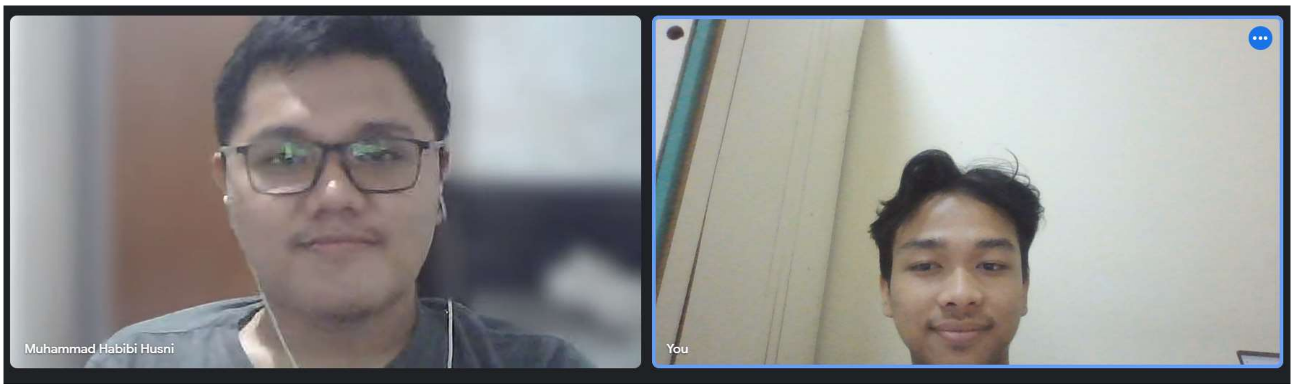
LAPORAN TUGAS BESAR 1 IF 2123 ALJABAR LINIER DAN GEOMETRI SISTEM PERSAMAAN LINIER, DETERMINAN, DAN APLIKASINYA

Link Repository: https://github.com/habibibi/Algeo01-21164



Kelompok 54

Akhmad Setiawan - 13521164

Muhammad Habibi Husni - 13521169

SEKOLAH TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

Semester 1 Tahun 2022/2023

BAB 1
DESKRIPSI MASALAH

Sistem persamaan linier (SPL) banyak ditemukan di dalam bidang sains dan rekayasa. Anda sudah mempelajari berbagai metode untuk menyelesaikan SPL, termasuk menghitung determinan matriks. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan ($x = A^{-1}b$), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak (tidak berhingga), atau hanya satu (unik/tunggal).

Di dalam Tugas Besar 1 ini, diminta untuk membuat satu atau lebih library aljabar linier dalam Bahasa Java. Library tersebut berisi fungsi-fungsi seperti eliminasi Gauss, eliminasi Gauss-Jordan, menentukan balikan matriks, menghitung determinan, kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan). Selanjutnya, gunakan library tersebut di dalam program Java untuk menyelesaikan berbagai persoalan yang dimodelkan dalam bentuk SPL, menyelesaikan persoalan interpolasi, dan persoalan regresi.

BAB 2
TEORI SINGKAT

Eliminasi Gauss

Eliminasi gauss ditemukan oleh Carl Friedrich Gauss, metode ini dapat dimanfaatkan untuk memecahkan sistem persamaan linear dengan merepresentasikan (mengubah) menjadi bentuk matriks, matriks tersebut lalu diubah menjadi Eselon Baris melalui Operasi Baris Elementer. Kemudian sistem diselesaikan dengan substitusi balik.

Untuk membentuk matriks Eselon Baris dapat menggunakan metode OBE. OBE adalah metode pengoperasian setiap baris matriks dengan pembagian satu baris dan pengurangan baris lainnya. Kedua operasi ini dilakukan secara terus menerus pada setiap barisnya sampai mendapatkan matriks Eselon Baris.

Eliminasi Gauss dapat dimanfaatkan untuk mendapatkan solusi dari Sistem Persamaan Linear. Ketika mendapatkan matriks Eselon Baris perlu dilakukan substitusi mundur untuk mendapatkan solusi tiap variabelnya.

Eliminasi Gauss-Jordan

Eliminasi Gauss-Jordan adalah pengembangan dari metode Eliminasi Gauss. Jika matriks yang dihasilkan Eliminasi Gauss adalah matriks Eselon Baris, Eliminasi Gauss-Jordan menghasilkan matriks Eselon Baris Tereduksi. Matriks Eselon Baris Tereduksi adalah matriks Eselon Baris tapi setiap elemen yang berada pada kolom 1 utama harus bernilai 0 semua kecuali 1 utama tersebut. Sehingga jika Eliminasi Gauss-Jordan diterapkan pada matriks yang jumlah baris dan kolomnya sama, akan menghasilkan matriks identitas.

Eliminasi Gauss-Jordan melalui proses OBE yang sama pada Eliminasi Gauss, perbedaannya ketika mendapatkan matriks Eselon Baris, dilakukan proses OBE kembali untuk setiap elemen diatas 1 utama. Metode eliminasi ini juga dapat dimanfaatkan untuk mendapatkan solusi SPL. Metode ini tidak memerlukan substitusi mundur.

