Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский химико-технологический университет имени Д.И. Менделеева»

Факультет цифровых технологий и химического инжиниринга

Кафедра информационных компьютерных технологий

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 6**

**ПО КУРСУ**

**«ЦИФРОВОЕ МОДЕЛИРОВАНИЕ ФИЗИКО-ХИМИЧЕСКИХ СИСТЕМ»:**

**«Определение порядка, скорости реакции. Расчет константы скорости реакции. Расчет текущих концентраций для сложных реакций»**

Ведущий преподаватель

Ст. преподаватель Скичко Е.А.

**СТУДЕНТ группы КС-26** Золотухин А.А.

**Москва**

**2024**

# Задание

1. Дана зависимость общего давления идеальной газовой смеси от времени *t* при постоянной температуре в ходе реакции (*Табл. 1*).

| Вариант 7 |  | | |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

По приведенным данным рассчитайте парциальное давление и концентрацию реагента в соответствующие моменты времени, постройте кинетическую кривую концентрации или парциального давления реагента. Графическим и аналитическим методами подбора определите константу скорости и порядок реакции . Рассчитайте время полупревращения исходного вещества в данном опыте, а также его концентрацию и степень превращения в момент времени после начала реакции. В момент начала опыта в системе присутствовал только исходный реагент.

1. Порядок выполнения задания:
   1. В соответствии с заданием (*Табл. 2*) составить математическое описание химического реактора.
   2. Разработать алгоритм и программу расчета.
   3. Провести расчеты изменения концентраций веществ от времени.
   4. Полученные результаты оформить в виде таблиц и графиков.
   5. Составить отчет о проделанной работе.

| № варианта | Тип реакции | Исходная концентрация, кмоль/м3 | Константы скорости | Энергии активации, кДж/моль | Температура, К |
| --- | --- | --- | --- | --- | --- |
| 10 | C8H18i-C8H18C4H10+C4H8 | CC8H18=0,036 | k1=0,12  k2=0,80 | E1=94,2  E2=81,2 | 620 |

Теоретическое обоснование решения

Есть реакция:

|  |  |
| --- | --- |

Реакция (1) есть реакция разложения оксида хлора (I) идет по **цепному механизму[[1]](#footnote-0)** , т.е. является **сложной[[2]](#footnote-1)**. При определении порядка и константы скорости данной реакции буду использовать методы **формальной кинетики[[3]](#footnote-2)**, считая реакцию **односторонней[[4]](#footnote-3)**.

Воспользуюсь интегральным методом подбора уравнения и координат. Для этого потребуется вычислить значения парциального давления оксида хлора (I). Составлю материальный баланс реакции, записывая данные под соответствующей формулой реагента или продукта (*Табл. 1*):

| Уравнение реакции: |  | | |
| --- | --- | --- | --- |
| Начало опыта () |  |  |  |
| Ушло к |  |  |  |
| Текущие значения () |  |  |  |

Общее давление в системе согласно **закону Дальтона[[5]](#footnote-4)**:

|  |  |
| --- | --- |

откуда

|  |  |
| --- | --- |

Теперь есть возможность выразить текущее парциальное давление оксида хлора (I) (обозначу его )через общее давление:

|  |  |
| --- | --- |

Рассчитаю парциальное давление оксида хлора (I) по формуле *(2)*.

Применив **уравнение состояния идеального газа**, или уравнение Менделеева-Клапейрона:

| *,* |  |
| --- | --- |

где *p* - давление газа, Па;

*V* - объем газа, м3;

*n* - количество вещества, моль;

*R* - универсальная газовая постоянная, Дж/(моль \* К);

*T* - термодинамическая температура, К

выражу концентрацию газа, зная, что она численно равна:

|  |  |
| --- | --- |

Объединив *(3)* и *(4)*, получаю:

|  |  |
| --- | --- |

Чтобы найти константу равновесия, можно вспомнить **основной постулат химической кинетики** (закон действующих масс Гульдберга-Вааге): скорость реакции в каждый момент времени пропорциональна произведению концентраций (Ci, моль/м3) реагирующих веществ, возведенных в некоторые степени (ni):

| , |  |
| --- | --- |

где *k* - константа скорости, (моль/м3)(1-n)/с.

С другой стороны, мгновенная скорость (r, моль/(м3\*с)) реакции может быть определена также как производная от концентрации по времени, т.е.:

| , |  |
| --- | --- |

где *V* - объем системы;

*ni* - количество *i*-го вещества, моль,

- производная от количества вещества по времени, которая положительная для продуктов и отрицательна для реагентов.

Перейду к такой кинетической характеристике процессов, как **инвариантная скорость[[6]](#footnote-5)**. Для этого используют понятие **глубины превращения** (, моль), которая определяется уравнением:

| , |  |
| --- | --- |

где *ni0* - количество *i*-го вещества в начале реакции, моль;

*nit* - количество *i*-го вещества в момент времени *t*, моль;

*vi* - стехиометрический коэффициент перед *i*-м веществом в уравнении химической реакции, которому приписывают знак “минус” (в случае реагентов) или знак “плюс” (в случае продуктов).

