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

Отчет

по лабораторной работе № 3

«Реализация и исследование решения дифференциального уравнения методом Эйлера 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.

Задание (вариант № 25)

- Разработать математический и машинный алгоритм решения дифференциального уравнения. методом Эйлера, рассчитать масштабы, написать и отладить программу на ассемблере, реализующую разработанный машинный алгоритм.

- Исследовать зависимость точности решения от величины шага (3, 4 шага; в качестве критерия точности взять дисперсии ошибок). Построить гистограммы зависимости дисперсий ошибок от шага интегрирования.

- За эталонное значение взять реализацию точного решения заданного уравнения.

- Исходные данные :

  - уравнение : $y''' - 2y'' + y = 0$;
  - интервал интегрирования : 0…1,2;
  - начальные условия :
    $y(0) = 0,5; y'(0) = 0; y''(0) = 0; y'''(0) = -y(0) + 2y''(0) = -0,5;$
  - шаг интегрирования h принять равным
    $2^{-4}(0,0625); 2^{-5}(0,03125); 2^{-6}(0,015625).$

Аналитическое (точное) решение заданного дифференциального уравнения

Решение данного однородного дифференциального уравнения имеет следующий вид:

$y(x) = C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x} + C_3 e^{\lambda_3 x}$, где λ_1, λ_2 и λ_3 - корни характеристического уравнения $\lambda^3 - 2\lambda^2 + 1 = 0$. Решая уравнение, получим: $\lambda_1 = 1$; $\lambda_2 = 0,5 + \frac{\sqrt{5}}{2} \approx 1,618$; $\lambda_3 = 0,5 - \frac{\sqrt{5}}{2} \approx -0,618$.

Коэффициенты C_1, C_2 и C_3 определяются исходя из начальных условий. Имеем следующую систему уравнений :

$$y(0) = C_1 + C_2 + C_3 = 0,5;$$

$$y'(0) = C_1 + C_2\left(0,5 + \frac{\sqrt{5}}{2}\right) + C_3\left(0,5 - \frac{\sqrt{5}}{2}\right) = 0;$$

$$y''(0) = C_1 + C_2\left(0,5 + \frac{\sqrt{5}}{2}\right)^2 + C_3\left(0,5 - \frac{\sqrt{5}}{2}\right)^2 = 0.$$

Решив эту систему, получим следующие значения постоянных :

$C_1 = 0,5$; $C_2 \approx -0.2236$; $C_3 \approx 0.2236$.

В результате искомое решение запишется следующим образом :

$y(x) \cong 0{,}5e^{x} - 0{,}2236e^{1{,}618x} + 0{,}2236e^{-0{,}618x}$.

Определим абсолютно максимальные значения для решения и первых трех его производных на заданном интервале (так как все эти функции – монотонно убывающие, то максимальное по модулю значение следует искать на одном из концов интервала) :

$|y|_{\max} = |y(0)| = 0{,}5$; $|y'|_{\max} = |y'(1{,}2)| \approx 0{,}9276$; $|y''|_{\max} = |y''(1{,}2)| \approx 2{,}3797$; $|y'''|_{\max} = |y'''(1{,}2)| = 4{,}9673$.

Математический алгоритм

Введем следующие обозначения: $y_1 = y'(x); y_2 = y''(x); y_3 = y'''(x)$.
Для решения д.у. весь интервал разбивается на n участков с величиной шага h. Решение для каждого участка можно найти по следующим формулам:

$$y_{3,k} = 2y_{2,k} - y_k \;;$$
$$y_{2,k+1} = y_{2,k} + h\, y_{3,k} \;;$$
$$y_{1,k+1} = y_{1,k} + h\, y_{2,k} \;;$$
$$y_{k+1} = y_k + h\, y_{1,k} \;.$$

На каждой итерации необходимо отслеживать момент достижения правой границы интервала. $y_{1,k}$, $y_{2,k}$ и y_k являются исходными значениями для данной итерации k+1. $y_{1,k+1}$, $y_{2,k+1}$ и y_{k+1} будут являться исходными значениями для следующей итерации k+2.

Таким образом, чтобы получить решение д.у. методом Эйлера, необходимо произвести n итераций для вычисления $y_{1,k}$ и $y_{2,k}$, где k=1..n, а $n = \frac{1{,}2}{h}$. Для первой итерации исходными данными будут являться начальные значения.

