

# Rozwiązywanie układów równań metodami bezpośrednimi

## Metoda Gaussa-Jordana

Michał Saturczak

Informatyka Stosowana, Akademia Górniczo-Hutnicza

Metody Numeryczne

9 marca 2024

# 1 Wstęp teoretyczny

**Metoda Gaussa-Jordana** jest techniką stosowaną w algebrze liniowej do rozwiązywania układów równań liniowych. Jest to rozszerzenie metody eliminacji Gaussa. Podczas gdy metoda eliminacji Gaussa przekształca system do postaci macierzy górnotrójkątnej, metoda Gaussa-Jordana przekształca system do postaci macierzy diagonalnej, co umożliwia bezpośrednie odczytanie rozwiązania.

Oto podstawowe kroki metody Gaussa-Jordana:

- **Tworzenie rozszerzonej macierzy:** Rozpoczyna się od utworzenia rozszerzonej macierzy układu, której kolumny odpowiadają współczynnikom równań, a ostatnia kolumna reprezentuje wektor prawych stron
- **Eliminacja w przód:** Następnie przekształca się macierz do postaci macierzy górnotrójkątnej, używając operacji elementarnych na wierszach (dodawanie, odejmowanie, mnożenie wiersza przez skalar i zamiana miejscami dwóch wierszy). Celem jest uzyskanie zera pod główną przekątną.
- **Eliminacja wstecz:** Kontynuowane jest przekształcanie macierzy do postaci diagonalnej, używając tych samych operacji elementarnych na wierszach. Celem jest uzyskanie zera powyżej głównej przekątnej.
- **Normalizacja:** Każdy wiersz dzieli się przez wartość na przekątnej, aby uzyskać jedynki na przekątnej. Macierz jest teraz w postaci kanonicznej, a rozwiązania układu równań można odczytać bezpośrednio z ostatniej kolumny rozszerzonej macierzy.

*Przykładowy układ równań:*

$$\begin{cases} a_1x_1 + b_1x_2 + c_1x_3 = d_1 \\ a_2x_1 + b_2x_2 + c_2x_3 = d_2 \\ a_3x_1 + b_3x_2 + c_3x_3 = d_3 \end{cases} \quad (1)$$

*Układ zapisany w postaci macierzowej:*

$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \quad (2)$$

Dla wyznacznika poniżej, który powstał z macierzy rozszerzonej powstałej z pierwszej i trzeciej macierzy, będziemy dokonywać przekształceń, by otrzymać wyznacznik macierzy jednostkowej

$$\left[ \begin{array}{ccc|c} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{array} \right] \quad (3)$$

Po przekształceniach otrzymujemy wyznacznik takiej postaci:

$$\left[ \begin{array}{ccc|c} 1 & 0 & 0 & e_1 \\ 0 & 1 & 0 & e_2 \\ 0 & 0 & 1 & e_3 \end{array} \right] \quad (4)$$

Poprzez posiadanie na przekątnych macierzy jedynek, gdzie reszta jest zerami wyniki mogą być odczytane.

## 2 Problem

Macierz  $A$  oraz wektor  $b$ , dla których rozpatrzono rozwiązywanie układów równań liniowych metodą Gaussa-Jordana:

$$A = \begin{bmatrix} 2q & 1 & 6 & 9 & 10 \\ 2 & 1 & 6 & 9 & 10 \\ 1 & 6 & 6 & 8 & 6 \\ 5 & 9 & 10 & 7 & 10 \\ 3 & 4 & 9 & 7 & 9 \end{bmatrix} \text{ oraz } b = \begin{bmatrix} 10 \\ 2 \\ 9 \\ 9 \\ 3 \end{bmatrix}.$$

Powyższy układ rozwiązano w zależności od parametru  $q$ , który należał od 0 do 3. Parametr  $q$  zmieniał się z krokiem 0.01. Rozwiązując dany układ otrzymano wektor  $x$ , będący wektorem rozwiązań układu równań.

