

## **SIMULACION DE INTERCAMBIADORES DE CALOR DE TIPO CONDUCTO ENTERRADO**

**Mónica Cruz\*** y **Elda Canterle\***, **Luis Cardón†**

\*Depto. de Matemática, UNSA

Buenos Aires 177, 4400 Salta, Argentina (cruz@unsa.edu.ar)

†INENCO, Facultad de Ciencias Exactas, UNSA

Buenos Aires 177, 4400 Salta, Argentina (cardon@unsa.edu.ar)

**Key Words:** bore hole heat exchangers, difusión, conducción, multigrillas.

**Abstract.** *Se desarrolló un procedimiento para calcular el flujo de calor desde conductos que transportan fluido al suelo en donde los conductos están enterrados. La aplicación tiene interés en el campo de transferencia de calor en "bore-holes", acumulación de calor en suelos y acondicionamiento térmico con pisos radiantes. El procedimiento iterativo calcula primero la temperatura del fluido a lo largo del conducto en base al balance de entalpía. Luego, la energía perdida por el fluido se introduce como una fuente de calor en el suelo adyacente al conducto. Después se resuelve la ecuación de calor para el suelo/conductos embebidos. El procedimiento fue implementado en C y en Mathematica. El paquete Mathematico desarrollado implementa también una variedad de algoritmos de multigrilla para acelerar la convergencia.*

## 1. INTRODUCCIÓN

Se ha resuelto el problema del cálculo del flujo de calor entre conductos por los que circula fluido y el suelo en donde los conductos están enterrados.

Esta situación surge en aplicaciones en donde se aprovecha el suelo como acumulador de calor. Se han utilizado conductos enterrados por debajo de invernaderos para atemperar las condiciones nocturnas, se han propuesto los tubos enterrados húmedos para introducir aire fresco (a la temperatura de bulbo húmedo lograda en el conducto) en habitaciones. En Europa se ha propuesto y usado la técnica para acumulación estacional de grandes cantidades de energía térmica, introduciendo y sacando agua caliente por un conducto (Reuss, Beck y Muller).<sup>1</sup> En aplicaciones de este último tipo los tubos se entierran verticalmente en el suelo. Las losas y pisos radiantes son otro ejemplo de la misma tecnología, con la diferencia que aquí los tubos se entierran horizontalmente y cerca de la superficie del suelo.

En este trabajo se desarrolla y experimenta con algoritmos apropiados para resolver el problema térmico del acoplamiento entre los conductos y el suelo de manera tal que puedan utilizarse en la simulación de sistemas que utilicen a aquel como acumulador de calor. El objetivo es que estos algoritmos puedan utilizarse como una subrutina a ser llamada por otros programas adecuados a la simulación del sistema completo.

Como entorno de desarrollo y prueba se ha utilizado el programa *Mathematica*. Es propósito de este trabajo presentar el *paquete* desarrollado para resolver problemas de flujo de calor en sólidos, 2D y 3D, que implementa técnicas de aceleración de multigrillas. El caso 2D, en una versión más simple sin multigrillas, se ha implementado también en lenguaje C.

El problema planteado se resuelve iterando la resolución para el flujo de calor en el conducto y la resolución del flujo de calor por conducción en el suelo. Para este último problema se utiliza la estructura de discretización de un problema de conducción pura. El flujo de calor perdido por el conducto es introducido como una fuente de calor en el suelo, en la adyacencia del conducto. Como aquella depende de las temperaturas del suelo, debe iterarse un cierto número de veces hasta obtenerse convergencia.

## 2. MÉTODO DE RESOLUCIÓN

Se requiere calcular el efecto que el calor transferido desde un conducto por el que circula fluido tiene en la temperatura del suelo, en donde el conducto está enterrado y utilizar para ello la estructura de cálculo de un programa general para resolver problemas de conducción de calor pura mediante el método de los volúmenes de control.

El aspecto fundamental de la metodología es que el cálculo de la conducción en el suelo debe hacerse mediante un subrutina o un paquete cerrado, al cual solo se tiene acceso a través de los parámetros físicos que describen el problema, tales como la conductividad o el término fuente. Otro requerimiento es que la resolución del problema en el suelo debe hacerse en un único dominio de cálculo, sin partición alguna, y en donde los conductos quedan embebidos.

Para cumplir estos objetivos se sigue la siguiente estrategia. Se discretiza el dominio de cálculo (suelo y conductos embebidos). Antes de resolver el problema de conducción en el suelo

se calcula el calor transferido desde el conducto al suelo. Este calor es introducido como una fuente en el suelo, en la adyacencia del conducto. Luego se resuelve el problema de conducción en el suelo.

