Projekt 3 - Obliczanie norm macierzowych i współczynnika uwarunkowania

Bartosz Hanc, Paweł Surdyka

1. Wprowadzenie

Celem niniejszego raportu jest przedstawienie algorytmów oraz implementacji obliczania norm macierzowych i współczynników uwarunkowania w języku Python. Przedstawione zostały normy: $\|M\|_1$, $\|M\|_\infty$, $\|M\|_2$, $\|M\|_p$ oraz odpowiadające im współczynniki uwarunkowania.

2. Wstęp teoretyczny

2.1. Normy macierzowe

Norma macierzy to funkcja przypisująca każdej macierzy nieujemną liczbę rzeczywistą, która w pewnym sensie mierzy jej "wielkość". Intuicyjnie można ją rozumieć jako odpowiednik długości wektora, ale rozszerzony na przestrzeń macierzy.

Dla macierzy $M \in \mathbb{R}^{m \times n}$ norma $\|M\|$ powinna spełniać następujące własności:

- $\|M\| \geq 0$ i $\|M\| = 0 \Leftrightarrow M = 0$
- $\|\alpha M\| = |\alpha| \cdot \|M\|$ dla każdego skalaru α
- $\|M + N\| \leq \|M\| + \|N\|$ (nierówność trójkąta)
- $\|M * N\| \leq \|M\| * \|N\|$

Szczególnym przypadkiem są **normy indukowane**, które dla macierzy traktowanej jako operator liniowy $M : \mathbb{R}^n \to \mathbb{R}^m$ definiujemy ogólnie jako:

$$\|M\| = \max_{\|x\|=1} \|Mx\|$$

A dla norm indukowanych z norm wektorowych jako:

$$\|M\|_p = \max_{\|x\|!=0} \frac{\|Mx\|_p}{\|x\|_p}$$

Przykładowe często stosowane normy:

- $\|M\|_1$ – Norma indukowana jedynkowa
- $\|M\|_\infty$ – Norma indukowana nieskończoność
- $\|M\|_2$ – Norma indukowana dwójkowa (norma spektralna)
- $\|M\|_p$ – Norma indukowana przez p

2.2. Współczynnik uwarunkowania

Współczynnik uwarunkowania (ang. *condition number*) mierzy wrażliwość rozwiązania układu równań liniowych na niewielkie zmiany danych wejściowych, czyli jak bardzo zmiana w b lub A wpływa na zmianę w rozwiązaniu x w równaniu $Ax = b$.

Dla odwracalnej macierzy A, współczynnik uwarunkowania względem wybranej normy definiujemy jako:

$$\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$$

Im większy współczynnik uwarunkowania, tym bardziej "niestabilna" numerycznie może być dana macierz. Jeśli $\mathrm{cond}(A) \approx 1$, układ jest dobrze uwarunkowany; jeśli bardzo duży, układ jest źle uwarunkowany.

Współczynnik uwarunkowania odgrywa kluczową rolę w analizie błędów i stabilności algorytmów numerycznych, szczególnie przy rozwiązywaniu układów równań oraz przy wyznaczaniu wartości własnych.

3. Normy macierzowe

Eksperymenty były wykonywane na macierzy M, gdzie:

$$M = \begin{bmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{bmatrix}$$

3.1. $\|M\|_1$ – norma kolumnowa

Algorytm:
Norma jedynkowa macierzy to maksymalna suma wartości bezwzględnych elementów w kolumnie. Dla każdej kolumny sumujemy moduły jej elementów, a następnie bierzemy maksimum z tych sum.

$$\|M\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{ij}|$$

Kod:

```
def norm_1(M):
    n = len(M[0])
    return max(
            sum(abs(M[i][j]) for i in range(n)
        ) for j in range(n)
    )
```

- abs(M[i][j]) – wartość bezwzględna elementu macierzy.

- Zagnieżdżone pętle sumują elementy kolumn, a następnie wybierają największą.

3.2. $\|M\|_\infty$ – norma wierszowa

Algorytm:
Norma nieskończoność macierzy to maksymalna suma wartości bezwzględnych elementów w wierszu. Podobnie jak wcześniej, sumujemy moduły, tym razem po wierszach.

$$\|M\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|$$

Kod:

```
def norm_inf(M):
    n = len(M[0])
    return max(
            sum(abs(M[i][j]) for j in range(ln))
        ) for i in range(n)
    )
```

- Pętla iteruje po wierszach i sumuje ich wartości bezwzględne.

3.3. $\|M\|_2$ – norma spektralna

Algorytm:
Norma dwójkowa macierzy to największa wartość własna (moduł) macierzy A. W przypadku macierzy ogólnej (niekoniecznie hermitowskiej), korzysta się z wartości własnych.

$$\|M\|_2 = \lambda_{\max}(A)$$

Kod:

```
def norm_2(M):
    return max(abs(np.linalg.eigvals(M)))
```

- np.linalg.eigvals(M) – zwraca wartości własne macierzy.

- Bierzemy największy moduł.

3.4. $\|M\|_p$ – aproksymacja normy p

Norma operatorowa $\|A\|_p$ dla macierzy $A \in \mathbb{R}^{m\times n}$ jest zdefiniowana jako:

$$\|A\|_p = \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p$$

W praktyce oznacza to maksymalne wzmocnienie, jakie macierz A może wprowadzić na dowolnym wektorze wejściowym w sensie normy p.

