**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)

\_26\_\_\_ \_ 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**

**3 вариант**

**Задание**

**Задание:** По данному механизму химической реакции и уравнениям материального баланса вычислить изменение концентраций химических веществ с использованием алгоритмов методов Эйлера и Рунге-Кутта.

При t=0; СА(0)=0,8 моль/л; СВ(0)=0,2 моль/л; СС(0)=0,3 моль/л;

*t = [0;10]* с шагом *h = 1*; *k1=0.05* с-1 *k2=0.98* л/моль с

Постройте график изменения расчетной концентрации каждого вещества во времени.





**Программная реализация:**

**1 способ:**

**Program** IDZ;

**type**

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

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.05, 0.98);

**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] \* c[2];

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

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

**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] := 0.3;

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

**end**.

**Ответ:**

0.0000 0.8000 0.2000 0.3000

1.0000 0.8188 0.1812 0.2812

2.0000 0.8278 0.1722 0.2722

3.0000 0.8323 0.1677 0.2677

4.0000 0.8347 0.1653 0.2653

5.0000 0.8359 0.1641 0.2641

6.0000 0.8366 0.1634 0.2634

7.0000 0.8369 0.1631 0.2631

8.0000 0.8371 0.1629 0.2629

9.0000 0.8372 0.1628 0.2628

10.0000 0.8373 0.1627 0.2627

**2 способ:**

**Program** IDZ;

**type**

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

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.05, 0.98);

**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] \* c[2];

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

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

**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] := 0.3;

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

**end**.

**Ответ:**

0.0000 0.8000 0.2000 0.3000

1.0000 0.8146 0.1854 0.2854

2.0000 0.8233 0.1767 0.2767

3.0000 0.8286 0.1714 0.2714

4.0000 0.8319 0.1681 0.2681

5.0000 0.8340 0.1660 0.2660

6.0000 0.8352 0.1648 0.2648

7.0000 0.8360 0.1640 0.2640

8.0000 0.8365 0.1635 0.2635

9.0000 0.8368 0.1632 0.2632

10.0000 0.8370 0.1630 0.2630

**Выводы**

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