

Programowanie i metody numeryczne

Zadania – seria 7.

Całkowanie II. Różniczkowanie.

Zadanie 1. diff – Różniczkowanie numeryczne I.

Napisz zestaw funkcji

```
double diffForward(double(*f)(double), double x, double h)
double diffBackward(double(*f)(double), double x, double h)
double diffCentral(double(*f)(double), double x, double h)
double diffRichardson(double(*f)(double), double x, double h)
```

obliczających pochodną zadanej funkcji $y = f(x)$, określonej przez `f`, w punkcie `x`. Wykorzystaj następujące wzory:

$$\begin{aligned}f'_{\text{forward}}(x) &= \frac{f(x+h) - f(x)}{h}, \\f'_{\text{backward}}(x) &= \frac{f(x) - f(x-h)}{h}, \\f'_{\text{central}}(x) &= \frac{f(x+h) - f(x-h)}{2h}, \\f'_{\text{richardson}}(x) &= \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}.\end{aligned}$$

Napisz program testowy sprawdzający poprawność działania tych funkcji dla kilku przykładowych różniczkowanych funkcji (np. x^2 , $\sin x$) i różnych wartości `h`.

Następnie, korzystając z tych funkcji, napisz program `diff`, który przyjmuje jako argumenty wywołania jedno ze słów: `forward`, `backward`, `central` lub `richardson`, określające metodę różniczkowania, oraz liczbę rzeczywistą określającą wartość `h`. Program powinien wczytywać ze standardowego wejścia liczbę rzeczywistą aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i obliczać pochodną z tego wielomianu, posługując się wskazaną metodą.

Zadanie 2. diffp – Różniczkowanie numeryczne II.

Napisz zestaw funkcji

```
std::vector<double> diffForwardP(const std::vector<double> & x, const
std::vector<double> & y)
std::vector<double> diffCentral(const std::vector<double> & x, const
std::vector<double> & y),
```

obliczających pochodną funkcji $y = f(x)$, dla której znane są wartości ($y_i = f(x_i)$) w zadanych punktach (x_i). Funkcja przyjmuje wektory `x` i `y`, zawierające, odpowiednio, wartości punktów (x_i) i (y_i) oraz zwraca wektor zawierający wartości (y'_i) pochodnej w punktach (x_i). Wykorzystaj następujące wzory:

$$\begin{aligned}f'_{\text{forward}}(x_i) &= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}, \\f'_{\text{central}}(x_i) &= \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}.\end{aligned}$$

Korzystając z tych funkcji, napisz program `diffp`, który przyjmuje jako argumenty wywołania jedno ze słów: `forward` lub `central`, określające metodę różniczkowania. Program powinien dwukrotnie wczytywać ze standardowego wejścia liczby rzeczywiste aż do napotkania znaku końca pliku, interpretując je, odpowiednio, jako wartości (x_i) i (y_i) , a następnie obliczać pochodną zadaną metodą i wypisywać na standardowe wyjście wartości (x_i) i (y'_i) w czytelny sposób.

* Zadanie 3. `intgauss` – Kwadratury Gaussa–Legendre’a dla wielomianów.

Napisz szablon funkcji

```
template<typename F>
double intGauss(F f, double a, double b, std::vector<double> w,
               std::vector<double> x)
```

obliczającej numeryczną wartość całki oznaczonej dowolnej funkcji całkownej. Zmienna `f` jest całkowaną funkcją, zaś zmienne `a` i `b` to, odpowiednio, dolna i górna granica całkowania. Wektor `w` jest zbiorem wag, zaś wektor `x` – zbiorem węzłów.

Napisz ponadto szablon funkcji

```
template<typename F>
double intGaussN(F f, double a, double b, int n)
```

działającej podobnie, jak funkcja `intGauss`, przyjmującej jednak zamiast zbiorów wag i węzłów liczbę naturalną ze zbioru $\{2, 3, 4, 5\}$ określającą ilość węzłów. Funkcja ta powinna zawierać w swoim ciele stałe wektory zawierające wartości wag i węzłów dla przypadku dwóch, trzech, czterech i pięciu węzłów oraz wywoływać funkcję `intGauss` z odpowiednimi argumentami. Wartości wag i węzłów należy obliczyć lub znaleźć w literaturze.

Napisz program testowy sprawdzający poprawność działania tych szablonów dla kilku przykładowych funkcji podcałkowych (np. x^2 , $\sin x$), wykorzystujący kwadratury z dwoma oraz z trzema węzłami.

Następnie, korzystając z tych szablonów, napisz program `intgauss`, który przyjmuje jako argumenty wywołania dwie liczby rzeczywiste określające granice całkowania oraz liczbę naturalną ze zbioru $\{2, 3, 4, 5\}$ określającą ilość węzłów. Program powinien wczytywać ze standardowego wejścia liczby rzeczywiste aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i obliczać całkę z tego wielomianu w zadanych granicach i przy zadanej liczbie węzłów metodą Gaussa–Legendre’a.

Opracowanie: Bartłomiej Zglinicki.