El recinto de cálculo se discretiza mediante una red de volúmenes de control. Para cada volumen de control se escribe la ecuación de discretización que resulta de integrar la ecuación de calor, ecuación 1 sobre el volumen de control.

El conducto es representado por una hilera unitaria (no necesariamente) de volúmenes de control ubicados según la dirección longitudinal del conducto. Sobre esta hilera de volúmenes de control se resolverá la ecuación de balance de entalpía correspondiente al conducto. Debe tenerse en cuenta que la resistencia que controla la pérdida de calor desde el conducto al suelo, es convectiva, ( $1/h$  con  $h$  el coeficiente de transferencia por convección). Luego en la etapa del cálculo conductivo, estos volúmenes de control serán tratados como cualquier otro dentro del recinto de cálculo, no obstante su conductividad térmica será anulada de manera de que la influencia térmica del conducto sólo se manifieste en el suelo a través del término fuente artificial en la adyacencia del conducto.

En la figura 3 se esquematiza un volumen de control típico para el conducto y su adyacencia.

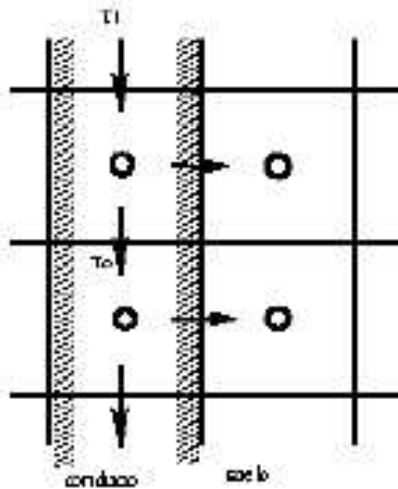


Figura 1: Volumen de control típico para el conducto y los volúmenes de control adyacentes

### 3. ECUACIONES GOBERNANTES Y DISCRETIZACION

#### 3.1. Suelo: difusión

El problema de conducción en el suelo está gobernado por la ecuación de difusión en sólidos con término fuente (la ecuación se puede adicionarse para considerar otros efectos, tales como

dispersión, convección, y se puede resolver acoplada a la convección difusión dispersión de fluidos o gases, ver ecuación en el modelo de de Reuss, por ejemplo)

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + S \quad (1)$$

donde  $T$  es la temperatura,  $k$  la conductividad térmica,  $\rho$  y  $c_p$  la densidad y calor específico y  $S$  la potencia generada internamente o fuente. La ecuación se resuelve sujeta a condiciones de borde sobre toda la frontera del dominio de cálculo, de tipo Dirichlet sobre  $\partial\Omega_D$ , Neumann sobre  $\partial\Omega_N$  o Robinson sobre  $\partial\Omega_R$

$$T = T_B \quad \text{sobre} \quad \partial\Omega_D \quad q = 0 \quad \text{sobre} \quad \partial\Omega_N \quad k \frac{\partial T}{\partial n} = h(T - T_a) \quad \text{sobre} \quad \partial\Omega_R \quad (2)$$

Esta ecuación se discretiza con el método de los volúmenes de control sobre redes estructuradas uniformes o no uniformes, según se requiera. La integración temporal se hace en forma totalmente implícita. Las ecuaciones de discretización, para el caso de dos dimensiones, en la nomenclatura de Patankar<sup>2</sup> que ya es estándar y donde los subíndices en mayúscula indican nodos y en minúscula indican caras entre volúmenes de control, son:

$$a_P T_P = \sum_{nv} a_{nv} T_{nv} + b \quad (3)$$

$$a_E = k_e \frac{A_e}{(\delta x)_e}, a_O = k_o \frac{A_o}{(\delta x)_o}, a_N = k_n \frac{A_n}{(\delta y)_n}, a_S = k_s \frac{A_s}{(\delta y)_s}, a_U = k_u \frac{A_u}{(\delta z)_u}, a_D = k_d \frac{A_d}{(\delta z)_d} \quad (4)$$

con

$$a_P = \sum_{nv=e,o,n,s,u,d} a_{nv} - S_P \Delta x \Delta y, \quad b = S_C \Delta x \Delta y - a_P^0 T_P^0 \quad y \quad a_P^0 = \rho c_p \Delta x \Delta y / dt \quad (5)$$

donde la conductividad térmica es  $k(x, y)$ , el calor específico es  $c_p(x, y)$ , la densidad es  $\rho(x, y)$ , todas función de la posición. El término fuente  $S$  se admite función lineal de la temperatura de la forma  $S = S_C + S_P T_P$ , donde  $S_C$  es la parte constante de la generación de calor mientras que  $S_P$  da cuenta de una dependencia lineal de la fuente con la temperatura. Para cada paso de tiempo se obtiene entonces un problema lineal dado por

$$A \mathbf{T} = \mathbf{b} \quad (6)$$

#### 4. BALANCE DE ENTALPÍA EN EL CONDUCTO

Para el conducto se ha utilizado el modelo más simple posible, un conducto por el que circula fluido sin retorno. En las aplicaciones se requieren otros modelos más complejos, tales como tubos en U o tubos concéntricos, donde debe tenerse en cuenta el retorno del fluido. A los

efectos del desarrollo de la metodología general, el conducto simple en estado estacionario es suficiente.

