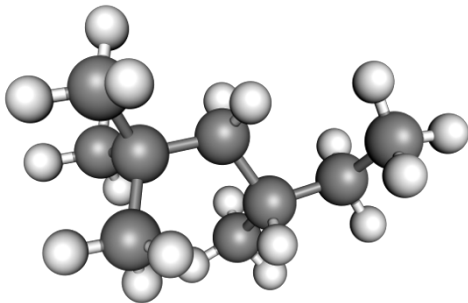


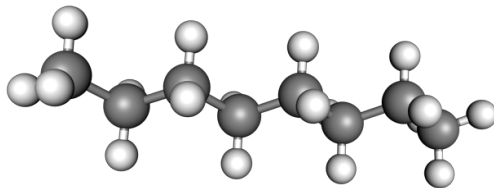
Октановое число

Октановое число – это условная величина, характеризующая детонационную стойкость и численно равная процентному содержанию изооктана в эталонной смеси с н-гептаном, которая по детонационной стойкости эквивалентна испытываемому топливу в условиях стандартного одноцилиндрового двигателя.

Эталонные углеводороды

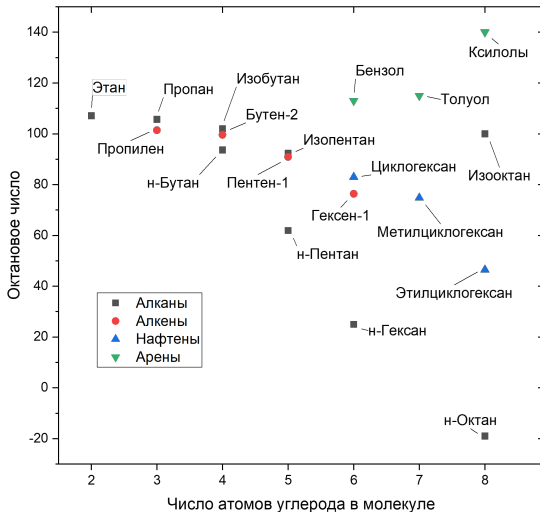


2,2,4-триметилпентан (изооктан)



н-гептан

Октановые числа индивидуальных углеводородов



- Наименьшим ОЧ обладают алканы нормального строения, наивысшим – ароматические УВ.
- ОЧ нормальных алканов резко снижается с увеличением их молекулярной массы.
- ОЧ изопарафинов значительно выше, чем у алканов нормального строения.
- Олефиновые УВ обладают более высокими ОЧ в сравнении с алканами с тем же числом атомов углерода.
- ОЧ аренов повышается с увеличением числа углеродных атомов.

Расчет октанового числа смешения

$$R = R_1 + C_1 \cdot (R_2 - R_1 \cdot J_x) + C_2 \cdot (O_1 - O_2) + C_3 \cdot (A_1 - A_2) \quad (1)$$

R – Октановое число смеси по исследовательскому методу;

R_0 – Октановое число каждого компонента по исследовательскому методу;

R_1 – Сумма произведений октанового числа каждого компонента (ОЧ ИМ) на его объемную долю (октановое число, средневзвешенное по объему);

R_2 – Сумма произведений R_0 и J каждого компонента, умноженных на его объемную долю (произведения, средневзвешенные по объему);

J_x – Сумма произведений чувствительности каждого компонента J на его объемную долю;

O_1 – Сумма произведений квадрата процентного содержания олефинов в каждом компоненте на его объемную долю;

O_2 – Квадрат суммы произведений процентного содержания олефинов в каждом компоненте на его объемную долю (квадрат суммы, средневзвешенной по объему);

A_1 – Сумма произведений квадрата содержания ароматических углеводородов в каждом компоненте на его объемную долю;

A_2 – Квадрат суммы произведений процентного содержания ароматических углеводородов в каждом компоненте на его объемную долю (квадрат суммы, средневзвешенной по объему).

