

IMPLEMENTAÇÃO DA FATORAÇÃO DE CHOLESKY COM ÊNFASE EM SUA REPRESENTAÇÃO VISUAL

MICHELLE LAU DE ALMEIDA,^a

VANESSA FERNANDES DA SILVA,^b

ISABELLE BARBALHO OLIVEIRA DE SOUZA,^c

MARCELO SCHOTS,^d ZOCHIL GONZÁLEZ ARENAS^e

Resumo

A decomposição ou Fatoração de Cholesky é um método direto de resolução de sistemas lineares, *i.e.*, produz uma solução exata do sistema após um número finito de operações aritméticas. Por se tratar de passos não triviais, o ensino deste método de decomposição pode se tornar desafiador. Neste contexto, representações visuais podem ser uma ferramenta de apoio vantajosa. Objetivando tornar o processo de ensino mais didático, este trabalho apresenta a implementação das etapas da decomposição utilizando a linguagem TuPy. Essa linguagem oferece uma abordagem gráfica que facilita a compreensão do método, já que é realizado o passo a passo da execução. Dessa forma, além de mostrar como funciona o método de fatoração do ponto de vista algorítmico, torna-se possível obter um entendimento mais efetivo através da visualização de cada etapa da decomposição.

Palavras-chave: Sistemas Lineares, Fatoração de Cholesky, Representação visual, TuPy Online.

Abstract

^aPrograma de Pós-Graduação em Ciências Computacionais (PPG-CCOMP), Instituto de Matemática e Estatística (IME), Universidade do Estado do Rio de Janeiro (UERJ), RJ, Brasil; **E-mail:** miiichellelau@gmail.com

^bPPG-CCOMP, IME/UERJ, RJ, Brasil; **E-mail:** vanessa.fernandes.silva@hotmail.com

^cPPG-CCOMP, IME/UERJ, RJ, Brasil; **E-mail:** ibarbalho26@gmail.com

^dDepartamento de Informática e Ciências da Computação, IME/UERJ, RJ, Brasil; ORCID: 0000-0003-2495-4637 **E-mail:** schots@ime.uerj.br

^eDepartamento de Matemática Aplicada, IME/UERJ, RJ, Brasil; ORCID: 0000-0001-8377-2687 **E-mail:** zochil@ime.uerj.br

The Cholesky decomposition or factorization is a direct method for solving linear systems, *i.e.*, it produces an exact solution of the system after a finite number of arithmetic operations. Considering that it is not a trivial algorithm, teaching this factorization method might be challenging. In this context, visual representations could be a useful support tool. For a more didactic teaching experience, this work presents the implementation of the decomposition stages by means of the TuPy language. This language offers a graphical approach, helping in the understanding of the method, since the execution is carried out step by step. Thus, in addition to showing how the factorization method works from an algorithmic point of view, it is possible to gain a more effective understanding through the visualization of each stage of the decomposition.

Keywords: Linear Systems, Cholesky Factorization, Visual representation, TuPy Online.

MSC2010: 65F05, 97D40, 68V99

1 Introdução

Encontrar a solução de sistemas lineares é um dos problemas básicos de álgebra linear, com aplicações em múltiplas áreas de conhecimentos. Os métodos numéricos usados para esse fim se classificam em dois grupos [11]: métodos diretos e métodos iterativos. Os métodos diretos encontram a solução exata do sistema linear, a menos de erros de arredondamento e caso ela exista, após um número finito de passos. Entre os métodos dessa categoria, são comuns aqueles que efetuam uma fatoração da matriz A do sistema linear. A fatoração consiste na decomposição da matriz no produto de dois ou mais fatores, para depois resolver uma sequência de sistemas lineares de forma mais eficiente computacionalmente. Além de ser usada na resolução de sistemas lineares, a fatoração de matrizes têm interesse em si mesma. Através da fatoração, muitas operações com matrizes são facilitadas, como cálculo do determinante, inversão de matrizes, etc. É uma técnica essencial nas aplicações de álgebra linear computacional, usada, em geral, na manipulação de grandes volumes de dados, com aplicações atuais em ciências de dados e aprendizado de máquinas.

A decomposição ou fatoração de Cholesky [7], que leva o nome do matemático e oficial militar francês André-Louis Cholesky (1875-1918) [3], é uma técnica importante dentro dessa área. Cholesky trabalhava em problemas de topografia e desenvolveu o método para resolver problemas de mínimos quadrados. A primeira publicação do método deveu-se a Ernest Benoît em 1924 [2]. A decomposição de Cholesky é amplamente usada em métodos estatísticos, como análise de regressão, assim como em métodos de simulação e otimização. Essa fatoração é uma variação da decomposição LU (*Lower-Upper*), na qual a matriz A do sistema possui

características específicas: é simétrica e definida positiva. Matrizes simétricas definidas positivas aparecem numa grande variedade de problemas nas ciências. Além de aplicações clássicas de Álgebra Linear, aparecem, por exemplo, em estudos de movimento de sistemas descritos por equações diferenciais, seja em problemas de transporte como muitos problemas na Física, nos quais são estudadas perturbações ao redor de um mínimo de energia. Outro exemplo muito comum encontra-se em áreas relacionadas a Estatística e Probabilidade, matrizes de covariâncias são sempre simétricas e definidas positivas

