

RESOLUCION LA ECUACION DE DIFUSION EN TRES DIMENSIONES MEDIANTE TECNICAS DE MULTIGRILLA

Luis Cardón y Juan Carlos Rosales

Facultad de Ciencias Exactas - INENCO

Universidad Nacional de Salta, Buenos Aires 177, 4400 Salta, Argentina

cardon@unsa.edu.ar

RESUMEN Se ha desarrollado un programa para la solución de la ecuación de difusión en tres dimensiones. Se usaron métodos iterativos, entre ellos SIPSOL, y se aceleró la convergencia con métodos de multigrilla. Se programó una serie de subrutinas que permiten implementar el algoritmo básico de dos redes, los algoritmos "V" y "W" entre múltiples niveles de redes, y el algoritmo de multigrilla completa. El programa trabaja con fuentes y difusividades dependientes de la posición. Se realizaron las pruebas básicas para validar el programa.

INTRODUCCION

Los problemas de transferencia de calor a través de pisos, pisos radiantes y suelos, han tenido reciente atención en la literatura internacional (Chiasson et al., 2000) y local (Hernández, 2001). Estos problemas están gobernados por la ecuación de difusión. Usualmente, para tener en cuenta posibles no linealidades, se resuelven mediante métodos numéricos iterativos. Si bien la velocidad de las computadoras modernas permite hacer cálculos en tres dimensiones, la convergencia a la solución puede ser lenta e incluso no alcanzarse si la red de discretización es muy grande, razón por la cual los cálculos tridimensionales no son comunes. Atendiendo a estas razones se ha escrito un programa que utiliza técnicas de aceleración de convergencia de tipo multigrilla. En este trabajo se informa de los fundamentos y algunos detalles de implementación del mismo. En trabajos subsecuentes aplicaremos el programa a la solución de problemas de pisos radiantes calentados por vía solar.

El programa es una derivación de otro desarrollado para resolver la ecuación de Poisson, (Cardón, 2000). La principal diferencia entre la ecuación de Poisson y la ecuación de Laplace con fuente, radica en que en la primera solo el "coeficiente de difusión" es constante e igual a uno. En el caso de la ecuación de Laplace, el método de volúmenes de control empleado requiere el cálculo de los coeficientes en las caras de los volúmenes de control y esto requiere interpolación entre los valores dados para los nodos. En el caso de nuestro problema de Poisson (un problema de flujo de fluidos), la condición de borde implementada fue la de derivada normal nula sobre toda la frontera del dominio de cálculo. En cambio, para la ecuación de Laplace, se requieren condiciones de borde más variadas. Por otro lado, para ambas ecuaciones, el método de multigrillas requiere la solución de ecuaciones para el error, y se especifica como condición de borde sobre la frontera del dominio: error nulo. Las condiciones de borde más variadas requeridas por el problema de difusión resultaron en una reprogramación de algunas de las subrutinas del paquete de multigrillas.

METODO DE DISCRETIZACION

La ecuación a resolver esta dada por

$$\nabla \cdot (k \nabla \phi) = G \quad (1)$$

sujeta a condiciones de contorno de tipo Dirichlet o Neumann sobre toda la frontera del dominio de cálculo.

Se discretizó con volúmenes de control sobre redes estructuradas no uniformes. Las ecuaciones de discretización, en la nomenclatura de Patankar (1980) que ya es estandar y donde los subíndices en mayúscula indican nodos y en minúscula indican caras entre volúmenes de control, son:

$$a_P \phi_P = \sum_{nv} a_{nv} \phi_{nv} + b \quad (2)$$

$$a_E = k_e \frac{A_e}{(\delta x)_e}, a_O = k_o \frac{A_o}{(\delta x)_o}, a_N = k_n \frac{A_n}{(\delta y)_n}, a_S = k_s \frac{A_s}{(\delta y)_s}, a_U = k_u \frac{A_u}{(\delta z)_u}, a_D = k_d \frac{A_d}{(\delta z)_d} \quad (3)$$

con

$$a_P = \sum_{nv=e,o,n,s,u,d} a_{nv} \quad y \quad b = \bar{G}\Delta V \quad (4)$$

Las ecuaciones de discretización se resolvieron con métodos iterativos. Estos presentan problemas de convergencia cuando la red de discretización es muy grande, haciéndose aquella muy lenta. Como acelerador de la convergencia se implementó una serie de métodos de multigrilla.

Es conocido que los métodos iterativos son rápidos en alisar los modos de alta frecuencia (longitud de onda comparable al orden del tamaño de la red) pero lentos para suavizar las frecuencias bajas. Por ello los métodos de multigrilla resuelven el problema original y varios problemas similares asociados al error sobre un anidamiento de redes, de manera de resolver cada frecuencia de la solución en una red que le resulta óptima.

