Sprawozdanie z zadania numerycznego NUM2

Paweł Nykiel

Październik 2024

1 Polecenie

Zadane są macierze

$$A_1 = \begin{pmatrix} 5.8267103432 & 1.0419816676 & 0.4517861296 & -0.2246976350 & 0.7150286064 \\ 1.0419816676 & 5.8150823499 & -0.8642832971 & 0.6610711416 & -0.3874139415 \\ 0.4517861296 & -0.8642832971 & 1.5136472691 & -0.8512078774 & 0.6771688230 \\ -0.2246976350 & 0.6610711416 & -0.8512078774 & 5.3014166511 & 0.5228116055 \\ 0.7150286064 & -0.3874139415 & 0.6771688230 & 0.5228116055 & 3.5431433879 \end{pmatrix}$$

oraz

$$A_2 = \begin{pmatrix} 5.4763986379 & 1.6846933459 & 0.3136661779 & -1.0597154562 & 0.0083249547 \\ 1.6846933459 & 4.6359087874 & -0.6108766748 & 2.1930659258 & 0.9091647433 \\ 0.3136661779 & -0.6108766748 & 1.4591897081 & -1.1804364456 & 0.3985316185 \\ -1.0597154562 & 2.1930659258 & -1.1804364456 & 3.3110327980 & -1.1617171573 \\ 0.0083249547 & 0.9091647433 & 0.3985316185 & -1.1617171573 & 2.1174700695 \end{pmatrix}$$

Zdefiniujmy wektor

$$b \equiv (-2.8634904630, -4.8216733374, -4.2958468309, -0.0877703331, -2.0223464006)^T$$

Używając wybranego pakietu algebry komputerowej lub biblioteki numerycznej, rozwiąż równania macierzowe $A_i y = b$ dla $i = 1, 2$. Ponadto, rozwiąż analogiczne równania z zaburzonym wektorem wyrazów wolnych, $A_i y = b + \Delta b$. Zaburzenie Δb wygeneruj jako losowy wektor o małej normie euklidesowej (np. $\|\Delta b\|_2 \approx 10^{-6}$). Przeanalizuj jak wyniki dla macierzy A_1 i A_2 zależą od Δb i zinterpretuj zaobserwowane różnice.

2 Cel ćwiczenia

Celem ćwiczenia jest zbadanie wrażliwości rozwiązań układów równań macierzowych na małe zaburzenia wektora wyrazów wolnych. Analiza wyników dla macierzy A_1 i A_2 pozwoli na porównanie stabilności obu układów. Czułość układów równań na zmiany w wektorze b może być w obliczeniach numerycznych.

3 Program

Obliczenia zostały wykonane w języku Python przy użyciu biblioteki `numpy` do operacji macierzowych oraz do generowania losowego wektora zaburzeń.

3.1 Kod programu

```python
import numpy as np

# Definiowanie macierzy A1 i A2
A1 = np.array([
    [5.8267103432, 1.0419816676, 0.4517861296, -0.2246976350, 0.7150286064],
    [1.0419816676, 5.8150823499, -0.8642832971, 0.6610711416, -0.3874139415],
    [0.4517861296, -0.8642832971, 1.5136472691, -0.8512078774, 0.6771688230],
    [-0.2246976350, 0.6610711416, -0.8512078774, 5.3014166511, 0.5228116055],
    [0.7150286064, -0.3874139415, 0.6771688230, 0.5228116055, 3.5431433879]
])

A2 = np.array([
    [5.4763986379, 1.6846933459, 0.3136661779, -1.0597154562, 0.0083249547],
    [1.6846933459, 4.6359087874, -0.6108766748, 2.1930659258, 0.9091647433],
    [0.3136661779, -0.6108766748, 1.4591897081, -1.1804364456, 0.3985316185],
    [-1.0597154562, 2.1930659258, -1.1804364456, 3.3110327980, -1.1617171573],
    [0.0083249547, 0.9091647433, 0.3985316185, -1.1617171573, 2.1174700695]
])

# Definiowanie wektora b
b = np.array([-2.8634904630, -4.8216733374, -4.2958468309, -0.0877703331, -2.0223464006])

# Rozwiązywanie układu równań A1 * y1 = b oraz A2 * y2 = b
y1 = np.linalg.solve(A1, b)
y2 = np.linalg.solve(A2, b)

# Wyświetlanie wyników
print("Rozwiązanie dla A1 * y1 = b:")
print(y1)

print("\nRozwiązanie dla A2 * y2 = b:")
print(y2)

# Wylosowanie wektora o losowych wartościach
n = len(b)  # rozmiar wektora
v = np.random.randn(n)  # generowanie wektora o rozmiarze n z rozkładem normalnym