El conducto pierde calor por sus caras laterales, que llamaremos calor lateral,  $Q_L$ , en balance con el cambio de entalpía del fluido que atraviesa el volumen de control en la dirección del flujo  $Q_{\dot{m}}$ . El fluido, con un caudal  $\dot{m}$  entra por una de las caras del volumen con una temperatura  $T^i$  y sale por la cara opuesta a una temperatura  $T^o$ , de manera que el calor intercambiado es

$$Q_{\dot{m}} = c_p \dot{m} (T_P^o - T_P^i) \quad (7)$$

mientras que el flujo lateral es

$$Q_L = U_e (T_E - T_P) \Delta y + U_w (T_W - T_P) \Delta y \quad (8)$$

donde las conductancias  $U_E$  y  $U_W$

$$U_E = \left( \frac{k_e}{\delta x_e} \right) \quad (9)$$

$$U_W = \left( \frac{k_w}{\delta x_w} \right) \quad (10)$$

se han calculado teniendo en cuenta que las conductividades en la interface están dadas por la media geométrica de las conductividades en los nodos que (para redes uniformes) está dada por

$$k_e = \frac{2K_P K_E}{K_P + K_E} \quad (11)$$

y que la conductividad equivalente para el nodo sobre el conducto, calculada de tal manera que explica la transferencia de calor convectiva debe ser

$$K_P = h \delta x_e^- \quad (12)$$

Resulta entonces que la temperatura de salida de un volumen de control cualquiera sobre el conducto se obtiene de una función de su temperatura de entrada y la temperatura del nodo.

$$T^o = T^i + \frac{\Delta y \sum_{nL} (U_{nL} T_{nL}) - \Delta y \sum_{nL} (U_{nL}) T_P}{c_P \dot{m}} \quad (13)$$

donde el subíndice  $nL$  indica nodos laterales. Esta ecuación tiene dos incógnitas  $T^o$  y  $T_P$ , por lo que debe agregarse una ecuación que establezca la relación adicional para cerrar el problema. La alternativa más sencilla es la de considerar, a los efectos de la pérdida de calor lateral, que la temperatura del fluido dentro del volumen de control es igual a la temperatura de entrada del fluido (upwinding), es decir  $T_P = T^i$ , de esta manera

$$T^o = T^i + \frac{\Delta y \sum_{nL} (U_{nL} T_{nL}) - \Delta y \sum_{nL} (U_{nL}) T^i}{c_P \dot{m}} \quad (14)$$

Esta alternativa tiene la ventaja que resulta en un cálculo explícito, aunque sobreestima en alguna medida la pérdida de calor lateral. Utilizando esta ecuación, y suponiendo las temperaturas del suelo conocidas, se puede precalcular la temperatura del fluido todo a lo largo del conducto. Como la temperatura del suelo está acoplada a la temperatura del conducto, el problema debe resolverse, al menos en principio, en forma iterativa alternado el cálculo de las temperaturas del conducto y del suelo sucesivamente hasta obtener convergencia.

Puede ocurrir que en un volumen de control, la sobrestimación de la pérdida lateral mencionada cause un enfriamiento del fluido tal que resulte en una temperatura de salida por debajo de la temperatura de la fuente fría, el suelo. Esta situación indica que en algún punto del volumen de control la temperatura del fluido y del suelo se han igualado y la transferencia de calor debe ser nula de allí en adelante (mientras la temperatura del suelo se mantenga) por lo que a partir de allí se considera  $T^o = T^i = T_E$ .

El flujo lateral se introduce en el volumen de control adyacente al conducto como una fuente uniformemente distribuida cuyo valor está dado por la relación

$$-\dot{m}C_p(T_P^o - T_P^i) = S_{\Delta h}\Delta x\Delta y \quad (15)$$

de donde

$$S_{\Delta h} = \frac{-\dot{m}C_p(T_P^o - T_P^i)}{\Delta x\Delta y} \quad (16)$$

#### 4.1. Algoritmo iterativo de resolución

El cálculo procede de la siguiente manera.

