PLANTEO MATRICIAL DE ESQUEMAS TEMPORALES

MECÁNICA COMPUTACIONAL

UNL - FICH - OCT - 2017

1. Ecuación de Transporte

La Ecuación de Transporte con todos sus términos se puede expresar como:

$$\rho c_p \dot{\phi} - \nabla \cdot (k \nabla \phi) + v \nabla \phi + c\phi = Q \tag{1}$$

donde:

$$\begin{aligned} \text{Término Temporal} &: \rho c_p \dot{\phi} \\ \text{Término Difusivo} &: \nabla \cdot (k \nabla \phi) \\ \text{Término Advectivo} &: v \nabla \phi \\ \text{Término Reactivo} &: c\phi \\ \text{Término Fuente} &: Q \end{aligned}$$

Si consideramos un sistema arbitrario en estado estacionario, es decir, $\rho c_p \dot{\phi} = 0$, luego el sistema de la *ec.* (1) queda reducido a:

$$-\nabla \cdot (k \nabla \phi) + v \nabla \phi + c\phi = Q \tag{2}$$

2. Planteo matricial del problema estacionario

La expresión (2) es una ecuación diferencial en derivadas parciales, la cual puede ser resuelta de distintas formas. Una de ella es plantear métodos numéricos como *Diferencias Finitas* o *Volúmenes Finitos* en forma de sistemas algebraicos de ecuaciones que permitan hallar de manera discreta una aproximación a la solución en un dominio dado. Por ejemplo, se puede pensar en una solución a la *ec.* (2) como el siguiente sistema:

$$\underline{\underline{K}}\,\underline{\phi} = \underline{F} \tag{3}$$

donde $\underline{\phi}$ representa el vector de incógnitas a resolver, $\underline{\underline{K}}$ es la matriz del sistema y $\underline{F}$ es el vector RHS. Luego el sistema anterior se puede resolver según:

$$\underline{\phi} = (\underline{\underline{K}})^{-1}\underline{F} \tag{4}$$

3. Discretización temporal

Para discretizar el término temporal de la *ec.* (1) se puede utilizar una aproximación de orden 1 según:

$$\rho c_p \dot{\underline{\phi}} \simeq \rho c_p \left[\frac{\underline{\phi}^{n+1} - \underline{\phi}^n}{\Delta t} \right] \tag{5}$$

donde, Δt es el paso temporal, $\underline{\phi}^n$ es el vector solución para el instante de tiempo $n = t(n)$ y $\underline{\phi}^{n+1}$ es el correspondiente al instante $n + 1 = t(n) + \Delta t$. Luego podemos combinar los resultados de las *ec.* (3) y (5) de la siguiente manera:

$$\rho c_p \left[\frac{\underline{\phi}^{n+1} - \underline{\phi}^n}{\Delta t} \right] + \underline{\underline{K}}\,\underline{\phi}^{n+\theta} = \underline{F} \tag{6}$$

siendo θ el parámetro temporal que determina el esquema a utilizar según:

$\theta = 0,$	Forward-Euler o Esquema Explícito
$\theta = 1,$	Backward-Euler o Esquema Implícito
$\theta = 1/2,$	Crank-Nicolson

A partir de la *ec.* (6) y con la correspondiente selección del parámetro θ se pueden econtrar las expresiones matriciales para cada esquema.

4. Forward-Euler o Esquema Explícito

Se parte de la *ec.* (6) con $\theta = 0$:

$$\rho c_p \left[\frac{\underline{\phi}^{n+1} - \underline{\phi}^n}{\Delta t} \right] + \underline{\underline{K}}\,\underline{\phi}^{n} = \underline{F}$$
$$\Rightarrow \underline{\phi}^{n+1} - \underline{\phi}^n + \left(\frac{\Delta t}{\rho c_p}\right) \underline{\underline{K}}\,\underline{\phi}^{n} = \left(\frac{\Delta t}{\rho c_p}\right) \underline{F}$$