Determinan Matriks

Determinan matriks adalah nilai skalar dari suatu matriks persegi. Menurut definisi ini, maka matriks yang ukurannya bukan $n \times n$ tidak memiliki determinan. Dalam laporan ini, akan dibahas 2 cara menghitung determinan matriks yaitu dengan ekspansi laplace atau ekspansi kofaktor dan dengan OBE. Dengan Metode OBE, determinan dapat dihitung dengan mengubah elemen-elemen di bawah diagonal utama menjadi 0 dan menghitung perkalian semua elemen diagonal utama. Dengan metode ekspansi kofaktor, determinan dapat dihitung dengan menghitung jumlah perkalian tiap elemen dalam suatu baris atau kolom dengan determinan matriks kofaktornya. Determinan memiliki banyak fungsi, contohnya adalah untuk menghitung matriks balikan dan menghitung solusi SPL dengan Kaidah Cramer.

Matriks Balikan

Matriks balikan atau invers matriks adalah suatu matriks yang jika dikalikan dengan matriks asalnya akan menghasilkan matriks identitas sama seperti pada aljabar biasa di mana bilangan X memiliki invers $X^{-1} = 1/X$, sehingga $XX^{-1} = 1$. Untuk mendapatkan matriks invers ini, dibahasdengan dua cara yaitu dengan rumus M^{-1} = (1/$determinan(M))adjoin(M)$ dengan adjoin merupakan transpose dari matriks Kofaktor M atau dengan eliminasi Gauss. Dapat dilihat dari rumusnya yang dibagi dengan determinan matriks maka matriks hanya memiliki matriks balikan jika determinan dari matriks tersebut tidak sama dengan nol dan merupakan matriks persegi. Metode eliminasi Gauss digunakan untuk menemukan matriks balikan dengan mengubah matriks tersebut menjadi matriks identitas dan melakukan perubahan yang sama terhadap matriks identitas sehingga matriks identitas tersebut berubah menjadi matriks balikan.

Matriks balikan ini dapat digunakan untuk menyelesaikan suatu sistem persamaan linier. Namun, metode matriks balikan ini hanya dapat digunakan jika SPL dapat diubah menjadi matriks persegi dan memiliki determinan yang tidak 0. Misalkan dimiliki SPL yang diubah menjadi bentuk matriks $AX = B$. Maka jika kedua ruas dikalikan dengan matriks balikan A didapat $A^{-1}AX = A^{-1}B$ yang menjadi $X = A^{-1}B$ karena matriks A dikali dengan matriks balikannya akan menjadi matriks identitas yang jika dikali dengan matriks X hasilnya akan sama dengan matriks X. Dari ini didapat solusi SPL dengan menggunakan matriks balikan.

Matriks Kofaktor

Matriks kofaktor adalah suatu matriks yang terdiri dari kofaktor matriks. Nilai kofaktor sebuah elemen adalah nilai determinan dari elemen-elemen matriks yang tidak sebaris dan tidak sekolom dengan elemen tersebut. Matriks kofaktor ini dipakai dalam perhitungan determinan matriks melalui ekspansi Laplace atau ekspansi kofaktor. Ekspansi kofaktor ini adalah metode perhitungan determinan matriks dengan memilih suatu baris atau kolom dalam matriks, mengalikan tiap elemen dalam baris atau kolom tersebut dengan determinan kofaktornya, dan menjumlahkan setiap nilai yang didapat dari perkalian elemen dengan determinan kofaktornya.

Matriks Adjoin

Matriks adjoin adalah transpose dari sebuah matriks kofaktor. Matriks adjoin ini dipakai dalam perhitungan matriks balikan suatu matriks dengan rumus

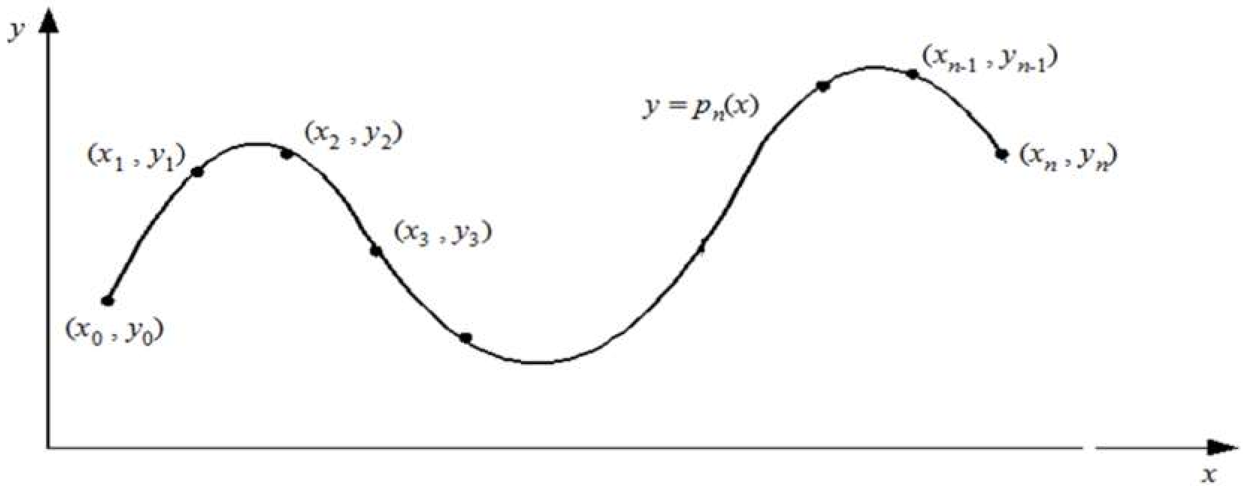
M^{-1} = (1determinan(M)) x adjoin(M).

