|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Лабораторная работа №1*

*По предмету: «Моделирование»*

**Тема: «Приближенный аналитический метод Пикара в сравнении с численными методами»**

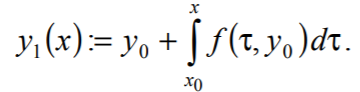
Преподаватель: Градов В.М.

Студент: Мирзоян С.А.

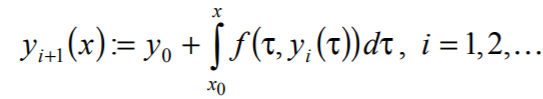
Группа: ИУ7-55Б

**Метод Пикара**

Одним из приближенно-аналитических методов решения задачи Коши (ЗК) является метод Пикара последовательных приближений. Применительно к ЗК для системы обычных дифференциальных уравнений первого порядка (y′ = f (x, y)), он заключается в том, что искомое решение y = y(x) в некоторой правой (или левой) окрестности точки 0 x является пределом ,где последовательность функций, строится по алгоритму, предложенному Пикаром для доказательства теоремы существования и единственности, носящей его имя, согласно которому первое приближение имеет вид:

 (1)

Все дальнейшие приближения строятся по формуле:

 (2)

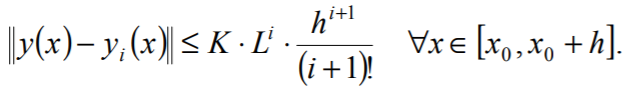
Теорема Пикара доставляет не только условия, при которых процесс (1), (2) сходится, но и оценку погрешности n-го приближения.

В силу трудностей с нахождением первообразных в чистом виде, метод (1)–(2) редко реализуем. Чаще применяется его псевдоаналитический вариант, когда в формулах (1) и (2) интегралы заменяются на квадратурные суммы. Но главную известность метод получил как инструмент для доказательства теоремы Пикара.

**Теорема Пикара:** Если в некоторой односвязной области G, содержащей точку  и f – липшицева по y с константой L:



то найдется такая постоянная h > 0, что на отрезке  , последовательность функций , порождаемая, согласно (1) – (2), равномерно сходится к решению ЗК для СОДУ первого порядка (1.5) и справедлива оценка погрешности



**Метод ломанных**

Это простейший численный метод. В практике вычислений он употребляется очень редко из-за невысокой точности. Но на его примере удобно пояснить способы построения и исследования численных методов.

Рассмотрим задачу Коши и выберем на отрезке  некоторую сетку  значений аргумента так, чтобы выполнялись соотношения  (сетка может быть неравномерной). Разлагая решение и u(x) по формуле Тейлора на интервале сетки  и обозначая и u(xn) = un получим

http://scask.ru/archive/arch.php?path=../htm/stu.sernam/book_dig_m/files.book&file=dig_m_114.files/image1.gif (3)

Стоящие в правой части производные можно найти, дифференцируя уравнение  требуемое число раз: (4)

http://scask.ru/archive/arch.php?path=../htm/stu.sernam/book_dig_m/files.book&file=dig_m_114.files/image2.gif (5)

и т. д. В принципе, если (4) имеет непрерывные производные по совокупности аргументов, то в разложении (3) можно удержать члены вплоть до

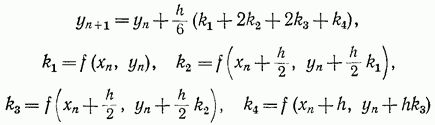
Однако использовать для расчетов формулу 3 с большим числом членов невыгодно. Во-первых, даже при сравнительно простой правой части выражения для производных могут оказаться громоздкими. Во-вторых, если правая часть известна лишь приближенно, то находить ее производные нежелательно. В простейшем случае, подставляя (5) в (3) и ограничиваясь только первым членом разложения, получим схему [ломаных](http://scask.ru/f_book_m_cat.php?id=45)

http://scask.ru/archive/arch.php?path=../htm/stu.sernam/book_dig_m/files.book&file=dig_m_114.files/image3.gif (6)

Поскольку при такой замене можно найти только приближенные значения искомой функции в узлах, то будем обозначать эти значения через y­n в отличие от точных значений un = u(x­n) Для численного расчета по схеме ломаных достаточно задать начальное значение . Затем по формуле (6) последовательно вычисляем величины

**Метод Рунге—Кутта**

можно строить схемы различного порядка точности. Например, схема ломаных (5) есть схема Рунге—Кутта первого порядка точности. Наиболее употребительны схемы четвертого порядка точности, образующие семейство четырехчленных схем. Приведем без вывода ту из них, которая записана в большинстве стандартных программ [ЭВМ](http://scask.ru/f_book_kiber2.php?id=815):



Формулы более высокого порядка точности практически не употребляются. Пятичленные формулы имеют всего лишь четвертый порядок точности; шестичленные имеют шестой порядок, но слишком громоздки. Кроме того, высокий порядок реализуется лишь при наличии у правой части непрерывных [производных](http://scask.ru/q_book_msh.php?id=117) соответствующего порядка.

