

# Pracownia z analizy numerycznej

## Sprawozdanie do zadania **P2.20**. Redukcja macierzy metodą Gaussa

Prowadzący: dr Witold Karczewski

Aleksander Balicki, nr indeksu: 220989  
Dominika Rogozińska, nr indeksu: 221094

Wrocław, 5 grudnia 2010r.

### 1. Wstęp

Metodę Gaussa redukcji macierzy wykorzystuje się do rozwiązywania takich problemów jak znajdowanie macierzy odwrotnej, obliczanie rzędu macierzy, a także rozwiązywanie układów równań z wieloma niewiadomymi. Efektywność tej metody zależy od szczegółów implementacji i modyfikacji algorytmu oraz wskaźnika uwarunkowania macierzy. Poniżej zostały zaprezentowane wyniki otrzymane dla eliminacji Gaussa bez i z następującymi modyfikacjami: wybór największego (co do modułu) «*«*«*«*reduktora*»»»*»»*» z wiersza, z kolumny, z podmacierzy (wybór pełny). Badania zostały przeprowadzone dla kilku rodzajów macierzy: macierzy Hilberta, macierzy Pei, macierzy losowej z dominującą przekątną oraz macierzy oraz macierzy losowej, w której większość elementów należy do przedziału  $(-1, 1)$ , a kilka jest wybranych z zakresu  $(-1000, 1000)$ .*

### 2. Definicje

**Definicja 1.** *Macierzą o wymiarach  $m \times n$  (macierzą o  $m$  wierszach i  $n$  kolumnach), nad ciałem  $K$  nazywamy każdą funkcję typu  $\{1, \dots, m\} \times \{1, \dots, n\} \rightarrow K$ .*

Interesującą będą nas macierze o rozmiarach  $n \times n$ , które przedstawiają układy  $n$  równań z  $n$  niewiadomymi. Przykładowymi danymi do badań sposobów rozwiązywania takich układów były macierze rzędu  $n$ , które są nieosobliwe, więc układy te zawsze mają rozwiązanie. Macierz zapisujemy w następujący sposób:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Weźmy przykładowy układ równań:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \end{cases}$$

Z tym układem wiążemy macierz układu  $A$  oraz wektor wyrazów wolnych  $b$ :

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

**Definicja 2.** *Macierzą Hilberta nazywamy macierz  $n \times n$ , w której*

$$a_{i,j} = \frac{1}{i+j-1}$$

**Definicja 3.** *Macierzą Pei nazywamy macierz  $n \times n$ , w której*

$$a_{i,j} = \begin{cases} 1 & i \neq j \\ d & i = j \end{cases}$$

gdzie  $d$  jest parametrem.

**Definicja 4.** *Macierzą o dominującej przekątnej nazywamy macierz, w której*

$$\bigwedge_{1 \leq i \leq n} \sum_{j \neq i} |a_{i,j}| \leq |a_{i,i}|$$

**Definicja 5.** *Normę maksimum dla wektora  $x = [x_1, x_2, \dots, x_n]$  definiujemy jako*

$$\|x\|_{\infty} = \max\{|x_i| : i = 1, 2, \dots, n\}$$

Normy maksimum użyjemy jako wskaźnika numerycznej poprawności metody Gaussa, porównując wartości  $\|b - A\tilde{x}\|_{\infty}$  dla wszystkich prób.  $\tilde{x}$  to nasze przybliżone rozwiązanie, więc  $\|b - A\tilde{x}\|_{\infty}$  oznacza największy spośród błędów przybliżeń  $x_i$ .

### 3. Metoda eliminacji Gaussa

Metoda ta została stworzona przez Carla Friedricha Gaussa. Daje ona algorytm do rozwiązania układu równań liniowych, obliczenia rzędu macierzy i znalezienia macierzy odwrotnej do danej. Algorytm składa się z 2 kroków, najpierw doprowadzamy macierz do postaci schodkowej, a następnie znajdujemy wynik układu poprzez podstawienie w tył (funkcja back substitution). W metodzie Gaussa stosuje się 3 operacje elementarne na wierszach macierzy. Te operacje to:

1. Zamiana kolejności wierszy
2. Pomnożenie wszystkich wartości w wierszu przez niezerowy skalar  $\lambda$
3. Dodanie do dowolnego wiersza kombinacji liniowej pozostałych wierszy

