Методы оптимизации
Решение задачи минимизации методом градиентного спуска

ЗАДАНИЕ.

Найти минимум функции двух переменных методом градиентного спуска с постоянным шагом, проверив применимость метода к заданной функции. Для решения составить компьютерную программу на любом языке программирования.

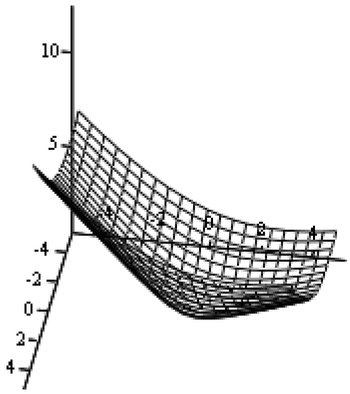
$f(\overline{x}) = \sqrt{x_1^2 + x_2^2 + 1} + \frac{1}{2}x_1 - \frac{1}{2}x_2$ $\qquad \overline{x}_0 = (1;2)$

РЕШЕНИЕ.

А) обоснование применимости метода к заданной функции;

Построим график данной функции.

$z(x1, x2) := \sqrt{x1^2 + x2^2 + 1} + \frac{x1}{2} - \frac{x2}{2}$



Видим, что функция имеет минимум.
Найдем вторые производные функции:

$$f\left(\bar{x}\right)=\sqrt{x_1^2+x_2^2+1}+\frac{1}{2}x_1-\frac{1}{2}x_2$$

$$\frac{df\left(\bar{x}\right)}{dx_1}=\frac{x_1}{\sqrt{x_1^2+x_2^2+1}}+\frac{1}{2}$$

$$\frac{df\left(\bar{x}\right)}{dx_2}=\frac{x_2}{\sqrt{x_1^2+x_2^2+1}}-\frac{1}{2}$$

$$\frac{d^2f\left(\bar{x}\right)}{dx_2x_1}=\frac{d^2f\left(\bar{x}\right)}{dx_1x_2}=-\frac{x_1x_2}{\sqrt{\left(x_1^2+x_2^2+1\right)^3}}$$

Б) краткое описание алгоритма метода;

В данном методе будем использовать формулу:

$$\bar{x}_{k+1}=\bar{x}_k-g(\bar{x}_k)\cdot\lambda_k$$

где $g(\bar{x}_k)$ - градиент $\begin{cases}\dfrac{df\left(\bar{x}\right)}{dx_1}=\dfrac{x_1}{\sqrt{x_1^2+x_2^2+1}}+\dfrac{1}{2}\\ \dfrac{df\left(\bar{x}\right)}{dx_2}=\dfrac{x_2}{\sqrt{x_1^2+x_2^2+1}}-\dfrac{1}{2}\end{cases}$

Повторяем процедуру поиска минимума до тех пор, пока не выполнится условие остановки итерационного процесса:

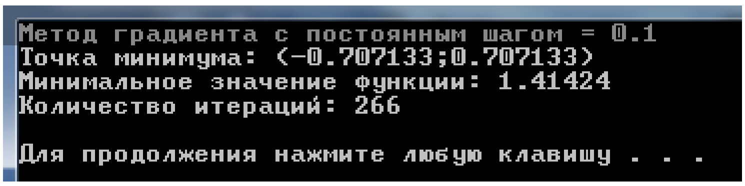
для быстро возрастающей функции $\left|f(x_{k+1})-f(x_k)\right|<\varepsilon$

В) программный код;

```
#include<iostream>
#include<iomanip>
using namespace std;

double f(double x1, double x2){  // функция
    return sqrt(x1*x1+x2*x2+1)+x1/2-x1/2;
}

double g1(double x1, double x2){  // градиент
    return x1/sqrt(x1*x1+x2*x2+1)+1.0/2;
}

double g2(double x1, double x2){  // градиент
```

```
	return x2/sqrt(x1*x1+x2*x2+1)-1.0/2;
}

int main(){
	setlocale(LC_ALL, "Russian"); // русский язык
	double x1_0=0, x2_0=0, lambda=0.00001, x1_1=1, x2_1=2; 
// начальные значения
	double e=0.000001; // точность
	int n; // количетсво итераций

	cout << "Метод градиента с постоянным шагом = " << 
lambda << endl;
	while (max(abs(x1_1-x1_0),abs(x2_1-x2_0))> e){	// 
разница в значениях x больше заданной точности
		x1_0=x1_1, x2_0=x2_1;
		x1_1=x1_0-lambda*g1(x1_0,x2_0); // находим новое 
значение x1
		x2_1=x2_0-lambda*g2(x1_0,x2_0); // находим новое 
значение x2
		n++; // наращиваем итерации
	}

	cout << "Точка минимума: (" << x1_1 << ";" << x2_1 << ")
" << endl;
	cout << "Минимальное значение функции: " << f(x1_1,x2_1) 
<< endl;
	cout << "Количество итераций: " << n << endl << endl;

	system("pause");

}
```

Г) результаты решения: координата точки минимума, минимальное значение функции, количество итераций при заданной точности 10^{-6};



```
Метод градиента с постоянным шагом = 0.01
Точка минимума: (-0.707383;0.707387)
Минимальное значение функции: 1.41449
Количество итераций: 2018

Для продолжения нажмите любую клавишу . . .
```

```
Метод градиента с постоянным шагом = 0.001
Точка минимума: (-0.70949;0.709791)
Минимальное значение функции: 1.41675
Количество итераций: 13939

Для продолжения нажмите любую клавишу . . .
```

```
Метод градиента с постоянным шагом = 0.0001
Точка минимума: (-0.717084;0.729585)
Минимальное значение функции: 1.43056
Количество итераций: 86439

Для продолжения нажмите любую клавишу . . .
```

```
Метод градиента с постоянным шагом = 1e-005
Точка минимума: (-0.652538;0.895552)
Минимальное значение функции: 1.49259
Количество итераций: 433311

Для продолжения нажмите любую клавишу . . .
```

Д) вывод о работе метода.

Как видим, увеличение шага дает худшие результаты и в плане минимизации функции и в плане итераций.