



ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ



СИСТЕМНЫЙ АНАЛИЗ ПРОЦЕССОВ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

ПРОЦЕСС КОМПАУНДИРОВАНИЯ ТОВАРНЫХ БЕНЗИНОВ

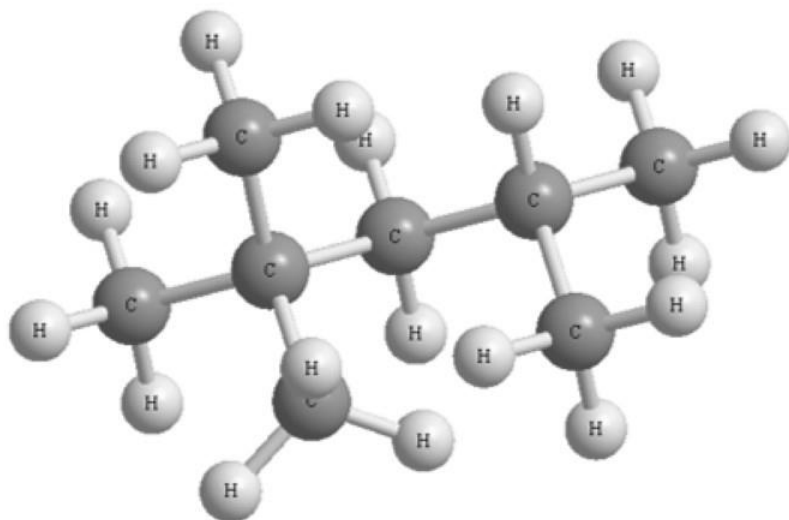
Выполнил доцент ОХИ ИШПР ТПУ, к.т.н.
Чузлов Вячеслав Алексеевич

2020

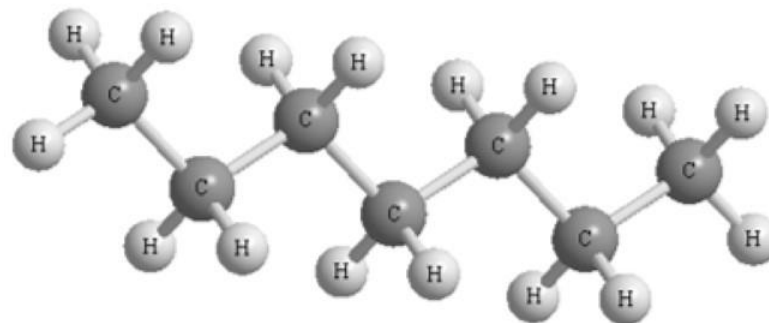


Октановое число – это условная величина, характеризующая детонационную стойкость и численно равная процентному содержанию изооктана в эталонной смеси с н-гептаном, которая по детонационной стойкости эквивалентна испытываемому бензину в условиях стандартного одноцилиндрового двигателя.

Эталонные углеводороды:



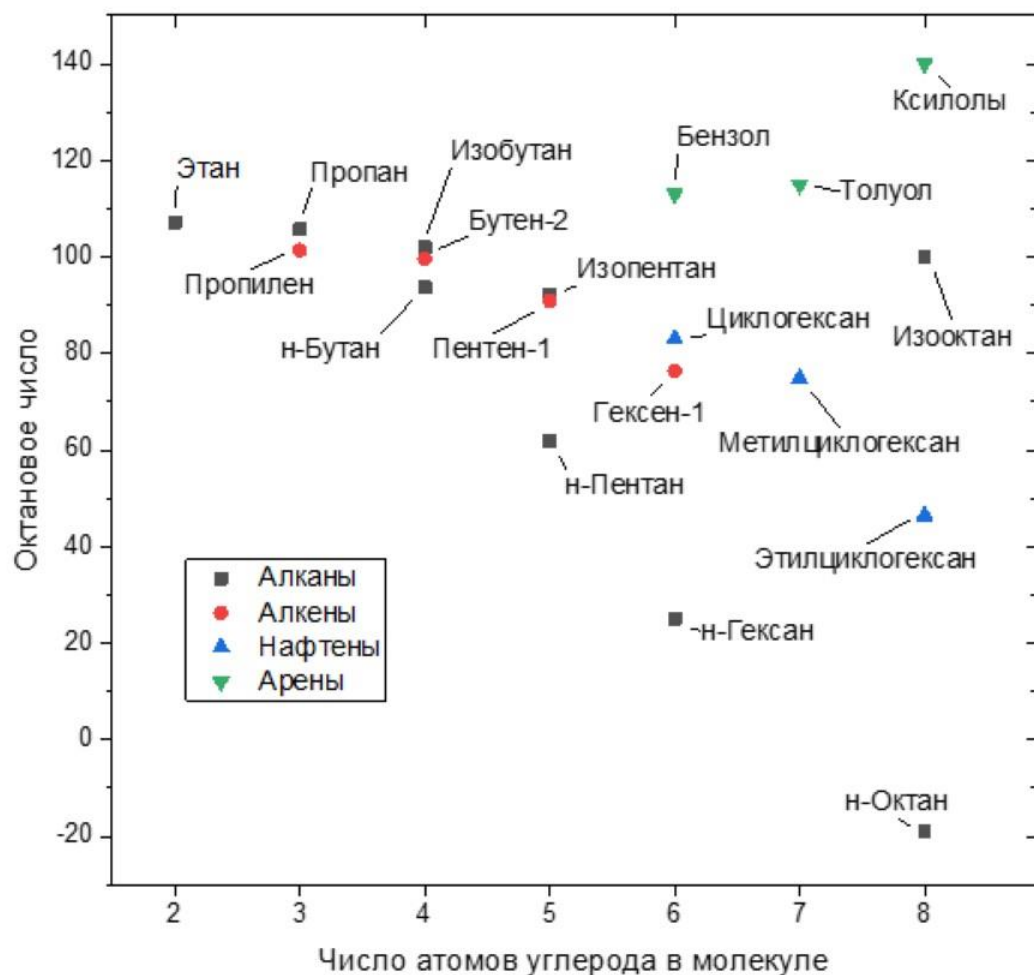
2,2,4-триметилпентан (изооктан)
ОЧ = 100



н-гептан
ОЧ = 0



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 ароматов повышается с увеличением числа углеродных атомов.



Углеводород	ОЧ	Поток 1	Поток 2	Поток 3	Поток 4	Поток 5	Поток 6
Пропан	100,00	1,75	0,00	0,71	0,00	0,09	0,01
Бутан	93,60	5,58	0,19	2,51	0,00	1,32	0,51
Пентан	61,70	20,46	2,34	20,97	8,98	5,20	1,38
Н-гексан	24,80	7,30	20,78	11,39	5,37	3,35	1,77
Н-гептан	0,00	0,01	4,52	2,89	3,06	2,73	2,89
Н-октан	-19,00	0,00	0,11	1,08	7,68	0,63	0,44
Н-нонан	-39,00	0,00	0,00	0,20	0,00	0,21	0,29
изобутан	101,00	0,76	0,04	0,56	6,64	0,52	0,14
2,2-диметилбутан	92,00	0,00	3,55	2,26	0,94	0,54	0,21
2,3-диметилбутан	101,70	2,81	3,68	2,48	1,27	0,52	0,30
2,2,3-триметилбутан	105,80	2,48	0,08	0,06	0,05	0,08	0,05
Изопентан	92,30	0,00	2,88	21,65	15,94	6,63	3,98
2-метилпентан	73,40	21,81	22,48	13,92	0,00	3,57	1,66
3-метилпентан	74,50	14,26	16,65	9,87	5,08	2,48	1,29
3,3-деметилпентан	80,80	10,24	0,56	0,39	0,38	0,43	0,33
2,3-деметилпентан	91,10	0,00	2,00	1,23	1,35	1,13	1,04
2,4-деметилпентан	83,10	0,20	0,94	0,56	0,56	0,49	0,44
2,2,3-триметилпентан	38,00	0,01	0,00	0,00	0,13	0,00	0,00
2-метил, 3-этилпентан	87,30	0,00	0,00	0,00	3,26	0,00	0,00
2,3,4-триметилпентан	101,30	0,00	0,00	0,03	0,08	0,02	0,00
2-метилгексан	42,40	0,00	5,13	3,11	3,49	2,91	2,78
3-метилгексан	52,00	0,04	5,31	3,82	4,25	3,58	3,46
2,5-диметилгексан	55,50	0,04	0,04	0,31	2,37	0,18	0,12
2,4-диметилгексан	65,20	0,00	0,07	0,00	2,90	0,17	0,12
2,3-диметилгексан	71,30	0,00	0,05	0,00	0,25	0,21	0,15
3,4-диметилгексан	76,30	0,00	0,03	0,00	1,37	0,10	0,07
2,3,5-ТМГ	90,00	0,00	0,00	0,00	0,08	0,02	0,04
2,2,3-триметилгексан	92,00	0,00	0,00	0,00	0,00	0,00	0,00
2-метил,4-этилгексан	30,00	0,00	0,00	0,00	0,00	0,02	0,03



