IV. MATRICES DISPERSAS (Sparse)

1. FORMATO DE ALMACENAMIENTO DE MATRICES EN ARCHIVO MATRIX MARKET (MM)

El formato MM proporciona un mecanismo sencillo para facilitar el intercambio de datos de matrices. En concreto, el objetivo ha sido definir una base mínima para almacenar los datos en un archivo en formato ASCII que puede ser muy fácil de explicar y analizar, pero puede adaptarse a aplicaciones con una estructura más compleja. El formato de intercambio MM de matrices es en realidad una colección de formatos que comparten elementos de diseño. En la especificación inicial son definidos dos formatos de matriz.

* Formato Coordenada

  Es un formato de archivo adecuado para la representación general de matrices dispersas. Sólo los valores no nulos se almacenan, y sus coordenadas se dan de forma explícita.

* Formato Matriz

  Un formato de archivo adecuado para la representación general de las matrices densas. Todas las entradas se proporcionan de forma definida (orientada a columnas).

Varias implementaciones están definidas de cada uno de estos formatos básicos. Estas se obtienen mediante la especificación del dominio al que pertenecen los valores (es decir, reales, complejos, enteros) y al patrón de simetría matriz que puede reducir el tamaño del archivo de datos (es decir; general , simétrica, antisimétrica, hermítica); ya que al ser suficiente el almacenamiento exclusivamente de los valores no-nulos situados encima de la diagonal principal o bien aquellos situados por debajo.

El formato MM es idóneo para la representación de matrices dispersas. Sólo los valores no nulos tienen que ser almacenadas, y las coordenadas de cada uno se dan de forma explícita. Esto se muestra en el siguiente ejemplo de una matriz de dimensiones 5x5 Real, general y dispersa.

```
1     0       0        6       0
0    10.5     0        0       0
0     0     .015       0       0
0   250.5     0      -280     33.32
0     0       0        0      12
```

El archivo MM correspondiente tendría el siguiente contenido:

```
%%MatrixMarket matrix coordinate real general
%=================================================================
================
%
% Este arcgivo ASCII representa una matriz dispersa MxN con L
% no nulos almacenada en el formato Matrix Market:
%
% +----------------------------------------------+
% |%%MatrixMarket matrix coordinate real general | <--- cabecera
% |%                                             | <--+
% |% comentarios                                 |    |-- 0 o más
comentarios
% |%                                             | <--+
% |    M  N  L                                   | <--- filas,
columnas, valores
% |    I1  J1  A(I1, J1)                         | <--+
% |    I2  J2  A(I2, J2)                         |    |
% |    I3  J3  A(I3, J3)                         |    |-- L líneas
% |        . . .                                 |    |
% |    IL JL  A(IL, JL)                          | <--+
% +----------------------------------------------+
%
% Los índices empiezan en 1, es decir A(1,1) es el primer
elemento.
%
%=================================================================
================
    5  5  8
    1     1   1.000e+00
    2     2   1.050e+01
    3     3   1.500e-02
    1     4   6.000e+00
    4     2   2.505e+02
    4     4  -2.800e+02
    4     5   3.332e+01
    5     5   1.200e+01
```

La primera línea contiene el código de tipo. En este ejemplo, se indica que el objeto representado es una matriz en formato de coordenadas y que los datos numéricos que contiene son reales y se representa en forma general. (Por "general" se entiende que en la representación de la matriz no se están aprovechando las propiedades de simetría.).

Las variantes del formato de coordenadas se definen para matrices dependiendo de la naturaleza de los valores, por ejemplo: valores complejos o enteros, así como para aquellos en los que sólo se almacenan los valores no-nulos. (En estos casos se cambiaría en el encabezado Real por Complejo o Entero y se incluiría el patrón de dispersión). En otras variantes se definen los casos en que las simetrías se pueden utilizar para reducir significativamente el tamaño de los datos: simétrica, antisimétrica y hermitiana. En estos casos, sólo se necesita almacenar los valores de la parte triangular inferior. En el caso de antisimétrica se eliminan los valores de la diagonal principal por ser nulos. (Esto se indicaría cambiando "general" por simétrico o hermitianos, en la línea de encabezado).

2. FORMATOS ADICIONALES PARA MATRICES DISPERSAS

Como veremos más adelante en la codificación del producto Matriz Dispersa-Vector (SpMV) y en la biblioteca CUSP, se hace uso de formatos de almacenamiento de matrices concretos que pretenden optimizar al máximo las operaciones en las que intervienen las matrices dispersas (Sparse).

El formato de almacenamiento nos permite lograr dos objetivos fundamentales:

- Ahorrar enormemente lo requisitos de almacenamiento. Ya que se ahorra mucha memoria al elegir el formato más adecuado para cada matriz.

