Wykresy poziomicowe i mapy barwne.

Zadanie

Uzupełnić brakujący fragment kodu programu rysującego funkcje typu *f(x,y)*. Funkcja podana będzie w postaci zbioru wartości w_i w różnych punktach (x_i,y_i) przy czym wiadomo, że wartości (x_i,y_i) zawsze mieszczą się w zakresie od -2.5 do 2.5 każda. Punkty są rozrzucone przypadkowo. Program pozwala na wybór jednej z pięciu zadanych funkcji. Każda funkcja może być wyrysowana jako mapa konturowa (poziomice), mapa barwna (w której różnym wartościom funkcji odpowiadają różne barwy) oraz jako kombinacja tych dwóch metod. Wszystkie funkcje kontrolno-sterujące programu zostały już zaimplementowane. Do uzupełnienia pozostaje jedynie fragment kodu odpowiedzialny za rysowanie poszczególnych map. W przypadku map konturowych użytkownik zadaje liczbę poziomic, które powinny być rozmieszczone równomiernie pomiędzy najmniejszą a największą wartością przyjmowaną przez funkcję. Dla map barwnych należy udostępnić trzy sposoby kolorowania:

- czerwono – niebieski (niebieski odpowiada najniższym wartościom funkcji, a czerwony najwyższym),
- czerwono – zielono – niebieski (czerwony i niebieski jak poprzednio, natomiast im wartości są bliższe połowy rozpiętości wartości funkcji tym kolor jest bliższy zielonego),
- odcienie szarości (od czarnego dla najniższych wartości do białego dla najwyższych).

Program pozwala również na wyświetlenie położenia punktów określających funkcję *f(x,y)*.

Cel

Praktyczne wykorzystanie metody Sheparda do interpolacji nieregularnie rozmieszczonych danych oraz implementacja wybranego algorytmu rysowania poziomic.

Środki

Biblioteka wxWidgets.

Zarys możliwego rozwiązania

Przygotowane fragmenty kodu obsługują sterowanie programem oraz odświeżanie okna z rysowanymi funkcjami. Okno to jest odświeżane na podstawie bitmapy znajdującej się w pamięci, która powinna zawierać prawidłową mapę. Po każdej zmianie parametru wołana jest funkcja **DrawMap(…)**, do której przekazywane są wszystkie dane potrzebne do narysowania funkcji. Zadaniem programisty jest napisanie odpowiedniej funkcji **DrawMap(…)**. Funkcja ta zawarta jest w pliku o nazwie **draw_map.cpp** i jest to jedyny plik, który wolno edytować!!! Wszystkie pomocnicze funkcje, dodatkowe zmienne i cokolwiek co będzie potrzebne powinno się znaleźć wyłącznie w tym pliku!

Opis funkcji DrawMap(…)

Funkcja **DrawMap(…)** ma następującą deklarację:

void GUIMyFrame1::DrawMap(int N, float d[100][3], bool Contour, int MappingType, int NoLevels, bool ShowPoints)

Jak widać, jest to metoda klasy głównego okna. Do funkcji przekazywane są następujące zmienne:

int N - liczba punktów, dla których podano wartośd funkcji

float d[100][3] - tablica zawierająca położenia punktów i wartości funkcji (pierwszy indeks numeruje punkty, drugi podaje: argument x – indeks 0, argument y – indeks 1 oraz wartość funkcji dla tych argumentów – indeks 2)

bool Contour - jeżeli **true** to ma być rysowany kontur

int MappingType - przyjmuje wartości od 0 do 3, gdzie:
0 oznacza, że mapa barwna nie będzie rysowana
1 oznacza, że używamy kolorystyki od niebieskiego do czerwonego
2 oznacza, że używamy kolorystyki od niebieskiego, poprzez zielony do czerwonego
3 oznacza, że rysujemy mapę w odcieniach szarości

int NoLevels - podaje liczbę rysowanych poziomic (w zakresie od 1 do 9)

bool ShowPoints - jeżeli true to rysowane będą pozycje punktów opisujących funkcję

Przykładowa zawartość funkcji DrawMap(…).

Pierwsze dwie linijki funkcji tworzą pamięciowy kontekst urządzenia rysującego i wiążą z nim bitmapę, która jest przechowywana w pamięci. Linie te należy pozostawić nie zmienione. Od tej pory korzystając z kontekstu **memDC** można rysować na bitmapie. **Uwaga: bitmapa ma rozmiary 500x500 pikseli i nie należy ich przekraczać.**

Następnych kilka linijek służy wyłącznie pokazaniu, że to co jest rysowane w funkcji będzie wyświetlane w oknie. Ten fragment kodu można wyrzucić. Resztę kodu trzeba uzupełnić. Dla przykładu, funkcja może wyglądać następująco:

1. Interpolujemy funkcję do regularnej siatki metodą Sheparda
2. Wyznaczamy najmniejszą i największą wartość funkcji
3. Jeśli trzeba rysujemy mapę barwną
4. Wyznaczamy wartości poziomic na podstawie ich liczby
5. Jeśli trzeba rysujemy poziomice metodą poznaną na wykładzie (tzw. „kafelkową”)
6. Jeśli trzeba rysujemy punkty, w których podano wartości funkcji