Métodos numéricos

Práctica N°9

Ecuaciones difenrenciales en derivadas parciales.

Se propone hallar la distribución de temperaturas en una placa rectangular que satisface la ecuación de Laplace:

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \text{ en } \Omega = [0,\ 2]\, x[0,\ 1] \\ T(0,\ y) = 0 \\ T(2,\ y) = y \\ \frac{\partial T}{\partial y}(x,\ 0) = 0 \\ \frac{\partial T}{\partial y}(x,\ 1) = 0 \end{cases} \qquad (1)$$

Se sabe que la solución que cumple con ésta ecuación es:

$$T(x,\ y) = \frac{x}{4} - 4 \sum_{n=1,3..}^{\infty} \frac{1}{(n\,\pi)^2 \operatorname{Sinh}[2\,n\,\pi]} \operatorname{Sinh}[n\,\pi\,x] \operatorname{Cos}[n\,\pi\,y]$$

Se propone discretizar el dominio de tal manera que tenga una cantidad de intervalos $1 + N_x$ y $1 + N_y$ con distancia para intervalos $h_x = \frac{2}{N_x+1}$ y $h_y = \frac{1}{N_y+1}$.

Utilizando esto puede llegarse a un sistema de ecuaciones de la forma A.T=b donde A es la matriz de los coeficientes en el interior de la placa, b contiene información de las condiciones de borde y T es el vector incógnita que se quiere hallar.

Una vez armada la matriz y el vector del segundo término (b) se propone hallar la solución T utilizando el método de resolución de Gauss-Seidel, utilizando como límite para el proceso iterativo las siguientes condiciones:

$$a - \left\| T^n - T^{n-1} \right\|_\infty < \text{tol}$$
$$b - \left\| r^n \right\|_\infty < \text{tol}$$

Donde tol es una tolerancia dada para la convergencia.
Se procedió a resolver de estas dos maneras para distintas cantidades $(N_x,\ N_y)$: (19,9),(39,19),(29,59). Se compararon ambos casos y se obtuvieron los siguientes resultados:

- para $(N_x,\ N_y)$ = (19,9) :
  - a- un error de 0.29215 con 18 iteraciones.
  - b- un error de 0.04865 con 313 iteraciones.
- para $(N_x,\ N_y)$ = (39,19):
  - a- un error de 0.43262 con 22 iteraciones.
  - b- un error de 0.08501 con 1335 iteraciones.
- para $(N_x,\ N_y)$ = (49,29):

- a- un error de 0.51298 con 24 iteraciones.
- b- un error de 0.10365 con 3063 iteraciones.

Puede verse en estos resultados que ultilizar el residuo para finalizar el proceso iterativo, conduce a la obtención de errores mas chicos pero con un número de iteraciones mucho mayor.

Luego se propuso comprobar las Temperaturas exacta obtenida en funcíón del paso utilizado con $h_x = h_y$ comparandolo con la solución otorgada por el comando de resolución de sistemas lineales del octave " \ ". utilizando norma infinito y buscando que esta sea menor que 10^{-3}, es decir:

$$||T - T_{\text{exacto}}||_{\infty} < 10^{-3}$$

Se obtuvo que para un paso $h_x = h_y = \frac{1}{28}$ el error caia por debajo de esa tolerancia pedida.

Por ultimo se graficaron las soluciones obtenidas para distintos (N_x, N_y) con el comando " \ ":

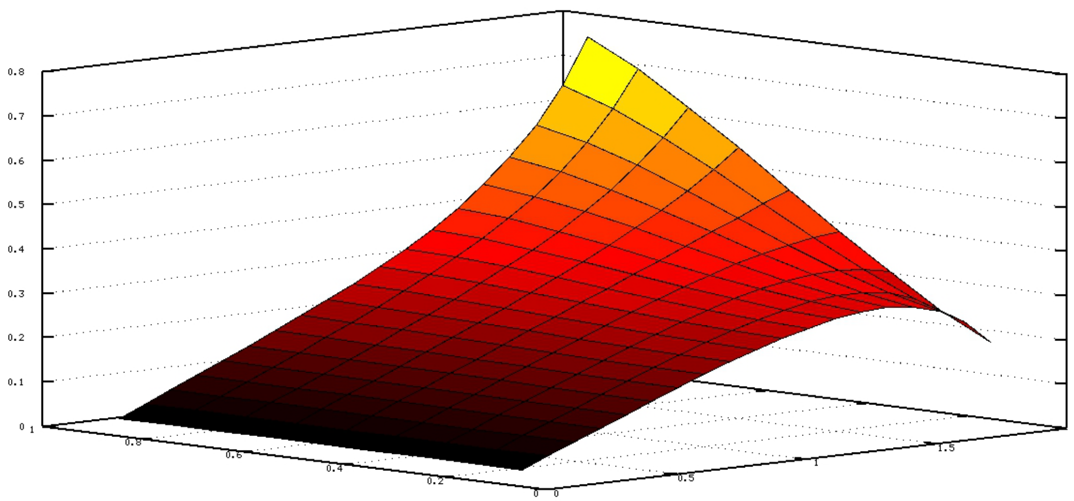

Figura 1: Temperaturas en función de la posición para (N_x, N_y) = (19,9)

