**Penjelasan Source Code Dekomposisi LU Gauss Pada Python**

Nama: Muhammad Nio Hastungkoro

NIM: 2112012214015

Matkul: Metode Numerik (A)

Source Code:

|  |
| --- |
| import numpy as np  def LU\_decomposition(A):  n = len(A)  L = np.eye(n)  U = A.astype(float) # Mengonversi tipe data matriks A ke float  for k in range(n-1):  for i in range(k+1, n):  if U[k, k] == 0:  return "Pembagian dengan nol terjadi, tidak dapat dilanjutkan"  factor = U[i, k] / U[k, k]  L[i, k] = factor  U[i, k:] -= factor \* U[k, k:]  return L, U  def solve\_linear\_eq\_LU\_gauss(A, b):  L, U = LU\_decomposition(A)  y = np.linalg.solve(L, b)  x = np.linalg.solve(U, y)  return x  # Testing  A = np.array([[2, 3], [4, 5]], dtype=float) # Mengonversi tipe data matriks A ke float  b = np.array([6, 7], dtype=float) # Mengonversi tipe data vektor b ke float  print("Solusi dengan metode dekomposisi LU Gauss:", solve\_linear\_eq\_LU\_gauss(A, b)) |

Penjelasan:

* import numpy as np

Baris pertama mengimpor pustaka NumPy dengan alias np. NumPy adalah pustaka Python yang populer untuk komputasi numerik, terutama dalam pemrosesan matriks dan vektor.

* def LU\_decomposition(A):

Ini adalah deklarasi fungsi LU\_decomposition yang menerima sebuah parameter, yaitu matriks 𝐴A (diwakili oleh A). Fungsi ini bertujuan untuk melakukan dekomposisi LU pada matriks 𝐴A.

* n = len(A)

Baris ini menghitung ukuran matriks A dan menyimpannya dalam variabel n.

* L = np.eye(n)

Baris ini membuat matriks identitas L dengan ukuran 𝑛×𝑛n×n menggunakan fungsi np.eye() dari NumPy. Matriks identitas ini akan digunakan sebagai matriks L dalam dekomposisi LU.

* U = A.astype(float):

Baris ini mengkonversi tipe data matriks A ke float menggunakan metode .astype(float). Hal ini diperlukan karena beberapa operasi dalam dekomposisi LU membutuhkan tipe data float.

* for k in range(n-1):

Ini adalah loop yang berjalan dari 0 hingga 𝑛−2n−2. Ini mengiterasi melalui setiap kolom matriks A, kecuali kolom terakhir, karena proses dekomposisi LU hanya berlaku untuk matriks persegi.

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

Ini adalah loop dalam loop luar yang berjalan dari 𝑘+1k+1 hingga 𝑛−1n−1. Ini mengiterasi melalui setiap baris di bawah diagonal utama untuk setiap kolom k.

* if U[k, k] == 0:

Baris ini memeriksa apakah elemen diagonal utama matriks U pada baris dan kolom ke k sama dengan nol. Jika ya, maka dekomposisi tidak dapat dilanjutkan karena akan terjadi pembagian dengan nol.

* factor = U[i, k] / U[k, k]

Baris ini menghitung faktor yang akan digunakan untuk mengeliminasi elemen di bawah diagonal utama. Faktor ini adalah rasio antara elemen pada baris saat ini dan elemen pada diagonal utama.

* L[i, k] = factor

Baris ini mengatur elemen matriks L pada baris 𝑖i dan kolom 𝑘k menjadi nilai faktor yang telah dihitung.

* U[i, k:] -= factor \* U[k, k:]

**B**aris ini melakukan operasi pengurangan pada bagian matriks U di bawah diagonal utama menggunakan faktor yang telah dihitung.

* return L, U

Fungsi mengembalikan matriks L dan matriks U.

* def solve\_linear\_eq\_LU\_gauss(A, b):

Ini adalah deklarasi fungsi solve\_linear\_eq\_LU\_gauss yang menerima dua parameter, yaitu matriks A dan vektor hasil b. Fungsi ini bertujuan untuk menyelesaikan sistem persamaan linear 𝐴𝑥=𝑏Ax=b dengan menggunakan metode dekomposisi LU Gauss.

* L, U = LU\_decomposition(A)

Baris ini memanggil fungsi LU\_decomposition untuk mendapatkan matriks L dan matriks U dari matriks A.

* y = np.linalg.solve(L, b)

Baris ini menggunakan fungsi np.linalg.solve() untuk menyelesaikan sistem 𝐿𝑦=𝑏Ly=b dengan matriks L dan vektor b. Hasilnya disimpan dalam variabel y.

* x = np.linalg.solve(U, y)

Baris ini menggunakan fungsi np.linalg.solve() untuk menyelesaikan sistem 𝑈𝑥=𝑦Ux=y dengan matriks U dan vektor y. Hasilnya adalah vektor solusi x, yang disimpan dalam variabel x.

* return x

Mengembalikan vektor solusi x.

* A = np.array([[2, 3], [4, 5]], dtype=float)

Inisialisasi matriks koefisien 𝐴A dengan nilai [[2, 3], [4, 5]] dan mengkonversi tipe datanya ke float.

* b = np.array([6, 7], dtype=float)

Inisialisasi vektor hasil 𝑏b dengan nilai [6, 7] dan mengkonversi tipe datanya ke float.

* print("Solusi dengan metode dekomposisi LU Gauss:", solve\_linear\_eq\_LU\_gauss(A, b))

Baris ini memanggil fungsi solve\_linear\_eq\_LU\_gauss dengan matriks koefisien A dan vektor hasil b sebagai argumen, kemudian mencetak solusi yang diperoleh menggunakan metode dekomposisi LU Gauss.