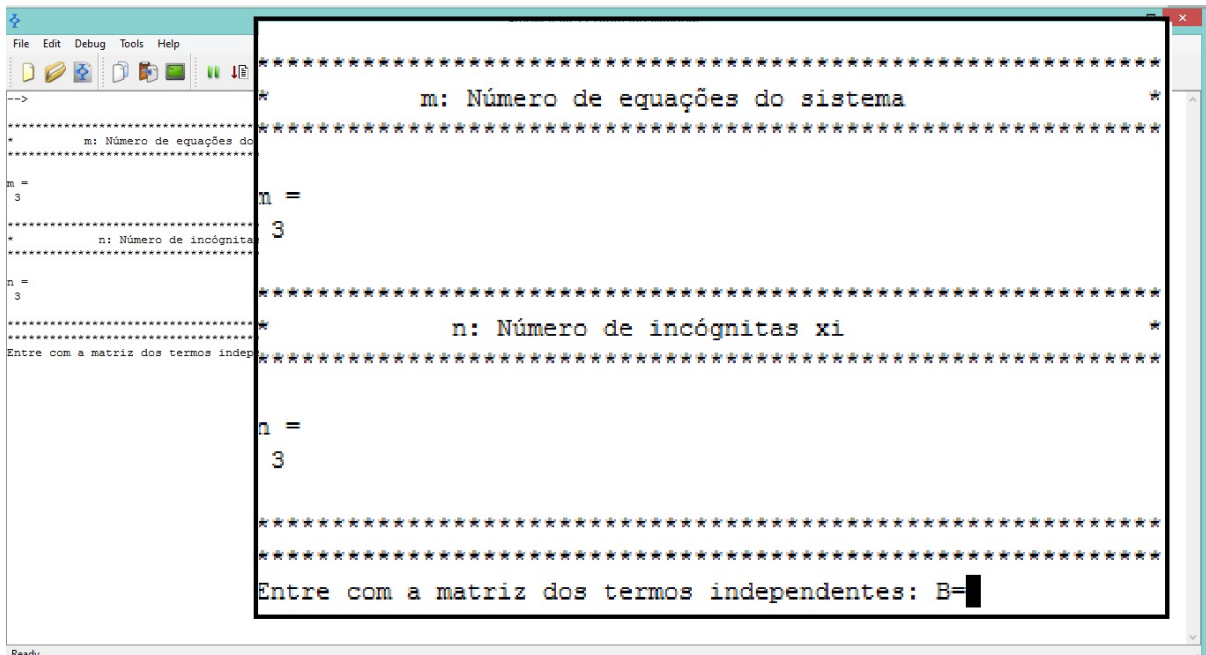


Figura 4.66: Desenvolvimento da Atividade 36, segunda etapa do programa Gauss Jordan

Passo 3: Inserir a matriz dos termos independentes.

Para inserir a matriz dos termos independentes, digita-se:

[15; 6; 12]

e, ao pressionar **Enter**, o programa reportará algumas informações sobre o sistema e sobre a matriz reduzida por linha à forma escada da matriz $C = [A|B]$, conforme mostra as figura 4.67 e 4.68.

```

*****
*                               Matriz ampliada C=(A|B)                               *
*****
C =
1 2 3 15
3 3 0 6
6 6 0 12
*****
*                               Posto da matriz A:                               *
*****
postoA =
2
*****
*                               Posto da matriz C:                               *
*****
postoC =
2
*****
*                               Como posto(A)=posto(C) e posto (A)< n o sistema é compatível e indeterminado                               *
*****

```

Figura 4.67: Desenvolvimento da Atividade 36, terceira etapa do programa Gauss Jordan

```

*****
*                               Número de parâmetros livres do sistema:                               *
*****
ans =
1
*****
*                               Como posto(A)=posto(C) e posto (A)< n o sistema é compatível e indeterminado                               *
*****
*                               Matriz C na forma linha escada reduzida                               *
*****
Escada =
1 0 -3 -11
0 1 3 13
0 0 0 0
*****
*                               Atribua quaisquer valores reais para cada parametro livre e substitua no sistema *
*                               escalonado, após, faça substituições regressivas obtendo um conjunto solução. *
*****
Deseja encerrar o programa? sim / nao

```

Figura 4.68: Desenvolvimento da Atividade 36, quarta etapa do programa Gauss Jordan

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.65, caso contrário, se inserir , o programa será encerrado, sendo apresentado o display da Figura 4.69.

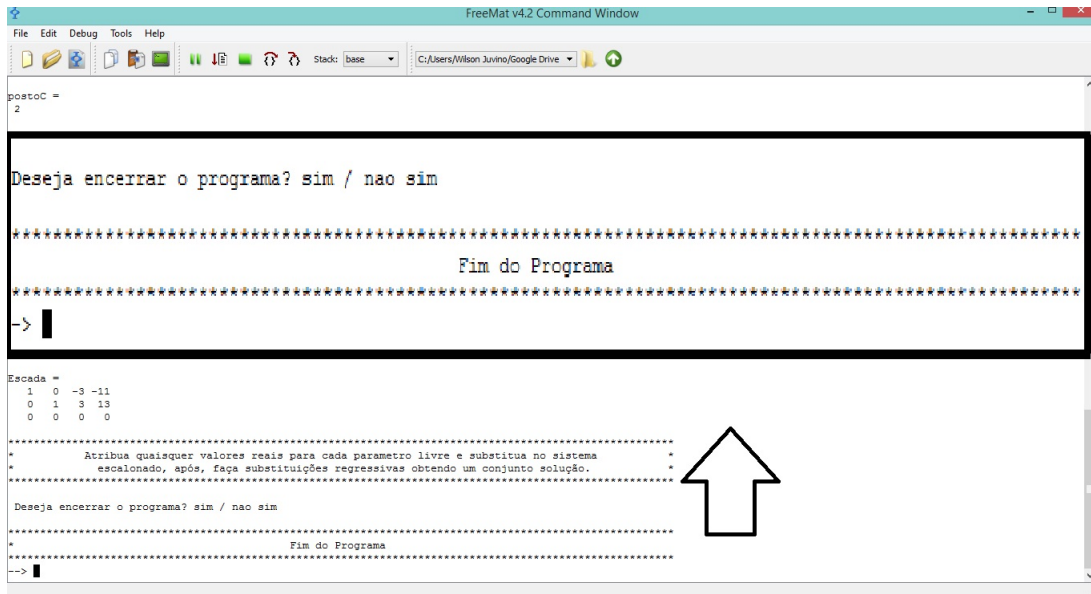


Figura 4.69: Desenvolvimento da Atividade 36, quinta etapa do programa Gauss Jordan

Passo 5: Explicitar a solução do sistema.

Pelo Passo 3, o sistema apresentado, no enunciado da Atividade 36, é equivalente ao sistema:

$$\begin{cases} x - 3z = -11 \\ y + 3z = 13 \end{cases}$$

Como existe um parâmetro livre, escolhemos $z = t$, com $t \in \mathbb{R}$. Substituindo neste sistema, obtemos $y = 13 - 3t$ e $x = -11 + 3t$.

