**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**

Выполнил студент гр. 2Д93 Ю.Н.Милованова

(Подпись)

12.05. 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить способ численного решения систем дифференциальных уравнений на примере кинетики химических реакций и реализовать соответствующие программы.

**Теоретическая часть**

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:



**Практическая часть**

**Задание 1**

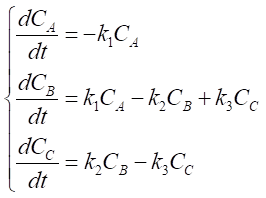
**Исходные данные:**



|  |  |
| --- | --- |
| *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).* |

**Задание:**

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



**Программная реализация:**

Метод Эйлера

**Program** lb141e;

**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(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]:8:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.8;

c[1] := 0.2;

c[2] := 0;

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

**end**.

**Ответы:**

0.0000 0.8000 0.2000 0.0000

0.1000 0.7520 0.2428 0.0052

0.2000 0.7069 0.2817 0.0115

0.3000 0.6645 0.3169 0.0187

0.4000 0.6246 0.3487 0.0267

0.5000 0.5871 0.3774 0.0355

0.6000 0.5519 0.4031 0.0450

0.7000 0.5188 0.4262 0.0550

0.8000 0.4877 0.4468 0.0655

0.9000 0.4584 0.4651 0.0765

1.0000 0.4309 0.4813 0.0878

1.1000 0.4050 0.4955 0.0995

1.2000 0.3807 0.5079 0.1114

1.3000 0.3579 0.5187 0.1234

1.4000 0.3364 0.5279 0.1357

1.5000 0.3162 0.5357 0.1481

1.6000 0.2973 0.5422 0.1605

1.7000 0.2794 0.5476 0.1730

1.8000 0.2627 0.5518 0.1855

1.9000 0.2469 0.5551 0.1980

2.0000 0.2321 0.5575 0.2105

Метод Рунге-Кутты

**Program** lb141r;

**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** rk(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum(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(h / 2, c, k1));

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

k4 := right\_parts(t + h, sum(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]:8:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.8;

c[1] := 0.2;

c[2] := 0;

print\_results(rk(0, 2, 0.1, c))

