**BAB I**

**PENDAHULUAN**

* 1. **Tujuan**
* Dapat menentukan penyelesaian sistem persamaan linear dengan metode Gauss, metode Decomposisi Choleski
* Mencari besarnya kesalahan dari suatu perhitungan solusi sistem persamaan linear dengan metode Gauss, dan metode Decomposisi Choleski
  1. **Rumusan Masalah**
* Bagaimana cara menentukan penyelesaian sistem persamaan linear dengan metode Gauss dan metode Decomposisi Choleski?
* Bagaimana cara mencari atau menentukan besarnya kesalahan dari suatu perhitungan solusi sistem persamaan linear dengan metode Gauss dan metode Decomposisi choleski?

**BAB II**

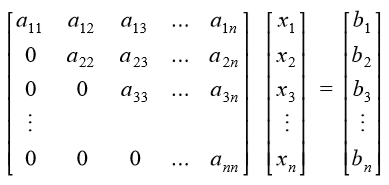
**DASAR TEORI**

1. **Solusi Sistem Persamaan Linear**

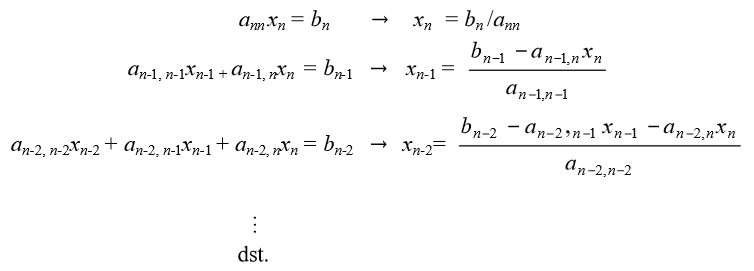
Dalam suatu penelitian atau praktik rekayasa, biasanya objek yang diteliti akan dimodelkan dalam bentuk matematika. Sering kali model matematika tersebut terdiri dari banyak persamaan yang harus diselesaikan secara serempak atau simultan. Jika sistem persamaan yang dihasilkan berbentuk aljabar linier, maka diperlukan suatu teknik penyelesaian, salah satunya adalah menggunakan matriks.

1. **Metode Gauss**

Metode ini memanfaatkan fakta bahwa perkalian matriks AX = B dengan A adalah matriks segitiga, X adalah matriks yang memiliki n baris 1 kolom, dan B adalah matriks yang memiliki n baris 1 kolom dapat dicari elemen X-nya dengan subtitusi. Hal ini bisa dilakukan karena baris ke-n matriks A hanya memiliki satu elemen tidak nol sehingga elemen dari X pada baris ke-n dapat dicari nilainya.. Langkah pertama metode Gauss adalah mereduksi matriks sehingga menjadi matriks segitiga. Langkah kedua adalah melakukan subtitusi mundur.



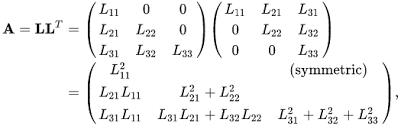
AX = B dengan A adalah matriks segitiga



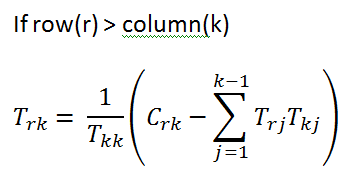
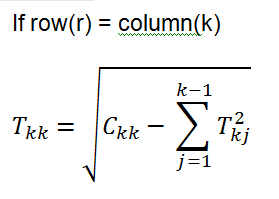
Teknik subtitusi mundur

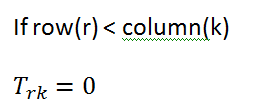
1. **Metode Dekomposisi Choleski**

Misalkan terdapat persamaan matriks AX = B dengan X adalah matriks yang dicari. Langkah pertama dari metode dekomposisi 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.