Portanto, o conjunto solução do sistema linear dado é $\{(-11 + 3t, 13 - 3t, t); t \in \mathbb{R}\}$.

Atividade 37. Utilize o programa ‘GAUSSJORDAN’, desenvolvido na Atividade 35, para determinar a solução do sistema:

$$\begin{cases} 1x + 2y + 3z = 15 \\ 3x + 3y = 6 \\ 6x + 6y = 13. \end{cases}$$

Passo 1 Abrir o programa ‘GAUSSJORDAN’.

Para abrir o programa, na janela de comandos, digita-se:

GAUSSJORDAN

Após, pressione **Enter** e o programa será compilado (ver Figura 4.70).

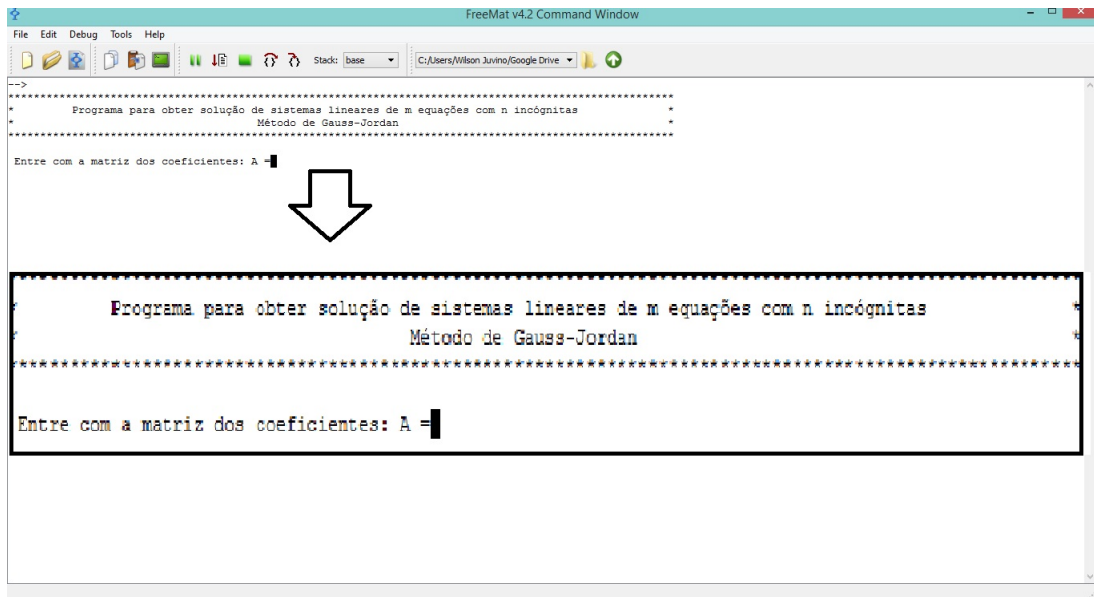


Figura 4.70: Desenvolvimento da Atividade 37, primeira etapa do programa Gauss Jordan

Passo 2: Inserir a matriz dos coeficientes $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 3 & 0 \\ 6 & 6 & 0 \end{bmatrix}$.

Para inserir a matriz dos coeficientes, digita-se:

[1 2 3; 3 3 0; 6 6 0]

e, ao pressionar **Enter**, o programa reportará o número de incógnitas e equações do sistema, conforme mostra a Figura 4.71.

Passo 3: Inserir a matriz dos termos independentes.

Para inserir a matriz dos termos independentes, digita-se:

[15; 6; 13]

e, ao pressionar **Enter**, o programa reportará algumas informações sobre o sistema linear e sobre a matriz reduzida por linha à forma escada da matriz $C = [A|B]$, conforme mostra a Figura 4.72.

Como $\text{posto}(A) < \text{posto}(C)$, o sistema não possui solução.

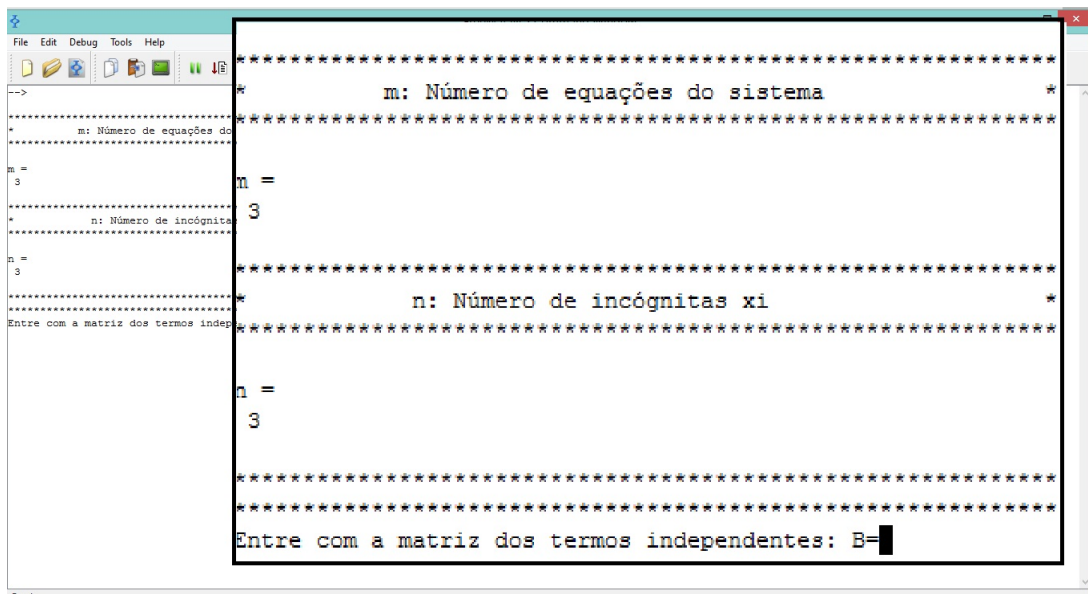


Figura 4.71: Desenvolvimento da Atividade 37, segunda etapa do programa Gauss Jordan

