Московский государственный технический университет им.Н.Э.Баумана
Калужский филиал
Приборостроительный факультет
Кафедра П2-КФ

Отчет

по лабораторной работе № 2

«Реализация полиномиальной функции на языке ассемблера»

по курсу

«Теория и проектирование ЭВМ и ВС»

Студент : Комиссаров А.В., гр.ЭВМ-92

Преподаватель : доцент Максимов А.В.

Калуга, 1998 г.

<u>Содержание.</u>

1. Задание на лабораторную работу

Вариант по списку - **№25** :

f(x) = csc x = 1 / sin x;
$|x_{max}|$ = 3π / 4.

Дополнительные указания :

- число разрядов сетки N = 16;
- необходимо взять первые четыре ненулевых члена разложения функции в ряд Маклорена.

Требуется :

- разработать математический алгоритм реализации функции;
- разработать машинный алгоритм реализации функции;
- рассчитать масштабные соотношения для аргумента функции, для промежуточных значений и для результата;
- написать и отладить программу на языке ассемблера;
- построить график эталонной функции, ее реализации на языке ассемблера и график ошибки;
- рассчитать математическое ожидание и дисперсию ошибки.

2. Математический алгоритм

Разложение исходной функции косеканса в ряд Маклорена, в котором взяты только первые четыре члена, имеет вид :

$$f(x) \approx \frac{1}{x} + \frac{1}{6}x + \frac{7}{360}x^3 + \frac{31}{15120}x^5. \qquad (2.1)$$

Разложение функции в ряд сходится на всем рассматриваемом интервале $x \in \left[-\frac{3\pi}{4}; \frac{3\pi}{4}\right]$ за исключением точки $x = 0$, в которой данная функция является неограниченной.

Рассмотрим далее несколько (три) варианта реализации функции и выберем наиболее оптимальный среди них (критерием оптимальности является здесь минимум сложных операций типа умножения или деления (idiv, imul), а также минимум общего числа операций).

- *Первый вариант.* Вводя промежуточные переменные u_i, согласно выражению (2.1) получим следующий алгоритм :

$u_1 = \frac{1}{x}$; $u_2 = \frac{x}{6}$; $u_3 = x \cdot x$; $u_4 = u_3 \cdot x$; $u_5 = \frac{7}{360} \cdot u_4$; $u_6 = u_4 \cdot x$; $u_7 = u_6 \cdot x$; $u_8 = \frac{31}{15120} \cdot u_7$;
$u_9 = u_1 + u_2$; $u_{10} = u_9 + u_5$; $y = u_{11} = u_{10} + u_8$.

Итого : 11 операций, из которых сложных – 6..8.

- *Второй вариант.* Преобразуем выражение (2.1), вынося за скобки сначала первую, затем вторую степень аргумента и части коэффициентов следующим образом :

$$f(x) \approx \frac{1}{x} + \frac{x}{6} \cdot \left[1 + \frac{x^2}{60} \cdot \left(7 + \frac{31}{42} \cdot x^2\right)\right]. \qquad (2.1.1)$$

В результате получаем следующий алгоритм :

$u_1 = \frac{1}{x}$; $u_2 = \frac{x}{6}$; $u_3 = x \cdot x$; $u_4 = \frac{u_3}{60}$; $u_5 = \frac{31}{42} \cdot u_3$; $u_6 = 7 + u_5$; $u_7 = u_4 \cdot u_6$; $u_8 = 1 + u_7$;
$u_9 = u_2 \cdot u_8$; $y = u_{10} = u_1 + u_9$.

Итого : 10 операций, из которых сложных – 6..7.

- *Третий вариант.* Данный вариант сходен с только что рассмотренным, но отличается тем, что выражение $\frac{1}{x}$ вынесено за общую скобку :

$$f(x) \approx \frac{1}{x} \cdot \left[1 + \frac{x^2}{6} \cdot \left[1 + \frac{x^2}{60} \cdot \left(7 + \frac{31}{42} \cdot x^2\right)\right]\right]. \qquad (2.1.2)$$

В результате получим следующий алгоритм :

$u_1 = \frac{1}{x}$; $u_2 = x \cdot x$; $u_3 = \frac{u_2}{60}$; $u_4 = \frac{31}{42} \cdot u_2$; $u_5 = 7 + u_4$; $u_6 = u_3 \cdot u_5$; $u_7 = 1 + u_6$; $u_8 = \frac{u_2}{6}$;
$u_9 = u_8 \cdot u_7$; $u_{10} = 1 + u_9$; $y = u_{11} = u_1 \cdot u_{10}$.

Итого : 11 операций, из них сложных – 7..8.

Делаем вывод, что предпочтение следует отдать второму варианту.

3. Машинный алгоритм и масштабные соотношения

Представим машинный алгоритм, а также масштабные соотношения для каждой из операций в виде таблицы (табл. 3.1). Каждая из операций реализуется с учетом всех возможностей по повышению точности.

