САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ

Направление подготовки: 09.03.04 «Программная инженерия»
б5130904/30022

Задание №1, вариант 14.

Выполнил студент: Лютов Александр Владимирович, группа 30022.

Преподаватель: Воскобойников Сергей Петрович.

Задание

В А Р И А Н Т N 14

Для $1 \le x \le 4$ с $h = 0.375$ вычислить значения $f(x) = \int_0^{20} \frac{dz}{e^z(z+x)}$,

используя для вычисления интеграла программу **QUANC8**. По полученным точкам построить сплайн-функцию и полином Лагранжа 8-й степени. Сравнить значения обеих аппроксимаций в точках $x_k = 1.1875 + 0.375k$ (k=0,1,..,7).

Результат работы программы

xk	f(xk)	spline(xk)	lagrange(xk)	f(xk)-spline(xk)	f(xk)-lagrange(xk)
1.1875	0.529808	0.530495	0.530405	0.000687	0.000597
1.5625	0.435033	0.434805	0.434916	0.000228	0.000117
1.9375	0.370219	0.370262	0.370263	0.000043	0.000044
2.3125	0.322808	0.322786	0.322779	0.000022	0.000028
2.6875	0.286487	0.286492	0.286515	0.000005	0.000028
3.0625	0.257707	0.257689	0.257663	0.000018	0.000044
3.4375	0.234303	0.234360	0.234417	0.000056	0.000113
3.8125	0.214876	0.214663	0.214311	0.000213	0.000565

Выводы

Сравнение аппроксимаций

1. **Сплайн-аппроксимация**:
   - Средняя ошибка: ≈0.00016.
   - Максимальная ошибка: 0.000687 при xk=1.1875.
   - Минимальная ошибка: 0.0000050 при xk=2.6875.
2. **Полином Лагранжа**:
   - Средняя ошибка: ≈0.00019.
   - Максимальная ошибка: 0.000565 при xk=3.8125.
   - Минимальная ошибка: 0.000028 при xk=2.6875.

Выводы

- **Точность**:
  - В большинстве точек сплайн-аппроксимация дает меньшую ошибку по сравнению с полиномом Лагранжа.
  - Исключение составляет точка xk=1.5625, где полином Лагранжа точнее.
  - В крайней точке xk=3.8125 ошибка полинома Лагранжа значительно возрастает.
- **Стабильность**:
  - Сплайн демонстрирует более стабильную точность на всем интервале.
  - Полином Лагранжа менее стабилен, особенно на краях интервала.

Итоговый ответ

Сплайн-аппроксимация в данном случае работает лучше полинома Лагранжа, так как обеспечивает меньшую среднюю ошибку и более стабильную точность на всем интервале.

Код программы

main.f90

```
program main
   use integral_func_mod
   implicit none

   interface
      subroutine SPLINE(N, X, Y, B, C, D)
         integer, intent(in) :: N ! Число заданных точек или узлов
         real, intent(in)    :: X(N) ! Абсциссы узлов в строго возрастающем порядке
         real, intent(in)    :: Y(N) ! Ординаты узлов
         real, intent(out)   :: B(N), C(N), D(N) ! Массивы определенных выше коэффициентов сплайна
      end subroutine SPLINE

      function SEVAL(N, Xi, X, Y, B, C, D) result(seval_value)
         integer, intent(in) :: N
         real, intent(in)    :: Xi, X(N), Y(N), B(N), C(N), D(N)
         real                :: seval_value
      end function SEVAL

   end interface

   external quanc8

   real    :: a, b, relerr, abserr, res, errest, flag
   integer :: nofun
   real, allocatable :: x_values(:), f_values(:)
   real, allocatable :: b_coef(:), c_coef(:), d_coef(:)
   integer :: i, k, x_n
   real    :: xk, spline_val, lagrange_val

   ! Установка параметров интегрирования
   a = 0.0
   b = 20.0
   relerr = 1.E-06
   abserr = 0.0
   h = 0.375
   x = 1.0
   x_n = 1

   DO WHILE (x <= 4)
      x = x + h
      x_n = x_n + 1
   END DO
   x = 1.0

   ALLOCATE(x_values(x_n), f_values(x_n), b_coef(x_n), c_coef(x_n), d_coef(x_n))

   i = 1
   ! Вычисление значений функции f(x) для 1 <= x <= 4 с шагом h
   DO WHILE (x <= 4.0)
      CALL quanc8(integral_func, a, b, abserr, relerr, res, errest, nofun, flag)
      x_values(i) = x
      f_values(i) = res

      x = x + h
      i = i + 1
   END DO



   ! Вызов подпрограммы SPLINE для вычисления коэффициентов сплайна
   CALL SPLINE(x_n, x_values, f_values, b_coef, c_coef, d_coef)

   ! Сравнение значений в точках xk = 1.1875 + 0.375k (k=0,1,...,7)
   WRITE(*, '(A20, A20, A20, A20, A20, A20)') &
      'xk', 'f(xk)', 'spline(xk)', 'lagrange(xk)', 'f(xk)-spline(xk)', 'f(xk)-lagrange(xk)'
   DO k = 0, 7
      xk = 1.1875 + 0.375 * k
      ! Вычисление значений сплайна, полинома Лагранжа и функции в точке xk
      spline_val = SEVAL(x_n, xk, x_values, f_values, b_coef, c_coef, d_coef)
      lagrange_val = compute_lagrange(xk, x_values, f_values)
      x = xk
```

```
      CALL quanc8(integral_func, a, b, abserr, relerr, res, errest, nofun, flag)

      WRITE(*, '(F20.4, F20.6, F20.6, F20.6, F20.6, F20.6)') &
         xk, res, spline_val, lagrange_val, abs(res-spline_val), abs(res-lagrange_val)
   END DO

   DEALLOCATE(x_values, f_values, b_coef, c_coef, d_coef)

contains

   REAL FUNCTION compute_lagrange(x, x_values, f_values) RESULT(lagrange_val)
      REAL, INTENT(IN) :: x, x_values(:), f_values(:)
      INTEGER :: n, i, j
      REAL :: term, prod

      n = SIZE(x_values)
      lagrange_val = 0.0

      DO i = 1, n
         term = f_values(i)
         prod = 1.0
         DO j = 1, n
            IF (j /= i) THEN
               prod = prod * (x - x_values(j)) / (x_values(i) - x_values(j))
            END IF
         END DO
         lagrange_val = lagrange_val + term * prod
      END DO
   END FUNCTION compute_lagrange

 end program main
```

integral_func.f90

```
MODULE integral_func_mod
   IMPLICIT NONE

   real :: x, h

CONTAINS

   real FUNCTION integral_func(z) RESULT(func_value)
      real, intent(in) :: z
      func_value = 1.0 / (EXP(z) * (z + x))
   END FUNCTION integral_func


END MODULE integral_func_mod
```