1. Se parte de un campo de prueba de temperatura sobre el suelo,  $T_P^*$ ,  $\forall P \in \Omega_s$ , la temperatura de entrada para el conducto  $T_i^0$ , y un valor de coeficiente convectivo  $h$ .
2. Se calcula  $K_P$  para los volúmenes del conducto, ecuación 12.
3. Para cada volumen de control sobre el conducto, comenzando por el correspondiente a la entrada, y luego para los siguientes en la dirección del flujo de fluido, se evalúa:
  - la temperatura de salida del volumen de control, ecuación 14.
  - el flujo lateral, ecuación 7 o 8.
  - el término fuente  $S_C$ , ecuación 16
4. Se anula las conductividades sobre el conducto.
5. Se calcula un nuevo campo de temperatura para el sólido,  $T_P$ .
6. Se compara  $T_P$  y  $T_P^*$ , si ambos difieren en más de una cantidad pequeña predeterminada, se reemplaza el campo de prueba por el campo encontrado

$$T_P^* = T_P \quad (17)$$

y se comienza una nueva iteración desde el punto 2. Si en cambio ambos valores coinciden, se ha encontrado la solución al problema.

7. Si el cálculo es temporal, se avanza un paso de tiempo.

## 5. IMPLEMENTACIÓN Y RESOLUCIÓN CON ACELERACIÓN CON MULTIGRI-LLAS

Las ecuaciones de discretización se resuelven con el método SOR (en C) y de Jacobi (y otros) acelerado por multigrillas en el paquete *Mathematico*. Los métodos iterativos convergen muy lentamente cuando la red de discretización es muy grande, y en el presente problema, el cálculo debido a la naturaleza iterativa del procedimiento empleado y al interés en los resultados temporales, el sistema matricial dado por la ecuación 6 deberá resolverse numerosas veces. Los métodos de multigrilla permiten resolver en redes adecuadas a la frecuencia característica de la solución que se busca acelerando la convergencia. Para ello se resuelve 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 resulte óptima.

Para resolver las ecuaciones de discretización se ha utilizado un programa que implementa una serie de métodos de multigrilla en el entorno *Mathematica*. El programa fué descrito en un trabajo precedente (Cruz, Canterle, Cardón).<sup>3</sup> La versión con que se hicieron los cálculos presente incluye numerosas mejoras de programación, no obstante los algoritmos empleados son los mismos. Se describen aquí a los efectos de completar la presentación.

El algoritmo básico es el método de dos redes que se esquematiza en la figura 2. Se trabaja con dos redes una fina, que denotamos  $\Omega^h$  o de 2ndo. nivel y otra menos fina,  $\Omega^H$  o de 1er. nivel. Se resuelve la ecuación 6 en la red  $\Omega^h$  y se resuelve la ecuación de difusión del error en la red  $\Omega^H$ . Los resultados de esta última se utilizan para corregir los resultados de la primera. En nuestro caso la red  $\Omega^H$  tiene el doble de volúmenes de control que la red  $\Omega^h$  por lo que la designaremos  $\Omega^{2h}$  en lo sucesivo. Llamaremos aquí  $u$  a la variable independiente, en este caso  $u = T$  y  $f = b$ . El error se define  $e = u - v$  y la ecuación para el mismo satisface

$$Ae = r = f - Av \quad (18)$$

Esto sugiere que podemos relajar directamente en el error usando la ecuación residual ya que relajar la ecuación original  $Au = f$  con una semilla inicial  $v$  es equivalente a relajar en la ecuación residual  $Ae = r$  con una semilla inicial  $e = 0$ . Esta íntima conexión entre la ecuación original y la ecuación residual motiva el uso de esta última.

Sea  $h$  la grilla de nivel 2 para la cual queremos resolver el problema. Tendríamos siguiente esquema:

- Relajar la ecuación  $Au = f$  en sobre  $\Omega^h$  y obtenemos una aproximación  $v^h$ .
- Computar el residuo  $r = f - Av^h$ .
- Relajar la ecuación residual  $Ae = r$  sobre  $\Omega^{2h}$  para obtener una aproximación del error  $e^{2h}$ .
- Corregir la aproximación  $v^h$  obtenida en  $\Omega^h$  con el error estimado obtenido en  $\Omega^{2h}$  reemplazando  $v^h$  por  $v^h + e^{2h}$ .

-Relajar  $Au = f$  en  $\Omega^h$  con semilla inicial  $v^h$  obtenida en el paso anterior

### 5.1. Los operadores de interpolación

Para poder pasar de una red  $\Omega^h$  a la red  $\Omega^{2h}$  o subir de esta a la primera se definen dos operadores que se denominan de restricción y de prolongación respectivamente. Estos se definen a continuación.

