**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 Е.С. Мухина

(Подпись)

\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ 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**

Исследование кинетических закономерностей протекания химической реакции методом математического моделирования заключается в определении изменения концентраций реагирующих веществ во времени при заданной температуре.

Систему обыкновенных дифференциальных уравнений первого порядка можно решить с использованием численного метода Эйлера, алгоритм которого записывается по уравнению.

Необходимо учитывать, что скорость химической реакции зависит от температуры, поэтому, чтобы использовать кинетическую модель для исследования процесса при различных температурах, необходимо ввести зависимость константы скорости химической реакции от температуры по уравнению Аррениуса.

**Практическая часть**

**Задание 1**

**Задание:** Дана схема химических превращений:



CA0 = 0.8 (моль / л);

СВ0 = 0.2 (моль / л);

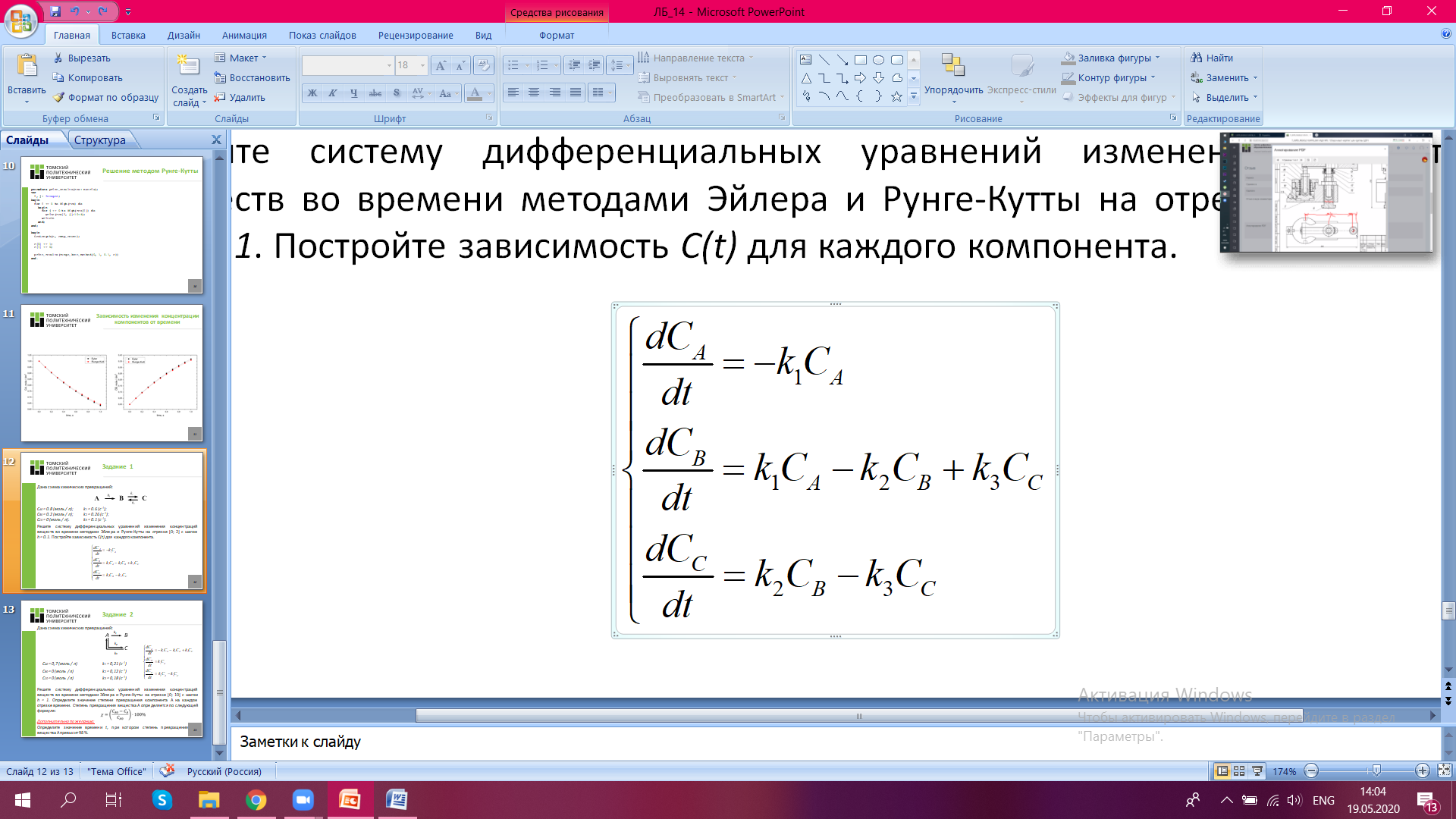
СС0 = 0 (моль / л).

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** lb14;

**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** lb14;

**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** lb14;

**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** lb14;

**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** lb14;

**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, stepen: 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, 100);

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

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

SetLength(c, Length(c0));

t := start\_t;

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

c[i] := c0[i];

i := 0;

stepen := 0;

**while** stepen <= 98 **do**

**begin**

result[i, 0] := t;

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

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

stepen := (0.7 - c[0]) / 0.7 \* 100;

result[i, j+2] := stepen;

**if** stepen > 98 **then**

**begin**

write('Степень превысила 98% и стала равной ', stepen, ' при температуре, равной ', t);

**exit**

**end**;

**end**;

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;

i:= i + 1;

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

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

**end**.

**Ответ:**

Степень превысила 98% и стала равной 98.1668730840961 при температуре, равной 30.

**2 способ:**

**Program** lb14;

**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, stepen: 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, 100);

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

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

SetLength(c, Length(c0));

t := start\_t;

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

c[i] := c0[i];

i := 0;

stepen := 0;

**while** stepen <= 98 **do**

**begin**

result[i, 0] := t;

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

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

stepen := (0.7 - c[0]) / 0.7 \* 100;

result[i, j+2] := stepen;

**if** stepen > 98 **then**

**begin**

write('Степень превысила 98% и стала равной ', stepen, ' при температуре, равной ', t);

**exit**

**end**;

**end**;

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;

i:= i + 1;

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

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

**end**.

**Ответ:**

Степень превысила 98% и стала равной 98.1668730840961 при температуре, равной 30.

**Выводы**

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