

Conduction instationnaire dans la paroi d'un four

| | | |
|----------|-----------------------------|----------|
| 1 | Résolution | 2 |
| 1.1 | Dans la paroi | 2 |
| 1.2 | Sur les surfaces | 2 |
| 2 | Le programme exemple | 3 |
| 2.1 | L'usage | 3 |
| 2.2 | Le code | 3 |
| 3 | À faire | 5 |

L'évolution des températures dans la paroi d'un four est calculée dans le temps et dans l'espace. Les différentes caractéristiques de la paroi peuvent évoluer dans l'espace (couches de matériaux différents) ou dans le temps (en général, la conductivité d'un matériau augmente avec la température).

À l'intérieur du four une loi de température est imposée en fonction du temps, à l'extérieur la convection naturelle et le rayonnement sont pris en compte.

Par hypothèse, le mur est un plan infini, le problème est monodimensionnel et les angles et les coins ne sont pas traités.

| | | |
|-------------------------------------|-----------|---|
| temps | t | s |
| épaisseur | e | m |
| masse volumique | ρ | $\text{kg}\cdot\text{m}^{-3}$ |
| conductivité | k | $\text{W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$ |
| capacité calorifique massique | c | $\text{J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$ |
| température | T | $^{\circ}\text{C}$ |
| épaisseur du mur | e | m |
| position dans le mur | x | m ($0 \geq x \geq e$) |
| flux de chaleur surfacique | φ | $\text{W}\cdot\text{m}^{-2}$ |
| conductance de convection naturelle | h_n | $\text{W}\cdot\text{m}^{-2}\cdot\text{K}^{-1}$ |
| conductance de rayonnement | h_r | $\text{W}\cdot\text{m}^{-2}\cdot\text{K}^{-1}$ |

Dans l'exemple de programme suivant, la température initiale et la température à l'infini extérieur sont fixées à 0°C , la température de la paroi intérieure est une rampe de chauffe jusqu'à 1300°C suivie d'une rampe de refroidissement, h et c sont constants, et k est une fonction de la température.

Ces valeurs sont fixées dans le code par des instructions comme «`const double e=0.200;`» ou des fonctions comme «`double k (double temper) { };`» toutes ces caractéristiques restant facilement modifiables.

1 Résolution

La paroi est maillée en n éléments, la face intérieure est notée $i = 0$ et la face extérieure $i = n + 1$.

1.1 Dans la paroi

À une position x quelconque, la différence entre les flux entrant et sortant d'un élément d'épaisseur Δx est égale à sa variation d'énergie interne, et

$$\varphi_x - \varphi_{x+\Delta x} = \rho c (T_x^{t+\Delta t} - T_x^t) \Delta x$$

avec

$$\varphi = k(T_{x-\Delta x} - T_x)/\Delta x$$

vient

$$T_x^{t+\Delta t} = T_x^t + \underbrace{\frac{k\Delta t}{\rho c(\Delta x)^2}}_{=b} (T_{x-\Delta x}^t - 2T_x^t - T_{x+\Delta x}^t)$$

Le calcul diverge si $b > 0,5$. Le programme exemple est alors interrompu avec un message d'erreur invitant à diminuer la valeur du Δt choisi.

1.2 Sur les surfaces

À la surface intérieure, deux simples rampes de température successives sont imposées sur $T(i = 0)$, une chauffe suivie d'un refroidissement ou d'un maintien à température maximale.

À la surface extérieure, le flux de conduction dans le dernier élément est imposé égal à ceux de la convection naturelle et du rayonnement, $h = h_n + h_r$

$$\varphi = -k \frac{\Delta T}{\Delta x} = h(T_{ext.} - T_\infty)$$

$$T_{n+1} - T_n = h \frac{\Delta x}{k} (T_{n+1} - T_\infty)$$

d'où

$$T_{n+1} = \frac{T_n - (h\Delta x/k)T_\infty}{1 + h\Delta x/k}$$

cette équation étant simplifiée dans le code puisqu'on a posé T_∞ égal à 0°C .