Таблица 3.1. Машинные алгоритмы и масштабные соотношения для каждой операции

Операция	Машинный алгоритм	Масштабные соотношения
$u_1 = \dfrac{1}{x}$	1. $\overline{b} := \overline{1}$; 2. $\overline{b} := \overline{b} \cdot 2^{-16}$; 3. $\overline{u}_1 := \dfrac{\overline{b}}{\overline{x}}$	$\mathcal{M}_1 = 2^{-1}$ $\mathcal{M}_X = 2^{-2}$ $\mathcal{M}_{u_1} = 2^{-15} \cdot \dfrac{2^{-1}}{2^{-2}} = 2^{-14}$
$u_2 = \dfrac{x}{6}$	1. $\overline{u}_2 := \overline{0}$; 2. $\overline{b} := \overline{6}$; 3. $\overline{c} := \overline{x}$; 4. $\overline{u}_2[15] := \overline{c}[0]$; 5. $\overline{c} := \overline{c} \cdot 2^{-r}$, *или* $\overline{c} \cdot 2^{-1}$; 6. $\overline{u}_2 := \dfrac{\overline{c}.\overline{u}_2}{\overline{b}}$.	$\mathcal{M}_X = 2^{-2}$ $\mathcal{M}_6 = 2^{-3}$ $\mathcal{M}_{u_2}(u_{2\max}) = \mathcal{M}_{u_2}(0{,}3927) = 2^1$ $r = 16 - \log_2\left(\dfrac{2^1}{\dfrac{2^{-2}}{2^{-3}} \cdot 2^{-15}}\right) = 1$
$u_3 = x \cdot x$	1. $\overline{u}_3 := \overline{x} \cdot \overline{x}$; 2. $\overline{u}_3 := \overline{u}_3 \cdot 2^1$; 3. $\overline{u}_3 := \overline{u}_3 \cdot 2^1$.	$\mathcal{M}_X = 2^{-2}$ $\mathcal{M}_{u_3}(u_{3\max}) = \mathcal{M}_{u_3}(5{,}55) = 2^{-3}$ $k = \log_2\left(\dfrac{2^{-3}}{2^{-2} \cdot 2^{-2} \cdot 2^{-1}}\right) = 2$
$u_4 = \dfrac{u_3}{60}$	1. $\overline{u}_4 := \overline{0}$; 2. $\overline{b} := \overline{60}$; 3. $\overline{c} := \overline{u}_3$; 4. $\overline{u}_4[15] := \overline{c}[0]$; 5. $\overline{c} := \overline{c} \cdot 2^{-r}$, *или* $\overline{c} \cdot 2^{-1}$; 6. $\overline{u}_4 := \dfrac{\overline{c}.\overline{u}_4}{\overline{b}}$.	$\mathcal{M}_{u_3} = 2^{-3}$ $\mathcal{M}_{60} = 2^{-6}$ $\mathcal{M}_{u_4}(u_{4\max}) = \mathcal{M}_{u_4}(0{,}0925) = 2^3$ $r = 16 - \log_2\left(\dfrac{2^1}{\dfrac{2^{-2}}{2^{-3}} \cdot 2^{-15}}\right) = 1$
$u_5 = \dfrac{31}{42} \cdot u_3$ *	1. $\overline{b} := \overline{\left(\dfrac{31}{42}\right)}$; 2. $\overline{u}_5 := \overline{b} \cdot \overline{u}_3$; 3. $\overline{u}_5 := \overline{u}_5 \cdot 2^1$.	$\mathcal{M}_{31/42} = 2^0 = 1$; $\mathcal{M}_{u_3} = 2^{-3}$ $\mathcal{M}_{u_5}(u_{5\max}) = \mathcal{M}_{u_5}(4{,}0977) = 2^{-3}$ $k = \log_2\left(\dfrac{2^{-3}}{1 \cdot 2^{-3} \cdot 2^{-1}}\right) = 1$

* Реализация данной операции умножения на константу методом сдвига и сложения не даст выигрыша в точности или времени исполнения : получаем довольно громоздкое выражение $\overline{u}_5 = \overline{\left(\dfrac{31}{42}\right)} \cdot \overline{u}_3 = 2^{-1}\left(\overline{u}_3 + 2^{-2}\overline{u}_3 + 2^{-3}\overline{u}_3 + 2^{-4}\overline{u}_3 + 2^{-5}\overline{u}_3 + 2^{-8}\overline{u}_3 + 2^{-9}\overline{u}_3 + 2^{-10}\overline{u}_3 + 2^{-11}\overline{u}_3 + 2^{-14}\overline{u}_3\right)$, причем можно показать, что вычисление суммы в скобках без учета сомножителя 1/2 вызвало бы переполнение при максимальном $\overline{u}_{3\max} = [0101100011010011]$, что не позволяет повысить точность этим способом.

