Faculté des Sciences de Base
Master in mathematics
Master in computational science and engineering
Scientific computing applied to linear algebra
March 20, 2011

Serie 6

Exercise 1 *The Morse structure for storing a square sparse matrix of order n, with m nonzero entries, has three variants. In each case, the nonzero entries are stored by rows in an array of real numbers* `double am[m]` *and the column index of* $am[k]$ *in the integer array* `int ic[m]`, *in the variable* `ic[k]`. *For finding the row index of* `am[k]`, *one defines an integer array* `ir`, *according to the three possibilities:*

1. `int ir[m], ir[k]`= *row index of* `am[k]`,
2. `int ir[n], ir[i]=`k_i, k_i *being the location in am of the fisrt nonzero entry of the i-th row,*
3. `int ir[n], ir[i]=`n_i, n_i *being the number of nonzero entries of the i-th row.*

For example, for the matrix:

$$A = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & a_{32} & 0 & a_{34} \end{pmatrix},$$

the three possibilities are:

am	a_{11}	a_{22}	a_{23}	a_{32}	a_{34}
ic	*1*	*2*	*3*	*2*	*4*
ir (1)	*1*	*2*	*2*	*3*	*3*
ir (2)	*1*	*2*	*4*	-	-
ir (3)	*1*	*2*	*2*	-	-

.

Write down (on paper) the C code doing the multiplication of this matrix by a vector `double x[n]`. *To your opinion, what is the best variant and why ?*

Exercise 2 *We consider a square banded matrix A with a symmetric structure (but not necessarly symmetric), which has $1 + 2k_b$ nonzero bands (included the main diagonal), but which can be far from the main diagonal (example: the matrix resulting from the discretization of the Laplacian of Exercise 2, Serie 4).*

1. *Set up a structure for storing only the nonzero bands of A and which allows to do the multiplication $\mathbf{y} = A\mathbf{x}$ "bandwise" (the inner loop runs on the entries of a band). One can be inspired by the structure of the Blas function* `dgbmv`, *in which the bands are stored by rows in a bidimensional array, the number of rows being equal to the number of bands.*
2. *Use your structure for the matrix of the Laplacian. Complete the program in the file* `gbmv.c` *for doing this multiplication and compare (CPU time) to the result obtained with* `dgbmv`. *Choose $N = 80$ and 2000 iterations. Explain the difference.*