Kaidah Cramer

Kaidah Cramer adalah suatu formula penyelesaian SPL yang dinamakan dari nama Gabriel Cramer yaitu orang yang mempublikasikan metode ini. Kaidah Cramer ini dapat menyelesaikan suatu SPL selama SPL itu memiliki solusi unik. Misal dimiliki SPL dalam bentuk matriks AX = B dengan ukuran matriks $A_{n\times n}$, matriks B, $X_{n\times 1}$. Maka menurut kaidah cramer nilai x_1 sama dengan determinan matriks A yang kolom 1-nya diganti dengan matriks B dibagi dengan determinan matriks A. Ini berlaku hingga x_n yang nilainya sama dengan determinan matriks A dengan kolom ke n-nya diganti dengan matriks B dibagi dengan determinan matriks A. Jika dituliskan dalam bentuk formula maka akan berbentuk $x_n = \det(A_n)/\det(A)$. Kaidah Cramer sebenarnya dapat diterapkan untuk matriks B, X ukuran N x M namun pada program ini diasumsikan input B selalu 1 kolom.

Interpolasi Polinom

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$. Tentukan polinom $p_n(x)$ yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga $y_i = p_n(x_i)$ untuk i = 0, 1, 2, ..., n.



Setelah polinom interpolasi $p_n(x)$ ditemukan, $p_n(x)$ dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang $[x_0, x_n]$.

Polinom interpolasi derajat n yang menginterpolasi titik-titik $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$. adalah berbentuk $p_n(x) = a_0 + a_1x + a_2x_2 + ... + a_nx_n$. Jika hanya ada dua titik, (x_0, y_0) dan (x_1, y_1), maka polinom yang menginterpolasi kedua titik tersebut adalah $p_1(x) = a_0 + a_1x$ yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x_0, y_0), (x_1, y_1), dan (x_2, y_2), maka polinom yang menginterpolasi ketiga titik tersebut adalah $p_2(x) = a_0 + a_1x + a_2x^2$ atau persaman kuadrat dan kurvanya berupa parabola.

Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data.

$$
\begin{aligned}
a_0 + a_1x_0 + a_2x_0^{\,2} + \ldots + a_n\, x_0^{\,n} &= y_0 \\
a_0 + a_1x_1 + a_2x_1^{\,2} + \ldots + a_n\, x_1^{\,n} &= y_1 \\
\ldots \qquad\qquad & \ldots \\
a_0 + a_1x_n + a_2x_n^{\,2} + \ldots + a_n\, x_n^{\,n} &= y_n
\end{aligned}
$$

Solusi sistem persamaan lanjar ini, yaitu nilai a_0, a_1, ..., a_n, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah dipelajari.

Bicubic Interpolation

Bicubic interpolation merupakan teknik interpolasi pada data 2D umumnya digunakan dalam pembesaran citra yang merupakan pengembangan dari interpolasi linear dan cubic yang telah dipelajari pada kuliah metode numerik di aljabar geometri. Diberikan sebuah matrix awal, misal M, kita akan mencari persamaan interpolasi f(x,y) dengan pemodelan sebagai berikut:

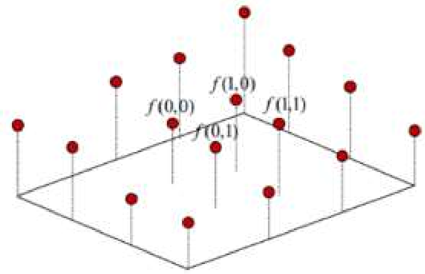
Normalization: $f(0,0), f(1,0)$

$f(0,1), f(1,1)$

Model: $$f(x,y) = \sum_{j=0}^{3}\sum_{i=0}^{3} a_{ij} x^i y^j$$

$$x = -1, 0, 1, 2$$

Solve: a_{ij}



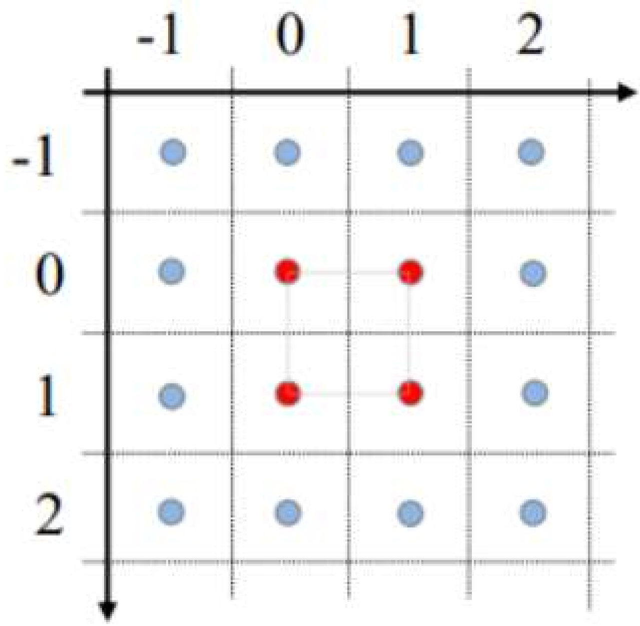
Melakukan substitusi nilai-nilai diketahui pada matriks 4 x 4 tersebut ke persamaan f(x,y) akan menghasilkan sebuah matriks persamaan:

$$y = Xa$$

$$\begin{bmatrix} f(-1,-1) \\ f(0,-1) \\ f(1,-1) \\ f(2,-1) \\ f(-1,0) \\ f(0,0) \\ f(1,0) \\ f(2,0) \\ f(-1,1) \\ f(0,1) \\ f(1,1) \\ f(2,1) \\ f(-1,2) \\ f(0,2) \\ f(1,2) \\ f(2,2) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 & 1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 & 1 & 2 & 4 & 8 \\ 1 & -1 & 1 & -1 & 2 & -2 & 2 & -2 & 4 & -4 & 4 & -4 & 8 & -8 & 8 & -8 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 1 & 2 & 4 & 8 & 2 & 4 & 8 & 16 & 4 & 8 & 16 & 32 & 8 & 16 & 32 & 64 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30} \\ a_{01} \\ a_{11} \\ a_{21} \\ a_{31} \\ a_{02} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{03} \\ a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

Elemen pada matrix X adalah koefisien a_{ij} yang diperoleh dari persamaan f(x,y) di atas. Sebagai contoh, elemen pada baris 4 kolom ke 10 adalah koefisien dari a_{12} dan diperoleh dari $2^1 * (-1)^2 = 2$, sesuai persamaan $x^i * y^j$.

Vektor a dapat dicari dari persamaan tersebut (menggunakan inverse), lalu vektor a digunakan sebagai nilai variabel dalam f(x,y). Sehingga terbentuk fungsi interpolasi bicubic sesuai model.



Regresi Linear

Regresi Linear merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \epsilon_i$$

Untuk mendapatkan nilai dari setiap βi dapat digunakan Normal Estimation Equation for Multiple Linear Regression sebagai berikut:

$$\begin{array}{lllllll}
nb_0 & + b_1 \sum_{i=1}^{n} x_{1i} & + b_2 \sum_{i=1}^{n} x_{2i} & + & \cdots & + b_k \sum_{i=1}^{n} x_{ki} & = \sum_{i=1}^{n} y_i \\
b_0 \sum_{i=1}^{n} x_{1i} & + b_1 \sum_{i=1}^{n} x_{1i}^2 & + b_2 \sum_{i=1}^{n} x_{1i} x_{2i} & + & \cdots & + b_k \sum_{i=1}^{n} x_{1i} x_{ki} & = \sum_{i=1}^{n} x_{1i} y_i \\
\vdots & \vdots & \vdots & & & \vdots & \vdots \\
b_0 \sum_{i=1}^{n} x_{ki} & + b_1 \sum_{i=1}^{n} x_{ki} x_{1i} & + b_2 \sum_{i=1}^{n} x_{ki} x_{2i} & + & \cdots & + b_k \sum_{i=1}^{n} x_{ki}^2 & = \sum_{i=1}^{n} x_{ki} y_i
\end{array}$$

Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

BAB 3
IMPLEMENTASI PROGRAM

Tabel Method Tiap Class

Class	Method	Tipe Data	Deskripsi
Determinan	detCofactorExpansion	Double	Mendapatkan determinan dengan cara kofaktor
	detReductionRow	Double	Mendapatkan determinan dengan reduksi baris
	rowReduction	Integer	Mendapatkan berapa banyak jumlah pertukaran baris
Invers	isExistInv	Boolean	Mendapatkan true jika matrix ada inversnya
	invGaussJordan	Matrix	Invers matriks dengan gauss jordan
	invAdjoint	Matrix	Invers matriks dengan matriks adjoint
Main	inputJenisMasukan	Integer	Kode input masukan user untuk mengakses program
	inputJenisKeluaran	Integer	Kode input masukan user untuk keluar
	inputNamaFileMasukan	String	Menerima file dari pengguna
	inputNamaFileKeluaran	String	Membaca nama file untuk output
SPL	isUniqueSol	Boolean	Memberi true jika solusi SPL unik
	isThereSol	Boolean	Mengecek apakah ada solusi SPL
	solGaussJordan	Array of double	Mencari solusi SPL dengan gauss jordan

	solInverse	Array of double	Mencari solusi SPL dengan invers
	solKaidahCramer	Array of double	Mencari solusi SPL dengan kaidah cramer
	solGauss	Array double	Mencari solusi SPL dengan metode Gauss

Garis Besar Program

Program dijalankan dengan menjalankan main.class. Setelah dijalankan, muncul tampilan menu. Pengguna dapat mengikuti panduan menu untuk menggunakan seluruh fitur program, antara lain: menyelesaikan SPL, mencari determinan matriks, mencari matriks balikan sebuah matriks, membuat polinom interpolasi, dan membuat persamaan regresi linier berganda.

