1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ОТЧЕТ**

**по лабораторной работе № 3**

1. по дисциплине «Вычислительная математика»
2. Выполнил
3. студент гр. 23508/4 Е.Г. Проценко
4. Проверил
5. профессор С.М. Устинов
6. Санкт-Петербург
7. 2016

# Формулировка задания (Вариант 29)

Решить систему дифференциальных уравнений:

; ;

,

Следующими способами с одним и тем же шагом печати :

1. По программе **RKF45** c EPS=0.0001;
2. Методом семейства Рунге-Кутты

; ;

;;

;

;

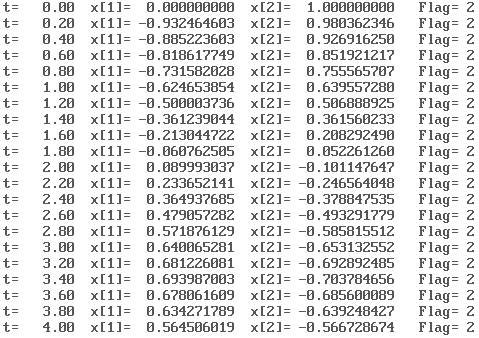
;

C двумя постоянными шагами интегрирования

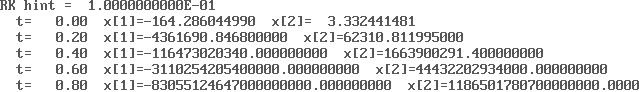
1. Любой другой, позволяющий получить качественно верное решение. Сравнить результаты.

# Результаты работы

1. **Результаты работы подпрограммы RKF45**



1. **Метод семейства Рунге-Кутты для**



Очевидно, что такой шаг интегрирования не подходит, тем более, что после 5 шага, на t = 1 происходит переполнение переменной типа float.

1. **Метод семейства Рунге-Кутты для любого другого шага интегрирования.**

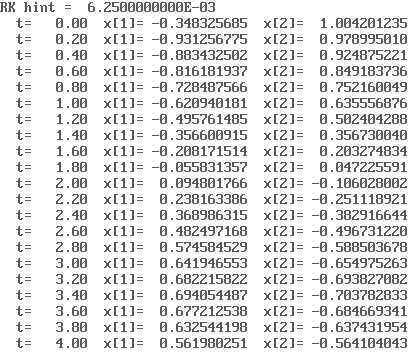
Чтобы разработанная подпрограмма давала качественно верное решение, нужно подобрать соответствующий шаг интегрирования. Условие:

Найдем собственные числа следующей матрицы:

Получаем, что .

Следовательно,

Был выбран шаг интегрирования .

**

Начиная со второго шага удалось достигнуть отклонения не более чем на EPS=0.01. При уменьшении шага интегрирования можно достигнуть большей точности.

# вывод

В данной работе я познакомился с подпрограммой RKF45 и методом Рунге-Кутты в целом.

Ясно, что для метода Рунге-Кутты 6 степени шаг, который был в условии задачи слишком большой, поэтому результат работы очень плохой, абсолютно не соответствующий действительности. После 5-го шага даже случилось переполнение переменной типа float.

Для того, чтобы решение было качественно верным нужен шаг, удовлетворяющий определенному условию. При этом, чем меньше шаг, тем лучше, тем меньше будет погрешность вычислений.

# Приложение

Листинг написанной программы:

uses FMM, CRT, MATH;

label rinse;

const N = 2;

Var

x, xp: floatvector;

t, tout, tfinal, tprint: float;

abserr, relerr: float;

iflag: integer;

work: rvecn;

iwork: ivec5;

hint: float;

i: integer;

hprint: float;

j: float;

counter: float;

{$F+}

procedure f(t: float; var x, xp: floatvector);

begin

xp[1] := -71\*x[1] - 70\*x[2] + exp(1 - power(t, 2));

xp[2] := x[1] + sin(1 - t);

end;

procedure RK(Var x, xp: floatvector; t, hint: float);