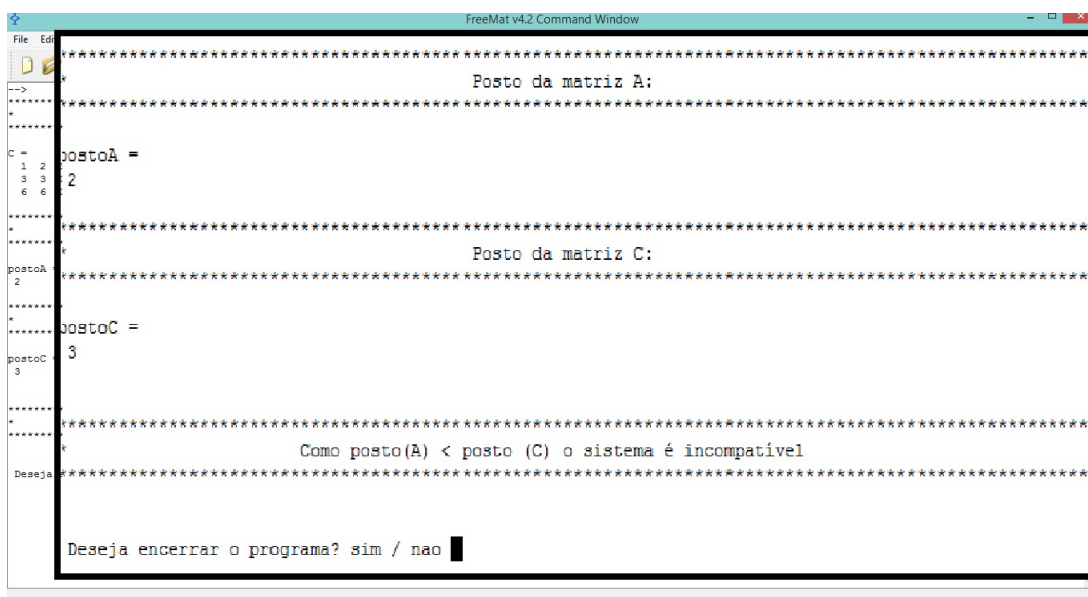


Figura 4.72: Desenvolvimento da Atividade 37, terceira etapa do programa Gauss Jordan

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.69, caso contrário, se inserir , o programa será encerrado, como mostra a Figura 4.73.

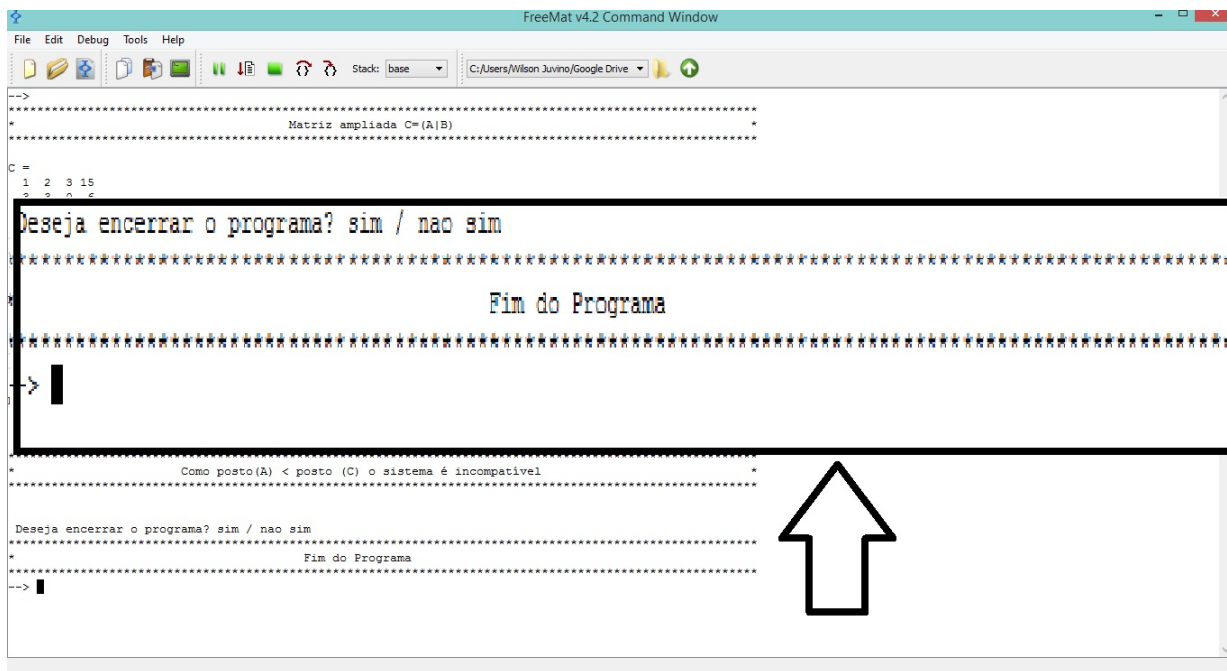


Figura 4.73: Desenvolvimento da Atividade 37, quarta etapa do programa Gauss Jordan

Atividade 38. Utilize o programa ‘GAUSSJORDAN’, desenvolvido na Atividade 35, para determinar a solução do sistema:

$$\begin{cases} 1x + 2y + 4z = 15 \\ 3x + 3y = 6 \\ 6x + 6y + 2z = 18. \end{cases}$$

Passo 1 Abrir o programa ‘GAUSSJORDAN’.

Para abrir o programa, na janela de comandos digite:

GAUSSJORDAN

Após, pressione **Enter** e o programa será compilado (ver Figura 4.74).

Passo 2: Inserir a matriz dos coeficientes $A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 0 \\ 6 & 6 & 2 \end{bmatrix}$.

Para inserir a matriz dos coeficientes, digita-se:

[1 2 4; 3 3 0; 6 6 2]

e, ao pressionar **Enter**, o programa reportará o número de incógnitas e equações do sistema, como mostra a Figura 4.75.

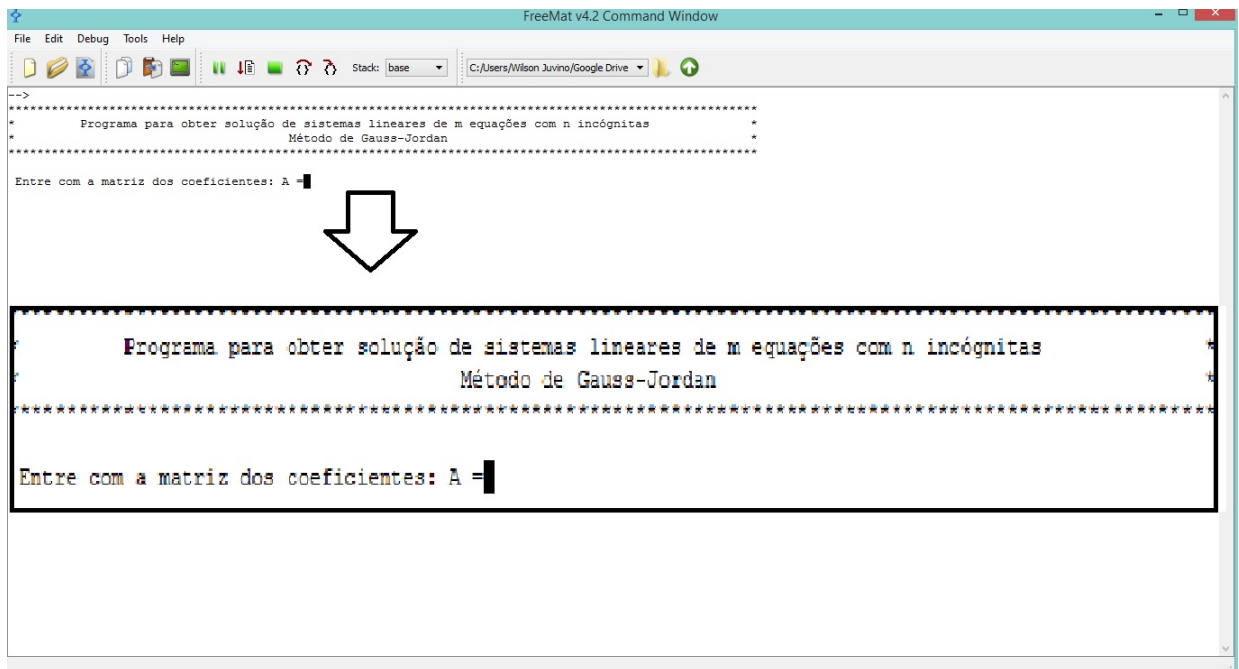


Figura 4.74: Desenvolvimento da Atividade 38, primeira etapa do programa Gauss Jordan

