Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
—
Институт компьютерных наук и технологий
Кафедра «Информационная безопасность компьютерных систем»

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине «Вычислительная математика»

Выполнил
студент гр. 23508/4 Е.Г. Проценко

Проверил
профессор С.М. Устинов

Санкт-Петербург
2016

1 ФОРМУЛИРОВКА ЗАДАНИЯ (ВАРИАНТ 29)

Решить систему дифференциальных уравнений:

$\frac{dx_1}{dt} = -71x_1 - 70x_2 + e^{1-t^2}$; $\frac{dx_2}{dt} = x_1 + \sin(1-t)$;

$x_1(0) = 0,$ $x_2(0) = 1;$ $t \in [0,\ 4]$

Следующими способами с одним и тем же шагом печати $h_{print} = 0.2$:

I) По программе **RKF45** с EPS=0.0001;

II) Методом семейства Рунге-Кутты

$$z_{n+1} = z_n + \frac{23k_1 + 125k_3 - 81k_5 + 125k_6}{192}; \qquad k_1 = hf(t_n, z_n);$$

$k_2 = hf(t_n + \frac{h}{3}, z_n + k_1/3)$; $k_3 = hf(t_n + 0.4h, z_n + 0.16k_1 + 0.24k_2)$;
$k_4 = hf(t_n + h, z_n + 0.25k_1 - 3k_2 + 3.75k_3)$;
$k_5 = hf(t_n + h, z_n + (6k_1 + 90k_2 - 50k_3 + 8k_4)/81)$;
$k_6 = hf(t_n + \frac{4h}{5}, z_n + (6k_1 + 36k_2 + 10k_3 + 8k_4)/75)$;

С двумя постоянными шагами интегрирования

a) $h_{int} = 0.1$
b) Любой другой, позволяющий получить качественно верное решение. Сравнить результаты.

2 РЕЗУЛЬТАТЫ РАБОТЫ

1) Результаты работы подпрограммы RKF45

```
t=   0.00  x[1]=  0.000000000  x[2]=  1.000000000   Flag= 2
t=   0.20  x[1]= -0.932464603  x[2]=  0.980362346   Flag= 2
t=   0.40  x[1]= -0.885223603  x[2]=  0.926916250   Flag= 2
t=   0.60  x[1]= -0.818617749  x[2]=  0.851921217   Flag= 2
t=   0.80  x[1]= -0.731582028  x[2]=  0.755565707   Flag= 2
t=   1.00  x[1]= -0.624653854  x[2]=  0.639557280   Flag= 2
t=   1.20  x[1]= -0.500003736  x[2]=  0.506888925   Flag= 2
t=   1.40  x[1]= -0.361239044  x[2]=  0.361560233   Flag= 2
t=   1.60  x[1]= -0.213044722  x[2]=  0.208292490   Flag= 2
t=   1.80  x[1]= -0.060762505  x[2]=  0.052261260   Flag= 2
t=   2.00  x[1]=  0.089993037  x[2]= -0.101147647   Flag= 2
t=   2.20  x[1]=  0.233652141  x[2]= -0.246564048   Flag= 2
t=   2.40  x[1]=  0.364937685  x[2]= -0.378847535   Flag= 2
t=   2.60  x[1]=  0.479057282  x[2]= -0.493291779   Flag= 2
t=   2.80  x[1]=  0.571876129  x[2]= -0.585815512   Flag= 2
t=   3.00  x[1]=  0.640065281  x[2]= -0.653132552   Flag= 2
t=   3.20  x[1]=  0.681226081  x[2]= -0.692892485   Flag= 2
t=   3.40  x[1]=  0.693987003  x[2]= -0.703784656   Flag= 2
t=   3.60  x[1]=  0.678061609  x[2]= -0.685600089   Flag= 2
t=   3.80  x[1]=  0.634271789  x[2]= -0.639248427   Flag= 2
t=   4.00  x[1]=  0.564506019  x[2]= -0.566728674   Flag= 2
```

2) Метод семейства Рунге-Кутты для $h_{int} = 0.1$

```
RK hint =  1.0000000000E-01
  t=   0.00  x[1]=-164.286044990  x[2]=  3.332441481
  t=   0.20  x[1]=-4361690.846800000  x[2]=62310.811995000
  t=   0.40  x[1]=-116473020340.000000000  x[2]=1663900291.400000000
  t=   0.60  x[1]=-3110254205400000.000000000  x[2]=44432202934000.000000000
  t=   0.80  x[1]=-83055124647000000000.000000000  x[2]=1186501780700000000.0000
```

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

3) Метод семейства Рунге-Кутты для любого другого шага интегрирования.

Чтобы разработанная подпрограмма давала качественно верное решение, нужно подобрать соответствующий шаг интегрирования. Условие: $\boldsymbol{h}|\boldsymbol{\lambda_{max}}| < 2$.

Найдем собственные числа следующей матрицы:

$$\begin{vmatrix} -71-\lambda & -70 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 + 71\lambda + 70 = (\lambda+1)(\lambda+70)$$

$$|\lambda_{max}| = 70$$

Получаем, что $h < 1/|\lambda_{max}|$.

Следовательно, $h < 0{,}0142857142857143$

Был выбран шаг интегрирования $h_{int} = 0.00625$.

```
RK hint =  6.2500000000E-03
  t=   0.00  x[1]= -0.348325685  x[2]=  1.004201235
  t=   0.20  x[1]= -0.931256775  x[2]=  0.978995010
  t=   0.40  x[1]= -0.883432502  x[2]=  0.924875221
  t=   0.60  x[1]= -0.816181937  x[2]=  0.849183736
  t=   0.80  x[1]= -0.728487566  x[2]=  0.752160049
  t=   1.00  x[1]= -0.620940181  x[2]=  0.635556876
  t=   1.20  x[1]= -0.495761485  x[2]=  0.502404288
  t=   1.40  x[1]= -0.356600915  x[2]=  0.356730040
  t=   1.60  x[1]= -0.208171514  x[2]=  0.203274834
  t=   1.80  x[1]= -0.055831357  x[2]=  0.047225591
  t=   2.00  x[1]=  0.094801766  x[2]= -0.106028002
  t=   2.20  x[1]=  0.238163386  x[2]= -0.251118921
  t=   2.40  x[1]=  0.368986315  x[2]= -0.382916644
  t=   2.60  x[1]=  0.482497168  x[2]= -0.496731220
  t=   2.80  x[1]=  0.574584529  x[2]= -0.588503678
  t=   3.00  x[1]=  0.641946553  x[2]= -0.654975263
  t=   3.20  x[1]=  0.682215822  x[2]= -0.693827082
  t=   3.40  x[1]=  0.694054487  x[2]= -0.703782833
  t=   3.60  x[1]=  0.677212538  x[2]= -0.684669341
  t=   3.80  x[1]=  0.632544198  x[2]= -0.637431954
  t=   4.00  x[1]=  0.561980251  x[2]= -0.564104043
```

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

3 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.

4 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.
```