**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 И.А. Самсонов

(Подпись)

11.06.2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить численное решение систем дифференциальных уравнений на примере кинетики химических реакций.

**Теоретическая часть**

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

Пусть дана схема химической реакции:

Изменение концентрации каждого компонента во времени:

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.

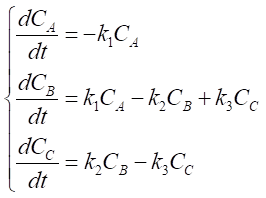
**Практическая часть**

**Задание 1**



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

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.



**Программа, метод Эйлера**

**program** lab\_14\_example;

**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**

**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;

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

**end**.

**Ответ**

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

**Программа, метод Рунге-Кутты**

**program** lb14\_1;

**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**

**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;

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

**end**.

**Ответ**

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

**Задание 2**



|  |  |
| --- | --- |
| *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:

**Программа, метод Эйлера**

**program** lb14\_2;

**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] - 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) + 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]: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**

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

**Программа, метод Рунге-Кутты**

**program** lb14\_2;

**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] - 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];

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

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] := 0.7;

c[1] := 0;

c[2] := 0;

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

**end**.

**Ответ**

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

**Выводы**

В ходе лабораторной работы было изучено численное решение систем дифференциальных уравнений на примере кинетики химических реакций. Были использованы методы Эйлера и Рунге-Кутты. Решены системы дифференциальных уравнений изменения концентраций веществ во времени этими методами. Для первого задания были составлены графики зависимости C(t) для компонентов: CA, CB, CC, для 2 задания была найдена степень превращения вещества А по формуле.