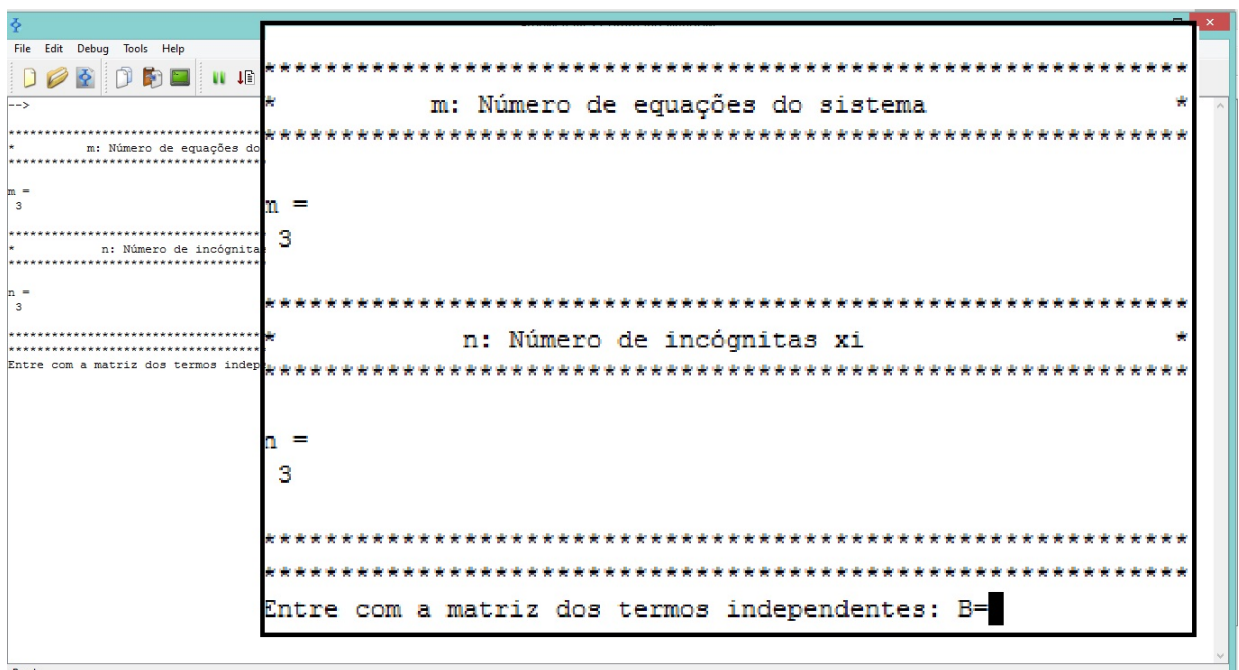


Figura 4.75: Desenvolvimento da Atividade 38, segunda etapa do programa Gauss Jordan

Passo 3: Inserir a matriz dos termos independentes B .

Para inserir a matriz dos termos independentes, digita-se:

[15; 6; 18]

e, ao pressionar **Enter**, o programa reportará algumas informações sobre o sistema linear e sobre a matriz reduzida por linha à forma escada da matriz $C = [A|B]$, como mostra as figuras 4.76 e 4.77.

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
-----
*
* Matriz ampliada C=[A|B]
*
-----
C =
1 2 4 15
3 3 0 6
6 6 2 18
-----
postoA =
3
-----
postoC =
3
-----
*
* Posto da matriz A:
*
-----
postoA =
3
-----
*
* Posto da matriz C:
*
-----
postoC =
3
-----
Escada =
1 0 0 1
0 1 0 1
0 0 1 3

```

Figura 4.76: Desenvolvimento da Atividade 38, terceira etapa do programa Gauss Jordan

