

## **Instruction**

**COLLABORATORS**

	<i>TITLE :</i> Instruction		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 15, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Instruction</b>	<b>1</b>
1.1	Manual - main menu	1
1.2	Prawa autorskie i rozpowszechnianie	2
1.3	Do czego ten program służy?	2
1.4	Opis biblioteki	2
1.5	Co to jest BlaBla?	3
1.6	Pozostałe informacje	4
1.7	Norma i Norma 2	4
1.8	Print	4
1.9	GaussToLU	4
1.10	Odwrotna	5
1.11	Transponuj	5
1.12	Print	5
1.13	Rozwiązanie	5
1.14	DokladnoscMO	6
1.15	RozwiązanieUkładu	6
1.16	Wektor i Macierz	6
1.17	Układ	7
1.18	Stałe	8
1.19	Copyrights and distribution	9
1.20	Introduction	9
1.21	Instructions	9
1.22	BlaBla - what's that?	9
1.23	Other information	10

# Chapter 1

## Instruction

### 1.1 Manual - main menu



TKAlgebra

Version 1.0

Date 24.11.1999

Copyright (c) 1999 Kaczuô/BlaBla

Instrukcja obsługi

User's Manual

Prawa autorskie, dystrybucja.

Copyrights and distribution.

Do czego ten program służy?

Introduction.

Opis programu.

Instructions.

Co to jest BlaBla?

BlaBla? What's that?

Pozostałe informacje.

Other information.

## 1.2 Prawa autorskie i rozpowszechnianie

Prawa autorskie

Biblioteka Algebra jest FreeWare. Co oznacza, iŹ za uŹywanie kodu programu pŹaciŹ nie musisz, jednak tylko do momentu, gdy wykorzystujesz kod w celach niekomercyjnych. By wykorzystaŹ kod biblioteki do celów komercyjnych, naleŹy skontaktowaŹ siŹ w tej sprawie z autorem.

## 1.3 Do czego ten program sŹuŹy?

Biblioteka ma zaimplementowane operacje na macierzach i wektorach. RównieŹ zaimplementowane sŹ metody umoŹliwiajŹce odwracanie macierzy, oraz rozwiŹzywanie układów liniowych standardowych, nadokreŹlonych i niedookreŹlonych. Układy nadokreŹlone i niedookreŹlone rozwiŹzywane sŹ z wykorzystaniem metod Źredniokwadratowych.

## 1.4 Opis biblioteki

Opis funkcji:

Wektor:

Norma

Norma2

Print

Macierz:

Print

GaussToLU

Odwrotna

Transponuj

Uklad:

Print

---

Globalne funkcje:

Rozwiazanie

DokladnoscMO

RozwiazanieUkladu  
Publiczne Pola:

Wektor

Macierz

Uklad  
Staie:

Staie

## 1.5 Co to jest BlaBla?

Co to jest BlaBla?

BlaBla jest to grupa zrzeszajaca programistow piszacych programy wspolpracujace z systemem. Gwarantuje to poprawna prace na wielu modelach Amig, oraz zapewnia zachowanie kompatybilnosci "w gore". Pozwala to rowniez na korzystanie z tak waznej cechy, jaka posiada Amiga, jak multitasking. Nasze programy maja zazwyczaj status PublicDomain, FreeWare lub ShareWare, co pozwala na ich darmowe rozpowszechnianie.

Grupa posiada wlasne konto Internetowe i stronę WWW. Nasze najnowsze programy mozna rowniez znalezc w wydawanym przez nas pakiecie polskich programow uzytkowych PolWare. Zapraszamy rowniez do lektury redagowanego przez nas magazynu dyskowego Izviestia. Traktuje on glownie o programowaniu i zawiera opisy wielu ciekawych programow.

Zainteresowanych wspolpracą z naszą grupą, bądú chętnych do wymiany doświadczeń i uwag na temat programów, prosimy o kontakt z którymś z członków BlaBla.

W skład grupy wchodzi (V'97):

- blabla:

Kordian Adamczyk	(Kordi)
Krzysztof Habowski	(Kysy)
Łukasz Prokulski	(LeMUr)
Marcin Ochocki	(ScoTT)
Dariusz J. Garbowski	(Thufor)
Tomasz Korolczuk	(Tomash)

- sub-blabla:

Krzysztof Wolski	(Futrzak)
Tomasz Kaczanowski	(Kaczuô)
Karol Bryd	(Karol)

Krzysztof Doliński  
Grzegorz Kraszewski (Krashan)  
Piotr Hoppe  
Piotr Wyderski  
Tomasz Bieliński (Siumot)  
Daniel Owsianski (Valar)  
Przemysław Jeû (Warhawk)

## 1.6 Pozostałe informacje

Skontaktowaê sië ze mnâ moûna:

Tomasz Kaczanowski  
Rydzowa 23m32  
91-211 Îódú

E-Mail: kaczus@poczta.onet.pl

## 1.7 Norma i Norma 2

```
double Wektor::Norma()
```

Funkcja zwraca liczbë, która jest normâ wektora.

```
double Wektor::Norma2()
```

Funkcja zwraca liczbë, która jest normâ ôredniokwadratowâ wektora

## 1.8 Print

```
void Wektor::Print(FILE *p=stdout)
```

```
void Macierz::Print(FILE *p=stdout)
```

Funkcja drukujâca elementy wektora (bâdú macierz). Argumentem jest wskaúnik na obiekt typu FILE (czyli na plik, który musi byê otwarty), domyôlnie ustawiony jest na obiekt stdout, który zazwyczaj jest wskaúnikiem na konsolë.

