**BAB I**

**PENDAHULUAN**

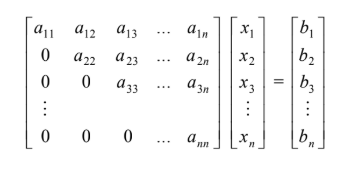
1. **Tujuan**
2. Dapat menentukan penyelesaian sistem persamaan linear dengan metode Gauss, metode Decomposisi Choleski.
3. Mencari besarnya kesalahan dari suatu perhitungan solusi sistem persamaan linear dengan metode Gauss, dan metode Decomposisi Choleski.
4. **Rumusan Masalah**
5. Mencari solusi sistem persamaan linear jika matriks dan dengan menggunakan:
6. Metode Eliminasi Gauss
7. Metode Cholesky
8. Bagaimana cara menentukan penyelesaian sistem persamaan linear dengan metode Gauss dan metode Decomposisi Choleski?
9. Bagaimana cara mencari atau menentukan besarnya kesalahan dari suatu perhitungan solusi sistem persamaan linear dengan metode Gauss dan metode Decomposisi choleski?

**BAB II**

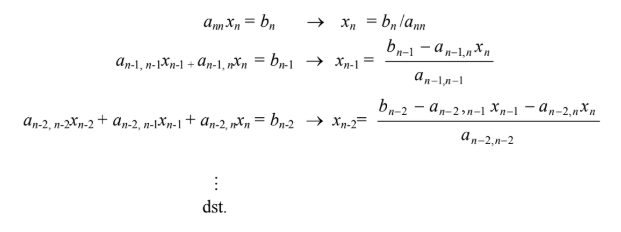
**DASAR TEORI**

1. **Metode Eliminasi Gauss**

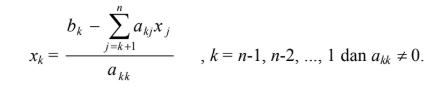
Metode ini bermula dari kenyataan bahwa bila matriks A berberntuk segitiga atas seperti sistem persamaan berikut ini



Maka solusinya dapat dihitung dengan teknik penyulihan mundur (*backward substitution*):



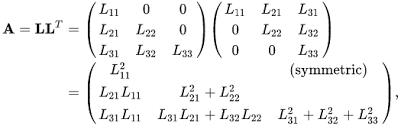
Sekali diketahui, maka nilai dapat dihitung dengan



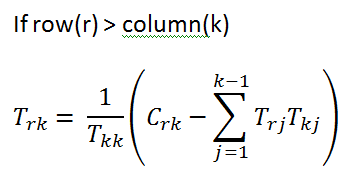
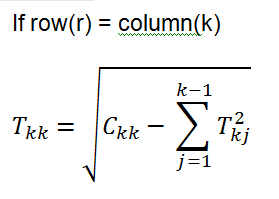
1. **Metode Decomposisi Cholesky**

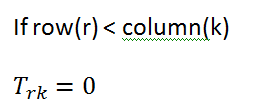
Metode Cholesky adalah sebuah penyelesaian persamaan linier simultan yang diperoleh dari rumusan matematika berdasarkan atas unsur koefisien variable yang simetris. Matriks yang diselesaikan harus matriks yang berordo sama atau yang biasa disebut adalah matriks simetris. Unsur matriks baris sama dengan unsur matriks kolom pada indeks baris dan kolom yang sama. Nilai di dalam tanda akar harus bernilai positif. Angka diluar diagonal utama harus memiliki nilai yang sama.

Misalkan terdapat persamaan matriks AX = B dengan X adalah matriks yang dicari. Langkah pertama dari metode decomposisi Choleski adalah membuat matriks segitiga atas dan segitiga bawah sehingga ketika kedua matriks segitiga terssebut dikalikan maka akan menghasilkan A.



Karena AX = B maka LX = B. Jika X = Y maka LY = B. Metode dekomposisi Choleski dimulai dari membuat matriks segitiga atas dan matriks segitiga bawah. Matriks segitiga bawah merupakan transpose dari matriks segitiga atas. T = L pada persamaan di bawah ini.





Setelah mengetahui L, Langkah selanjutnya adalah mencari Y dari persamaan LY = B menggunakan teknik subtitusi. Setelah Y ditemukan, lalu dilanjutkan dengan mencari X dari persamaan X = Y. X dapat dicari dengan menggunakan teknik subtitusi mundur.

1. **Sistem Persamaan Linear**

Dalam *project* penelitian dapat dipastikan objek yang diteliti dimodelkan dalam bentuk matematika. Model matematika tersebut terdiri dari banyak persamaan yang harus diselesaikan secara simultan. Jika sistem persamaan yang dihasilkan berbentuk aljabar liniear, diperlukan suatu teknik penyelesaian, salah satunya dengan menggunakan matriks.

**BAB III**

**PEMBAHASAN**

1. **Penyelesaian dengan Metode Eliminasi Gauss**
2. ***Source Code***

## module gaussElimin

'''x = gaussElimin(a,b).

penyelesaian[a]{b}={x} dengan metode Eliminasi Gauss.