```

FreeMat v4.2 Command Window
File Edit Debug Tools Help
-----
*
* Como posto(A)=posto(C) e posto (A)= n o sistema é compatível e determinado
*
-----
*
* Matriz C na forma linha escada reduzida
*
-----
Escada =
1 0 0 1
0 1 0 1
0 0 1 3
-----
*
* Como posto
*
-----
Deseja revelar cada incógnita e o seu respectivo valor? sim/ nao
-----
Escada =
1 0 0 1
0 1 0 1
0 0 1 3
-----
Deseja revelar cada incógnita e o seu respectivo valor? sim/ nao

```

Figura 4.77: Desenvolvimento da Atividade 38, terceira etapa do programa Gauss Jordan

Passo 4: Reinicializar ou encerrar o programa.

Ao inserir e teclar **Enter**, o programa é reinicializado, sendo apresentado o display da Figura 4.74, caso contrário, se inserir , o programa reportará o valor de cada incógnita, como mostra a Figura 4.78.

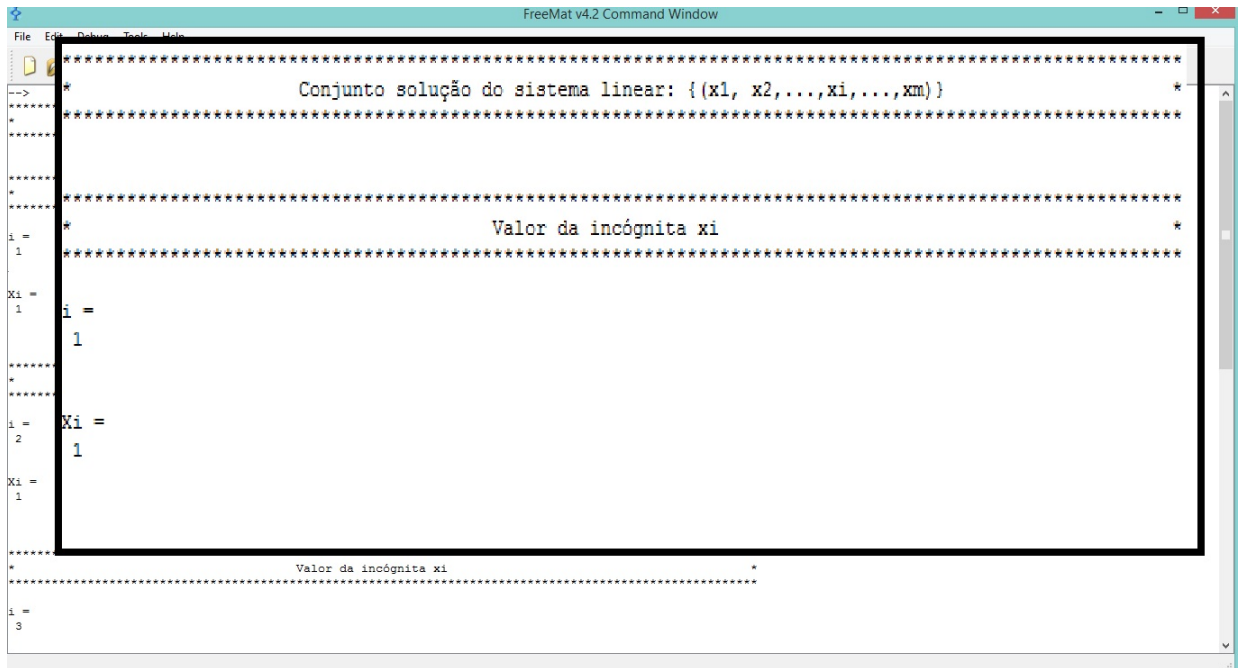


Figura 4.78: Desenvolvimento da Atividade 38, quarta etapa do programa Gauss Jordan

Passo 5: Explicitar a solução do sistema.

Pelo Passo 3, a solução do sistema é dada pelos elementos da última coluna da matriz reduzida por linha à forma escada, sendo estes:

$$x = 1, \quad y = 1, \quad z = 3.$$

Portanto, o conjunto solução do sistema é dado por $\{(1, 1, 3)\}$.

Considerações finais

As funções são apresentadas inicialmente no ensino fundamental e, a partir daí, nos outros níveis de ensino são aprofundados seus estudos. As aplicações deste conteúdo são amplas e abordadas nos livros pedagógicos, bem como, vários trabalhos acadêmicos discutem e apresentam estas aplicações. No âmbito do estudo destas funções, exemplos e exercícios tratados em salas de aulas do nível médio geralmente não apresentam um tratamento adequado, pois, são abordados como uma equação polinomial e isto, também é, observado em algumas literaturas pedagógicas. Para o estudo do esboço gráfico das funções, existem diversos softwares computacionais que podem auxiliar os docentes, porém, percebe-se esta prática, ainda incomum, no Ensino Médio.

Desta forma, com as propostas apresentadas, busca-se inserir a utilização do software em sala de aula e auxiliar os docentes com atividades e uma sequência didática que possibilite isto.

Durante a aplicação da Proposta I, observou-se que, além do entusiasmo dos discentes ao utilizar o software, existiram poucas dificuldades com a inserção dos comandos. Também, ressalta-se que através do desenvolvimento das atividades da proposta, os discentes interpretaram facilmente um gráfico, ressaltando que, quando comparado aos gráficos construídos sem uso de softwares, a facilidade no esboço trouxe oportunidade de estudar gráficos mais complexos para este nível de ensino.

Dos obstáculos para desenvolvimento das atividades da Proposta I, destacou-se, as dificuldades de interpretação de problemas matemáticos pelos discentes e para definir os comandos de condição e repetição. Diante disto, sugere-se que, para a aplicação da Proposta I, seria interessante fazer um algoritmo em português no quadro, usando-se os condicionais: *se*, *enquanto*, *senão*, *então*, *etc.*, e depois passar para a linguagem de programação do FreeMat, semelhante ao que foi apresentado na Seção 2.3. Todavia, a inserção

dos comandos e atividades voltadas para a programação, trouxe novas interpretações e leituras da resolução de um problema pelos discentes.

Portanto, observou-se que a proposta traz um ensino diferenciado, mais dinâmico e, conjuntamente, traz um aprofundamento nos conteúdos trabalhados em sala, oportunizando algumas experiências na área de programação. Por fim, destaca-se a vontade apresentada pelos discentes em continuar estudando dentro desta metodologia, a facilidade em trabalhar com os comandos do programa e a fixação do conteúdo pelos mesmos.

Quanto ao estudo de matrizes e sistemas lineares, sendo estes, conteúdos que na disciplina de Álgebra Linear do Ensino Superior, tem seu estudo aprofundado, e de grande relevância, devido apresentar várias aplicações e um tratamento algébrico formal, exige-se do discente um bom conhecimento de suas propriedades algébricas. No Ensino Médio, o estudo de matrizes e suas aplicações na resolução de sistemas lineares, geralmente não é aprofundado, também, percebe-se grandes dificuldades em apresentar aplicações deste conteúdo para o aluno.

Neste sentido, as propostas II e III, foram elaboradas buscando inserir o estudo das matrizes e de suas aplicações. A sequência das atividades apresentadas, constituem um caminho e orientação aos docentes para inserção do software FreeMat no ensino de matrizes e sistemas lineares na educação básica.

Anseia-se que, na aplicação destas propostas nas escolas de nível médio, os docentes consigam aprofundar os estudos do software junto aos alunos, construindo outros programas, de modo a utilizar cada vez menos as linguagens pré-definidas deste recurso computacional. Ressalta-se que, após o desenvolvimento das propostas, é sempre necessário uma abordagem algébrica usual no quadro.

Referências Bibliográficas

- BATISTA, S. C. F.(2004). *Sofmat: Um repositório de softwares para matemática do ensino médio- um instrumento em prol de posturas mais conscientes na seleção de softwares educacionais*. Campos Goytacazes, 2004. 202 f. Dissertação (Mestrado em Engenharia de Produção) - Centro de Ciência e Tecnologia, Universidade Estadual do Norte Fluminense.
- BELINE, W. COSTA, N. M. L. *Educação matemática, tecnologia e formação de professores: Algumas reflexões*. Fecilcan, Paraná, 2010, p.94–220–218.
- BOLDRINI, J. L.; et al. *Álgebra Linear*. Editora Harbra, 3 ed. São Paulo, 1980.
- CALLIOLI, C. A.; HYGINO, H. D.; COSTA, R. *Álgebra Linear e Aplicações*. ATUAL EDITORA, 6 ed. São Paulo, 2000.
- BRASIL (2000). *Parametros Curriculares Nacionais do Ensino Médio: ciencias da natureza, matemática e suas tecnologias*. Ministério da Educação : Secretaria de Educação Básica, Brasília, pg.41.
- BRASIL (2014). *Guia de livros didáticos: PNLD 2015. Matemática: ensino médio*. Ministério da Educação : Secretaria de Educação Básica, Brasília, p. 108
- CURY, H. N. (2004). *Disciplinas Matemáticas em Cursos Superiores*. EDIPICRS, Porto Alegre, p.17.
- FIPIAK, E. FIGUEIREDO, F. R. SOMAVILLA, F. *Introdução ao Matlab*. Santa Maria/SC, 2012. Disponível em: <http://w3.ufsm.br/petmatematica/arquivos/matlab.2012.pdf>. Acesso em: 28 jul. 2014.

- FUGIMOTO, S. M. A. *Informática na educação: A questão da utilização do computador na escola em uma perspectiva construtivista*. XVII Congresso de Leitura do Brasil, UNICAMP, Campinas/SP, 2009, p.9. Disponível em: <http://alb.com.br/arquivo-morto/edicoesanteriores/anais17/txtcompletos/sem16/COLE1066.pdf>. Acesso em: 14 out. 2014.
- FREEMAT, Website. Disponível em: <http://freemat.sourceforge.net/>. Acesso em: 29 Agos. 2014.
- HEFEZ, A. FERNANDEZ, C. d. S. *Introdução à Álgebra Linear*. Coleção Profmat, SBM, 1 ed. Rio de Janeiro, 2012.
- HOWARD, A. RORRES, C. *Álgebra Linear com Aplicações*. Editora Boockman, 8 ed. Porto Alegre, 2001.
- HYGINO, H. D. IEZZI, G. *Álgebra Moderna*. Editora Atual, 4 ed. São Paulo, 2003.
- IEZZI, G. MURAKAMI, C. *Fundamentos de Matemática Elementar*. ATUAL EDITORA, vol 1, 7 ed. São Paulo, 2002.
- LIMA, E. L. *Números e Funções Reais*. Coleção Profmat, SBM, 1 ed. Rio de Janeiro, 2013.
- LLANO, J. G. (2006). *A informática educativa na escola*. Edições Loyola, São Paulo, p. 27–31.
- MARIANI, V. C. SOUZA, V. D. M. *Um Breve Relato do Desenvolvimento do Conceito de Função*. Educare PUCPR, 2005, PARANÁ, Disponível em: <http://www.pucpr.br/eventos/educere/educere2005/anaisEvento/documentos/com/TCCI021.pdf>; Acesso em: 15 Maio. de 2015.
- NOBRE, M. *O uso do software Matlab para o estudo de alguns tópicos de Álgebra Linear*. pg.1–2. Disponível em: <http://www.ucb.br/sites/100/103/TCC/22005/MarcelloNobreCardoso.pdf>, Acesso em: 21 Agos. de 2014.
- POLCINO, f. C. M. *Anéis e Módulos*. 1. ed. São Paulo: IME-USP, 1972. 202 p.

- SANTOS, R. LORETO, A. B. GONÇALVES, J. L. *Avaliação de softwares matemáticos quanto a sua funcionalidade e tipo de licença para uso em sala de aula*. REnCiMa, vol.1, n.1, p. 47-65, 2010.
- SILVA, A. A. *Introdução à Álgebra Linear*. Ed. Universitária/UFPB, 1 ed. João Pessoa, 2007.
- SHARMA, N. GOBBERT, M. K. *A comparative evaluation comparative of Matlab, Octave, FreeMat, and Scilab for research and teaching*. Department of Mathematics and Statistics, University of Maryland, Baltimore County, Disponível em: <http://userpages.umbc.edu/~gobbert/papers/SharmaGobbertTR2010.pdf>. Acesso em: 22 agos. 2014.
- SODRE, U. OLIVEIRA, A. S. CORRÊA, T. P. *FreeMat: Uma introdução*. Londrina-PR, 2009, pg.4—6. Disponível em: <http://www.uel.br/projetos/matessencial/superior/pdfs/freemat.pdf>. Acesso em: 20 agos. 2014.

Apêndice: Programas escritos em FreeMat

A.1 Programa para determinar o valor a pagar para um certo consumo de água das residências em Cuiabá/MT