Составим список используемых простейших арифметических операций:

- $z_1 = 2 \cdot y_{2,k}$
- $z_2 = z_1 - y_k$ (z_2 – значение $y_{3,k}$.)
- $z_3 = h\, y_{3,k}$
- $z_4 = z_3 + y_{2,k}$ (z_4 - значение $y_{2,k+1}$)
- $z_5 = h\, y_{2,k}$
- $z_6 = z_5 + y_{1,k}$ (z_6 - значение $y_{1,k+1}$)
- $z_7 = h\, y_{1,k}$
- $z_8 = z_7 + y_k$ (z_8 - значение y_{k+1})
- $z_9 = x_k + h$ (z_9 - значение x_{k+1}).

Машинный алгоритм и масштабирование

Представим машинный алгоритм, а также масштабные соотношения для каждой из операций в виде таблицы. Каждая из операций реализуется с учетом всех возможностей по повышению точности. Последовательность этих операций необходимо выполнять в цикле n раз. Для сокращения записи индекс k у переменных писать не будем. Масштаб шага h вычисляется исходя из наибольшего значения шага 2^{-4}, а сам шаг считаем переменной величиной.

Таблица Машинные алгоритмы и масштабные соотношения для каждой операции

Операция	Машинный алгоритм	Масштабные соотношения
$z_1 = 2 \cdot y_2$	$\bar{z}_1 := \bar{y}_2$	$\mathcal{M}_{y_2} = 2^{-2}$ $\mathcal{M}_{z_1} = \mathcal{M}_{y_2} 2^{-1} = 2^{-3}$
$z_2 = z_1 - y$	1. $\bar{b} := \bar{z}_1$; 2. $\bar{z}_2 := -\bar{y}$; 3. $\bar{z}_2 := \bar{z}_2 \cdot 2^{-3}$; 4. $\bar{z}_2 := \bar{z}_2 + \bar{b}$	$\mathcal{M}_{z_1} = 2^{-3}$ $\mathcal{M}_{y} = 2^{0} = 1$ $\mathcal{M}_{z_2} = \min\{2^{-3}; 2^{0}; 2^{-3}\} = 2^{-3}$ * Коэффициенты выравнивания : $K_{z_1} = \dfrac{\mathcal{M}_{z_2}}{\mathcal{M}_{z_1}} = \dfrac{2^{-3}}{2^{-3}} = 2^{0} = 1$ $K_{y} = \dfrac{\mathcal{M}_{z_2}}{\mathcal{M}_{y}} = \dfrac{2^{-3}}{2^{0}} = 2^{-3}$
$z_3 = h \cdot y_3$	1. $\bar{z}_3 := \bar{h} \cdot \bar{y}_3$; 2. $\bar{z}_3 := \bar{z}_3 \cdot 2^{2}$.	$\mathcal{M}_{y_3} = 2^{-3}$ $\mathcal{M}_{h} = 2^{3}$ $\mathcal{M}_{z_3} = \mathcal{M}\left(\lvert z_3 \rvert_{\max}\right) = \mathcal{M}(0{,}3105) = 2^{1}$ $k = \log_2\left(\dfrac{2^{1}}{2^{-3} \cdot 2^{3} \cdot 2^{-1}}\right) = 2$
$z_4 = z_3 + y_2$	1. $\bar{b} := \bar{z}_3$; 2. $\bar{b} := \bar{b} \cdot 2^{-3}$; 3. $\bar{z}_4 := \bar{y}_2$; 4. $\bar{z}_4 := \bar{z}_4 + \bar{b}$	$\mathcal{M}_{z_3} = 2^{1}$ $\mathcal{M}_{y_2} = 2^{-2}$ $\mathcal{M}_{z_4} = \min\{2^{1}; 2^{-2}; 2^{-2}\} = 2^{-2}$ *** Коэффициенты выравнивания : $K_{z_3} = \dfrac{\mathcal{M}_{z_4}}{\mathcal{M}_{z_3}} = \dfrac{2^{-2}}{2^{1}} = 2^{-3}$ $K_{y_2} = \dfrac{\mathcal{M}_{z_4}}{\mathcal{M}_{y_2}} = \dfrac{2^{-2}}{2^{-2}} = 2^{0} = 1$