- Ahorrar en el número de operaciones y cálculos a realizar, (en nuestro caso el producto Matriz-Vector). Ya que no se computan aquellos en los que intervienen los valores nulos.

Existen, de hecho, varias representaciones de matrices dispersas, cada una con distintos requerimientos de almacenamiento, características computacionales y métodos de acceso y manipulación de los elementos matriciales. Ya que en el contexto, del producto Matriz-Vector, no se considera la transformación de matrices, sólo se considerarán formatos de matrices dispersas estáticos, en contraste con aquellos formatos adaptados a una rápida inserción y borrado de elementos.

La principal diferencia entre representaciones de matrices dispersas es el patrón de dispersión, es decir; la disposición que adoptan los elementos no nulos, para los cuales está mejor adaptado. Aunque los formatos adaptados para casos muy específicos son computacionalmente más interesantes, es importante considerar esquemas de almacenamiento generales que permiten el almacenamiento de matrices dispersas que obedecen a patrones arbitrarios.

a. FORMATO DIAGONAL (DIA)

Cuando los elementos no nulos se restringen a un reducido número de diagonales, como muestra la figura 1, el formato DIA es una representación indicada. Aunque no es un formato de almacenamiento de propósito general, el almacenamiento diagonal permite codificar matrices procedentes de la aplicación de "stencils" a mallas regulares y también es indicado para almacenar precondicionadores diagonales.

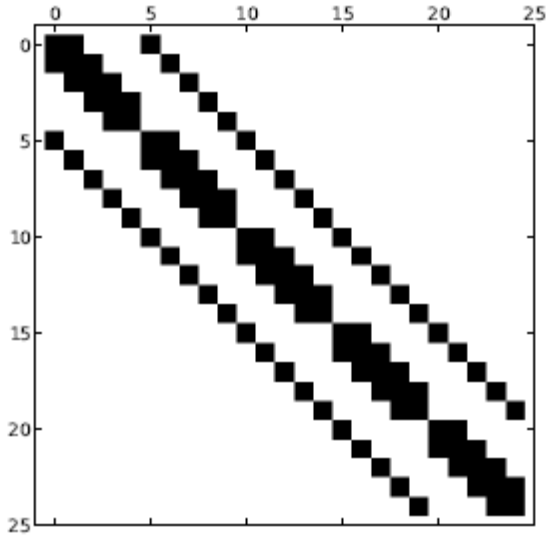

Fig. 1 Patrón de distribución de valores no nulos en una matriz con Dispersión Diagonal

El formato diagonal está formado por dos arrays: datos (que almacena los valores no nulos) y los offsets (que almacena el offset de cada diagonal con respecto a la diagonal principal). Por convenio a la diagonal principal le corresponde el offeset 0. Los valores de "i" positivos corresponden a diagonales superiores y los negativos a las inferiores.

Como ejemplo, sea la matriz A:

$$A = \begin{matrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{matrix}$$

Su representación en formato DIA sería:

$$datos = \begin{matrix} * & 1 & 7 \\ * & 2 & 8 \\ 5 & 3 & 9 \\ 6 & 4 & * \end{matrix} \; y \; offsets = \begin{matrix} -2 & 0 & 1 \end{matrix}$$

El almacenamiento se realiza según las columnas y los (*) indican relleno y pueden almacenar un valor arbitrario.

Las ventajas del formato son dos:

- Los índices de un elemento no nulo están definidos implícitamente según su posición dentro de la diagonal correspondiente y el offset de esta con respecto a la principal. El indexado implícito reduce el tamaño de la matriz en la memoria y el número de datos transferidos a la memoria durante la operación SpMV.

- Los accesos a los datos son contiguos mejorando la eficiencia.

Los inconvenientes son evidentes; este formato reserva valores fuera de la matriz y explícitamente almacena valores nulos que aparecen en las diagonales.

b. FORMATO ELLPACK

Para una matriz MxN con un máximo de k valores no nulos por fila, este formato almacena los valores no nulos en una matriz densa MxK, donde las filas con menos de k valores no nulos son rellenados con ceros. De igual modo, los índices de las columnas correspondientes son almacenados en índices, con relleno con ceros. Es un sistema más general que DIA porque las columnas no nulas no obedecen a ningún patrón en concreto.