Углеводород	ОЧ	Поток 1	Поток 2	Поток 3	Поток 4	Поток 5	Поток 6
2-метилгептан	21,70	0,00	0,14	0,00	0,00	0,69	0,47
4-метилгептан	26,70	0,00	0,07	0,00	4,38	0,34	0,24
3-метилгептан	26,80	0,00	0,22	0,00	3,41	1,04	0,71
2,2-диметилгептан	66,00	0,00	0,00	0,00	0,00	0,04	0,05
2,4-диметилгептан	62,00	0,00	0,00	0,00	0,32	0,07	0,10
2,6-диметилгептан	33,00	0,00	0,00	0,00	0,00	0,05	0,09
2,5-диметилгептан	50,00	0,00	0,00	0,00	0,08	0,12	0,19
3,3-диметилгептан	33,70	0,00	0,00	0,00	0,08	0,00	0,00
изононаны	40,00	0,00	0,02	0,00	0,00	0,59	2,36
непредельные	0,00	0,10	0,00	0,00	0,40	0,16	0,30
циклопентан	87,00	3,34	0,44	0,00	0,13	0,62	0,08
циклогексан	83,00	0,72	0,46	0,00	0,00	0,12	0,05
Метилциклопентан	91,30	6,19	0,08	0,00	0,83	1,87	0,81
1,3-диметилциклопентан (цис)	40,00	0,00	0,14	0,00	0,19	0,09	0,10
1,3-диметилциклопентан (транс)	40,00	0,00	0,75	0,00	0,62	0,44	0,44
1,2-диметилциклопентан (транс)	40,00	0,00	0,18	0,00	0,27	0,11	0,06
1,2-диметилциклопентан (цис)	40,00	0,00	0,10	0,00	1,37	0,14	0,10
1,1,3-триметилциклопентан	38,00	0,00	0,60	0,00	0,19	0,07	0,02
1-2-4-триметилциклопентан	38,00	0,00	0,00	0,00	0,29	0,08	0,02
1,2,3-триметилциклопентан	38,00	0,00	0,00	0,00	0,21	0,06	0,02
1,1,2-триметилциклопентан	38,80	0,00	0,00	0,00	0,11	0,02	0,01
этилциклопентан	67,20	0,00	0,09	0,00	1,23	0,18	0,14
Метилциклогексан	74,80	0,00	0,00	0,00	0,25	0,11	0,09
1,2-диметилциклогексан (транс)	80,90	0,00	0,03	0,00	0,00	0,01	0,00
1,3-диметилциклогексан (транс)	66,90	0,00	0,00	0,00	0,00	0,02	0,00
Нафтеновые до С8	50,00	0,00	0,04	0,00	1,20	0,20	0,08
Нафтеновые до С9	45,00	0,00	0,00	0,00	0,00	0,00	0,01
бензол	115,00	1,79	0,05	0,00	8,96	3,69	3,00
толуол	114,00	0,05	0,36	0,00	0,00	22,13	18,30
п-ксилол	120,00	0,01	0,06	0,00	0,00	1,53	2,38
м-ксилол	120,00	0,03	0,13	0,00	0,00	3,68	5,68
о-ксилол	120,00	0,02	4,39	0,00	0,00	2,45	2,45
ЭЦГ+этилбензол	114,00	0,00	0,03	0,00	0,00	2,37	2,97
Тяжелые С9+	110,00	0,00	0,19	0,00	0,00	19,11	20,96
МТБЭ	130,00	0,00	0,00	0,00	0,00	0,67	14,22



Необходимо определить долю каждого потока в смеси при заданном октановом числе смешения.

Исходные данные:

1. Углеводородные составы потоков, направляемых на смешение.
2. ОЧ индивидуальных углеводородов.

Требуемый результат:

1. Доля каждого потока, направляемого на смешение (в %).
2. ОЧ смесового потока.



