Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowania Komputerów 3

Temat: Biblioteka do numerycznego rozwiązywania układów równań liniowych

Autor	Piotr Hasiec
Prowadzący	Dr inż. Piotr Pecka
Rok akademicki	2020/2021
Kierunek	Informatyka
Rodzaj studiów	SSI
Semestr	3
Sekcja	2
Termin oddania sprawozdania	2020-

Treść zadania

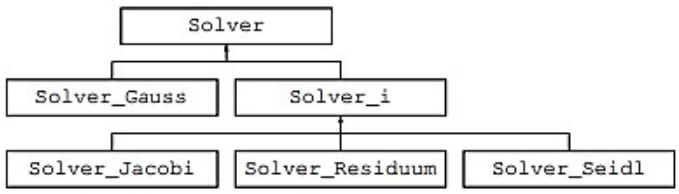
Napisać zbiór bibliotekę do numerycznego rozwiązywania układów równań linowych. Metody mają operować na dynamicznie alokowanych macierzach. Biblioteka ma implementować zarówno metody dokładnego rozwiązywania równań – metod eliminacji Gaussa, jak i iteracyjne – metoda Jacobiego i metoda Seidla. Zadanie ma być wykonane tak aby spełniać paradygmaty programowania obiektowego.

1. Analiza zadania

Z racji podobieństwa pomiędzy metodą Jacobiego, a metodą Seidla postanowiłem zaimplementować jeszcze dodatkowo metodę residuów.

1.1. Struktury danych

W zadaniu należy zaimplementować dynamiczną strukturę Macierzy – dwuwymiarowa dynamicznie alokowana tablica z metodami i operatorami pozwalającymi na jej obsługę oraz Solvery – obiekty operujące na macierzach i rozwiązujące układy równań.



Rysunek 1: Diagram dziedziczenia dla klasy Solver

1.2. Algorytmy

Algorytmy operują na układach równań w postaci macierzowej – macierz współczynników z dołączoną macierzą wyników.

Jeśli układ równań zapisany jest w postaci $A \times X = B$ to macierz A ma wymiary $n \times n$ gdzie n to liczba zmiennych, a macierz B $n \times 1$. Macierz dołączona na której operują solvery ma postać $[A|B]$ oraz ma wymiary $n \times (n+1)$ gdzie n to liczba zmiennych.

1.2.1. Metoda eliminacji Gaussa

Metoda Gaussa polega na sprowadzeniu układu równań (zapisanego w postaci macierzowej) do postaci macierzy trójkątnej. Do wykonania tej operacji służy potrójnie zagnieżdżona pętla. Zewnętrzna pętla wykonuje się n-1 razy za każdym przebiegiem eliminując i-tą zmienną w i+1 równaniu układu poprzez odjęcie i-tego równania przemnożonego przez odpowiedni współczynnik od równania (i+1)-ego.

1. Krok pętli

$$
\begin{aligned}
a_{00}x_0 + a_{01}x_1 + \cdots + a_{0n}x_n &= b_0 \\
a_{11}x_0 + a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_0 + a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\
. \quad\quad\quad ... \quad\quad\quad . \\
. \quad\quad\quad ... \quad\quad\quad . \\
. \quad\quad\quad ... \quad\quad\quad . \\
a_{n1}x_0 + a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

2. Krok pętli:

$$\begin{array}{l} a_{00}x_0 + a_{01}x_1 + \cdots + a_{0n}x_n = b_0 \\ 0 \quad + c_{11}x_1 + \cdots + c_{0n}x_n = b'_1 \\ 0 \quad + c_{21}x_1 + \cdots + c_{2n}x_n = b'_2 \\ . \qquad\qquad ... \qquad\qquad . \\ . \qquad\qquad ... \qquad\qquad . \\ . \qquad\qquad ... \qquad\qquad . \\ 0 \quad + c_{n1}x_1 + \cdots + c_{nn}x_n = b'_n \end{array}$$

3. Krok pętli:

$$\begin{array}{l} a_{00}x_0 + a_{01}x_1 + \cdots + a_{0n}x_n = b''_0 \\ 0 \quad + c_{11}x_1 + \cdots + c_{0n}x_n = b''_1 \\ 0 \quad + \quad 0 \quad + \cdots + d_{2n}x_n = b''_2 \\ . \qquad\qquad ... \qquad\qquad . \\ . \qquad\qquad ... \qquad\qquad . \\ . \qquad\qquad ... \qquad\qquad . \\ 0 \quad + \quad 0 \quad + \cdots + d_{nn}x_n = b''_n \end{array}$$

Eliminacja przebiega tak aż do momentu w którym w ostatnim równaniu zostanie tylko jedna niewiadoma. Ten etap ma złożoność $O(n) = \mathrm{n}^3$. Następnie należy wstawić zmienną wyliczoną z ostatniego równania do równania wyżej i wyliczyć kolejną zmienną, i tak dalej aż do uzyskania wszystkich zmiennych.

1.2.2. Metoda Jacobiego oraz Seidla

Metody te polegają na przekształceniu układu równań do postaci:

$$x_0^{k+1} = \frac{b_1 - (a_{01}x_1^k + \cdots + a_{0n}x_n^k)}{a_{00}}$$

$$x_1^{k+1} = \frac{b_2 - (a_{11}x_1^k + \cdots + a_{1n}x_n^k)}{a_{11}}$$

$$x_2^{k+1} = \frac{b_3 - (a_{21}x_1^k + \cdots + a_{2n}x_n^k)}{a_{22}}$$

$$\begin{array}{ccc} . & ... & . \\ . & ... & . \\ . & ... & . \end{array}$$

$$x_n^{k+1} = \frac{b_n - (a_{n1}x_1^k + \cdots + a_{nn}x_n^k)}{a_{nn}}$$

Gdzie x_j^k to przybliżenie x_j w k-tej iteracji. Zmienne x w zerowej iteracji mogą być losowe lecz zwykle przyjmuje się je jako 0.