С учетом *(6)* количество *i*-го вещества:

| . |  |
| --- | --- |

Учитывая *(6)* и *(7)*, получаю выражение для инвариантной скорости:

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

Возвращаясь к *(1)* и используя *(5)* и *(8)*, найду составлю дифференциальное уравнение для **скорости реакции нулевого порядка**:

| . |  |
| --- | --- |

Умножив обе части на *dt*, проинтегрирую в пределах от (начальная концентрация) до (текущая концентрация) и от *0* до *t*:

| , |  |
| --- | --- |

получаю интегральную форму кинетического уравнения:

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

Из *(9)* можно выразить текущую концентрацию:

|  |  |
| --- | --- |

и константу скорости:

|  |  |
| --- | --- |

**Время полупревращения[[7]](#footnote-6)** необратимой реакции нулевого порядка, исходя из *(11)*:

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

Возвращаясь к *(1)* и используя *(5)* и *(8)*, найду составлю дифференциальное уравнение для **скорости реакции первого порядка**:

| . |  |
| --- | --- |

Умножив обе части на *dt* и разделив на , проинтегрирую в пределах от (начальная концентрация) до (текущая концентрация) и от *0* до *t*:

| , |  |
| --- | --- |

получаю интегральную форму кинетического уравнения:

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

Из *(9)* можно выразить текущую концентрацию:

|  |  |
| --- | --- |

и константу скорости:

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

Время полупревращения необратимой реакции первого порядка, исходя из *(14)*:

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

Возвращаясь к *(1)* и используя *(5)* и *(8)*, найду составлю дифференциальное уравнение для **скорости реакции второго порядка**:

| . |  |
| --- | --- |

Умножив обе части на *dt* и разделив на , проинтегрирую в пределах от (начальная концентрация) до (текущая концентрация) и от *0* до *t*:

| , |  |
| --- | --- |

получаю интегральную форму кинетического уравнения:

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

Из *(9)* можно выразить текущую концентрацию:

|  |  |
| --- | --- |

и константу скорости:

|  |  |
| --- | --- |

Время полупревращения необратимой реакции второго порядка, исходя из *(17)*:

|  |  |
| --- | --- |

Найду **степень превращения** в момент времени после начала реакции как отношение убыли концентрации реагента *Cl2* в ходе реакции (, моль/м3) к исходной концентрации (, моль/м3):

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

Есть необратимая последовательная реакция:

| C8H18i-C8H18C4H10+C4H8 |  |
| --- | --- |

**Принцип независимости** элементарных реакций утверждает, что константа скорости элементарной химической реакции не зависит от того, протекают ли в данной системе одновременно другие элементарные реакции. При справедливости этого положения скорость реакции по компоненту равна алгебраической сумме скоростей его образования и расходования во всех стадиях, т.е.:

|  |  |
| --- | --- |

где *rs* - скорость отдельной стадии, моль / (м3 \* с);

*vi* - стехиометрический коэффициент перед компонентом в уравнении реакции для *i*-ой стадии сложного процесса.

Исходя из *(18)* и *(19)*, получаю систему дифференциальных уравнений:

|  |  |
| --- | --- |

Систему из дифференциальных уравнений *(19)-(22)* буду решать с помощью **метода Рунге-Кутты 7-го порядка**. Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка (далее ):

|  |  |
| --- | --- |

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле:

|  |  |
| --- | --- |

Метод седьмого порядка должен иметь по меньшей мере девять стадий:

|  |  |
| --- | --- |

**Уравнение Аррениуса**:

|  |  |
| --- | --- |

где *k* - константа скорости реакции;

*e* - основание натурального логарифма;

*T* - термодинамическая температура, К;

*R* - универсальная газовая постоянная, Дж/(моль \* К);

*Ea* - энергия активации, Дж/моль;

*A* - предэкспоненциальный множитель, показывающий общее число столкновений.

Код

**use** std**::**{fs**::**File, io**::**prelude**::\***};

*// объявление структуры исходных данных*

#[derive(Debug)]

**struct** InitialData {

universal\_gas\_equation\_system**:** f32, *// универсальная газовая постоянная*

temperature\_1**:** f32, *// температура газовой смеси (задание 1)*

time**:** Vec<i32>, *// время (задание 1)*

pressure**:** Vec<f32>, *// давление газовой смеси (задание 1)*

time\_1**:** i32, *// момент времени после начала реакции (задание 1)*

temperature\_2**:** f32, *// температура (задание 2)*

k\_1**:** f32, *// константа скорости первой реакции (задание 2)*

k\_2**:** f32, *// константа скорости второй реакции (задание 2)*

c\_c8h18**:** f32, *// начальная концентрация вещества C8H18 (задание 2)*

c\_ic8h18**:** f32, *// начальная концентрация вещества i-C8H18 (задание 2)*

c\_c4h10**:** f32, *// начальная концентрация вещества C4H10 (задание 2)*

c\_c4h8**:** f32, *// начальная концентрация вещества C4H8 (задание 2)*

dt**:** f32, *// шаг в методе Рунге-Кутты 4-го порядка (задание 2)*

number\_of\_iterations**:** Vec<usize>, *// количество итераций*

}

**impl** InitialData {

*// вычисление давления вещества A*

**fn** pressure\_calc(**&***self*) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..***self***.**pressure**.**len() {

temporary**.**push(*self***.**pressure[0] **-** (2.0 **\*** *self***.**pressure[i] **-** 2.0 **\*** *self***.**pressure[0]));

}

**return** temporary;

}

*// вычисление концентрации вещества A*

**fn** concentration\_calc(**&***self*, pressure\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**pressure\_a**.**len() {

temporary**.**push(pressure\_a[i] **/** (*self***.**temperature\_1 **\*** *self***.**universal\_gas\_equation\_system));

}

**return** temporary;

}

*// вычисление анаморфозы нулевого порядка*

**fn** anamorphosis0(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push(concentration\_a[0] **-** concentration\_a[i]);

}

**return** temporary;

}

*// вычисление анаморфозы первого порядка*

**fn** anamorphosis1(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push((concentration\_a[0] **/** concentration\_a[i])**.**ln());

}

**return** temporary;

}

*// вычисление анаморфозы второго порядка*

**fn** anamorphosis2(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push(1.0 **/** concentration\_a[i] **-** 1.0 **/** concentration\_a[0]);

