
Anexo: Refrigeradores moleculares

Cálculo de las temperaturas en condiciones adiabáticas T_{ad} :

Para el cálculo numérico de T_{ad} parto de la siguiente expresión:

$$\int_{t_i}^t \frac{k(T_i - T)}{T} dt = \int_{T_{ad}}^T \frac{C}{T'} dT', \quad (1)$$

donde T_i es la temperatura en el instante inicial t_i tal, C calor específico y k la conductancia térmica. Añadiendo ahora a ambos lados de la igualdad la entropía total de la muestra a temperatura T :

$$\int_0^T \frac{C}{T'} dT' - \int_{t_i}^t \frac{k(T_i - T)}{T} dt = \int_0^T \frac{C}{T'} dT' - \int_{T_{ad}}^T \frac{C}{T'} dT' \quad (2)$$

$$\int_0^T \frac{C}{T'} dT' - \int_{t_i}^t \frac{k(T_i - T)}{T} dt = \int_0^{T_{ad}} \frac{C}{T'} dT' \quad (3)$$

Renombrando los términos de la ecuación anterior:

$$S_j(T) - \Delta S k_j = S_j(T_{ad}) \quad (4)$$

Esta es una expresión más práctica para buscar T_{ad} para cada punto (t_j, T_j, B_j) , ya que calculando la entropía total de la muestra a la temperatura medida T en el instante j si se resta la variación por las pérdidas debidas a la falta de adiabaticidad del proceso a través de los puentes hasta ese instante, se obtiene que debe ser lo mismo que la entropía total que habría tenido la muestra en condiciones adiabáticas, lo que da una forma en la que T_{ad} está determinada a la hora de hacer un programa que la calcule, utilizando la k definida en el texto principal y valores de calor específico C obtenidos a muchas temperaturas y campos.

Para el cálculo de T_{ad} que cumple la ecuación (3) he escrito un programa en lenguaje C. Las integrales las he hecho usando la regla del trapecio.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define invierte /* Según el fichero inputCp esté ordenado creciente o
decreciente con T */
int main(){
double dt,dT,DeltaSK,SC,SCinf,SCsup,t1,t2,T1,T2,B1,B2,Ti,Tad,f1,f2,g1;
double g2,K1,K2,A,DiferenciaSAinf,DiferenciaSAsup;
double Tcp,Bcp,Cp,DiferenciaTinf,DiferenciaTsup,DiferenciaT;
double DiferenciaB,Cpinf,Cpsup,Bsup,Binf,DiferenciaSA;
double DiferenciaSMinima,DeltaSCFinal,DiferenciaBinf,DiferenciaBsup;
double Tsup,Tinf,SA,SAinf,SAsup;
int p,N,i,j;
N=0;
FILE *input;
FILE *output;
FILE *inputCp;
```

```

output=fopen("out","wt"); /* Salida de la corrección */
input=fopen("direct.dat","rt"); /* Entrada de los puntos (tj,Tj,Bj) */
inputCp=fopen("Cparalela.dat","rt"); /* Entrada calores específicos */

/* N cuenta el número de temperaturas para las que se ha obtenido
   calor específico a campo 0. N será el mismo para todos los demás
   campos disponibles */

while(!feof(inputCp)){
    fscanf(inputCp,"%lf %lf %lf",&Tcp,&Bcp,&Cp);
    if(Bcp==0) N=N+1;
}
rewind(inputCp);
double Tcps[N], Cpssup[N], Cpsinf[N], Cps[N], SCps[N], AuxT[N], AuxC[N];

/* A^-p es el coeficiente en el cálculo de la conductancia
K(T)=A^-p*T^(1.88) */

A=2;
p=8;
fscanf(input,"%lf %lf %lf",&t1,&B1,&T1);
if(B1<0)B1=0;

/* Ti es la temperatura del baño térmico */

Ti=0.5;
DeltaSK=0;

/* El este bucle recorrerá cada punto (tj,Tj,Bj) */

while(!feof(input)){
    fscanf(input,"%lf %lf %lf",&t2,&B2,&T2);
    if(B2<0) B2=0;
    dt=t2-t1;
    K1=A*pow(T1,1.88);
    f1=(double)((Ti-T1)/T1)*K1;
    K2=A*pow(T2,1.88);
    f2=(double)((Ti-T2)/T2)*K2;

    /* Calculo de forma acumulativa de la entropía conforme pasa el tiempo
(DeltaSkj) a partir de la conductancia */

    DeltaSK=DeltaSK+(f1+f2)/2*dt;
    i=j=0;
    DiferenciaTinf=DiferenciaBinf=DiferenciaSAinf=10000000000000;
    DiferenciaBsup=DiferenciaTsup=DiferenciaSAsup=-10000000000000;

```

