1. L\*L^t

1. Hàm nhập ma trận (inputMatrix):

Input: Ma trận A kích thước n x m.

Output: Nhập các giá trị cho ma trận A từ người dùng.

Hàm inputMatrix(A, n, m):

Nếu n != m, thông báo "Ma trận không phải hình vuông".

Nhập các giá trị của ma trận A (n x m) từ người dùng.

2. Hàm kiểm tra ma trận có thể phân rã Cholesky (checkMatrixCholesky):

Input: Ma trận A kích thước n x m.

Output: Trả về true nếu ma trận có thể phân rã Cholesky, ngược lại false.

Hàm checkMatrixCholesky(A, n, m):

Nếu n != m, trả về false.

Nếu ma trận không đối xứng, trả về false.

Nếu ma trận không dương xác định, trả về false.

Trả về true nếu ma trận thỏa mãn điều kiện.

3. Hàm kiểm tra ma trận đối xứng (SymmetricalMatrix):

Input: Ma trận A kích thước n x n.

Output: Trả về true nếu ma trận A đối xứng, ngược lại false.

Hàm SymmetricalMatrix(A, n):

Duyệt qua từng phần tử A[i][j]:

Nếu A[i][j] != A[j][i], trả về false.

Trả về true nếu tất cả phần tử đối xứng.

4. Hàm đổi chỗ hai phần tử (swap):

Input: Hai tham số a và b.

Output: Đổi giá trị của a và b.

Hàm swap(a, b):

Đổi giá trị giữa a và b.

5. Hàm tính định thức của ma trận (DetMatrix):

Input: Ma trận A kích thước n x n.

Output: Trả về định thức của ma trận A.

Hàm DetMatrix(A, n):

Tạo một bản sao của ma trận A.

Tính định thức của ma trận bằng phương pháp Gauss.

Nếu có hàng hoặc cột có giá trị bằng 0, kiểm tra hoán vị các hàng để làm không có số 0 trên đường chéo chính.

Trả về định thức của ma trận.

6. Hàm kiểm tra ma trận dương xác định (checkpositive):

Input: Ma trận A kích thước n x n.

Output: Trả về true nếu ma trận là dương xác định, ngược lại false.

Hàm checkpositive(A, n):

Duyệt qua các ma trận con bậc k từ 1 đến n:

Tính định thức của ma trận con này.

Nếu định thức nhỏ hơn hoặc bằng 0, trả về false.

Trả về true nếu tất cả các ma trận con đều có định thức dương.

7. Hàm phân rã Cholesky (Cholesky\_Decomposition):

Input: Ma trận A kích thước n x n.

Output: Phân rã ma trận A thành L (ma trận tam giác dưới) sao cho A = L \* L^T.

Hàm Cholesky\_Decomposition(A, n):

Khởi tạo ma trận L bằng 0.

Duyệt qua tất cả các phần tử A[i][j] để tính giá trị tương ứng của ma trận L.

Nếu i == j, tính giá trị L[i][i] dựa trên các giá trị trước đó.

Nếu i != j, tính giá trị L[i][j] dựa trên các giá trị trước đó.

Trả về ma trận L sau khi phân rã.

8. Hàm in ma trận (printMatrix):

Input: Ma trận L kích thước n x n.

Output: In ma trận L và L^T ra màn hình.

Hàm printMatrix(L, n):

In ra ma trận L và L^T theo định dạng cụ thể.

Sử dụng setprecision để in các giá trị với độ chính xác 2 chữ số sau dấu thập phân.

In ma trận theo dạng ma trận tam giác dưới và ma trận chuyển vị của nó.

9. Hàm main:

Input: Nhập giá trị n và m từ người dùng, sau đó nhập ma trận A.

Output: Kiểm tra nếu ma trận có thể phân rã Cholesky, nếu không thông báo lỗi. Nếu có, thực hiện phân rã Cholesky và in kết quả.

