**TUGAS MANDIRI**

**PERANCANGAN & ANALISIS ALGORITMA**

**“Gauss Elimination”**

**202323430048**

****

**DOSEN PENGAMPU:**

**Randi Proska Sandra, S.Pd, M.Sc**

**OLEH:**

**Billy Kelvin Tressa**

**22343039**

**Informatika (NK)**

**PRODI SARJANA INFORMATIKA**

**DEPARTEMEN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2024**

1. Penjelasan Singkat

Dalam matematika, **eliminasi Gauss** adalah [algoritma](https://p2k.stekom.ac.id/ensiklopedia/Algoritme" \o "Algoritme) yang digunakan untuk menyelesaikan [sistem persamaan linear](https://p2k.stekom.ac.id/ensiklopedia/Sistem_persamaan_linear" \o "Sistem persamaan linear). Algoritma ini terdiri dari serangkaian operasi yang dilakukan pada matriks koefisien dari sistem persamaan tersebut. Walau akan mengubah bentuk matriks, operasi-operasi tersebut tidak akan mengubah solusi dari sistem persamaan. Hal ini memungkinkan matriks koefisien dibentuk menjadi sebuah [matriks segitiga atas](https://p2k.stekom.ac.id/ensiklopedia/Matriks_segitiga_atas" \o "Matriks segitiga atas), sehingga solusi sistem persamaan dapat ditentukan dengan cukup melakukan eliminasi variabel secara berulang. Eliminasi Gauss juga dapat digunakan untuk menghitung [rank](https://p2k.stekom.ac.id/ensiklopedia/Rank_(aljabar_linear)) dari matriks, [determinan](https://p2k.stekom.ac.id/ensiklopedia/Determinan" \o "Determinan) dari matriks persegi, dan invers dari [matriks nonsingular](https://p2k.stekom.ac.id/ensiklopedia/Matriks_terbalikkan" \o "Matriks terbalikkan). Metode ini dinamai dari matematikawan [Carl Friedrich Gauss](https://p2k.stekom.ac.id/ensiklopedia/Carl_Friedrich_Gauss) (1777–1855), walaupun beberapa kasus khusus dari metode ini — tapi tanpa dilengkapi bukti — sudah dikenal oleh matematikawan Tionghoa semenjak tahun 179 M.

Matriks segitiga atas yang didapat dari algoritma ini akan memiliki bentuk eselon baris (*row echelon form*). Jika semua koefisien utama (nilai bukan nol pertama pada sebuah baris) matriks bernilai 1, dan kolom-kolom yang mengandung koefisien utama memiliki bentuk yang sama dengan kolom pada matriks identitas, matriks tersebut dikatakan memiliki bentuk eselon baris tereduksi (*reduced row echelon form*). Eliminasi Gauss yang dilakukan untuk mengubah matriks koefisien *sampai* menjadi bentuk eselon baris tereduksi terkadang disebut sebagai **eliminasi Gauss–Jordan**. Karena alasan komputasi, operasi baris untuk mencari solusi sistem persamaan terkadang dihentikan sebelum matriks berada dalam bentuk tereduksinya.

Kompleksitas komputasi eliminasi Gauss untuk sebuah matriks berukuran adalah . Dalam bentuk paling sederhana, secara numerik algoritma ini rentan terhadap galat pembulatan. Namun hal ini dapat diatasi dengan menggunakan metode pivot; menjadikannya cara standar menemukan solusi sistem persamaan linear, dan menjadi bagian pustaka program [aljabar linear](https://p2k.stekom.ac.id/ensiklopedia/Aljabar_linear" \o "Aljabar linear) yang penting seperti NAG, IMSL, dan LAPACK.[1]

