**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ НА ПРИМЕРЕ КИНЕТИКИ ХИМИЧЕСКИХ РЕАКЦИЙ**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 А.А. Циттель

(Подпись)

\_25\_\_\_ \_ мая\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** научиться составлять программы для численных методов решения систем дифференциальных уравнений на примере кинетики химических реакций.

**Теоретическая часть**

**Закон действующих масс:**

Скорость химической реакции прямо пропорциональна произведению концентраций реагирующих веществ, возведенных в степени, равные стехиометрическим коэффициентам.

Пусть дана схема химической реакции:



Скорость данной реакции можно выразить:



где *k* – константа скорости химической реакции; *CA1, CA2, CA3, CB* – концентрации веществ (моль/л), участвующих в химической реакции, *n1, n2, n3* – стехиометрические коэффициенты в уравнении реакции.

Изменение концентрации каждого компонента во времени:



Пусть дана схема химической реакции:



Константы скоростей реакций:

k1 = 0.5

k2 = 0.2

***СА****,* ***СВ***– концентрации компонентов А и В

Необходимо определить изменение концентраций каждого компонента по времени методами Эйлера и Рунге-Кутты на отрезке *[0; 1]* с шагом *h = 0.1*.

Начальные условия: *CA(0) = 1 (моль / л); CB(0) = 0 (моль / л)*.

**Метод Эйлера:**

**Program** lab\_14\_example;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 2;

k: arr = (0.5, 0.2);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] + k[1] \* c[1];

result[1] := k[0] \* c[0] - k[1] \* c[1]

**end**;

**function** eyler\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

f, c: arr;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

f := right\_parts(t, c);

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h \* f[j];

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 1;

c[1] := 0;

print\_results(eyler\_method(0, 1, 0.1, c))

**end**.

**Ответ:**

0.0000 1.0000 0.0000

0.1000 0.9500 0.0500

0.2000 0.9035 0.0965

0.3000 0.8603 0.1397

0.4000 0.8200 0.1800

0.5000 0.7826 0.2174

0.6000 0.7479 0.2521

0.7000 0.7155 0.2845

0.8000 0.6854 0.3146

0.9000 0.6574 0.3426

1.0000 0.6314 0.3686

**Метод Рунге-Кутты:**

**Program** lab\_14\_example;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 2;

k: arr = (0.5, 0.2);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] + k[1] \* c[1];

result[1] := k[0] \* c[0] - k[1] \* c[1]

**end**;

**function** runge\_kutt\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum\_map(a: real; array1, array2: arr): arr;

**var**

i: integer;

**begin**

SetLength(result, Length(array1));

**for** i := 0 **to** High(result) **do**

result[i] := array1[i] + array2[i] \* a

**end**;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

k1 := right\_parts(t, c);

k2 := right\_parts(t + h / 2, sum\_map(h / 2, c, k1));

k3 := right\_parts(t + h / 2, sum\_map(h / 2, c, k2));

k4 := right\_parts(t + h, sum\_map(h, c, k3));

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h / 6 \* (k1[j] + 2 \* k2[j] + 2 \* k3[j] + k4[j]);

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 1;

c[1] := 0;

print\_results(runge\_kutt\_method(0, 1, 0.1, c))

**end**.

**Ответ:**

0.0000 1.0000 0.0000

0.1000 0.9517 0.0483

0.2000 0.9067 0.0933

0.3000 0.8647 0.1353

0.4000 0.8256 0.1744

0.5000 0.7891 0.2109

0.6000 0.7550 0.2450

0.7000 0.7233 0.2767

0.8000 0.6937 0.3063

0.9000 0.6661 0.3339

1.0000 0.6404 0.3596

**Практическая часть**

**Задание 1**

**Задание:** Дана схема химических превращений:



CA0 = 0.8 (моль / л);

СВ0 = 0.2 (моль / л);