Se debe recordar que un vector puede expresarse como el producto de él mismo por la matriz identidad: $\underline{\underline{I}}\,\underline{\phi}^n = \underline{\phi}^n$. En otras palabras, si multiplicamos un vector por una matriz cuadrada con 1 en la diagonal principal y 0 fuera de ella obtendremos como resultado el mismo vector. Luego utilizando esta propiedad se arriba a:

$$\underline{\phi}^{n+1} - \underline{\underline{I}}\,\underline{\phi}^n + \left(\frac{\Delta t}{\rho c_p}\right) \underline{\underline{K}}\,\underline{\phi}^{n} = \left(\frac{\Delta t}{\rho c_p}\right) \underline{F}$$
$$\Rightarrow \underline{\phi}^{n+1} - \left[\underline{\underline{I}} - \left(\frac{\Delta t}{\rho c_p}\right) \underline{\underline{K}}\right] \underline{\phi}^{n} = \left(\frac{\Delta t}{\rho c_p}\right) \underline{F}$$

$$\boxed{\underline{\phi}^{n+1} = \left(\frac{\Delta t}{\rho c_p}\right) \underline{F} + \left[\underline{\underline{I}} - \left(\frac{\Delta t}{\rho c_p}\right) \underline{\underline{K}}\right] \underline{\phi}^{n}} \tag{7}$$

Puede comprobarse en la *ec.* (7) que el vector $\underline{\phi}^{n+1}$ puede ser obtenido como resultado de multiplicaciones entre matrices y vectores con datos conocidos para el tiempo $n = t(n)$.

5. Backward-Euler o Esquema Implícito

Se parte de la *ec.* (6) con $\theta = 1$:

$$
\begin{aligned}
\rho c_p \left[\frac{\underline{\phi}^{n+1} - \underline{\phi}^{n}}{\Delta t} \right] + \underline{\underline{K}}\, \underline{\phi}^{n+1} &= \underline{F} \\
\Rightarrow \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n+1} - \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n} + \underline{\underline{K}}\, \underline{\phi}^{n+1} &= \underline{F} \\
\Rightarrow \left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}}\, \underline{\phi}^{n+1} - \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n} + \underline{\underline{K}}\, \underline{\phi}^{n+1} &= \underline{F} \\
\Rightarrow \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \underline{\underline{K}} \right] \underline{\phi}^{n+1} &= \underline{F} + \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n}
\end{aligned}
$$

$$
\boxed{\underline{\phi}^{n+1} = \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \underline{\underline{K}} \right]^{-1} \left[\underline{F} + \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n} \right]} \tag{8}
$$

Obsérvese que la *ec.* (8) conduce a la resolución de un sistema de ecuaciones similar al planteado previamente en la *ec.* (4):

$$
\underline{\phi}^{n+1} = (\underline{\underline{K}}^{imp})^{-1}\, \underline{F}^{imp}
$$

donde

$$
\begin{aligned}
\underline{\underline{K}}^{imp} &= \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \underline{\underline{K}} \right] \\
\underline{F}^{imp} &= \left[\underline{F} + \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n} \right]
\end{aligned}
$$

6. Crank-Nicolson

El método Crank-Nicolson produce una aproximación de orden 2 al promediar las soluciones para los instantes $n = t(n)$ y $n + 1 = t(n) + \Delta t$. Partiendo de la (6) con $\theta = 1/2$ tiene entonces:

$$
\begin{aligned}
\rho c_p \left[\frac{\underline{\phi}^{n+1} - \underline{\phi}^{n}}{\Delta t} \right] + \underline{\underline{K}} \left[\frac{\underline{\phi}^{n+1} + \underline{\phi}^{n}}{2} \right] &= \underline{F} \\
\Rightarrow \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n+1} - \left(\frac{\rho c_p}{\Delta t} \right) \underline{\phi}^{n} + \frac{1}{2} \underline{\underline{K}}\, \underline{\phi}^{n+1} + \frac{1}{2} \underline{\underline{K}}\, \underline{\phi}^{n} &= \underline{F} \\
\Rightarrow \left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}}\, \underline{\phi}^{n+1} - \left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}}\, \underline{\phi}^{n} + \frac{1}{2} \underline{\underline{K}}\, \underline{\phi}^{n+1} + \frac{1}{2} \underline{\underline{K}}\, \underline{\phi}^{n} &= \underline{F} \\
\Rightarrow \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \frac{1}{2} \underline{\underline{K}} \right] \underline{\phi}^{n+1} &= \underline{F} + \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} - \frac{1}{2} \underline{\underline{K}} \right] \underline{\phi}^{n}
\end{aligned}
$$

$$
\boxed{\underline{\phi}^{n+1} = \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \frac{1}{2} \underline{\underline{K}} \right]^{-1} \left[\underline{F} + \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} - \frac{1}{2} \underline{\underline{K}} \right] \underline{\phi}^{n} \right]} \tag{9}
$$

Nuevamente tenemos un sistema de la forma:

$$
\underline{\phi}^{n+1} = (\underline{\underline{K}}^{cn})^{-1}\, \underline{F}^{cn}
$$

donde

$$\underline{\underline{K}}^{cn} = \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} + \frac{1}{2} \underline{\underline{K}} \right]$$
$$\underline{F}^{cn} = \left[\underline{F} + \left[\left(\frac{\rho c_p}{\Delta t} \right) \underline{\underline{I}} - \frac{1}{2} \underline{\underline{K}} \right] \underline{\phi}^n \right]$$

7. Pautas para la implementación en Matlab o GNU Octave

A continuación se presenta un conjunto de pautas a forma de guía para la implementación de esquemas temporales en Diferencias y Volúmenes Finitos. De manera general, se plantean los esquemas temporales como variaciones a los resultados obtenidos para estado estacionario, donde se reutilizan las matrices y vectores "ensamblados" previamente (matriz $\underline{\underline{K}}$ y vector $\underline{F}$) y se modifican acorde al esquema temporal elegido. En la *fig.* (1) se presenta un diagrama de la interacción entre los resultados para estado estacionario con los esquemas temporales.

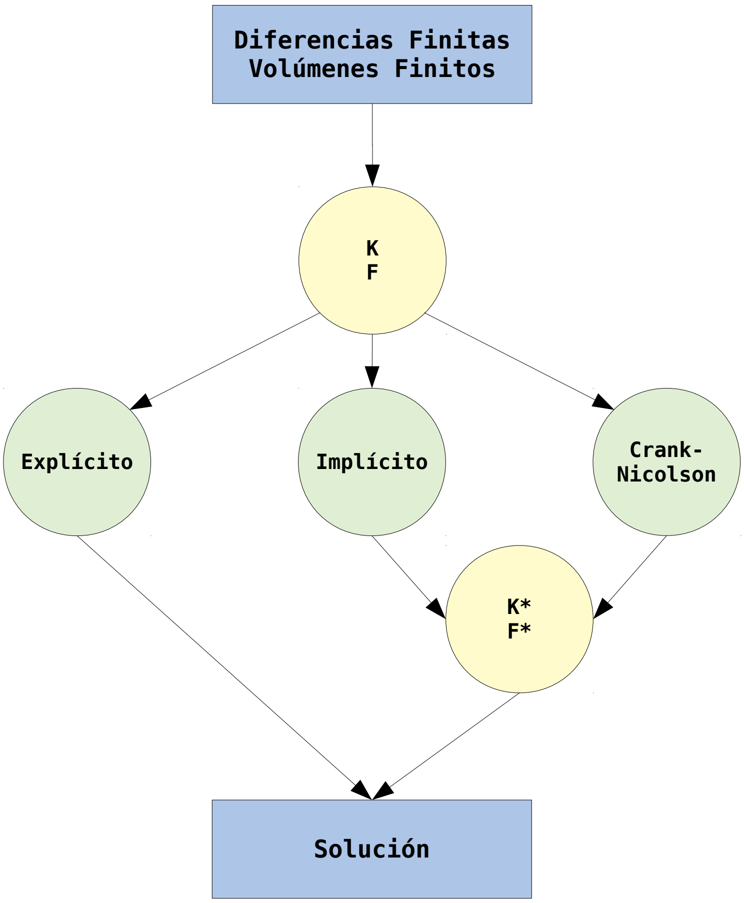

Figura 1: Diagrama general de implementación de esquemas temporales.