}

**return** temporary;

}

*// вычисление константы скорости аналитическим методом для 0-го порядка*

**fn** rate\_constant0\_calc(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push((1 **as** f32 **/** *self***.**time[i] **as** f32) **\*** (concentration\_a[0] **-** concentration\_a[i]));

}

**return** temporary;

}

*// вычисление константы скорости аналитическим методом для 1-го порядка*

**fn** rate\_constant1\_calc(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push((1 **as** f32 **/** *self***.**time[i] **as** f32) **\*** (concentration\_a[0] **/** concentration\_a[i])**.**ln());

}

**return** temporary;

}

*// вычисление константы скорости аналитическим методом для 2-го порядка*

**fn** rate\_constant2\_calc(**&***self*, concentration\_a**:** Vec<f32>) **->** Vec<f32> {

**let** **mut** temporary**:** Vec<f32> **=** vec![];

**for** i **in** 0**..**concentration\_a**.**len() {

temporary**.**push((1 **as** f32 **/** *self***.**time[i] **as** f32) **\*** (1.0 **/** concentration\_a[i] **-** 1.0 **/** concentration\_a[0]));

}

**return** temporary;

}

*// вычисление времени полупревращения, концентрации и степени превращения в момент времени t1*

**fn** time\_moment(**&***self*, moment**:** f32, rate\_constant**:** f32, concentration0**:** f32) **->** (f32, f32, f32) {

**let** **mut** half\_turn\_time**:** f32 **=** 0.0;

**let** **mut** concentration\_time\_moment**:** f32 **=** 0.0;

**let** **mut** tranformation\_degree**:** f32 **=** 0.0;

half\_turn\_time **=** 1.0 **/** (rate\_constant **\*** concentration0);

concentration\_time\_moment **=** 1.0 **/** (rate\_constant **\*** moment **+** 1.0 **/** concentration0);

tranformation\_degree **=** rate\_constant **\*** moment **\*** concentration\_time\_moment;

**return** (half\_turn\_time, concentration\_time\_moment, tranformation\_degree);

}

*// решение системы дифференциальных уравнений методом Рунге-Кутты 7-го порядка*

**fn** equation\_system(**&***self*, **mut** c\_c8h18\_mod**:** Vec<f32>, **mut** c\_ic8h18\_mod**:** Vec<f32>, **mut** c\_c4h10\_mod**:** Vec<f32>, **mut** c\_c4h8\_mod**:** Vec<f32>) **->** (Vec<f32>, Vec<f32>, Vec<f32>, Vec<f32>) {

**for** i **in** 0**..***self***.**number\_of\_iterations**.**len() {

**let** h **=** *self***.**dt;

**let** k1\_c8h18 **=** **-***self***.**k\_1 **\*** c\_c8h18\_mod[i];

**let** k1\_ic8h18 **=** *self***.**k\_1 **\*** c\_c8h18\_mod[i] **-** *self***.**k\_2 **\*** c\_ic8h18\_mod[i];

**let** k1\_c4h10 **=** *self***.**k\_2 **\*** c\_ic8h18\_mod[i];

**let** k1\_c4h8 **=** *self***.**k\_2 **\*** c\_ic8h18\_mod[i];

**let** k2\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (1 **as** f32 **/** 12 **as** f32) **\*** k1\_c8h18);

**let** k2\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (1 **as** f32 **/** 12 **as** f32) **\*** k1\_c8h18) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (1 **as** f32 **/** 12 **as** f32) **\*** k1\_ic8h18);

**let** k2\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (1 **as** f32 **/** 12 **as** f32) **\*** k1\_ic8h18);

**let** k2\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (1 **as** f32 **/** 12 **as** f32) **\*** k1\_ic8h18);

**let** k3\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-**10.0 **\*** k1\_c8h18 **+** 11.0 **\*** k2\_c8h18)**/**12.0);

**let** k3\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-**10.0 **\*** k1\_c8h18 **+** 11.0 **\*** k2\_c8h18)**/**12.0) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**10.0 **\*** k1\_ic8h18 **+** 11.0 **\*** k2\_ic8h18)**/**12.0);

**let** k3\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**10.0 **\*** k1\_ic8h18 **+** 11.0 **\*** k2\_ic8h18)**/**12.0);

**let** k3\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**10.0 **\*** k1\_ic8h18 **+** 11.0 **\*** k2\_ic8h18)**/**12.0);

**let** k4\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (2 **as** f32 **/** 12 **as** f32) **\*** k3\_c8h18);

**let** k4\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (2 **as** f32 **/** 12 **as** f32) **\*** k3\_c8h18) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (2 **as** f32 **/** 12 **as** f32) **\*** k3\_ic8h18);

**let** k4\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (2 **as** f32 **/** 12 **as** f32) **\*** k3\_ic8h18);

**let** k4\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (2 **as** f32 **/** 12 **as** f32) **\*** k3\_ic8h18);

**let** k5\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (157.0 **\*** k1\_c8h18 **-** 318.0 **\*** k1\_c8h18 **+** 4.0 **\*** k1\_c8h18 **+** 160.0 **\*** k1\_c8h18)**/**9.0);

**let** k5\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (157.0 **\*** k1\_c8h18 **-** 318.0 **\*** k1\_c8h18 **+** 4.0 **\*** k1\_c8h18 **+** 160.0 **\*** k1\_c8h18)**/**9.0) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (157.0 **\*** k1\_ic8h18 **-** 318.0 **\*** k1\_ic8h18 **+** 4.0 **\*** k1\_ic8h18 **+** 160.0 **\*** k1\_ic8h18)**/**9.0);

**let** k5\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (157.0 **\*** k1\_ic8h18 **-** 318.0 **\*** k1\_ic8h18 **+** 4.0 **\*** k1\_ic8h18 **+** 160.0 **\*** k1\_ic8h18)**/**9.0);