Apabila pengguna ingin memberi masukan dari file, pengguna perlu membuat file masukan dengan format .txt, lalu memasukkannya ke dalam folder test. Setelah itu, pengguna dapat mengikuti panduan menu untuk memberi masukan dari file. Pada program terdapat fitur untuk menyimpan luaran program dalam file. Pengguna dapat mengikuti panduan menu untuk menggunakan fitur tersebut.

BAB 4
EKSPERIMEN

1. Temukan solusi SPL $Ax = b$,berikut:

   a.

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \\ 4 \\ 6 \end{bmatrix}$$

Output Program:

```
Masukkan banyak baris (m): 4
Masukkan banyak kolom (n): 4
Masukkan nilai Aij secara berurut :
1 1 -1 -1
2 5 -7 -5
2 -1 1 3
5 2 -4 2
Masukkan nilai Bi secara berurut :
1 -2 4 6

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
SPL tidak konsisten, tidak ada solusi!
Tekan ENTER untuk kembali ke menu utama
```

b.

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

Output Program:

```
Masukkan banyak baris (m): 4
Masukkan banyak kolom (n): 5
Masukkan nilai Aij secara berurut :
1 -1 0 0 1
1 1 0 -3 0
2 -1 0 1 -1
-1 2 0 -2 -1
Masukkan nilai Bi secara berurut :
3 6 5 -1

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x1 = -1.00a1 + 3.00
x2 = -2.00a1
x3 = a0
x4 = -1.00a1 - 1.00
x5 = a1
Tekan ENTER untuk kembali ke menu utama
```

c.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

Output Program:

```
Masukkan banyak baris (m): 3
Masukkan banyak kolom (n): 6
Masukkan nilai Aij secara berurut :
0 1 0 0 1 0
0 0 0 1 1 0
0 1 0 0 0 1
Masukkan nilai Bi secara berurut :
2 -1 1

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x1 = a0
x2 = 1.00a2 + 1.00
x3 = a1
x4 = 1.00a2 - 2.00
x5 = -1.00a2 + 1.00
x6 = a2
Tekan ENTER untuk kembali ke menu utama
```

d.

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

H adalah matriks Hilbert. Cobakan untuk n = 6 dan n = 10.

Output Program:

n = 6

```
Masukkan banyak baris (m): 6
Masukkan banyak kolom (n): 6
Masukkan nilai Aij secara berurut :
1.0000000000 0.5000000000 0.3333333333 0.2500000000 0.2000000000 0.1666666667
0.5000000000 0.3333333333 0.2500000000 0.2000000000 0.1666666667 0.1428571429
0.3333333333 0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000
0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111
0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000
0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909
Masukkan nilai Bi secara berurut :
1 0 0 0 0 0

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x0 : 35.9984332426154
x1 : -629.9561348353582
x2 : 3359.7082215341184
x3 : -7559.251479106894
x4 : 7559.183048492995
x5 : -2771.681076322824
Tekan ENTER untuk kembali ke menu utama
```

n = 10

```
Masukkan nilai Aij secara berurut :
1.0000000000 0.5000000000 0.3333333333 0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000
0.5000000000 0.3333333333 0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909
0.3333333333 0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333
0.2500000000 0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769
0.2000000000 0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714
0.1666666667 0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714 0.0666666667
0.1428571429 0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714 0.0666666667 0.0625000000
0.1250000000 0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714 0.0666666667 0.0625000000 0.0588235294
0.1111111111 0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714 0.0666666667 0.0625000000 0.0588235294 0.0555555556
0.1000000000 0.0909090909 0.0833333333 0.0769230769 0.0714285714 0.0666666667 0.0625000000 0.0588235294 0.0555555556 0.0526315789
Masukkan nilai Bi secara berurut :
1 0 0 0 0 0 0 0 0 0

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x0 : 31.524720064396885
x1 : 53.531925846826425
x2 : -11856.868239527721
x3 : 111687.21658825113
x4 : -414690.1563623029
x5 : 694143.016176682
x6 : -360708.7969884537
x7 : -389724.1153111986
x8 : 572636.4642543644
x9 : -201602.08093172696
Tekan ENTER untuk kembali ke menu utama
```

2. SPL berbentuk matriks *augmented*

a.

$$\begin{bmatrix} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{bmatrix}.$$

Output Program:

```
Masukkan nama file input yang sudah diletakkan di test/input/ (termasuk ekstensi) :
SPL2a.txt

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x1 = -1.00a1 - 1.00
x2 = -2.00a0
x3 = a0
x4 = a1
Tekan ENTER untuk kembali ke menu utama
```

b.

$$\begin{bmatrix} 2 & 0 & 8 & 0 & 8 \\ 0 & 1 & 0 & 4 & 6 \\ -4 & 0 & 6 & 0 & 6 \\ 0 & -2 & 0 & 3 & -1 \\ 2 & 0 & -4 & 0 & -4 \\ 0 & 1 & 0 & -2 & 0 \end{bmatrix}.$$