* Третий масштаб в фигурных скобках выбирается исходя из $\lvert z_2 \rvert_{\max} = \lvert y_3 \rvert_{\max} = 4{,}9673$

*** Третий масштаб в фигурных скобках выбирается исходя из $\lvert z_4 \rvert_{\max} = \lvert y_2 \rvert_{\max} = 2{,}3797$

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

Операция	Машинный алгоритм	Масштабные соотношения
$z_5 = h \cdot y_2$	1. $\bar{z}_5 := \bar{h} \cdot \bar{y}_2$; 2. $\bar{z}_5 := \bar{z}_5 \cdot 2^2$.	$\mathcal{M}_{y_2} = 2^{-2}$ $\mathcal{M}_h = 2^3$ $\mathcal{M}_{z_5} = \mathcal{M}\left(\lvert z_5\rvert_{\max}\right) = \mathcal{M}(0{,}1487) = 2^2$ $k = \log_2\left(\dfrac{2^2}{2^{-2} \cdot 2^3 \cdot 2^{-1}}\right) = 2$
$z_6 = z_5 + y_1$	1. $\bar{b} := \bar{z}_5$; 2. $\bar{b} := \bar{b} \cdot 2^{-2}$; 3. $\bar{z}_6 := \bar{y}_1$; 4. $\bar{z}_6 := \bar{z}_6 + \bar{b}$	$\mathcal{M}_{z_5} = 2^2$ $\mathcal{M}_{y_1} = 2^0$ $\mathcal{M}_{z_6} = \min\{2^2; 2^0; 2^0\} = 2^0$ * Коэффициенты выравнивания : $K_{z_5} = \dfrac{\mathcal{M}_{z_6}}{\mathcal{M}_{z_5}} = \dfrac{2^0}{2^2} = 2^{-2}$ $K_{y_1} = \dfrac{\mathcal{M}_{z_6}}{\mathcal{M}_{y_1}} = \dfrac{2^0}{2^0} = 2^0 = 1$
$z_7 = h \cdot y_1$	1. $\bar{z}_7 := \bar{h} \cdot \bar{y}_1$; 2. $\bar{z}_7 := \bar{z}_7 \cdot 2^2$.	$\mathcal{M}_{y_1} = 2^0$ $\mathcal{M}_h = 2^3$ $\mathcal{M}_{z_7} = \mathcal{M}\left(\lvert z_7\rvert_{\max}\right) = \mathcal{M}(0{,}058) = 2^4$ $k = \log_2\left(\dfrac{2^4}{2^0 \cdot 2^3 \cdot 2^{-1}}\right) = 2$
$z_8 = z_7 + y$	1. $\bar{b} := \bar{z}_7$; 2. $\bar{b} := \bar{b} \cdot 2^{-4}$; 3. $\bar{z}_8 := \bar{y}$; 4. $\bar{z}_8 := \bar{z}_8 + \bar{b}$	$\mathcal{M}_{z_7} = 2^4$ $\mathcal{M}_y = 2^0$ $\mathcal{M}_{z_8} = \min\{2^4; 2^0; 2^0\} = 2^0$ ** Коэффициенты выравнивания : $K_{z_7} = \dfrac{\mathcal{M}_{z_8}}{\mathcal{M}_{z_7}} = \dfrac{2^0}{2^4} = 2^{-4}$ $K_y = \dfrac{\mathcal{M}_{z_8}}{\mathcal{M}_y} = \dfrac{2^0}{2^0} = 2^0 = 1$

[*] Третий масштаб в фигурных скобках выбирается исходя из $\lvert z_6\rvert_{\max} = \lvert y_1\rvert_{\max} = 0{,}9276$

[**] Третий масштаб в фигурных скобках выбирается исходя из $\lvert z_8\rvert_{\max} = \lvert y\rvert_{\max} = 0{,}5$

Сделаем 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 :
$M_x = 2^{-1} \cdot 2^{15} = 2^{14}$; $M_h = 2^3 \cdot 2^{15} = 2^{18}$; $M_y = 2^0 \cdot 2^{15} = 2^{15}$; $M_{y1} = 2^0 \cdot 2^{15} = 2^{15}$;
$M_{y2} = 2^{-2} \cdot 2^{15} = 2^{13}$; $M_{y3} = 2^{-3} \cdot 2^{15} = 2^{12}$.