**let** k5\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (157.0 **\*** k1\_ic8h18 **-** 318.0 **\*** k1\_ic8h18 **+** 4.0 **\*** k1\_ic8h18 **+** 160.0 **\*** k1\_ic8h18)**/**9.0);

**let** k6\_c8h18 **=** (**-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-**322.0 **\*** k1\_c8h18 **+** 199.0 **\*** k1\_c8h18 **+** 108.0 **\*** k1\_c8h18 **-** 131.0 **\*** k1\_c8h18)**/**30.0)) **as** f32;

**let** k6\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-**322.0 **\*** k1\_c8h18 **+** 199.0 **\*** k1\_c8h18 **+** 108.0 **\*** k1\_c8h18 **-** 131.0 **\*** k1\_c8h18)**/**30.0) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**322.0 **\*** k1\_ic8h18 **+** 199.0 **\*** k1\_ic8h18 **+** 108.0 **\*** k1\_ic8h18 **-** 131.0 **\*** k1\_ic8h18)**/**30.0);

**let** k6\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**322.0 **\*** k1\_ic8h18 **+** 199.0 **\*** k1\_ic8h18 **+** 108.0 **\*** k1\_ic8h18 **-** 131.0 **\*** k1\_ic8h18)**/**30.0);

**let** k6\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-**322.0 **\*** k1\_ic8h18 **+** 199.0 **\*** k1\_ic8h18 **+** 108.0 **\*** k1\_ic8h18 **-** 131.0 **\*** k1\_ic8h18)**/**30.0);

**let** k7\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** ((3158 **as** f32 **/** 45 **as** f32) **\*** k1\_c8h18 **-** (638 **as** f32 **/** 6 **as** f32) **\*** k2\_c8h18 **-** (23 **as** f32 **/** 2 **as** f32) **\*** k3\_c8h18 **+** (157 **as** f32 **/** 3 **as** f32) **\*** k4\_c8h18 **+** (157 **as** f32 **/** 45 **as** f32) **\*** k6\_c8h18));

**let** k7\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** ((3158 **as** f32 **/** 45 **as** f32) **\*** k1\_c8h18 **-** (638 **as** f32 **/** 6 **as** f32) **\*** k2\_c8h18 **-** (23 **as** f32 **/** 2 **as** f32) **\*** k3\_c8h18 **+** (157 **as** f32 **/** 3 **as** f32) **\*** k4\_c8h18 **+** (157 **as** f32 **/** 45 **as** f32) **\*** k6\_c8h18)) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((3158 **as** f32 **/** 45 **as** f32) **\*** k1\_ic8h18 **-** (638 **as** f32 **/** 6 **as** f32) **\*** k2\_ic8h18 **-** (23 **as** f32 **/** 2 **as** f32) **\*** k3\_ic8h18 **+** (157 **as** f32 **/** 3 **as** f32) **\*** k4\_ic8h18 **+** (157 **as** f32 **/** 45 **as** f32) **\*** k6\_ic8h18));

**let** k7\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((3158 **as** f32 **/** 45 **as** f32) **\*** k1\_ic8h18 **-** (638 **as** f32 **/** 6 **as** f32) **\*** k2\_ic8h18 **-** (23 **as** f32 **/** 2 **as** f32) **\*** k3\_ic8h18 **+** (157 **as** f32 **/** 3 **as** f32) **\*** k4\_ic8h18 **+** (157 **as** f32 **/** 45 **as** f32) **\*** k6\_ic8h18));

**let** k7\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((3158 **as** f32 **/** 45 **as** f32) **\*** k1\_ic8h18 **-** (638 **as** f32 **/** 6 **as** f32) **\*** k2\_ic8h18 **-** (23 **as** f32 **/** 2 **as** f32) **\*** k3\_ic8h18 **+** (157 **as** f32 **/** 3 **as** f32) **\*** k4\_ic8h18 **+** (157 **as** f32 **/** 45 **as** f32) **\*** k6\_ic8h18));

**let** k8\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-** (53 **as** f32 **/** 14 **as** f32) **\*** k1\_c8h18 **+** (38 **as** f32 **/** 7 **as** f32) **\*** k2\_c8h18 **-** (3 **as** f32 **/** 14 **as** f32) **\*** k3\_c8h18 **-** (65 **as** f32 **/** 72 **as** f32) **\*** k5\_c8h18 **+** (29 **as** f32 **/** 90 **as** f32) **\*** k7\_c8h18));

**let** k8\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** (**-** (53 **as** f32 **/** 14 **as** f32) **\*** k1\_c8h18 **+** (38 **as** f32 **/** 7 **as** f32) **\*** k2\_c8h18 **-** (3 **as** f32 **/** 14 **as** f32) **\*** k3\_c8h18 **-** (65 **as** f32 **/** 72 **as** f32) **\*** k5\_c8h18 **+** (29 **as** f32 **/** 90 **as** f32) **\*** k7\_c8h18)) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-** (53 **as** f32 **/** 14 **as** f32) **\*** k1\_ic8h18 **+** (38 **as** f32 **/** 7 **as** f32) **\*** k2\_ic8h18 **-** (3 **as** f32 **/** 14 **as** f32) **\*** k3\_ic8h18 **-** (65 **as** f32 **/** 72 **as** f32) **\*** k5\_ic8h18 **+** (29 **as** f32 **/** 90 **as** f32) **\*** k7\_ic8h18));

**let** k8\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-** (53 **as** f32 **/** 14 **as** f32) **\*** k1\_ic8h18 **+** (38 **as** f32 **/** 7 **as** f32) **\*** k2\_ic8h18 **-** (3 **as** f32 **/** 14 **as** f32) **\*** k3\_ic8h18 **-** (65 **as** f32 **/** 72 **as** f32) **\*** k5\_ic8h18 **+** (29 **as** f32 **/** 90 **as** f32) **\*** k7\_ic8h18));