A fatoração de Cholesky é amplamente usada e forma parte das ementas clássicas na disciplina de Cálculo Numérico para diferentes cursos de ensino superior. Contudo, o ensino do funcionamento do método e das etapas envolvidas na fatoração pode ser desafiador: tais cursos possuem alunos com uma ampla variedade de perfis, de modo que alguns conseguem rapidamente captar os detalhes, enquanto outros têm maior dificuldade em assimilar passos específicos. Isso repercute diretamente nas aulas, pois a explicação do método repetidas vezes toma um tempo de aula muitas vezes não previsto. Além disso, o processo de aprendizagem fica prejudicado, pois alunos que já entenderam a fatoração ficam desmotivados, enquanto alunos que ainda não a entenderam podem se sentir constrangidos e às vezes não manifestar o não-entendimento.

Com o objetivo de apoiar o ensino e aprendizagem da decomposição de Cholesky, este trabalho apresenta a implementação do método na linguagem TuPy [6]. Essa linguagem é uma ferramenta online que possibilita que a implementação do código seja feita em português e também que seja observado o passo a passo da execução do programa. Isso facilita o entendimento do aluno não somente em relação à escrita do código, mas também ao esquema algorítmico da fatoração.

O trabalho está organizado na seguinte forma. Na próxima seção, são apresentados alguns aspectos teóricos da fatoração de Cholesky, bem como as suas restrições. Na Seção 3 é discutida a importância do uso de recursos de visualização e introduzida a ferramenta utilizada na implementação. Na seção seguinte é apresentado o código do algoritmo desenvolvido na linguagem TuPy, assim como figuras que ilustram o andamento da fatoração, possibilitando a visualização da interface da ferramenta para implementação e da interface de execução do código. Por fim, na Seção 5, encontram-se as conclusões do trabalho.

2 Fatoração de Cholesky

Como foi indicado na seção anterior, a fatoração de matrizes desempenha um importante papel em Álgebra Linear Computacional. A fatoração de Cholesky é uma variação da decomposição LU , porém, com algumas restrições quanto à matriz A que se deseja fatorar, que deve ser simétrica e definida positiva. A seguir, são definidos alguns conceitos e aspectos teóricos necessários na compreensão e utilização do método. A exposição é baseada nas referências [1, 7, 11]. Essa apresentação é útil para definir a notação usada e apresentar um texto autossuficiente, servindo como um guia rápido dos conceitos básicos de álgebra linear necessários para a compreensão do que se segue.

Seja \mathbb{R} o conjunto dos números reais. Denota-se o espaço vetorial de todas as matrizes reais de dimensão $m \times n$ por $\mathbb{R}^{m \times n}$. Os elementos da matriz A são escritos como $a_{ij} \in \mathbb{R}$, onde i refere-se à linha e j , à coluna da matriz. Escreve-se, às vezes, $A = (a_{ij})$ para denotar a matriz.

A transposição é uma das operações básicas de matrizes. Na transposição, as linhas da matriz são transformadas em colunas e as colunas em linhas.

Definição 2.1. Se $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, então $B = (b_{ij}) = A^T$, onde $b_{ij} = a_{ji}$, é a transposta de A .

Nota-se que a matriz $A^T \in \mathbb{R}^{n \times m}$.

Definição 2.2. Uma matriz simétrica é uma matriz quadrada tal que $A = A^T$, *i.e.*, $a_{ij} = a_{ji}$ para todo i e j .

Matrizes simétricas são atrativas desde o ponto de vista computacional, pois para elas é usado apenas metade do armazenamento que para uma outra matriz qualquer. Igualmente, de forma geral, é necessária apenas a metade do tempo nos cálculos envolvidos.

Definição 2.3. Uma matriz é dita triangular quando os elementos acima ou abaixo da diagonal principal são iguais a zero. São chamadas de matriz triangular inferior e matriz triangular superior, respectivamente.

Existem vários critérios para definir uma matriz definida positiva. Uma apresentação exaustiva fica fora do interesse nesse trabalho. Assim, optamos por definir da seguinte forma:

Definição 2.4. Seja A uma matriz quadrada de dimensão n . A é definida positiva se $\mathbf{x}^T A \mathbf{x} > 0$ para todo $\mathbf{x} \in \mathbb{R}^n$ não nulo.

Usando a definição anterior, pode ser demonstrado o seguinte Teorema, que estabelece uma equivalência entre a matriz definida positiva e a sua fatoração. A demonstração do Teorema pode ser encontrada em [1, p.117]. A fatoração de Cholesky é obtida como um caso particular.

Teorema 2.5. *Seja A uma matriz real simétrica, de dimensão n . Então A é definida positiva se e somente se existe uma matriz triangular inferior invertível G tal que $A = GG^T$. Os elementos da diagonal de G , g_{ii} , com $1 \leq i \leq n$ podem ser escolhidos como sendo números positivos.*

Observação: A decomposição com elementos g_{ii} positivos se chama de **Fatoração de Cholesky**. Pode ser demonstrado que essa decomposição é única.

No exemplo a seguir mostra-se que existem várias possibilidades de decomposição no caso em que a exigência de ter todos os elementos g_{ii} positivos não seja satisfeita.

$$A = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}, \quad G = \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} -2 & 0 \\ -1 & -1 \end{pmatrix}, \quad G = \begin{pmatrix} -2 & 0 \\ -1 & 1 \end{pmatrix}$$

Seja A uma matriz simétrica e definida positiva, então, pelo Teorema 2.5, ela possui uma decomposição de Cholesky e pode ser decomposta como $A = GG^T$, com elementos positivos na diagonal da matriz G . Isso é representado da seguinte forma:

$$\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{21} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} g_{11} & 0 & \cdots & 0 \\ g_{21} & g_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix} \begin{pmatrix} g_{11} & g_{21} & \cdots & g_{n1} \\ 0 & g_{22} & \cdots & g_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & g_{nn} \end{pmatrix} \quad (1)$$

Na representação da matriz A em (1) já foi considerado o fato de $a_{ij} = a_{ji}$. A seguir, será feita uma dedução detalhada das expressões necessárias para calcular os elementos da matriz G . Essa dedução é baseada na apresentação oferecida em [11, p.151-152]. Leitores mais avançados podem ir diretamente nas expressões (3), (4) e (5).

A identificação dos elementos da matriz G é baseada no produto matricial representado em (1). Inicialmente, será definida a forma para encontrar os elementos da primeira coluna de G . Em seguida, serão definidos os elementos da segunda coluna de G e assim, sucessivamente, até encontrar os elementos da última coluna de G , cujo único elemento não nulo é g_{nn} , chegando-se a uma forma geral para encontrá-los.

Efetuando-se o produto das matrizes G e G^T e igualando à matriz A , a começar

com o produto da primeira linha de G pela primeira coluna de G^T , obtemos:

$$a_{11} = g_{11}g_{11} \longrightarrow a_{11} = g_{11}^2 \quad (2)$$

Para cumprir a condição da fatoração de Cholesky, a diagonal da matriz G tem de ser positiva. Dessa forma, será escolhida a raiz positiva na expressão (2), $g_{11} = \sqrt{a_{11}}$.

Agora que o elemento g_{11} foi encontrado, é possível obter os outros elementos da primeira coluna de G da seguinte forma,

$$\begin{aligned} a_{21} = g_{21}g_{11} &\longrightarrow g_{21} = \frac{a_{21}}{g_{11}} \\ &\vdots \\ a_{n1} = g_{n1}g_{11} &\longrightarrow g_{n1} = \frac{a_{n1}}{g_{11}} \end{aligned}$$

Logo, o caso geral, para a primeira coluna de G , resulta em:

$$g_{i1} = \frac{a_{i1}}{g_{11}}, \quad \text{para } 2 \leq i \leq n. \quad (3)$$

Encontrados os elementos da primeira coluna, procuram-se os elementos da segunda coluna de G . Como a matriz A é simétrica, não é necessário realizar o produto matricial para, por exemplo, o caso $a_{12} = g_{11}g_{21}$, pois $a_{12} = a_{21}$, e o elemento g_{21} já foi calculado anteriormente. Então, iniciando o cálculo dos elementos da segunda coluna pelo elemento da diagonal, tem-se

$$a_{22} = g_{21}g_{21} + g_{22}g_{22} \longrightarrow a_{22} = g_{21}^2 + g_{22}^2 \longrightarrow g_{22}^2 = a_{22} - g_{21}^2$$

Novamente, para satisfazer a condição de que a matriz G possua diagonal positiva, escolhemos g_{22} como sendo a raiz positiva da expressão $\sqrt{a_{22} - g_{21}^2}$. Por outra parte, como está se trabalhando com matrizes reais, $a_{22} - g_{21}^2$ não poderá ser negativo. O fato da matriz A ser positiva definida garante a condição $a_{22} - g_{21}^2 > 0$.

Uma vez encontrado o elemento da diagonal, podem-se obter os elementos restantes da segunda coluna de G ,

$$\begin{aligned} a_{32} = g_{31}g_{21} + g_{32}g_{22} &\longrightarrow g_{32} = \frac{a_{32} - g_{31}g_{21}}{g_{22}} \\ &\vdots \\ a_{n2} = g_{n1}g_{21} + g_{n2}g_{22} &\longrightarrow g_{n2} = \frac{a_{n2} - g_{n1}g_{21}}{g_{22}} \end{aligned}$$

Dessa forma, chega-se, no caso geral, à seguinte expressão para os elementos da

segunda coluna,

$$g_{i2} = \frac{1}{g_{22}} [a_{i2} - g_{i1}g_{21}] \quad \text{para } 3 \leq i \leq n.$$

.

