

Ta lista dotyczy głównie zainstalowania i uruchomienia biblioteki ARMADILLO (m.in. do liczenia wartości i wektorów własnych)

## 1 Instalacja i uruchomienie biblioteki ARMADILLO

Załóżmy, że Twój projekt znajduje się w katalogu `qt-projekty/projekt1`

1. Ze strony <http://arma.sourceforge.net/download.html> ściągnij ostatnią "Stable version", obecnie jest to `armadillo-4.100.2.tar.gz` i nagraj ją w katalogu `qt-projekty/projekt1`

2. Rozpakuj ów plik:

```
chaos:~$ cd qt-projekty/projekt1
```

```
chaos:~/qt-projekty/projekt1$ tar xzf armadillo-4.100.2.tar.gz
```

i zmieńmy nazwę `armadillo-4.100.2.tar.gz` na `armadillo` wydając polecenie:

```
chaos:~/qt-projekty/projekt1$ mv armadillo-4.100.2 armadillo
```

3. Następnie w pliku

```
qt-projekty/projekt1/armadillo/include/armadillo_bits/config.hpp
```

odkomentuj linijkę zawierającą

```
#define ARMA_USE_LAPACK
```

(Uwaga: Jeśli chcesz tego używać w domu, musisz jeszcze się upewnić czy masz zainstalowaną bibliotekę LAPACK - niektóre dystrybucje linuxa mają ją zainstalowaną domyślnie)

Teraz otwieramy projekt w Qt-Creatorze. Musimy tutaj jeszcze ustawić dwie rzeczy:

4. w pliku `.pro` (tzn. w `projekt1.pro` jeśli tak nazywa się Twój projekt) dodaj linijkę

```
LIBS += /usr/lib/liblapack.so.3
```

5. W pliku `main.cpp` dodaj

```
#include "armadillo/include/armadillo"
```

oraz (gdzieś przed `int main(int argc, char *argv[])`)

```
using namespace arma;
```

W tym momencie biblioteka ARMADILLO wraz z LAPACK powinny działać w Twoim projekcie.

## 2 ARMADILLO - podstawy

Zapoznaj się ze stroną

[arma.sourceforge.net/docs.html](http://arma.sourceforge.net/docs.html)

Nie przeraż się - będziemy używali tylko kilku funkcji i typów danych. Podstawą jest typ `mat`, który to jest macierzą (`mat-rix`). W poniższym przykładzie macierze  $A$  i  $B$  są macierzami  $4 \times 4$ , a  $C = A * B$ , program również wyświetla macierz  $C$ :

```
mat A(4,4);
mat B(4,4);

for(int i=0; i<4;i++)
    for(int j=0; j<4; j++)
    {
        A(i,j)=i+j+1;
        B(i,j)=i*j+1;
    }

mat C = A * B; // = A*B;

qDebug() << "C=A*B=";
for(int i=0; i<4;i++)
{
    QString wiersz="";
    for(int j=0; j<4; j++)
        { wiersz = wiersz + QString::number(C(i,j))+"\t"; }
    qDebug()<<wiersz;
}
```

Natomiast liczenie wartości własnych i wektorów własnych prezentuje poniższy przykład (typ `vec` to wektor):

```

vec eigval(4);
mat eigvec(4,4);

//liczenie wektorow wlasnych i wartosci wlasnych macierzy SYMETRYCZNEJ C
eig_sym(eigval, eigvec, C);

qDebug()<<"Wektory_wlasne_macierzy_C: \n";
for(int i=0; i<4; i++)
{
    QString wiersz="";
    for(int j=0; j<4; j++) { wiersz = wiersz + QString::number(eigvec(i,j))+"\t";}
    qDebug()<<wiersz;
}

qDebug()<<"Wartosci_wlasne_macierzy_C: \n";
QString wiersz="";
for(int j=0; j<4; j++) wiersz = wiersz + QString::number(eigval(j))+"\t";
qDebug()<<wiersz;

```

### 3 Zadanie

(Więcej informacji do tego zadania na wykładzie i laboratorium)

- Weź jakiś mały obrazek  $f$  (np.  $20 \times 20$ ,  $N = 20$ ) i “przepisz” go do macierzy typu `mat`. Policz wartości własne i wektory własne macierzy  $m_1 = f \cdot f^T$ . Załóżmy, iż mamy  $r$  różnych wartości własnych  $\lambda_0, \dots, \lambda_{r-1}$ .
- Stwórz macierz  $D$  diagonalną rozmiaru  $r \times r$ :  $D(i, i) = \lambda_i, D(i, j) = 0$  dla  $i \neq j$ .
- Stwórz macierz  $U$  rozmiaru  $N \times r$ ,  $i$ -ty wektor tej macierzy to wektor własny macierzy  $m_1$  odpowiadający wartości własnej  $\lambda_i$ .
- Podobnie:  $m_2 = f^T \cdot f$ . Macierze  $m_1$  i  $m_2$  mają te same wartości własne. Stwórz macierz  $V$  rozmiaru  $N \times r$ , której  $i$ -ta kolumna to wektor własny macierzy  $m_2$  odpowiadający wartości własnej  $\lambda_i$ .

- SPRAWDŹ, że

$$f = UD^{\frac{1}{2}}V^T \quad (1)$$

- Zauważ, że wzór (1) można zapisać jako:

$$f = \sum_{i=1}^r \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i^T, \quad (2)$$

gdzie  $\mathbf{u}_i$  to  $i$ -ta kolumna macierzy  $U$ , a  $\mathbf{v}_i$  to  $i$ -ta kolumna macierzy  $V$  W powyższym wzorze pomiń kilka wyrazów i zapisz pozostałą sumę jako obraz.