

Metody Numeryczne 2

Laboratorium 5

Obliczanie wskaźnika uwarunkowania macierzy
 $A = A^T$ i diagonalnie silnie dominującej

Szymon Adach

16 grudnia 2015

1 Treść zadania

Zadanie 6: Obliczanie wskaźnika uwarunkowania macierzy $A = A^T$ i diagonalnie silnie dominującej ($cond_2(A) = \frac{\lambda_{max}}{\lambda_{min}}$). Do obliczenia λ_{max} należy zastosować zwykłą metodę potęgową, a do obliczenia λ_{min} odwrotną metodę potęgową. Odpowiedni układ równań należy rozwiązać metodą iteracyjną Jacobiego.

2 Opis metody

λ_{max} obliczana jest za pomocą prostej metody potęgowej. Aby wyeliminować nadmiar bądź niedomiar zastosowano normowanie wektora x . Algorytm znajdowania tej wartości własnej macierzy A przedstawia się następująco:

$$\begin{cases} x^{(0)} \neq 0 \\ x^{(k+1)} = A \cdot x^{(k)} \text{ dla } k = 0, 1 \dots n \end{cases}$$

Ponadto w każdej iteracji wykonywane jest normowanie wektora x :

$$x^{(k+1)} = \frac{x^{(k+1)}}{\|x^{(k+1)}\|_2}$$

Wektor $x^{(0)}$ jest losowany podczas uruchamiania programu. Końcowy wynik obliczany jest ze wzoru:

$$\lambda_{max} = \frac{(Ax, x)}{\|x^{(k+1)}\|_2^2}$$

λ_{min} obliczana jest za pomocą odwrotnej metody potęgowej. Algorytm znajdowania tej wartości własnej macierzy A przedstawia się następująco:

$$\begin{cases} \lambda^* \approx \lambda_i \\ x^{(0)} \neq 0 \\ (A - \lambda^* I)y^{(k+1)} = x^{(k)} \\ \lambda_i^{(k)} = \frac{1}{(y^{(k+1)}, x^{(k)})} - \lambda^* \\ x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|_2} \end{cases}$$

Wektor $x^{(0)}$ jest losowany podczas uruchamiania programu. Równanie $(A - \lambda^* I)y^{(k+1)} = x^{(k)}$ rozwiązywane jest za pomocą metody iteracyjnej Jacobiego z zadaną dokładnością.

3 Działanie programu

Program jest uruchamiany poleceniem:

conditional_number(A, n, e)

- **A** - macierz, której wskaźnik uwarunkowania zostanie obliczony
- **n** - liczba iteracji w metodach potęgowej i odwrotnej potęgowej
- **e** - błąd bezwzględny metody Jacobiego rozwiązywania równania $(A - \lambda^* I)y^{(k+1)} = x^{(k)}$ w odwrotnej metodzie potęgowej

Po wykonaniu obliczeń prezentowane są wartości λ_{max} , λ_{min} oraz $cond_2(A)$:

Lambda_min =

Lambda_max =

cond_2(A) =

4 Przykłady

1. **Wywołanie:**

$$A = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}$$

condition_number(A, 100, 0.001)

Wyjście:

Lambda_min = 2.588539

Lambda_max = 5.414214

cond_2(A) = 2.091610

eig(A) = 2.5858 4.0000 5.4142

Obserwacja: Dla tego problemu podobny wpływ na dokładność programu ma zwiększenie liczby iteracji oraz zmniejszenie dopuszczalnego błędu metody Jacobiego.

2. **Wywołanie:**

$$A = \begin{bmatrix} 8 & 3 & 2 \\ 3 & 9 & 1 \\ 2 & 1 & 4 \end{bmatrix}$$

```
condition_number(A, 1000, 0.0001)
```

Wyjście:

```
Lambda_min = 3.159160
```

```
Lambda_max = 12.072603
```

```
cond_2(A) = 3.821460
```

```
eig(A) = 3.159273139447023 5.768124085466175 12.072602775086800
```

3. **Wywołanie:**

$$A = \begin{bmatrix} 10 & 2 & -2 & 3 \\ 2 & -8 & 1 & -4 \\ -2 & 1 & 7 & -1 \\ 3 & -4 & -1 & -15 \end{bmatrix}$$

```
condition_number(A, 100000, 0.00001)
```

Wyjście:

```
Lambda_min = 6.220290
```

```
Lambda_max = -17.281854
```

```
cond_2(A) = -2.778304
```

```
eig(A) = -17.281853841873332 -6.352788379417891
```

```
6.220274282228035 11.414367939063178
```

Obserwacja: Prezentowana metoda znajduje wartości własne najmniejsze i największe co do modułu, zatem obliczony wskaźnik uwarunkowania w pewnych przypadkach nie być poprawny.