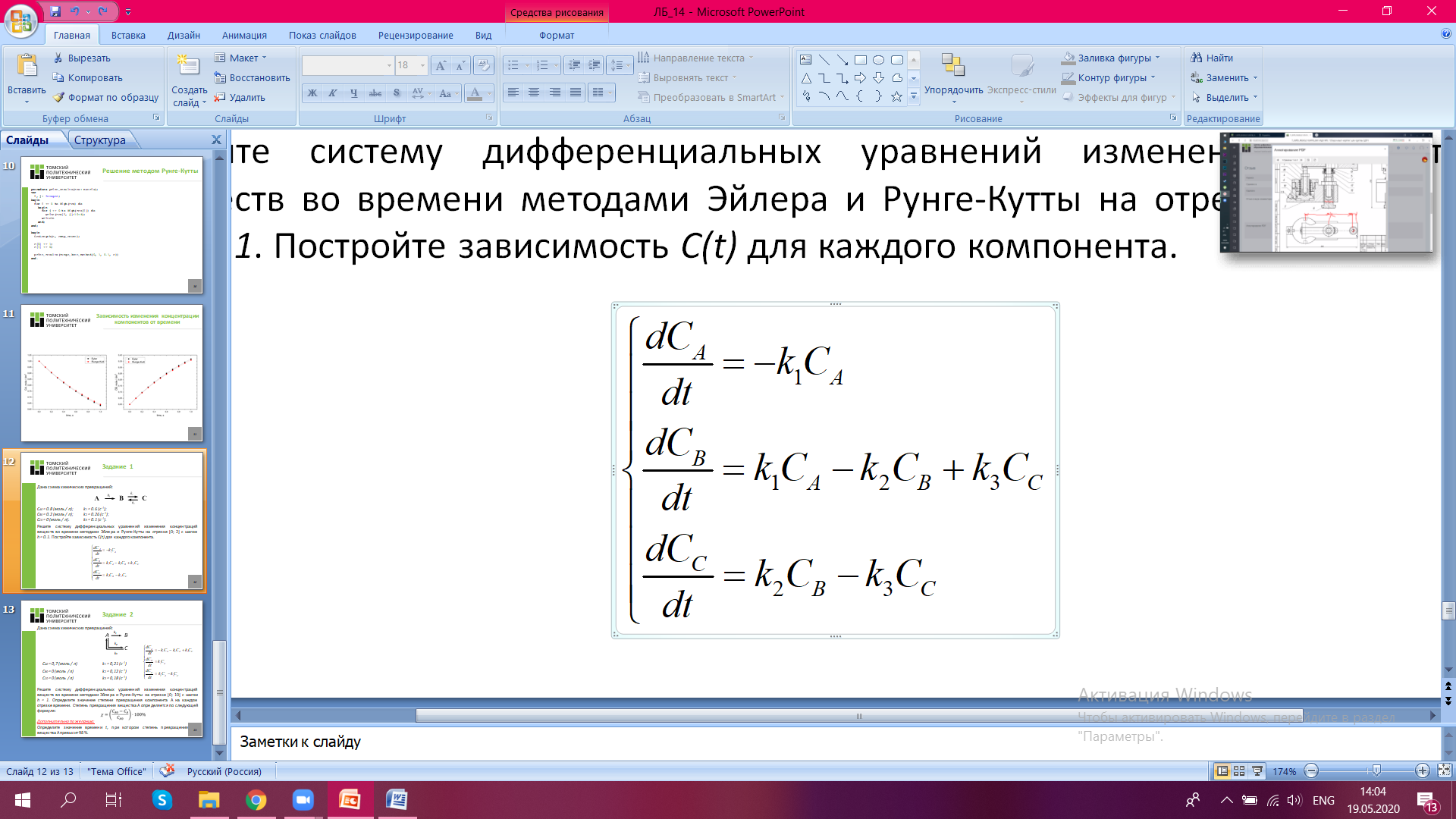
СС0 = 0 (моль / л).

k1 = 0.6 (c-1);

k2 = 0.26 (c-1);

k3 = 0.1 (c-1).

Решите систему дифференциальных уравнений изменения концентраций веществ во времени методами Эйлера и Рунге-Кутты на отрезке [0; 2] с шагом h = 0.1. Постройте зависимость С(t) для каждого компонента.



**Программная реализация:**

**1 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.6, 0.26, 0.1);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0];

result[1] := k[0] \* c[0] - k[1] \* c[1] + k[2] \* c[2];

result[2] := k[1] \* c[1] - k[2] \* c[2]

**end**;

**function** eyler\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

f, c: arr;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

f := right\_parts(t, c);

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h \* f[j];

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.8;

c[1] := 0.2;

c[2] := 0.1;

print\_results(eyler\_method(0, 2, 0.1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t)