'''

import numpy as np

def gaussElimin(a,b):

n = len(b)

# Elimination Phase

for k in range(0,n-1):

for i in range(k+1,n):

if a[i,k] != 0.0:

lam = a [i,k]/a[k,k]

a[i,k+1:n] = a[i,k+1:n] - lam\*a[k,k+1:n]

b[i] = b[i] - lam\*b[k]

# Back substitution

for k in range(n-1,-1,-1):

b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]

return b

## Contoh Metode eliminasi Gauss

a = np.array([[4.0, 1.0, 2.0], [1.0, 3.0, 1.0], [1.0, 2.0, 5.0]])

b = np.array([[16.0],[10.0],[12.0]])

aOrig = a.copy() # Save original matrix

bOrig = b.copy() # and the constant vector

x = gaussElimin(a,b)

det = np.prod(np.diagonal(a))

print('a = ',aOrig)

print('b = ',bOrig)

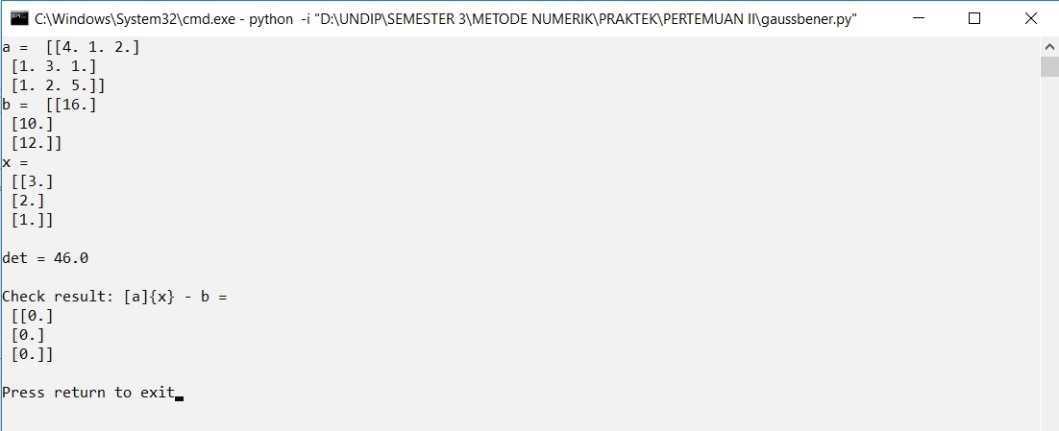
print('x =\n',x)

print('\ndet =',det)

print('\nCheck result: [a]{x} - b =\n',np.dot(aOrig,x) - bOrig)

input('\nPress return to exit')

1. ***Screenshot* Hasil atau Keluaran**



1. **Penjelasan**

Program di atas menampilkan solusi dari sistem persamaan linier dengan metode eliminasi gauss. Fungsi gaussElimin() akan mengembalikan nilai dari matriks x yang merupakan solusi dari sistem persamaan linier yang dicari.

for k in range(0,n-1):

for i in range(k+1,n):

if a[i,k] != 0.0:

lam = a [i,k]/a[k,k]

a[i,k+1:n] = a[i,k+1:n] - lam\*a[k,k+1:n]

b[i] = b[i] - lam\*b[k]

Baris kode di atas akan mengubah matriks yang sudah ada menjadi matriks segitiga atas. Matriks segitiga atas dibutuhkan untuk selanjutnya dilakukan proses substitusi balik.

for k in range(n-1,-1,-1):

b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]

return b

Baris kode di atas akan melakukan proses *back substitution* di mana hal tersebut dilakukan mulai dari baris yang paling bawah dan seterusnya ke atas.

Program utamanya akan memasukkan nilai dari matriks a dan b yang akan dicari solusi persamaan liniernya. Lalu setelahnya akan mencetak ke layar matriks a, b, dan x yang merupakan solusi dari sistem persamaan linier tersebut, juga mencetak determinan dari matriks a. Selain itu, program utamanya juga akan melakukan cek apakah jawaban yang ditampilkan sudah benar dengan melakukan operasi pada persamaan . Apabila hasilnya 0, maka solusi yang ditampilkan pada layar sudah tepat.

1. **Penyelesaian dengan Metode Cholesky**
2. ***Source Code***

## module choleski

''' L = choleski(a)

Dekomposisi Choleski : [L][L]transpose = [a]

x = choleskiSol(L,b)

'''

import numpy as np

import math

def choleski(a):

n = len(a)

for k in range(n):

try:

a[k,k] = math.sqrt(a[k,k] \

- np.dot(a[k,0:k],a[k,0:k]))

except ValueError:

error.err('Matrix is not positive definite')

for i in range(k+1,n):

a[i,k] = (a[i,k] - np.dot(a[i,0:k],a[k,0:k]))/a[k,k]

for k in range(1,n): a[0:k,k] = 0.0

return a

## module error

''' err(string).