**BAB III**

**PEMBAHASAN**

1. **Metode Gauss**

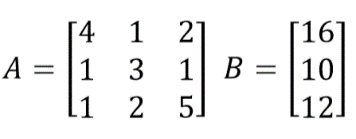
Berikut ini merupakan Implementasi metode Gauss dalam bahasa pemrograman python dengan menggunakan modul numpy

gaussElimin.py

|  |
| --- |
| ## 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 |

* Pertama-tama fungsi gaussElimin mencari tahu panjangnya matriks b sehingga fungsi memiliki batasan pada loop eliminasi dan loop subtitusi.
* Pada loop eliminasi, fungsi melakukan reduksi matriks a sehingga matriks a menjadi matriks segitiga. Pada awal loop, pertama fungsi membaca apakah elemen di bawah pivot sudah 0, jika belum maka akan diubah menjadi 0. Elemen akan diubah menjadi 0 menggunakan operasi baris elementer. Fungsi akan mencari tahu ratio elemen yang akan direduksi dengan elemen pivot, kemudian nilai ratio tersebut dimasukkan ke dalam variabel lam. Setelah variabel lam diketahui, maka fungsi akan mereduksi matriks A dan B pada baris ke-i. Setelah semua proses tersebut selesai, i mengalami increment. Hal ini menandakan bahwa fungsi akan memproses baris berikutnya. Setelah baris habis j akan mengalami increment. Hal ini menandakan fungsi akan memproses kolom selanjutnya.
* Pada loop subtitusi, nilai x dikalkulasi, kemudian dimasukkan ke varibel matriks b. Implementasi ini menerapkan teknik subtitusi mundur yang sudah dijelaskan pada dasar teori.

Berikut ini merupakan contoh dari penggunaan modul gaussElimin di atas pada bahasa pemrograman python dimana :

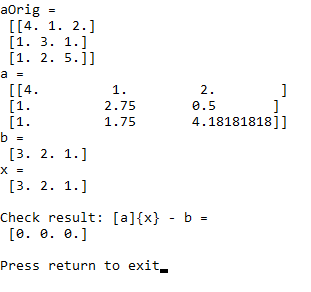


mgaussElimin.py

|  |
| --- |
| ## Contoh Metode eliminasi Gauss  **import** numpy **as** np  **from** gaussElimin **import** **\***  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  **print(**'aOrig = \n'**,** aOrig**)**  x **=** gaussElimin**(**a**,**b**)**  **print(**'a = \n'**,**a**)**  **print(**'b = \n'**,**b**)**  **print(**'x =\n'**,**x**)**  **print(**'\nCheck result: [a]{x} - b =\n'**,**np**.**dot**(**aOrig**,**x**)** **-** bOrig**)**  input**(**'\nPress return to exit'**)** |

* Pertama, program impor modul numpy dan modul gaussElimin yang sudah dibuat di atas agar fungsi dapat digunakan pada program.
* Selanjutnya, program mendeklarasikan nilai dari matriks a dan b. Kemudian program menyalin data dari matriks a dan b dan dimasukkan ke variabel aOrig dan bOrig. Hal ini dilakukan kerena matriks a dan b akan berubah bentuk jika diproses dengan fungsi gaussElimin.
* Kemudian program akan memanggil fungsi gaussElimin untuk mengkalkulasi x dari persamaan ax = b. Setelah itu, program memberi output a, b, x, dan hasil pengecekan ax = b. Matriks a dan b digantikan oleh aOrig dan bOrig karena nilai a dan b sudah berubah.

Hasil output dari program tersebut adalah sebagai berikut



* Matriks paling atas adalah matriks aOrig, Matriks kedua adalah matriks a yang sudah tereduksi. Hasil pengecekan menunjukkan error dari metode Gauss. Dalam kasus ini tidak ada error.

