POLITECHNIKA KRAKOWSKA  
2014

Kalkulator macierzy

Mateusz Bryzik  
Mateusz Brycki

Przedmiot: Metody obliczeniowe

Prowadzący: Anna Romańska

1. Wstęp

Program o nazwie „Kalkulator macierzy” został wykonany na potrzeby przedmiotu Metody obliczeniowe w roku 2014.

Podstawowymi zadaniami kalkulatora jest wykonywanie następujących działań:

1. dodawanie macierzy
2. mnożenie macierzy
3. obliczanie wyznacznika macierzy
4. generowanie macierzy odwrotnej
5. generowanie macierzy transponowanej
6. Zastosowane metody obliczeń – opis algorytmów
7. Dodawanie macierzy

Przy dodawaniu macierzy użyty został najprostszy algorytm dodawania kolejnych odpowiadających sobie elementów macierzy A i B (przy założeniu, że obie dodawane macierze mają takie same wymiary).

1. Mnożenie macierzy

Mnożenie macierzy odbywa się poprzez sumowanie kolejnych ilorazów odpowiednich elementów macierzy.

//pętla odpowiedzialna za wykonywanie mnożenia poszczególnych elementów macierzy

for (int i = 0; i < first->y; i++)

{

for(int j = 0; j < second->x; j++)

{

int s = 0; //zmienna pomocnicza na sumy obliczane w kolejnych krokach

for(int k = 0; k < first->x; k++)

{

s += first->table[i][k] \* second->table[k][j];

}

C->table[i][j] = s; //przypisanie tablicy wynikowej wartości

}

}

Dokładniejszy opis metody można znaleźć w programie, w zakładce Podstawy teoretyczne.

1. Obliczanie wyznacznika macierzy

W obliczaniu wyznacznika macierzy użyliśmy metody rozkładu LU. Po pierwsze zauważmy, iż wyznacznik macierzy trójkątnej jest równy iloczynowi elementów leżących na głównej przekątnej. Można to bardzo prosto wyprowadzić z rozwinięcia Laplace'a.   
  
  
  
  
Pomiędzy macierzami A, L i U zachodzą odpowiednie zależności:

***A*** = ***L*** × ***U***det ***A*** = det(***L*** × ***U***)  
det ***A*** = det ***L*** × det ***U***

det ***L*** = 1, ponieważ jest macierzą trójkątną, a wszystkie elementy przekątnej są równe 1

det ***A*** = 1 × det ***U***

det ***A*** = det ***U***

Zatem w celu obliczenia wyznacznika macierzy A wystarczy dokonać jej rozkładu LU i obliczyć iloczyn elementów leżących na przekątnej macierzy U.

if(LU(temp)) //dokonujemy rozkładu LU macierzy temp, jeżeli się powiódł

{

C->simple\_result = temp->table[0][0]; //wyznacznik ustawiamy na element [0][0] macierzy

for(int i = 1; i < temp->x; i++) //w kolejnych krokach obliczamy wyznacznik

{

C->simple\_result \*= temp->table[i][i]; //zapisujemy wyznacznik macierzy do wyniku

}

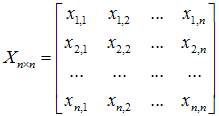
}

1. Generowanie macierzy odwrotnej

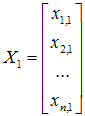
Oznaczmy X = A-1. Wtedy powyższe równanie możemy zapisać jako:

A × X = I

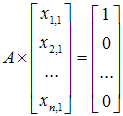
Niech:



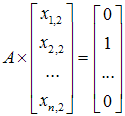
Weźmy następnie pierwszą kolumnę macierzy X:



Zgodnie z regułami mnożenia macierzy mamy:



W wyniku otrzymujemy pierwszą kolumnę macierzy jednostkowej, gdyż z założenia X = A-1. Wyrazy kolumny X1 możemy potraktować jako niewiadome, a cały iloczyn jako układ n równań liniowych względem tych niewiadomych. Układ ten rozwiązujemy metodą LU, którą opisaliśmy w [poprzednim rozdziale](http://edu.i-lo.tarnow.pl/inf/alg/001_search/0080.php). Następnie tak samo postępujemy z drugą kolumną X2 (rozkład LU wykonujemy tylko jeden raz na początku obliczeń – później wykorzystujemy go do rozwiązania kolejnych układów równań).



Postępując podobnie dla pozostałych kolumn X3, X4 ... Xn wyznaczymy całą macierz X, która jest odwrotnością macierzy A. W algorytmie macierz X może jednocześnie być macierzą jednostkową, której kolejne kolumny przekształcamy w kolumny Xi. Macierz jednostkowa na końcu zostanie więc zastąpiona macierzą odwrotną do A.