Таблица 3.1. Машинные алгоритмы и масштабные соотношения для каждой операции (продолжение)

Операция	Машинный алгоритм	Масштабные соотношения
$u_6 = 7 + u_5$	1. $\overline{b} := \overline{7}$; 2. $\overline{b} := \overline{b} \cdot 2^{-1}$; 3. $\overline{u}_6 := \overline{u}_5$; 4. $\overline{u}_6 := \overline{u}_6 \cdot 2^{-1}$; 5. $\overline{u}_6 := \overline{u}_6 + \overline{b}$	$\mathcal{M}_7 = 2^{-3}$ $\mathcal{M}_{u_5} = 2^{-3}$ $\mathcal{M}_{u_6} = \min\left\{2^{-3}; 2^{-3}; 2^{-4}\right\} = 2^{-4\,*}$ Коэффициенты выравнивания : $K_7 = \dfrac{\mathcal{M}_{u_6}}{\mathcal{M}_7} = \dfrac{2^{-4}}{2^{-3}} = 2^{-1}$ $K_{u_5} = \dfrac{\mathcal{M}_{u_6}}{\mathcal{M}_{u_5}} = \dfrac{2^{-4}}{2^{-3}} = 2^{-1}$
$u_7 = u_4 \cdot u_6$	1. $\overline{u}_7 := \overline{u}_4 \cdot \overline{u}_6$; 2. $\overline{u}_7 := \overline{u}_7 \cdot 2^1$.	$\mathcal{M}_{u_4} = 2^3$ $\mathcal{M}_{u_6} = 2^{-4}$ $\mathcal{M}_{u_7}(u_{7\max}) = \mathcal{M}_{u_7}(1{,}0268) = 2^{-1}$ $k = \log_2\left(\dfrac{2^{-1}}{2^3 \cdot 2^{-4} \cdot 2^{-1}}\right) = 1$
$u_8 = 1 + u_7$	1. $\overline{b} := \overline{1}$; 2. $\overline{b} := \overline{b} \cdot 2^{-1}$; 3. $\overline{u}_8 := \overline{u}_7$; 4. $\overline{u}_8 := \overline{u}_8 \cdot 2^{-1}$; 5. $\overline{u}_8 := \overline{u}_8 + \overline{b}$	$\mathcal{M}_1 = 2^{-1}$ $\mathcal{M}_{u_7} = 2^{-1}$ $\mathcal{M}_{u_8} = \min\left\{2^{-1}; 2^{-1}; 2^{-2}\right\} = 2^{-2\,**}$ Коэффициенты выравнивания : $K_1 = \dfrac{\mathcal{M}_{u_8}}{\mathcal{M}_1} = \dfrac{2^{-2}}{2^{-1}} = 2^{-1}$ $K_{u_7} = \dfrac{\mathcal{M}_{u_6}}{\mathcal{M}_{u_5}} = \dfrac{2^{-2}}{2^{-1}} = 2^{-1}$
$u_9 = u_2 \cdot u_8$	1. $\overline{u}_9 := \overline{u}_2 \cdot \overline{u}_8$; 2. $\overline{u}_9 := \overline{u}_9 \cdot 2^1$; 3. $\overline{u}_9 := \overline{u}_9 \cdot 2^1$.	$\mathcal{M}_{u_2} = 2^1$ $\mathcal{M}_{u_8} = 2^{-2}$ $\mathcal{M}_{u_9}(u_{9\max}) = \mathcal{M}_{u_9}(0{,}7959) = 2^0 = 1$ $k = \log_2\left(\dfrac{2^0}{2^1 \cdot 2^{-2} \cdot 2^{-1}}\right) = 2$
$y = u_{10} = u_1 + u_9$	1. $\overline{b} := \overline{u}_9$; 2. $\overline{b} := \overline{b} \cdot 2^{-14}$; 3. $\overline{u}_{10} := \overline{u}_1$; 4. $\overline{u}_{10} := \overline{u}_{10} + \overline{b}$.	$\mathcal{M}_{u_1} = 2^{-14}$ $\mathcal{M}_{u_9} = 1$ $\mathcal{M}_{u_{10}} = \min\left\{2^{-14}; 2^0; 2^{-14}\right\} = 2^{-14}$ Коэффициенты выравнивания : $K_{u_1} = \dfrac{2^{-14}}{2^{-14}} = 1;\ K_{u_9} = \dfrac{2^{-14}}{2^0} = 2^{-14}.$

---

* Третье значение масштаба для выбора (2^{-4}) рассчитывается исходя из максимального значения $u_{6\max} \approx 11{,}097648$.

** Третье значение масштаба для выбора (2^{-2}) рассчитывается исходя из максимального значения $u_{8\max} \approx 2{,}0268$.