**let** k8\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** (**-** (53 **as** f32 **/** 14 **as** f32) **\*** k1\_ic8h18 **+** (38 **as** f32 **/** 7 **as** f32) **\*** k2\_ic8h18 **-** (3 **as** f32 **/** 14 **as** f32) **\*** k3\_ic8h18 **-** (65 **as** f32 **/** 72 **as** f32) **\*** k5\_ic8h18 **+** (29 **as** f32 **/** 90 **as** f32) **\*** k7\_ic8h18));

**let** k9\_c8h18 **=** **-***self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** ((56 **as** f32 **/** 25 **as** f32) **\*** k1\_c8h18 **+** (283 **as** f32 **/** 14 **as** f32) **\*** k2\_c8h18 **-** (119 **as** f32 **/** 6 **as** f32) **\*** k3\_c8h18 **-** (26 **as** f32 **/** 7 **as** f32) **\*** k4\_c8h18 **-** (13 **as** f32 **/** 15 **as** f32) **\*** k5\_c8h18 **+** (149 **as** f32 **/** 32 **as** f32) **\*** k6\_c8h18 **-** (25 **as** f32 **/** 9 **as** f32) **\*** k7\_c8h18 **+** (27 **as** f32 **/** 25 **as** f32) **\*** k8\_c8h18));

**let** k9\_ic8h18 **=** *self***.**k\_1 **\*** (c\_c8h18\_mod[i] **+** h **\*** ((56 **as** f32 **/** 25 **as** f32) **\*** k1\_c8h18 **+** (283 **as** f32 **/** 14 **as** f32) **\*** k2\_c8h18 **-** (119 **as** f32 **/** 6 **as** f32) **\*** k3\_c8h18 **-** (26 **as** f32 **/** 7 **as** f32) **\*** k4\_c8h18 **-** (13 **as** f32 **/** 15 **as** f32) **\*** k5\_c8h18 **+** (149 **as** f32 **/** 32 **as** f32) **\*** k6\_c8h18 **-** (25 **as** f32 **/** 9 **as** f32) **\*** k7\_c8h18 **+** (27 **as** f32 **/** 25 **as** f32) **\*** k8\_c8h18)) **-** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((56 **as** f32 **/** 25 **as** f32) **\*** k1\_ic8h18 **+** (283 **as** f32 **/** 14 **as** f32) **\*** k2\_ic8h18 **-** (119 **as** f32 **/** 6 **as** f32) **\*** k3\_ic8h18 **-** (26 **as** f32 **/** 7 **as** f32) **\*** k4\_ic8h18 **-** (13 **as** f32 **/** 15 **as** f32) **\*** k5\_ic8h18 **+** (149 **as** f32 **/** 32 **as** f32) **\*** k6\_ic8h18 **-** (25 **as** f32 **/** 9 **as** f32) **\*** k7\_ic8h18 **+** (27 **as** f32 **/** 25 **as** f32) **\*** k8\_ic8h18));

**let** k9\_c4h10 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((56 **as** f32 **/** 25 **as** f32) **\*** k1\_ic8h18 **+** (283 **as** f32 **/** 14 **as** f32) **\*** k2\_ic8h18 **-** (119 **as** f32 **/** 6 **as** f32) **\*** k3\_ic8h18 **-** (26 **as** f32 **/** 7 **as** f32) **\*** k4\_ic8h18 **-** (13 **as** f32 **/** 15 **as** f32) **\*** k5\_ic8h18 **+** (149 **as** f32 **/** 32 **as** f32) **\*** k6\_ic8h18 **-** (25 **as** f32 **/** 9 **as** f32) **\*** k7\_ic8h18 **+** (27 **as** f32 **/** 25 **as** f32) **\*** k8\_ic8h18));

**let** k9\_c4h8 **=** *self***.**k\_2 **\*** (c\_ic8h18\_mod[i] **+** h **\*** ((56 **as** f32 **/** 25 **as** f32) **\*** k1\_ic8h18 **+** (283 **as** f32 **/** 14 **as** f32) **\*** k2\_ic8h18 **-** (119 **as** f32 **/** 6 **as** f32) **\*** k3\_ic8h18 **-** (26 **as** f32 **/** 7 **as** f32) **\*** k4\_ic8h18 **-** (13 **as** f32 **/** 15 **as** f32) **\*** k5\_ic8h18 **+** (149 **as** f32 **/** 32 **as** f32) **\*** k6\_ic8h18 **-** (25 **as** f32 **/** 9 **as** f32) **\*** k7\_ic8h18 **+** (27 **as** f32 **/** 25 **as** f32) **\*** k8\_ic8h18));

c\_c8h18\_mod**.**push(c\_c8h18\_mod[i] **+** h**/**840.0 **\*** (41.0**\***k1\_c8h18 **+** 216.0**\***k4\_c8h18 **+** 27.0**\***k5\_c8h18 **+** 272.0**\***k6\_c8h18 **+** 27.0**\***k7\_c8h18 **+** 216.0**\***k8\_c8h18 **+** 41.0**\***k9\_c8h18));

c\_ic8h18\_mod**.**push(c\_ic8h18\_mod[i] **+** h**/**840.0 **\*** (41.0**\***k1\_ic8h18 **+** 216.0**\***k4\_ic8h18 **+** 27.0**\***k5\_ic8h18 **+** 272.0**\***k6\_ic8h18 **+** 27.0**\***k7\_ic8h18 **+** 216.0**\***k8\_ic8h18 **+** 41.0**\***k9\_ic8h18));

c\_c4h10\_mod**.**push(c\_c4h10\_mod[i] **+** h**/**840.0 **\*** (41.0**\***k1\_c4h10 **+** 216.0**\***k4\_c4h10 **+** 27.0**\***k5\_c4h10 **+** 272.0**\***k6\_c4h10 **+** 27.0**\***k7\_c4h10 **+** 216.0**\***k8\_c4h10 **+** 41.0**\***k9\_c4h10));

c\_c4h8\_mod**.**push(c\_c4h8\_mod[i] **+** h**/**840.0 **\*** (41.0**\***k1\_c4h8 **+** 216.0**\***k4\_c4h8 **+** 27.0**\***k5\_c4h8 **+** 272.0**\***k6\_c4h8 **+** 27.0**\***k7\_c4h8 **+** 216.0**\***k8\_c4h8 **+** 41.0**\***k9\_c4h8));