```
function VALORES
sim=1; nao=0; continuar=1; while continuar==1
clc
disp(' ')
disp('*****')
disp('*Programa para determinar o valor decorrente do consumo de água*')
disp('*****')
disp(' ')
x=input ('Insira o consumo em m3==>')
clc
if x>=0 & x<=10
disp (' ')
disp('*****')
disp('*          Consumo em m3:          *')
disp('*****')
```

```

consumo=x
y= 4.313*x;
disp('*****')
disp('*                Valor a pagar em reais :                *')
disp('*****')
valor=y
elseif x>10 & x<=20
disp(' ')
disp('*****')
disp('*                Consumo em m3:                *')
disp('*****')
consumo=x
y= 5.282*x-9.69;
disp('*****')
disp('*                Valor a pagar em reais :                *')
disp('*****')
valor=y
elseif x>20 & x<=30
disp(' ')
disp('*****')
disp('*                Consumo em m3:                *')
disp('*****')
consumo=x
y= 8.816*x-80.37;
disp('*****')
disp('*                Valor a pagar em reais :                *')
disp('*****')
valor=y
elseif x>30 & x<=50
disp(' ')
disp('*****')
disp('*                Consumo em m3:                *')

```



```

disp('*****')
consumo=x
y= 10.792*x-139.65;
disp('*****')
disp('*                Valor a pagar em reais :                *')
disp('*****')
valor=y
elseif x>50
disp(' ')
disp('*****')
disp('*                Consumo em m3:                *')
disp('*****')
y=14.269*x-313.5;
disp('*****')
disp('*                Valor a pagar em reais :                *')
disp('*****')
valor=y
end
continuar=input('Deseja obter o valor para outro consumo ?  sim /nao');
disp(' ')
disp('*****')
disp('*                Fim do programa                *')
disp('*****')
end

```

A.2 Programa para criptografar mensagens pelo sistema poligráfico, baseado na multiplicação de matrizes com entradas reais

```

function CRIPTOGRAFAR
sim=1; nao=0; reiniciar=0; while reiniciar==0
clc
a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;
n=14;o=15;p=16;q=17;r=18;s=19;t=20; u=21; v=22; w=23; x=24; y=25; z=26;
disp ( ' ')
disp ( '*****')
disp ( '* Programa para criptografar mensagens pelo sistema poligráfico,*')
disp ( '* baseado na multiplicação de matrizes com entradas reais *')
disp ( '*****')
disp ( ' ')
disp ( ' ')
disp ( '*****')
disp ( '* Tabela de correspondência numérica *')
disp ( '*****')
disp ( '* *')
disp ( '* *')
disp ( '*****')
disp ( '* *')
disp ( '* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14' *)
disp ( '* *')
disp ( '* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp ( '* *')
disp ( '*****')
disp ( ' 0 caractere 0 corresponde a um espaço vazio ')
disp ( '*****')
disp ( ' ')

```

```

disp(' ')
A=input(' Entre com a matriz chave de ordem n: A=')
disp(' ')
disp(' ')
[I, J]=size(A)
clc
disp(' ')
disp(' ')
if det(A)==0
disp ('*****')
disp ('* A matriz chave não é válida, pois a matriz não é inversível. *')
disp ('*****')
reiniciar=input('Deseja encerrar o programa? sim / nao');
disp ('*****')
disp ('*                               Fim do Programa                               *')
disp ('*****')
else
disp ('*****')
disp ('*           A matriz M deve possuir J elementos em cada linha           *')
disp ('*****')
J=J
disp ('*****')
M=input('Entre com a matriz cujas entradas são caracteres da mensagem:M= ')
clc
disp(' ')
disp ('*****')
disp ('*           Cada linha da matriz MA é um bloco do criptograma           *')
disp ('*****')
disp(' ')
disp(' ')
MA=M*A
reiniciar=input('Deseja encerrar o programa? sim /nao');

```

```

disp ( ' ')
disp ( ' ')
disp ( '*****')
disp ( '*                Fim do Programa                *')
disp ( '*****')
end
end

```

A.3 Programa para decifrar mensagens pelo sistema poligráfico, baseado na multiplicação de matrizes com entradas reais