1. **Metode Dekomposisi Choleski**

Berikut ini merupakan Implementasi metode dekomposisi Choleski dalam bahasa pemrograman python dengan menggunakan modul numpy.

choleski.py

|  |
| --- |
| ## module choleski  ''' L = choleski(a)  Dekomposisi Choleski : [L][L]transpose = [a]  x = choleskiSol(L,b)  '''  **import** numpy **as** np  **import** math  **import** error  **def** choleski**(**a**):**  n **=** len**(**a**)**  #Dekomposisi Choleski  **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**]**  **#Matriks segitiga**  **for** k **in** range**(**1**,**n**):** a**[**0**:**k**,**k**]** **=** 0.0  **return** a |

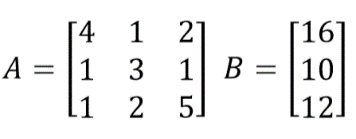
* Modul choleski digunakan untuk mencari dekomposisi choleski dari matriks a dengan menggunakan teknik yang sudah dijelaskan di dasar teori. Jika elemen yang diproses merupakan elemen diagonal utama maka akan dikalkulasi menggunakan persamaan 2.3.1. Jika elemen yang diproses bukan merupakan elemen diagonal utama, maka akan dikalkulasi menggunakan persamaan 2.3.2
* Setelah loop dekomposisi choleski selesai, matriks a diubah menjadi matriks segitiga atas dengan membuat elemen di atas diagonal utama menjadi 0.0.

choleskiSol.py

|  |
| --- |
| ## module choleskiSol  ''' L = choleski(a)  Dekomposisi Choleski : [L][L]transpose = [a]  x = choleskiSol(L,b)  '''  **import** numpy **as** np  **import** math  **import** error  **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 |

* Modul choleskiSol digunakan untuk mencari menyelesaikan persamaan LY = B dan X = Y. Pertama-tama fungsi choleskiSol mencari nilai Y terlebih dahulu menggunakan teknik subtitusi. Kemudian hasilnya dimasukkan ke matriks b.
* Setelah Y diketahui, fungsi choleskiSol mencari nilai X dari persamaan X = Y dengan menggunakan teknik subtitusi mundur. Nilai X dimasukkan ke matriks b.

Berikut ini merupakan contoh dari penggunaan modul gaussElimin di atas pada bahasa pemrograman python dimana

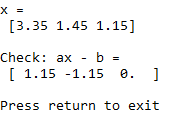


mCholeski.py

|  |
| --- |
| ## contoh dekomposisi choleski  **import** numpy **as** np  **from** choleski **import** **\***  **from** choleskiSol **import** **\***  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**()**  bOrig **=** b**.**copy**()**  L **=** choleski**(**a**)**  x **=** choleskiSol**(**L**,**b**)**  **print(**"x =\n"**,**x**)**  **print(**'\nCheck: ax - b =\n'**,**np**.**dot**(**aOrig**,**x**)** **-** bOrig**)**  input**(**"\nPress return to exit"**)** |

* Pertama, program mengimpor modul numpy, modul choleski, dan modul choleskiSol. Kemudian program mendeklarasikan a dan b sesuai dengan soal.
* Matriks a dan b kemudian disalin ke matriks aOrig dan bOrig karena matriks a dan b akan berubah ketika diproses oleh fungsi choleski dan choleskiSol.
* Terakhir, program mengeluarkan output matriks yang dicari yaitu matriks x dan errornya.

Hasil output dari program tersebut adalah sebagai berikut



* x adalah hasil dari matriks yang dicari dari persamaan ax = b, matriks paling bawah adalah hasil pengecekan error. Dalam kasus ini errornya besar.

**BAB IV**

**PENUTUP**

**4.1 Kesimpulan**

Berikut adalah kesimpulan dari praktikum yang sudah dilakukan.

* Mencari solusi dari banyak persamaan linear merupakan permasalahan yang dihadapi oleh para ilmuan dan ahli rekayasa. Sering kali, persamaan yang akan dicari solusinya merupakan persamaan yang rumit sehingga sulit untuk ditemukan solusinya. Oleh karena itu, muncul metode-metode untuk mencari akar seperti metode Gauss dan metode dekomposisi choleski.
* Namun penggunaan metode-metode tersebut untuk mencari akar sulit jika menggunakan perhitungan secara manual oleh manusia. Oleh karena itu, manusia menggunakan komputer untuk mempermudah perhitungan dengan metode-metode tersebut.

**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