Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

**Отчет по лабораторной работе**

Вариант 29. Применение RKF45

Выполнил студент гр. в3530904/00022 <подпись> А.И. Чигирь

Руководитель

доцент, к.ф.- м.н. <подпись> С.П. Воскобойников

Санкт-Петербург

2022г.

**Оглавление**

Условия задачи 3

Исходный код 4

Вывод программы 9

Вывод 13

**Условия задачи**

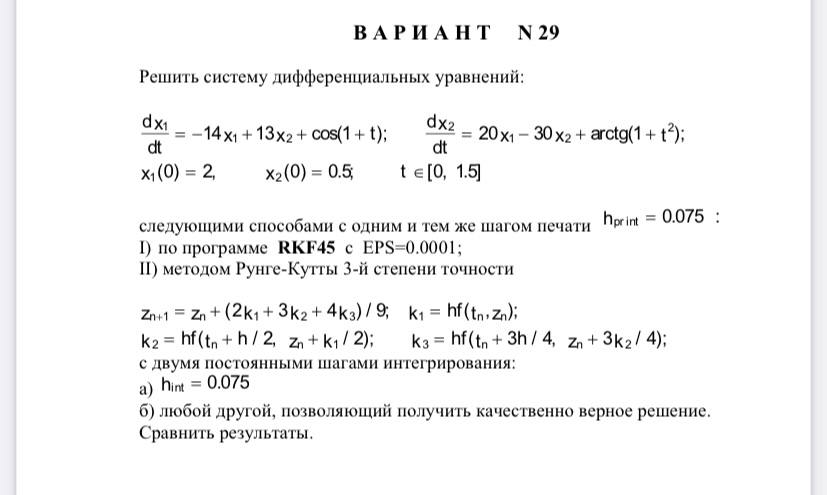
****

Рисунок 1. Условие задачи

**Исходный код**

main.f90:

program lab3\_RKF

! модули

use Environment

use Group\_Process

! Важные переменные для RKF45

real(R\_) :: LOW, HIGH, RELERR, ABSERR, STEP

real(R\_), allocatable :: WORK(:), x(:)

integer(I\_) :: NEQN, IFLAG

integer(I\_), allocatable :: IWORK(:)

! Важные переменные для RKF3 (Метод Рунге-Кутты 3ьей степени точности)

real(R\_) :: LOW\_RKF3\_1, LOW\_RKF3\_2, CUSTOM\_STEP

real(R\_), allocatable :: x\_rkf3\_1(:), x\_rkf3\_2(:)

! Для циклов, ввода-вывода и прочего

integer(I\_) :: i = 0, Out = 1

character(:), allocatable :: output\_file, format

! подготовка к решению по RKF45

output\_file = "results .txt" ! создания файла с результатми

! и его чистка

open (file=output\_file, newunit=Out, position="rewind")

close(Out)

NEQN = 2 ! число уравнений

ABSERR = 1.E-4 ! EPS = 0.0001

RELERR = 1.E-4 ! EPS = 0.0001

allocate(x(NEQN)) ! выделение память под узлы для RKF45

allocate(x\_rkf3\_1(NEQN)) ! выделение память под узлы для RKF3

allocate(x\_rkf3\_2(NEQN)) ! выделение память под узлы для RKF3

allocate(WORK(3+6\*NEQN)) ! (требования по документации RKF45)

allocate(IWORK(5)) ! (требования по документации RKF45)

x(1) = 2 ! начальное условие x1

x\_rkf3\_1(1) = x(1)

x\_rkf3\_2(1) = x(1)

x(2) = 0.5 ! начальное условие x2

x\_rkf3\_1(2) = x(2)

x\_rkf3\_2(2) = x(2)

! границы

LOW = 0.0 ! нижняя граница для RKF45

LOW\_RKF3\_1 = LOW ! нижняя граница для RKF3

LOW\_RKF3\_1 = LOW ! нижняя граница для RKF3

HIGH = 0.0 ! верхняя граница

STEP = 0.075 ! шаг увелечения границы

CUSTOM\_STEP = 0.001 ! шаг увелечения границы для RKF3

IFLAG = +1 ! указатель настройки программы

! режим: многошаговый интегратор

do while (HIGH < 1.501)