```

function DECIFRAR
sim=1; nao=0; reiniciar=0;
while reiniciar==0
clc
disp ( ' ')
disp ( '*****')
disp ( '* Programa para decifrar mensagens pelo sistema poligráfico, *')
disp ( '* baseado na multiplicação de matrizes com entradas reais *')
disp ( '*****')
disp ( ' ')
disp ( ' ')
disp ( '*****')
disp ( '*                Tabela de correspondência numérica                *')
disp ( '*****')
disp ( '* *')
disp ( '* *')
disp ( '*****')

```

```

disp ('*
                                     *')
disp ('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14' *)
disp ('*
                                     *')
disp ('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp ('*
                                     *')
disp ('*****')
disp ('          0 caractere 0 corresponde a um espaço vazio          ')
disp ('*****')
disp(' ')
disp(' ')
A=input(' Entre com a matriz chave de ordem n:  A=')
[I, J]=size(A);
clc
disp(' ')
disp(' ')
if det(A)==0
disp ('*****')
disp ('* A matriz chave não é válida, pois a matriz não é inversível. *')
disp ('*****')
reiniciar=input('Deseja encerrar o programa?  sim / nao');
disp ('*****')
disp ('*
                                     Fim do Programa                                     *')
disp ('*****')
else
disp ('*****')
disp ('*          A matriz C deve possuir J elementos em cada linha          *')
disp ('*****')
J=J
disp ('*****')
C=input('Entre com a matriz cujos elementos caracteres do criptograma:C= ')
clc
disp(' ')

```

```

disp ('*****')
disp ('*          Cada linha da matriz M é um bloco do texto comum          *')
disp ('*****')
disp(' ')
disp(' ')
M=C*inv(A)
disp ('* *')
disp ('* *')
disp ('*****')
disp ('          Utilize a tabela abaixo para escrever o texto comum          ')
disp ('*****')
disp ('* *')
disp ('* *')
disp ('*****')
disp ('*                                     *')
disp ('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14' *)
disp ('*                                     *')
disp ('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp ('*                                     *')
disp ('*****')
disp ('          0 caractere 0 corresponde a um espaço vaziao          ')
disp ('*****')
disp(' ')
disp(' ')
reiniciar=input('Deseja encerrar o programa?  sim /nao');
disp (' ')
disp (' ')
disp ('*****')
disp ('*                               Fim do Programa                               *')
disp ('*****')
end
end

```

A.4 Programa para criptografar mensagens pelo sistema poligráfico: 2-Cifras de Hill

```
function CRIPTOGRAFARHILL2
sim=1; nao=0; reiniciar=0;
while reiniciar==0
    clc
    disp(' ')
    disp('*****')
    disp('*Programa para criptografar mensagens pelo sistema poligráfico, *')
    disp('*                2-Cifras de Hill                *')
    disp('*****')
    disp(' ')
    disp(' ')
    disp('*****')
    disp('*                Tabela de correspondência numérica                *')
    disp('*****')
    disp('* *')
    disp('* *')
    disp('*****')
    disp('* *')
    disp('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14 *')
    disp('* *')
    disp('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
    disp('* *')
    disp('*****')
    disp('                0 caractere 0 corresponde a um espaço vaziao                ')
    disp('*****')
    disp(' ')
endwhile
```

```

disp(' ')
A=input(' Entre com a matriz chave de ordem 2: A=')
clc
disp(' ')
disp(' ')
if size(A)~= [2,2] || mod(A(1)*A(4)-A(2)*A(3),3)==0
disp ('*****')
disp ('* A matriz chave não é válida, pois a matriz não é inversível. *')
disp ('*****')
reiniciar=input('Deseja encerrar o programa? sim / nao');
disp ('*****')
disp ('*                               Fim do Programa                               *')
disp ('*****')
else
disp ('*****')
disp ('*          A matriz M deve possuir 2 elementos em cada linha          *')
disp ('*****')
disp(' ')
disp(' ')
disp ('*****')
M=input('Entre com a matriz cujos elementos caracteres do texto comum:M= ')
clc
disp(' ')
disp ('*****')
disp ('*          Cada linha da matriz MA é um bloco do criptograma          *')
disp ('*****')
disp(' ')
disp(' ')
MA=mod(M*A,27)
disp ('* *')
disp ('* *')
disp ('*****')

```



```

disp ('          Utilize a tabela abaixo para escrever o criptograma          ')
disp ('*****')
disp ('* *')
disp ('* *')
disp ('*****')
disp ('*                                     *')
disp ('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14' *)
disp ('*                                     *')
disp ('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp ('*                                     *')
disp ('*****')
disp ('          0 caractere 0 corresponde a um espaço vazio          ')
disp ('*****')
disp(' ')
disp(' ')
reiniciar=input('Deseja encerrar o programa?  sim /nao');
disp (' ')
disp (' ')
disp ('*****')
disp ('*                                     Fim do Programa                                     *')
disp ('*****')
end
end

```

A.5 Programa para decifrar mensagens pelo sistema poligráfico: 2-Cifras de Hill

```

function DECIFRARHILL2
sim=1; nao=0; reiniciar=0;
while reiniciar==0

```

```

clc
disp(' ')
disp('*****')
disp('* Programa para decifrar mensagens pelo sistema poligráfico, *')
disp('*                2-Cifras de Hill                *')
disp('*****')
disp(' ')
disp(' ')
disp('*****')
disp('*                Tabela de correspondência numérica                *')
disp('*****')
disp('* *')
disp('* *')
disp('*****')
disp('* *')
disp('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14 *')
disp('* *')
disp('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp('* *')
disp('*****')
disp('                0 caractere 0 corresponde a um espaço vazio                ')
disp('*****')
disp(' ')
disp(' ')
A=input(' Entre com a matriz chave de ordem 2: A=')
D=[A(4) -A(3); -A(2) A(1)];
clc
disp(' ')
disp(' ')
if size(A)~= [2,2] || mod(A(1)*A(4)-A(2)*A(3),3)==0
disp('*****')
disp('* A matriz chave não é válida, pois a matriz não é inversível. *')

```

```

disp ('*****')
reiniciar=input('Deseja encerrar o programa? sim / nao');
disp ('*****')
disp ('*                Fim do Programa                *')
disp ('*****')
else
elseif mod(A(1)*A(4)-A(2)*A(3),27)==1
inv=1*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==2
inv=14*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==4
inv=7*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==5
inv=11*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==7
inv=4*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==8
inv=17*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==10
inv=19*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==11
inv=5*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==13
inv=25*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==14
inv=2*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==16
inv=22*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==17
inv=8*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==19
inv=10*D;

```

```

elseif mod(A(1)*A(4)-A(2)*A(3),27)==20
inv=23*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==22
inv=16*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==23
inv=20*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==25
inv=13*D;
elseif mod(A(1)*A(4)-A(2)*A(3),27)==26
inv=26*D;
end
disp('*****')
disp('*          A matriz C deve possuir 2 elementos em cada linha          *')
disp('*****')
disp(' ')
disp(' ')
C=input('Entre com a matriz cujos elementos caracteres do criptograma:C= ')
clc
disp(' ')
disp(' ')
disp('*****')
disp('*          Cada linha da matriz M é um bloco do texto comum          *')
disp('*****')
disp(' ')
disp(' ')
M=mod(C*inv,27)
disp(' * *')
disp(' * *')
disp('*****')
disp('          Utilize a tabela abaixo para escrever o texto comum          ')
disp('*****')
disp(' * *')

```

