

Metody Numeryczne

Projekt 2 – Układy równań liniowych

dr hab. inż. Grzegorz Fotyga, prof. PG

24 marca 2025

1. Wstęp

Celem projektu jest implementacja i analiza dwóch metod iteracyjnych (Jacobiego i Gaussa-Seidla) oraz jednej metody bezpośredniej (faktoryzacja LU) rozwiązywania układów równań liniowych. Testy poszczególnych metod będą przeprowadzane na układach równań, które mogą powstać w wyniku dyskretyzacji równań różniczkowych i są powszechnie stosowane w takich zagadnieniach jak: elektronika, elektrodynamika, mechanika (zastosowania lotnicze, biomechanika, motoryzacja), badanie wytrzymałości materiałów i konstrukcji, symulacje odkształceń, naprężeń, przemieszczeń i drgań, akustyka, fotonika, termodynamika, dynamika płynów i wiele innych.

W rzeczywistych problemach rozwiązywane są układy równań zawierające setki milionów niewiadomych, dla których obliczenia trwają często wiele godzin, a nawet dni, mimo wykorzystywania najnowszych superkomputerów. Opracowanie nowych efektywnych metod rozwiązań (dostosowanych do współczesnych architektur komputerowych) jest dużym wyzwaniem zarówno z punktu widzenia matematyki, jak i informatyki. Jest ono przedmiotem badań wielu ośrodków naukowych, ponieważ bez niego rozwój wymienionych wyżej dziedzin wiedzy byłby **niemożliwy**.

W praktyce najczęściej stosuje się tak zwany rzadki format przechowywania macierzy (który przechowuje tylko wartości niezerowe i ich położenie w macierzy), ponieważ zdecydowana większość elementów ma wartość 0. Jednak ze względu na prostotę, w ramach tego projektu domyślnie będzie wykorzystywany tzw. format pełny (przechowujący wszystkie wartości, również 0), który może być stosowany w problemach zawierających zazwyczaj nie więcej niż kilka tysięcy niewiadomych. Mimo, że testowane będą jedynie podstawowe metody rozwiązań, wykonanie projektu będzie dobrym fundamentem do poznania bardziej zaawansowanych metod iteracyjnych (np. metody gradientów sprzężonych, GMRES, QMR itp.).

2. Konstrukcja układu równań

Układ równań liniowych ma następującą postać:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (1)$$

gdzie \mathbf{A} jest macierzą systemową¹, \mathbf{b} jest wektorem pobudzenia², natomiast \mathbf{x} jest wektorem rozwiązań reprezentującym szukaną wielkość fizyczną³.

- Na potrzeby projektu należy przyjąć, że \mathbf{A} jest tzw. macierzą pasmową o rozmiarze $N \times N$ i zdefiniowaną w (2), gdzie N ma wartość $1200 + 10c + d$, c jest przedostatnią cyfrą numeru Twojego indeksu, natomiast d ostatnią (np. dla indeksu 102263 $N = 1263$). Macierz \mathbf{A} zawiera więc pięć diagonali - główna z elementami a_1 , dwie sąsiednie z elementami a_2 i dwie skrajne diagonale z elementami a_3 .
- Prawa strona równania to wektor \mathbf{b} o długości N .
- W wyniku rozwiązania układu równań (1) otrzymujemy wektor \mathbf{x} .

¹W zależności od problemu może ona reprezentować np. obwód elektroniczny, geometrię sali koncertowej, turbinę, karoserię samochodu itp.

²np. impuls elektroniczny, wektor siły, fala dźwiękowa itp.

³np. rozkład pola elektromagnetycznego, natężenie dźwięku itp.

$$\mathbf{A} = \begin{bmatrix} a1 & a2 & a3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a2 & a1 & a2 & a3 & 0 & 0 & 0 & \dots & 0 \\ a3 & a2 & a1 & a2 & a3 & 0 & 0 & \dots & 0 \\ 0 & a3 & a2 & a1 & a2 & a3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & a3 & a2 & a1 \end{bmatrix} \quad (2)$$

3. Wektor residuum

Ważnym elementem algorytmów iteracyjnych (np. Jacobiego i Gaussa-Seidla) jest określenie w której iteracji algorytm powinien się zatrzymać. W tym celu najczęściej korzysta się z residuum [1], czyli wektora który dla k – tej iteracji przyjmuje postać:

$$\mathbf{r}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}. \quad (3)$$

Badając normę euklidesową residuum ($norm(\mathbf{r}^{(k)})$), możemy w każdej iteracji algorytmu obliczyć jaki błąd wnosi wektor $\mathbf{x}^{(k)}$. Jeżeli algorytm zbiegnie się do dokładnego rozwiązania, to residuum stanowić będzie wektor zerowy. Ze względu na to, że metody iteracyjne praktycznie nigdy nie generują wektora residuum równego zero, podstawowe kryterium zakończenia obliczeń określone jest jako osiągnięcie normy residuum mniejszej niż zadana wartość, np. 10^{-6} .

