**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

**Индивидуальное домашнее задание «Приближенные методы решения**

**систем дифференциальных уравнений на примере кинетики химических реакций»**

**2 вариант**

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.

**Задание**

1. По 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)=0,64 моль/л; СВ(0)=0,8 моль/л; СС(0)=0,5 моль/л;

*t = [0;1]* с шагом *h = 0.1*; *k1=0.45* *k2=0.15*

1. Построить график изменения расчетной концентрации каждого вещества во времени.

**Программная реализация, метод Эйлера**

**program** eyler;

**type**

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

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.45, 0.15);

**var**

c: arr;

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

**begin**

SetLength(result, comp\_count);

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

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

result[2] := 3 \* (k[0] \* c[0]\* sqr(c[1]) - k[1] \* sqr(c[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] := 0.64;

c[1] := 0.8;

c[2] := 0.5;

writeln(' t ', ' CA ', ' CB ', ' CC ');

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

**end**.

**Ответ**

t CA CB CC

0.0000 0.6400 0.8000 0.5000

0.1000 0.6234 0.7669 0.5497

0.2000 0.6094 0.7389 0.5917

0.3000 0.5976 0.7151 0.6273

0.4000 0.5875 0.6950 0.6574

0.5000 0.5790 0.6780 0.6830

0.6000 0.5718 0.6636 0.7046

0.7000 0.5657 0.6515 0.7228

0.8000 0.5606 0.6412 0.7382

0.9000 0.5563 0.6325 0.7512

1.0000 0.5526 0.6252 0.7622

**Программная реализация, метод Рунге-Кутты**

**program** runge\_kutt\_method1;

**type**

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

arr = **array of** real;

**const**

comp\_count = 3;

k: arr = (0.45, 0.15);

**var**

c: arr;

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

**begin**

SetLength(result, comp\_count);

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

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

result[2] := 3 \* (k[0] \* c[0]\* sqr(c[1]) - k[1] \* sqr(c[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] := 0.64;

c[1] := 0.8;

c[2] := 0.5;

writeln(' t ', ' CA ', ' CB ', ' CC ');

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

**end**.

**Ответ**

t CA CB CC

0.0000 0.6400 0.8000 0.5000

0.1000 0.6247 0.7693 0.5460

0.2000 0.6115 0.7430 0.5855

0.3000 0.6002 0.7204 0.6194

0.4000 0.5905 0.7010 0.6485

0.5000 0.5822 0.6844 0.6734

0.6000 0.5751 0.6701 0.6948

0.7000 0.5690 0.6579 0.7131

0.8000 0.5637 0.6475 0.7288

0.9000 0.5593 0.6385 0.7422

1.0000 0.5554 0.6309 0.7537