2019 год

Практикум по численным методам: вычисление функции

Мусаева Аида, группа 208

1 Цель работы

Целью работы является овладение практическим навыком решения обратной задачи теории погрешностей, а также применение результатов этой работы в вычислении значения функции.

2 Постановка задачи

1) По указанной точности ($\epsilon = 10^{-6}$) решить обратную задачу теории погрешности для функции $z(x) = \sqrt{sin(x + 0.74)}sh(0.8x^2 + 0.1)$, где x=0.1(0.01)0.2.

2) Построить с требуемой точностью таблицу значений этой функции (квадратный корень вычислять по формуле Герона, остальные простейшие элементарные функции вычислять с использованием степенных рядов).

3) Составить ту же таблицу, используя встроенные функции и сравнить обе таблицы.

3 Аналитические вычисления

$u(x) = sin(x + 0.74)$
$v(x) = sh(0.8x^2 + 0.1)$
$f(u, v) = \sqrt{u} * v$
Найдем пределы изменения величин u, v при $x \in [0.1; 0.2]$. Функции $u(x), v(x)$ монотонно возрастают на [0.1;0.2]. Интервал изменения u, v можно расширить, чтобы не вычислять верхние и нижние границы изменения этих функций с большой точностью. На промежутке [0.1; 0.2]: $0.74 <= u <= 0.81$, $0.1 <= v <= 0.14//$ Таким образом, $G = \{(u, v)$: $0.74 <= u <= 0.81, 0.1 <= v <= 0.14\}$
$|f_u| = |\frac{v}{2\sqrt{u}}| < 0.09$
$|f_v| = |\sqrt{u}| < 0.9$
Таким образом, $\epsilon_u = \frac{10^{-6}}{0.27}, \epsilon_v = \frac{10^{-6}}{2.7}, \epsilon_f = \frac{10^{-6}}{3}$

4 Код программы

```
#include "stdafx.h"
#include <iostream>
#include <math.h>

using namespace std;

long int factorial(int i) {
if (i == 0) return 1;
else return i * factorial(i - 1);
}

long double sqr(long double n, const double eps) {
long double x = 1, nx;
while (true)
{
nx = (x + n / x) / 2;
if (fabs(x - nx) < eps)
break;
else x = nx;
}
```

```
return x;
}

long double sh(double n, const double eps) {
long double ans = 0;
int k = 0;

while (true)
{
long double u = pow(n, 2 * k + 1) / factorial(2 * k + 1);
ans += u;
if (abs(u) < eps)
break;
k++;
}
return ans;
}

long double sin(long double n, const double eps) {
long double ans = 0;
int k = 0;

while (true)
{
long double u = pow(n, 2 * k + 1) / factorial(2 * k + 1);
ans += pow(-1, k)*u;

if (abs(u) < eps)
break;
k += 1;
}
return ans;
}

long double f(long double x) {
return sqr(sin(x + 0.74, pow(10, -6) / 0.27), pow(10, -6) / 3)*sh(0.8*pow(x,2)+0.1, pow(
}

long double math_f(long double x) {
return sqrt(sin(x+0.74))*sinh(0.8*pow(x,2)+0.1);
}

int main() {

for (double x = 0.1; x <= 0.2; x += 0.01) {
double res1 = f(x);
cout << res1 << endl;
}

cout << endl;

for (double x = 0.1; x <= 0.2; x += 0.01)
{
```

```
double res2 = math_f(x);
cout << res2 << endl;
}

cout << endl;

for (double x = 0.1; x <= 0.2; x += 0.01) {

double res3 = fabs(math_f(x) - f(x));
cout << res3 << endl;
}
}
```

5 Таблицы

Значения при использовании моей реализации функций

x	$z(x)$
0.1	0.09337740236
0.11	0.09525743088
0.12	0.09728414920
0.13	0.09945918858
0.14	0.10178413613
0.15	0.10426053623
0.16	0.10688989193
0.17	0.10967366641
0.18	0.11261328449
0.19	0.11571013413

Значения при использовании стандартных функций

x	$z(x)$
0.1	0.09337740200
0.11	0.09525743052
0.12	0.09728414883
0.13	0.09945918819
0.14	0.10178413571
0.15	0.10426053577
0.16	0.10688989142
0.17	0.10967366584
0.18	0.11261328385
0.19	0.11571013341

Разница между значениями

Δ
3.60824e-10
3.59994e-10
3.6994e-10
3.9006e-10
4.20053e-10
4.59883e-10
5.09742e-10
5.7003e-10
6.41332e-10
7.24404e-10