W metodzie Jacobiego do wyliczenie wszystkich zmiennych w k-tej iteracji używa się tylko zmiennych z k-1 iteracji. W metodzie Seidla do wyliczenie wszystkich zmiennych w k-tej iteracji używa się także zmiennych wyliczonych już w tej (k-tej) iteracji. Metoda Seidla zwykle jest szybciej zbieżna niż metoda Jacobiego.

Jako warunek stopu w obu metodach przyjmuje się żeby maksymalna zmiana, co do modułu, pomiędzy dowolną zmienną w k-taj iteracji i k-1 iteracji była mniejsza od jakiejś ustalonej wartości.

Zbieżność obu metod zależy od współczynników zawartych w macierzy A. Aby układ równań był zbieżny dowolna z poniższych macierzy musi być mniejsza od 1:

$$\|A\|_I = max_i \sum_{j=0}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| < 1$$

$$\|A\|_{II} = max_j \sum_{i=0}^{n} \left|\frac{a_{ij}}{a_{jj}}\right| < 1$$

$$\|A\|_{III} = \sum_{i=0}^{n}\sum_{j=0}^{n} \left|\frac{a_{ij}}{a_{ii}}\right|^2 < 1$$

Złożoność wprowadzania poprawek do wszystkich zamiennych trakcie jednej iteracja pętli zewnętrznej jest kwadratowa względem ilości zmiennych, zatem złożoność całej metody wynosi:

$O(n) = n^2 \cdot I(A)$

Gdzie n jest liczbą zmiennych, a I(A) jest liczbą iteracji zależną od współczynników układu.

1.2.3. Metoda residuów

Metoda residuów opiera się o przekształcenie układu równań do postaci:

$$R^k = D - C \times X^k$$

Gdzie $c_{ij} = -\frac{a_{iij}}{a_{ii}}$, a $d_{i0} = \frac{b_{i0}}{a_{ii}}$

R^k i X^k to kolejno wektor residuów i wektor przybliżeń zmiennych w katej iteracji.

W każdej iteracji wybierane jest największe co do modułu residuum (niech będzie z s-tego równania) i zerowane. W równaniu s-tym zmienna x_s ma współczynnik $c_{ij} = -1$ zatem żeby wyzerować residuum należy dodać jego wartość do zmiennej x_s (i otrzymać nowe przybliżenie zmiennej x_s), a następnie wnieś poprawkę do każdego z pozostałych residuów, znów wybrać największe itd.

Obliczenia zakańczamy tak jak w poprzednich przypadkach jeśli największa zmiana pomiędzy dowolną zmienną w pewnej i kolejnej iteracji jest mniejsza od zadanej arbitralnie dokładności. Warunki zbieżności są również takie same jak w poprzednich metodach iteracyjnych.

Złożoność wybierania maksymalnego residuum i wprowadzania poprawek jest liniowa względem liczby zmiennych. Nie jesteśmy jednak w stanie przewidzieć ile iteracji nastąpi przed osiągnięciem zadanej dokładności. Złożoność algorytmu można szacować na $O(n) = n \cdot I(A)$

Gdzie n jest liczbą zmiennych, a I(A) jest liczbą iteracji zależną od współczynników układu.

2. Specyfikacja zewnętrzna

W treści zadani niewymagane było napinanie programu zatem napisany został tylko program sprawdzający czas rozwiązywania się losowo generowanych układów o zdanych wartościach liczbach zmiennych.

3. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem programowania obiektowego.

3.1. Ogólna struktura programu

Jeśli układ równań zapisany jest w postaci $A \times X = B$ to macierz A ma wymiary $n \times n$ gdzie n to liczba zmiennych, a macierz B $n \times 1$. Macierz dołączona na której operują solvery ma postać $[A|B]$ oraz wymiary $n \times (n+1)$ gdzie n to liczba zmiennych.

Metoda solv() klasy Solver jest typu bool i zwraca true w przypadku gdy układ jest rozwiązywalny i false w przypadku gdy układ jest nierozwiązywalny – sprzeczny tożsamościowy lub niezbieżny w przypadku metod iteracyjnych.

Solver przechowuje rozwiązania w wewnętrznej zmiennej typu Matrix i nie nadpisuje ich aż do ponownego użycia metody solv.

3.2. Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

4. Testowanie

Program został przetestowany na układach równań spełniających warunki zbieżności i nie spełniających. Maksymalna liczba zmiennych dla których program został przetestowany wynosi 1000 zmiennych o losowo wygenerowanych współczynnikach. Testy nie wykazały niepoprawnego działania programu szybkość metody Jacobiego jest mocno losowa i czasem rozwiązywanie trwa dłużej niż metodą Gaussa.

5. Wnioski

W przypadku dużych układów równań – ponad kilkaset zmiennych – efektywniejsze wydają się być iteracyjne metody rozwiązywania równań. Niestety metody te działają jedynie w określonych warunkach, tylko na układy spełniające warunki zbieżności, co ogranicza ich stosowalność. Na szczęście dzięki odpowiednim przekształceniom liniowym można sprowadzić układy niezbieżne do postaci zbieżnej. Dzięki temu jesteśmy, choć niestety w przybliżeniu, szybko rozwiązywać duże układy równań.

Źródła

[1] Jerzy Klamka, Zbigniew Ogonowski „Metody numeryczne”

[2] http://www.algorytm.org/procedury-numeryczne/