Сделаем одно замечание. Для правильного моделирования работы контура САУ с ЦВМ в цепи обратной связи на языке высокого уровня (например, на языке Turbo Pascal) необходимо осуществить перевод дробных масштабов аргумента и результата в целочисленные. Для этого необходимо соответствующие дробные двоичные масштабы умножить на величину $2^n = 2^{15} = 32768$. Таким образом, получаем :

$$M_X = 2^{-2} \cdot 2^{15} = 2^{13};$$
$$M_Y = 2^{-14} \cdot 2^{15} = 2^{1}.$$

Машинное представление констант (с учетом соответствующих константам двоичных масштабов) имеет следующий вид :

$$\overline{1} = 4000h; \overline{6} = 6000h; \overline{60} = 7800h; \overline{\left(\frac{31}{42}\right)} = 5E79h; \overline{7} = 7000h.$$

4. Расчет математического ожидания и дисперсии ошибки

Разобьем весь интервал изменения аргумента на 1024 подынтервала.

Абсолютная ошибка между i-м значением реализации функции на языке ассемблера и эталонной функцией может быть вычислена следующим образом :

$$E_i = \left| y_i - f_i \right|, \qquad (4.1)$$

где y_i - текущее значение реализуемой функции, а f_i - текущее значение эталонной функции (первые четыре члена ряда).

Математическое ожидание ошибки реализации можно подсчитать по формуле :

$$M[E] = \frac{\sum_i E_i}{2n}, \qquad (4.2)$$

После прогона программы получаем значение математического ожидания ошибки, приблизительно равное 0,468.

Дисперсию (квадрат среднеквадратического отклонения) ошибки реализации можно подсчитать по формуле :

$$D[E] = M\left[E^2\right] - M^2[E]. \qquad (4.3)$$

После прогона программы получаем значение дисперсии ошибки, приблизительно равное 0,0625.

Максимально зафиксированное программой значение функции равно 8192,5, а ошибки - 0,715.

