ЦЕЛИ ВЫПОЛНЕНИЯ ЗАДАНИЯ

- Изучить интерполяцию функций с помощью интерполяционных многочленов Лагранжа и Ньютона.

ЗАДАНИЕ

Вариант 5. Построить интерполяционные многочлены в форме Лагранжа и Ньютона, используя номер варианта k, соответствующие значения параметров m и p_i и значения x_i, y_i из таблиц:

$\mathbf{x_i}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\mathbf{p_i}$	0.0	0.41	0.79	1.13	1.46	1.76	2.04	2.3	2.55	2.79	3.01

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0	1	1.5	2	2.5	3	3.5	4	4.5	1.8	2.53	3.96	5.33	1.96

$$\boldsymbol{y_i = p_i + (-1)^k * m}$$

Оценить погрешность. Вычислить значение функции в точке 0.47 с помощью интерполяционного многочлена и многочлена наилучшего приближения. Сравнить значения.

ХОД РАБОТЫ

Перед началом выполнения работы были изучены теоретические материалы на тему «Интерполяция с помощью интерполяционных многочленов Лагранжа и Ньютона».

Далее были составлены алгоритмы процедур вычисления интерполяционных многочленов Лагранжа и Ньютона (рис. 1, 2).

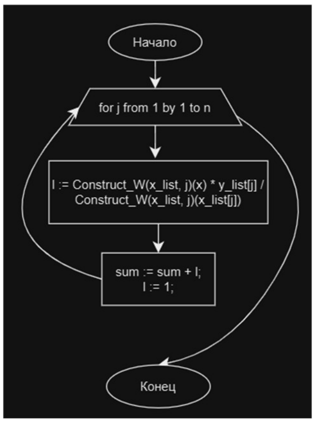

Рисунок 1 – Алгоритм составления интерполяционного многочлена Лагранжа

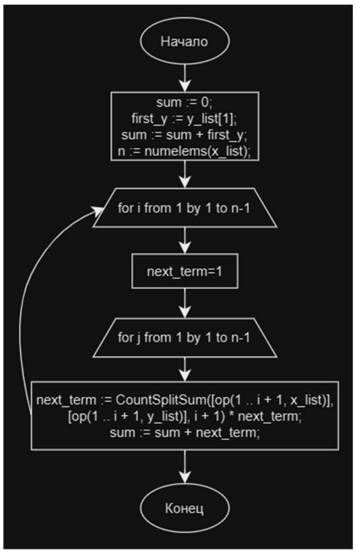

Рисунок 2 – Алгоритм составления интерполяционного многочлена методом Ньютона

После чего были реализованы обе такие процедуры (рис. 3, 4).

```
> Lagrange_P := proc(x_list,y_list)
  local l, i, j, n, sum;
  n := numelems(x_list);
  sum := 0;

   for j from 1 by 1 to n do
     l := Construct_W(x_list,j)(x) / Construct_W(x_list,j)(x_list[j]) · y_list[j];
     sum := sum + l;
     l := 1
   end do;

  return unapply(sum, x);
  end proc:
```

Рисунок 3 – Реализация процедуры (метод Лагранжа)

```
> Newthon_P := proc(x_list, y_list)
  local first_y, n, sum, next_term, i, j;
  sum := 0;
  first_y := y_list[1];
  sum := sum + first_y;
  n := numelems(x_list);

  for i from 1 by 1 to n − 1 do
   next_term := 1;
     for j from 1 by 1 to i do
     next_term := next_term · (x − x_list[j])
     end do;
   next_term := CountSplitSum([op(1..(i + 1), x_list)], [op(1..(i + 1), y_list)], i + 1) · next_term;
   sum := sum + next_term;
  end do;

  return unapply(sum, x);
  end proc:
```

Рисунок 4 – Реализация процедуры (метод Ньютона)

Для чистоты кода алгоритм вычисления $\omega(x) = (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n)$ (в методе Лагранжа) был вынесен в отдельную функцию (рис. 5).

```
> Construct_W := proc(x_list, j)
   local w, i;
   w := 1;

   for i from 1 to numelems(x_list) do
    if i ≠ j then
    w := w·(x − x_list[i]);
    end if;
   end do;

   return unapply(w, x);
  end proc:
```

Рисунок 5

То же проделано с алгоритмом вычисления конечной разности k-го порадка, используемом в методе Ньютона (рис. 6).

```
> CountSplitSum := proc(args, res, k)
  local new_args_l, new_args_r, new_res_l, new_res_r;
  if numelems(args) = 2 then
   return ((args[k] − args[k − 1]) / (res[k] − res[k − 1]))^(−1);
  end if;
  new_args_l := subsop(1 = NULL, args);
  new_args_r := subsop(numelems(args) = NULL, args);
  new_res_l := subsop(1 = NULL, res);
  new_res_r := subsop(numelems(res) = NULL, res);
  return (CountSplitSum(new_args_l, new_res_l, k − 1) − CountSplitSum(new_args_r, new_res_r, k − 1)) / (args[k] − args[1]);
  end proc:
```