! вывод результатов пошагово

open (file=output\_file, newunit=Out, position="append")

format = "(2(a, 1f5.3), a)"

write(Out, format) "Отрезок (", LOW, ";", HIGH, ")"

call RKF45(diff\_system, NEQN, x, LOW, HIGH, RELERR, ABSERR, IFLAG, WORK, IWORK)

call RKF3(diff\_system, x\_rkf3\_1, LOW\_RKF3\_1, HIGH, STEP)

call RKF3(diff\_system, x\_rkf3\_2, LOW\_RKF3\_2, HIGH, CUSTOM\_STEP)

format = "(a, 1f5.3, a, 1f14.5)"

write(Out, format) "RKF-45, step: ", STEP, ", x1: ", x(1)

write(Out, format) "RKF-3 , step: ", CUSTOM\_STEP, ", x1: ", x\_rkf3\_2(1)

write(Out, format) "RKF-3 , step: ", STEP, ", x1: ", x\_rkf3\_1(1)

write(Out, format) "RKF-45, step: ", STEP, ", x2: ", x(2)

write(Out, format) "RKF-3 , step: ", CUSTOM\_STEP, ", x2: ", x\_rkf3\_2(2)

write(Out, format) "RKF-3 , step: ", STEP, ", x2: ", x\_rkf3\_1(2)

if (IFLAG /= 2) then

write(Out, \*) "Flag :", IFLAG

end if

write(Out, \*) " "

close(Out)

! изменение переменных для следующего шага

LOW = HIGH

LOW\_RKF3\_1 = HIGH

LOW\_RKF3\_2 = HIGH

HIGH = HIGH + STEP

end do

contains

end program lab3\_RKF

group\_process.f90:

module Group\_Process

use Environment

contains

! dx1 по dt

real(R\_) function diff\_eq1(x1, x2, t)

real(R\_) :: x1, x2, t

diff\_eq1 = -14 \* x1 + 13 \* x2 + cos(1 + t)

end function diff\_eq1

! dx2 по dt

real(R\_) function diff\_eq2(x1, x2, t)

real(R\_) :: x1, t

diff\_eq2 = 20 \* x1 - 30 \* x2 + atan(1 + t \* t)

end function diff\_eq2

! дифф. система уравнений

subroutine diff\_system(t, x, dx)

real(R\_), intent(inout) :: t, x(2)

real(R\_), intent(inout) :: dx(2)

dx(1) = diff\_eq1(x(1), x(2), t)

dx(2) = diff\_eq2(x(1), x(2), t)

end subroutine diff\_system

! метод Рунге-КУтты 3-й степени точности

subroutine RKF3(func, x, LOW, HIGH, STEP)

! входные переменные

external :: func

real(R\_), intent(inout) :: x(2)

real(R\_), intent(inout) :: LOW

real(R\_), intent(in) :: HIGH, STEP

! локальные

real(R\_) :: k1(2), k2(2), k3(2)

do while (LOW < HIGH)

k1(1) = step \* diff\_eq1(x(1), x(2), LOW)

k1(2) = step \* diff\_eq2(x(1), x(2), LOW)

k2(1) = step \* diff\_eq1(x(1) + k1(1) / 2, x(2) + k1(2) / 2, LOW + STEP / 2)

k2(2) = step \* diff\_eq2(x(1) + k1(1) / 2, x(2) + k1(2) / 2, LOW + STEP / 2)

k3(1) = step \* diff\_eq1(x(1) + 3 \* k2(1) / 4, x(2) + 3 \* k2(2) / 4, LOW + 3 \* STEP / 4)

k3(2) = step \* diff\_eq2(x(1) + 3 \* k2(1) / 4, x(2) + 3 \* k2(2) / 4, LOW + 3 \* STEP / 4)

