Programowanie i metody numeryczne

Zadania – seria 8.

Miejsca zerowe funkcji jednej zmiennej.

Zadanie 1. `bisection` – Metoda bisekcji.

Napisz szablon funkcji

```
template<typename F>
double rootBisection(F f, double a, double b, double eps)
```

która posługując się metodą bisekcji znajduje miejsce zerowe ciągłej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ położone w przedziale $[a, b] \subset X$, przy czym $f(a)f(b) < 0$, z dokładnością ε. Funkcja `rootBisection` przyjmuje argument `f` – implementację funkcji f oraz trzy argumenty `a`, `b` i `eps` typu `double`, odpowiadające kolejno liczbom a, b i ε. Wartością zwracaną przez funkcję `rootBisection` powinno być znalezione przez nią miejsce zerowe funkcji f. Załóż, że w przedziale $[a, b]$ znajduje się co najwyżej jedno miejsce zerowe funkcji f.

Funkcja `rootBisection` powinna sprawdzać, czy wartości jej argumentów są poprawne oraz czy spełniają założenia (a więc np. czy odpowiadają $f(a)f(b) < 0$, $a < b$, $0 < \varepsilon \leqslant b - a$). Jeśli okaże się, że tak nie jest, funkcja powinna zgłosić odpowiedni wyjątek (pochodzący z biblioteki standardowej lub napisany specjalnie na jej potrzeby) opatrzony komunikatem wyjaśniającym przyczynę jego wystąpienia.

Napisz program testowy sprawdzający poprawność działania tego szablonu dla trzech przykładowych funkcji: $f(x) = x^2$, $g(x) = x^2 - 2$ oraz $h(x) = \mathrm{e}^x + x - 1$, poszukujący dla każdej z nich miejsca zerowego w przedziale $[-1, 6]$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001, 0.0001, 0.0000001.

Następnie, korzystając z tego szablonu, napisz program `bisection`, który przyjmuje jako argumenty wywołania trzy liczby zmiennoprzecinkowe określające wartości a, b i ε. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować miejsce zerowe tego wielomianu w zadanym przedziale i z zadaną dokładnością. Możesz założyć, że w zadanym przedziale znajduje się co najwyżej jedno miejsce zerowe tego wielomianu.

Zadanie 2. `newton` – Metoda Newtona.

Napisz szablon funkcji

```
template<typename F>
double rootNewton(F f, double x0, double eps)
```

która posługując się metodą Newtona znajduje miejsce zerowe ciągłej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ położone najbliżej punktu $x_0 \in X$ z dokładnością ε. Funkcja `rootNewton` przyjmuje argument `f` – implementację funkcji f oraz dwa argumenty `x0` i `eps` typu `double`, odpowiadające kolejno liczbom x_0 i ε. Wartością zwracaną przez funkcję `rootNewton` powinno być znalezione przez nią miejsce zerowe funkcji f.

Napisz program testowy sprawdzający poprawność działania tego szablonu dla trzech przykładowych funkcji: $f(x) = x^2$, $g(x) = x^2 - 2$ oraz $h(x) = \mathrm{e}^x + x - 1$, poszukujący dla każdej z nich miejsca zerowego w pobliżu punktu $x_0 = 1,4$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001, 0.0001, 0.0000001.

Następnie, korzystając z tego szablonu, napisz program `newton`, który przyjmuje jako argumenty wywołania dwie liczby zmiennoprzecinkowe określające wartości x_0 i ε. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować miejsce zerowe tego wielomianu w pobliżu zadanego punktu i z zadaną dokładnością.

Zadanie 3. `steffensen` – Metoda Steffensena.

Napisz szablon funkcji

```
template<typename F>
double rootSteffensen(F f, double x0, double eps)
```

która posługując się metodą Steffensena znajduje miejsce zerowe ciągłej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ położone najbliżej punktu $x_0 \in X$ z dokładnością ε. Funkcja `rootSteffensen` przyjmuje argument `f` – implementację funkcji f oraz dwa argumenty `x0` i `eps` typu `double`, odpowiadające kolejno liczbom x_0 i ε. Wartością zwracaną przez funkcję `rootSteffensen` powinno być znalezione przez nią miejsce zerowe funkcji f.

Napisz program testowy sprawdzający poprawność działania tego szablonu dla trzech przykładowych funkcji: $f(x) = x^2$, $g(x) = x^2 - 2$ oraz $h(x) = \mathrm{e}^x + x - 1$, poszukujący dla każdej z nich miejsca zerowego w pobliżu punktu $x_0 = 1{,}4$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001, 0.0001, 0.0000001.

Następnie, korzystając z tego szablonu, napisz program `steffensen`, który przyjmuje jako argumenty wywołania dwie liczby zmiennoprzecinkowe określające wartości x_0 i ε. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować miejsce zerowe tego wielomianu w pobliżu zadanego punktu i z zadaną dokładnością.

* Zadanie 4. `secant` – Metoda siecznych.

Napisz szablon funkcji

```
template<typename F>
double rootSecant(F f, double x0, double x1, double eps)
```

która posługując się metodą siecznych znajduje miejsce zerowe ciągłej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ dla punktów początkowych x_0, $x_1 \in X$ z dokładnością ε. Funkcja `rootSecant` przyjmuje argument `f` – implementację funkcji f oraz trzy argumenty `x0`, `x1` i `eps` typu `double`, odpowiadające kolejno liczbom x_0, x_1 i ε. Wartością zwracaną przez funkcję `rootSecant` powinno być znalezione przez nią miejsce zerowe funkcji f.

Napisz program testowy sprawdzający poprawność działania tego szablonu dla trzech przykładowych funkcji: $f(x) = x^2$, $g(x) = x^2 - 2$ oraz $h(x) = \mathrm{e}^x + x - 1$, poszukujący dla każdej z nich miejsca zerowego z punktami początkowymi $x_0 = 1{,}4$ i $x_1 = -2$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001, 0.0001, 0.0000001.

Następnie, korzystając z tego szablonu, napisz program `secant`, który przyjmuje jako argumenty wywołania trzy liczby zmiennoprzecinkowe określające wartości x_0, x_1 i ε. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować miejsce zerowe tego wielomianu z zadanymi punktami początkowymi i z zadaną dokładnością.

* Zadanie 5. `exnewton` – Znajdowanie ekstremów funkcji metodą Newtona.

Napisz szablon funkcji

```
template<typename F>
double exNewton(F f, double x0, double eps)
```

która posługując się metodą Newtona znajduje ekstremum ciągłej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ położone najbliżej punktu $x_0 \in X$ z dokładnością ε. Funkcja `rootNewton` przyjmuje argument `f` – implementację funkcji f

oraz dwa argumenty `x0` i `eps` typu `double`, odpowiadające kolejno liczbom x_0 i ε. Wartością zwracaną przez funkcję `rootNewton` powinno być znalezione przez nią ekstremum funkcji f.

Napisz program testowy sprawdzający poprawność działania tego szablonu dla dwóch przykładowych funkcji: $f(x) = (x - 3)^2$ oraz $g(x) = \sin x$, poszukujący dla każdej z nich ekstremum w pobliżu punktu $x_0 = 1$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001.

Następnie, korzystając z tego szablonu, napisz program `exnewton`, który przyjmuje jako argumenty wywołania dwie liczby zmiennoprzecinkowe określające wartości x_0 i ε. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować ekstremum tego wielomianu w pobliżu zadanego punktu i z zadaną dokładnością.

Opracowanie: Bartłomiej Zglinicki.