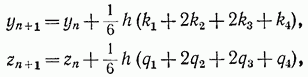
Схемы Рунге—Кутта имеют ряд важных достоинств.

1. Все они (кроме схемы ломаных) имеют хорошую точность.
2. Они являются явными, т. е. значение http://scask.ru/img_page/54.gif вычисляется по ранее найденным значениям за определенное число действий по определенным формулам.
3. Все схемы допускают расчет переменным шагом; значит, нетрудно уменьшить шаг там, где функция быстро меняется, и увеличить его в обратном случае.
4. 4) Для начала расчета достаточно выбрать сетку http://scask.ru/img_page/55.gif и задать значение http://scask.ru/img_page/56.gif далее вычисления идут по одним и тем же формулам. Все эти свойства схем очень ценны при расчетах на [ЭВМ](http://scask.ru/f_book_kiber2.php?id=815).

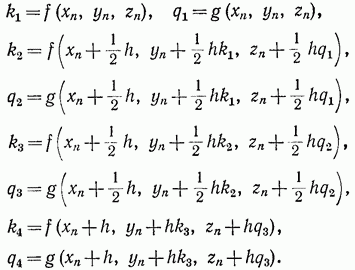
На случай систем уравнений схемы Рунге—Кутта легко переносятся, как во всех других методах, при помощи формальной замены у, f(х, у) на **у**, f(x, **у**). Нетрудно произвести покомпонентную запись этих схем. Например, для системы двух уравнений

http://scask.ru/archive/arch.php?path=../htm/stu.sernam/book_dig_m/files.book&file=dig_m_114.files/image22.gif

обозначая через у, z приближенные значения функций u(x) v(x) запишем аналогичную четырехчленную схему следующим образом:



где



**Листинг**

#include <stdio.h>

#include <math.h>

#define h 0.01

///

/// \brief function

/// \param x

/// \param u

/// \return

///

double **function**(double x, double u)

{

return pow(x,2)+pow(u,2);

}

///

/// \brief Picar3

/// \param x

/// \return

double **Picar3**(double x)

{

double y = pow(x, 3) / 3;

double tmp = 1 + pow(x, 4) / 21;

tmp += 2 \* pow(x, 3) / (693);

tmp += pow(x, 12) / (19845);

return y\*tmp;

}

///

/// \brief Picar4

/// \param x

/// \return

///

double **Picar4**(double x)

{

double y = pow(x, 31)/109876902975;

y += 2 \* pow(x, 23) / 86266215;

y += 2 \* pow(x, 22) / 1361505915;

y += 2 \* pow(x, 19) / 3393495;

y += pow(x, 15) / 59535;

y += 2 \* pow(x, 14) / 916839;

y += pow(x, 13) / 56189133;

y += 2 \* pow(x, 11)/2079;

y += 2 \* pow(x, 10)/31185;

y += 2 \* pow(x, 7)/63;

y += pow(x, 3) / 3;

return y;

}

//Численный метод (Метод ломанных)

///

/// \brief polyline

/// \param x

/// \param y

/// \return

///

double **polyline**(double x, double y)

{

return (y + h \* function(x, y));

}

double **Runge2**(double x, double y)

{

return y + h \* function(x + h / 2, y + h / 2 \* function(x, y));

}

double **Runge4**(double x, double y)

{

double K1 = function(x, y);

double K2 = function(x + h / 2, y + h \* K1 / 2);

double K3 = function(x + h / 2, y + h \* K2 / 2);

double K4 = function(x + h, y + h \* K3);

return y + h / 6 \* (K1 + 2 \* K2 + 2 \* K3 + K4);

}

int **main**()

{

// Границы

double x = 0.0;

double maxX = 1.0;

double poly = 0.0;

double run2 = 0.0;

double run4 = 0.0;

puts("X\t\t|Picar3 \t|Picar4 \t|Polyline\t|Runge\t\t|Runge\t\t");//|function\t|");

puts("\t\t| \t\t| \t\t|yavnyi \t|2d step \t|4th step\t");//\t\t|");

puts("-------------------------------------------------------------------------------------------");

while(x <= maxX)