Następnie policzono wektor  $c$ , który w idealnej sytuacji mógł być wektorem  $b$ . Wektor  $c$  powstał w wyniku przemnożenia macierzy  $A$  przez wektor rozwiązań, czyli wektor  $x$ .

Na końcu policzono odchylenie między wektorami  $c$  i  $b$  dla wszystkich wartości  $q$  z poniższego wzoru:

$$o(q) = 1/5\sqrt{\sum (c_i - b_i)^2}.$$

Powyższy problem rozwiązano korzystając z biblioteki **GSL** w języku  $C++$ , oraz przez zaimplementowane funkcje pomocnicze:

- **gsl\_matrix\_scale\_row** funkcja skaluje (mnoży) wszystkie elementy w  $i$ -tym wierszu macierzy  $A$  przez podany factor
- **gsl\_matrix\_sub\_row** funkcja odejmuje od  $i$ -tego wiersza macierzy  $A$   $j$ -ty wiersz macierzy  $B$  pomnożony przez factor

```
void gsl_matrix_scale_row(gsl_matrix *A, int i, double factor);
void gsl_matrix_sub_row(gsl_matrix *A, int i, const gsl_matrix *B, int j, double factor);

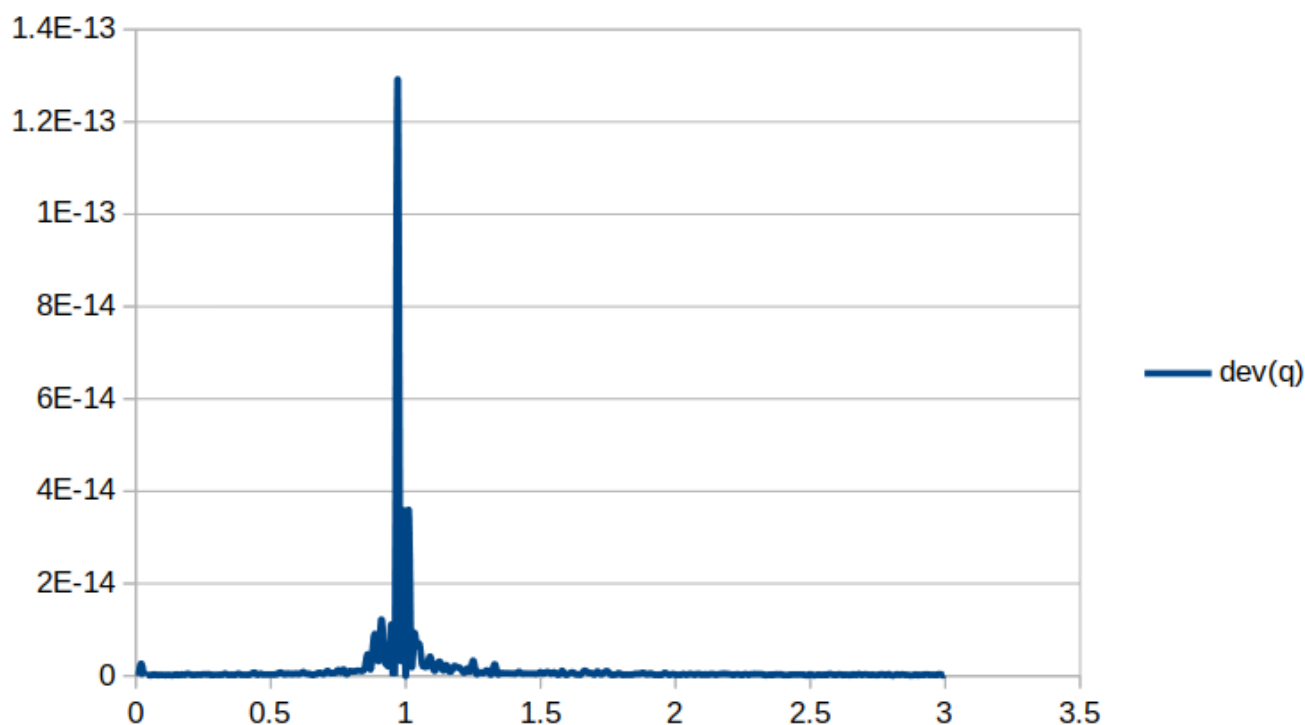
void gauss_jordan(gsl_matrix *A, gsl_vector *b)
{
    int n = A->size1;
    for (int i = 0; i < n; i++)
    {
        double Aii = gsl_matrix_get(A, i, i);
        gsl_matrix_scale_row(A, i, 1.0 / Aii);
        gsl_vector_set(b, i, gsl_vector_get(b, i) / Aii);
        for (int j = 0; j < n; j++)
        {
            if (i != j)
            {
                double factor = gsl_matrix_get(A, j, i);
                gsl_matrix_sub_row(A, j, A, i, factor);
                gsl_vector_set(b, j, gsl_vector_get(b, j) - factor * gsl_vector_get(b, i));
            }
        }
    }
}
```