METODO DE DOS REDES

El métodos de dos redes es el algoritmo básico. Se presenta un esquema del mismo en la figura 1 . Sobre el se desarrollarán otros algoritmos más complejos:

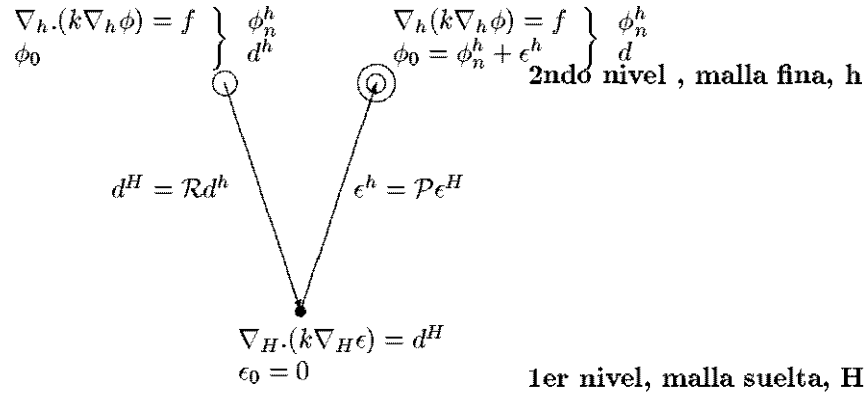


Figure 1: Algoritmo de dos redes. El subíndice cero indica campo de prueba para el comienzo de la iteración

Se quiere resolver con un método iterativo el problema:

$$\delta^2 \phi^h = f^h \quad (5)$$

con alguna condición de borde adecuada (Dirichlet o Neumann) sobre la frontera Γ y con una solución de prueba ϕ_0^h . El operador δ^2 está dado por nuestro método de discretización, ecuación 2. Se ha indicado con h la discretización en una red fina y con H la discretización sobre una red la mitad de densa que la primera.

Luego de n iteraciones se obtiene la solución aproximada ϕ_n^h . El residuo o defecto es:

$$d^h = \delta^2 \phi_n^h - f^h \quad (6)$$

y el error

$$\epsilon^h = \phi_n^h + \phi^h \quad (7)$$

Por la linealidad del operador δ^2 se puede construir el siguiente problema para el error:

$$\delta^2 \epsilon^h = d^h \quad (8)$$

sujeto a $\epsilon^h = 0$ sobre Γ . Podríamos resolver este problema y corregir la primera solución obtenida.

Ahora, como se dijo, los métodos iterativos son rápidos en alisar los modos de alta frecuencia (longitud de onda coparable al orden del tamaño de la red) pero lentos para suavizar las frecuencias bajas. Por eso se elije resolver el error sobre la red H de menos densidad. Se aplica el mismo operador de discretización a la ecuación diferencial, ahora sobre red H . La matriz de coeficientes del problema en ϵ se calcula de la misma forma que la correspondiente para el problema en ϕ , Se tranfiere el defecto d^h a la red H donde se usa como término fuente del problema en el error. Llamamos a esta última operación, restricción.

$$d^H = \mathcal{R}d^h \quad (9)$$

donde \mathcal{R} es el operador restricción.

Resolvemos finalmente el siguiente problema

$$\delta^2 \epsilon^H = d^H \quad (10)$$

con condiciones de borde $\epsilon^H = 0$ sobre Γ y solución de prueba $\epsilon_0^H = 0$.

Sobre la malla H buscamos la mejor solución que se pueda para el error, ϵ^H . Luego, aquel debe ser interpolado a la red h para corregir allí la solución aproximada ϕ_n^h .

La transformación de la red gruesa a la fina se denomina prolongación

$$\epsilon^h = \mathcal{P}\epsilon^H \quad (11)$$

donde \mathcal{P} es el operador prolongación. Luego, puede corregirse la solución en h :

$$\phi^h = \phi_n^h + \epsilon^h \quad (12)$$

Si es necesario, pueden realizarse nuevas iteraciones utilizando los valores corregidos de ϕ por el método de dos redes como valor de prueba.

En la figura 2 se esquematiza la implementación del operador restricción. Se ha usado una red bidimensional para facilitar el dibujo, pero los cálculos se hacen todos en tres dimensiones. Las ecuaciones de interpolación también se presentan en dos dimensiones. A la izquierda se muestra la malla fina y a la derecha la malla gruesa. Abajo se muestran los nodos, cuatro nodos marcados con cruces, en los que se dispone el defecto. Estos nodos se usan para interpolar un valor para el nodo de la red gruesa encerrado por ellos.

En la figura 3 se esquematiza la operación de prolongación. A la derecha se muestra la red gruesa, para cuyos nodos se tiene calculado el error. A la izquierda se muestra la red fina. Para prolongar los valores del error de la red gruesa a la fina se toman cuatro nodos de la red gruesa. Estos nodos se usan para interpolar los valores de los cuatro nodos de la red fina que aquellos encierra.

$$\epsilon^h = \mathcal{P}\epsilon^H \quad (13)$$

La figura 4 esquematiza los detalles del mismo cálculo. En la figura, xc e yc se refieren a las coordenadas de los nodos, los subíndices ii y jj se refieren a la red gruesa y los subíndices $ii f$ y $jj f$ se refieren a la red fina.