**end**.

**Ответы:**

0.0000 0.8000 0.2000 0.0000

0.1000 0.7534 0.2409 0.0057

0.2000 0.7095 0.2781 0.0124

0.3000 0.6682 0.3119 0.0199

0.4000 0.6293 0.3425 0.0282

0.5000 0.5927 0.3702 0.0371

0.6000 0.5581 0.3952 0.0467

0.7000 0.5256 0.4177 0.0567

0.8000 0.4950 0.4378 0.0672

0.9000 0.4662 0.4557 0.0781

1.0000 0.4390 0.4716 0.0893

1.1000 0.4135 0.4857 0.1008

1.2000 0.3894 0.4980 0.1126

1.3000 0.3667 0.5088 0.1245

1.4000 0.3454 0.5181 0.1365

1.5000 0.3253 0.5261 0.1487

1.6000 0.3063 0.5328 0.1609

1.7000 0.2885 0.5384 0.1731

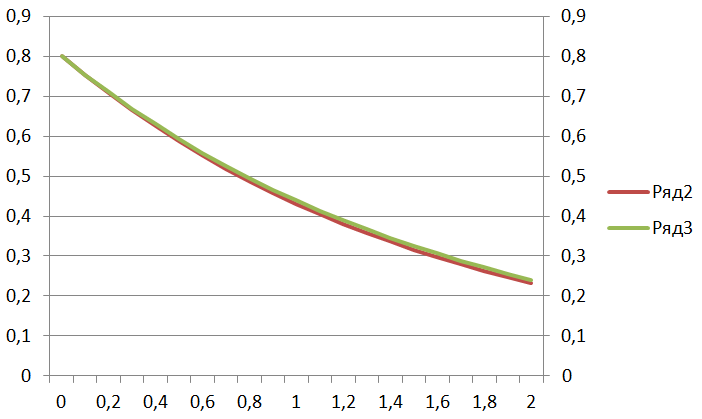
1.8000 0.2717 0.5429 0.1854

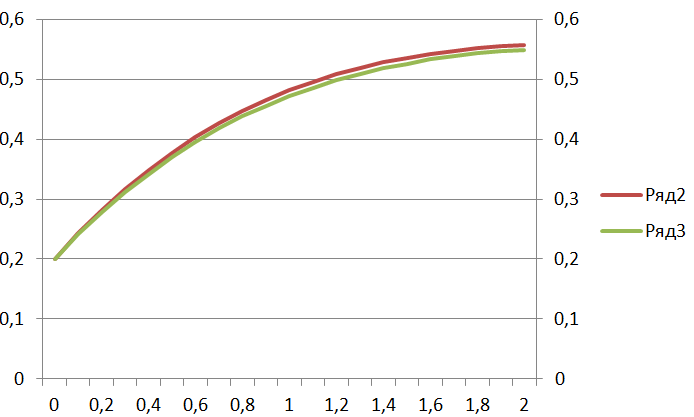
1.9000 0.2559 0.5465 0.1977

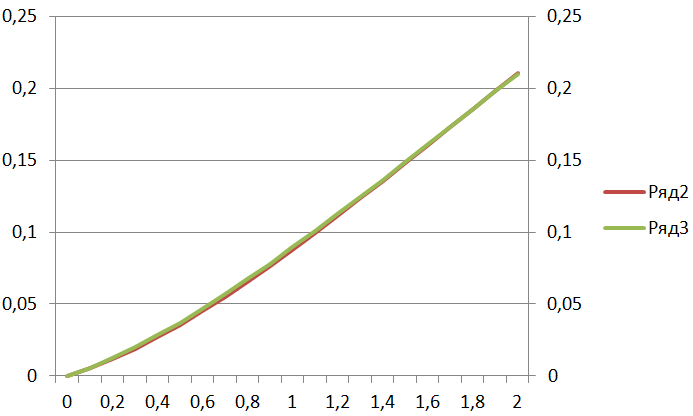
2.0000 0.2410 0.5492 0.2099

Графики зависимости C(t) для каждого компонента представлены ниже.

Где ряд 2 – множество результатов, полученных при вычислении методом Эйлера, ряд 3 – при вычислении методом Рунге-Кутты.







Исходя из полученных графиков, можно сделать вывод о том, что, в целом, разница ответов, полученных при решении этими двумя разными методами не так велика.

**Задание 2**

**Исходные данные:**



|  |  |
| --- | --- |
| *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)* |



**Задание:**

Решить 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**

Метод Эйлера

**Program** lb142e;

**type**

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

arr = **array of** real;

**const**

comp\_count = 4;

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];

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(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];

result[i,4]:= ((c0[0] - c[0]) / c0[0]) \* 100;

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]:8:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

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

**end**.

**Ответы:**

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.5530 0.1470 0.0000 21.0000

2.0000 0.4369 0.2455 0.0176 37.5900

3.0000 0.3451 0.3109 0.0439 50.6961

4.0000 0.2727 0.3540 0.0733 61.0499

5.0000 0.2154 0.3820 0.1026 69.2294

6.0000 0.1702 0.3999 0.1300 75.6913

7.0000 0.1344 0.4110 0.1546 80.7961

8.0000 0.1062 0.4177 0.1761 84.8289

9.0000 0.0839 0.4216 0.1945 88.0148

10.0000 0.0663 0.4236 0.2101 90.5317

Метод Рунге-Кутты:

**Program** lb142r;

**type**

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

arr = **array of** real;

**const**

comp\_count = 4;

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];

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** rk(start\_t, stop\_t, h: real; c0: arr): matrix;

**var**

i, j: integer;

t: real;

k1, k2, k3, k4: arr;

c: arr;

**function** sum(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];

result[i, 4] := ((c0[0] - c[0]) / c0[0]) \* 100;

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

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

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

k4 := right\_parts(t + h, sum(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]:8:4);

writeln

**end**;

**end**;

**begin**

SetLength(c, comp\_count);

c[0] := 0.7;

c[1] := 0;

c[2] := 0;

print\_results(rk(0, 10, 1, c))

**end**.

**Ответы:**

0.0000 0.7000 0.0000 0.0000 0.0000

1.0000 0.5674 0.1251 0.0075 18.9412

2.0000 0.4599 0.2147 0.0253 34.2948

3.0000 0.3728 0.2786 0.0486 46.7402

4.0000 0.3022 0.3239 0.0739 56.8282

5.0000 0.2450 0.3559 0.0992 65.0055

6.0000 0.1986 0.3782 0.1233 71.6339

7.0000 0.1610 0.3936 0.1454 77.0068

8.0000 0.1305 0.4042 0.1653 81.3620

9.0000 0.1058 0.4113 0.1829 84.8923

10.0000 0.0857 0.4160 0.1982 87.7539

**Выводы**

В ходе лабораторной работы был изучен способы численного решения систем дифференциальных уравнений на примере кинетики химических реакций методами Эйлера и Рунге-Кутты. Были реализованы программы по вычислению изменения концентраций веществ во времени с помощью дифференциальных уравнений.

Также для наглядного результата были построены графики зависимости для отображения конечного результата.