## 3 Wyniki

### 3.1 Część surowych wyników z pliku dla poszczególnych wartości $q$

27	$q = 0.260000$	$x = [-5.405405$	$2.821453$	$-1.010389$	$-2.343074$	$3.713936$	$]$
28	$q = 0.270000$	$x = [-5.479452$	$2.841353$	$-1.015711$	$-2.385188$	$3.767851$	$]$
29	$q = 0.280000$	$x = [-5.555556$	$2.861806$	$-1.021181$	$-2.428472$	$3.823264$	$]$
30	$q = 0.290000$	$x = [-5.633803$	$2.882835$	$-1.026805$	$-2.472975$	$3.880238$	$]$
31	$q = 0.300000$	$x = [-5.714286$	$2.904464$	$-1.032589$	$-2.518750$	$3.938839$	$]$
32	$q = 0.310000$	$x = [-5.797101$	$2.926721$	$-1.038542$	$-2.565851$	$3.999139$	$]$
33	$q = 0.320000$	$x = [-5.882353$	$2.949632$	$-1.044669$	$-2.614338$	$4.061213$	$]$
34	$q = 0.330000$	$x = [-5.970149$	$2.973228$	$-1.050979$	$-2.664272$	$4.125140$	$]$
35	$q = 0.340000$	$x = [-6.060606$	$2.997538$	$-1.057481$	$-2.715720$	$4.191004$	$]$
36	$q = 0.350000$	$x = [-6.153846$	$3.022596$	$-1.064183$	$-2.768750$	$4.258894$	$]$
37	$q = 0.360000$	$x = [-6.250000$	$3.048438$	$-1.071094$	$-2.823438$	$4.328906$	$]$
38	$q = 0.370000$	$x = [-6.349206$	$3.075099$	$-1.078224$	$-2.879861$	$4.401141$	$]$
39	$q = 0.380000$	$x = [-6.451613$	$3.102621$	$-1.085585$	$-2.938105$	$4.475706$	$]$
40	$q = 0.390000$	$x = [-6.557377$	$3.131045$	$-1.093186$	$-2.998258$	$4.552715$	$]$
41	$q = 0.400000$	$x = [-6.666667$	$3.160417$	$-1.101042$	$-3.060417$	$4.632292$	$]$
42	$q = 0.410000$	$x = [-6.779661$	$3.190784$	$-1.109163$	$-3.124682$	$4.714566$	$]$
43	$q = 0.420000$	$x = [-6.896552$	$3.222198$	$-1.117565$	$-3.191164$	$4.799677$	$]$
44	$q = 0.430000$	$x = [-7.017544$	$3.254715$	$-1.126261$	$-3.259978$	$4.887774$	$]$
45	$q = 0.440000$	$x = [-7.142857$	$3.288393$	$-1.135268$	$-3.331250$	$4.979018$	$]$
46	$q = 0.450000$	$x = [-7.272727$	$3.323295$	$-1.144602$	$-3.405114$	$5.073580$	$]$
47	$q = 0.460000$	$x = [-7.407407$	$3.359491$	$-1.154282$	$-3.481713$	$5.171644$	$]$
48	$q = 0.470000$	$x = [-7.547170$	$3.397052$	$-1.164328$	$-3.561203$	$5.273408$	$]$
49	$q = 0.480000$	$x = [-7.692308$	$3.436058$	$-1.174760$	$-3.643750$	$5.379087$	$]$
50	$q = 0.490000$	$x = [-7.843137$	$3.476593$	$-1.185600$	$-3.729534$	$5.488909$	$]$
51	$q = 0.500000$	$x = [-8.000000$	$3.518750$	$-1.196875$	$-3.818750$	$5.603125$	$]$
52	$q = 0.510000$	$x = [-8.163265$	$3.562628$	$-1.208610$	$-3.911607$	$5.722003$	$]$
53	$q = 0.520000$	$x = [-8.333333$	$3.608333$	$-1.220833$	$-4.008333$	$5.845833$	$]$
54	$q = 0.530000$	$x = [-8.510638$	$3.655984$	$-1.233577$	$-4.109176$	$5.974934$	$]$
55	$q = 0.540000$	$x = [-8.695652$	$3.705707$	$-1.246875$	$-4.214402$	$6.109647$	$]$
56	$q = 0.550000$	$x = [-8.888889$	$3.757639$	$-1.260764$	$-4.324306$	$6.250347$	$]$
57	$q = 0.560000$	$x = [-9.090909$	$3.811932$	$-1.275284$	$-4.439205$	$6.397443$	$]$
58	$q = 0.570000$	$x = [-9.302326$	$3.868750$	$-1.290480$	$-4.559448$	$6.551381$	$]$
59	$q = 0.580000$	$x = [-9.523810$	$3.928274$	$-1.306399$	$-4.685417$	$6.712649$	$]$
60	$q = 0.590000$	$x = [-9.756098$	$3.990701$	$-1.323095$	$-4.817530$	$6.881784$	$]$