Output Program:

```
Masukkan nama file input yang sudah diletakkan di test/input/ (termasuk ekstensi) :
SPL2b.txt

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x1 = 0.000000
x2 = 2.000000
x3 = 1.000000
x4 = 1.000000
Tekan ENTER untuk kembali ke menu utama
```

3. SPL berbentuk

a. $$\begin{aligned} 8x_1 + x_2 + 3x_3 + 2x_4 &= 0 \\ 2x_1 + 9x_2 - x_3 - 2x_4 &= 1 \\ x_1 + 3x_2 + 2x_3 - x_4 &= 2 \\ x_1 + 6x_3 + 4x_4 &= 3 \end{aligned}$$

Output Program:

```
Masukkan nama file input yang sudah diletakkan di test/input/ (termasuk ekstensi) :
SPL3a.txt

Jenis keluaran :
1. Layar
2. File
Masukkan pilihan Anda : 1
x0 : -0.2243243243243243
x1 : 0.18243243243243243
x2 : 0.7094594594594594
x3 : -0.2581081081081081
Tekan ENTER untuk kembali ke menu utama
```

b.

$$\begin{aligned} x_7 + x_8 + x_9 &= 13.00 \\ x_4 + x_5 + x_6 &= 15.00 \\ x_1 + x_2 + x_3 &= 8.00 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_6 + x_8) + 0.61396x_9 &= 14.79 \\ 0.91421(x_3 + x_5 + x_7) + 0.25(x_2 + x_4 + x_6 + x_8) &= 14.31 \\ 0.04289(x_3 + x_5 + x_7) + 0.75(x_2 + x_4) + 0.61396x_1 &= 3.81 \\ x_3 + x_6 + x_9 &= 18.00 \\ x_2 + x_5 + x_8 &= 12.00 \\ x_1 + x_4 + x_7 &= 6.00 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_2 + x_6) + 0.61396x_3 &= 10.51 \\ 0.91421(x_1 + x_5 + x_9) + 0.25(x_2 + x_4 + x_6 + x_8) &= 16.13 \\ 0.04289(x_1 + x_5 + x_9) + 0.75(x_4 + x_8) + 0.61396x_7 &= 7.04 \end{aligned}$$

Output Program:

```
Masukkan pilihan Anda : 1
Masukkan banyak baris (m): 12
Masukkan banyak kolom (n): 9
Masukkan nilai Aij secara berurut :
0 0 0 0 0 0 1 1 1
0 0 0 1 1 1 0 0 0
1 1 1 0 0 0 0 0 0
0 0 0.04289 0 0.4289 0.75 0.04289 0.75 0.61
396
0 0.25 0.91421 0.25 0.91421 0.25 0.91421 0.25 0
0.61396 0.75 0.04289 0.75 0.04289 0 0.04289 0 0
0 0 1 0 0 1 0 0 1
0 1 0 0 1 0 0 1 0
1 0 0 1 0 0 1 0 0
0.04289 0.75 0.61396 0 0.04289 0.75 0 0 0.0
4289
0.91421 0.25 0 0.25 0.91421 0.25 0 0.25 0.9
1421
0.04289 0 0 0.75 0.04289 0 0.61396 0.75 0.0
4289
Masukkan nilai Bi secara berurut :
13
15
8
SPL tidak konsisten, tidak ada solusi!
Tekan ENTER untuk kembali ke menu utama
```

4. **Studi Kasus Interpolasi**

a. Gunakan tabel di bawah ini untuk mencari polinom interpolasi dari pasangan titik-titik yang terdapat dalam tabel. Program menerima masukan nilai x yang akan dicari nilai fungsi $f(x)$.

x	0.4	0.7	0.11	0.14	0.17	0.2	0.23
$f(x)$	0.043	0.005	0. 058	0.072	0.1	0.13	0.147

Lakukan pengujian pada nilai-nilai default berikut:

$x = 0.2$ f(x) = ?
$x = 0.55$ f(x) = ?

$x = 0.85$ f(x) = ?
$x = 1.28$ f(x) = ?

```
Dimulai dari x0 y0, x1 y1, ... xn yn :
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
Persamaan f(x) = 13.874967163303259x^0.0 -430.7440627232415x^1.0
5287.853257111551x^2.0 -32545.52758378136x^3.0 104472.99091577974
x^4.0 -162780.63626808464x^5.0 94639.90480746611x^6.0
Masukkan nilai x yang ingin dicari nilainya : 0.2
4.401812248033821E-12
Tekan ENTER untuk kembali ke menu utama
```

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
Persamaan f(x) = 13.874967163303259x^0.0 -430.7440627232415x^1.0
5287.853257111551x^2.0 -32545.52758378136x^3.0 104472.99091577974
x^4.0 -162780.63626808464x^5.0 94639.90480746611x^6.0
Masukkan nilai x yang ingin dicari nilainya : 0.55
-51.091068691641794
Tekan ENTER untuk kembali ke menu utama
```

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
Persamaan f(x) = 13.874967163303259x^0.0 -430.7440627232415x^1.0
5287.853257111551x^2.0 -32545.52758378136x^3.0 104472.99091577974
x^4.0 -162780.63626808464x^5.0 94639.90480746611x^6.0
Masukkan nilai x yang ingin dicari nilainya : 0.85
1483.5097516047754
Tekan ENTER untuk kembali ke menu utama
```

