Sparse matrix storage methods

Only the non-zero elements of the matrix are stored and enough information about the indices of these elements, such that it makes it possible to restore completely the matrix

Assume that the matrix ***A*** has ***NN*** non-zero elements.

Compressed row storage

One uses 3 vectors:

- ***valori*** – vector with real values of size ***NN***
- ***ind_col*** – vector with indices of size ***NN***
- ***inceput_linii*** – vector of indices of size ***n+1***

In the vector ***valori*** are stored the non-zero elements of matrix ***A*** row-wise. The vector ***ind_col*** contains the column indices of the elements from vector ***valori***. In vector ***inceput_linii*** are stored the index/position in vectors ***valori / ind_col*** of the first non-zero element on row ***i*** stores in vectors ***valori / ind_col***.

- ***inceput_linii(n+1) = NN+1***
- ***inceput_linii(i+1) – inceput_linii(i) =***

  number of non-zero elements on row ***i, i=1,n***

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 3.5 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

n=5 , NN=12

valori = (102.5, 2.5, 0.33, 1.05, 104.88, 3.5, 100.0, 101.3, 1.3, 1.5, 0.73, 102.23)

ind_col = (1, 3, 5, 3, 2, 1, 3, 4, 2, 4, 1, 5)

inceput_linii = (1, 3, 7, 8, 10, 13)

If the sparse matrix has maximum ***n_max*** non-zero elements on each row, one can use only 2 matrices for the sparse storage:

valori – matrix with real values of size ***n x n_max***
ind_col – matrix of indices of size ***n x n_max***

In matrix ***valori,*** on each row ***i*** are stores the non-zero elements from row ***i*** of matrix ***A*** and in matrix ***ind_col*** are stores the column indices of the corresponding elements from matrix ***valori***.

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 0.0 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

$$valori = \begin{pmatrix} 102.5 & 2.5 & 0 \\ 104.88 & 1.05 & 0.33 \\ 100.0 & 0 & 0 \\ 101.3 & 1.3 & 0 \\ 102.23 & 1.5 & 0.73 \end{pmatrix} \quad ind\_col = \begin{pmatrix} 1 & 3 & 0 \\ 2 & 3 & 5 \\ 3 & 0 & 0 \\ 4 & 2 & 0 \\ 5 & 4 & 1 \end{pmatrix}$$

The diagonals of matrix ***A*** are denoted by:

$$\boldsymbol{d_0} : (\boldsymbol{a_{11}, a_{22}, \ldots, a_{nn}})$$

$$\boldsymbol{d_1} : (\boldsymbol{a_{12}, a_{23}, \ldots, a_{n-1n}})$$

$$\boldsymbol{d_{-1}} : (\boldsymbol{a_{21}, a_{32}, \ldots, a_{nn-1}})$$

$$\boldsymbol{d_2} : (\boldsymbol{a_{13}, a_{24}, \ldots, a_{n-2n}})$$

$$\boldsymbol{d_{-2}} : (\boldsymbol{a_{31}, a_{42}, \ldots, a_{nn-2}})$$

$$\vdots$$

For a matrix that has the non-zero elements placed on some of the diagonals of matrix ***A*** (***n_d*** diagonals with non-zero elements) it is possible to economically store the matrix using a matrix and a vector:

diag – matrix with real values of size ***n x n_d***
diag_no – vector of indices of size ***n_d***

On the columns of matrix ***diag*** are stored the non-zero diagonals of the sparse matrix and in vector ***diag_no*** it is specified the number of the corresponding diagonal stored in column ***j*** of matrix ***diag***.

$$\boldsymbol{diag(i,j) = a_{i\ i+diag\_no(j)}}$$

$$A = \begin{pmatrix} 20.5 & 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 40.5 & 3.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 2.3 & 0.0 & 101.5 & 4.0 \\ 0.0 & 0.0 & 3.0 & 0.0 & 102.5 \end{pmatrix}$$

$$diag = \begin{pmatrix} * & 20.5 & 2.0 \\ * & 40.5 & 3.0 \\ 1.0 & 100.0 & 0.0 \\ 2.3 & 101.5 & 4.0 \\ 3.0 & 102.5 & * \end{pmatrix} \quad diag\_no = (-2, 0, 1)$$

Other types of sparse storages:
http://netlib.org/linalg/html_templates/node90.html