Var

i: integer;

k\_1, k\_2, k\_3, k\_4, k\_5, k\_6: floatvector;

temp: floatvector;

begin

{k1}

f(t, x, k\_1);

for i := 1 to N do begin

k\_1[i] := hint \* k\_1[i];

end;

{k2}

for i := 1 to N do begin

temp[i] := x[i] + k\_1[i] / 3;

end;

f(t + hint/3, temp, k\_2);

for i := 1 to N do begin

k\_2[i] := hint \* k\_2[i];

end;

{k3}

for i := 1 to N do begin

temp[i] := x[i] + 0.16 \* k\_1[i] + 0.24 \* k\_2[i];

end;

f(t + 0.4 \* hint, temp, k\_3);

for i := 1 to N do begin

k\_3[i] := hint \* k\_3[i];

end;

{k4}

for i := 1 to N do begin

temp[i] := x[i] + 0.25\*k\_1[i] - 3\*k\_2[i] + 3.75\*k\_3[i];

end;

f(t + hint, temp, k\_4);

for i := 1 to N do begin

k\_4[i] := hint \* k\_4[i];

end;

{k5}

for i := 1 to N do begin

temp[i] := x[i] + (6\*k\_1[i] + 90\*k\_2[i] - 50\*k\_3[i] + 8\*k\_4[i]) / 81;

end;

f(t + 2 \* hint / 3, temp, k\_5);

for i := 1 to N do begin

k\_5[i] := hint \* k\_5[i];

end;

{k6}

for i := 1 to N do begin

temp[i] := x[i] + (6\*k\_1[i] + 90\*k\_2[i] - 50\*k\_3[i] + 8\*k\_4[i]) / 75;

end;

f(t + 4 \* hint / 5, temp, k\_6);

for i := 1 to N do begin

k\_6[i] := hint \* k\_6[i];

end;

{xp}

for i := 1 to N do begin

xp[i] := x[i] + (23\*k\_1[i] + 125\*k\_3[i] - 81\*k\_5[i] + 125\*k\_6[i]) / 192;

end;

end;

{$F-}

begin

clrscr;

t := 0;

tfinal := 4;

tout := t;

tprint := 0.2;

relerr := 0.0001;

abserr := 0;

iflag := 1;

x[1] := 0;

x[2] := 1;

{RFK45}

rinse:

rkf45(@f,N,x,t,tout,relerr,abserr,iflag,work,iwork);

writeln(' t= ',t:6:2, ' x[1]=',x[1]:13:9,' x[2]=',

x[2]:13:9,' Flag=',iflag:2);

case iflag of

1, 8 : exit;

2 : begin

tout := t + tprint;

if t < tfinal then goto rinse

end;

4 : goto rinse;

5 : begin

abserr := 1E-9;

goto rinse

end;

6 : begin

relerr := 10 \* relerr;

iflag := 2;

goto rinse

end;

7 : begin

iflag := 2;

goto rinse

end;

end;

readln;

{RK}

t := 0;

hint := 0.1;

x[1] := 0;

x[2] := 1;

hprint := tprint;

counter := hprint / hint;

j := counter;

writeln('RK hint = ', hint);

while t <= tfinal/4 do begin

RK(x, xp, t, hint);

x[1] := xp[1];

x[2] := xp[2];

if j = counter then begin

writeln(' t= ', t:6:2, ' x[1]=',x[1]:13:9,' x[2]=', x[2]:13:9);

j := 1;

end

else j := j + 1;

t := t + hint;

end;

readln;

t := 0;

hint := 0.00625;

x[1] := 0;

x[2] := 1;

hprint := tprint;

counter := hprint / hint;

j := counter;

writeln('RK hint = ', hint);

while t <= tfinal do begin

RK(x, xp, t, hint);

x[1] := xp[1];

x[2] := xp[2];

if j = counter then begin

writeln(' t= ',t:6:2, ' x[1]=',x[1]:13:9,' x[2]=', x[2]:13:9);

j := 1;

end

else j := j + 1;

t := t + hint;

end;

readln;

end.