```

/* Aquí conocido el campo B, busco el campo inferior y superior
   más próximos para los que hay valor de calor específico */

while(!feof(inputCp)){
    fscanf(inputCp,"%lf %lf %lf",&Tcp,&Bcp,&Cp);
    if(Cp<0) Cp=0;
    DiferenciaB=B2-Bcp;
    if(DiferenciaB<=DiferenciaBinf && DiferenciaB>=0){
        Cpinf=Cp;
        DiferenciaBinf=DiferenciaB;
        Binf=Bcp;
    }
    if(DiferenciaB>=DiferenciaBsup && DiferenciaB<=0){
        Cpsup=Cp;
        DiferenciaBsup=DiferenciaB;
        Bsup=Bcp;
    }
}
rewind(inputCp);

/* Ahora una vez encontrados los campos inferior y superior, almaceno
   todos los calores específicos y temperaturas para los que hay medida
   a dichos campos */

while(!feof(inputCp)){
    fscanf(inputCp,"%lf %lf %lf",&Tcp,&Bcp,&Cp);
    if(Cp<0) Cp=0;
    if(Bcp==Binf){
        Tcps[i]=Tcp;
        Cpsinf[i]=Cp;
        i=i+1;
    }
    if(Bcp==Bsup){
        Cpsup[j]=Cp;
        j=j+1;
    }
}

/* A continuación, a partir de los calores específicos anteriores
   y para cada temperatura interopolo linealmente
   el calor específico para el campo B real */

for(i=0;i<N;i++){
    if(Bsup==Binf) Cps[i]=Cpsinf[i];
    if(Bsup!=Binf){

```

```

        Cps[i]=Cpsinf[i]+(Cpssup[i]-Cpsinf[i])/(Bsup-Binf)*(B2-Binf);
    }
}

/* Cambio de unidades para pasar de unidades de R
a J/K para la muestra de 50 microgramos de Formiato de Gadolinio
con masa molar de 292.3 g/mol */

for(i=0;i<N;i++){
    Cps[i]=Cps[i]*8.314/292.3*50*pow(10,-6+p);
}
#ifdef invierte
    for(i=0;i<N;i++){
        AuxT[i]=Tcps[N-1-i];
        AuxC[i]=Cps[N-1-i];
    }
    for(i=0;i<N;i++){
        Tcps[i]=AuxT[i];
        Cps[i]=AuxC[i];
    }
#endif // invierte

/* Calculo la entropía de forma acumulativa al subir la temperatura
a partir de los calores específicos anteriores */

SCps[0]=0;
for(i=1;i<N;i++){
    dT=Tcps[i]-Tcps[i-1];
    g1=Cps[i]/Tcps[i];
    g2=Cps[i-1]/Tcps[i-1];
    SCps[i]=SCps[i-1]+(g1+g2)/2*dT;
}

/* Busco los valores de temperatura más próximos
a la temperatura T en el instante j y sus respectivas entropías
entre las calculadas anteriormente */

for(i=0;i<N;i++){
    DiferenciaT=T2-Tcps[i];
    if(DiferenciaT<=DiferenciaTinf && DiferenciaT>=0){
        SCinf=SCps[i];
        DiferenciaTinf=DiferenciaT;
        Tinf=Tcps[i];
    }
    if(DiferenciaT>=DiferenciaTsup && DiferenciaT<=0){
        SCsup=SCps[i];
    }
}

```

```

        DiferenciaTsup=DiferenciaT;
        Tsup=Tcps [i];
    }
}

/* Interpolo linealmente para la temperatura Tj la entropía total
a partir de las más próximas encontradas lo que da Sj(T) */

if(Tsup==Tinf) SC=SCinf;
if(Tsup!=Tinf) SC=SCinf+(SCsup-SCinf)/(Tsup-Tinf)*(T2-Tinf);

/*Definiendo SA=Sj(T)-DeltaSkj, busco entre las entropías
calculadas anteriormente las más próximas a SA,
ya que SA=Sj(Tad) y con ello ya se puede interpolar Tad */

SA=SC-DeltaSK;
for(i=0;i<N;i++){
    DiferenciaSA=SA-SCps [i];
    if(DiferenciaSA<=DiferenciaSAinf && DiferenciaSA>=0){
        SAinf=SCps [i];
        DiferenciaSAinf=DiferenciaSA;
        Tinf=Tcps [i];
    }
    if(DiferenciaSA>=DiferenciaSAsup && DiferenciaSA<=0){
        SAsup=SCps [i];
        DiferenciaSAsup=DiferenciaSA;
        Tsup=Tcps [i];
    }
}

/* Interpolando linealmente calculo Tad con
las entropías anteriores */

if(SAsup==SAinf) Tad=Tinf;
if(SAsup!=SAinf) Tad=Tinf+(Tsup-Tinf)/(SAsup-SAinf)*(SA-SAinf);
printf("%.14lf %.14lf %.14lf %.14lf\n",t2,B2,T2,Tad);
fprintf(output, "%.14lf %.14lf %.14lf %.14lf\n",t2,B2,T2,Tad);
t1=t2;
T1=T2;
B1=B2;
rewind(inputCp);
}
}

```