## 1.9 GaussToLU

```
int Macierz::GaussToLU();
```

Funkcja ta sprowadza Macierz do postaci rozkîadu LU. Zwracana jest wartoê 1- gdy operacja sië powiodâ, bâdú 0 w razie niepowodzenia.

---

## 1.10 Odwrotna

```
Macierz *Macierz::Odwrotna();
```

Funkcja tworzy nową macierz, która jest macierzą odwrotną do danej. Zwracany jest wskaźnik na nowoutworzony obiekt, bądź NULL w przypadku, gdy macierz nie jest macierzą kwadratową.

Przykład:

```
Macierz *A,*B;  
....  
B=A->Odwrotna();
```

## 1.11 Transponuj

```
Macierz *Macierz::Transponuj();
```

Jest to metoda transponująca daną macierz. Wynikiem działania tej funkcji jest nowy obiekt typu Macierz, którego wskaźnik jest zwracany.

## 1.12 Print

```
void Układ::Print(const char)
```

Jest to funkcja drukująca do pliku niektóre obiekty zawarte w obiektach potomnych dla klasy Układ. Argumentem jest nazwa pliku, do którego mają być drukowane informacje o obiekcie.

## 1.13 Rozwiązanie

```
Wektor *Rozwiazanie(Macierz &A,Wektor &b);
```

Funkcja Rozwiazanie przyjmuje jako argumenty dwa obiekty, jeden typu Macierz, drugi typu Wektor. Zakłada się iż Macierz jest przekazana w postaci rozkładu LU. Zwracany jest wskaźnik na Wektor (będący rozwiązaniem układu równań), bądź NULL (w przypadku błędu przy rozwiązywaniu układu).

Przykład:

```
Macierz *A;  
Wektor *x,*b;  
...  
if(A->GaussToLU())
```



```
x=Rozwiazanie(*A,*b);
```

Jak widać bezpośrednio stosowanie tej funkcji nie jest najwygodniejsze, jest ona jednak niezbędna przy implementacji niektórych operacji. Dużo wygodniejsze do rozwiązania układu będzie użycie funkcji `RozwiazUklad`, opisanej w dalszej części pracy.

## 1.14 DokladnoscMO

```
double DokladnoscMO(Macierz &,Macierz &);
```

Argumentami są obiekty typu `Macierz`. Funkcja zwraca wielkość błędu po obliczeniu macierzy odwrotnej, dlatego argumentami powinny być `Macierz` i `Macierz` do niej odwrotna.

## 1.15 RozwiazanieUkladu

```
Uklad *RozwiazanieUkladu(Macierz &A, Wektor &b, char Flaga=0);
```

Funkcja tworzy obiekt typu `Uklad`. Argumentami są `Macierz A` - macierz współczynników układu równań, `Wektor b` - wektor wyrazów wolnych układu równań, `Flaga` (nie jest to argument konieczny), która mówi o sposobie rozwiązywania układu. Domyślnie przypisana jest wartość, o automatycznym doborze algorytmu rozwiązania układu równań. Dostępne są trzy wartości flagi:

## 1.16 Wektor i Macierz

```
unsigned int Ile_Wierszy;
```

Przechowuje informacje o ilości wierszy danego obiektu.

```
unsigned int Ile_Kolumn;
```

Przechowuje informacje o ilości kolumn danego obiektu.

```
char Bledy;
```

Tu zapisywane są informacje o błędach które wystąpiły w trakcie istnienia obiektu. Informacja jest tak skonstruowana, iż można łączyć kody błędów (kolejne bity odpowiadają informacjom o kolejnych błędach).

---

## 1.17 Układ

Macierz \*A;

Wskaźnik na Macierz współczynników układu równań.

Wektor \*b;

Wektor wyrazów wolnych układu równań.

Wektor \*x;

Rozwiązanie układu równań.

char Bledy;

Flaga błędu (definiowana dokładnie tak samo jak dla obiektów pochodnych od klasy Algebra)

double ERR;

Pole z informacją o dokładności obliczeń (maksymalny błąd)

```
double SrBlad
```

Pole z informacją o dokładności obliczeń (średni błąd obliczeń). Błędy liczone są jako różnica pomiędzy wartością obliczoną z lewej części równania po podstawieniu obliczonych wartości  $x$ , a wyrazem wolnym (prawą częścią równania).

```
double Czas;
```

Pole z informacją o czasie obliczeń (czas podawany jest w sekundach).

