**Лабораторная работа № 9. Решение инженерных задач на основе циклических программ**

В инженерной практике часто встречаются задачи, для решения которых требуется использовать численные методы. Численные методы или методы вычислительной математики характеризуются тем, что решение конкретной задачи сводится к выполнению арифметических действий.

Рассмотрим способы использования циклических алгоритмов для вычисления площади криволинейной трапеции и для решения уравнений.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 4. Изучить способ нахождения корня уравнения в приложении Excel с помощью команды **Подбор параметра**.  5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200.  Вариант 8  x3 – 1 a = 0, b = 3  6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.  Написать программу вычисления корня уравнения методом ***дихотомии***. Точность вычислений принять равной **e** = 0,0001 для всех вариантов.  Найти корень уравнения с помощью приложения Excel.  Вариант 8  x3 + x – 4  **Дополнительные задания**  Вариант 13(+5)  e x + x − 4  Вариант 2(-6)  cos(x) + x – 7 | **#include <iostream>**  **using namespace std;**  **int main() {**  **double a = 0, b = 3, n = 200, h, s = 0, x;**  **h = (b - a) / n;**  **x = a;**  **while (x <= (b - h)) {**  **s += h \* ((pow(x, 3) - 1) + (pow((x + h), 3) - 1)) / 2;**  **x += h;**  **}**  **cout << "S = " << s;**  **return 0;**  **}**    методом *трапеций*    **#include <iostream>**  **using namespace std;**  **int main() {**  **double a = 0, b = 3, n = 200, h, s1 = 0, s2 = 0, x;**  **int i = 1;**  **h = (b - a) / (2 \* n);**  **x = a + 2 \* h;**  **while (i < n) {**  **s2 += (pow(x, 3) - 1);**  **x += h;**  **s1 += (pow(x, 3) - 1);**  **x += h;**  **i++;**  **}**  **double S = h / 3 \* ((pow(a, 3) - 1) + 4 \* (pow((a + h), 3) - 1) + 4 \* s1 + 2 \* s2 + (pow(b, 3)) - 1);**  **cout << "S = " << S;**  **return 0;**  **}**    методом ***парабол***    **#include <iostream>**  **using namespace std;**  **int main() {**  **double a = 0, b = 1, e = 0.0001, x;**  **while (fabs(a - b) > 2 \* e) {**  **x = (a + b) / 2;**  **double f\_x = pow(x, 3) + 3 \* x - 1;**  **double f\_a = pow(a, 3) + 3 \* a - 1;**  **if (f\_x \* f\_a <= 0) {**  **b = x;**  **}**  **else {**  **a = x;**  **}**  **}**  **cout << "X = " << x;**  **return 0;**  **}**  *A graph of function and a line  Description automatically generated with medium confidence*      #include <iostream>  using namespace std;  int main() {  double a = 0, b = 3, e = 1e-4, x = 0;  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (((exp(x) + x - 4) \* (exp(a) + a - 4)) <= 0) {  b = x;  }  else {  a = x;  }  }  cout << x;  }    ^  ||  #include <iostream>  using namespace std;  int main() {  double pi = 3.14159265359, a = pi, b = 3\*pi, e = 1e-4, x = 0;  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (((cos(x) + x -7) \* (cos(a) + a -7)) <= 0) {  b = x;  }  else {  a = x;  }  }  cout << x;  } |