Por fim, generalizando os procedimentos feitos anteriormente, obtem-se a forma geral para encontrar todos os elementos da matriz G . No caso geral, para os elementos da diagonal da matriz G , tem-se

$$g_{jj} = \begin{cases} \sqrt{a_{11}} & \text{para } j = 1 \\ \left[a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2 \right]^{\frac{1}{2}} & \text{para } 2 \leq j \leq n. \end{cases} \quad (4)$$

Para completar os elementos das colunas de G , tem-se a expressão (3) como forma geral da primeira coluna, sendo o caso geral para as seguintes colunas, *i.e.*, para $j = 2, 3, \dots, n-1$, dado por

$$g_{ij} = \frac{1}{g_{jj}} \left[a_{ij} - \sum_{k=1}^{j-1} g_{ik}g_{jk} \right] \quad \text{para } j+1 \leq i \leq n, \quad (5)$$

com os g_{jj} calculados usando a expressão (4). O índice j vai até $n-1$, visto que na n -ésima coluna só precisa ser calculado o elemento g_{nn} , o que é feito através da expressão (4).

Portanto, usando as expressões gerais (3), (4) e (5), é possível encontrar todos os elementos da matriz G , observando que o último cálculo a ser efetuado é na determinação do elemento g_{jj} , $j = n$, através da expressão (4). Nessa etapa, já são conhecidos todos os elementos das colunas da matriz G com índice menor do que n . A matriz G é conhecida como fator de Cholesky. Essas expressões são a base do algoritmo que foi implementado [1, 11], do que trata a Seção 4.

3 O Uso de Recursos de Visualização

A visualização de informação é uma área de estudo recente e refere-se ao uso de representações visuais de dados abstratos para amplificar a cognição [4]. As visualizações efetivas ajudam os observadores a identificar rapidamente informações relevantes e a entender as relações do que estão vendo [12]. A visualização de software é considerada uma subárea da visualização de informação. Nesse caso, o objetivo é tornar

mais compreensíveis a estrutura, o comportamento e a evolução do software, como a organização do código, o estado do software e os erros apresentados no decorrer de sua execução. Os desenvolvedores utilizam a visualização de software para melhorar a qualidade do software e aumentar o conhecimento em relação aos artefatos de software [10].

Observa-se um grande potencial no uso de representações visuais para explicar conceitos complexos na educação em Matemática, de forma a facilitar mecanismos de raciocínio. Em particular, em áreas da matemática aplicada, como o cálculo numérico, as representações visuais tornam-se úteis na explicação dos passos de execução dos métodos usados e, talvez, nas adaptações necessárias de resultados teóricos. Unindo a representação da fatoração de Cholesky como algoritmo e a visualização de software aplicada ao mesmo, é possível obter grandes benefícios no que tange ao aprendizado, diminuindo consideravelmente o esforço cognitivo na compreensão de nuances relacionadas aos diferentes passos.

3.1 TuPy Online

O *TuPy Online*¹ é um ambiente de desenvolvimento (IDE) online [6], baseado no trabalho de Guo [8] e foi adaptado por professores pesquisadores e alunos do IME-UERJ com o objetivo de auxiliar a aprendizagem dos alunos no estudo de algoritmos [9]. A ferramenta trabalha com pseudocódigos e é uma linguagem de programação mínima inspirada, em parte, por Python, porém o código é escrito em português. Ele permite acompanhar o passo a passo da execução do algoritmo, mostrando o conteúdo das variáveis envolvidas e também a visualização simultânea das estruturas dos dados correspondentes, já que é integrado à ferramenta *Graphviz*² para visualização de gráficos.

A ferramenta foi desenvolvida no contexto de software livre baseada no *Online Python Tutor*³ e pode ser usada diretamente na rede ou a partir de uma cópia local. Além disso, o TuPy Online é uma iniciativa de código aberto e seu código-fonte está disponível no *GitHub* [6] para quem desejar contribuir em seu desenvolvimento.

¹<https://gvirtu.github.io/TuPyOnline/>

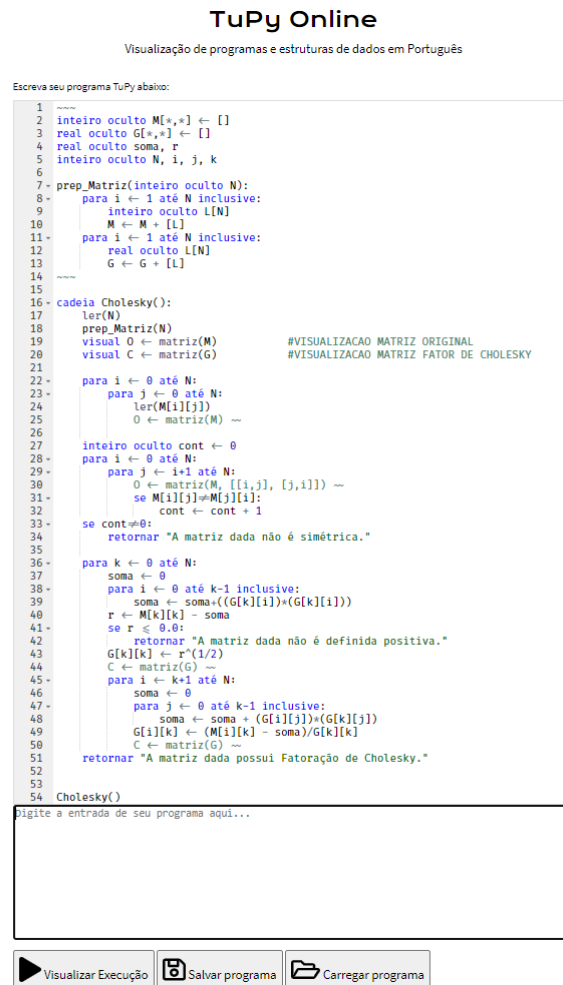
²<https://www.graphviz.org/>

³<http://www.pythontutor.com/>

4 Algoritmo da fatoração de Cholesky

Nesse trabalho foi implementado o algoritmo da fatoração de Cholesky baseado nas expressões (3), (4) e (5) da Seção 2, utilizando a ferramenta TuPy. Na Figura 1, pode-se observar a interface inicial da linguagem e o código do programa. Uma versão básica do algoritmo pode ser encontrada em. [11, pág. 153].

