Unidad 4

Ecuaciones diferenciales parciales. (a) Ecuaciones elípticas.

La determinación del potencial electrostático en el interior de un cuadrado de lado unidad libre de carga, donde tres de sus lados están conectados a tierra mientras el restante se mantiene a un potencial constante de 100 V, conduce al siguiente *problema de Cauchy para la ecuación de Laplace*: Hallar la función $u(x, y)$ que satisface la ecuación

$$u_{xx} + u_{yy} = 0$$

para $0 \leq x \leq 1$, $0 \leq y \leq 1$, y que cumple la condición de contorno:

$$\begin{aligned} u(x, 0) &= 0, \quad u(x, 1) = 100, \quad 0 \leq x \leq 1, \\ u(0, y) &= 0, \quad u(1, y) = 0, \quad 0 \leq y \leq 1. \end{aligned}$$

Ejercicio 1. Mostrar que la solución *analítica* del problema, obtenida por el *método de separación de variables*, está dada por

$$u(x, y) = \sum_{k=0}^{\infty} \frac{400}{(2k+1)\pi} \operatorname{sen}[(2k+1)\pi\, x] \times \frac{\operatorname{sh}[(2k+1)\pi\, y]}{\operatorname{sh}[(2k+1)\pi]}$$

Ejercicio 2. Una solución *numérica* del problema es obtenida por el *método de diferencias finitas*. Considerando una grilla con un paso Δ del mismo tamaño en ambas direcciones x e y, tenemos la red de puntos:

$$(x_i = i\Delta, y_j = j\Delta) \quad i, j = 0, 1, \ldots, n = 1/\Delta.$$

Mostrar que el algoritmo de *diferencia finita centrada* determina la aproximación $u_{i,j}$ de $u(x_i, y_j)$ según la fórmula:

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}),$$

con las condiciones de contorno:

$$\begin{aligned} u_{i,0} &= 0 \quad i = 0, \ldots, n \\ u_{i,n} &= 100 \quad i = 0, \ldots, n \\ u_{0,j} &= 0, \quad j = 1, \ldots, n-1 \\ u_{n,j} &= 0, \quad j = 1, \ldots, n-1 \end{aligned}$$

Ejercicio 3. El programa, cuyo código se muestra más adelante, implementa la determinación de la solución numérica por iteraciones sucesivas hasta que los cambios en el potencial son menores que cierta tolerancia prefijada.

a) Con $n = 49$, determine la solución numérica. Grafique la misma junto con las curvas de nivel $u =$ cte.

b) Compare la solución numérica con la solución analítica. Nótese que la solución analítica requerirá considerar un gran número de términos para su suma, y por lo tanto, el seno hiperbólico puede conducir a un desbordamiento positivo para k grande. Para evitarlos, exprese a los mismos en términos de exponenciales.

c) Repita el proceso para otros tamaños de paso.

Programa para la ecuación de Laplace

```
      program laplace
*     ------------------------------------------------------
*     Problema de Dirichlet para la ecuación de Laplace
*     en el cuadrado [0,l]x[0,l]
*     ------------------------------------------------------
*     Declaración de tipo
*     ------------------------------------------------------
      implicit none
      integer maxpoints ! Máximo tamaño de la red
      parameter (maxpoints = 1000)
      double precision u(0:maxpoints,0:maxpoints)
      double precision l,delta
      double precision tol,unew,maxdif
      integer i,j,n,maxiter,niter
*     ------------------------------------------------------
*     Parametros del problema
*     ------------------------------------------------------
      maxiter = 5000   ! Número máximo de iteraciones
      tol     = 1d-8   ! Tolerancia para la solución
      n       = 49     ! N+1 puntos de la malla en x e y
      l       = 1.0d0  ! Lado del cuadrado
      delta   = l/n    ! Paso de la malla en x
*     ------------------------------------------------------
*     Asignar condiciones de contorno
*     ------------------------------------------------------
      do i=0,n
         u(i,0) = 0.0d0    ! Borde inferior
         u(i,n) = 100.0d0  ! Borde superior
      enddo
      do j=1,n-1
         u(0,j) = 0.0d0    ! Borde lateral izquierdo
         u(n,j) = 0.0d0    ! Borde lateral derecho
      enddo
*     ------------------------------------------------------
*     Valores iniciales para los puntos interiores de la red
*     ------------------------------------------------------
      do i=1,n-1
         do j=1,n-1
            u(i,j) = 0.0d0
         end do
      end do
*     ------------------------------------------------------
*     Computar la solucion en forma iterativa
*     ------------------------------------------------------
      do niter = 1,maxiter
         maxdif = 0.0d0
         do j=1,n-1
            do i=1,n-1
               unew  = 0.25d0*(u(i+1,j)+u(i-1,j)+u(i,j-1)+u(i,j+1))
               maxdif = max(maxdif,abs(unew-u(i,j)))
               u(i,j) = unew
            enddo
         enddo
         if (maxdif.lt.tol) then
               write (0,*) 'Procedimiento completado con éxito'
               write (0,*) 'en iteración:',niter
               ! Imprimir solución en formato 3D de gnuplot
```

```
            do i=0,n
                do j=0,n
                    write (*,*) i*delta,j*delta,u(i,j)
                enddo
                write (*,*)
            enddo
            stop
    endif
enddo
write (0,*) 'Número máximo de iteraciones excedido'
stop
end
```