}

**return** (c\_c8h18\_mod, c\_ic8h18\_mod, c\_c4h10\_mod, c\_c4h8\_mod);

}

}

**fn** main() {

**let** init **=** InitialData {

temperature\_1**:** 458.0,

time**:** vec![0, 30, 60, 120, 240, 300],

pressure**:** vec![4.21e3, 4.51e3, 4.73e3, 5.04e3, 5.40e3, 5.52e3],

time\_1**:** 250,

universal\_gas\_equation\_system**:** 8.31,

temperature\_2**:** 620.0,

k\_1**:** 0.12,

k\_2**:** 0.80,

c\_c8h18**:** 0.036e3,

c\_ic8h18**:** 0.0,

c\_c4h10**:** 0.0,

c\_c4h8**:** 0.0,

dt**:** 0.01,

number\_of\_iterations**:** (0**..**1000)**.**collect(),

};

println!("Task\_1:");

**let** pressure\_a **=** init**.**pressure\_calc();

**let** concentration\_a **=** init**.**concentration\_calc(pressure\_a**.**clone());

**let** concentration\_anamorphosis\_0 **=** init**.**anamorphosis0(concentration\_a**.**clone());

**let** concentration\_anamorphosis\_1 **=** init**.**anamorphosis1(concentration\_a**.**clone());

**let** concentration\_anamorphosis\_2 **=** init**.**anamorphosis2(concentration\_a**.**clone());

**let** rate\_constant0 **=** init**.**rate\_constant0\_calc(concentration\_a**.**clone());

**let** rate\_constant1 **=** init**.**rate\_constant1\_calc(concentration\_a**.**clone());

**let** rate\_constant2 **=** init**.**rate\_constant2\_calc(concentration\_a**.**clone());

println!("\tTask\_1\_1:");

println!("\t\tt, sec\t\tp\_A, Pa\t\tC\_A, mol/m^3");

**for** i **in** 0**..**pressure\_a**.**len() {

println!("\t\tt[{}] = {}\tp\_A[{}] = {}\tC\_A[{}] = {}", i, init**.**time[i], i, pressure\_a[i], i, concentration\_a[i]);

}

println!("\n\tTask\_1\_2:");

println!("\t\tk0, mol/(m^3 \* sec)\tk1, 1/sec\t\tk2, 1/(mol \* m^3 \* sec))");

**for** i **in** 0**..**pressure\_a**.**len() {

println!("\t\tk0[{}] = {}\tk1[{}] = {}\tk2[{}] = {}", i, rate\_constant0[i], i, rate\_constant1[i], i, rate\_constant2[i]);

}

println!("\n\tTask\_1\_3:");

println!("\t\ta\_0, mol/m^3\t\ta\_1, mol/m^3\t\ta\_2, mol/m^3");

**for** i **in** 0**..**pressure\_a**.**len() {

println!("\t\ta\_0[{}] = {}\t\ta\_1[{}] = {}\ta\_2[{}] = {}", i, concentration\_anamorphosis\_0[i], i, concentration\_anamorphosis\_1[i], i, concentration\_anamorphosis\_2[i]);

}

write\_to\_file(concentration\_a**.**clone(), init**.**time**.**clone(), "D:/lab\_PhysChem/Lab6/C(dt)/C(dt).txt");

write\_to\_file(concentration\_anamorphosis\_0**.**clone(), init**.**time**.**clone(), "D:/lab\_PhysChem/Lab6/C\_a0(dt)/C\_a0(dt).txt");

write\_to\_file(concentration\_anamorphosis\_1**.**clone(), init**.**time**.**clone(), "D:/lab\_PhysChem/Lab6/C\_a1(dt)/C\_a1(dt).txt");

write\_to\_file(concentration\_anamorphosis\_2**.**clone(), init**.**time**.**clone(), "D:/lab\_PhysChem/Lab6/C\_a2(dt)/C\_a2(dt).txt");

println!("\n\tTask\_1\_4:");

**let** (half\_turn\_time, concentration\_time\_moment, tranformation\_degree) **=** init**.**time\_moment(init**.**time[1] **as** f32, rate\_constant2[1], concentration\_a[0], concentration\_a[1]);

println!("\n\tt(1/2) = {} sec\n\tC(t[1]) = {} mol/m^3\n\ta(t[1]) = {}", half\_turn\_time, concentration\_time\_moment, tranformation\_degree);

**let** **mut** c\_c8h18\_mod**:** Vec<f32> **=** vec![];

c\_c8h18\_mod**.**push(init**.**c\_c8h18);

**let** **mut** c\_ic8h18\_mod**:** Vec<f32> **=** vec![];

c\_ic8h18\_mod**.**push(init**.**c\_ic8h18);

**let** **mut** c\_c4h10\_mod**:** Vec<f32> **=** vec![];

c\_c4h10\_mod**.**push(init**.**c\_c4h10);

**let** **mut** c\_c4h8\_mod**:** Vec<f32> **=** vec![];

c\_c4h8\_mod**.**push(init**.**c\_c4h8);