Residuum nazywane jest również wektorem reszt [2] lub wektorem residualnym [3]. W literaturze anglojęzycznej residuum określone jest jako *residual vector* lub *residual* [4].

4. Zadania

Sprawozdanie powinno zawierać m.in. analizę rezultatów osiągniętych w zadaniach **B**, **C**, **D**, **E** wraz z wykresami wskazanymi w opisie zadań.

- **Zadanie A** – Stwórz układ równań dla $a1 = 5 + e$, gdzie e jest czwartą cyfrą Twojego indeksu, $a2 = a3 = -1$. Rozmiar macierzy N zdefiniowano w punkcie 2 tej instrukcji. \mathbf{b} jest wektorem o długości N , którego n -ty element ma wartość $\sin(n \cdot (f + 1))$, gdzie f jest trzecią cyfrą Twojego indeksu. We wstępie sprawozdania opisz rozwiązywane równanie macierzowe. (5%)
- **Zadanie B** – Zaimplementuj metody iteracyjne rozwiązywania układów równań liniowych: Jacobiego i Gaussa-Seidla. Opisz ile iteracji potrzebuje każda z tych metod do wyznaczenia rozwiązania układu równań z zadania **A**, przy założeniu, że warunkiem zakończenia obliczeń jest osiągnięcie normy residuum mniejszej niż 10^{-9} . Dla obu metod przedstaw na **wykresie** jak zmienia się norma residuum w kolejnych iteracjach wykonywanych w celu wyznaczenia rozwiązania (oś y w **skali logarytmicznej**). Porównaj czasy działania obu algorytmów. (30%)
- **Zadanie C** – Zdefiniuj układ równań dla $a1 = 3$, $a2 = a3 = -1$, natomiast N i wektor \mathbf{b} określ zgodnie z treścią zadania **A**. Czy metody iteracyjne dla takich wartości elementów macierzy **A** zbiegają się? Dla obu metod przedstaw na **wykresie** jak zmienia się norma residuum w kolejnych iteracjach (oś y w **skali logarytmicznej**). (10%)
- **Zadanie D** – Zaimplementuj metodę bezpośredniego rozwiązywania układów równań liniowych: metodę rozkładu LU. Zastosuj tę implementację do wyznaczenia rozwiązania równania z zadania **C**. Ile wynosi norma residuum w tym przypadku? (30%)
- **Zadanie E** – Stwórz **dwa wykresy** zależności czasu wyznaczenia rozwiązania dla trzech badanych metod w zależności od liczby niewiadomych $N = \{100, 500, 1000, 2000, 3000 \dots\}$ dla macierzy opisanej w zadaniu **A**. Oba wykresy powinny ilustrować identyczne dane, przy czym pierwszy powinien posiadać liniową skalę osi Y, a drugi skalę logarytmiczną. (10%)
- **Zadanie F** – Zwięźle opisz swoje obserwacje po wykonaniu zadań **A–E**. (15%)

Program komputerowy opracowany w ramach tego projektu powinien zawierać samodzielnie napisaną implementację trzech badanych algorytmów, bez korzystania z zewnętrznych bibliotek (poza biblioteką `numpy`). Użycie `numpy` w obliczeniach numerycznych jest silnie rekomendowane ze względu na wysoką wydajność tej biblioteki, jednak nie jest obowiązkowe (operacje na macierzach można definiować z zastosowaniem np. funkcji `np.dot()` lub `np.matmul()`).

Zadanie bonusowe. Opracuj modyfikację dowolnego fragmentu badanych algorytmów, tak aby obliczenia były wykonywane wyłącznie na elementach niezerowych macierzy, oraz przeanalizuj wpływ tej zmiany na wydajność obliczeń. Zadanie bonusowe nie jest wymagane do uzyskania 10 punktów, lecz może stanowić dodatkowe źródło punktów w sytuacji, gdy podstawowa część sprawozdania zostanie oceniona na mniej niż 10 punktów.

Kody źródłowe w języku Python należy przesyłać wyłącznie jako pliki o rozszerzeniu `.py` – inne formaty, takie jak pliki Jupyter Notebook, nie będą akceptowane. Wskazane jest dołączenie pliku `requirements.txt` zawierającego listę pakietów wykorzystywanych w projekcie (można go wygenerować automatycznie, np. przy użyciu `pipreqs`), jednak nie stanowi to obowiązku. Ponadto, przesłane archiwum `.zip` nie powinno zawierać katalogu środowiska wirtualnego: `venv`.

Literatura

- [1] Bjorck A., Dahlquist G., *Metody numeryczne*, PWN, 1987
- [2] Fortuna Z., Macukow B., Wąsowski J., *Metody numeryczne*, PWN, 2017
- [3] Kincaid, Cheney, *Analiza numeryczna*, WNT, 2006
- [4] Saad, Yousef, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.