Приложение П1. Текст программы на языке Turbo Pascal со вставкой фрагмента на языке ассемблера

```
program Alex2pas;
Uses Graph, CRT;
const
  XPhysMax = 3*Pi/4;  {Наибольшее по модулю физическое значение аргумента}
  XMachMax = 3*Pi/16; {Наибольшее по модулю машинное значение аргумента}
  AlgScaleX = 8192;   {Масштаб аргумента в целочисленной арифметике}
  AlgScaleY = 2;      {Масштаб результата в целочисленной арифметике}
  GraphScaleY = 480/136.5; {Масштаб функции графический}
  GraphScaleError = 240/1.1866; {Масштаб ошибки графический}
  n = 320;  {Количество точек графика по одну сторону от начала координат}
var
  CurMachX, CurMachY : integer;
  {Машинные значения аргумента и реализации функции}
  CurPhysX, {Физическое значение аргумента}
  CurPhysY, {Физическое значение реализации функции}
  CurError, {Текущее значение ошибки реализации}
  AverageError, {Суммарное, а впоследствии - среднее значение ошибки
реализации}
  AverageSquaredError,
  {Суммарное, а впоследствии - среднее значение квадрата ошибки реализации}
  Disperse : double;  {Дисперсия ошибки реализации}
  CurTemplateYRowSum, CurTemplateYOriginal : double;
{Текущие значения суммы ряда и функции косеканса csc(x), вычисленные при}
{помощи средств языка высокого уровня}
  TmpTemplateX2 : double; {Промежуточное выражение, содержащее квадрат
{физического значения аргумента}
  grdrv,grmode : integer; {Номера графического драйвера и режима}
  k : word;
  i : integer;            {Переменные, используемые при организации циклов}
  Y_gr_prev, Error_gr_prev,Templ_gr_prev : integer;
  {Значения реализации функции на ассемблере, ошибки и четырех членов ряда}
  {в графических координатах, вычисленные в предыдущей итерации цикла}
  {построения графика}
  breakline : boolean; {Признак того, что в процессе вычислений имел место}
  {нулевой результат, и необходимо прервать линию графика}

function AsmCalc(X : integer) : integer;
{Возвращает машинное значение реализуемой функции в точке X}
const
    const1 = $4000;
    const6 = $6000;
    const60 = $7800;
    const31_42 = $5E79;
    const7 = $7000;
var
  u1, u2, u3, Y : integer;
  label AboveZero;
  label Next;
begin
     asm
      {u1 = 1/x}
      mov     ax, const1
      cwd
      mov     bx, X
      idiv    bx
      mov     u1, ax
      {u2 = x/6}
      xor     ax, ax
      mov     dx, X
      mov     bx, const6
      sar     dx, 1
      rcr     ax, 1
      idiv    bx
      mov     u2, ax
      {u3 = x*x}
```

```
        mov   ax, X
        mov   bx, X
        imul  bx     {результат - в dx : ax}
        sal   ax, 1
        rcl   dx, 1
        sal   ax, 1
        rcl   dx, 1
        mov   u3, dx

        {u4(cx)=u3/60}
        xor     ax, ax
        mov     dx, u3
        mov     bx, const60
        sar     dx, 1
        rcr     ax, 1
        idiv    bx
        mov     cx, ax
        {u5=(31/42)*u3}
        mov   ax, const31_42
        mov   bx, u3
        imul  bx     { результат - в dx : ax}
        sal   ax, 1
        rcl   dx, 1  {16-битный результат - в in dx}

        {u6(ax) = 7+u5=7+[dx]}
        mov   ax, const7
        sar   ax, 1
        sar   dx, 1
        add   ax, dx {result in ax}

        {u7(dx) = u4(cx)*u6}
        imul  cx    {result in dx : ax}
        sal   ax, 1
        rcl   dx, 1 {16-bit result in dx}

        {u8(ax) = 1+u7(dx)}
        mov   ax, const1
        sar   ax, 1
        sar   dx, 1
        add   ax, dx {result in ax}

        {u9(dx)=u2*u8(ax)}
        mov   bx, u2
        imul  bx    {result in dx : ax}
        sal   ax, 1
        rcl   dx, 1
        sal   ax, 1
        rcl   dx, 1 {16-bit result in dx}

        {u10 = u1 + u9(dx)}
        mov   ax, u1
        cmp   dx, 0
        jge   AboveZero
        mov   cx, 0FFFFh
        jmp   Next
AboveZero:
        xor   cx, cx
Next:
        sal   dx, 1
        rcl   cx, 1
        sal   dx, 1
        rcl   cx, 1
        add   ax, cx
        mov   Y, ax
      end;
      AsmCalc := Y;
end;
```

```
procedure draw_axes;
begin
  setcolor(10);
  line(0,240,640,240);
  line(320,0,320,480);
  setcolor(15);
  outtextxy(323,240+3,'0');
  setcolor(13);
  outtextxy(330,260,'Теоретический csc(x)');
  setcolor(11);
  outtextxy(330,270,'По степенному ряду');
  setcolor(12);
  outtextxy(330,280,'На ассемблере');
  setcolor(15);
  outtextxy(330,290,'Абсолютная погрешность');
end;

procedure RecalcY_Error(PhysX : double);
begin
  CurMachX := PhysX*AlgScaleX;
  CurMachY := AsmCalc(CurMachX);
  CurPhysY := CurMachY / AlgScaleY;
  TmpTemplateX2 := PhysX * PhysX;
  CurTemplateYRowSum :=
    (1/PhysX)+(1/6)*PhysX+(7/360)*TmpTemplateX2*PhysX+
    + (31/15120)*TmpTemplateX2*TmpTemplateX2*PhysX;
  CurError := abs(CurTemplateYRowSum - CurPhysY);
  Averageerror := Averageerror + CurError;
  AverageSquaredError := AverageSquaredError + CurError*CurError;
end;

begin {Главный модуль}
  grdrv := 9;
  grmode := 2;
  breakline := true;
  initgraph(grdrv, grmode,'c:\bp\bgi\');
  draw_axes;
  for i := -n to n-1 do
  begin
    CurPhysX := XPhysMax*i/n;
    if (CurPhysX <> 0) then
    begin
     CurMachX := trunc(CurPhysX*AlgScaleX);
     if (CurMachX <> 0) then
     begin
      inc(k);
      CurMachY := AsmCalc(CurMachX);
      CurPhysY := CurMachY / AlgScaleY;
      TmpTemplateX2 := CurPhysX * CurPhysX;
      CurTemplateYOriginal := 1 / sin(CurPhysX);
      CurTemplateYRowSum :=
      (1/CurPhysX)+(1/6)*CurPhysX+(7/360)*TmpTemplateX2*CurPhysX+
      + (31/15120)*TmpTemplateX2*TmpTemplateX2*CurPhysX;
      CurError := abs(CurTemplateYRowSum - CurPhysY);
     {Строим точку графика csc(x)}
      putpixel(i+320, 240 - round(CurTemplateYOriginal*GraphScaleY),13);
     {Строим линию эталонного графика суммы четырех членов ряда}
      setcolor(11);
      if (breakline) then
      begin
       putpixel(i+320, 240-round(CurTemplateYRowSum*GraphScaleY),11);
      end
      else
       line(i+319,Templ_gr_prev,i+320,
        240 - round(CurTemplateYRowSum*GraphScaleY));
      {Строим линию графика реализации функции на языке ассемблера}
      setcolor(12);
      if (breakline) then
      begin
```

```
     putpixel(i+320, 240 - round(CurPhysY*GraphScaleY),12);
    end
    else
    begin
     line(i+319,Y_gr_prev,i+320, 240 - round(CurPhysY*GraphScaleY));
    end;
    {Строим линию графика абсолютной погрешности (ошибки реализации)}
    setcolor(15);
    if (breakline) then
    begin
     putpixel(i+320, 240 - round(CurError*GraphScaleError),15);
     breakline := false
    end
    else
    begin
     line(i+319,Error_gr_prev,i+320,
      240 - round(CurError*GraphScaleError));
    end;
    Templ_gr_prev := 240 - round(CurTemplateYRowSum*GraphScaleY);
    Y_gr_prev := 240 - round(CurPhysY*GraphScaleY);
    error_gr_prev := 240 - round(CurError*GraphScaleError);
   end
   else
   begin
    breakline := true
   end;
  end
  else
  begin
   breakline := true
  end;
 end;
 readln;
 closegraph;
 k := 0;
 AverageError := 0;
 AverageSquaredError := 0;
 CurPhysX := 1/8192;
 while (CurPhysX <= XPhysMax) do
 {В данном цикле вычисляются суммы значений абсолютной ошибки}
 {и квадрата абсолютной ошибки в переменных AverageError и
SquaredAverageError}
 begin
  RecalcY_Error(CurPhysX);
  inc(k);
  RecalcY_Error(-CurPhysX);
  inc(k);
  CurPhysX := CurPhysX + 3*Pi/1024;
 end;
 AverageError := AverageError / k;
 writeln(' Математическое ожидание ошибки = ',AverageError : 10 : 10);
 AverageSquarederror := AverageSquaredError/k;
 Disperse := AverageSquarederror - Averageerror*Averageerror;
 writeln('Дисперсия ошибки = ',Disperse : 10 : 10);
 readln;
end.
```