Figura 1: Interface de edição do TuPy Online

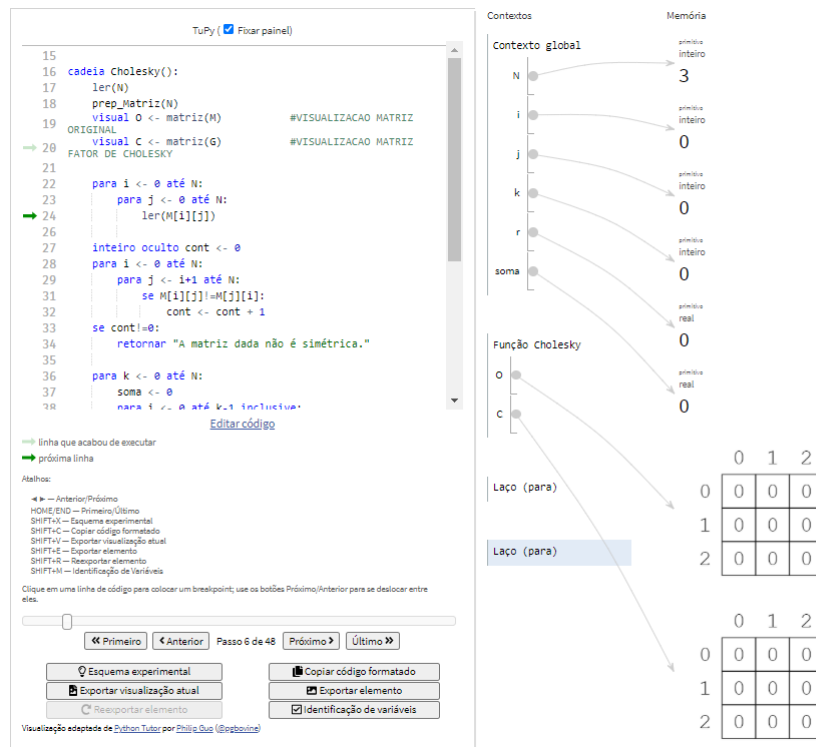


Fonte: Autoral.

Inicialmente foi feita a declaração das variáveis que serão utilizadas. Elas foram declaradas de forma *oculta*. Esse recurso é utilizado para que as variáveis, assim declaradas, não apareçam de forma explícita no decorrer da visualização da execução do código. Para utilizá-lo, é necessário declarar a variável no código com o seu tipo e a palavra *oculto* logo após; por exemplo, se declaramos uma variável do tipo inteiro, declaramos como *inteiro oculto*. Contudo, esse recurso pode não ser utilizado, indicando na visualização os valores que as variáveis possuem a cada

iteração, como pode ser observado na Figura 2.

Figura 2: Execução do código mostrando as variáveis



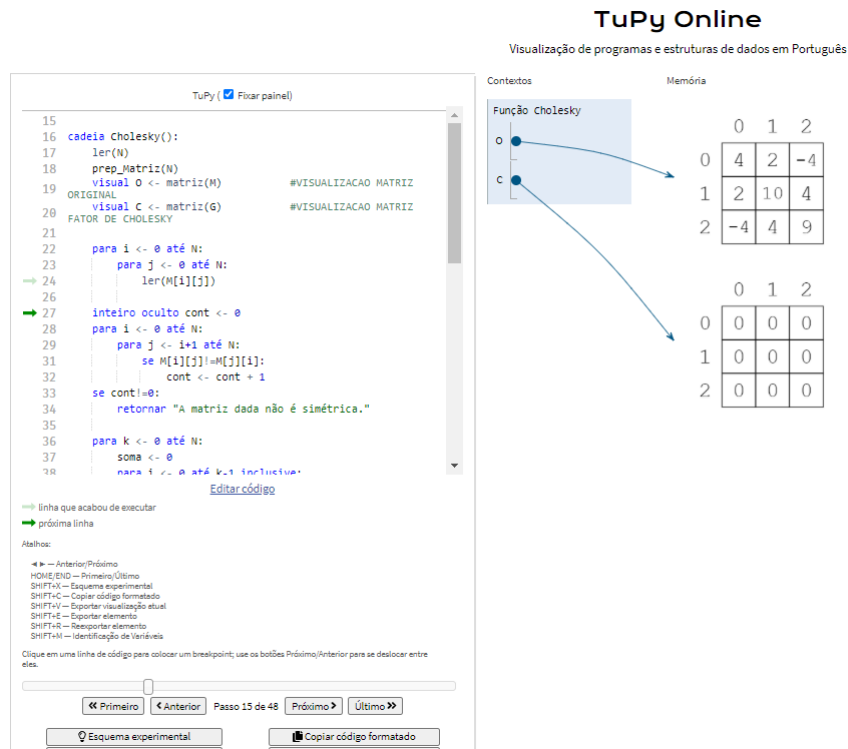
Fonte: Autoral.