(c\_c8h18\_mod, c\_ic8h18\_mod, c\_c4h10\_mod, c\_c4h8\_mod) **=** init**.**equation\_system(c\_c8h18\_mod, c\_ic8h18\_mod, c\_c4h10\_mod, c\_c4h8\_mod);

write\_to\_file\_conc(c\_c8h18\_mod, init**.**number\_of\_iterations**.**clone(), "D:/lab\_PhysChem/Lab6/Concentrations/C\_C8H18/c\_c8h18(t).txt");

write\_to\_file\_conc(c\_ic8h18\_mod, init**.**number\_of\_iterations**.**clone(), "D:/lab\_PhysChem/Lab6/Concentrations/C\_i-C8H18/c\_ic8h18(t).txt");

write\_to\_file\_conc(c\_c4h10\_mod, init**.**number\_of\_iterations**.**clone(), "D:/lab\_PhysChem/Lab6/Concentrations/C\_C4H10/c\_c4h10(t).txt");

write\_to\_file\_conc(c\_c4h8\_mod, init**.**number\_of\_iterations**.**clone(), "D:/lab\_PhysChem/Lab6/Concentrations/C\_C4H8/c\_c4h8(t).txt");

}

*// вывод в файл анаморфоз*

**fn** write\_to\_file(vector**:** Vec<f32>, time**:** Vec<i32>, path**:** **&**str) **->** std**::**io**::**Result<()> {

**let** **mut** file **=** File**::**create(path)**?**;

**for** i **in** 0**..**time**.**len() {

write!(file, "{} {}\n", time[i], vector[i])**?**;

}

Ok(())

}

*// вывод в файл системы уравнений*

**fn** write\_to\_file\_conc(vector**:** Vec<f32>, number\_of\_iterations**:** Vec<usize>, path**:** **&**str) **->** std**::**io**::**Result<()> {

**let** **mut** file **=** File**::**create(path)**?**;

**for** i **in** 0**..**number\_of\_iterations**.**len() {

write!(file, "{} {}\n", number\_of\_iterations[i], vector[i])**?**;

}

Ok(())

}

Результаты расчетов

Task\_1:

Task\_1\_1:

t, sec p\_A, Pa C\_A, mol/m^3

t[0] = 0 p\_A[0] = 4210 C\_A[0] = 1.106154

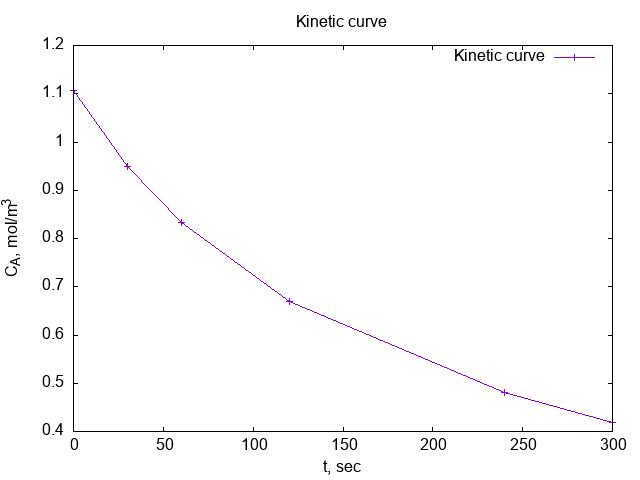
t[1] = 30 p\_A[1] = 3610 C\_A[1] = 0.9485073

t[2] = 60 p\_A[2] = 3170 C\_A[2] = 0.83289975

t[3] = 120 p\_A[3] = 2550 C\_A[3] = 0.6699982

t[4] = 240 p\_A[4] = 1830 C\_A[4] = 0.48082227

t[5] = 300 p\_A[5] = 1590 C\_A[5] = 0.4177636



Task\_1\_2:

k0, mol/(m^3 \* sec) k1, 1/sec k2, 1/(mol \* m^3 \* sec))

k0[0] = NaN k1[0] = NaN k2[0] = NaN

k0[1] = 0.005254889 k1[1] = 0.005125165 k2[1] = 0.005008495

k0[2] = 0.0045542372 k1[2] = 0.004728852 k2[2] = 0.00494319

k0[3] = 0.0036346314 k1[3] = 0.0041780784 k2[3] = 0.004904234

k0[4] = 0.0026055488 k1[4] = 0.0034714448 k2[4] = 0.0048989053

k0[5] = 0.0022946347 k1[5] = 0.0032457623 k2[5] = 0.00496555

Из результатов аналитического метода видно, что наименьшее отклонение результатов дает третий столбец, соответственно это и есть наша константа скорости, а порядок реакции - 2.

Task\_1\_3:

a\_0, mol/m^3 a\_1, a\_2, m^3/mol

a\_0[0] = 0 a\_1[0] = 0 a\_2[0] = 0

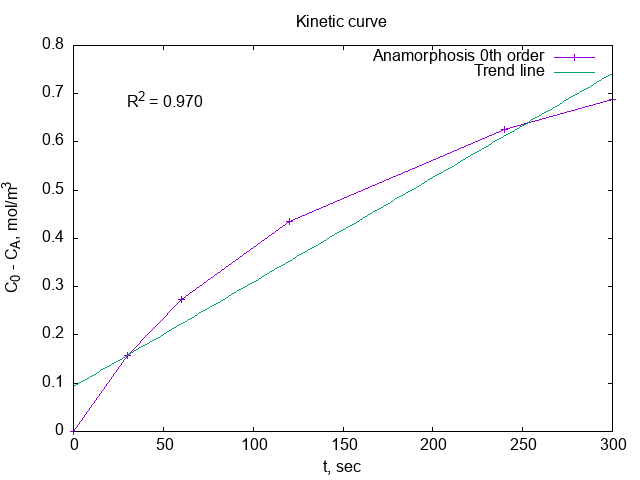
a\_0[1] = 0.15764666 a\_1[1] = 0.15375493 a\_2[1] = 0.15025485