```

disp ('* *')
disp ('*****')
disp ('*                                     *')
disp ('* a=1;b=2;c=3;d=4;e=5;f=6;g=7;h=8;i=9;j=10;k=11;l=12;m=13;n=14' *)
disp ('*                                     *')
disp ('* o=15;p=16;q=17;r=18;s=19;t=20;u=21;v=22;w=23;x=24;y=25;z=26 * ')
disp ('*                                     *')
disp ('*****')
disp ('          0 caractere 0 corresponde a um espaço vazio          ')
disp ('*****')
disp(' ')
disp(' ')
reiniciar=input('Deseja encerrar o programa?  sim /nao');
disp (' ')
disp (' ')
disp ('*****')
disp ('*                               Fim do Programa                               *')
disp ('*****')
end
end

```

A.6 Programa para resolução de sistemas pela Regra de Cramer

```

function CRAMER
sim=1; nao=0; reiniciar=0;
while reiniciar==0
clc
disp (' ')
disp ('*****')

```

```

disp ('*          Programa para obter solução de sistemas lineares de:          *')
disp ('*                  m equações com m incógnitas                          *')
disp ('*                  Regra de Cramer                                       *')
disp ('*****')
A=input('Entre com a matriz dos coeficientes: A =')
clc
disp (' ')
disp ('*****')
disp ('*                  Matriz A                                               *')
disp ('*****')
disp (' ')
A=A
disp (' ')
disp ('*****')
disp ('*                  Determinante da matriz A                               *')
disp ('*****')
disp (' ')
detA=det(A)
if det(A)  sim=0
disp ('*****')
disp ('* Como det(A) é diferente de zero, o sistema possui solução única *')
disp ('*****')
disp (' ')
B=input('Para obter a solução, insira a matriz dos termos independentes: B =')
clc
disp (' ')
disp ('*****')
disp ('*                  solução do sistema linear:                             *')
disp ('*****')
X=inv(A)*B
reiniciar=input('Deseja revelar cada incógnita e o respectivo valor?  sim/não')
clc

```

```

disp ( ' ')
disp ( '*****')
disp ( '*      Conjunto solução do sistema linear: {(x1, x2, ..., xi, ..., xm)}:      *')
disp ( '*****')
disp ( ' ')
disp ( ' ')
[m n]=size(X);
for i=1:m
disp ( '*****')
disp ( '*                          valor da incógnita xi                          *')
disp ( '*****')
i=i
xi=[X(i)]
end
disp ( ' ')
disp ( '*****')
disp ( '*                          Fim do programa                          *')
disp ( '*****')
elseif det(A)==0
disp ( '*****')
disp ( '*                          Como det(A)=0, informações sobre                          *')
disp ( '*                          as soluções do sistema linear são inconclusivas                          *')
disp ( '*****')
disp ( ' ')
disp ( '*****')
disp ( '*                          Fim do programa                          *')
disp ( '*****')
end
end

```

A.7 Programa para resolução de sistemas pelo método de Gauss-Jordan

```
function GAUSSJORDAN
sim=1; nao=0; reiniciar=0;
while reiniciar==0
clc
disp ( ' ')
disp ( '*****')
disp ( '*      Programa para obter solução de sistemas lineares de      *')
disp ( '*                               m equações com n incógnitas                               *')
disp ( '*                               Método de Gauss- Jordan                               *')
disp ( '*****')
disp ( ' ')
A=input(' Entre com a matriz dos coeficientes : A = ');
[m, n]=size(A);
clc
disp ( ' ')
disp ( ' ')
disp ( '*****')
disp ( '*                m: Número de equações do sistema                *')
disp ( '*****')
m=m
disp ( '*****')
disp ( '*                n: Número de incógnitas xi                *')
disp ( '*****')
n=n
disp ( '*****')
B=input(' Entre com a matriz dos termos independentes: B = ');
clc
disp ( '*****')
```



```

disp ('*                               Matriz ampliada C=(A|B):                               *')
disp ('*****')
C=[A B]
disp ('*****')
disp ('*                               Posto da matriz A:                               *')
disp ('*****')
postoA=rank(A)
disp ('*****')
disp ('*                               Posto da matriz C:                               *')
disp ('*****')
postoC=rank(C)
if rank(A)==rank(C) & rank(A)==n
disp (' ')
disp ('*****')
disp ('*          Como posto(A)=posto(C) e posto (A)= n          *')
disp ('*          o sistema é compatível e determinado          *')
disp ('*****')
disp (' ')
disp (' ')
disp ('*****')
disp ('*          Matriz C na forma linha escada reduzida          *')
disp ('*****')
Escada=rref (C)
disp (' ')
disp (' ')
reiniciar=input('Deseja revelar cada incógnita e seu valor?  sim /nao');
clc
disp (' ')
disp ('*****')
disp ('* Conjunto solução do sistema linear {(x1, x2,...,xi,...,xm)}:*')
disp ('*****')
disp (' ')

```

```

disp ( ' ')
[m n]=size(X);
for i=1:m
disp ( '*****')
disp ( '*                Valor da incógnita xi                *')
disp ( '*****')
i=i
Xi=Escada(m*(n-1)+1*i)
disp ( '*****')
disp ( ' ')
disp ( ' ')
end
disp ( '*****')
disp ( '*                Fim do programa                *')
disp ( '*****')
elseif rank(A)< rank(C)
disp ( ' ')
disp ( '*****')
disp ( '*      Como posto(A) < posto (C) o sistema é incompatível      *')
disp ( '*****')
disp ( ' ')
disp ( ' ')
reiniciar=input(' Deseja encerrar o programa? sim / nao ');
disp ( ' ')
disp ( '*****')
disp ( '*                Fim do programa                *')
disp ( '*****')
elseif rank(A)==rank(C) & rank(A)<n
disp ( ' ')
disp ( '*****')
disp ( '*      Como posto(A)=posto(C) e posto (A)< n                *')
disp ( '*      o sistema é compatível e indeterminado                *')

```

```

disp ('*****')
disp (' ')
disp (' ')
disp ('*****')
disp ('*          Número de parâmetros livres do sistema:          *')
disp ('*****')
n-rank(A)
disp (' ')
disp (' ')
disp ('*****')
disp ('*          Matriz C na forma linha escada reduzida          *')
disp ('*****')
Escada=rref (C)
disp ('*****')
disp ('* Atribua quaisquer valores reais para cada incógnita livre e *')
disp ('* substitua no sistema escalonado, após, faça substituições *')
disp ('*          regressivas obtendo um conjunto solução.          *')
disp ('*****')
reiniciar=input(' Deseja encerrar o programa? sim / nao ');
disp (' ')
disp ('*****')
disp ('*          Fim do programa          *')
disp ('*****')
end
end

```