Sea  $N$  la cantidad de puntos de la grilla menos 1, el operador  $I_{2h}^h$  nos permite pasar de  $\Omega^{2h}$  a  $\Omega^h$ , tenemos  $I_{2h}^h(v^{2h}) = v^h$ , entonces las componentes de  $v^h$  están dadas por:

$$\begin{aligned} v_{2i,2j}^h &= v_{ij}^{2h} \\ v_{2i+1,2j}^h &= \frac{1}{2} (v_{ij}^{2h} + v_{i+1,j}^{2h}) \\ v_{2i,2j+1}^h &= \frac{1}{2} (v_{ij}^{2h} + v_{i,j+1}^{2h}) \\ v_{2i,2j+1}^h &= \frac{1}{2} (v_{ij}^{2h} + v_{i,j+1}^{2h}) \\ v_{2i+1,2j+1}^h &= \frac{1}{4} (v_{ij}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}) \quad \text{con } 0 \leq i, j \leq \frac{N}{2} - 1 \end{aligned}$$

El operador que permite pasar de  $\Omega^h$  a  $\Omega^{2h}$  es  $I_h^{2h}(v^h) = v^{2h}$  tenemos que

$$v_{ij}^{2h} = \frac{1}{16} \left[ \begin{array}{c} v_{2i-1,2j-1}^h + v_{2i-1,2j+1}^h + v_{2i+1,2j-1}^h + v_{2i+1,2j+1}^h \\ + 2(v_{2i,2j-1}^h + v_{2i,2j+1}^h + v_{2i-1,2j}^h + v_{2i+1,2j}^h) \\ 4v_{2i,2j}^h \end{array} \right] \quad \text{con } 1 \leq i, j \leq \frac{N}{2} - 1 \quad (19)$$

Para implementar estos dos operadores hemos definido dos funciones: `interha2h2D` y `inter2hah2D`.

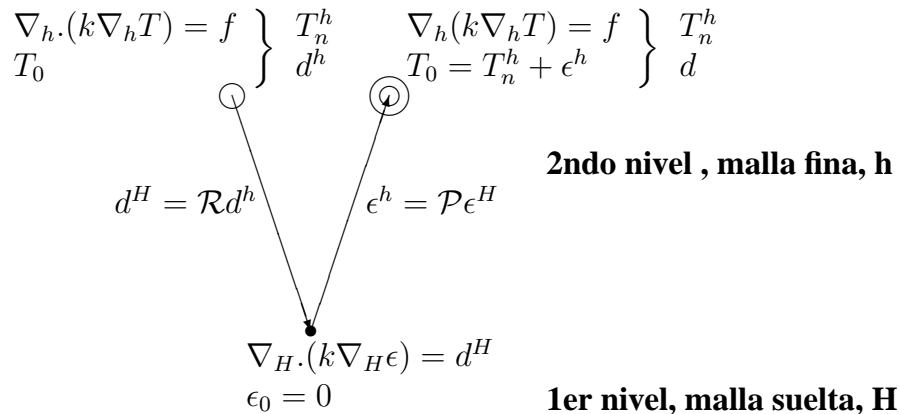


Figura 2: Algoritmo de dos redes. El subíndice cero indica campo de prueba para el comienzo de la iteración



El algoritmo de dos redes se puede combinar de muchas maneras, una de las cuales es el denominando un ciclo  $\mu$ , o algoritmo de multigrilla completo.

## 6. PAQUETES PARA *MATHEMATICA* DESARROLLADOS

A los efectos de resolver las ecuaciones enunciadas arriba se ha creado una serie de paquetes para el Programa *Mathematica*. Describiremos solo dos de ellos, adecuados a la resolución del caso estacionario bidimensional. En el primero se desarrollan la discretización de las ecuaciones y los métodos iterativos básicos para su resolución, en el segundo se desarrollan los métodos de aceleración de convergencia.

### 6.1. Discretización y relajación básico

Las funciones principales del primero de ellos son básicamente dos, la función **jacobi2D** que calcula los coeficientes de la matriz de discretización y aplica una paso de iteración con el método de Jacobi. La segunda, la función **relajarMetPon2D**, permite aplicar multiples iteraciones de Jacobi hasta alcanzar la convergencia de la solución. El paquete presenta además otras funciones auxiliares, otros métodos iterativos que no describiremos aquí, entre ellos una serie de métodos ponderados. La importancia del método de relajación empleado es relativa, si se tiene en cuenta que el propósito es usarlo en combinación con un método de aceleración de multigrillas.