0.0000 0.8000 0.2000 0.1000

0.1000 0.7520 0.2438 0.1042

0.2000 0.7069 0.2836 0.1095

0.3000 0.6645 0.3198 0.1158

0.4000 0.6246 0.3525 0.1229

0.5000 0.5871 0.3820 0.1309

0.6000 0.5519 0.4086 0.1395

0.7000 0.5188 0.4325 0.1487

0.8000 0.4877 0.4539 0.1585

0.9000 0.4584 0.4729 0.1687

1.0000 0.4309 0.4898 0.1793

1.1000 0.4050 0.5047 0.1902

1.2000 0.3807 0.5178 0.2015

1.3000 0.3579 0.5292 0.2129

1.4000 0.3364 0.5390 0.2245

1.5000 0.3162 0.5475 0.2363

1.6000 0.2973 0.5546 0.2482

1.7000 0.2794 0.5605 0.2601

1.8000 0.2627 0.5653 0.2721

1.9000 0.2469 0.5690 0.2841

2.0000 0.2321 0.5719 0.2960

**2 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.6, 0.26, 0.1);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0];

result[1] := k[0] \* c[0] - k[1] \* c[1] + k[2] \* c[2];

result[2] := k[1] \* c[1] - k[2] \* c[2]

**end**;

**function** runge\_kutt\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum\_map(a: real; array1, array2: arr): arr;

**var**

i: integer;

**begin**

SetLength(result, Length(array1));

**for** i := 0 **to** High(result) **do**

result[i] := array1[i] + array2[i] \* a

**end**;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

k1 := right\_parts(t, c);

k2 := right\_parts(t + h / 2, sum\_map(h / 2, c, k1));

k3 := right\_parts(t + h / 2, sum\_map(h / 2, c, k2));

k4 := right\_parts(t + h, sum\_map(h, c, k3));

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h / 6 \* (k1[j] + 2 \* k2[j] + 2 \* k3[j] + k4[j]);

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.8;

c[1] := 0.2;

c[2] := 0.1;

print\_results(runge\_kutt\_method(0, 2, 0.1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t)

0.0000 0.8000 0.2000 0.1000

0.1000 0.7534 0.2419 0.1047

0.2000 0.7095 0.2800 0.1104

0.3000 0.6682 0.3147 0.1170

0.4000 0.6293 0.3463 0.1244

0.5000 0.5927 0.3748 0.1325

0.6000 0.5581 0.4006 0.1413

0.7000 0.5256 0.4238 0.1505

0.8000 0.4950 0.4447 0.1603

0.9000 0.4662 0.4634 0.1704

1.0000 0.4390 0.4800 0.1809

1.1000 0.4135 0.4948 0.1917

1.2000 0.3894 0.5078 0.2028

1.3000 0.3667 0.5192 0.2141

1.4000 0.3454 0.5291 0.2255

1.5000 0.3253 0.5377 0.2371

1.6000 0.3063 0.5450 0.2487

1.7000 0.2885 0.5511 0.2604

1.8000 0.2717 0.5562 0.2722

1.9000 0.2559 0.5603 0.2839

2.0000 0.2410 0.5634 0.2956

**Задание 2**

**Задание:** Дана схема химических превращений:



|  |  |
| --- | --- |
| СА0 = 0,7 (моль / л) | k1 = 0,21 (c-1) |
| СB0 = 0 (моль / л) | k2 = 0,12 (c-1) |
| CC0 = 0 (моль / л) | k3 = 0,18 (c-1) |

****

Решите систему дифференциальных уравнений изменения концентраций веществ во времени методами Эйлера и Рунге-Кутты на отрезке [0; 10] с шагом h = 1. Определите значение степени превращения компонента А на каждом отрезке времени. Степень превращения вещества А определяется по следующей формуле:

**Программная реализация:**

**1 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.21, 0.12, 0.18);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] - k[1] \* c[0] + k[2] \* c[2];

result[1] := k[0] \* c[0];

result[2] := k[1] \* c[0] - k[2] \* c[2]

**end**;

**function** eyler\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

f, c: arr;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 2);

SetLength(c, Length(c0) + 1);

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

c[3] := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

f := right\_parts(t, c);

**for** j := 0 **to** High(c)-1 **do**

c[j] := c[j] + h \* f[j];

c[3] := (c0[0] - c[0]) / c0[0] \* 100;

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

write('X':6);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

print\_results(eyler\_method(0, 10, 1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t) X