Если рассматривать приведенный выше машинный алгоритм как единое целое, то можно заметить, что некоторые операции в нем можно опустить (например, если производится сначала сдвиг вправо, а затем – сдвиг влево на одно и то же число разрядов, а промежуточная переменная используется только в следующей операции). Поэтому окончательно машинный алгоритм будет выглядеть следующим образом (введены следующие обозначения : $u_1 = z_7$, $u_2 = z_8$, $u_3 = z_2$, $u_4 = z_3$, $u_5 = z_4$, $u_6 = z_5$, $u_7 = z_6$; Y – формируемый массив машинных значений Y; j – указатель смещения в массиве Y; i – счетчик цикла) :

- Перед циклом :

$i := n$; $j := 0$; $Y[0] := \bar{y}$; $j := j + 2$

- **начало цикла**

$\bar{u}_1 := \bar{h}$; $\bar{u}_1 := \bar{u}_1 \cdot \bar{y}_1$; $\bar{u}_1 := \bar{u}_1 \cdot 2^{-2}$.

$\bar{u}_2 := \bar{y}$; $\bar{u}_2 := \bar{u}_2 + \bar{u}_1$; $Y[j] := \bar{u}_2$;

$j := j + 2$; $i := i - 1$;

- Если $i \le 0$, то **конец цикла**; иначе :

$\bar{y} := \bar{u}_2$; $\bar{u}_3 := -\bar{y}$; $\bar{u}_3 := \bar{u}_3 \cdot 2^{-3}$; $\bar{u}_3 := \bar{u}_3 + \bar{y}_2$

$\bar{u}_4 := \bar{h}$; $\bar{u}_4 := \bar{u}_4 \cdot \bar{y}_3$; $\bar{u}_4 := \bar{u}_4 \cdot 2^{-1}$

$\bar{u}_5 := \bar{y}_2$; $\bar{u}_5 := \bar{u}_5 + \bar{u}_4$; $\bar{y}_2 := \bar{u}_5$

$\bar{u}_6 := \bar{h}$; $\bar{u}_6 := \bar{u}_6 \cdot \bar{y}_2$;

$\bar{u}_7 := \bar{y}_1$; $\bar{u}_7 := \bar{u}_7 + \bar{u}_6$; $\bar{y}_1 := \bar{u}_7$

- Перейти к : **начало цикла**

- **конец цикла**

Расчет математического ожидания и дисперсии ошибки

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

$$E_i = y_i - f_i ,$$

где y_i - текущее значение реализуемой функции, а f_i - текущее значение эталонной функции.

Математическое ожидание ошибки реализации можно подсчитать по формуле :

$$M[E] = \frac{\sum_{i=0}^{l-1} E_i}{l},$$

l– количество точек, в которых вычисляются значения математического ожидания, это значение на единицу больше, чем количество подынтервалов n, на которые разбивается весь интервал интегрирования.

После прогона программы получаем следующие приближенные значения математического ожидания ошибки : 0,0289 (шаг равен 1/16); 0,0155 (шаг равен 1/32); 0,0073 (шаг равен 1/64).

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

$$D[E] = M[E^2] - M^2[E].$$

После прогона программы получаем следующие значения дисперсии ошибки : 0,00113 (шаг равен 1/16); 0,00031 (шаг равен 1/32); 0,00007 (шаг равен 1/64).

Максимальное значение функции $y(x)$ на заданном интервале равно 0,5 , а ошибки - соответственно 0,1128 (шаг равен 1/16); 0,0619 (шаг равен 1/32); 0,0301 (шаг равен 1/64). Минимальное значение функции на рассматриваемом интервале равно 0,2194.

Выводы :

Из гистограммы дисперсии ошибки (см. рис. 4) видно, что с уменьшением величины шага происходит повышение точности (дисперсия ошибки уменьшается), причем с уменьшением величины шага скорость изменения дисперсии ошибки уменьшается. По виду кривой нарастания ошибки можно заключить, что характер функции ошибки близок к показательному (возможно, экспоненциальному) закону. Во всех трех случаях ошибка, ее математическое ожидание и дисперсия находятся в допустимых диапазонах (относительное усредненное значение ошибки не превышает 6% даже в случае h = 1/16). Следовательно, метод Эйлера в рассматриваемом примере дает неплохие результаты наряду с его достаточно простой реализацией (по сравнению с методами Адамса, Рунге-Кутта и.т.д.).