Для решения поставленной задачи необходимо разбить ее на более простые подзадачи и последовательно решить их:

1. Расчет ОЧ каждого потока:

$$\text{ОЧ}_{\text{потока}} = \sum_{i=1}^k C_i \cdot \text{ОЧ}_i$$

где $\text{ОЧ}_{\text{потока}}$ – октановое число потока смешения;
 C_i – доля i -го углеводорода в составе потока смешения;
 ОЧ_i – октановое число i -го углеводорода;
 k – количество компонентов.

2. Расчет ОЧ смеси потоков:

$$\text{ОЧ}_{\text{см}} = \sum_{i=1}^n x_i \cdot \text{ОЧ}_{\text{потока}, i}$$

где $\text{ОЧ}_{\text{см}}$ – октановое число смеси;
 x_i – доля i -го потока смешения;
 $\text{ОЧ}_{\text{потока}}$ – октановое число потока смешения;
 n – количество потоков смешения.

3. Расчет долей каждого из потоков, при которых достигается заданное ОЧ смешения.



1. Поиск потока с ОЧ, максимально близким к заданному.
 - Доля этого потока принимается за 1, доли остальных потоков за ноль (обозначим его *поток N*).

2. Выполняется проверка условия:

$|\text{Расчетное ОЧ} - \text{Заданное ОЧ}| \leq \text{eps}$ (заранее выбранная точность)

- Если условие выполнилось, то расчет прекращается, если нет, то переходим к следующему этапу.

3. Выполняется проверка условия:

$\text{Расчетное ОЧ} > \text{Заданное ОЧ}$

Условие выполняется

- Доля *потока N* уменьшается;
- Если ОЧ потока $<$ ОЧ *потока N*, его доля увеличивается, если $>$, то уменьшается.

Условие не выполняется

- Доля *потока N* увеличивается;
- Если ОЧ потока $>$ ОЧ *потока N*, его доля увеличивается, если $<$, то уменьшается.

Пока не выполнится условие: $|\text{Расчетное ОЧ} - \text{Заданное ОЧ}| \leq \text{eps}$





Расчет ОЧ потоков смешения:

Необходимо описать следующий тип:

type

TArrOfDouble = **array of double**;

function get_flow_RON (comp_count:integer; RON: TArrOfDouble;
flow_composition: TArrOfDouble): **double**;

var

i: **integer**;

s: **double**;

begin

s := 0;

for i := 0 **to** comp_count-1 **do**

s := s + RON[i] * flow_composition[i];

Result := s;

end;



Расчет ОЧ смеси:

```
function get_mix RON(flow_count: integer; RON: TArrOfDouble;  
    mix_composition: TArrOfDouble): double;  
var  
    i: integer;  
    s: double;  
begin  
    s := 0;  
    for i := 0 to flow_count-1 do  
        s := s + RON[i] * mix_composition[i];  
    Result := s;  
end;
```




Функция поиска потока, ОЧ которого максимально близко к заданному ОЧ:

```
function get_min(flow_count: integer; flows_RON: TArrOfDouble;  
    treb_RON: double): integer;  
var  
    i: integer;  
    d: double;  
begin  
    d := abs(flows_RON[0] - treb_RON);  
    Result := 0;  
    for i := 1 to flow_count-1 do  
        if abs(flows_RON[i] - treb_RON) < d then  
            begin  
                d := abs(flows_RON[i] - treb_RON);  
                Result := i;  
            end;  
    end;
```



Функция нормировки состава смеси:

```
function normalization(flow_count: integer;  
    mix_composition: TArrayOfDouble): TArrayOfDouble;  
var  
    i: integer;  
    s: double;  
begin  
    SetLength(Result, flow_count);  
    s := 0;  
    for i := 0 to flow_count-1 do  
        s := s + abs(mix_composition[i]);  
    for i := 0 to flow_count-1 do  
        Result[i] := abs(mix_composition[i]) / s;  
end;
```



Функция расчета состава ОЧ смеси:

```
function get_mix_composition(eps: double; h: double; flow_count: integer;  
    min: integer; flows RON: TArrOfDouble; trebRON: double): TArrOfDouble;  
var  
    i, n: integer;  
    RONc: double;  
begin  
    SetLength(Result, flow_count);  
    n := 1;  
    Result[min] := 1;  
    for i := 0 to flow_count-1 do  
        if i <> min then  
            Result[i] := 0;  
    RONc := get_mix RON(flow_count, flows RON, Result);  
    if abs(RONc - trebRON) > eps then  
        repeat  
            if flows RON[min] > trebRON then  
                begin  
                    Result[min] := Result[min] - h;  
                    for i := 0 to flow_count-1 do  
                        if i <> min then  
                            if flows RON[i] < flows RON[min] then  
                                Result[i] := Result[i] + h * abs(flows RON[i] - trebRON)  
                            else  
                                Result[i] := Result[i] - h * abs(flows RON[i] - trebRON)  
                        end
```