Operacje elementarne mają ciekawe własności, mianowicie:

- Nie zmieniają rzędu macierzy
- Dowolną macierz można za pomocą skończonej liczby kroków doprowadzić do macierzy w postaci schodkowej

### 3.1. Metoda eliminacji Gaussa bez wyboru elementów głównych

Wykonujemy następujący algorytm:

```
for k from 1 to N
    przeskaluj k-ty wiersz, aby  $m[k,k] = 1$ 
    for i from k+1 to N
        odejmij od i-tego wiersza k-ty wiersz  $m[i,k]$  razy
    wykonaj back_substitution
```

Powyższy algorytm ma pewien problem. Co jeżeli  $m[k,k] = 0$  dla któregoś  $k$ . Nie jesteśmy wtedy w stanie znaleźć takiego skalaru, aby po pomnożeniu na tym miejscu znalazło się 1. Nawet jeżeli odrzucimy te przypadki, to na przekątnej mogą się znaleźć elementy bardzo bliskie zeru. Wtedy cały nasz wiersz zostanie podzielony przez liczbę bliską zeru i wygeneruje duże błędy.

### 3.2. Metoda eliminacji Gaussa z wyborem elementów głównych z wiersza lub kolumny

Jednym sposobem poradzenia sobie z tym problemem jest wybór elementów głównych z wiersza lub kolumny. W wyborze głównego elementu z wiersza, w każdym kroku głównej pętli naszego programu wyszukujemy maksymalny element z wiersza i zamieniamy kolumnę obecną i tą z maksymalnym elementem. Wiemy, że zamiana kolumn przestawia nam tylko kolejność zmiennych w rozwiązaniu. Tym sposobem zapewniamy to, że zawsze dzielimy przez największy element z wiersza, czyli dzielenie przez liczby bliskie zeru występuje rzadziej. Analogicznie działa wybór głównego elementu z kolumny.

```
for k from 1 to N
    znajdź maksymalną wartość  $|m[k,i]|$  dla wszystkich  $i > k$ 
    zamień kolumnę z maksymalną wartością z k-tą kolumną
    przeskaluj k-ty wiersz, aby  $m[k,k] = 1$ 
    for i from k+1 to N
        odejmij od i-tego wiersza k-ty wiersz  $m[i,k]$  razy
    wykonaj back_substitution
```

### 3.3. Metoda eliminacji Gaussa z pełnym wyborem elementów głównych(z podmacierzy)

Jest to rozszerzenie poprzedniego pomysłu z wyborem elementów, tym razem wyszukujemy maksymalnego elementu z podmacierzy i dokonujemy zamiany wiersza oraz kolumny. Metoda ta jest kosztowna obliczeniowo, bo w każdym kroku musimy znaleźć maksymalny element spośród  $O(n^2)$  elementów.