2 Le programme exemple

2.1 L'usage

Dans un environnement Unix ou Linux, on compile le code par la commande
`gcc -lm code.c`

L'exécutable prend par défaut le nom "*a.out*", et la commande
`./a.out` affiche le résultat à l'écran sous la forme d'un tableau. Celui-ci comprend en 1^{ère} colonne la date en secondes, puis les températures depuis l'intérieur (à gauche) vers l'extérieur (à droite). Pour « alléger » l'affichage, celui-ci n'est effectué qu'à certains pas de temps et d'espace, ces pas sont réglables avec les variables *di* et *dj* du programme.

Pour rediriger ce résultat dans un fichier, "*tutu.dat*" par exemple, on utilise la commande

```
./a.out>tutu.dat
```

2.2 Le code

```
/* thermique instationnaire dans la paroi d'un four
d'après le poly de Bernard LEVET
*/

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

double Tint // température intérieure
( double t, double deg_par_sec, double Tmax, int chauff ) {
if (chauff==1) return (t*deg_par_sec) ;
//else return (2.0*Tmax - t*deg_par_sec) ; //même pente au refroidissement
else return (1.5*Tmax - t*deg_par_sec/2.0) ; // pente 2X plus petite
//else return (Tmax) ; // un palier
}

double k ( double temper ) { // conductivité
if (temper<540) return(0.09);
else return (-0.045+0.00025*temper) ;
}

double hr(double tp,double ta){ // conductance de rayonnement (émissivité=1)
return ((5.67e-8*(tp*tp*tp*tp-ta*ta*ta*ta))/(tp-ta));
}
```

```

int main (int argc, char *argv[])
{
const double dt = 1.0 ;           // incrément temps
const double deg_par_sec=0.002; // degrés/seconde paroi intérieure (default 0,
const double e=0.200;           // épaisseur totale mur
const double hn=4.0 ;           // coef.de convection naturelle extérieure
const double rho=300.0 ;        // masse volumique mur
const double c=2000.0 ;         // chaleur massique mur
const int n = 100 ;             // épaisseur découpée en n intervalles
const double Tmax=1300 ;
double T[n+1], T1[n+1] ;
const int di=10 ;               // on affiche en espace seulement tous les di
const int dj=500 ;              // on affiche en temps seulement tous les dj*
double t, dx ;                  // t=temps, dx=incrément espace
int i, j ;
int chauf ;                      // 1: chauffe, 0 refroidissement
double b ;                       // k*dt/(rho*c*dx*dx)

dx = e/(double)n ;              // incrément espace

for( i=0;i<(n+2);i++){ T[i]=0.0; } // température initiale = 0 Celsius
t=0.0 ; j=2 ; chauf=1 ;

printf("%.0lf s ", t ) ;
for( i=0; i<(n+1); i=i+di ){ printf(" %4.0lf", T[i] ) ; }
printf("\n");
do{
    t= t + dt ;
    T[0]=Tint(t,deg_par_sec,Tmax,chauf);
    if (T[0]>Tmax) chauf=0;
    for( i=1; i<(n+1); i++ ){
        b = k(T[i])*dt/(rho*c*dx*dx) ;
        if (b>0.5){
            printf("b=%f, diminuez dt !\n", b ); return(-1) ;
        }
        T1[i] = T[i] + b*(T[i-1]+T[i+1]-2.0*T[i]) ;
    }
    for( i=1; i<n; i++ ){ T[i] = T1[i] ; }
    T[n] = T[n-1]/(1+ (hn+hr(T[n-1],273.0))*dx/k(t)) ; // t extérieure = 0C
    if (j>dj){
        printf("%.0lf s ", t ) ;

```

```
for( i=0; i<(n+1); i=i+di ){ printf("  %.0lf", T[i] ) ; }
printf("\n");
j=1;
}
j++ ;
} while ( t<48000.0 ) ;
printf("\n  dt: %4.2lf s   dx: %.4lf m\n", dt,dx ) ;
printf("  épaisseur %.4lf m \n", e ) ;
printf("  pas d'affichage %.3lf m  et  %.1lf s\n", dx*di, dt*dj ) ;
printf("  hr: %.2lf W/(m2.K)\n", hr(T[n-1],273.0));
return 0 ;
}
```

3 À faire

- Les valeurs des variables sont actuellement fixées « en dur » dans le code, et pourraient être des fonctions externes plus élaborées ;
- rendu graphique automatique ;
- calcul des chaleurs entrante, sortante et accumulée dans la paroi ;
- relevé des températures maximales en fonction de l'espace.