{

printf("%.2f\t\t|%.7f\t|%.7f\t|%.7f\t|%.7f\t|%.7f\t\n", x, Picar3(x), Picar4(x), polyline(x, poly), run2, run4);

poly = polyline(x, poly);

run2 = Runge2(x, run2);

run4 = Runge4(x, run4);

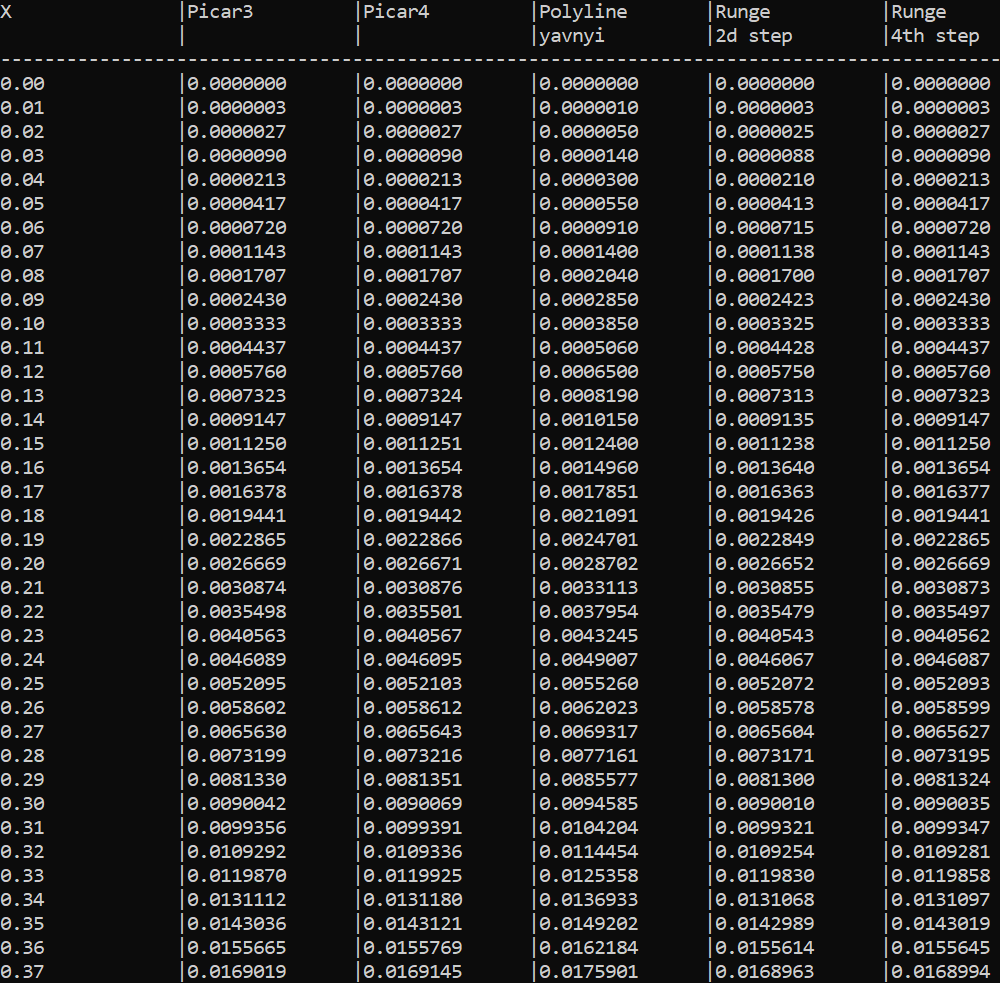
x += h;