Em seguida, foi realizada a implementação de uma função chamada *prep_Matriz* para a inicialização das matrizes que serão utilizadas no decorrer da execução do código. A criação de matrizes no TuPy é feita a partir de vetores. Na obtenção de uma visualização personalizada das matrizes se faz necessário utilizar a função *visual*, que recebe os valores das matrizes e as mostra de forma gráfica. Essa função foi utilizada internamente na função “*Cholesky()*”, a qual foi criada como uma função *cadeia*.

Uma cadeia em TuPy representa cadeias de caracteres, comumente conhecidas como *strings*, pela sua denominação em inglês. Com isso, a função implementada retorna uma mensagem indicando se é possível realizar a fatoração ou não, a depender das propriedades da matriz de entrada. As possíveis mensagens de retorno serão destacadas mais na frente.

Antes de inicializar a execução do código é necessário inserir a ordem da matriz de entrada, a variável *N*, e também os elementos dessa matriz. Esses valores devem ser inseridos na caixa de texto apresentada na Figura 1, separados por espaço, como entrada do programa. Na Figura 3 pode-se observar que foi realizada a leitura dos valores inseridos na caixa de texto e apresentada de forma gráfica.

Figura 3: Leitura dos elementos da matriz dos coeficientes



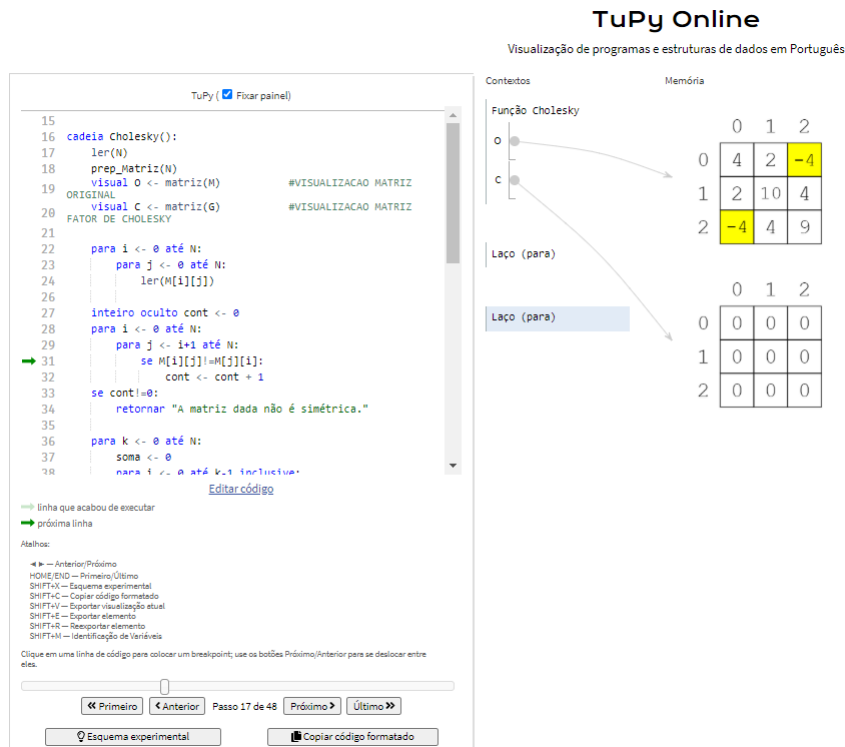
Fonte: Autoral.

No algoritmo foi feita uma adaptação para determinar se a matriz A é simétrica. Esse aspecto também foi considerado na parte da visualização, onde implementamos um recurso que destaca os elementos que estão sendo comparados para avaliar a simetria. Isso pode ser observado na Figura 4. Caso a matriz seja simétrica, o algoritmo prossegue com a execução do código, caso contrário, o algoritmo retorna a mensagem “*A matriz dada não é simétrica*”.

É importante notar que, como o TuPy é inspirado na linguagem Python, a inicialização dos índices é feita a partir do valor 0. Quando definimos, por exemplo, *para $i \leftarrow 0$ até N* ;, a iteração da variável i será de 0 até $N - 1$, pois a linguagem não considera que o N está incluso no intervalo de iteração.