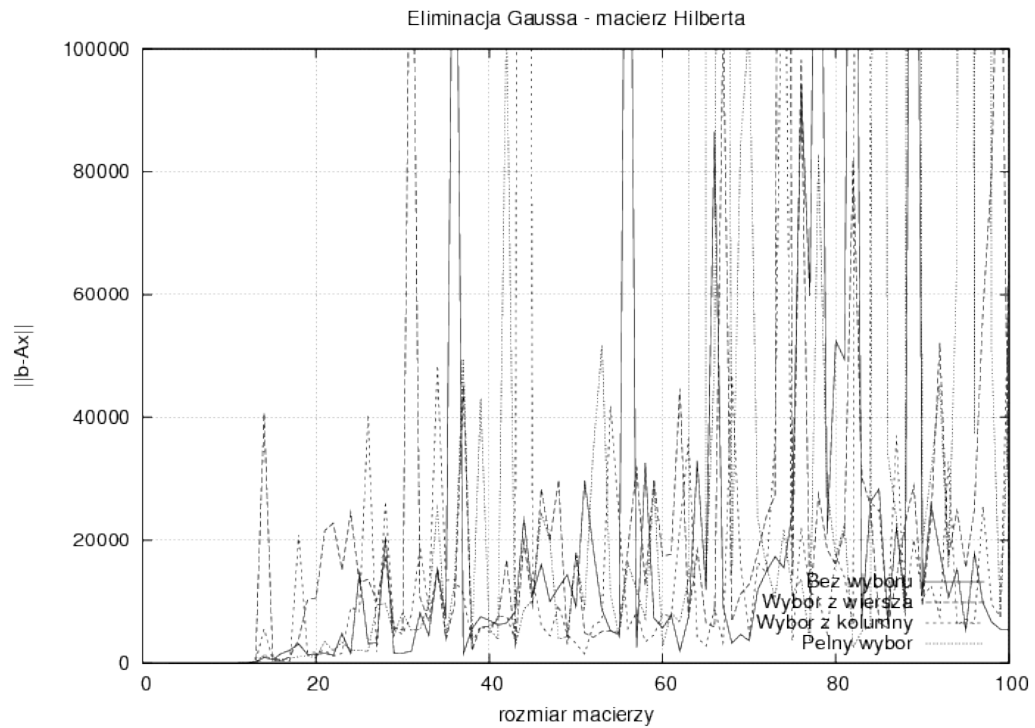
```
for k from 1 to N-1
    znajdź i,l takie, że  $|m[i,l]|$  jest maksymalne dla wszystkich  $i > k, l > k$ 
    zamień l-tą kolumnę z k-tą kolumną
    zamień i-ty wiersz z k-tym wierszem
    przeskaluj k-ty wiersz, aby  $m[k,k] = 1$ 
    for i from k+1 to N
        odejmij od i-tego wiersza k-ty wiersz  $m[i,k]$  razy
    wykonaj back_substitution1
```

## 4. Program

Program testujący jest napisany w języku C++. Użyto typu podwójnej precyzji (double). Wyniki zostały zapisane jako plik z rozszerzeniem .csv, a następnie opracowane graficznie za pomocą programu gnuplot.