Приложение П2. Графики : эталонной функции, ошибки и реализации функции на языке ассемблера

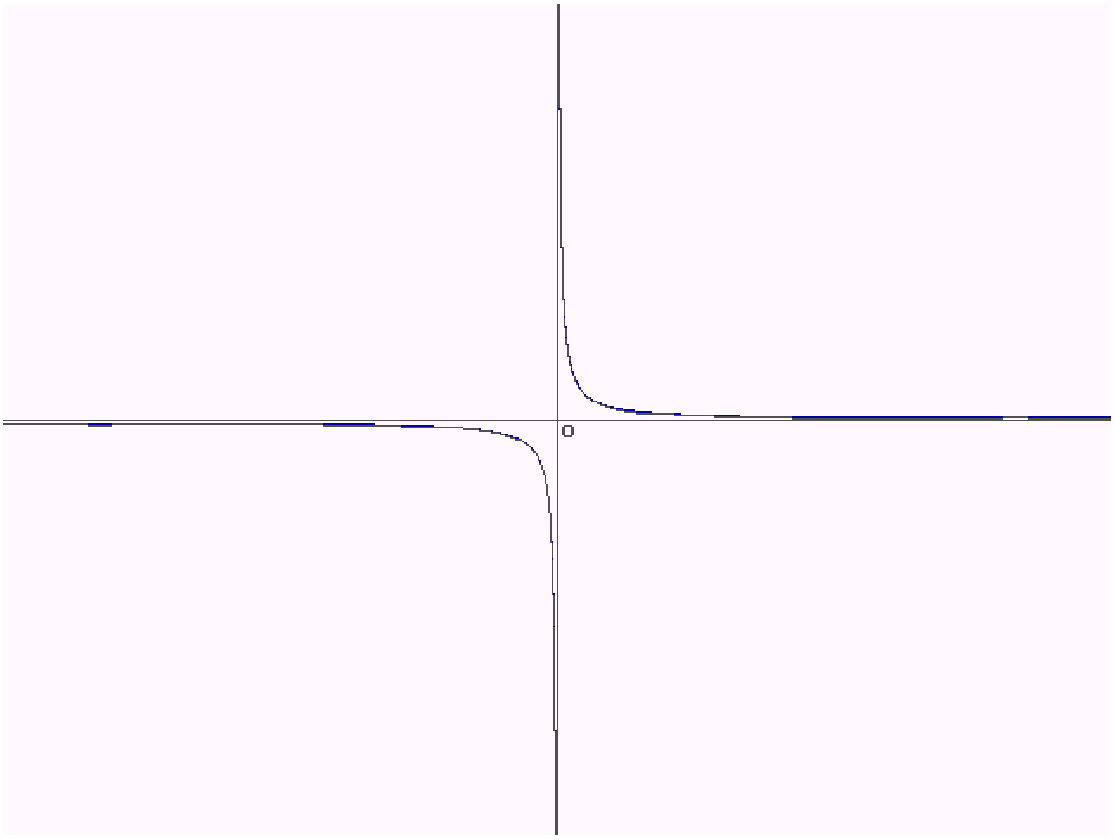

Рис. П.2.1. Графики csc(x), первых четырех членов ряда и реализации на ассемблере. Максимальное значение функции по модулю равно 136,5. Полный интервал изменения аргумента.