Por outra parte, de acordo com o Teorema 2.5, para verificar a propriedade da matriz ser definida positiva e, ao mesmo tempo, obter uma fatoração de Cholesky, os elementos da diagonal principal da matriz G têm que ser escolhidos de modo a serem estritamente positivos. Como foi indicado na Seção 2, isso é feito escolhendo a raiz positiva na expressão (4). Além disso, precisa-se ter também solução real da raiz quadrada considerada. A nível do algoritmo, essa verificação é feita analisando o sinal do argumento da expressão (4), *i.e.*, o que está dentro da raiz quadrada, que é considerado no código pela variável r . Por exemplo, se considerarmos o cálculo do

Figura 4: Comparação dos elementos para avaliar a simetria



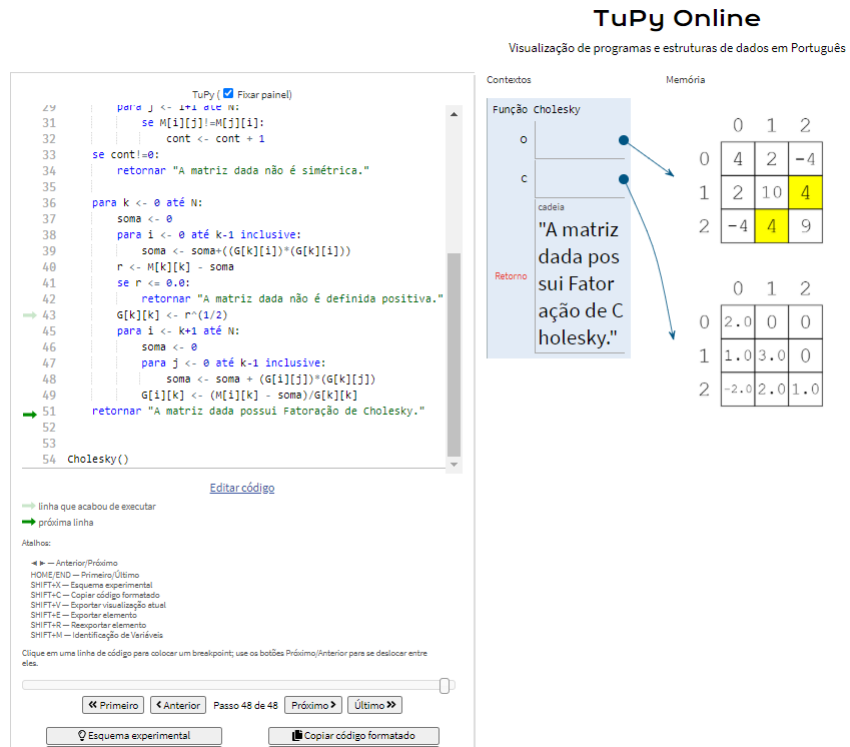
Fonte: Autoral.

elemento g_{kk} , temos que $g_{kk} = \sqrt{a_{kk} - g_{k1}^2 - g_{k2}^2 - \dots - g_{k,k-1}^2}$, onde $r = a_{kk} - g_{k1}^2 - g_{k2}^2 - \dots - g_{k,k-1}^2$. Assim, analisa-se o sinal de r . Se a condição $r \leq 0$ for verdadeira, o processo de fatoração será interrompido e o código retornará a mensagem “A matriz dada não é definida positiva”. Quando o processo for completado satisfatoriamente, será obtida a matriz G , o fator de Cholesky, e retornada a seguinte mensagem “A matriz dada possui Fatoração de Cholesky”.

No processo de ensino de algoritmos, como os de fatoração de matrizes, se faz difícil para os alunos o entendimento das expressões gerais como as apresentadas em (3), (4) e (5), ainda mais quando índices são usados. O código aqui desenvolvido pode ser usado de forma auxiliar no processo de ensino. Na Figura 5 pode-se observar a execução do código finalizada, utilizando como exemplo uma matriz que possui fatoração de Cholesky.

O programa computacional permite acompanhar o passo a passo da execução, como também voltar a um passo que não tenha sido totalmente entendido, aumentando o seu poder didático. De fato, é possível voltar quantos passos for necessário e ver, dinamicamente, o que está sendo feito. Assim, a contribuição fundamental da ferramenta é a de permitir o acompanhamento visual de cada etapa do cálculo dos

Figura 5: Interface da ferramenta com o código finalizado



Fonte: Autoral.

elementos da matriz G , facilitando o entendimento do algoritmo.

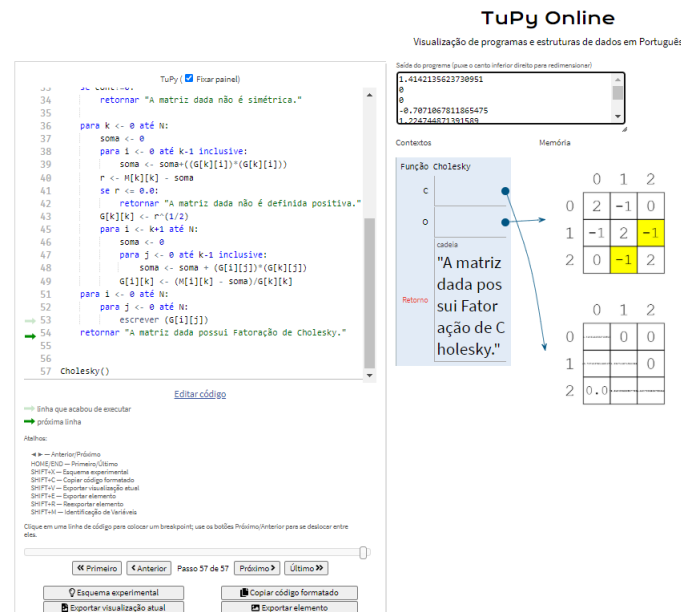
Vale notar que, ao se trabalhar com números reais, algumas matrizes podem ter como fator de Cholesky uma matriz com números de muitos dígitos, os quais não se adequam ao tamanho padronizado da visualização no TuPy. Na Figura 6, mostramos um exemplo simples disso. Com a intenção de contornar essa limitação visual, foi inserida uma caixa de diálogo na parte superior, onde são listados os valores, por linha, da matriz final G . Contudo, ressalta-se que o objetivo do código é didático, com a intenção de mostrar visualmente a evolução da fatoração, ajudando no melhor entendimento do algoritmo. Assim, o docente que estiver usando o aplicativo no processo de ensino, pode antecipadamente encontrar exemplos de matrizes que produzam um resultado útil visualmente.