продолжение:

```
else
  begin
    Result[min] := Result[min] + h;
    for i := 0 to flow_count-1 do
      if i <> min then
        if flows RON[i] < flows RON[min] then
          Result[i] := Result[i] - h * abs(flows RON[i] - trebRON)
        else
          Result[i] := Result[i] + h * abs(flows RON[i] - trebRON)
        end;
    Result := normalization(flow_count, Result);
    RONc := get_mix RON(flow_count, flows RON, Result);
    if n >= 1e5 then
      begin
        writeln('Выполнено 100 000 итераций, но решение не было найдено!');
        break;
      end;
    n := n + 1;
  until abs(RONc - trebRON) <= eps;
end;
```



«Основная» процедура:

Необходимо описать следующий тип:

```
type
  TArrOfArrOfDouble = array of array of double;

procedure blending(comp_count: integer; flow_count: integer;
  comp_RON: TArrOfDouble; flow_composition: TArrOfArrOfDouble;
  treb_RON: double; eps: double; h: double;
  var mix_composition: TArrOfDouble; var RONc: double);
var
  min: integer;
  i, j: integer;
  flows_RON: TArrOfDouble;
  flow_comp: TArrOfDouble;

begin
  SetLength(comp_RON, comp_count);
  SetLength(flow_comp, comp_count);
  SetLength(flows_RON, flow_count);
  SetLength(mix_composition, flow_count);
  SetLength(flows_RON, flow_count);
```



продолжение:

```
for i := 0 to flow_count-1 do
  begin
    for j := 0 to comp_count-1 do
      flow_comp[j] := flow_composition[j, i] / 100;
      flows_RON[i] := get_flow_RON (comp_count, comp_RON, flow_comp);
    end;

    min := get_min(flow_count, flows_RON, treb_RON);

    mix_composition := get_mix_composition(eps, h, flow_count, min,
      flows_RON, treb_RON);
    RONc := get_mix_RON(flow_count, flows_RON, mix_composition);
  end;
```




Блок описаний:

```
program Main_blending;
```

```
uses
```

```
  UBlending; // Имя модуля, содержащего описание расчетов
```

```
const
```

```
  flow_count = 6;
```

```
  comp_count = 64;
```

```
var
```

```
  comp RON: TArrayOfDouble;
```

```
  flow_composition: TArrayOfArrOfDouble;
```

```
  mix_composition: TArrayOfDouble;
```

```
  RONc: double;
```

```
  treb RON: double;
```

```
  i: integer;
```



Процедура для считывания исходных данных:

```
procedure get_data(var RON: TArray;
                   flow_composition: TArrayOfArrOfDouble);

var
    f1, f2: text;
    i, j: integer;
begin
    assign(f1, 'RON.txt');
    assign(f2, 'flow_comp.txt');
    reset(f1);
    reset(f2);
    for i := 0 to comp_count-1 do
        begin
            readln(f1, RON[i]);
            for j := 0 to flow_count-1 do
                read(f2, flow_composition[i, j]);
            readln(f2);
        end;
    close(f1);
    close(f2);
end;
```



Процедура для вывода результатов:

```
procedure get_result(mix_composition: TArrOfDouble; RONc: double);  
var  
    f: text;  
    i: integer;  
begin  
    assign(f, 'result.txt');  
    rewrite(f);  
    for i := 0 to flow_count-1 do  
        writeln(f, 'Доля потока ', i+1, ' = ',  
                mix_composition[i] * 100:8:2);  
    writeln(f);  
    writeln(f, 'Октановое число смешения = ', RONc:8:2);  
    close(f);  
end;
```




Код основной программы:

Begin

```
SetLength(comp_RON, comp_count);
SetLength(flow_composition, comp_count);
for i := 0 to comp_count-1 do
    SetLength(flow_composition[i], flow_count);
SetLength(mix_composition, flow_count);

write('Введите требуемое ОЧ: ');
readln(treb_RON);

get_data(comp_RON, flow_composition);
blending(comp_count, flow_count, comp_RON, flow_composition,
        treb_RON, 5e-2, 5e-6, mix_composition, RONc);
get_result(mix_composition, RONc);
```

end.