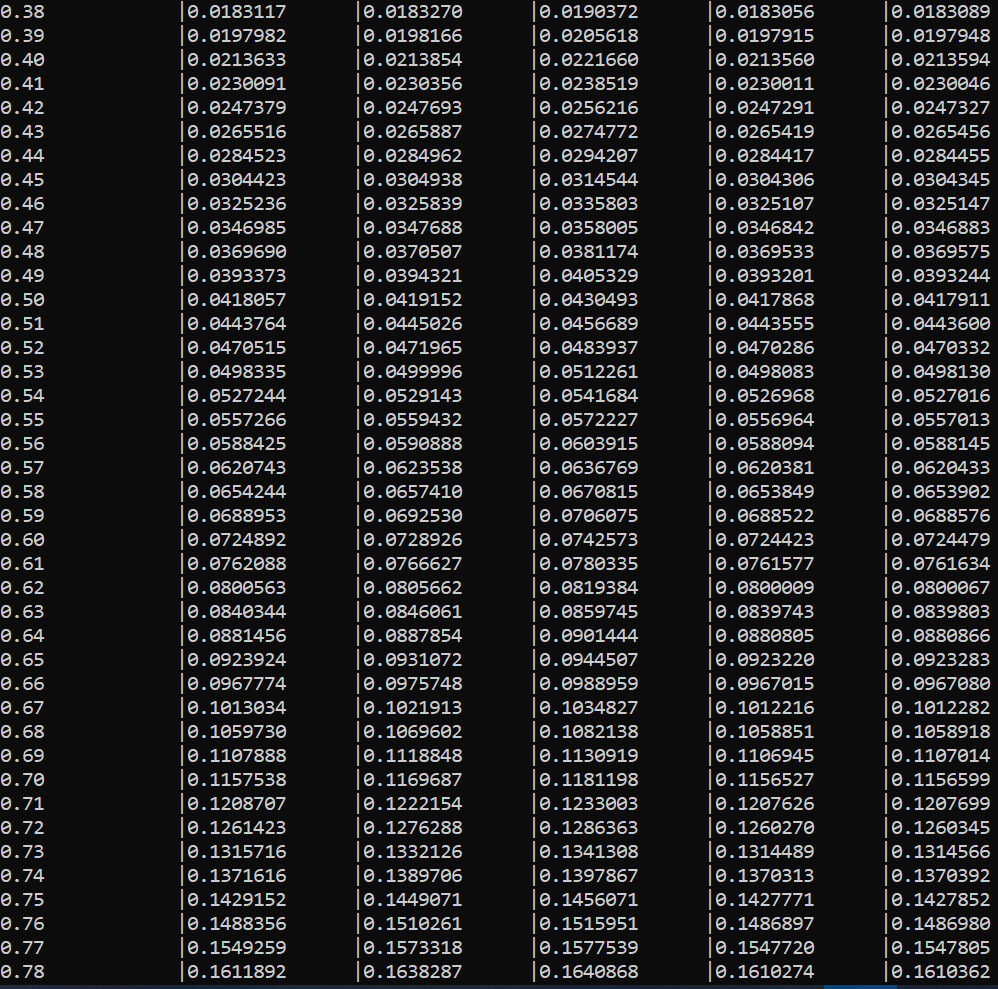
}

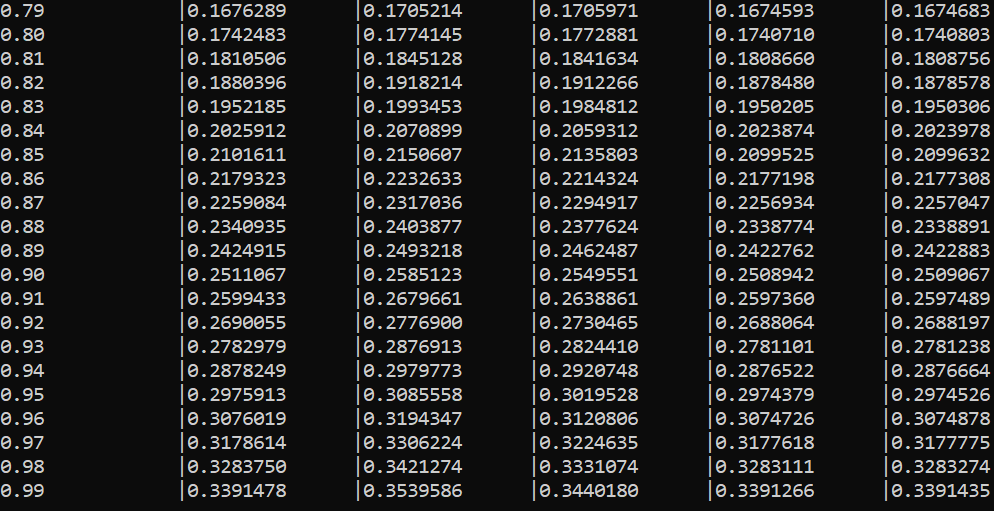
return 0;

}

**Результат работы программы**

****

****



**Заключение**

На практике метод Пикара используется очень редко. Одна из причин – та, что интегралы, которые необходимо вычислять при построении очередных приближений, чаще всего аналитически не находятся, а применение для их вычисления численных методов так усложняет решение, что становится гораздо удобнее непосредственно применять численными методы.