Расчет октанового числа смешения

Изучение данных по смешению компонентов бензина показало, что нелинейность поведения компонентов бензина при смешении можно описать следующим образом:

$$P_{\text{расч}} = P_{\text{об}} + I_{(1,2)} \cdot X_1 \cdot X_2 + I_{(1,3)} \cdot X_1 \cdot X_3 + \dots + I_{(8,9)} \cdot X_8 \cdot X_9 \quad (2)$$

$P_{\text{расч}}$ — Расчетное свойство;

$P_{\text{об}}$ — Средневзвешенное (по объему) свойство;

$I_{(1,2)} \dots I_{(8,9)}$ — Коэффициенты взаимодействия компонентов;

$X_1 \dots X_9$ — Объемная доля каждого компонента.

Коэффициент взаимодействия для бинарной смеси можно рассчитать по следующей формуле:

$$I_{(A,B)} = \frac{P_{\text{факт}} - P_{\text{об}}}{V_A \cdot V_B} \quad (3)$$

$I_{(A,B)}$ — Коэффициент взаимодействия компонентов A и B ;
 $P_{\text{факт}}$ — Свойство смеси, определенное в лаборатории;
 $P_{\text{об}}$ — Средневзвешенное (по объему) свойство смеси;
 V_A, V_B — Объемная доля компонентов A и B .

*Паркаш Суриндер Справочник по переработке нефти / Перевод с английского. – М: ООО Премиум Инжиниринг, 2012. – 776 с., ил. – (Промышленный инжиниринг)

Расчет октанового числа смешения

Октановое число смешения можно представить в виде суммы двух составляющих: аддитивной и неаддитивной:

$$\text{ОЧ}_{\text{см}} = \sum_{i=1}^n (\text{ОЧ}_i \cdot C_i) + B \quad (4)$$

$\text{ОЧ}_{\text{см}}$ — Октановое число смешения компонентов бензина;

B — Суммарное отклонение октановых чисел от аддитивности;

C_i — Концентрация i -го компонента, отн. ед.;

ОЧ_i — Октановое число i -го компонента;

n — Количество компонентов.

Суммарное отклонение B определяется следующим образом:

$$B = \sum_{i=1}^n \sum_{j=i+1}^n B_i \cdot B_j \cdot C_i \cdot C_j \quad (5)$$

B_i, B_j — величины, характеризующие склонность i -й молекулы к межмолекулярному взаимодействию с j -й молекулой.

Программная реализация



```
1 def get_octane_nummber(  
2     flow: Flow,  
3     ron: np.ndarray = const.RON,  
4     bi: np.ndarray = const.Bi  
5 ) -> float:  
6     mf = flow.mass_fractions  
7     additive = (ron * mf).sum()  
8  
9     non_additive = 0  
10    for i, _ in enumerate(mf):  
11        for j in range(i+1, mf.shape[0]):  
12            non_additive += bi[i] * bi[j] * mf[i] * mf[j]  
13  
14    return additive + non_additive  
15
```

Задача

Определить соотношение 6-ти потоков различного состава при котором достигается заданное октановое число (92, 95, пункта по исследовательскому методу)

Этапы решения

- ✓ Реализовать расчет октанового числа смешения
- Реализовать целевую функцию
- Применить один из методов минимизации функции
- Вывести результаты расчета

Описание класса Blend



Методы класса Blend

Метод	Описание
<code>def blend(self, *flows: Flow) -> Flow</code>	Расчет октанового числа смешения неограниченного количества объектов класса Flow
<code>def calculate_ratio(self, expected_value: float, *flows: Flow) -> dict</code>	Расчет соотношения потоков для достижения требуемого октанового числа смешения

Метод SLSQP

- Метод SLSQP использует последовательное квадратичное программирование для минимизации функций нескольких переменных с любой комбинацией границ, а также ограничений на равенство и неравенство.
- Метод включает в себя подпрограмму оптимизации SLSQP, первоначально реализованную Дитером Крафтом (Dieter Kraft).
- Обратите внимание, что оболочка обрабатывает бесконечные значения в границах, преобразуя их в большие значения с плавающей точкой.