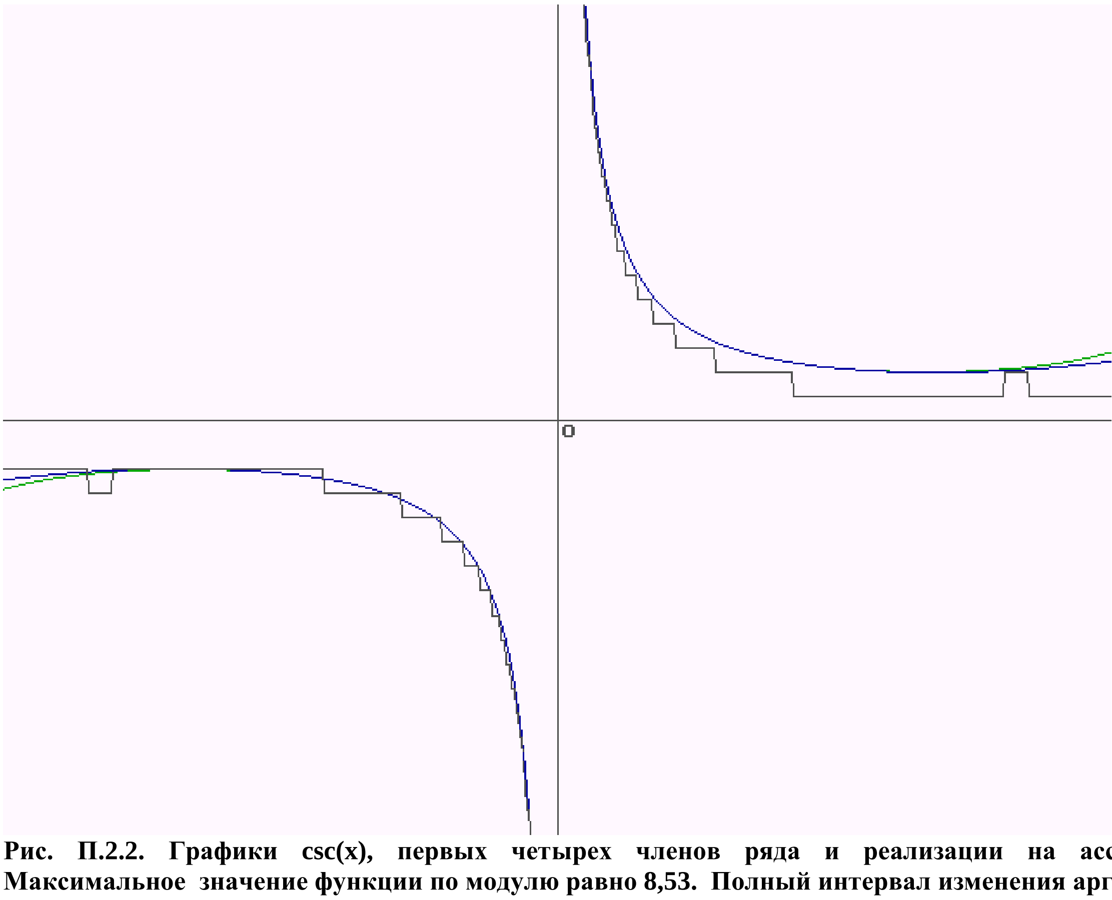

Рис. П.2.2. Графики csc(x), первых четырех членов ряда и реализации на ассемблере. Максимальное значение функции по модулю равно 8,53. Полный интервал изменения аргумента

