MAT0122 ALGEBRA LINEAR I

FOLHA DE SOLUÇÃO

Nome: Gabriel Geraldino de Souza **Número USP: 12543885**

SOLUÇÃO

(2) — escreva um algoritmo que calcula a inversa de uma matriz triangular inferior.

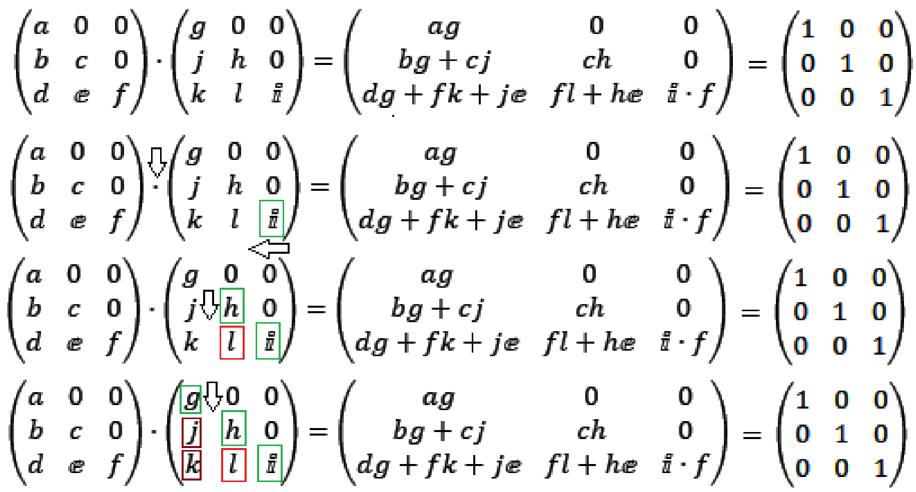
Doravante, M será a matriz a ser invertida e I sua inversa.

É fácil observar que em uma matriz triangular, $I_{i,i} = 1/M_{i,i}$; daí se tem os elementos da diagonal. Também se sabe que a inversa de uma matriz triangular também é uma matriz triangular e, por definição, o produto entre a matriz e sua inversa será a matriz identidade.

Assim, um possível algoritmo para calcular a inversa é:

- Começar da última coluna e ir em direção à primeira;
- Preencher a diagonal desta;
- Abaixo da diagonal, de cima para baixo (em relação às linhas), fazer a soma dos produtos com termos já conhecidos da matriz; o termo a ser calculado será o oposto da divisão dessa soma pela diagonal;
- Continuar a mesma etapa nas linhas debaixo; note que o item descoberto na etapa anterior será utilizado nas próximas linhas, o que justifica o caminho cima-baixo;
- Fazer estes mesmos passos para todas as colunas.

A imagem a seguir tenta demonstrar o processo.



fl+he=0 -> l=-he/f

dg+ej+fk=0 -> k=-(dg+ej)/f

(1) — escreva um algoritmo que calcula a inversa de uma matriz triangular superior.

O seguinte algoritmo será análogo ao do item 2. A única diferença será a direção a ser seguida; Pode ser visto como o mesmo algoritmo, "ao contrário".

- Começar da primeira linha e ir em direção à última;
- Preencher a diagonal desta;
- Calcular os itens da matriz inversa à direita da diagonal com base nos itens já descobertos (soma de produtos e posterior divisão pela diagonal);
- Continuar a mesma etapa para as linhas debaixo.

Para a codificação deste algoritmo, o código foi quase inteiramente reutilizado do item 2, invertendo as posições, a fim de ter o mesmo funcionamento com a ordem descrita.

(3) — escreva um algoritmo que calcula a inversa de uma matriz nxn.

Para este item, foi utilizado o algoritmo já conhecido de Gauss-Jordan.

- À direita da matriz a ser invertida, coloca-se a identidade de mesma ordem;
- Passa-se por todas as linhas e toma essa posição como pivô;

  Para todas as colunas que não sejam uma diagonal da matriz inicial, divide-se o elemento pela diagonal da matriz;

  Passa-se por todas linhas e por todas colunas; soma-se ao elemento nesta posição o oposto do produto desta linha na coluna pivô com a linha pivô desta coluna;
- A matriz inversa estará onde estava inicialmente a matriz identidade.

Note que o cálculo dos elementos com o pivô segue a mesma lógica do cálculo dos elementos aplicado nos itens 1 e 2.

(4) — Implemente o algoritmo do item 1 e teste o seu código para a matrizes aleatórias. Analise o resultado dos seus testes.

A implementação de todos os códigos está no arquivo enviado em anexo junto com este documento. Esta função chama-se 'inverse_upper'.

A função mostrou-se precisa até para matrizes grandes ($\approx$50) e realiza substancialmente menos operações que a função 'inverse'.

(5) — Implemente o algoritmo do item 2 e teste o seu código para a matrizes aleatórias. Analise o resultado dos seus testes.

A implementação de todos os códigos está no arquivo enviado em anexo junto com este documento. Esta função chama-se 'inverse_lower'.

A função mostrou-se precisa até para matrizes grandes ($\approx$50) e realiza substancialmente menos

operações que a função ‘inverse‘.

(6) — Implemente o algoritmo do item 3 e teste o seu código para a matriz de Hilbert de dimensões de 1 a 50. Analise os resultados dos seus testes

A implementação de todos os códigos está no arquivo enviado em anexo junto com este documento. Esta função chama-se ‘inverse‘.

A complexidade de tempo é maior do que para calcular a inversa de matrizes triangulares, uma vez que para matrizes triangulares alguns elementos são pulados.

As matrizes inversas de matrizes de Hilbert contém números de ordem de grandeza suficientemente grande para afetar a precisão na linguagem escolhida. A função apresenta resultados precisos até aproximadamente a dimensão 12; Acima disso, desvios começam a aparecer.

Apenas como palpite, o mesmo código ou escrito em C++ utilizando long double ou mesmo em Python utilizando ‘Fractions‘ ou notação científica talvez traga resultados melhores em questão de precisão; No entanto, creio que esses aperfeiçoamentos não estejam dentro do escopo proposto.