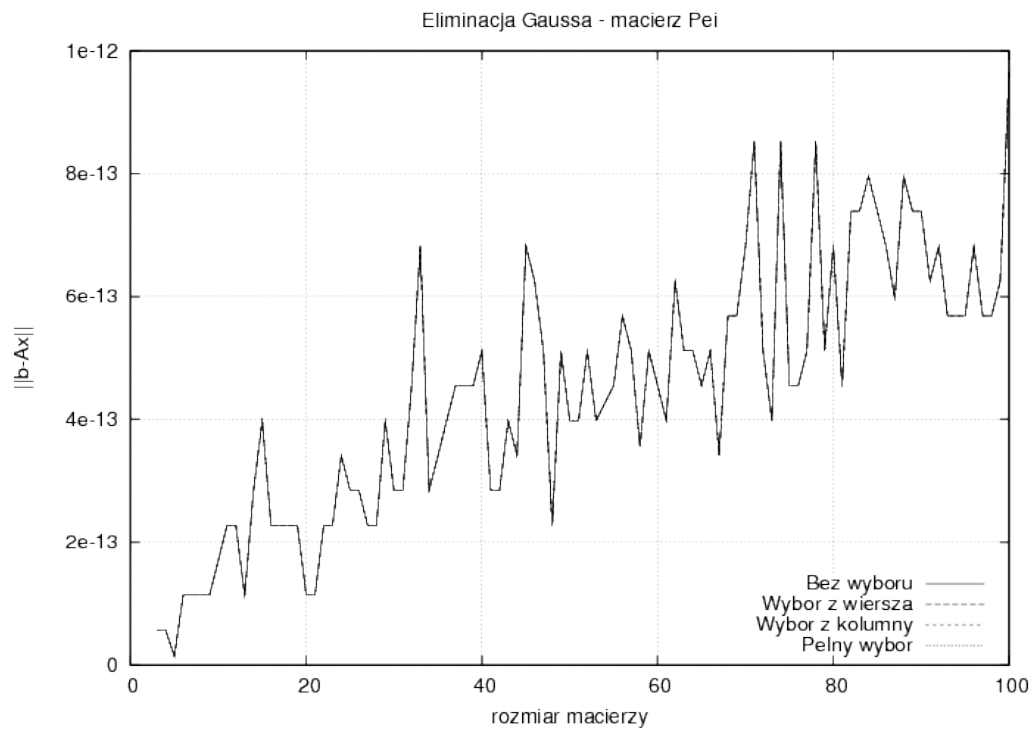
## 5. Wyniki prób

### 5.1. Macierz Hilberta



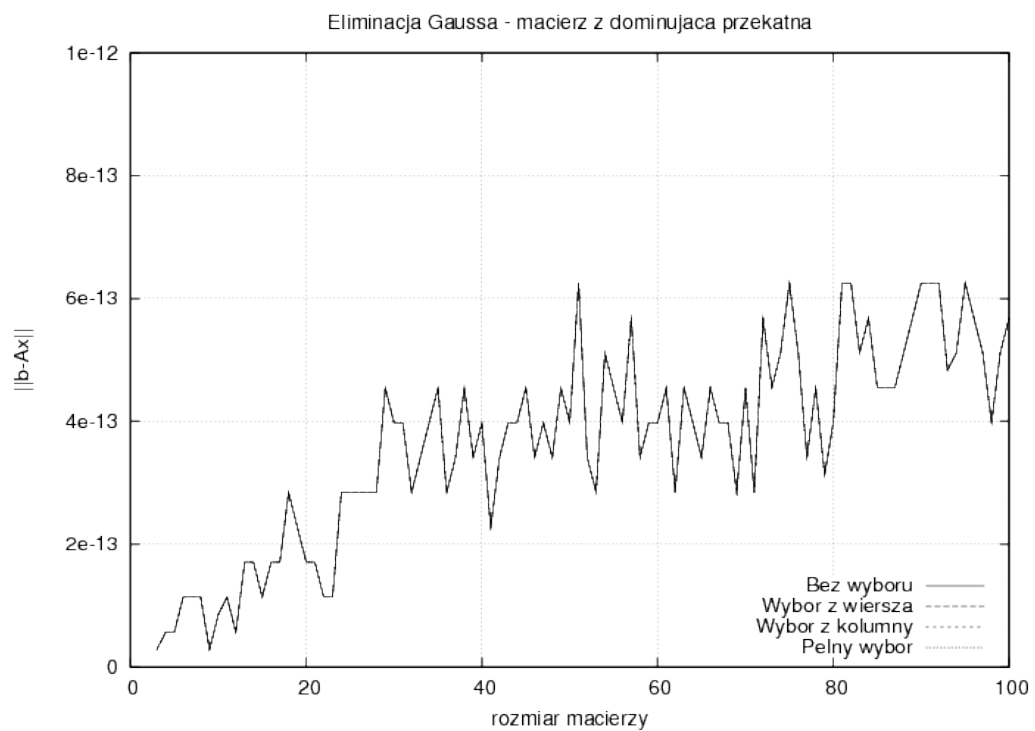
Źle uwarunkowana, leci w kosmos. Wykres nie obejmuje całego zakresu.