Рисунок 6

Применив реализованные методы на исходные данные, получим

```
> Newthon := simplify(Newthon);
```

$$Newthon := \frac{1759}{2800}x + \frac{265625}{81}x^{10} - \frac{6359375}{378}x^9 - \frac{11620625}{252}x^7 + \frac{7018750}{189}x^8 - \frac{5843983}{6048}x^3 + \frac{13654985}{2592}x^4 - \frac{4951475}{288}x^5 + \frac{15259175}{432}x^6 + \frac{1558189}{16800}x^2 - \frac{5}{2}$$

```
> Lagrange := simplify(Lagrange);
```

$$Lagrange := \frac{1759}{2800}x + \frac{265625}{81}x^{10} - \frac{6359375}{378}x^9 - \frac{11620625}{252}x^7 + \frac{7018750}{189}x^8 - \frac{5843983}{6048}x^3 + \frac{13654985}{2592}x^4 - \frac{4951475}{288}x^5 + \frac{15259175}{432}x^6 + \frac{1558189}{16800}x^2 - \frac{5}{2}$$

Далее оценим погрешность. Т.к. функция в задании работы задана таблично, невозможно вычислить погрешность интерполяции. Поэтому для её вычисления рассмотрим пример (рис. 7). Погрешность ищется по формуле

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x)$$

.

#Оценка погрешности

```
> f := ln(x) :
> x_list := [1, 1.1, 1.2, 1.3, 1.4, 1.5] :
  y_list := [0, ln(1.1), ln(1.2), ln(1.3), ln(1.4), ln(1.5)] :
>
> Lagrange := simplify(Lagrange_P(x_list, y_list)(x));
```

$$Lagrange := -2.07791150000000 + 0.0682501\,x^5 - 0.532919000000000\,x^4 + 1.76879800000000\,x^3 - 3.28982100000000\,x^2 + 4.06361000000000\,x$$

```
> Newthon := simplify(Newthon_P(x_list, y_list)(x));
```

$$Newthon := -2.07791418200000 + 0.06825091700\,x^5 - 0.532922085500000\,x^4 + 1.76880586000000\,x^3 - 3.28982942100000\,x^2 + 4.06360891200000\,x$$

$$> pres := \frac{evalf(maximize(diff(f, x\$7), x = 1..1.5)) \cdot 0.5}{7!};$$

$$pres := 0.07142857145$$

Рисунок 7

Для наглядности пострим график рассматриваемой функции (ln(x)), 5 точек, по которым методом Лагранжа ищется интерполяционный многочлен, а также покажем погрешность (рис. 8).

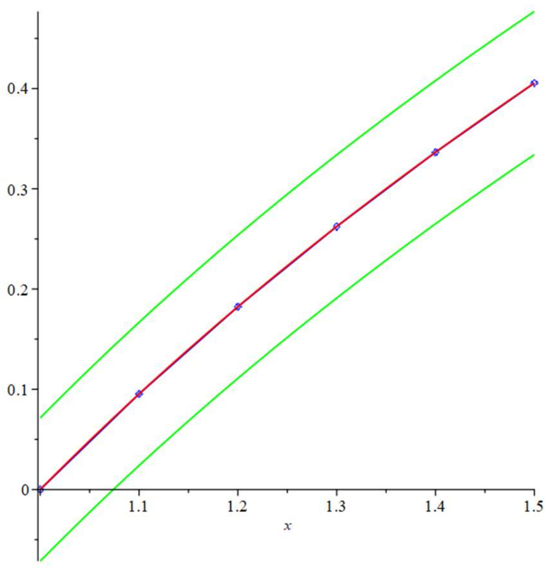

Рисунок 8

Теперь необходимо вычислить значение функции в точке 0.47 для интерполяционного многочлена и многочлена наилучшего приближения.

Для примера рассмотрим функцию e^x. Получим следующий интерполяционный многочлен

```
> Newthon := simplify(evalf(Newthon_P(x_list, y_list)(x)));
```

$$Newthon := 1.00000010600000 - 0.000248\,x^9 + 0.00120600000000000\,x^8 - 0.00214160000000000\,x^7 + 0.00396100000000000\,x^6 + 0.00659758959999998\,x^5 + 0.0424115669999999\,x^4 + 0.166464387800000\,x^3 + 0.500033237600000\,x^2 + 0.999997042799998\,x$$

Аппроксимацией методом наименьших квадратов найдем многочлен наилучшего приближения. Получим

```
> BestByStandardDeviation := unapply( a·sin(x) + b·cos(x) + c, x)
```

$$BestByStandardDeviation := x \mapsto 0.7015785300 \cdot \sin(x) - 2.330189767 \cdot \cos(x) + 3.356566068$$

Теперь осталось сравнить полученные значения: значение самой функции e^x, значение интерполяционного многочлена и значение многочлена наилучшего приближения:

```
> #Сравнение значений
> val := evalf(exp(0.47));
                                        val := 1.599994193
> interpol_value := evalf(subs(x = 0.47, Newthon));
                                        interpol_value := 1.599994192
> best_approach_value := BestByStandardDeviation(0.47);
                                        best_approach_value := 1.596778060
```

По полученным данным, интерполяционный многочлен дал более близкое к правильному ответу значение.

ВЫВОДЫ

В ходе выполнения данной работы была изучена интерполяция функции методами Лагранжа и Ньютона.

Были построены и запрограммированы алгоритмы методов, получен интерполяционный многочлен методами Лагранжа и Ньютона, была вычислена погрешность на примере функции ln(x), а также получены значения функции (взятой для примера e^x) при x=0.47. Более точный результат был получен при вычислении значения интерполяционного многочлена.