La sintaxis de la función **jacobi2D** es la siguiente

```
jacobi2D[T_List,sc_List,sp_List,KK_List,G_List,Ti_Integer,Tf_Integer,Cbs_List,Cbi_List]
```

donde *T\_List* es el campo de temperatura de prueba definido sobre la red *G\_List*. Esta matriz contiene las coordenadas de los nodos de los volúmenes de control. Los demás parámetros se usa para dar los valores de la matriz de conductividades, los datos de la matriz de términos fuente con sus dos partes *Sc* y *Sp*, las datos de la temperatura sobre los bordes de Dirichlet *Ti*, *Tf*, *Cbs*, *Cbi*. La subrutina devuelve una matriz con el nuevo valor de campo *T* luego de la iteración.

La importancia del algoritmo de relajación es relativa en los cálculos acelerados por multigrillas, no obstante es conocida la poca eficiencia del método de Jacobi. La presente versión del programa permitirá incluir con cierta facilidad otros métodos de relajación.

La sintaxis de la función **relajarMetPon2D** es la siguiente

```
relajarMetPon2D[m_,T_List,s_List,KK_List,G_List,Ti_,Tf_,Cbs_List,Cbi_List,j_Integer]
```

permite realizar una cantidad *j* de iteraciones del método *m* que puede ser cualquiera de los programados (Jacobi o Jacobi Ponderado) u otro que se programe en el futuro.

## 6.2. Aceleración por multigrillas

El paquete de multigrillas implementa varios algoritmos de aceleración, entre ellos, el ciclo  $V$ ,  $W$  y  $\mu$  o multigrilla completa.

El ciclo  $V$  es una extensión del algoritmo de dos redes a múltiples redes. De la red fina se puede ir bajando el defecto a una red menos fina, donde se resuelve la ecuación residual, luego nuevamente se baja de nivel de la misma manera y así hasta que sea posible una solución exacta. Luego se invierte el ciclo corrigiendo las varias soluciones intermedias, usando en cada caso, la solución corregida como semilla para iniciar la relajación correspondiente al nivel. El ciclo  $V$  se ha implementado en la subrutina `esquemaCicloV2D`. Una vez realizado un ciclo  $V$  puede tomarse el resultado como valor de prueba y comenzarse tantos otros ciclos  $V$  como se quieran, esto se denomina ciclo  $W$  y se ha implementado en la subrutina `esquemaCicloW2D`. 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 deseado. En cada nivel puede hacerse ciclos  $V$  o  $W$ . Este algoritmo se ha implementado en la subrutina `esquemaCicloMu2D`.

La sintaxis de `esquemaCicloV2D` es la siguiente

```
esquemaCicloV2D[T_List,s_List,KK_List,G_List,Ti_,Tf_,
Cbs_List,Cbi_List,m_,w_?Positive,r1_Integer,
r2_Integer,g_?Positive]
```

la función devuelve la corrección de la semilla inicial  $T$  al aplicar un esquema de ciclo  $V$ . Además de los parámetros que definen la matriz  $A$  y que fueron descritos en el caso de la subrutina `Jacobi2D`, esta subrutina recibe como parámetros a  $r1$ ,  $r2$  que definen el número de iteraciones de relajación durante la bajada y subida respectivamente. También recibe el parámetro  $g$  relacionado con el número de niveles del ciclo.

La sintaxis de `esquemaCicloW2D` es la siguiente

```
esquemaCicloW2D[T_List,s_List,KK_List,G_List,Ti_,Tf_,
Cbs_List,Cbi_List,m_,w_?Positive,r1_Integer,
r2_Integer,g_?Positive]
```

Esta función devuelve la corrección de la semilla inicial al aplicar un esquema de ciclo  $W$ ,

La sintaxis de `esquemaCicloMu2D` es

```
esquemaCicloMu2D[T_List,s_List,KK_List,G_List,Ti_,Tf_,
Cbs_List,Cbi_List,m_,w_?Positive,r1_Integer,r2_Integer,
g_?Positive,u_Integer]
```

Esta función devuelve la corrección de la semilla inicial al aplicar un esquema de ciclo  $\mu$   $u$  veces.

