Universidade Estadual De Campinas

MS211 - Cálculo Numérico

Prof. Kelly Cristina Poldi

Projeto Computacional 1

Felipe Lobão Melara RA:247844
Gabriel Mattias Antunes RA:281199

1. Introdução

O seguinte relatório contempla o desenvolvimento do trabalho computacional 1 da matéria MS211 - Cálculo numérico, ministrada pela Prof. Dra.Kelly Cristina Poldi do instituto de matemática, estatística e computação científica da Universidade Estadual de Campinas.

O trabalho explora a fatoração de Cholesky, que decompõe uma matriz simétrica e definida positiva em duas matrizes, uma matriz triangular inferior e sua transposta. Essa decomposição é útil por exemplo para soluções numéricas eficientes e simulações de Monte Carlo.

Busca-se, portanto, por meio deste, implementar um algoritmo capaz de realizar a fatoração nos casos em que a mesma é possível e analisar os casos em que não é.

2. Desenvolvimento

A fatoração de Cholesky busca encontrar as matrizes G e G^t de uma matriz M tal que $M = GG^t$. Para implementar o código, será usada a linguagem de programação python, com a inclusão da biblioteca numpy, que facilita a manipulação de matrizes.

O primeiro passo do trabalho é analisar se a matriz A =

$$\begin{vmatrix} 4 & 6 \\ 6 & 13 \end{vmatrix}$$

é simétrica e positiva. Para isso, foi feito o seguinte código que, dada uma dimensão N e uma matriz quadrada NxN, verifica se a matriz é simétrica, analisando par a par de valores simetricamente opostos:

```python
def simetrica(n, mat):
    for i in range(n):
        for j in range(i, n):
            if mat[i][j] != mat[j][i]:
                return False

    return True
```

Em seguida, foi implementada uma função para analisar se uma dada matriz é definida positiva. A função usa uma função do numpy que calcula os autovalores de uma matriz e depois checa se todos são positivos.

```python
def definida_positiva(mat):
    autovalores = np.linalg.eigvals(mat)

    return np.all(autovalores > 0)
```

Ao executar o código, temos que a matriz A é simétrica e definida positiva.

O próximo passo é realizar a multiplicação das matrizes G e G^t, com os respectivos valores:

$$\begin{vmatrix} g11 & 0 \\ g21 & g22 \end{vmatrix} \qquad \begin{vmatrix} g11 & g21 \\ 0 & g22 \end{vmatrix}$$

Encontrando assim GG^t:

$$\begin{vmatrix} g11^2 & g11.g21 \\ g11.g21 & g21^2 + g22^2 \end{vmatrix}$$

Comparando GG^t com A, obtém-se o seguinte sistema:

$g11^2 = 4 \rightarrow g11 = 2$

$g11.g21 = 6 \rightarrow g21 = 3$

$g21^2 + g22^2 = 13 \rightarrow g22 = 2$

Substituindo $g11$, $g21$ e $g22$ em G e G^t obtém-se:

$$\begin{vmatrix} 2 & 0 \\ 3 & 2 \end{vmatrix} \quad \begin{vmatrix} 2 & 3 \\ 0 & 2 \end{vmatrix}$$

Calculando novamente GG^t:

$$\begin{vmatrix} 4 & 6 \\ 6 & 13 \end{vmatrix}$$

Tendo explorado o funcionamento da fatoração de Cholesky, foi feita a implementação de um algoritmo capaz de realizá-la. Calcula-se o valor de cada valor da matriz triangular inferior G, quando esse valor faz parte da diagonal principal, cujos valores devem ser todos positivos para que a matriz seja definida positiva, ele é calculado como a raiz quadrada do elemento correspondente da matriz original, subtraído pelo produto escalar das partes relevantes da matriz G. Já quando o valor não faz parte da diagonal principal, ele é calculado como o valor correspondente da matriz original, subtraído subtraído pelo produto escalar das partes relevantes da matriz G e dividido pelo valor da diagonal correspondente, de forma a garantir que se obtenha uma matriz triangular inferior. Nota-se que divisões por zero podem ocorrer, interrompendo o algoritmo caso a matriz sendo fatorada não seja definida positiva e simétrica. Com isso, obtém-se a matriz G.

```
def cholesky(n, mat):
    G = [[0.0] * n for _ in range(n)]

    for i in range(n):
        for j in range(i + 1):
            if i == j:
                soma_diag = sum(G[i][k] ** 2 for k in range(j))
                valor = mat[i][i] - soma_diag
                if valor <= 0:
                    raise ValueError("matriz não definida positiva")
                G[i][j] = math.sqrt(valor)

            else:
                soma_n_diag = sum(G[i][k] * G[j][k] for k in
range(j))
```

```
                if G[j][j] == 0:
                    raise ZeroDivisionError("divisão por zero em
G[j][j]")
                G[i][j] = (mat[i][j] - soma_n_diag) / G[j][j]

    return G
```

3. Conclusões

Portanto, a fatoração de Cholesky provou-se ser de fácil implementação e uma forma mais simples de descobrir se uma matriz é definida positiva, podendo ser implementada sem muito conhecimento sobre computação. Apesar de realizar muitas operações, em matrizes de dimensões maiores, muito possivelmente a fatoração de Cholesky deve performar mais rápido do que a abordagem tradicional de checar se uma matriz é definida positiva, uma vez que ela não requer a solução de equações para encontrar os autovalores.

Conclui-se então que a fatoração de Cholesky é uma maneira mais eficiente de definir se uma matriz é definida positiva por ser um método iterativo. Além disso, é possível melhorar ainda mais a performance do código caso fossem utilizadas mais funções da biblioteca Numpy.