A fatoração de Cholesky requer $n^3/6 + \mathcal{O}(n^2)$ operações no desenvolvimento do algoritmo para o cálculo dos fatores. Assim, para n grande, serão realizadas aproximadamente metade das operações necessárias quando comparado com o algoritmo clássico da Eliminação Gaussiana.

O código desenvolvido encontra-se disponível [5] no GitHub ⁴ e divulgado no

⁴https://github.com/miiichellelau/cholesky_tupy

Figura 6: Visualização de matrizes com muitos dígitos



Fonte: Autoral.

site do Programa de Pós-Graduação em Ciências Computacionais da UERJ (PPG-CCOMP) ⁵, possibilitando que outras pessoas possam utilizá-lo para fins didáticos.

5 Conclusão

A ferramenta TuPy Online é uma grande aliada para o ensino de algoritmos, já que possibilita a visualização do passo a passo da execução do código. Nesse trabalho, considera-se uma aplicação da ferramenta no ensino de Cálculo Numérico, onde são estudados vários tipos de fatoração para resolver sistemas lineares. No algoritmo básico da fatoração de Cholesky encontrado na literatura foram acrescentadas condições para comprovar que a matriz de coeficientes do sistema linear cumpre os requerimentos para o cálculo do fator de Cholesky. Com o auxílio do TuPy Online é possível visualizar cada etapa da fatoração, facilitando a compreensão do método.

Com vistas a avaliar a utilidade do trabalho desenvolvido, foi apresentado em sala de aula para uma turma mista de Cálculo Numérico, com alunos de graduação dos cursos de Estatística, Ciências Atuariais e Ciência da Computação, da UERJ. Os alunos se mostraram interessados no decorrer da atividade. Com o intuito de avaliar a eficácia da ferramenta na aprendizagem, encontra-se um projeto em andamento, que prevê um protocolo de experimentação em várias turmas de graduação.

⁵<https://ccomp.ime.uerj.br/producao-tecnologica/>

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e da Fundação de Amparo à Pesquisa do Rio de Janeiro (FAPERJ). Agradecemos as sugestões e comentários recebidos no processo de revisão, que beneficiaram o trabalho.

Referências

- [1] ACKLEH, A. S. et al.: **Classical and modern numerical analysis; Theory, Methods and Practice**. CRC Press, Taylor & Francis, 2009.
- [2] BREZINSKI, C.; TOURNÈS, D.: **André-Louis Cholesky: Mathematician, topographer and Army Officer**. Birkhäuser, Springer Switzerland, 2014.
- [3] BREZINSKI, C.; WUYTACK, L.: Numerical Analysis in the twentieth century, **Numerical Analysis: Historical Developments in the 20th Century**. Elsevier Science B. V., 2001
- [4] CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. (Ed.): **Readings in information visualization: using vision to think**. Morgan Kaufmann, 1999.
- [5] CÓDIGO-FONTE DA IMPLEMENTAÇÃO DA FATORAÇÃO DE CHOLESKY NA LINGUAGEM TUPY.
Disponível em: https://github.com/miichellelau/cholesky_tupy
Último acesso em: mai.2020.
- [6] CÓDIGO-FONTE DO TUPY ONLINE.
Disponível em: <https://github.com/gVirtu/TuPyOnline>
Último acesso em: mai.2020.
- [7] GOLUB, G. H.; VAN LOAN, C. F.: **Matrix Computations**. 3^a ed., Johns Hopkins University Press, 1996.
- [8] GUO, P. J.. **Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education**. In: 44th ACM Technical Symposium on Computer Science Education (SIGCSE), Denver, CO, USA, pp. 579-584, 2013.

- [9] GOULART, J. L. J.: **Avaliação Sistemática de Eficácia na Aprendizagem de Algoritmos com o uso do TuPy Online**. Dissertação (Mestrado em Ciências Computacionais) - Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.
- [10] PAREDES, J.; ANSLOW, C.; MAURER, F. **Information visualization for agile software development**. In: 2nd IEEE Working Conference on Software Visualization (VISSOFT), Victoria, BC, Canada, pp. 157-166, 2014.
- [11] RUGGIERO, M. A. G; LOPES, V. L. R.: **Cálculo numérico: aspectos teóricos e computacionais**, 2^a ed., Pearson, 2000.
- [12] WARE, C. **Information Visualization: Perception for Design**. Elsevier, 2012.