Definicja normy operatorowej opiera się na naturalnym pytaniu: jak bardzo A może "rozciągnąć" dowolny wektor? Szukamy więc maksymalnej wartości $\|Ax\|_p$ przy założeniu, że $\|x\|_p = 1$.

Wzór:

$$\|A\|_p = \max_{\|x\|_p=1} \|Ax\|_p$$

można interpretować jako ekstremum funkcji $\|Ax\|_p$ na jednostkowej kuli w normie p. Dla ogólnego p (np. $p = 1.5$, $p = 4$) nie istnieje zamknięta forma tej normy, dlatego stosuje się metody numeryczne, takie jak:

- generowanie losowych wektorów x z rozkładu normalnego,
- normalizacja ich do długości 1 w normie p,
- obliczanie $\|Ax\|_p$ i wybieranie maksymalnej z wartości.

Choć nie daje to dokładnej wartości, pozwala uzyskać dobrą **aproksymację** tej normy przy odpowiednio dużej liczbie prób (np. 10 000). Jest to metoda Monte Carlo dla estymacji wartości ekstremalnych funkcji.

Uwaga praktyczna

W zastosowaniach numerycznych dokładne obliczanie $\|A\|_p$ może być bardzo kosztowne obliczeniowo (szczególnie dla dużych macierzy i nietypowych wartości p), dlatego podejścia przybliżone są powszechnie akceptowane i używane np. w optymalizacji, analizie układów równań czy uczeniu maszynowym.

Algorytm: Metoda Monte Carlo – losowanie wielu wektorów, normalizacja w normie p, obliczenie wartości $\|Mx\|_p$ i wybór największej wartości .

Kod:

```
def normalize_vector_p(x, p):
    norm = np.sum(np.abs(x) ** p) ** (1 / p)
    return x / norm

def norm_p(M, p, samples=10000, seed=42):
    np.random.seed(seed)
    max_norm = 0
    n = M.shape[1]
    for _ in range(samples):
        x = np.random.randn(n)
        x = normalize_vector_p(x, p)
        Mx = M @ x
        Mx_norm = np.sum(np.abs(Mx) ** p) ** (1 / p)
        if Mx_norm > max_norm:
            max_norm = Mx_norm
    return max_norm
```

3.5. Wyniki

Dla zaimplementowanych algorytmów:

```
Normy macierzowe:
‖M‖₁ = 15
‖M‖₂ = 15.000000000000002
‖M‖ₚ (p=3) = 14.995541552860754
‖M‖∞ = 15
```

Figure 1: Wartości norm

Dla porównania wartości z biblioteki numpy:

```
Normy macierzowe:
‖M‖₁ = 15.0
‖M‖₂ = 15.000000000000002
‖M‖ₚ (p=3), ta wartość normy nie jest obsługiwna
‖M‖∞ = 15.0
```

Figure 2: Wartości norm

4. Współczynnik uwarunkowania

4.1. $\mathbf{cond}_1$

$$\mathrm{cond}_1(M) = \|M\|_1 \cdot \|M^{-1}\|_1$$

```
def cond_1(M):
    return norm_1(M) * norm_1(inverse(M))
```

4.2. $\mathbf{cond}_2$

$$\mathrm{cond}_2(M) = \|M\|_2 \cdot \|M^{-1}\|_2$$

```
def cond_2(M):
    return norm_2(M) * norm_2(inverse(M))
```

4.3. $\mathbf{cond}_\infty$

$$\mathrm{cond}_\infty(M) = \|M\|_\infty \cdot \|M^{-1}\|_\infty$$

```
def cond_inf(M):
    return norm_inf(M) * norm_inf(inverse(M))
```

4.4. $\mathbf{cond}_p$

$$\mathrm{cond}_p(M) = \|M\|_p \cdot \|M^{-1}\|_p$$

```
def cond_p(M, p):
    return norm_p(M, p) * norm_p(inverse(M), p)
```

4.5. Wyniki

Dla zaimplementowanych algorytmów:

Współczynniki uwarunkowania:
$\mathrm{cond}_1(M)$ = 5.333333333333333
$\mathrm{cond}_2(M)$ = 3.0618621784789704
$\mathrm{cond}_p(M)$ (p=3) = 4.502777545487174
$\mathrm{cond}\infty(M)$ = 5.333333333333333

Figure 3: Wartości norm

Dla porównania wartości z biblioteki numpy:

Współczynniki uwarunkowania macierzowego:
$\mathrm{cond}_1(M)$ = 5.333333333333333
$\mathrm{cond}_2(M)$ = 4.330127018922192
$\mathrm{cond}_p(M)$ (p=3)), ta wartość normy nie jest obsługiwna
$\mathrm{cond}\infty(M)$ = 5.333333333333333

Figure 4: Wartości norm

5. Podsumowanie

Normy macierzowe zostały zaimplementowane poprawnie i nie różnią się od tych które można uzyskać z biblioteki numpy. Jednak dla współczynników uwarunkowania występuje delikatna równica dla normy $\|M\|_2$ (tj. normy spektralnej).