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.4690 0.1470 0.0840 33.0000

2.0000 0.3294 0.2455 0.1252 52.9500

3.0000 0.2432 0.3147 0.1422 65.2581

4.0000 0.1885 0.3657 0.1457 73.0676

5.0000 0.1525 0.4053 0.1421 78.2074

6.0000 0.1278 0.4373 0.1349 81.7440

7.0000 0.1099 0.4642 0.1259 84.3007

8.0000 0.0963 0.4873 0.1164 86.2436

9.0000 0.0855 0.5075 0.1070 87.7890

10.0000 0.0765 0.5254 0.0980 89.0662

**2 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.21, 0.12, 0.18);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] - k[1] \* c[0] + k[2] \* c[2];

result[1] := k[0] \* c[0];

result[2] := k[1] \* c[0] - k[2] \* c[2]

**end**;

**function** runge\_kutt\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum\_map(a: real; array1, array2: arr): arr;

**var**

i: integer;

**begin**

SetLength(result, Length(array1));

**for** i := 0 **to** High(result) **do**

result[i] := array1[i] + array2[i] \* a

**end**;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

k1 := right\_parts(t, c);

k2 := right\_parts(t + h / 2, sum\_map(h / 2, c, k1));

k3 := right\_parts(t + h / 2, sum\_map(h / 2, c, k2));

k4 := right\_parts(t + h, sum\_map(h, c, k3));

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h / 6 \* (k1[j] + 2 \* k2[j] + 2 \* k3[j] + k4[j]);

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

write('X':6);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

write(((0.7 - res[i, 1]) / 0.7 \* 100):10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

print\_results(runge\_kutt\_method(0, 10, 1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t) X

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.5090 0.1256 0.0654 27.2820

2.0000 0.3793 0.2180 0.1027 45.8130

3.0000 0.2902 0.2877 0.1221 58.5416

4.0000 0.2281 0.3417 0.1301 67.4096

5.0000 0.1841 0.3848 0.1311 73.6973

6.0000 0.1523 0.4199 0.1278 78.2498

7.0000 0.1286 0.4493 0.1221 81.6258

8.0000 0.1106 0.4743 0.1150 84.1956

9.0000 0.0966 0.4960 0.1074 86.2054

10.0000 0.0853 0.5151 0.0997 87.8195

**Дополнительно:**

**Задание:** Определите значение времени *t*, при котором степень превращения вещества А превысит 98 %.

**Программная реализация:**

**1 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.21, 0.12, 0.18);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] - k[1] \* c[0] + k[2] \* c[2];

result[1] := k[0] \* c[0];

result[2] := k[1] \* c[0] - k[2] \* c[2]

**end**;

**function** eyler\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

f, c: arr;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 2);

SetLength(c, Length(c0) + 1);

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

c[3] := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

f := right\_parts(t, c);

**for** j := 0 **to** High(c)-1 **do**

c[j] := c[j] + h \* f[j];

c[3] := (c0[0] - c[0]) / c0[0] \* 100;

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

write('X':6);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

print\_results(eyler\_method(0, 30, 1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t) X

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.4690 0.1470 0.0840 33.0000

2.0000 0.3294 0.2455 0.1252 52.9500

3.0000 0.2432 0.3147 0.1422 65.2581

4.0000 0.1885 0.3657 0.1457 73.0676

5.0000 0.1525 0.4053 0.1421 78.2074

6.0000 0.1278 0.4373 0.1349 81.7440

7.0000 0.1099 0.4642 0.1259 84.3007

8.0000 0.0963 0.4873 0.1164 86.2436

9.0000 0.0855 0.5075 0.1070 87.7890

10.0000 0.0765 0.5254 0.0980 89.0662

11.0000 0.0689 0.5415 0.0896 90.1537

12.0000 0.0623 0.5560 0.0817 91.0998

13.0000 0.0565 0.5691 0.0745 91.9356

14.0000 0.0512 0.5809 0.0679 92.6816

15.0000 0.0465 0.5917 0.0618 93.3519

16.0000 0.0423 0.6015 0.0562 93.9571

17.0000 0.0385 0.6103 0.0512 94.5049

18.0000 0.0350 0.6184 0.0466 95.0017

19.0000 0.0318 0.6258 0.0424 95.4529

20.0000 0.0290 0.6324 0.0386 95.8629

21.0000 0.0264 0.6385 0.0351 96.2357

22.0000 0.0240 0.6441 0.0320 96.5747

23.0000 0.0218 0.6491 0.0291 96.8831

24.0000 0.0199 0.6537 0.0265 97.1637

25.0000 0.0181 0.6578 0.0241 97.4191

26.0000 0.0164 0.6616 0.0219 97.6514

27.0000 0.0150 0.6651 0.0199 97.8628

28.0000 0.0136 0.6682 0.0182 98.0551

29.0000 0.0124 0.6711 0.0165 98.2302

30.0000 0.0113 0.6737 0.0150 98.3895

**2 способ:**

**Program** lab14;

**type**

matrix = **array of array of** real;