Se observa una diferenciación entre el esquema explícito y los demás. Tal como fuera explicado previamente, el método explícito no supone resolver un sistema de la forma $\underline{\underline{K}}\,\underline{\phi}^{n+1} = \underline{F}$, sino que los valores de $\underline{\phi}^{n+1}$ se obtienen directamente a partir de la multiplicación de matrices y vectores de datos conocidos para el instante de tiempo $n = t(n)$. En cambio los restantes métodos modifican la entrada ($\underline{\underline{K}}$ y $\underline{F}$) y resuelven un sistema con los valores modificados.

7.1. Diferencias Finitas

En el *pseudo-código* (1) se presenta un esquema posible para la implementación de Diferencias Finitas con esquemas temporales.

```
1  function [K,F,PHI] = fdm(model,NEU,ROB,DIR,...)
2    %inicializar la matriz de coef. K y el vector del lado derecho F
3    K = sparse(N,N);
4    F = sparse(N,1);
5    %calcular los coeficientes para los nodos interiores (no borde)
6    [K,F] = fdm_nodos_interiores(K,F,...);
7    %calcular los coeficientes para los nodos frontera Neumann
8    [K,F] = fdm_neumann(K,F,NEU,...);
9    %calcular los coeficientes para los nodos frontera Robin
10   [K,F] = fdm_robin(K,F,ROB,...);
11   %calcular los coeficientes para los nodos frontera Dirichlet
12   [K,F] = fdm_dirichlet(K,F,DIR,...);
13   %seleccion del metodo para resolucion
14   if model.ts == 0 %explicito
15     [PHI] = fdm_explicito(K,F,DIR,...);
16   elseif model.ts == 1 %implicito
17     [PHI] = fdm_implicito(K,F,DIR,...);
18   else %estacionario
19     PHI = K\F; %resuelve el sistema de ecuaciones
20     fdm_graficar(...);
21   end
22 end
```

Pseudo-código 1: Método de Diferencias Finitas

Como puede observarse en la líneas 6,8,10 y 12, cada llamada a función recibe como parámetros de entrada $\underline{\underline{K}}$ y $\underline{F}$ y producen como salida valores modificados de ellos. Este procedimiento es lo que se denominó previamente como “ensamble”, donde la matriz $\underline{\underline{K}}$ y el vector $\underline{F}$ se van completando con los coeficientes característicos de cada método numérico (en este caso, Diferencias Finitas) y donde también se incluyen las modificaciones pertinentes para incluir las condiciones de borde (obsérvese las entradas NEU, ROB y DIR, que corresponden a la información de los nodos fronteras para cada tipo de condición de borde).