x(1) = x(1) + (2 \* k1(1) + 3 \* k2(1) + 4 \* k3(1)) / 9

x(2) = x(2) + (2 \* k1(2) + 3 \* k2(2) + 4 \* k3(2)) / 9

LOW = LOW + STEP

end do

end subroutine rkf3

end module Group\_Process

**Вывод программы**

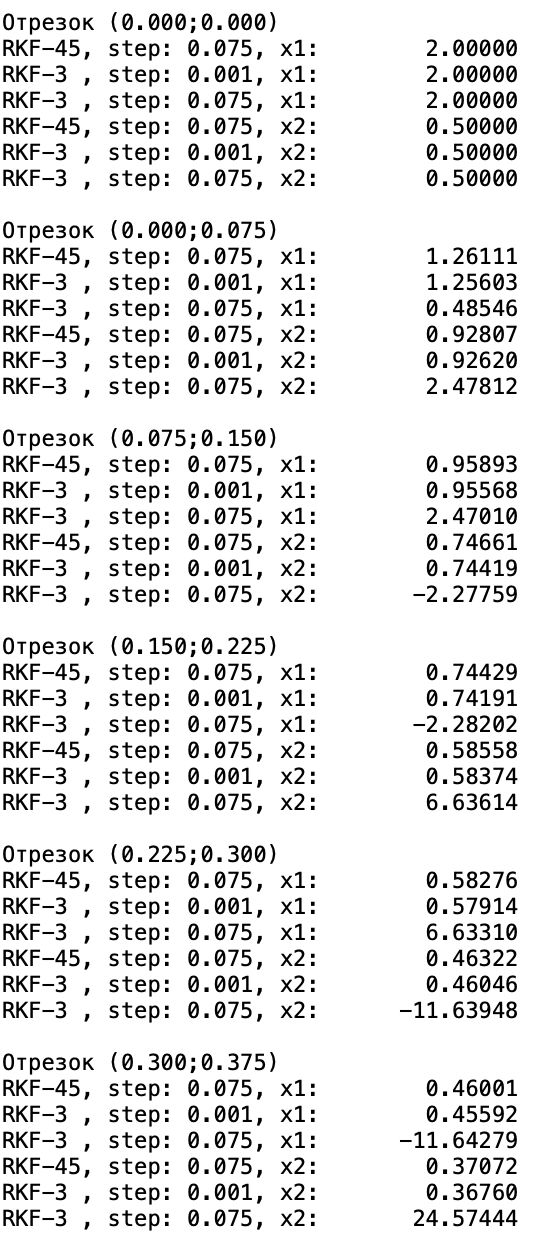


Рисунок 2. Скриншот файла вывода 1

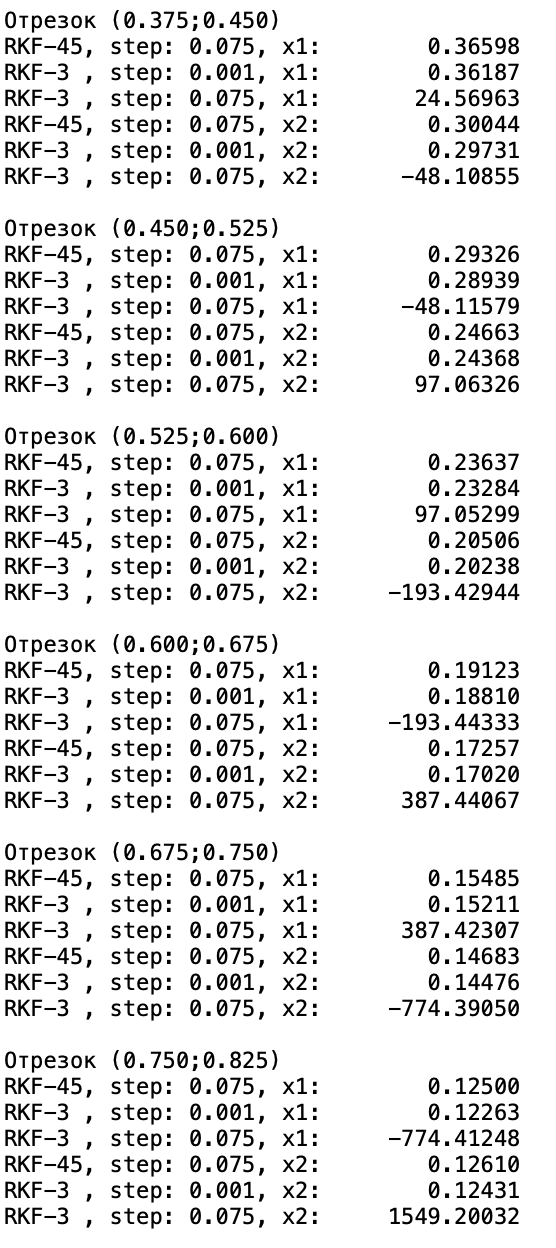


Рисунок 3. Скриншот файла вывода 2

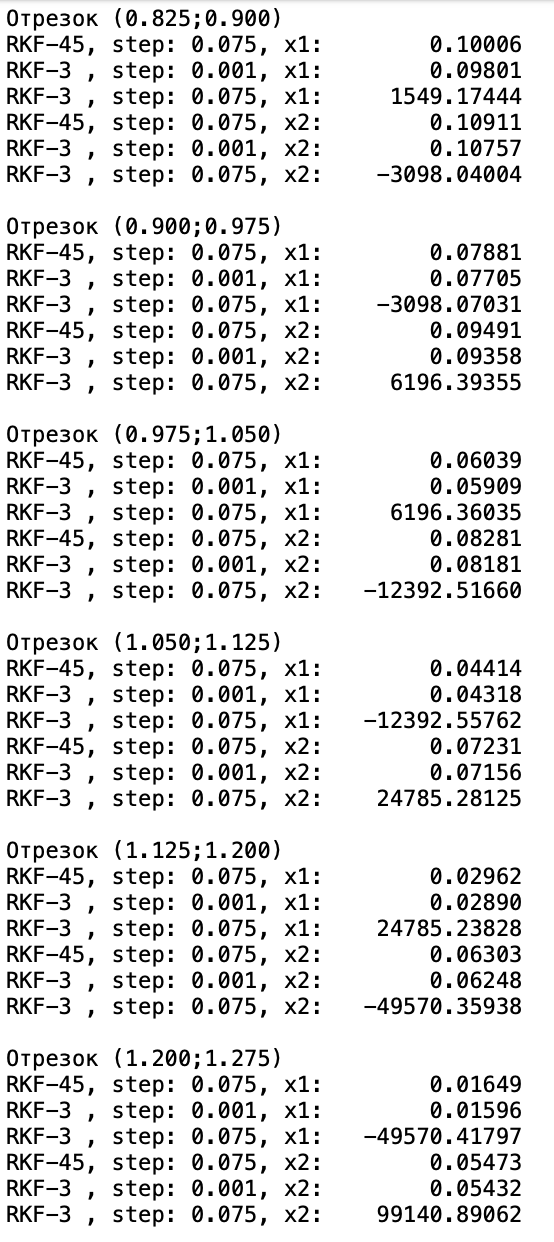
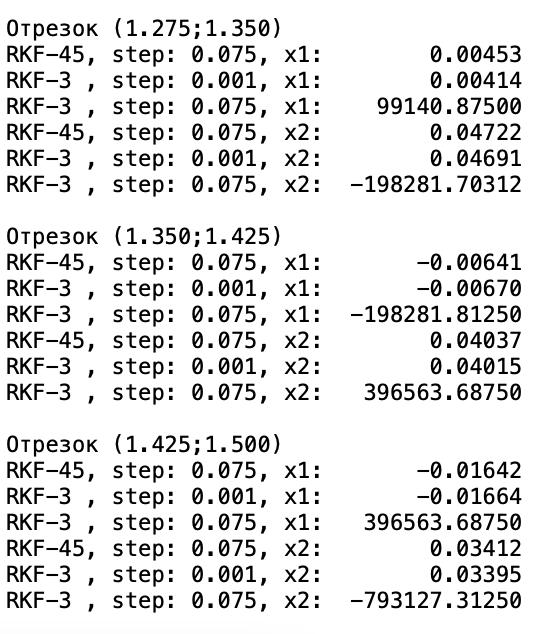


Рисунок 4. Скриншот файла вывода 3



**Вывод**

При воспроизведении метода Рунге-Кутты 3ей степени точности стало понятно, что постоянный шаг интегрирования, равный 0.075, является большим и вычисление происходит неправильно. Поэтому необходимо установить границу устойчивости для данного метода:

**Составление неравенства:**

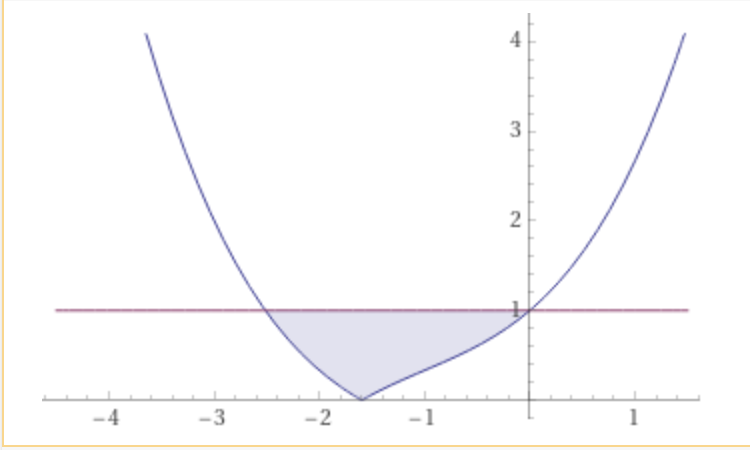
Неравенство ограничения на постоянный шаг интегрирования для метода из семейства Рунге-Кутты 3 степени:

,

- это наибольшее по модулю отрицательное число матрицы А.

**Решение неравенства для нахождения константы, за которой решение является неустойчивым**

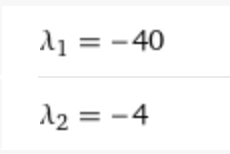
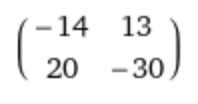
Заменим = x и решим данное неравенство и построим интервал. Его (интервала) левая граница и будет данной константой, которая является верхней границей для произведения , выход за которую означает неустойчивость системы.



Отсюда вывод, что:

**Составление окончательного вида неравенство, нахождение**

Составим матрицу A и найдем собственные числа:



Исходя из этого за принимается соб. число, равное 40.

Финальный вид неравенства:

**Нахождение максимального постоянного шага интегрирования, при котором метод считается устойчивым**

**Генеральный вывод:**

Программа RKF45 оказалась точнее, нежели программа, где реализовывался метод Рунге-Кутты 3ей степени точности. Это объясняется тем, что в RKF45 используется метод Рунге-Кутты 4-5 степени точности.