Метод SLSQP предназначен для решения задач оптимизации функций в следующей форме:

$$\begin{aligned}
 & \min_x f(x) \\
 & \text{при } c_j(x) = 0, \quad j \in \mathcal{E} \\
 & \quad c_j(x) \geq 0, \quad j \in \mathcal{I} \\
 & lb_i \leq x_i \leq ub_i, \quad i = 1, \dots, N.
 \end{aligned} \tag{6}$$

Здесь \mathcal{E} и \mathcal{I} – множества индексов выражений, описывающих ограничения в виде равенств или неравенств.

Метод SLSQP

`scipy.optimize.minimize(method='SLSQP')`

Основные параметры:

`fun: callable`

Объектная функция для минимизации.

`fun(x, *args) -> float`

где `x` – одномерный массив размера `(n,)`; `args` – кортеж параметров, необходимых для вызова функции.

`x0: ndarray, shape (n,)`

Начальное приближение. Массив элементов размера `(n,)`, где `n` – количество независимых переменных.

`args: tuple, optional`

Дополнительные аргументы, передаваемые объектной функции и ее производным (функциям `fun`, `jac` и `hess`)

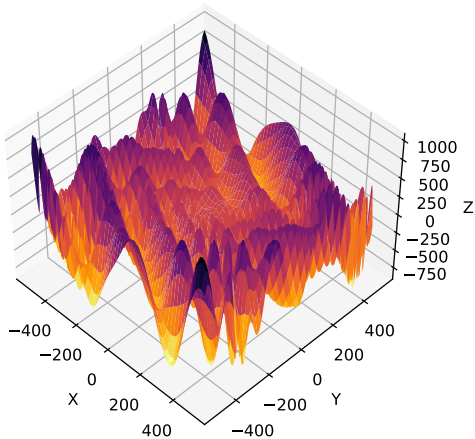
`bounds: sequence, optional`

Предельные значения для переменных.

`constraints: {Constraint, dict} or List of {Constraint, dict}, optional`

Описание функций, накладывающих ограничения на решение.

Пример



Рассмотрим следующую функцию:

$$z = (-y + 47) \cdot \sin \sqrt{\left| \frac{x}{2} + y + 47 \right|} - x \cdot \sin \sqrt{\left| \frac{x}{2} - y - 47 \right|}$$

Найдем минимум данной функции при значениях $x \in [400, 500]$; $y \in [400, 500]$ и при условии, что $x = y$.

Пример

```
1 import numpy as np
2 import scipy.optimize as opt
3
4
5 def func(x):
6     return (
7         (-x[1] + 47) * np.sin(np.sqrt(abs(x[0] / 2 + x[1] + 47)))
8         - x[0] * np.sin(np.sqrt(abs(x[0] - x[1] - 47)))
9     )
10
11
12 x = np.arange(-512, 513)
13 y = np.arange(-512, 513)
14 xgrid, ygrid = np.meshgrid(x, y)
15 xy = np.stack([xgrid, ygrid])
16
17 bounds = (400, 500), (400, 500)
18 constr = {
19     'type': 'eq',
20     'fun': lambda x: x[0] - x[1]
21 }
```

Пример

```
22 |
23 |
24 | res = opt.minimize(
25 |     eggholder,
26 |     x0=(450, 450),
27 |     bounds=bounds,
28 |     constraints=constr,
29 |     method='SLSQP'
30 | )
31 | print(res)
32 |
    fun: -641.3734202306166
    jac: array([ 27.49101257, -27.49103546])
message: 'Optimization terminated successfully'
   nfev: 12
    nit: 4
   njev: 4
status: 0
success: True
      x: array([448.93802864, 448.93802864])
```