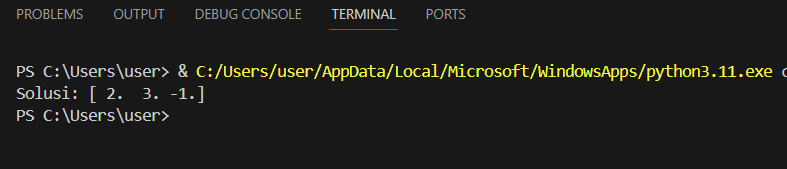
1. Pseudocode

|  |
| --- |
| Procedure eliminasiGauss(A, b):  n = jumlah persamaan (jumlah baris) dalam matriks A    for k dari 1 sampai n-1 lakukan:  # Pivoting (opsional namun direkomendasikan untuk stabilitas numerik)  indeks\_maks = k  nilai\_maks = abs(A[k][k])  for i dari k+1 sampai n lakukan:  jika abs(A[i][k]) > nilai\_maks maka:  indeks\_maks = i  nilai\_maks = abs(A[i][k])  Tukar baris A[k] dan A[indeks\_maks]  Tukar elemen b[k] dan b[indeks\_maks]    for i dari k+1 sampai n lakukan:  faktor = A[i][k] / A[k][k]  for j dari k sampai n lakukan:  A[i][j] = A[i][j] - faktor \* A[k][j]  b[i] = b[i] - faktor \* b[k]    # Substitusi mundur  x[n] = b[n] / A[n][n]  for i dari n-1 sampai 1 lakukan:  jumlah = 0  for j dari i+1 sampai n lakukan:  jumlah = jumlah + A[i][j] \* x[j]  x[i] = (b[i] - jumlah) / A[i][i]    return x |

1. Source Code

|  |
| --- |
| import numpy as np  def eliminasi\_gauss(A, b):  n = len(A)  # Menggabungkan matriks A dan vektor b  Ab = np.column\_stack((A, b))  # Eliminasi maju (forward elimination)  for k in range(n-1):  # Pivoting parsial  indeks\_max = np.argmax(np.abs(Ab[k:, k])) + k  Ab[[k, indeks\_max]] = Ab[[indeks\_max, k]]  for i in range(k+1, n):  faktor = Ab[i, k] / Ab[k, k]  Ab[i, k:] -= faktor \* Ab[k, k:]  # Substitusi mundur (back substitution)  x = np.zeros(n)  for i in range(n-1, -1, -1):  x[i] = (Ab[i, -1] - np.dot(Ab[i, :-1], x)) / Ab[i, i]  return x  # Contoh penggunaan  A = np.array([[2, 1, -1],  [-3, -1, 2],  [-2, 1, 2]], dtype=float)  b = np.array([8, -11, -3], dtype=float)  solusi = eliminasi\_gauss(A, b)  print("Solusi:", solusi) |

Hasil Screenshot :



1. Analisis Kebutuhan Waktu

Untuk analisis kompleksitas waktu algoritma eliminasi Gauss, kita dapat menggunakan tiga pendekatan:

1. Analisis berbasis operasi/instruksi:

Untuk setiap iterasi eliminasi maju, terdapat perulangan untuk menghitung faktor eliminasi dan mengurangi elemen-elemen matriks, dengan kompleksitas sekitar O(n^2).

Substitusi mundur memiliki kompleksitas O(n^2).

Sehingga, total kompleksitas waktu adalah O(n^2).

1. Analisis berbasis jumlah operasi abstrak:

Dalam eliminasi maju, terdapat sekitar (n^3)/3 operasi abstrak (aproximasi kasar).

Substitusi mundur juga membutuhkan sekitar (n^2)/2 operasi abstrak.

Jadi, kompleksitas totalnya adalah sekitar (n^3)/3 + (n^2)/2.

1. Analisis menggunakan pendekatan best-case, worst-case, dan average-case:

Best-case: Jika matriks A sudah dalam bentuk segitiga atas, maka tidak diperlukan eliminasi maju. Kompleksitasnya adalah O(n^2).

Worst-case: Jika matriks A membutuhkan eliminasi penuh, kompleksitasnya adalah O(n^3).

Average-case: Kompleksitas rata-ratanya juga sekitar O(n^3), mengingat eliminasi memerlukan perulangan di setiap baris dan kolom.

1. Referensi

<https://p2k.stekom.ac.id/ensiklopedia/Eliminasi_Gauss>

[1] “Introduction to the Design and Analysis of Algorithms (3rd ed.) [Levitin 2011-10-09]”.

1. Link Github

<https://github.com/billy098/GaussElimination>