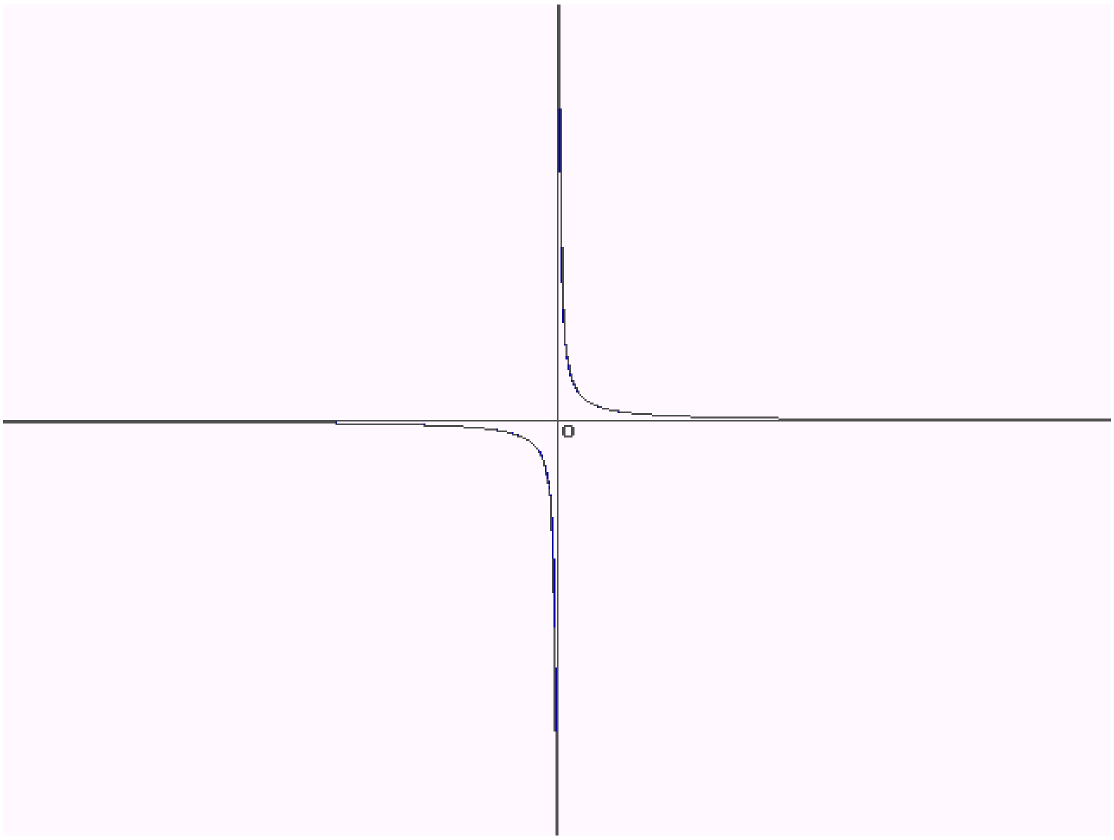

Рис. П.2.3. Графики csc(x), первых четырех членов ряда и реализации на ассемблере. Максимальное значение функции по модулю равно 8193,42 (это приблизительно соответствует достижению максимально реализуемого значения). Интервал изменения аргумента показан частично (сокращен в 96 раз), т.е. $\left[-\frac{3\pi}{192};\frac{3\pi}{192}\right]$.

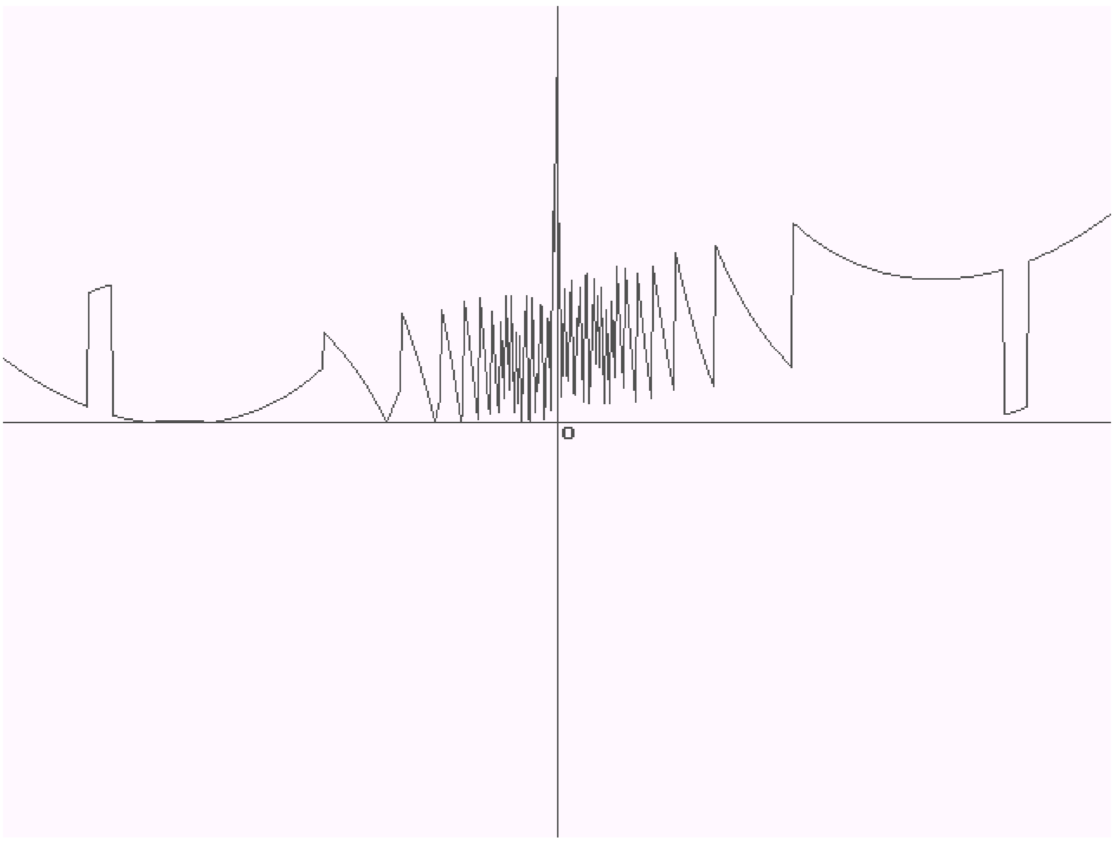

Рис. П.3.3. График ошибки. Максимальное значение ошибки примерно равно 0,715. Интервал изменения аргумента показан полностью