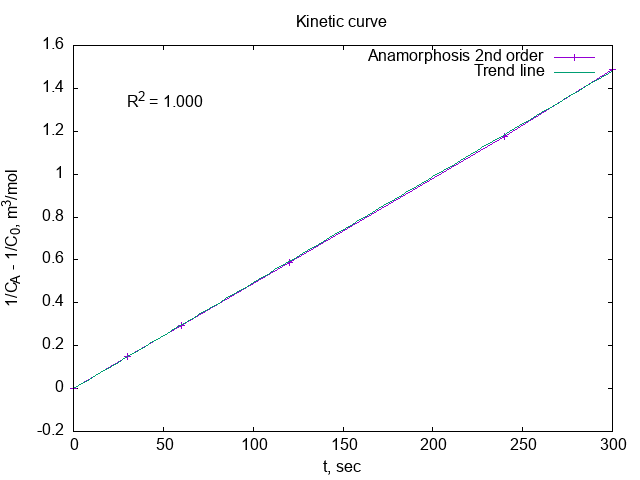
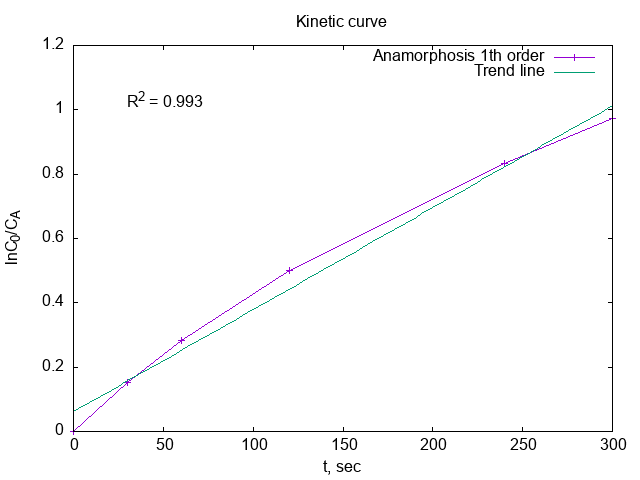
a\_0[2] = 0.27325422 a\_1[2] = 0.2837311 a\_2[2] = 0.2965914

a\_0[3] = 0.43615574 a\_1[3] = 0.50136936 a\_2[3] = 0.588508

a\_0[4] = 0.6253317 a\_1[4] = 0.8331467 a\_2[4] = 1.1757373

a\_0[5] = 0.6883904 a\_1[5] = 0.97372866 a\_2[5] = 1.4896649





Решая графическом способом, можно заметить, что сумма квадратов отклонений от линейной функции наблюдается на третьем графике, который является 2 порядком протекания реакции. Константа скорости реакции равна угловому коэффициенту прямой, т.е. *k = 0.00494 c-1*.

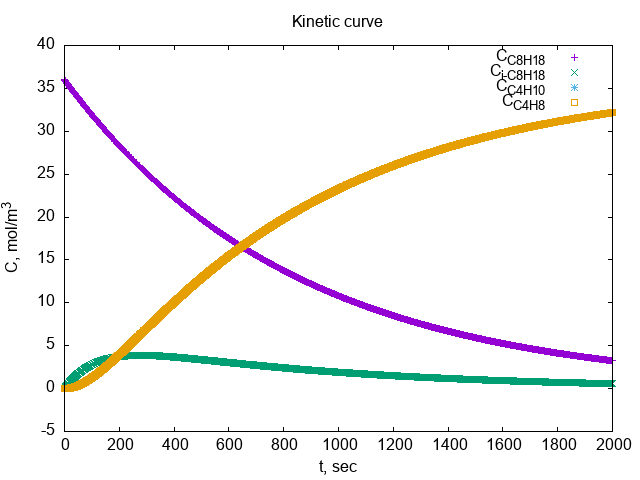
Task\_1\_4:

t(1/2) = 182.88458 sec

C(t\_1) = 0.46732667 mol/m^3

a(t\_1) = 0.57752115

Task\_2:



Для продукта реакции, из-за образования промежуточного вещества, на кинетической кривой имеется точка перегиба, а скорость образования в начальный момент равна нулю, затем проходит через максимум и к концу реакции снова стремится к нулю. Кинетическая кривая промежуточного вещества имеет вид кривой с максимумом, так как в одних стадиях промежуточное вещество получается, а в других расходуется. К концу реакции концентрация промежуточного вещества уменьшается до нуля. Скорость получения промежуточного вещества с течением времени уменьшается, а скорость его расходования увеличивается, поэтому кинетическая кривая проходит через максимум. В максимуме кинетической кривой концентрация промежуточного вещества постоянна и скорость образования равна нулю.

1. механизм, при котором исходные вещества вступают в цепь превращений с участием промежуточных активных частиц (*интермедиатов*) и их регенерацией в каждом *элементарном акте* (в одну стадию) реакции. [↑](#footnote-ref-0)
2. многостадийная реакция. [↑](#footnote-ref-1)
3. раздел химической кинетики, в котором рассматривают зависимость скорости реакции от концентраций реагирующих веществ в закрытых системах при постоянной температуре. [↑](#footnote-ref-2)
4. реакции, в которых конечные продукты отсутствуют в начальный момент времени. [↑](#footnote-ref-3)
5. давление смеси идеальных газов (p, Па) равно сумме парциальных давлений (pi, Па), входящих в нее газов (n). [↑](#footnote-ref-4)
6. скорость, которую используют при процессах с известной стехиометрией и которая не зависит от выбора компонента. [↑](#footnote-ref-5)
7. время, за которое исходная концентрация уменьшится в 2 раза. [↑](#footnote-ref-6)