Практикум по вычислительным методам: Вычисление функций

Блинов Иван Сергеевич

01 марта 2019

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

2.1

По указанной точности $E = 10^{-6}$ решить обратную задачу теории погрешностей для функции $z(x) = exp(1 + x) * cos(\sqrt{1 + x})$, где $x = 0.01(0.005)0.06$

2.2

Построить с требуемой точностью таблицу значений этой функции (квадратный корень вычислить с помощью формулы Герона, остальное - степенными рядами)

2.3

Составить ту же таблицу с использованием встроенных функций языка и сравнить их

3 Аналитические вычисления

Пусть:
$\phi(x) = \exp(1 + x) = U,\ g(x) = cos(\sqrt{1 + x}) = V,\ f(x) = U * V$
При $x \in [0.01; 0.06]$:

$$exp(1.01) <= U <= exp(1.06)$$

$$2.74 <= U <= 2.89$$

$$cos(\sqrt{1.01}) <= V <= cos(\sqrt{1.06})$$

$$0.51 <= V <= 0.54$$

Таким образом, $G = \{(U, V) : 2.74 <= U <= 2.89, 0.51 <= V <= 0.54\}$.
Оценим в G частные производные:

$$|\frac{\partial z}{\partial u}| = |V| <= 0.54$$

$$|\frac{\partial z}{\partial v}| = |U| <= 2.89$$

Функцию ϕ вычисляем с точностью $E_1 = \frac{E}{1.62}$, g с $E_2 = \frac{E}{8.67}$, f с $E_3 = \frac{E}{3}$

4 Код

```
import math
import numpy as np


def exp(n, eps=1e-6):
    ans = 0
    k = 0
    while(True):
        add = n**k/math.factorial(k)
        if add < eps:
            break
        ans += add
        k += 1
    return ans


def cos(n, eps=1e-6):
    ans = 0
    k = 0
    while(True):
        add = n ** (2 * k) / math.factorial(2 * k)
        if add < eps:
            break
        ans += ((-1) ** k) *add
        k += 1

    return ans


def sqrt(n, eps=1e-15):
    x = 1
    while(True):
        nx = (x + n / x) / 2
        if abs(x - nx) < eps:
            break
        x = nx
    return x


def f(x, eps=1e-6):
    return exp(1+x, eps=eps/1.62)*cos(sqrt(1+x), eps=eps/8.67)
```

```
def f_exact(x):
    return math.exp(1+x)*math.cos(math.sqrt(1+x))


print('f')
for x in np.arange(0.1, 0.21, 0.01):
    print('{} & {} \\\\'.format(x, f(x)))


print('f_exact')
for x in np.arange(0.1, 0.21, 0.01):
    print('{} & {} \\\\'.format(x, f_exact(x)))


print('max difference')
res = -1
for x in np.arange(0.1, 0.21, 0.01):
    res = max(res, abs(f_exact(x) - f(x)))
print(res)
```

5 Таблицы

5.1 Моя реализация

x	$z(x)$
0.1	1.497888927919311
0.11	1.5004149270233296
0.12	1.5028632192878844
0.13	1.5052321931154162
0.13999999999999999	1.5075202142060602
0.14999999999999997	1.5097256252805324
0.15999999999999998	1.5118467457998839
0.16999999999999998	1.5138818716820868
0.17999999999999997	1.5158292750154312
0.18999999999999995	1.517687203768673
0.19999999999999996	1.5194538814979364

5.2 Реализация встроенными методами

x	$z(x)$
0.1	1.4978889781684341
0.11	1.5004149817890935
0.12	1.5028632789333878
0.13	1.505232258029139
0.13999999999999999	1.5075202848033342
0.14999999999999997	1.509725702005078
0.15999999999999998	1.5118468291254101
0.16999999999999998	1.5138819621139576
0.17999999999999997	1.5158293730923977
0.18999999999999995	1.5176873100646828
0.19999999999999996	1.5194539966240022

Максимальная разница между значениями функции: 1.1512606579167084e-07