Na rzecz obiektu UkładNO zdefiniowane zostały dwie zmienne statyczne:

```
static double Dokladnosc
```

Zmienna informująca z jaką dokładnością ma być sprawdzana zerowość elementów normy liczonej dla kolejnej kolumny ortogonalizowanej macierzy. Jest to zmienna publiczna, tak więc jest możliwość ustawienia tej wartości według własnego uznania.

```
static char Czy_Zmieniac
```

Jest to flaga informująca metody wywoływane na rzecz obiektów typu UkładNO (i~potomnych), czy podczas ortogonalizacji należy zamieniać kolejno kolumny. 1 - oznacza, że kolumny mają być zamieniane podczas ortogonalizacji, 0 - że nie mają być zamieniane. Jest to zmienna dostępna jako zmienna publiczna, tak więc jest możliwość zmiany wartości tej flagi. Domyślnie ustawiona jest tak, iż ortogonalizując macierz kolumny są zamieniane.

---

## 1.18 Staïe

W module macierz.h zdefiniowane sã staïe wykorzystywane przy obsludze bïedów. Oto zdefiniowane i uÿwane przeze mnie kody bïedów:

```
#define BEZ_BLEDU 0
```

Brak bïedu (wyzerowane wszystkie bity).

```
#define BRAK_PAMIECI 1
```

Kïopoty przy alokowaniem niezbédnej iloœci pamieci (ustawiona 1 na pierwszym bicie).

```
#define BLEDNE_OPERACJE 2
```

Informacja, ùe wystãpiï bïad podczas próby wykonania pewnej operacji na obiektach (ustawiana jest 1 na 2 bicie).

```
#define BLEDNE_DANE 4
```

Informacja, iÛ podane dane byïy bïedne (np. przy operacji dodawania wektorów, sumowane wektory majã bïedy). Informacja o bïedzie ustawiana jest na 3 bicie.

```
#define ZLE_UWARUNKOWANIE 8
```

Informacja o ùle uwarunkowanej macierzy.

```
#define POZA_ZAKRESEM 16
```

Informacja, iÛ byïa próba pobrania nieistniejàcego elementu. Informacja ustawiana jest na 5 bicie.

Funkcja tworzy obiekt typu Ukklad. Argumentami sã Macierz A - macierz wspólczynników ukïadu równaï, Wektor b - wektor wyrazów wolnych ukïadu równaï, Flaga (nie jest to argument konieczny), która mówi o sposobie rozwiãzywania ukïadu. Domyœlnie przypisana jest wartoœê, o automatycznym doborze algorytmu rozwiãzania ukïadu równaï. Dostêpne sã trzy wartoœci flagi:

```
#define AUTODETECT_ 0
```

Oznacza automatyczny dobór algorytmu rozwiãzania ukïadu równaï.

```
#define GAUSS_ 1
```

Rozwiãzanie metodã Gaussa (w razie ukïadu nadokreœlonego, ukïad zostanie przemnoÿony lewostronnie przez macierz transponowanã do macierzy wspólczynników ukïadu równaï).

```
#define ORTOGONALIZACJAGS_ 2
```

Rozwiãzanie z wykorzystaniem metody ortogonalizacji Grama Schmidta (moÿliwe

jest równieŝ rozwiązanie tã metodã układu niedookreœlonego).

## 1.19 Copyrights and distribution

Copyrights

This program is FreeWare.

## 1.20 Introduction

Some of linear algebra algoritms implementation.

## 1.21 Instructions

Sorry, only in Polish

## 1.22 BlaBla - what's that?

What is BlaBla?

BlaBla is a Polish biggest system programmers group. We make the system friendly programms. We guarantee that our programs will run on almost every machine (Amiga of course), including future ones. All our programs are running in multitasking, and all of them are PD, Freeware, or Shareware.

Our organisation has a few members (we "absorb" only good programmers), an own e-mail address and own BBS. Our newest productions you can find in our PD library called "Polware". Our texts, articles you can also find in our disk-mag "Izviestia" (sorry, in Polish!).

BlaBla memberlist (V'97):

- blabla:
  - Kordian Adamczyk (Kordi)
  - Krzysztof Habowski (Kysy)
  - Łukasz Prokulski (LeMUr)
  - Marcin Ochocki (ScoTT)
  - Dariusz J. Garbowski (Thufor)
  - Tomasz Korolczuk (Tomash)
- sub-blabla:
  - Krzysztof Wolski (Futrzak)
  - Tomasz Kaczanowski (Kaczuô)
  - Karol Bryd (Karol)
  - Krzysztof Doliński (Krashan)
  - Grzegorz Kraszewski (Krashan)

Piotr Hoppe  
Piotr Wyderski  
Tomasz Bieliński (Siumot)  
Daniel Owsianski (Valar)  
Przemysław Jeŭ (Warhawk)

BlaBla group is well-known in Poland, and programs signed with "Blabla" are for sure very good. We made many productions, we help many young programmers, we write articles to Polish edition of Amiga Magazine, we try to do all, for our beloved computer - Amiga!

## 1.23 Other information

Contact with me:

Tomasz Kaczanowski  
Rydzowa 23m32  
91-211 Ńódú

E-Mail: [kaczus@poczta.onet.pl](mailto:kaczus@poczta.onet.pl)

---