```

```
# Obliczanie normy euklidesowej
norma_v = np.linalg.norm(v)

# Przeskalowanie wektora do normy około 10^-6
v_scaled = v * (1e-6 / norma_v)
# Wektor zaburzeń (b +   b )
b_zaburzone = b + v_scaled
# Rozwiązywanie układów równań A1 * y1 = b +   b  oraz A2 * y2 = b
    +   b
y1_zaburzone = np.linalg.solve(A1, b_zaburzone)
y2_zaburzone = np.linalg.solve(A2, b_zaburzone)

print("\n␣␣␣Wektor␣zaburzeń␣( b ):")
print(v_scaled)

print("\nWektor␣b␣+␣ b :")
print(b_zaburzone)

print("\nRozwiązanie␣dla␣A1␣*␣y1␣=␣b␣+␣ b :")
print(y1_zaburzone)

print("\nRozwiązanie␣dla␣A2␣*␣y2␣=␣b␣+␣ b :")
print(y2_zaburzone, end="\n\n\n")
```

4 Wyniki

Wyniki dla macierzy A_1 i A_2 przed i po dodaniu zaburzeń pokazują różnice w czułości obliczeń.

- **Rozwiązanie dla A_1:**
  - Przed zaburzeniem: y_1
  - Po zaburzeniu: $y_1 + \Delta b$
- **Rozwiązanie dla A_2:**
  - Przed zaburzeniem: y_2
  - Po zaburzeniu: $y_2 + \Delta b$

4.1 Rezultaty działania programu

- Rozwiązanie dla $A_1 * y_1 = b$:
  $y_1 =$[0.02556195, -1.35714283, -3.94075752, -0.48893629, 0.10097805]

- Rozwiązanie dla $A_2 * y_2 = b$:
  $y_2 =$[-0.40875861, -0.56030137, -4.11200026, -1.52420117, -0.77520141]

- Dla losowo wygenerowanego wektora zaburzeń Δb
  $\Delta b =$[7.38480937e^{-7}, $5.04284540e^{-7}, -2.48422035e^{-7}, 6.57309714e^{-8}, 3.66481842e^{-7}$]

- Wektor $b + \Delta b$:
  $b + \Delta b =$[-2.86348972, -4.82167283, -4.29584708, -0.08777027, -2.02234603]

- Rozwiązanie dla $A_1 * y_1 = b + \Delta b$:
  $y_1 =$ [0.02556207, -1.35714279, -3.94075779, -0.48893634, 0.10097819]

- Rozwiązanie dla $A_2 * y_2 = b + \Delta b$:
  $y_2 =$ [712.13575595, -1307.85512989, 276.82812361, 1696.92708915, 1436.68122035]

4.2 Analiza - stabilność rozwiązania

- Dla macierzy A_1, rozwiązanie y_1 nieznacznie zmienia się po dodaniu zaburzeń do wektora b. Wartości y_1 przed i po perturbacji są zbliżone, co sugeruje, że układ równania jest stabilny na niewielkie zmiany w b.
- Natomiast w przypadku macierzy A_2, rozwiązanie y_2 wykazuje znaczne różnice po dodaniu zaburzeń. Wartości y_2 są znacznie bardziej wrażliwe na zmiany w b, co sugeruje, że ten układ jest mniej stabilny.

Macierz A_1 wykazała większą stabilność na zaburzenia niż A_2.

5 Wnioski

Różne układy równań mogą wykazywać odmienną wrażliwość na zaburzenia wyrazów wolnych. W tym przypadku macierz A_1 okazała się bardziej stabilna niż A_2, co sugeruje, że jest lepiej uwarunkowana numerycznie.