

Programowanie i metody numeryczne, 04.04.2024

Krótki plan zajęć:

- Klasy, iteratory: kontynuacja
- Pliki nagłówkowe, źródłowe, CMake
- Interfejsy graficzne ImGui/ImPlot
- Metody numeryczne: interpolacje wielomianów, różniczki, całki
- Funkcja jako argument innej funkcji

Zadania:

Zadanie 1

Napisać iterator dla uprzednio zdefiniowanej klasy. Iterator ten powinien zawierać:

- konstruktor przyjmujący wskaźnik jako argument
- zdefiniowany operator `*`
- zdefiniowany operator `->`
- zdefiniowany operator `++` oraz `++(int)`, innymi słowy lewo- i prawostronny operator `++`
- zdefiniowany operator `==` oraz `!=`

Tak zdefiniowany iterator można załączyć do własnej klasy wektor. Zdefiniować w kalsie iterator `begin()` oraz `end()`.

Sprawdzić czy iterator spełnia swoje zadania i wypełnić wektor wartościami korzystając z `std::iota` albo `std::transform`.

Dopisać aliasy do iteratora: `'iterator_category'`, `'value_type'`, `'reference'`, `'pointer'`, `'difference_type'`.

Zadanie 2

Zadeklarować ww. klasę w pliku nagłówkowym a jej definicję umieścić w adekwatnym pliku źródłowym. Wywołać tę klasę w innym pliku źródłowym korzystając z pliku nagłówkowego oraz poprawnie skompilować cały kod.

Zadanie 3

Zaimplementować różniczkowanie numeryczne w postaci różnicy skończonej (finite difference) oraz całkowanie numeryczne w postaci reguły trapezów (trapezoidal rule). Każda z tych funkcji powinna przyjmować jako argument dwa wektory i zwracać wektor. Wyrysować na jednym wykresie funkcje pierwotną, jej pochodną oraz całkę.

Zadanie 4

Zaimplementować różniczkowanie oraz całkowanie numeryczne metodą Simpsona, gdzie funkcja przyjmuje jako argument inną funkcję i zwraca wartość pochodnej/całki w danym punkcie. Wyrysować na jednym wykresie funkcje pierwotną, jej pochodną oraz całkę.

Zadanie 5

Zaimplementować interpolację Lagrangea. Funkcja ma przyjmować jako argument listę/wektor punktów, na podstawie których ma się dokonać interpolacja oraz listę/wektor punktów, w których ma dokonać tej interpolacji. Wyrysować punkty pierwotne oraz interpolację na wykresie.

Zadanie 6

Zaimplementować interpolację Neville'a. Funkcja ma przyjmować jako argument listę/wektor punktów, na podstawie których ma się dokonać interpolacja oraz listę/wektor punktów, w których ma dokonać tej interpolacji. Wrysować punkty pierwotne oraz interpolację na wykresie.

Zadanie 7

Zaimplementować metodę równego podziału (bisekcji) szukania miejsc zerowych. Funkcja powinna jako argument przyjmować inną funkcję oraz krańce przedziału i zwracać położenie miejsca zerowego.

Zadanie 8

Zaimplementować metodę Newtona szukania miejsc zerowych. Funkcja powinna jako argument przyjmować inną funkcję i zwracać położenie miejsca zerowego.