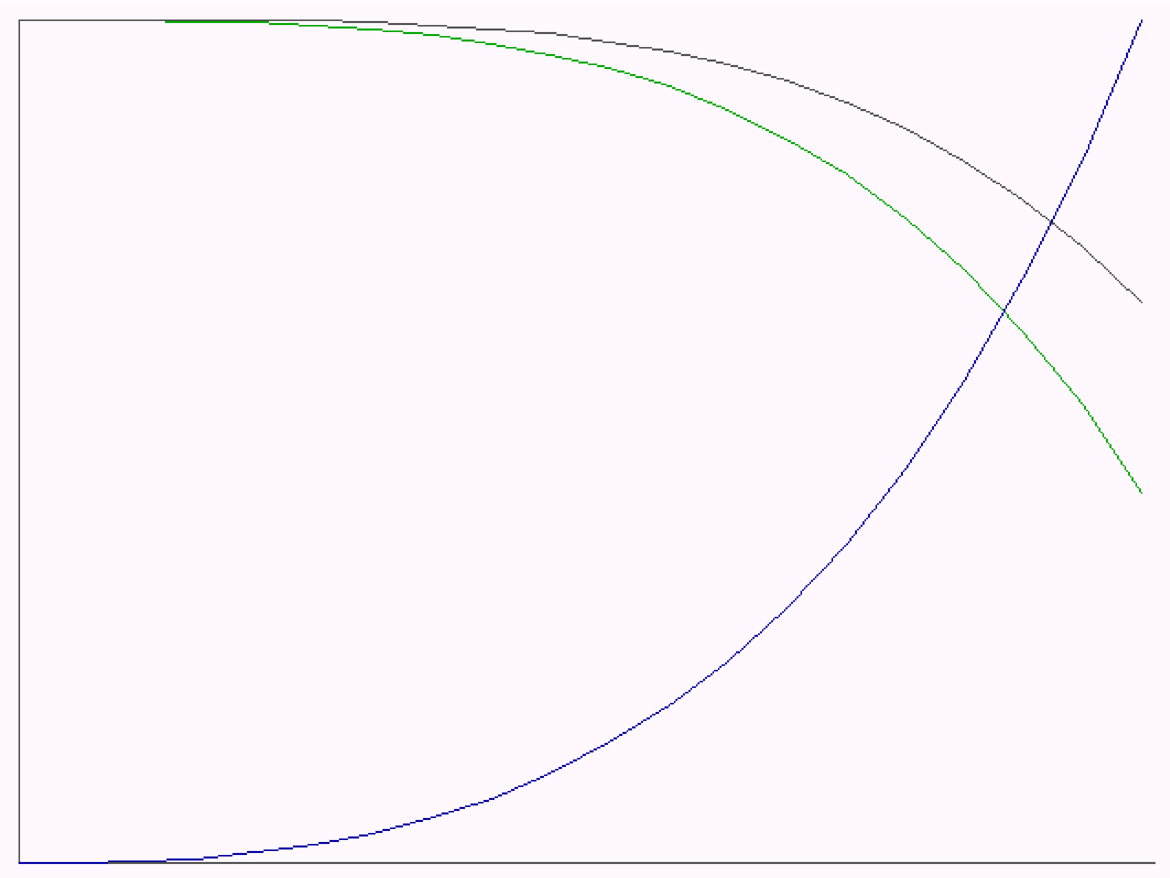

Рис. 1. Графики эталонного решения, результата интегрирования уравнения на языке ассемблера и ошибки численного интегрирования. Шаг h = 1/16.