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.21, 0.12, 0.18);

**var**

c: arr;

**function** right\_parts(t: real; c: arr): arr;

**begin**

SetLength(result, comp\_count);

result[0] := -k[0] \* c[0] - k[1] \* c[0] + k[2] \* c[2];

result[1] := k[0] \* c[0];

result[2] := k[1] \* c[0] - k[2] \* c[2]

**end**;

**function** runge\_kutt\_method(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum\_map(a: real; array1, array2: arr): arr;

**var**

i: integer;

**begin**

SetLength(result, Length(array1));

**for** i := 0 **to** High(result) **do**

result[i] := array1[i] + array2[i] \* a

**end**;

**begin**

SetLength(result, Trunc((stop\_t - start\_t) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], Length(c0) + 1);

SetLength(c, Length(c0));

t := start\_t;

**for** i := 0 **to** High(c0) **do**

c[i] := c0[i];

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := t;

**for** j := 0 **to** High(c) **do**

result[i, j+1] := c[j];

k1 := right\_parts(t, c);

k2 := right\_parts(t + h / 2, sum\_map(h / 2, c, k1));

k3 := right\_parts(t + h / 2, sum\_map(h / 2, c, k2));

k4 := right\_parts(t + h, sum\_map(h, c, k3));

**for** j := 0 **to** High(c) **do**

c[j] := c[j] + h / 6 \* (k1[j] + 2 \* k2[j] + 2 \* k3[j] + k4[j]);

t := t + h

**end**;

**end**;

**procedure** print\_results(res: matrix);

**var**

i, j: integer;

**begin**

write('t':6);

write('CA(t)':14);

write('CB(t)':10);

write('CC(t)':10);

write('X':6);

writeln;

**for** i := 0 **to** High(res) **do**

**begin**

**for** j := 0 **to** High(res[i]) **do**

write(res[i, j]:10:4);

write(((0.7 - res[i, 1]) / 0.7 \* 100):10:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

print\_results(runge\_kutt\_method(0, 30, 1, c))

**end**.

**Ответ:**

t CA(t) CB(t) CC(t) X

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.5090 0.1256 0.0654 27.2820

2.0000 0.3793 0.2180 0.1027 45.8130

3.0000 0.2902 0.2877 0.1221 58.5416

4.0000 0.2281 0.3417 0.1301 67.4096

5.0000 0.1841 0.3848 0.1311 73.6973

6.0000 0.1523 0.4199 0.1278 78.2498

7.0000 0.1286 0.4493 0.1221 81.6258

8.0000 0.1106 0.4743 0.1150 84.1956

9.0000 0.0966 0.4960 0.1074 86.2054

10.0000 0.0853 0.5151 0.0997 87.8195

11.0000 0.0760 0.5320 0.0921 89.1483

12.0000 0.0681 0.5471 0.0848 90.2666

13.0000 0.0614 0.5607 0.0779 91.2255

14.0000 0.0556 0.5729 0.0715 92.0603

15.0000 0.0504 0.5841 0.0655 92.7959

16.0000 0.0458 0.5942 0.0600 93.4504

17.0000 0.0417 0.6033 0.0549 94.0367

18.0000 0.0380 0.6117 0.0502 94.5648

19.0000 0.0347 0.6193 0.0460 95.0423

20.0000 0.0317 0.6263 0.0420 95.4754

21.0000 0.0289 0.6327 0.0384 95.8691

22.0000 0.0264 0.6385 0.0351 96.2274

23.0000 0.0241 0.6438 0.0321 96.5539

24.0000 0.0220 0.6486 0.0293 96.8517

25.0000 0.0201 0.6530 0.0268 97.1235

26.0000 0.0184 0.6571 0.0245 97.3716

27.0000 0.0168 0.6608 0.0224 97.5981

28.0000 0.0154 0.6642 0.0205 97.8051

29.0000 0.0140 0.6672 0.0187 97.9941

30.0000 0.0128 0.6701 0.0171 98.1669

**Выводы**

В ходе работы мы научились составлять программы для численных методов решения систем дифференциальных уравнений на примере кинетики химических реакций.