## 7. EXPERIMENTOS NUMÉRICOS

A los efectos de comprobar la metodología expuesta se ha diseñado un experimento numérico sencillo, bidimensional. El dominio de cálculo es rectangular, con condiciones de borde laterales de tipo Dirichlet y condiciones de borde superior e inferior de tipo Neuman. El conducto se hace pasar, centrado con respecto a los bordes laterales, de arriba hacia abajo, sin retorno. Los parámetros se especifican para una zona de suelo definida por  $Lx \times Ly$  cuyos valores se dan en la Tabla ?? junto con los demás parámetros físicos para el suelo: la densidad  $\rho_s$ , el calor específico,  $c_s$ , y la conductividad térmica  $k_s$ . Para el fluido en el conducto se especifica la densidad  $\rho_f$ , el calor específico,  $c_f$ , y la conductividad térmica  $k_f$ , el flujo másico entrante,  $\dot{m}$ , la temperatura de entrada  $T_0^i$  y el valor del coeficiente de transferencia de calor convectiva  $h$ . El

Parámetro	suelo	fluido
$D, m$		0.015
$H, m$	10	—
$L, m$	1	1
$\rho, kg/m^3$	2050	0.991
$c_p, J/kgK$	1840	4179
$k, W/mK$	0.52	0.634
$\dot{m}, kg/s$	—	1.0
$h, W/m^2K$	—	9175
$T_0^i, C$	—	40.
$T_s$	10	—

Cuadro 1: Parámetros correspondientes al experimento numérico 1. Propiedades de agua a 315K, y del suelo 300K. Datos de Incropera y DeWitt.

coeficiente de transferencia de calor se calcula a partir de la correlación de Dittus y Boelter  $Nu_D = 0,023Re_D^{4/5}Pr^n$  con  $n$  igual a 0,4 o 0,3 según el fluido se caliente o enfríe respectivamente, dando un valor de  $h = 9175$  para los valores de la Tabla 1. Los resultados mostrados en la figura 3 fueron calculados por un programa escrito en Lenguaje C. Cada corrida demora unos pocos segundos. Resultados similares se obtienen en *Mathematica*, aunque los cálculos demoran un poco más. Con este último programa, el ambiente interactivo permite cambiar las condiciones del problema y examinar los resultados con mucha facilidad. En la figura de la derecha se muestra la evolución del campo de temperatura en el suelo y la de la izquierda la evolución de la temperatura del conducto. Se observa que luego de cierto tiempo la temperatura del fluido en el conducto es lineal, suposición que se hace con frecuencia en cálculos semi empíricos.

La figura 4 muestra una situación similar a la descrita anteriormente con varios tubos en vez de uno, en donde el flujo tiene siempre la misma dirección y la temperatura de entrada es la misma para todos ellos. Se realizaron ensayos para el caso en un único conducto que recorre el

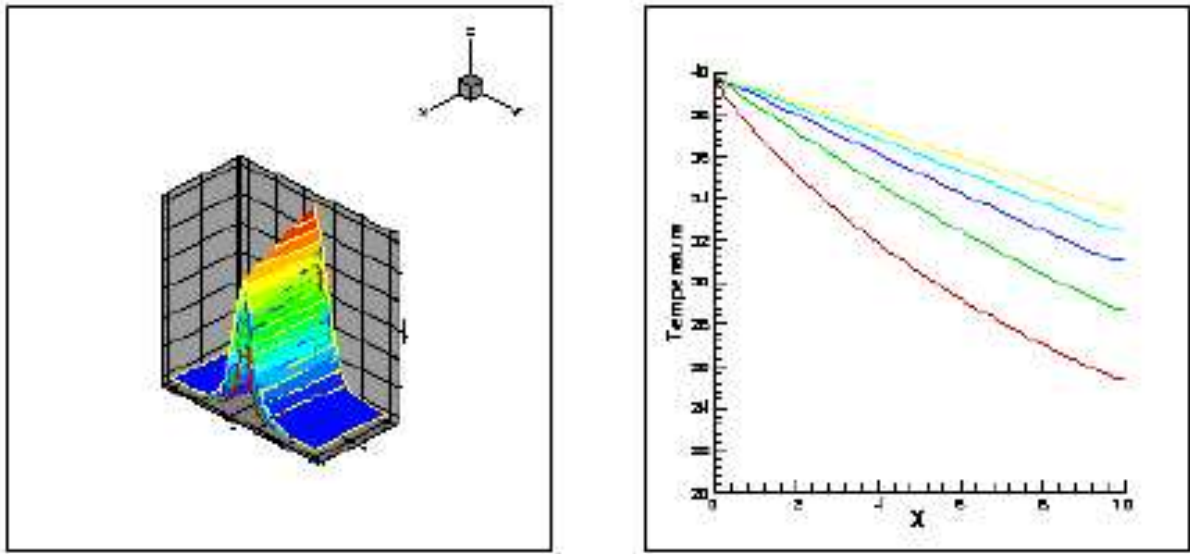


Figura 3: a) Evolución del campo de temperatura en el suelo y b) evolución de la temperatura en el conducto, mostradas cada 2 horas, durante 10 horas.

suelo en uno y otro sentido, como un serpentin, aunque hemos encontrado alguna dificultad en la que trabajamos al momento de esta presentación.