Para el caso puntual de Diferencias Finitas la llamada a función de la línea 12 modifica el sistema de ecuaciones poniendo 1 en la diagonal principal de $\underline{\underline{K}}$ y 0 fuera de ella, y reemplazando el valor de $\underline{F}$ por el valor fijado para cada nodo frontera Dirichlet (temperatura impuesta). En el *pseudo-código* (2) se presenta una la implementación de esquema explícito para Diferencias Finitas.

```
function [PHI] = fdm_explicito(K,F,DIR,model,...)
  %calcular el paso temporal
  dt = fdm_explicito_delta_t(...);
  A = dt/(model.rho*model.cp);  %variable auxiliar con constantes del modelo
  I = eye(model.nnod);          %matriz identidad de NxN (N = model.nnod)
  %inicializar vectores para instante actual y siguiente
  PHI = model.PHI_n;
  PHI_n = model.PHI_n;
  %bucle temporal
  for n = 1 : model.maxit
    %LINEA PRINCIPAL DEL ESQUEMA EXPLICITO
    PHI = A*F + (I - A*K)*PHI_n;
    %DIRICHLET
    PHI(DIR(:,1)) = DIR(:,2);
    %calcular el error
    err = norm(PHI-PHI_n,2)/norm(PHI_n,2);
    %graficar solucion parcial
    fdm_graficar(...);
    if err < tol
      disp('Metodo terminado por tolerancia de error.');
      return;
    end
    %actualizo phi(n+1) sera phi(n) para el siguiente paso
    PHI_n = PHI;
  end
  disp('Metodo terminado por limite de iteraciones.');
end
```

Pseudo-código 2: Esquema Explícito en Diferencias Finitas

La línea 12 corresponde a lo expresado previamente en la *ec.* (7). En la línea 14 se reemplazan directamente los valores de la solución por los impuestos para la frontera Dirichlet.

En el *pseudo-código* (3) se presenta la implementación para esquema implícito en Diferencias Finitas.

```
1  function [PHI] = fdm_implicito(K,F,DIR,model,...)
2    %calcular el paso temporal (arbitrario para esquema implicito)
3    dt = fdm_explicito_delta_t(...)*10;
4    I = eye(model.nnod);          %matriz identidad de NxN (N = model.nnod)
5    %LINEAS EXCLUSIVAS DEL METODO IMPLICITO
6    K = K + (model.rho*model.cp/dt)*I;   %modifica la matriz K agregando valores a la diag. ppal.
7    F0 = F;               %guarda una copia del vector F
8    %inicializar vector para instante actual
9    PHI_n = model.PHI_n;
10   %bucle temporal
11   for n = 1 : model.maxit
12     %ACTUALIZA EL VECTOR F EN CADA INSTANTE DE TIEMPO
13     F = F0 + (model.rho*model.cp/dt)*PHI_n;
14     %DIRICHLET
15     [K,F] = fdm_dirichlet(K,F,DIR,...);
16     %RESOLUCION DEL SISTEMA
17     PHI = K\F;
18     %calcular el error
19     err = norm(PHI-PHI_n,2)/norm(PHI_n,2);
20     %graficar solucion parcial
21     fdm_graficar(...);
22     if err < tol
23       disp('Metodo terminado por tolerancia de error.');
24       return;
25     end
26     %actualizo phi(n+1) sera phi(n) para el siguiente paso
27     PHI_n = PHI;
28   end
29   disp('Metodo terminado por limite de iteraciones.');
30 end
```

Pseudo-código 3: Esquema Implícito en Diferencias Finitas

En la línea 6 se modifica por única vez la matriz $\underline{\underline{K}}$ agregando valores constantes a la diagonal principal. En la línea 13 se modifica en cada iteración el vector $\underline{F}$ ya que este depende de la solución para el instante $n = t(n)$. En la línea 17 se resuelve el sistema $\underline{\underline{K}}\,\underline{\phi}^{n+1} = \underline{F}$.

7.2. Volúmenes Finitos

Para el caso de Volúmenes Finitos la metodología a seguir es idéntica, excepto que la frontera Dirichlet ya no modifica el sistema de ecuaciones alterando filas enteras del mismo con ecuaciones triviales como antes, sino que completa la matriz $\underline{\underline{K}}$ y el vector $\underline{F}$ con coeficientes propios del método.

Nuevamente, los resultados del "ensamble" para el caso estacionario se reutilizan en cada uno de los esquemas temporales y la estructura general del código será similar a lo mostrado previamente en los pseudo-códigos para Diferencias Finitas.

8. Notas finales

Más adelante se verá cómo este razonamiento es aplicable también al método de *Elementos Finitos* con algunas modificaciones.