1. Generowanie macierzy transponowanej

Macierz transponowana jest generowana na podstawie zamiany wartości kolumn z wartościami wierszy macierzy pierwotnej.

for(int y = 0; y < first->x; y++)

{

for(int x = 0; x < first->y; x++)

{

C->table[y][x] = first->table[x][y];

}

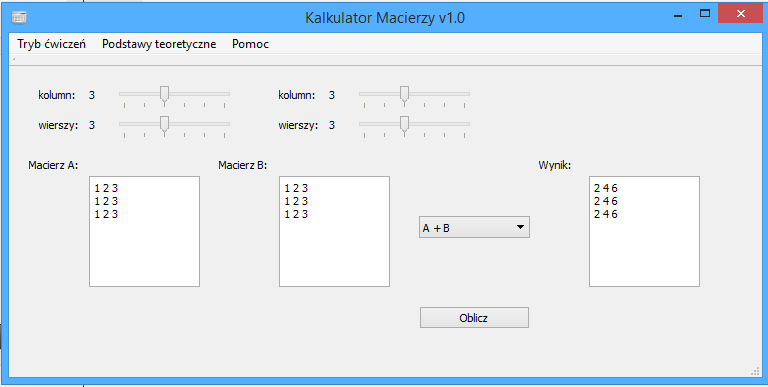
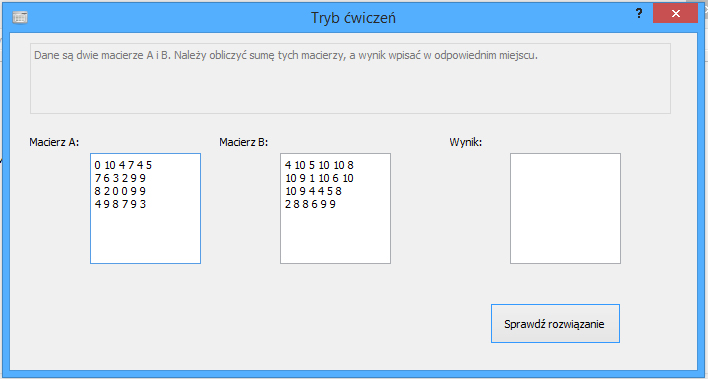
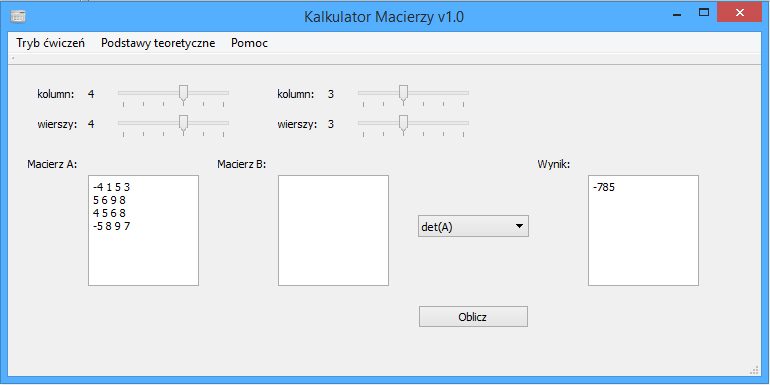
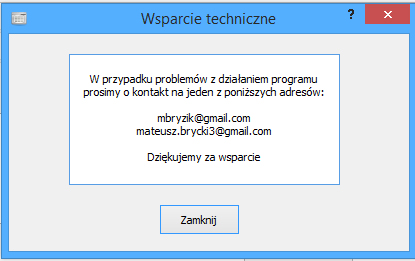
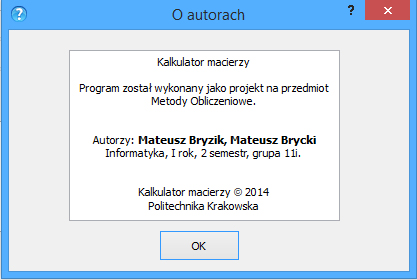
}

1. Opis bibliotek zastosowanych do tworzenia interfejsu użytkownika.

Cały interfejs graficzny został utworzony w bibliotekach Qt w wersji 5.2.1. Kompilacja projektu została uruchomiona w narzędziu o nazwie QtCreator w wersji 3.0.1, działającym na bibliotekach Qt 5.2.1

Twórcą Qt jest obecnie organizacja [Qt Project](http://pl.wikipedia.org/w/index.php?title=Qt_Project&action=edit&redlink=1" \o "Qt Project (strona nie istnieje)), do której oprócz niezależnych twórców [Open Source](http://pl.wikipedia.org/wiki/Otwarte_oprogramowanie) należą m.in. firmy [Digia](http://pl.wikipedia.org/w/index.php?title=Digia&action=edit&redlink=1" \o "Digia (strona nie istnieje)), [KDAB](http://pl.wikipedia.org/w/index.php?title=KDAB&action=edit&redlink=1), [ICS](http://pl.wikipedia.org/wiki/ICS" \o "ICS).[Digia](http://pl.wikipedia.org/w/index.php?title=Digia&action=edit&redlink=1) jest właścicelem praw do komercyjnej wersji Qt oraz do znaków towarowych, po zakupie od firmy [Nokia](http://pl.wikipedia.org/wiki/Nokia).

1. Przykładowe zrzuty z programu.

1. Problemy napotkane przy tworzeniu programu

Problem, z którym mieliśmy bardzo dużo problemów to wycieki pamięci. W dniu pisania tego dokumentu wydaje nam się, że prawdopodobieństwo powstania wycieków zostało zredukowane do minimum.

Kolejnym problemem było odpowiednie dostosowanie formularza do wpisywania danych tak, aby można było wpisać np. tylko jedną macierz i na niej wykonywać odpowiednie obliczenia (obliczanie wyznacznika, generowanie macierzy transponowanej).

Przy rozwiązywaniu każdej z tych trudności bardzo pomagała nam dokumentacja bibliotek Qt dostarczona przez wydawców. Kolejnym źródłem, z którego czerpaliśmy wiedzę były zmagania programistów Qt z takimi samymi lub podobnymi problemami. Bardzo często na forach lub blogach znajdowaliśmy rozwiązanie.