

Wydział EiTI PW

Dokumentacja

Projekt 2.2

Paweł Maśluch, nr albumu 262955

Treść projektu

Napisać program do rozwiązywania układu N (np.5) równań liniowych metodą wyznaczników. Program powinien uwzględniać przypadki szczególne i odpowiednio je identyfikować oraz umożliwić edycję wprowadzonych danych po dokonaniu obliczeń i wyświetleniu rozwiązania.

Uwagi:

- Program powinien posiadać czytelny i wygodny w obsłudze interfejs użytkownika.
- Maksymalna ilość równań N musi być zdefiniowana w programie jako stała – nie ma konieczności wykorzystywania dynamicznej alokacji pamięci.
- Program powinien sprawdzać poprawność wprowadzanych danych i informować użytkownika o popełnianych błędach podczas wprowadzania owych danych
- Program powinien być napisany tak, aby wprowadzenie niepoprawnych danych nie powodowało jego niekontrolowanego zakończenia

Dodatkowe uwagi:

- Program **nie powinien** zakładać poprawności wprowadzanych danych
- Nieodłączną częścią programu jest dokumentacja a z kolei jej najważniejszym elementem jest obrazek (rysunek) przedstawiający algorytm działania programu (pozostałe szczegóły dotyczące dokumentacji opisane są w pliku „dokumentacja.pdf”)

Opis programu

Stałe globalne

W programie wykorzystuję dokładnie 1 stałą globalną, mianowicie `MAX_N`. Stała ta jest typu `int` oraz wynosi 5. Oznacza ona maksymalną liczbę równań, którą będziemy chcieli rozwiązać.

Zmienne globalne

W programie nie używam zmiennych globalnych.

Funkcje i procedury

W programie wykorzystuję poniższe funkcje oraz procedury:

- a) double det(int n, int w, int * stare_kolumny, double ** A)
Jest to funkcja zwracająca wyznacznik (jako liczbę zmiennoprzecinkową) macierzy A. Funkcja ta korzysta z metody rozwinięcia Laplace'a. Macierz A jest rozmiaru $n \times n$. Zakładamy, że rozwijamy wyznacznik względem wiersza o numerze w (dodatkowo, do obliczenia wyznacznika nie uwzględniamy wierszy o numerach mniejszych niż w). Dalej, zbiór numerów kolumn, uwzględnianych przy wyliczeniu wyznacznika, znajduje się w tablicy stare_kolumny (tablica stare_kolumny ma n komórek).
- b) void podmiana(double ** A, int j, double * b, int n)
Jest to procedura dokonująca zamiany miejscami j-tej kolumny macierzy A, oraz tablicy b. Macierz A ma wymiar $n \times n$, natomiast tablica b ma n komórek.
- c) int Get_Int(int a, int b)
Jest to funkcja, której zadaniem jest pobranie od użytkownika liczby całkowitej, mieszczącej się w przedziale $[a, b]$, gdzie $a \leq b$. Funkcja ta zwraca poprawnie podaną liczbę całkowitą.
- d) double Get_Double()
Jest to funkcja, której zadaniem jest pobranie od użytkownika liczby zmiennoprzecinkowej. Funkcja ta zwraca poprawnie podaną liczbę zmiennoprzecinkową.
- e) int main()
Jest to funkcja główna całego programu, realizująca projekt. Dalej podaję skrócony opis działania tej funkcji. Najpierw tworzymy dynamiczne struktury danych. Dopóki użytkownik ma zamiar rozwiązywać układy równań (przy czym rozwiążemy co najmniej 1 układ równań), dopóty realizowana jest funkcja main. Za każdym razem podajemy liczbę równań, współczynniki każdego równania oraz wyrazy wolne. Dalej, do określenia, z jakim układem równań mamy do czynienia, oraz ewentualnego wyznaczenia rozwiązania, korzystamy z twierdzenia, zwanego wzorami Cramera. Kiedy tylko użytkownik nie ma zamiaru dalej rozwiązywać układów równań, usuwamy dynamiczne struktury danych, a następnie kończymy działanie funkcji main, zwracając wartość 0.