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
0.4 0.043
0.7 0.005
0.11 0.058
0.14 0.072
0.17 0.1
0.2 0.13
0.23 0.147
Persamaan f(x) = 13.874967163303259x^0.0 -430.7440627232415x^1.0
5287.853257111551x^2.0 -32545.52758378136x^3.0 104472.99091577974
x^4.0 -162780.63626808464x^5.0 94639.90480746611x^6.0
Masukkan nilai x yang ingin dicari nilainya : 1.28
77236.46843645384
Tekan ENTER untuk kembali ke menu utama
```

b. Jumlah kasus positif baru Covid-19 di Indonesia semakin fluktuatif dari hari ke hari. Di bawah ini diperlihatkan jumlah kasus baru Covid-19 di Indonesia mulai dari tanggal 17 Juni 2022 hingga 31 Agustus 2022:

Tanggal	Tanggal (desimal)	Jumlah Kasus Baru
17/06/2022	6,567	12.624
30/06/2022	7	21.807
08/07/2022	7,258	38.391
14/07/2022	7,451	54.517
17/07/2022	7,548	51.952
26/07/2022	7,839	28.228
05/08/2022	8,161	35.764
15/08/2022	8,484	20.813
22/08/2022	8,709	12.408
31/08/2022	9	10.534

Tanggal (desimal) adalah tanggal yang sudah diolah ke dalam bentuk desimal 3 angka di belakang koma dengan memanfaatkan perhitungan sebagai berikut:

tanggal(desimal) = bulan + (tanggal / jumlah hari pada bulan tersebut)

Sebagai **contoh**, untuk tanggal 17/06/2022 (dibaca: 17 Juni 2022) diperoleh tanggal(desimal) sebagai berikut:

$$\text{Tanggal(desimal)} = 6 + (17/30) = 6{,}567$$

Gunakanlah data di atas dengan memanfaatkan **polinom interpolasi** untuk melakukan prediksi jumlah kasus baru Covid-19 pada tanggal-tanggal berikut:

a. 16/07/2022
b. 10/08/2022
c. 05/09/2022
d. beserta masukan user lainnya berupa **tanggal (desimal) yang sudah diolah** dengan asumsi prediksi selalu dilakukan untuk tahun 2022.

Output Program:

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
6.567 12624
7 21807
7.258 38391
7.451 54517
7,548 51952
7.839 28228
8.161 35764
8.484 20813
8.709 12408
9 10534
Persamaan f(x) = -4.485472253653449E9x^0.0 4.39893816470358E9x^1.
0 -1.8708230299540336E9x^2.0 4.499733843295453E8x^3.0 -6.68089687
6593643E7x^4.0 6252957.043917188x^5.0 -358878.77705438185x^6.0 11
483.66711858679x^7.0 -155.75377567899253x^8.0 0.02043437225770643
2x^9.0
Masukkan nilai x yang ingin dicari nilainya : 7.516
1518824.2732649301
Tekan ENTER untuk kembali ke menu utama
```

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
6.567 12624
7 21807
7.258 38391
7.451 54517
7,548 51952
7.839 28228
8.161 35764
8.484 20813
8.709 12408
9 10534
Persamaan f(x) = -4.485472253653449E9x^0.0 4.39893816470358E9x^1.
0 -1.8708230299540336E9x^2.0 4.499733843295453E8x^3.0 -6.68089687
6593643E7x^4.0 6252957.043917188x^5.0 -358878.77705438185x^6.0 11
483.66711858679x^7.0 -155.75377567899253x^8.0 0.02043437225770643
2x^9.0
Masukkan nilai x yang ingin dicari nilainya : 8.322
1998529.6904437863
Tekan ENTER untuk kembali ke menu utama
```

```
Masukkan titik-titik x dan y (cukup dipisah spasi tanpa kurung)
Dimulai dari x0 y0, x1 y1, ... xn yn :
6.567 12624
7 21807
7.258 38391
7.451 54517
7,548 51952
7.839 28228
8.161 35764
8.484 20813
8.709 12408
9 10534
Persamaan f(x) = -4.485472253653449E9x^0.0 4.39893816470358E9x^1.
0 -1.8708230299540336E9x^2.0 4.499733843295453E8x^3.0 -6.68089687
6593643E7x^4.0 6252957.043917188x^5.0 -358878.77705438185x^6.0 11
483.66711858679x^7.0 -155.75377567899253x^8.0 0.02043437225770643
2x^9.0
Masukkan nilai x yang ingin dicari nilainya : 9.166
2613893.6662086155
Tekan ENTER untuk kembali ke menu utama
```

4. **Studi Kasus Interpolasi Bicubic**

Diberikan matriks input:

$$\begin{bmatrix} 153 & 59 & 210 & 96 \\ 125 & 161 & 72 & 81 \\ 98 & 101 & 42 & 12 \\ 21 & 51 & 0 & 16 \end{bmatrix}$$

Tentukan nilai:

f(0,0) = ?

f(0.5, 0.5) = ?

f(0.25, 0.75) = ?

f(0.1, 0.9) = ?

Output Program:

```
Nilai dari f(0.00,0.00) adalah 161.00
Tekan ENTER untuk kembali ke menu utama
```

```
Nilai dari f(0.50,0.50) adalah 97.73
Tekan ENTER untuk kembali ke menu utama
```

```
Nilai dari f(0.25,0.75) adalah 105.51
Tekan ENTER untuk kembali ke menu utama
```

```
Nilai dari f(0.10,0.90) adalah 104.23
Tekan ENTER untuk kembali ke menu utama
```

5. **Studi Kasus Regresi Linier Berganda**

Diberikan sekumpulan data sesuai pada tabel berikut ini.

Table 12.1: Data for Example 12.1

Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3	Nitrous Oxide, y	Humidity, x_1	Temp., x_2	Pressure, x_3
0.90	72.4	76.3	29.18	1.07	23.2	76.8	29.38
0.91	41.6	70.3	29.35	0.94	47.4	86.6	29.35
0.96	34.3	77.1	29.24	1.10	31.5	76.9	29.63
0.89	35.1	68.0	29.27	1.10	10.6	86.3	29.56
1.00	10.7	79.0	29.78	1.10	11.2	86.0	29.48
1.10	12.9	67.4	29.39	0.91	73.3	76.3	29.40
1.15	8.3	66.8	29.69	0.87	75.4	77.9	29.28
1.03	20.1	76.9	29.48	0.78	96.6	78.7	29.29
0.77	72.2	77.7	29.09	0.82	107.4	86.8	29.03
1.07	24.0	67.7	29.60	0.95	54.9	70.9	29.37

Source: Charles T. Hare, "Light-Duty Diesel Emission Correction Factors for Ambient Conditions," EPA-600/2-77-116. U.S. Environmental Protection Agency.

Gunakan *Normal Estimation Equation for Multiple Linear Regression* untuk mendapatkan regresi linear berganda dari data pada tabel di atas, kemudian

estimasi nilai Nitrous Oxide apabila Humidity bernilai 50%, temperatur 76°F, dan tekanan udara sebesar 29.30.

Dari data-data tersebut, apabila diterapkan *Normal Estimation Equation for Multiple Linear Regression*, maka diperoleh sistem persamaan linear sebagai berikut.

$$20b_0 + 863.1b_1 + 1530.4b_2 + 587.84b_3 = 19.42$$

$$863.1b_0 + 54876.89b_1 + 67000.09b_2 + 25283.395b_3 = 779.477$$

$$1530.4b_0 + 67000.09b_1 + 117912.32b_2 + 44976.867b_3 = 1483.437$$

$$587.84b_0 + 25283.395b_1 + 44976.867b_2 + 17278.5086b_3 = 571.1219$$

```
Rumus y yang didapat :
y = -3.507778 - 0.002625x1 + 0.000799x2 + 0.154155x3
Nilai yang didapatkan dari f(x1k,x2k,..,xnk)  :
0.9384342262216849
Tekan ENTER untuk kembali ke menu utama
```

BAB 5
KESIMPULAN, SARAN, DAN REFLEKSI

Kesimpulan

Program yang dibuat pada tugas ini dapat:

1. menghitung solusi SPL dengan metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer;
2. menghitung matriks balikan;
3. menghitung determinan matriks dengan metode reduksi baris dan metode ekspansi kofaktor.
4. menyelesaikan persoalan interpolasi polinom dan bicubic, serta regresi linier berganda;

Saran

- Memeriksa setiap kasus khusus untuk penyelesaian SPL yang memiliki solusi banyak/tidak unik, karena matriks eselonnya dapat melalui kasus-kasus yang tidak dapat diselesaikan dengan OBE bertahap dari baris dan kolom 1, hingga baris dan kolom terakhir. Tetapi memerlukan pertukaran dan pemeriksaan ketika ada koefisien variabel yang elemen 0 pada diagonal utamanya.
- Meningkatkan presisi perhitungan terutama dalam perhitungan determinan karena ada kemungkinan terbuat determinan 0 dari yang sebenarnya mempunyai nilai namun sangat kecil sehingga membuat perhitungan metode-metode yang memakai determinan menjadi tidak mungkin.

Refleksi

- Tugas ini mendorong penulis untuk belajar tentang aljabar linier, bahasa Java, dan berkolaborasi dengan platform Github

REFERENSI

https://www.profematika.com/eliminasi-gauss-dan-contoh-penerapannya/
https://en.wikipedia.org/wiki/Laplace_expansion
https://en.wikipedia.org/wiki/Minor_(linear_algebra)
https://en.wikipedia.org/wiki/Invertible_matrix
https://www.uniksharianja.com/2015/03/minor-kofaktor-matrik-kofaktor-dan.html
Laaksonen, Antti. 2017. Competitive Programmer’s Handbook. diambil dari https://cses.fi/book/book.pdf.