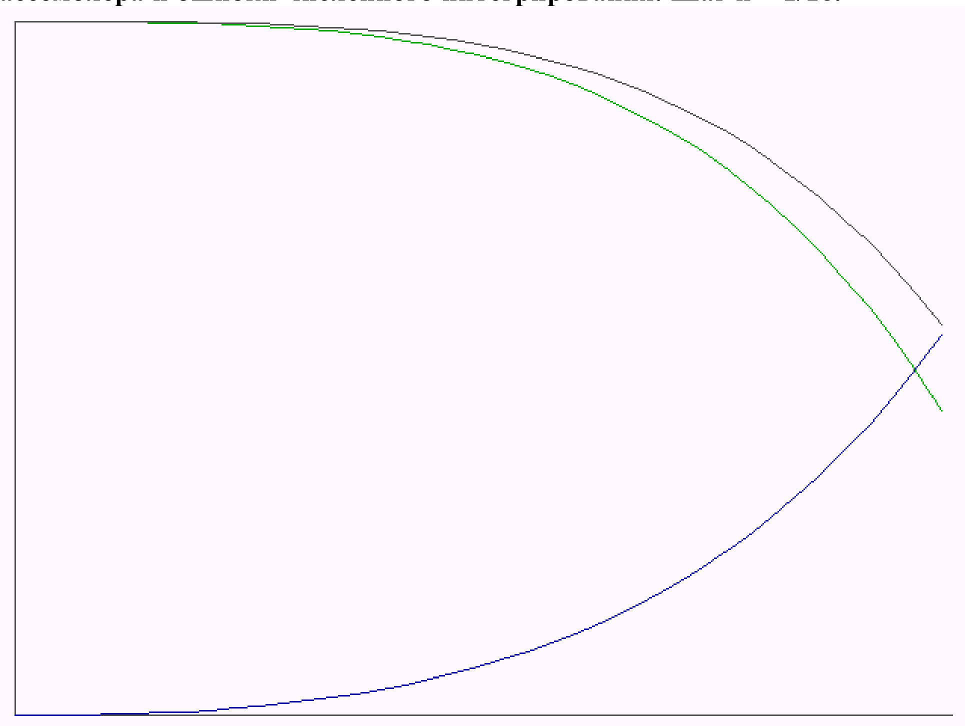

Рис. 2. Графики эталонного решения, результата интегрирования уравнения на языке ассемблера и ошибки численного интегрирования. Шаг h = 1/32.

