**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение

высшего образования



**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа – Инженерная школа ядерных технологий

Направление подготовки – 14.03.02 «Ядерные физика и технологии»

Отделение школы (НОЦ) – Отделение ядерно-топливного цикла

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

по дисциплине

“Междисциплинарный проект”

|  |
| --- |
| **Тема работы** |
| Расчет изменения изотопной концентрации по каскаду в стационарном режиме |

Выполнил

|  |  |  |  |
| --- | --- | --- | --- |
| **Группа** | **ФИО** | **Подпись** | **Дата** |
| 0А8Д | Кузьменко Анна Сергеевна |  |  |

Проверил

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Должность** | **ФИО** | **Ученая степень, звание** | **Подпись** | **Дата** |
| Профессор ОЯТЦ | Видяев Д.Г. | д.т.н. |  |  |

Томск – 2021 г.

**3. ИСХОДНЫЕ ДАННЫЕ**

Концентрация отбора ;

Концентрация питания ;

Концентрация отвала ;

Температура ;

Поток отбора ;

Доля сокращения потока на одной ТТ ;

Количество ТТ в одной колонне .

**4. РАСЧЕТЫ**

Расчеты проведены на языке C++, листинг главного файла приведен в приложении А.

Поток потока переведен в моль/ч:



Рассчитано значение коэффициента разделения по формуле (1):

 (1)



Рассчитано значение коэффициента обогащения по формуле (2):

 (2)



По формуле (3) определено число теоретических тарелок:

, (3)

где , 







Количество колонн:



Рассчитано изменение концентрации целевого изотопа в безотборном режиме по колоннам каскада с помощью дифференциального уравнения (4):

 (4)







, , 











Выразим концентрацию *c*1:









 (5)













Определена величина начального потока при работе каскада с заданным отбором по формуле (6):

, (6)

где ** – коэффициент для сшивки каскада по концентрации отвала.



Сокращение потока *L* по колоннам каскада:

 (7)



 (8)

Рассчитан средний поток для каждой колонны:









 (9)













Рассчитано изменение концентрации целевого изотопа по колоннам по дифференциальному уравнению (10):

 (10)







Разложим знаменатель на множители:





, ,













Выразим концентрацию *c*2:













, (11)

где 

























В таблицах 1 и 2 приведены результаты расчета изменения концентрации по колоннам каскада в режимах без отбора и с отбором.

Таблица 1 – Изменение концентрации целевого изотопа по колоннам каскада для безотборного режима

|  |  |
| --- | --- |
| *n*кол. | *c*1 |
| 0 | 0,900000000000000 |
| 1 | 0,964015441661586 |
| 2 | 0,987615040886966 |
| 3 | 0,995804769961711 |
| 4 | 0,998586675005177 |
| 5 | 0,999524747379797 |
| 6 | 0,999840288460193 |

Таблица 2 – Изменение концентрации целевого изотопа по колоннам каскада для режима с отбором

|  |  |
| --- | --- |
| *n*кол. | *c*2 |
| 0 | 0,900019491782748 |
| 1 | 0,940607497653780 |
| 2 | 0,963256992695849 |
| 3 | 0,976195345393696 |
| 4 | 0,984184558984906 |
| 5 | 0,989868268716236 |
| 6 | 0,995000000000000 |

График изменения концентрации в режиме с отбором и без представлен на рисунке 1.

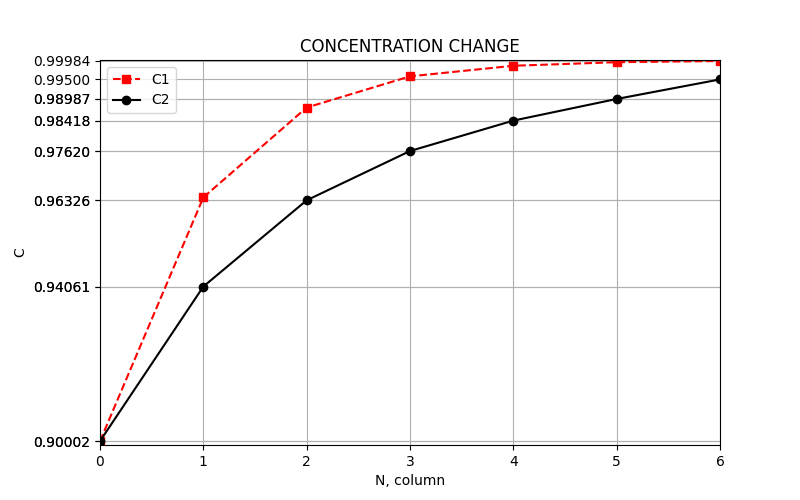


Рисунок 1 – Изменение концентрации по колоннам каскада

*c*1 – без отбора; *c*2 – с отбором

Определены величины потоков отвала и питания из уравнения материального баланса (12):

 (11)

В данной системе уравнений неизвестными являются потоки питания (*F*) и отвала (*W*).







 (12)

 (13)





**ВЫВОДЫ**

Проведен расчет изменения концентрации целевого изотопа по колоннам каскада в режимах без отбора и с отбором при заданных параметрах его работы. Построен график изменения концентрации целевого изотопа в обоих режимах работы каскада. Рассчитано, что для обеспечения целевой концентрации на выходе из каскада колонн в безотборном режиме необходимо минимально шесть колонн. В режиме без отбора концентрация по целевому изотопу на выходе из каскада колонн 0,99984. Определено, что необходимо увеличить минимальный начальный поток в 1,576 раз для сшивки каскада по концентрации отвала. Изменение концентрации по колоннам каскада для режима без отбора превышает изменение для режима с отбором в среднем на 1,39 %.

**ПРИЛОЖЕНИЕ А**Листинг main.cpp

#include <iostream>

#include <fstream>

#include <cmath>

#include "matplotlibcpp.h"

#include <vector>

namespace plt = matplotlibcpp;

const double T = 15; // Celsius

const double CW = 0.9;

const double CF = 0.925;

const double CP = 0.995;

const double r = 0.5; // percents

const double N = 20;

double P = 150; // kg per year

int main()

{

std::cout << "Calculations start" << std::endl;

std::cout << std::endl;

const double mu = (CP\*7.0+(1-CP)\*6.0)\*pow(10, -3);

P = P/mu/(365.0\*24.0); // mol per hour

double e = 4755.0/pow((T+273), 2) - 0.803 / (T+273);

double nE = 2.0/e\*log(CP\*(1.0-CF)/(CF\*(1.0-CP))); nE = ceil(nE);

double nR = 2.0/e\*log(CF\*(1.0-CW)/(CW\*(1.0-CF))); nR = ceil(nR);

double nB = nE + nR;

const int Nc = ceil(nB/N);

double dN[Nc+1];

for (int i = 0; i <= Nc; i++)

{

dN[i] = i;

}

double dC1[Nc+1]; dC1[0] = CW;

double dC2[Nc+1]; dC2[Nc+1] = CP;

// without P

for (int i = 1; i <= Nc; i++)

{

dC1[i] = (CW/(1-CW)\*exp(e\*i\*N)) / (1.0 + CW/(1-CW)\*exp(e\*i\*N));

}

// with P

double Lb[Nc];

double Le[Nc];

double L[Nc];

double Lin = 2.0\*P\*(CP-CF)/(e\*CF\*(1.0-CF));

for (int i = 0; i <= Nc; i++)

{

Lb[i] = Lin \* pow((1.0-r/100.0), N\*i);

Le[i] = Lin \* pow((1.0-r/100.0), N\*(i+1));

L[i] = (Lb[i] + Le[i]) / 2.0;

}

double x1, x2;

for (int i = 0; i <= Nc; i++)

{

x1 = 0.5 \* (1.0 + P/(e\*L[i])) + sqrt(0.25\*(pow(1.0 + P/(e\*L[i]), 2.0)) - P \* CP / (e\*L[i]));

x2 = 0.5 \* (1.0 + P/(e\*L[i])) - sqrt(0.25\*(pow(1.0 + P/(e\*L[i]), 2.0)) - P \* CP / (e\*L[i]));

dC2[i] = (x2 \* (x1 - CP) / (CP - x2) \* exp(e\*(Nc-i)\*N\*(x1 - x2)) + x1) / (1 + (x1 - CP) / (CP - x2) \* exp(e\*(Nc-i)\*N\*(x1 - x2)));

}

// changing Lin

double dh = 0.0001;

int k = 0;

while ((abs(dC2[0]-CW) > 0.00009) || (dC2[0] < CW))

{

Lin = 2.0\*P\*(CP-CF)/(e\*CF\*(1.0-CF));

Lin = Lin \* (1-k\*dh);

for (int i = 0; i <= Nc; i++)

{

Lb[i] = Lin \* pow((1.0-r/100.0), N\*i);

Le[i] = Lin \* pow((1.0-r/100.0), N\*(i+1));

L[i] = (Lb[i] + Le[i]) / 2.0;

}

for (int i = 0; i <= Nc; i++)

{

x1 = 0.5 \* (1.0 + P/(e\*L[i])) + sqrt(0.25\*(pow(1.0 + P/(e\*L[i]), 2.0)) - P \* CP / (e\*L[i]));

x2 = 0.5 \* (1.0 + P/(e\*L[i])) - sqrt(0.25\*(pow(1.0 + P/(e\*L[i]), 2.0)) - P \* CP / (e\*L[i]));

dC2[i] = (x2 \* (x1 - CP) / (CP - x2) \* exp(e\*(Nc-i)\*N\*(x1 - x2)) + x1) / (1 + (x1 - CP) / (CP - x2) \* exp(e\*(Nc-i)\*N\*(x1 - x2)));

}

k++;

if (1 - k\*dh < 0)

{

std::cout << "Error, too much increasing Lin: 1-k\*dh < 0" << std::endl;

break;

}

}

std::cout << Nc <<std::endl;

// output into terminal

std::cout<< "Lin is increased in " << 1 - k\*dh << std::endl;

std::cout << std::endl;

std::cout << "N " << "L, mol/h " << "C1 " << "C2" << std::endl;

std::cout << dN[0] << " " << "- " << dC1[0] << " " << dC2[0] << std::endl;

for (int i = 1; i <= Nc; i++)

{

std::cout << dN[i] << " " << L[i-1] << " " << dC1[i] << " " << dC2[i] << std::endl;

}

// output into file

std::ofstream out("output.txt", std::ios\_base::out | std::ios\_base::trunc);

if (!out.is\_open())

{

std::cout << "Error, file is not able to open" << std::endl;

}

else

{

out<< "Lin is increased in " << 1 - k\*dh << std::endl;

out << std::endl;

out << "N " << "L, mol/h " << "C1 " << "C2" << std::endl;

out << dN[0] << " " << "- " << dC1[0] << " " << dC2[0] << std::endl;

for (int i = 1; i <= Nc; i++)

{

out << dN[i] << " " << L[i-1] << " " << dC1[i] << " " << dC2[i] << std::endl;

}

}

std::cout << std::endl;

double W = P \* (dC2[Nc] - CF) / (CF - dC2[0]);

double F = P + W;

std::cout << "W is " << W << std::endl;

std::cout << "F is " << F << std::endl;

if (out.is\_open())

{

out << std::endl;

out << "W is " << W << std::endl;

out << "F is " << F << std::endl;

out.close();

}

std::cout << std::endl;

std::cout << "Calculations end" << std::endl;

// plot a graph using matplotlibcpp

std::vector <double> y1(Nc+1);

std::vector <double> y2(Nc+1);

std::vector <double> x(Nc+1);

std::vector <double> y(2\*Nc+2);

for (int i = 0; i <= Nc; i++)

{

y1.at(i) = dC1[i];

y2.at(i) = dC2[i];

x.at(i) = i;

if (abs(dC1[i] - dC2[i]) > 0.03)

{

y.at(2\*i) = dC1[i];

y.at(2\*i+1) = dC2[i];

}

else

{

y.at(2\*i) = dC2[i];

y.at(2\*i+1) = dC2[i];

}

}

y.at(2\*Nc+1) = dC1[Nc];

plt::figure\_size(800,500);

plt::plot(x,y1,{{"label", "C1"}, {"linestyle", "--"}, {"marker", "s"}, {"color", "r"}});

plt::xticks(x); plt::yticks(y);

plt::plot(x,y2,{{"label", "C2"}, {"linestyle", "-"}, {"marker", "o"}, {"color", "k"}});

plt::ylim(CW-0.001, 1.0001); plt::xlim(0,Nc);

plt::grid(1);

plt::title("CONCENTRATION CHANGE",{{"color", "k"}});

plt::xlabel("N, column"); plt::ylabel("C");

plt::legend();

plt::save("result.tiff"); plt::show();

return 0;

}