### 3.2 Wykres parametru $q$ od odchylenia dla poszczególnych wartości



## 4 Wnioski

Analizując powyższe wyniki, można zauważyć, że odchylenia pomiędzy wektorami  $b$  i  $c$  są bardzo małe dla wszystkich wartości  $q$  w zakresie  $(0; 3)$  z krokiem 0.01. Zgodnie z definicją, wektor  $c$  jest wynikiem mnożenia macierzy  $A$  przez wektor rozwiązania  $x$ , a więc w idealnej sytuacji powinien być równy wektorowi  $b$ . Małe odchylenia sugerują, że rozwiązania znalezione dla układu równań są bliskie prawdziwym wartościom, co wskazuje na dokładność metody użytej do rozwiązania układu.

Ponadto, warto zauważyć, że odchylenia nie są stałe dla różnych wartości  $q$ , ale wykazują pewne niewielkie fluktuacje. Wartości te są jednak na tyle małe (rzędu  $10^{-15}$ ), że mogą być spowodowane błędami numerycznymi wynikającymi z precyzji reprezentacji liczb zmiennoprzecinkowych w komputerze, a nie rzeczywistymi różnicami w rozwiązaniach układu równań.

Na podstawie powyższych obserwacji, można stwierdzić, że mimo zmienności parametru  $q$ , układ równań rozwiązany metodą Gaussa-Jordana daje stabilne rozwiązania dla  $x$  w badanym zakresie  $q$ . Ta stabilność jest istotna, ponieważ pokazuje, że rozwiązania układu równań nie są wrażliwe na małe zmiany w macierzy  $A$ , co jest pożądaną właściwością w wielu zastosowaniach, takich jak rozwiązywanie równań różniczkowych, optymalizacja, czy analiza danych.

Podsumowując, analiza wyników pokazuje, że metoda użyta do rozwiązania układu równań jest dokładna i stabilna dla różnych wartości parametru  $q$  w zakresie  $(0; 3)$ . Odchylenia pomiędzy wektorami  $b$  i  $c$  są bardzo małe i mogą wynikać z błędów numerycznych, co sugeruje, że rozwiązania są bliskie prawdziwym wartościom. Ta analiza podkreśla znaczenie dokładności i stabilności metod numerycznych w kontekście rozwiązywania układów równań liniowych.