

# Obliczenia numeryczne w C++ przy użyciu biblioteki Armadillo

Czy odkrywanie koła na nowo jest czymś, na co warto poświęcać czas? Prawdopodobnie nie. To samo tyczy się implementacji często używanych struktur danych i algorytmów do obliczeń numerycznych. Na szczęście możemy skorzystać z prostego w użyciu rozwiązania, które łączy w sobie najlepsze cechy wielu popularnych bibliotek i dodaje do nich szereg optymalizacji. Zapoznajmy się z Armadillo – biblioteką do realizacji obliczeń numerycznych w C++.

## I CZYM JEST ARMADILLO?

Armadillo jest biblioteką C++ służącą do realizacji obliczeń algebry liniowej, mającą na celu zachowanie równowagi pomiędzy łatwością użycia i szybkością działania. Dostarcza generycznych typów danych oraz funkcji i operatorów umożliwiających realizację wielu działań i operacji na macierzach i wektorach. Korzysta przy tym z szeregu implementacji dostarczonych przez sprawdzone biblioteki do obliczeń numerycznych i dodaje do tego automatyczne zrównoleglenie obliczeń z wykorzystaniem OpenMP.

Każdy, kto korzystał z Matlaba, może odnieść wrażenie, że kod napisany w Armadillo wygląda znajomo. Jednym z założeń podczas tworzenia biblioteki było umożliwienie prostej konwersji kodu Matlab/Octave na C++. Wiele rzeczy realizowanych jest bez wiedzy programisty. Całe wyrażenia są optymalizowane, aby łączyć łańcuchy wyrażeń dla zwiększenia szybkości ich obliczania.

Istotną zaletą jest również licencja Apache 2.0, która pozwala na wykorzystanie Armadillo zarówno do zastosowań prywatnych, open-sourcowych, jak i komercyjnych, pod warunkiem załączenia właściwych informacji dotyczących licencjonowania.

## I BIBLIOTEKI DODATKOWE

Armadillo ma szereg wbudowanych struktur danych i algorytmów. Są one wystarczająco dobre do realizacji prostych obliczeń na małych i średnich macierzach. Aby jednak wydobyć pełną moc biblioteki, konieczne jest zainstalowanie kilku dodatkowych zależności. Szczegóły dotyczące instalacji przedstawiono w dalszej części artykułu, tutaj natomiast skupimy się na tym, jakie korzyści dają nam poszczególne biblioteki i z których warto skorzystać. Poniżej znajdują się jedynie krótkie opisy niektórych z nich, ponieważ w większości przypadków można stosować ich różne odpowiedniki.

### I OpenMP

OpenMP (ang. *Open Multi-Processing*) [4] jest biblioteką, której nie trzeba przedstawiać większości osób programujących w C, C++ i Fortranie. Jest to wieloplatformowe API dostarczające szereg dyrektyw kompilatora i funkcji umożliwiających zrównoleglenie kodu. OpenMP najczęściej wykorzystuje się do tworzenia równoległego kodu drobnoziarnistego (czyli zrównoleglenie pętli) oraz gruboziar-

nistego (tj. tworzenie sekcji programu, które mogą być wykonywane równoległe). Pełna specyfikacja API OpenMP dostępna jest na stronie internetowej biblioteki [4].

Korzystając z Armadillo, nie musimy zastanawiać się, które fragmenty kodu mogą być zrównoleglone, a które nie. Co najważniejsze, nie musimy martwić się zachowaniem zależności w kodzie, co przy bardziej skomplikowanych programach bywa problematyczne. Wszystko dzieje się automatycznie, gdy korzystamy z kompilatora obsługującego standard C++11/C++14 i OpenMP w wersji co najmniej 3.1.

### I LAPACK

LAPACK (ang. *Linear Algebra PACKage*) [5] jest biblioteką napisaną w języku Fortran zawierającą szereg funkcji służących do numerycznego rozwiązywania najpopularniejszych problemów algebry liniowej. Zawiera ona API dla języka C o nazwie LAPACKE, z którego korzysta Armadillo. API to jest wykorzystywane m.in. do rozkładów macierzy (jak rozkład LU, QR, Schura etc.), a także do realizacji działań na macierzach gęstych.

Dzięki Armadillo możemy korzystać z wszelkich dobrodziejstw, które daje nam standard C++11 i jego następcy. Nie ma potrzeby zgłębiania tajników wysoko- i średniopoziomowego API LAPACKE dla języka C, konwencji nazewniczych i całego gąszczu wyspecjalizowanych funkcji [10]. Możemy za to używać wygodnych typów generycznych i operujących na nich funkcji dostarczonych przez Armadillo.

### I OpenBLAS

OpenBLAS (ang. *Open Basic Linear Algebra Subprograms*) [6] jest wydajną, open-sourcową implementacją BLAS. Odpowiada za najpopularniejsze niskopoziomowe operacje algebry liniowej wektor-vektor, wektor-macierz oraz macierz-macierz (np. dodawanie wektorów, mnożenie wektora przez macierz czy mnożenie macierzy przez macierz).

Wybierając tę implementację, otrzymujemy dodatkowo szereg optymalizacji dedykowanych różnym platformom sprzętowym. Ponadto OpenBLAS korzysta z wielowątkowości, jednak z pewnymi zastrzeżeniami dotyczącymi zrównoleglenia naszego programu. Na szczęście i w tym przypadku szczegóły konfiguracyjne nas nie interesują, bo zajmuje się nimi Armadillo.