Los experimentos numéricos realizados mostraron una dificultosa convergencia cuando se intenta resolver el estado estacionario directamente. En esta situación el término fuente ocasionado por el conducto en los bordes adyacentes al mismo, calculado en base a la temperatura de suelo de prueba, resulta exagerado, por lo que las temperaturas resultantes en la inmediación del conducto superan en mucho las temperatura de conducto mismo, dificultando así la convergencia. Todos los resultados numéricos mostrados aquí se han realizado simulando la evolución temporal de la temperatura del suelo y del conducto.

## 8. CONCLUSION

Se ha desarrollado un algoritmo iterativo para el cálculo del flujo de calor en tubos enterados. El algoritmo se ha implementado en un programa en C para dos dimensiones y en el programa para *Mathematica* en dos y tres dimensiones, con aceleración de multigrilla en este último caso.

Los programas pueden manejar con facilidad una amplia geometría de conductos, aunque aquí solo se muestran conductos paralelos. También se pueden manejar con facilidad distintas condiciones de borde en el suelo y distintos materiales para el mismo.

El *paquete* desarrollado para el programa *Mathematica* para resolver la ecuación de difusión en tres dimensiones tiene las siguientes características: trabaja sobre dominios rectangulares uni, bi y tridimensionales, la discretización se basa en volúmenes de control con redes es-

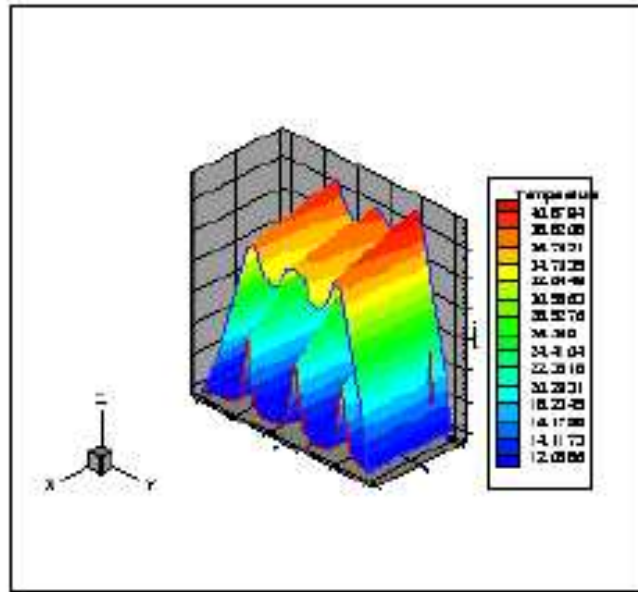


Figura 4: Campo de temperatura en el suelo para un caso con tres conductos ,  $t = 900s$ .

estructuradas no necesariamente uniformes, implementa el método iterativo de Jacobi y Jacobi ponderado, implementa una variedad de algoritmos de multigrilla: el algoritmo de dos redes como caso especial de ciclo V entre dos únicos niveles de red, ciclos V y W que pueden abarcar varios niveles de redes y algoritmo de multigrilla completa, permite la introducción de fuentes por zonas, permite trabajar con zonas de distintos materiales caracterizados por su coeficiente de difusión, permite resolver el caso estacionario y no estacionario, permite obtener una representación gráfica de los resultados en forma inmediata, la organización del programa permite el desarrollo de un archivo para el usuario donde se definen todos los datos concernientes al problema en forma separada del paquete con las subrutinas centrales. Esta última característica se ha usado para implementar el acoplamiento conductos-suelo.

Los cálculos efectuados muestran que con el paquete desarrollado se pueden resolver problemas de conducción transitoria con el programa *Mathematica* de una forma muy conveniente, ya que el programa es suficientemente veloz (aunque en el problema 2D, con redes pequeñas donde las ventajas de las multigrillas no se hacen notar, nuestro programa en C es más rápido) y posee todas las capacidades para hacer un posprocesamiento gráfico inmediato de los resultados. Los resultados son suficientemente satisfactorios como para alentar el desarrollo de paquete para cubrir problemas de tipo convección difusión y de mecánica de fluidos. En particular, el problema del transporte de calor por fluido en las capas freáticas es muy fácil de implementar (se ha hecho en nuestro programa en C) y este es un problema de interés práctico.

## **REFERENCIAS**

- [1] Reuss M., Beck M., and Muller J.P. *Solar Energy*, **59** (1997).
- [2] *Numerical Heat Transfer*. Taylor and Francis, (1980).
- [3] Cruz M., Canterle E., and Cardón L. *AVERMA*, (2003).