**Самостоятельная работа.**

**Тема 8. Функции.**

1. Напишите программу для вычисления матричного синуса

Математическая модель:  где

A – некоторая квадратная матрица, а n – целое число считается от единицы до некоторого конечного числа.

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Тип |
| i | Счетчик цикла, индекс строки | int |
| j | Счетчик цикла, индекс столбца | int |
| n | Счетчик цикла | int |
| a | Исходный массив | float |
| arr | Массив, передаваемый в функцию | float |
| parr | Вспомогательный массив в функции | float |
| tarr | Временный массив для хранения данных. | float |
| tarr1 | Временный массив для хранения данных при возведении в степень. | float |
| tarr2 | Временный массив для хранения данных при возведении в степень. | float |
| oarr | Матрица источник для копирования | float |
| iarr | Матрица приемник для копирования | float |
| k | Факториал и степень для матрицы | int |
| fm | Множитель -1 в степени деленная на факториал k | float |
| p | Степень для -1 | int |
| msin | Матричный синус | float |

Код программы:

#include <stdio.h>

#include <math.h>

void copy\_mtr(float (\*oarr)[2], float(\*iarr)[2]) {

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

iarr[i][j] = oarr[i][j];

}

}

}

int fact(int k) {

if (k == 1)

return 1;

return fact(k - 1) \* k;

}

void pow\_arr(float(\*parr)[2], float (\*arr)[2], int k) {

float tarr1[2][2];

float tarr2[2][2];

copy\_mtr(arr, tarr1);

for (int n = 1; n < k; n++) {

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

tarr2[i][j] = tarr1[i][0] \* arr[0][j] + tarr1[i][1] \* arr[1][j];

}

}

copy\_mtr(tarr2, tarr1);

}

copy\_mtr(tarr2, parr);

}

void matrsin(float(\*arr)[2]) {

float msin[2][2];

float parr[2][2];

double fm;

int n = 2, k, p;

copy\_mtr(arr, msin);

do {

p = n + 1;

k = 2 \* n - 1;

fm = pow(-1, p) / fact(k);

pow\_arr(parr, arr, k);

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

msin[i][j] += fm \* parr[i][j];

}

}

n++;

} while (n != 7);

copy\_mtr(msin, arr);

}

int main(){

float a[2][2] = { {1, 2}, {3, 4} };

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++)

printf("%g ", a[i][j]);

printf("\n");

}

matrsin(a);

printf("\n");

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++)

printf("%g ", a[i][j]);

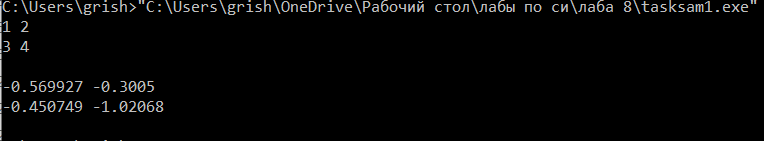
printf("\n");

}

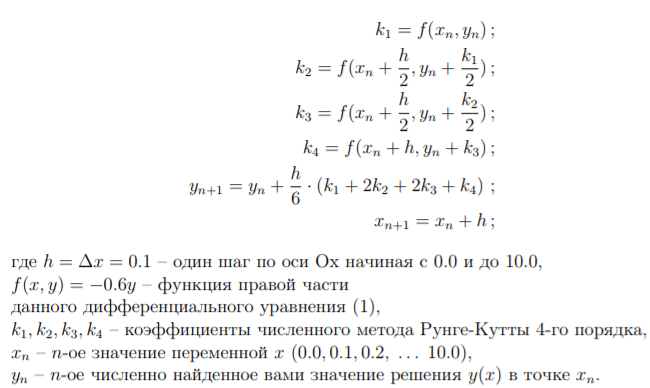
return 0;

}

Результат выполненной работы:



2. Напишите все необходимые функции для численного интегрирования простого дифференциального уравнения с начальным условием y(0) = 2, в интервале [0, 10] по переменной x классическим методом Рунге-Кутты 4-гопорядка с шагом по оси Ox ∆x = 0.1

Математическая модель: 

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Тип |
| x | Массив значений x | float |
| y | Массив значений y, текущее значение y | float |
| h | Шаг по x | float |
| k1 | Первый коэффициент численного метода Рунге-Кутты | float |
| k2 | Второй коэффициент численного метода Рунге-Кутты | float |
| k3 | Третий коэффициент численного метода Рунге-Кутты | float |
| k4 | Четвертый коэффициент численного метода Рунге-Кутты | float |

Код программы:

#include <stdio.h>

#define h 0.1

float func(float y) {

return -0.6 \* y;

}

void integ(float \*x, float \*y) {

float k1, k2, k3, k4;

for (int i = 0; i < 100; i++) {

k1 = func(y[i]);

k2 = func(y[i] + k1 / 2);

k3 = func(y[i] + k2 / 2);

k4 = func(y[i] + k3);

y[i+1] = y[i]+h/6\*(k1 + 2\*k2 + 2\*k3 + k4);

x[i+1] = x[i]+h;

}

}

int main() {

float x[101];

float y[101];

x[0] = 0;

y[0] = 2;

integ(x, y);

for (int i = 0; i < 101; i++)

printf("x =%g y = %g\n", x[i], y[i]);

return 0;

}

Результат выполненной работы:

