A1) The result of the simulation with N=1, γ_0=0.05 is as below.

[Input Parameter]

number of reactor	: N =	1
initial concentration of monomer	: γ0 =	0.050
temperature of coolant	: T2 =	-15.000[°C]

[Constances]

product flow rate	: Gp =	63.000[ton/day]
total conversion rate	: ζ =	0.600
residence time of one reactor	: τ =	18000.000[s]
mixture density	: ρ =	850.000[kg/m^3]
reactor aspect rate	: α =	1.300
polymerization heat	: ⊿H =	72.800[kJ/mol]
mixture temperature	: T1 =	50.000[°C]
thermal conductivity of toluene	: λ =	0.128[W/K*m]
specific heat of toluene	: Cp =	1.680[J/K*g]
molecular weight of toluene	: M =	54.000[g/mol]
monomer length	: ML =	50
reaction rate const	: K =	0.000[1/s]

[Reactors Overview]

feed rate	: F =	0.029[m^3/hour]
volume	: V =	514.706[m^3]
diameter	: D =	7.959[m]
height	: H =	10.346[m]
side area	: S =	258.689[m^2]

[Reactor 1]

concentration of Butadien of out	: C_1	=	0.020
product heat of polymerization	: Qp	=	983024.691[W]
viscosity	: μ_1	=	0.616[Pa*s]
Nusselt number	: Nu	=	3635.013
Prandtl number	: Pr	=	8086.036
Reynolds number	: Re	=	6893.441
Power number	: Np	=	1.229
revolution number	: n_1	=	0.316[rps]
power consumption	: P_1	=	32745.985[W]

[Total]

total power consumption	: P	=	32745.985[W]

So, the size of the reactor and the total power consumption are as below.

Size of the reactor:

Volume : **514.706[m^3]**

Diameter: **7.959[m]**

Height : **10.346[m]**

Total power consumption:

32745.985[W]

A2) The result of the simulation with N=3, γ_0=0.04 is as below.

[Input Parameter]

number of reactor	: N =	3
initial concentration of monomer	: γ0 =	0.040
temperature of coolant	: T2 =	-15.000[°C]

[Constances]

product flow rate	: Gp =	63.000[ton/day]
total conversion rate	: ζ =	0.600
residence time of one reactor	: τ =	4286.506[s]
mixture density	: ρ =	850.000[kg/m^3]
reactor aspect rate	: α =	1.300
polymerization heat	: ⊿H =	72.800[kJ/mol]
mixture temperature	: T1 =	50.000[°C]
thermal conductivity of toluene	: λ =	0.128[W/K*m]
specific heat of toluene	: Cp =	1.680[J/K*g]
molecular weight of toluene	: M =	54.000[g/mol]
monomer length	: ML =	50
reaction rate const	: K =	0.000[1/s]

[Reactors Overview]

feed rate	: F =	0.036[m^3/s]
volume	: V =	153.215[m^3]
diameter	: D =	5.314[m]
height	: H =	6.908[m]
side area	: S =	115.329[m^2]

```
[Reactor 1]
  concentration of Butadien of out : C_1  =      0.029
  product heat of polymerization   : Qp   = 431209.843[W]
  viscosity                        : μ_1  =      0.064[Pa*s]
  Nusselt number                   : Nu   =   2388.085
  Prandtl number                   : Pr   =    835.304
  Reynolds number                  : Re   =  11420.827
  Power number                     : Np   =      1.067
  revolution number                : n_1  =      0.121[rps]
  power consumption                : P_1  =    213.450[W]

-----------------------------------------------------------------

[Reactor 2]
  concentration of Butadien of out : C_2  =      0.022
  product heat of polymerization   : Qp   = 317718.129[W]
  viscosity                        : μ_2  =      0.253[Pa*s]
  Nusselt number                   : Nu   =   1759.556
  Prandtl number                   : Pr   =   3320.648
  Reynolds number                  : Re   =   3622.750
  Power number                     : Np   =      1.472
  revolution number                : n_2  =      0.153[rps]
  power consumption                : P_2  =    590.307[W]

-----------------------------------------------------------------

[Reactor 3]
  concentration of Butadien of out : C_3  =      0.016
  product heat of polymerization   : Qp   = 234096.719[W]
  viscosity                        : μ_3  =      0.499[Pa*s]
  Nusselt number                   : Nu   =   1296.452
  Prandtl number                   : Pr   =   6554.413
  Reynolds number                  : Re   =   1630.846
  Power number                     : Np   =      1.840
  revolution number                : n_3  =      0.136[rps]
  power consumption                : P_3  =    517.832[W]
```

[Total]
  total power consumption   : P =   1321.589[W]

So, the size of the reactor and the total power consumption are as below
        Size of the reactor:
                Volume : **153.215[m^3]**
                Diameter: **5.314[m]**
                Height :   **6.908[m]**
        Total power consumption:
                        1321.589 [W]

A3)Table 1 and figure 1, 2 represent the relation between the number of reactors, monomer concentration in the feed of the reactor train, and the total power consumption.

		Γ0 : Monomer concentration in the feed of the reactor train									
		0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
N : Number of reactors	1	541.1	2469.4	6923.9	15903.3	32746.0	62923.0	115277.2	203945.6	351320.6	592594.1
	2	105.2	480.2	1346.4	3092.4	6367.5	12235.5	22415.9	39657.6	68315.0	115231.1
	3	45.0	205.2	575.4	1321.6	2721.2	5229.0	9579.7	16948.2	29195.4	49245.6
	4	26.2	119.5	335.0	769.4	1584.2	3044.1	5576.8	9866.4	16996.0	28668.2
	5	17.6	80.5	225.7	518.4	1067.3	2051.0	3757.4	6647.5	11451.2	19315.4

Table 1: The total power consumption[w] of each numbers of reactors and each monomer concentrations in the feed of the reactor train.

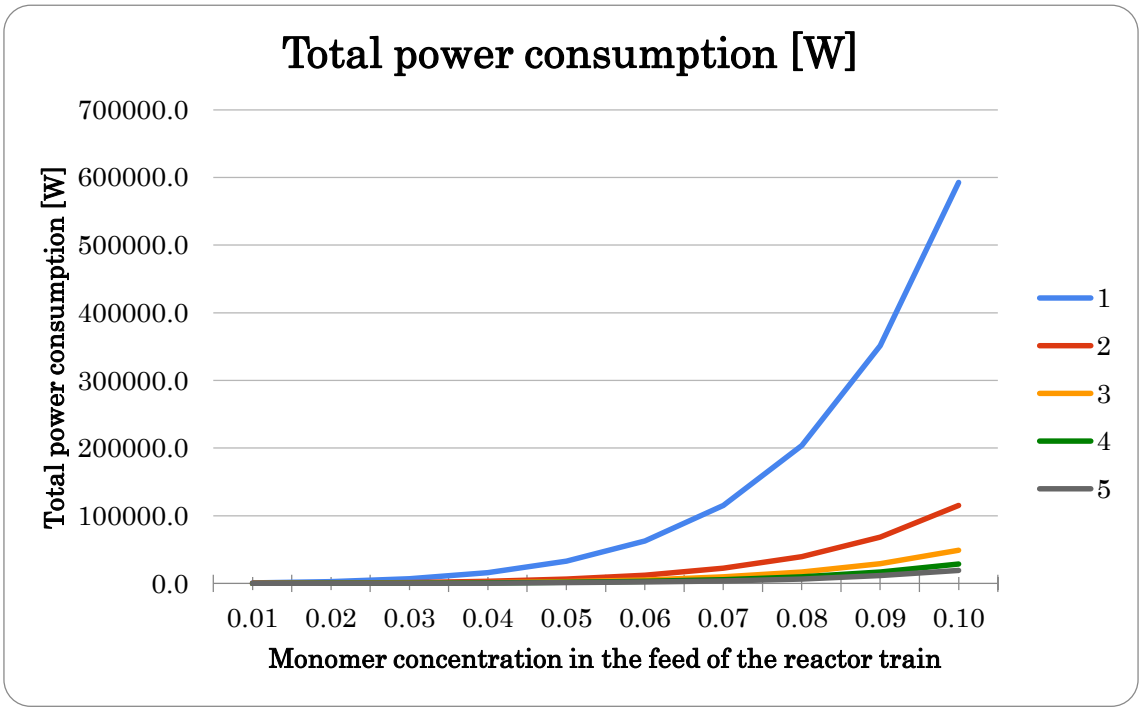

Figure 1: The relation between the total power consumption and the monomer concentration in the feed of the reactor train.

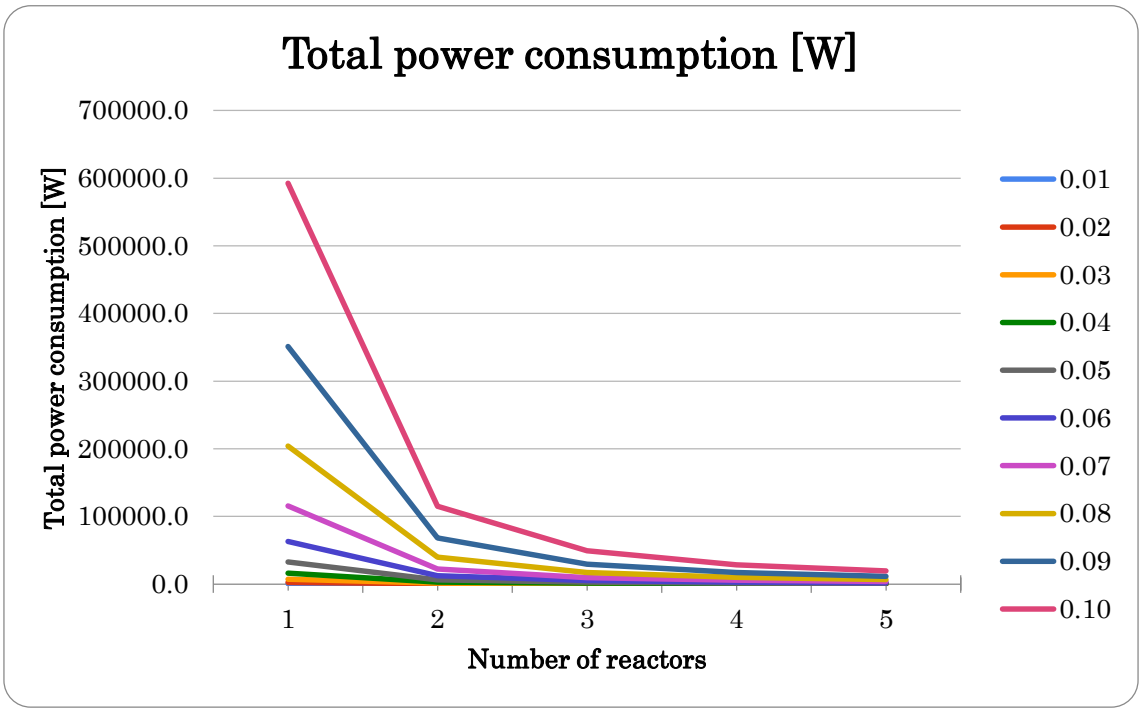

Figure 2: The relation between the total power consumption and the number of reactors.

These results show that increasing the number of reactors is very effective to save the power consumption. With increasing the monomer concentration of the feed, the power consumption is increasing exponentially. Therefore, these results show that using larger reactor and lower concentration mixture is better. However, as larger reactors, as much cost of circulation of the coolant, so we must also consider of that cost.

Program

- Each constant parameters are written with **#define** syntax, and if need, converted unit order in other constant for readability and editability.
- **showUsage** method shows usage of this program.
- When 3 parameter were passed for this program, it calculates total power consumption and show. If parameter is invalid format, only the usage is shown.

```
//
//  main.cpp
//  work
//
//  Created by KikuraYuichiro on 2014/10/22.
//  Copyright (c) 2014年 KikuraYuichiro. All rights reserved.
//

#include <stdio.h>
#include <iostream>
#include <math.h>

#define PRODUCT_FLOW_RATE__             63.0    //  Gp[ton/day]
#define TOTAL_CONVERSION_RATE           0.6     //  ζ
#define RESIDENCE_TIME_IN_ONE_REACTOR__ 5       //  τ [hour]
#define MIXTURE_DENSITY                 850.0   //  ρ [kg/m^3]
#define ASPECT_RATE                     1.3     //  α
#define POLYMERIZATION_HEAT__           72.8    //  ⊿H[kJ/mol]
#define MIXTURE_TEMP                    50.0    //  T1[℃]
#define TOLUENE_THERMAL_CONDUCTIVITY    0.128   //  λ [W/K*m]
#define TOLUENE_SPECIFIC_HEAT__         1.68    //  Cp[J/g*K]
#define TOLUENE_MOLECULAR_WEIGHT__      54.0    //  M [g/mol]
#define ML                              50      //  Monomer Length

//unit-converted parameters
#define PRODUCT_FLOW_RATE
    (PRODUCT_FLOW_RATE__*1.0e3/24/60/60)    //  Gp[kg/s]
#define RESIDENCE_TIME_IN_ONE_REACTOR   (RESIDENCE_TIME_IN_ONE_REACTOR__*60*60
    ) //  τ0[s]
#define POLYMERIZATION_HEAT
    (POLYMERIZATION_HEAT__*1.0e3)           //  ⊿H[J/mol]
#define TOLUENE_SPECIFIC_HEAT
    (TOLUENE_SPECIFIC_HEAT__*1.0e3)         //  Cp[J/kg*K]
#define TOLUENE_MOLECULAR_WEIGHT
    (TOLUENE_MOLECULAR_WEIGHT__*1.0e-3)     //  M [kg/mol]

#define printLine
    printf("\n---------------------------------------------------------------
    \n\n")

float getPowerConsumption(int reactorNumber,
                          double initialConcentration,
                          double coolantTemp,
                          bool flagLog) {

    if (flagLog) {
        printLine;
        printf("[Input Parameter]\n\n");
        printf("  number of reactor               : N  = %6d\n",
            reactorNumber);
        printf("  initial concentration of monomer : γ0 = %10.3lf\n",
            initialConcentration);
        printf("  temperature of coolant          : T2 = %10.3lf[°C]\n",
            coolantTemp);
    }

    //---------------------------------------------------------------
    //  calculate common reactor information.
```

```
    double z = 1.0/(1.0 - TOTAL_CONVERSION_RATE);
    double reactionRateConst = 1.0 / RESIDENCE_TIME_IN_ONE_REACTOR * (z - 1);
    double residenceTime = (pow(z, 1.0/reactorNumber) - 1) / (z - 1) *
        RESIDENCE_TIME_IN_ONE_REACTOR;
    double totalResidenceTime = residenceTime * reactorNumber;

    double feed = PRODUCT_FLOW_RATE / (MIXTURE_DENSITY * initialConcentration
        * TOTAL_CONVERSION_RATE);
    double volume = feed * residenceTime;

    double diameter = cbrt(4.0*volume/ASPECT_RATE/M_PI);
    double height = diameter * ASPECT_RATE;
    double sideArea = M_PI*diameter*height;

    //-------------------------------------------------------------
    //  output common reactor information.
    if (flagLog) {
        printLine;
        printf("[Constances]\n");
        printf("  product flow rate               : Gp = %10.3lf[ton/day]\n",
            PRODUCT_FLOW_RATE__);
        printf("  total conversion rate           : ζ  = %10.3lf\n",
            TOTAL_CONVERSION_RATE);
        printf("  residence time of one reactor   : τ  = %10.3lf[s]\n",
            residenceTime);
        printf("  mixture density                 : ρ  = %10.3lf[kg/m^3]\n",
            MIXTURE_DENSITY);
        printf("  reactor aspect rate             : α  = %10.3lf\n",
            ASPECT_RATE);
        printf("  polymerization heat             : ⊿H = %10.3lf[kJ/mol]\n",
            POLYMERIZATION_HEAT__);
        printf("  mixture temperature             : T1 = %10.3lf[°C]\n",
            MIXTURE_TEMP);

        printf("  thermal conductivity of toluene : λ  = %10.3lf[W/K*m]\n",
            TOLUENE_THERMAL_CONDUCTIVITY);
        printf("  specific heat of toluene        : Cp = %10.3lf[J/K*g]\n",
            TOLUENE_SPECIFIC_HEAT__);
        printf("  molecular weight of toluene     : M  = %10.3lf[g/mol]\n",
            TOLUENE_MOLECULAR_WEIGHT__);
        printf("  monomer length                  : ML = %6d\n",
            ML);
        printf("  reaction rate const             : K  = %10.3lf[1/s]\n",
            reactionRateConst);

        printLine;
        printf("[Reactors Overview]\n");
        printf("  feed rate                       : F  = %10.3f[m^3/s]\n",
            feed);
        printf("  volume                          : V  = %10.3f[m^3]\n",
            volume);
        printf("  diameter                        : D  = %10.3lf[m]\n",
            diameter);
        printf("  height                          : H  = %10.3lf[m]\n",
            height);
        printf("  side area                       : S  = %10.3lf[m^2]\n",
            sideArea);
    }

    //-------------------------------------------------------------
```

```
//  calculate for each reactant value

int i;
double
    concentRation,
    conversionRate,
    productHeat,
    heatTransCof,
    viscosity,
    Nu,
    Pr,
    Re,
    Np,
    revolutionNumber,
    powerConsumption,
    totalPowerConsumption = 0;

for (i = 1; i <= reactorNumber; i++) {
    concentRation = initialConcentration * pow(z, -1.0*i/reactorNumber);

    productHeat = POLYMERIZATION_HEAT * MIXTURE_DENSITY * concentRation /
        TOLUENE_MOLECULAR_WEIGHT * reactionRateConst * volume;

    heatTransCof = productHeat / sideArea / (MIXTURE_TEMP - coolantTemp);

    conversionRate = 1.0 - pow(1.0 - TOTAL_CONVERSION_RATE, 1.0*i/
        reactorNumber);

    viscosity = pow(ML, 1.7) * pow(conversionRate, 2.5) * exp(21 *
        initialConcentration) * 1.0e-3;

    Nu = heatTransCof * diameter / TOLUENE_THERMAL_CONDUCTIVITY;
    Pr = viscosity / TOLUENE_THERMAL_CONDUCTIVITY * TOLUENE_SPECIFIC_HEAT;
    Re = pow(2.0*Nu*pow(Pr, -1.0/3), 3.0/2);
    Np = 14.6 * pow(Re, -0.28);
    revolutionNumber = Re * ((viscosity / MIXTURE_DENSITY) / pow(diameter/
        2, 2));

    powerConsumption = Np * MIXTURE_DENSITY * pow(revolutionNumber, 3)*pow
        (diameter/2, 5);

    totalPowerConsumption += powerConsumption;

    //-------------------------------------------------------------------
    //  output for each reactant value

    if (flagLog) {
        printLine;
        printf("[Reactor %d]\n",
            i);
        printf("  concentration of Butadien of out : C_%-2d = %10.3lf\n",
            i,  concentRation);
        printf("  product heat of polymerization   : Qp   = %10.3lf[W]\n",
            productHeat);
        printf("  viscosity                        : μ_%-2d =
            %10.3lf[Pa*s]\n", i,  viscosity);
        printf("  Nusselt number                   : Nu   = %10.3lf\n",
            Nu);
        printf("  Prandtl number                   : Pr   = %10.3lf\n",
            Pr);
```

```
            printf("  Reynolds number                  : Re   = %10.3lf\n",
                Re);
            printf("  Power number                     : Np   = %10.3lf\n",
                Np);
            printf("  revolution number                : n_%-2d = %10.3lf[rps]
                \n",  i,  revolutionNumber);
            printf("  power consumption                : P_%-2d = %10.3lf[W]
                \n",    i,  powerConsumption);
        }
    }

    //------------------------------------------------------------------
    //  output for each reactant value

    if (flagLog) {
        printLine;
        printf("[Total]\n");
        printf("  total power consumption          : P  = %10.3f[W]\n",
            totalPowerConsumption);
    }

    if (flagLog) {
        printLine;
    }

    return totalResidenceTime;
}

void showUsage(const char *argv[]) {
    fprintf(stderr, "Usage : %s N y0 T2 [--log]\n", argv[0]);
    fprintf(stderr, "\n");
    fprintf(stderr, "    N : number of reactor\n");
    fprintf(stderr, "    y0: initial concentration of monomer\n");
    fprintf(stderr, "    T2: temperature of coolant [°C]\n");
    fprintf(stderr, " --log: when true, output full logs.\n");
    fprintf(stderr, "\n");
}

int main(int argc, const char *argv[]) {
    int reactorNumber;
    double initialConcentration, coolantTemp;
    bool flagLog = false;

    flagLog = (argc == 5 && (strcmp(argv[4], "--log") == 0));

    if (!((argc == 4) || (flagLog && argc == 5))) {
        showUsage(argv);
        exit(-1);
    }

    reactorNumber = atoi(argv[1]);
    initialConcentration = atof(argv[2]);
    coolantTemp = atof(argv[3]);

    printf("%lf\n", getPowerConsumption(reactorNumber, initialConcentration,
        coolantTemp, flagLog));

    return 0;
}
```