Retomando el mismo ejemplo anterior de la matriz A:

$$A = \begin{matrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{matrix}$$

Su representación en formato ELL sería:

$$datos = \begin{matrix} 1 & 7 & * \\ 2 & 8 & * \\ 5 & 3 & 9 \\ 6 & 4 & * \end{matrix} \quad e \quad índices = \begin{matrix} 0 & 1 & * \\ 1 & 2 & * \\ 0 & 2 & 3 \\ 6 & 4 & * \end{matrix}$$

Cuando el número de elementos no nulos por fila no difiere sustancialmente de la media, este formato es muy eficiente. Se debe observar que al igual que el formato DIA, los índices de las columnas se almacenan explícitamente mientras que los de las filas lo son implícitamente.

Las matrices obtenidas mediante mallas semiestructuradas y en especial las mallas no estructuradas de comportamiento correcto se ajustan a este esquema.

En la práctica sin embargo las mallas no estructuradas pueden resultar en una relación arbitraria entre el máximo de no nulos y la media. En estos casos se usa el formato HYB.

c. FORMATO COORDENADA (COORDINATE) COO:

Es un formato sencillo. Los arrays: fila, columna y dato almacenan los índices de la fila, el índice de la columna y el valor de los elementos no nulos de la matriz. COO es una representación general de una matriz dispersa, donde la capacidad de almacenamiento requerida depende de los valores no nulos. A diferencia de DIA y ELL, los índices de filas y columnas se almacenan explícitamente.

La representación de la matriz A en formato COO:

$$A = \begin{matrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{matrix}$$

Sería en forma de tres arrays de tamaño "número de no ceros":

Fila	=[0	0	1	1	2	2	2	3	3]
Columna	=[0	1	1	2	0	2	3	1	3]
Dato	=[1	7	2	8	5	3	9	6	4]

d. FORMATO DE FILA DISPERSA COMPRIMIDA (COMPRESSED SPARSE ROW FORMAT) CSR:

Igual que el formato COO, lo índices de columnas se almacenan explícitamente, al igual que los valores no-nulos en arrays de índices y de datos. Se añade un tercer array de punteros a filas llmado "ptr" con formato CSR. Para una matriz MxN, la tabla de ptr tiene M+1 elementos y almacena el offset de la fila i-ésima en ptr[i]. La última posición de ptr que correspondería a la línea (M+1) almacena el valor NNZ, el total de valores no nulos de la matriz.

Este formato puede ser considerado como una evolución del COO, con un esquema de compresión aplicado a los índices de fila. Por ello la conversión entre ambos formatos es inmediata. El uso de punteros facilita la localización de valores en la matriz, también permite la rápida computación de valores de interés como el número de no nulos en una fila en concreto.

La representación de la matriz A en formato CSR:

$$A = \begin{matrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{matrix}$$

Sería en forma de tres arrays:

Punteros =[0 2 4 7 9], de tamaño número de filas+1 y

Columna =[0 1 1 2 0 2 3 1 3]

Dato =[1 7 2 8 5 3 9 6 4] de tamaño igual al "número de no ceros"

e. FORMATO HÍBRIDO

El formato ELLPACK es idóneo para la arquitectura de vectores. Sin embargo, Cuando el número de valores no nulos varía mucho, la eficiencia del formato ELLPACK disminuye. La eficiencia de dicho formato en el caso de la multiplicación de una matriz por un vector es muy superior al de COO. En cambio la eficiencia del formato COO es invariante con respecto al número de valores no nulos por fila. Por ello se opta por un formato híbrido donde la mayoría de los elementos son almacenados en formato ELLPACK y las restantes en COO.

En el caso de generación de mallas por ejemplo; es a priori sencillo determinar el número de valores no nulos para la mayoría de las filas ya que depende de las vecindades en dicha malla. De ahí que esos elementos se almacenen en formato ELL y dejando los demás en COO.

3. CONCLUSIÓN:

Debido a que como punto de partida sabemos que la Matriz de Rigidez del sistema es una Matriz Real, Cuadrada, Simétrica, con Diagonal Principal No Nula y Dispersa; era obvio que se debían aprovechar algunos de los formatos descritos en este apartado para su almacenamiento y manejo.

Sin embargo se plantearon distintas alternativas atendiendo a cada tarea en particular.

Por ello se concluyó que no se podía usar un formato único ya que disminuiría la eficiencia del método.

Partiendo de este hecho y del hecho de que el sistema de resolución se basaría en un método iterativo donde un paso fundamental lo constituiría el producto Matriz-Vector en la codificación ofrecida por NVIDIA SpMV se concluyó que:

- **Para la generación y almacenamiento de la matriz de rigidez se usaría el formato COO.**
- **Para el almacenamiento en archivo se usaría un formato MM sencillo.**
- **Para la realización de los productos Matriz Dispersa-Vector, donde fuera necesario, se usaría el formato CSR.**
- **Para la generación, almacenamiento y uso del precondicionador diagonal donde fuera necesario se usaría el formato DIA.**