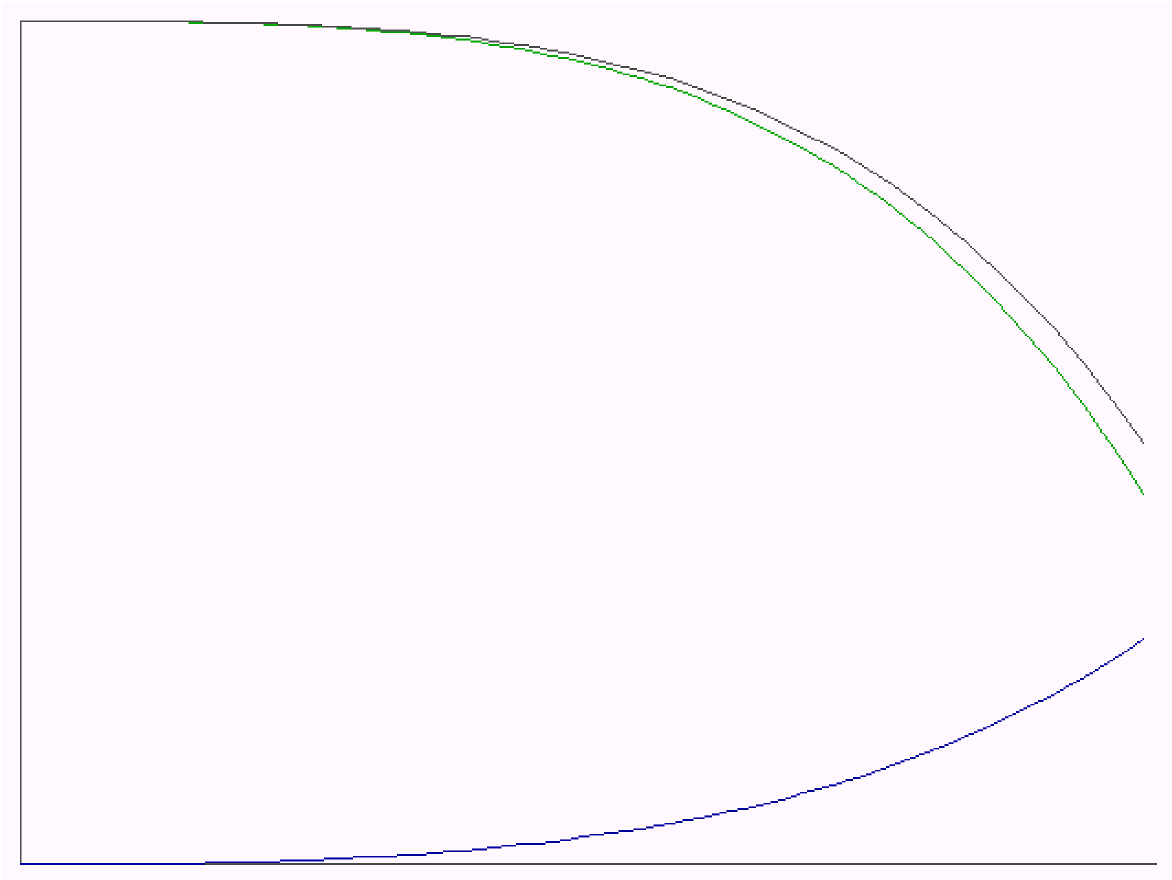

Рис. 3. Графики эталонного решения, результата интегрирования уравнения на языке ассемблера и ошибки численного интегрирования. Шаг h = 1/64.

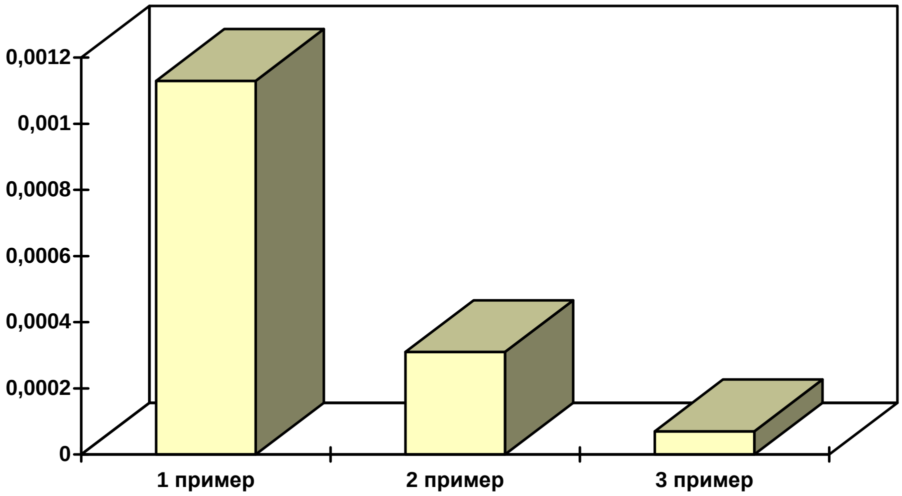

Рис. 4. Гистограмма значений дисперсии в зависимости от величины шага интегрирования.

	1 пример	2 пример	3 пример
Величина шага h	0,0625	0,03125	0,015625
Число шагов n	19	38	76
Дисперсия D	0,0011298323	0,003144875	0,0000716355

Листинг ядра программы (процедура вычисления машинных значений результата) :

```
procedure AsmCalc(N,Y,Y1,Y2 : integer);
label Next, Finish;
begin
      asm
        mov     cx, N
        xor     si, si
        mov     ax, Y
        mov     word ptr ds : MachValsY [si], ax
        or      si, 2
        mov     ax, Y1
        mov     dx, HMach
Next:
        {u1(dx) = h*y1}
        imul      dx  {dx : ax}
        sar       dx, 1
        sar       dx, 1
        {u2(ax) = u1(dx) + y}
        mov       ax, Y
        add       ax, dx
        mov       word ptr ds : MachValsY [si], ax
        inc     si
        inc     si
        dec     cx
        cmp     cx, 0
        jle     Finish
        mov     Y, ax
        {u3(dx) = y3 = y2 - y}
        mov      bx, y2
        push     bx
        neg      ax
        sar      ax, 1
        sar      ax, 1
        sar      ax, 1
        add      ax, bx  {y3}
        {u4(dx) = h*y3}
        mov      bx, HMach
        imul     bx  {dx : ax}
        sar      dx, 1
        {u5(dx) = u4(dx) + y2}
        pop      bx  {y2}
        add      dx, bx
        mov      Y2, dx
        {u6(dx) = h*y2(bx)}
        mov      ax, bx
        mov      dx, HMach
        push     dx
        imul     dx   {dx : ax}
        {u7(ax) = u6(dx) + y1}
        mov      ax, Y1
        push     ax
        add      ax, dx
        mov      Y1, ax

        pop       ax {y1}
        pop       dx {h}
        jmp       Next
Finish:
     end;
end;
```