Hàm main:

Nhập n và m từ người dùng.

Gọi hàm inputMatrix để nhập ma trận A.

Nếu không thể phân rã Cholesky, thông báo lỗi và kết thúc chương trình.

Gọi hàm Cholesky\_Decomposition để phân rã ma trận A thành ma trận L.

In kết quả ma trận L và L^T bằng cách gọi hàm printMatrix.

1. L D L^T

BẮT ĐẦU

ĐỊNH NGHĨA HÀM inputMatrix(A, n, m):

KIỂM TRA nếu n khác m:

IN "Ma trận không phải là ma trận vuông"

TRẢ VỀ

IN "Nhập ma trận A: "

LẶP qua i từ 0 đến n-1:

LẶP qua j từ 0 đến m-1:

IN "Nhập A[i][j]: "

NHẬP giá trị vào A[i][j]

KẾT THÚC LẶP

KẾT THÚC HÀM

ĐỊNH NGHĨA HÀM isSymmetric(A, n):

LẶP qua i từ 0 đến n-1:

LẶP qua j từ 0 đến n-1:

KIỂM TRA nếu A[i][j] không bằng A[j][i]:

TRẢ VỀ FALSE

TRẢ VỀ TRUE

KẾT THÚC HÀM

ĐỊNH NGHĨA HÀM checkMatrixCholesky(A, n, m):

KIỂM TRA nếu n khác m:

TRẢ VỀ FALSE

KIỂM TRA nếu isSymmetric(A, n) là FALSE:

TRẢ VỀ FALSE

TRẢ VỀ TRUE

KẾT THÚC HÀM

ĐỊNH NGHĨA HÀM CholeskyLDL\_Decomposition(A, L, D, n):

KHỞI TẠO ma trận L là ma trận 0

LẶP qua j từ 0 đến n-1:

LẶP qua k từ 0 đến j-1:

A[j][j] -= L[j][k] \* L[j][k] \* D[k]

KẾT THÚC LẶP

D[j] = A[j][j]

LẶP qua i từ j+1 đến n-1:

L[i][j] = A[i][j]

LẶP qua k từ 0 đến j-1:

L[i][j] -= L[i][k] \* L[j][k] \* D[k]

KẾT THÚC LẶP

L[i][j] /= D[j]

KẾT THÚC LẶP

L[j][j] = 1

KẾT THÚC LẶP

KẾT THÚC HÀM

ĐỊNH NGHĨA HÀM printMatrix(L, D, n):

IN "Ma trận L | D | L^T"

LẶP qua i từ 0 đến n-1:

LẶP qua j từ 0 đến n-1:

KIỂM TRA nếu j lớn hơn i:

IN 0.00

KHÁC IN L[i][j]

KẾT THÚC LẶP

IN " | "

LẶP qua j từ 0 đến n-1:

KIỂM TRA nếu i bằng j:

IN D[i]

KHÁC IN 0.00

KẾT THÚC LẶP

IN " | "

LẶP qua j từ 0 đến n-1:

KIỂM TRA nếu j nhỏ hơn i:

IN 0.00

KHÁC IN L[j][i]

KẾT THÚC LẶP

IN dòng mới

KẾT THÚC LẶP

KẾT THÚC HÀM

BẮT ĐẦU CHÍNH:

KHAI BÁO ma trận A[n][n], L[n][n], D[n]

KHAI BÁO n, m

IN "Nhập kích thước ma trận (n): "

NHẬP n

IN "Nhập kích thước ma trận (m): "

NHẬP m

KIỂM TRA nếu checkMatrixCholesky(A, n, m) là FALSE:

IN "Ma trận không phải là ma trận xác định dương"

KẾT THÚC

KHÁC:

GỌI hàm CholeskyLDL\_Decomposition(A, L, D, n)

GỌI hàm printMatrix(L, D, n)

KẾT THÚC

KẾT THÚC