Para hacer los cálculos de prolongación y restricción se calcula en coordenadas normalizadas. Las coordenadas de los nodos en la red fina, $xc(ii f)$ y $xc(jj f)$ se expresan en coordenadas locales normalizadas s y e , representadas en la figura 4. Las transformaciones requeridas están dadas por

$$s = 2 \frac{xc(ii f) - xc(ii)}{xc(ii + 1) - xc(ii)} - 1 \quad (14)$$

$$e = 2 \frac{yc(jj f) - yc(jj)}{yc(jj + 1) - yc(jj)} - 1 \quad (15)$$

Luego se calculan los factores de peso que se usan para hacer la interpolación:

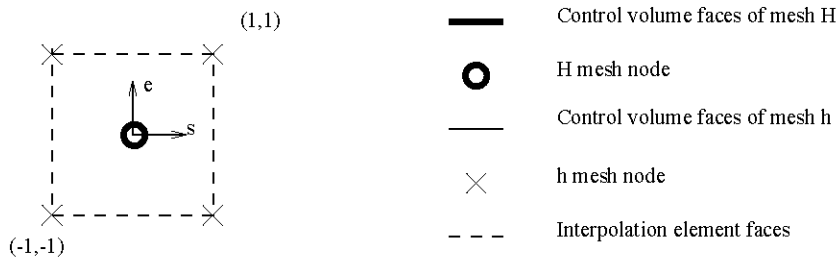
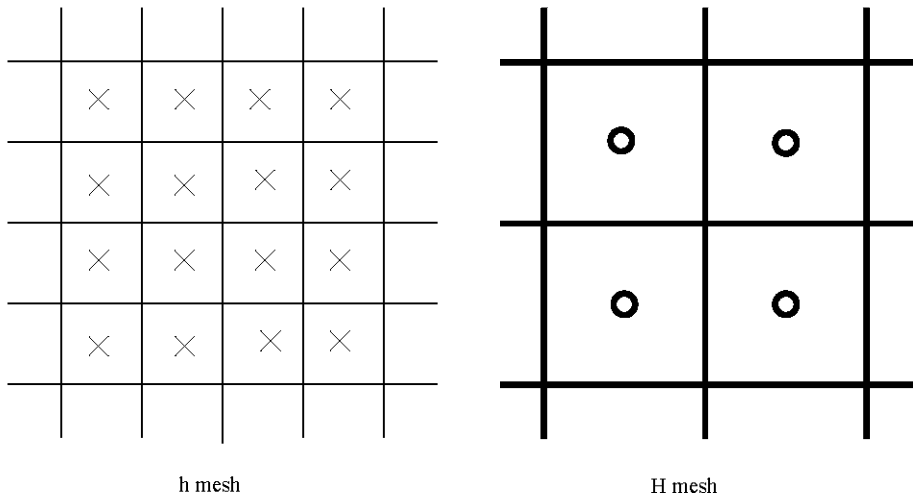
$$N_1 = 0.25(1 - s)(1 - e) \quad (16)$$

$$N_2 = 0.25(1 + s)(1 - e) \quad (17)$$

$$N_3 = 0.25(1 + s)(1 + e) \quad (18)$$

$$N_4 = 0.25(1 - s)(1 + e) \quad (19)$$

Finalmente se interpola:



H nodes limit a normalized interpolation region

Figure 2: Esquema de implementación del operador restricción

$$\epsilon^h = \sum_1^4 N_i \epsilon_i^H \quad (20)$$

La restricción se hace utilizando el mismo método de interpolación.

OTROS METODOS DE MULTIGRILLA IMPLEMENTADOS

Se programaron los componentes para implementar varios métodos:

- **Ciclos V.** Se puede extender el algoritmo de dos redes a múltiples redes. Comenzando de la red fina se puede ir bajando el defecto a una red menos densa, y de allí a otra de densidad menor y así hasta que sea posible una solución exacta. Luego se invierte el ciclo corrigiendo las varias soluciones intermedias. Si se requiere, puede alisarse la solución corregida en cada nivel de red. El ciclo V se ha implementado en la subrutina **VCY**
- **Ciclos W.** Una vez realizado un ciclo V puede tomarse el resultado como valor de prueba y comenzarse tantos otros ciclos V como se quieran. Esto también se ha implementado en la subrutina **VCY**.
- **Multigrilla completa.** Es sabido que los métodos iterativos actúan mejor si se dispone de una buena solución de prueba. Se puede construir una buena solución de prueba partiendo de una red poco densa y prolongando el resultado a una red más fina, y así tantas veces hasta llegar a la red de nivel desado. En cada nivel puede hacerse ciclos V o W. El algoritmo se esquematiza en la figura 5. Este algoritmo se ha implementado en la subrutina **MGSIP2**.