## 5.2. Macierz Pei



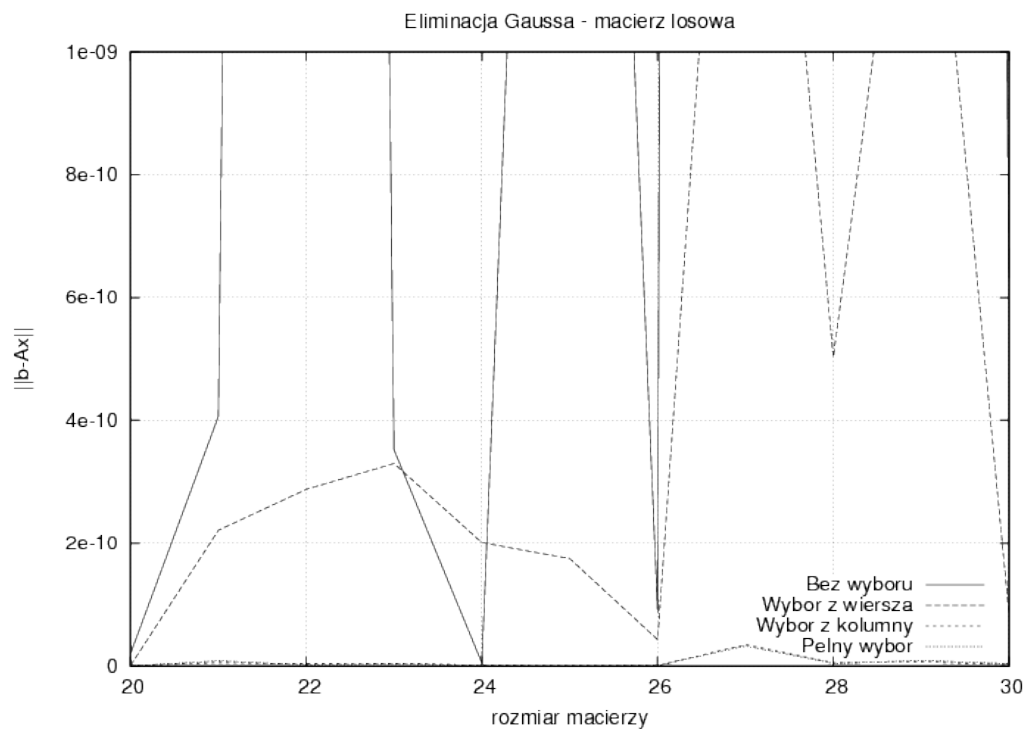
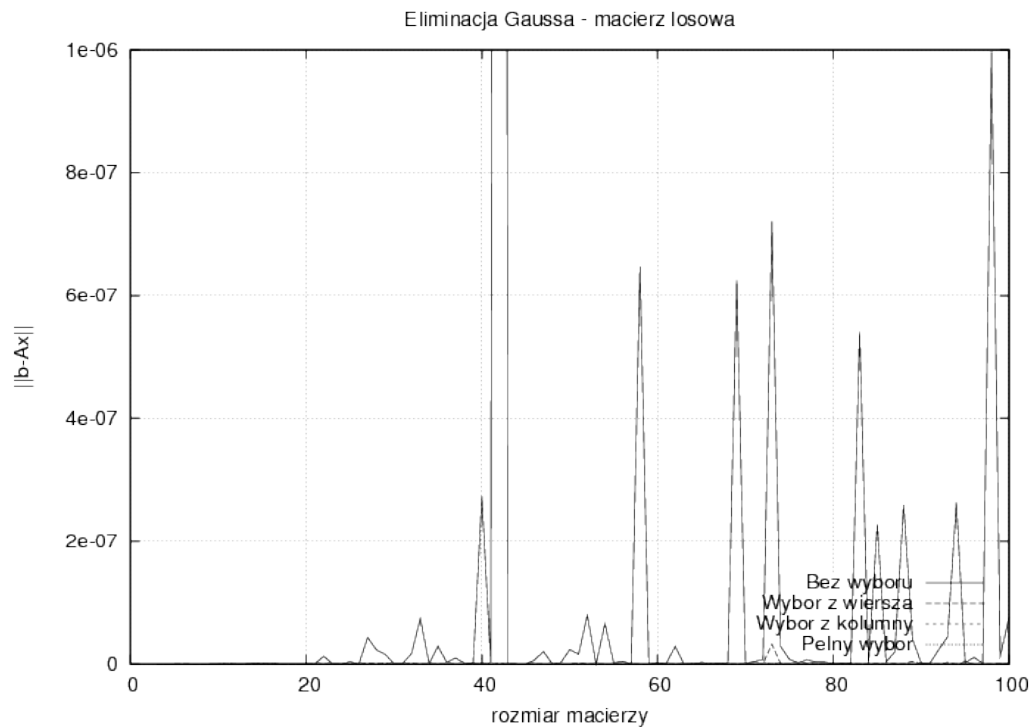
Dobrze uwarunkowana, wygląda tak samo dla każdej modyfikacji algorytmu -i, na przekątnej są największe liczby i są równe.

## 5.3. Macierz z dominującą przekątną



Dobrze uwarunkowana, dla metody bez modyfikacji i wyboru z wiersza(po którym zmieniamy kolumny) nie za dobrze, dla wyboru z kolumn i full lepiej i prawie tak samo.

#### 5.4. Macierz losowa



## 6. Wnioski

### Literatura

- [1] Notatki z wykładu Algebra Emanuela Kierońskiego
- [2] Notatki z wykładu Analiza Numeryczna Stanisława Lewanowicza
- [3] Kincaid David, Cheney Ward, Analiza numeryczna
- [4] J. M. Jankowscy, Przegląd metod i algorytmów numerycznych
- [5] <http://wolframalpha.com/>
- [6] <http://en.wikipedia.org/>
- [7] <http://wazniak.mimuw.edu.pl/>