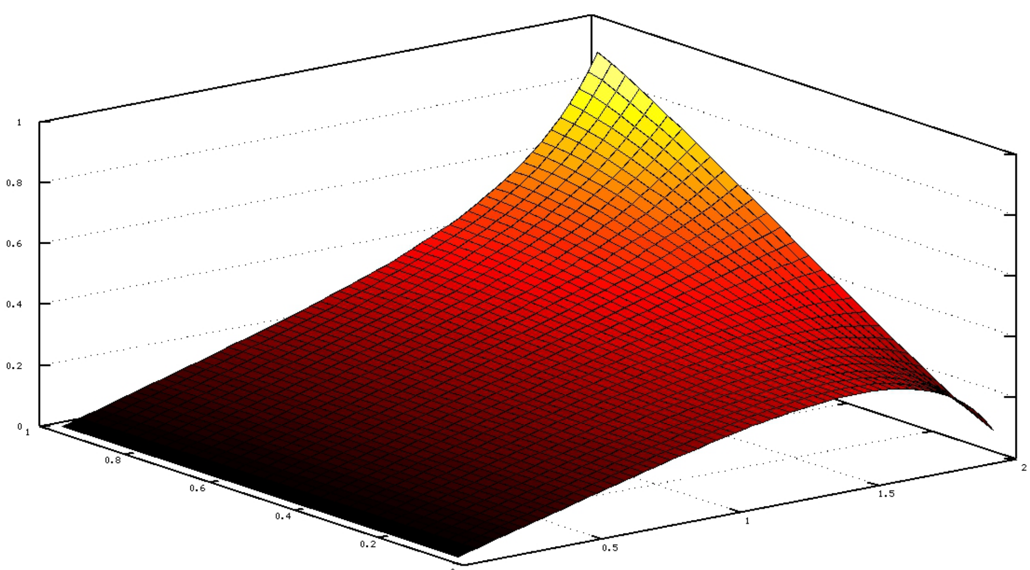

Figura 2: Teperaturas en función de la posición para (N_x, N_y) = (59,29).

Código del programa

```
function [A,b]=sist(h) #función que arma la matriz A y el vector b de segundo miembro.
   hx=hy=h;
   nx=-1+2/h;
   ny=-1+1/h;
   nt=nx*ny;
   A=spalloc(nt,nt,5*nt);

   A+=sparse(1:nt,1:nt,-2/hx^2-2/hy^2,nt,nt,0);

   A+=sparse(1+ny:nt,1:nt-ny,1/hx^2,nt,nt,0);
   A+=sparse(1:nt-ny,1+ny:nt,1/hx^2,nt,nt,0);

   A+=sparse(2:nt,1:nt-1,1/hy^2,nt,nt,0);
   A+=sparse(1:nt-1,2:nt,1/hy^2,nt,nt,0);

   A+=sparse(1:ny:nt,1:ny:nt,7/(2*hy)^2,nt,nt,0);
   A+=sparse(ny:ny:nt,ny:ny:nt,7/(2*hy)^2,nt,nt,0);

   A+=sparse(1:ny:nt , 2:ny:nt , -1/hy^2   ,nt,nt,0);
   A+=sparse(ny+1:ny:nt , ny:ny:nt-1,  -1/hy^2  ,nt,nt,0);

   A+=sparse(ny:ny:nt-1,ny+1:ny:nt,-1/hy^2,nt,nt,0);
   A+=sparse(ny:ny:nt,ny-1:ny:nt,-1/hy^2,nt,nt,0);

   A+=sparse(1:ny:nt,3:ny:nt,1/4/hy^2,nt,nt,0);
   A+=sparse(ny:ny:nt,ny-2:ny:nt,1/4/hy^2,nt,nt,0);

   b =sparse(nt-ny+1:nt  ,  1,-[hy:hy:1-hy]/hx^2 ,nt,1,0);
endfunction
function Te=temperaturaexacta(nx,ny,nt)
suma=0;
  hx=2/(nx+1);
   hy=1/(ny+1);
Te=zeros(nt,1);

for k=1:nx
        x=k*hx;
        for l=1:ny
                y=l*hy;
                for n=1:2:100
                suma=suma+sinh(n*pi*x)*cos (n*pi*y) / ((n*pi)^2 * sinh(2*n*pi));
                endfor
                Te(l+ny*(k-1),1)=x/4-4*suma;
                suma=0;
        endfor
endfor
```

endfunction

```
tol=0.01; #resolución por Gauss-Seidel
error=1;
r=1;
T=zeros(nt,1);
itee=0;
itrr=0;
[A,b]=sist(1/30);
t=time;
do
  ++itee; #++itrr
  for i=1:nt
        T1=b(i);
        if(i>1)
              T1=T1-A(i,i-1)*T(i-1);
        endif
        if(i>2)
              T1=T1-A(i,i-2)*T(i-2);
        endif
        if(i>ny)
              T1=T1-A(i,i-ny)*T(i-ny);
        endif
        if(i<nt)
              T1=T1-A(i,i+1)*T(i+1);
        endif
        if(i<nt-1)
              T1=T1-A(i,i+2)*T(i+2);
        endif
        if(i<(nx-1)*ny+1)
              T1=T1-A(i,i+ny)*T(i+ny);
        endif
        T1=T1/A(i,i);
        e(i)=T1-T(i,1);
        T(i)=T1;
  endfor
  error=max(abs(e))
  #R=b-A*T;
  #r=max(abs(R(:,1)));
until(error<tol) #until(r<tol)

t=time-t

Te=temperaturaexacta(nx,ny,nt);
E=T-Te;
Error=max(abs(E(:,1)))        #error en norma infinito
```

```
n=9; #cálculo del maximo hx
do
```

```
    ++n
    ny=n;
    nx=2*n+1;
    h=1/(n+1);
    [A,b]=sist(h);
    T=A\b;
    nt=nx*ny;
    Te=temperaturaexacta(nx,ny,nt);
    e=max(abs(T-Te));
until(e<1E-3)
h
```