Prints ’string’ and terminates program

'''

import sys

def err(string):

print(string)

input('Press return to exit')

sys.exit()

## contoh dekomposisi choleski

a = np.array([[4.0, 1.0, 2.0], [1.0, 3.0, 1.0], [1.0, 2.0, 5.0]])

b = np.array([[16.0],[10.0],[12.0]])

aOrig = a.copy()

L = choleski(a)

x = choleskiSol(L,b)

print("x =",x)

print('\nCheck: A\*x =\n',np.dot(aOrig,x))

input("\nPress return to exit")

x = choleskiSol(L,b)

'''

import numpy as np

import math

def choleski(a):

n = len(a)

for k in range(n):

try:

a[k,k] = math.sqrt(a[k,k] \

- np.dot(a[k,0:k],a[k,0:k]))

except ValueError:

error.err('Matrix is not positive definite')

for i in range(k+1,n):

a[i,k] = (a[i,k] - np.dot(a[i,0:k],a[k,0:k]))/a[k,k]

for k in range(1,n): a[0:k,k] = 0.0

return a

## module choleskiSol

''' L = choleski(a)

Dekomposisi Choleski : [L][L]transpose = [a]

x = choleskiSol(L,b)

'''

def choleskiSol(L,b):

n = len(b)

# Solution of [L]{y} = {b}

for k in range(n):

b[k] = (b[k] - np.dot(L[k,0:k],b[0:k]))/L[k,k]

# Solution of [L\_transpose]{x} = {y}

for k in range(n-1,-1,-1):

b[k] = (b[k] - np.dot(L[k+1:n,k],b[k+1:n]))/L[k,k]

return b

## module error

''' err(string).

Prints ’string’ and terminates program

'''

import sys

def err(string):

print(string)

input('Press return to exit')

sys.exit()

## contoh dekomposisi choleski

a = np.array([[4.0, 1.0, 2.0], [1.0, 3.0, 1.0], [1.0, 2.0, 5.0]])

b = np.array([[16.0],[10.0],[12.0]])

aOrig = a.copy()

L = choleski(a)

x = choleskiSol(L,b)

print("x =",x)

print('\nCheck: A\*x =\n',np.dot(aOrig,x))

input("\nPress return to exit")

''' err(string).

Prints ’string’ and terminates program

'''

import sys

def err(string):

print(string)

input('Press return to exit')

sys.exit()

## contoh dekomposisi choleski

a = np.array([[4.0, 1.0, 2.0], [1.0, 3.0, 1.0], [1.0, 2.0, 5.0]])

b = np.array([[16.0],[10.0],[12.0]])

aOrig = a.copy()

L = choleski(a)

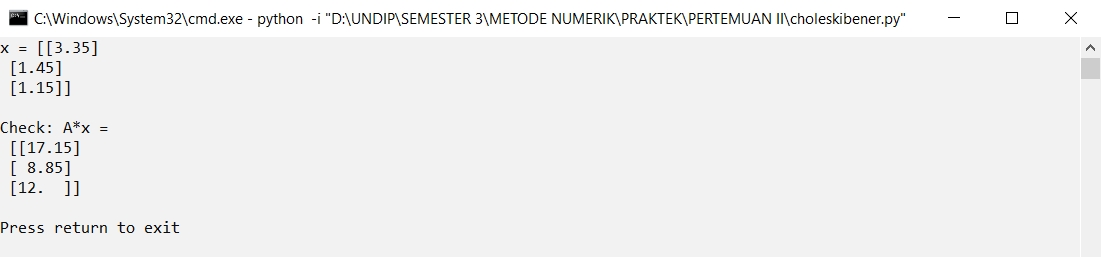
x = choleskiSol(L,b)

print("x =",x)

print('\nCheck: A\*x =\n',np.dot(aOrig,x))

input("\nPress return to exit")

**3.2.2 *Screenshot* Hasil atau Keluaran**



1. **Penjelasan**

Program di atas akan menampilkan solusi dari sistem persamaan linier dengan metode Cholesky. Fungsi Cholesky() akan membentuk matris L yang merupakan hasil perkalian dari matriks lower dan transposenya. Fungsi CholeskySol() akan mencari matriks dari persamaan yang kemudian akan digunakan untuk mencari matriks dari persamaan . Fungsi choleskySol() akan mengembalikan nilai dari matriks x yang merupakan solusi dari sistem persamaan linier yang dicari.

Program utamanya akan mencetak matriks x dan juga melakukan cek apakah jawaban yang dihasilkan sudah benar dengan melakukan perkalian *dot product* pada matriks A dan x. Apabila hasilnya mendekati matriks b yang dimasukkan di awal maka solusi tersebut benar.

**BAB IV**

**PENUTUP**

1. **Kesimpulan**

Berdasarkan hasil yang diperoleh melalui program Eliminasi Gauss dan Cholesky, solusi sistem persamaan linier jika mariks dan   
, yaitu:

1. **Metode Eliminasi Gauss**
2. **Metode Cholesky**

**DAFTAR PUSTAKA**

Munir, Rinaldi. 2015. *Metode Numerik.* Bandung : Informatika

Sasongko, Priyo Sidik dan Suhartono. 2019. Modul Praktikum Metode Numerik. Semarang: Departemen Informatika/Ilmu Komputer Fakultas Sains dan Matematika Universitas Diponegoro