Faculté des Sciences de Base
Master in mathematics
Master in computational science and engineering
Scientific computing applied to linear algebra
March 29, 2011

Solution of serie 6

Exercise 1 *One declares two vectors* `double x[n], y[n]` *and initializes* `x`.

1.
```
for (k=0; k<n; k++) y[k]=0;
for (k=0; k<m; k++) y[ir[k]]+=am[k]*x[ic[k]];
```

2.
```
for (i=0; i<n-1; i++)
{ y[i]=0;
for (k=ir[j]; k<ir[j+1]-1; k++) y[i]+=am[k]*x[ic[k]];}
y[n]=0;
for(k=ir[n]; k<m; k++) y[n]+=am[k]*x[ic[k]];
```

3. *One defines an integer variable: counter.*
```
int counter=0;
for(i=0; i<n; i++)
{ y[i]=0;
for(k=0; k<ir[i]; k++)
   { y[i]+=am[counter]*x[ic[counter]];
counter++;}
}
```

The second method is better, because there is only one indirection in the assignments. The inner loop can be partially vectorized by some compilers, which will put to a register several values of ic[0], ic[1], ic[2], ... and use them for vectorizing.

Exercise 2 *One stores the nonzero bands of the matrix A into the rows of the array AB of dimensions $(1+2k_b)\times n$, in such a way that the entries of a band are in the same column of AB as the column of A. The main diagonal is in the row k_b+1, the first nonzero upper diagonal in the row k_b, the second one in the rowd k_b-1, an so on; the first lower diagonal will be in the row k_b+2, the second one in the row k_b+3, a.s.o. Furthermore, one needs the integer array IB of dimension kb such that $IB(i)$ is equal to the column number of the column containing the first entry of the upper band in the i-th row of AB. For example, we consider the following matrix:*

$$A=\begin{pmatrix}
A_{11} & A_{12} & 0 & \cdots & 0 & A_{1s} & 0 & \cdots & 0\\
A_{21} & A_{22} & \ddots & & & & \ddots & \ddots & \vdots\\
0 & \ddots & \ddots & & & & & \ddots & 0\\
\vdots & & & & & & & & A_{n-s+1,n}\\
0 & & & & & & & & 0\\
A_{s1} & & & & & & & & \vdots\\
0 & \ddots & & & & & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & & & & \ddots & A_{n-1,n-1} & A_{n-1,n}\\
0 & \cdots & 0 & A_{n,n-s+1} & 0 & \cdots & 0 & A_{n,n-1} & A_{nn}
\end{pmatrix}.$$

Then the array AB is given by:

$$AB=\begin{pmatrix}
* & \cdots & * & A_{1s} & \cdots & A_{n-s+1,n}\\
* & A_{12} & \cdots & \cdots & A_{n-2,n-1} & A_{n-1,n}\\
A_{11} & A_{22} & \cdots & \cdots & A_{n-1,n-1} & A_{n,n}\\
A_{21} & A_{32} & \cdots & \cdots & A_{n,n-1} & *\\
A_{s1} & \cdots & A_{n,n-s+1} & * & \cdots & *
\end{pmatrix},$$

and the array IB by:

$$IB = \begin{pmatrix} s & 2 & 1 \end{pmatrix}.$$

Hence, in this particular case, the vector $y = Ax$ is obtained from the following pseudocode.

```
for i=1 to n do y(i):=0
for i=1 to 3 do
 for j=ib(i) to n do y(1-ib(i)+j):=y(1-ib(i)+j)+ab(i,j)*x(j)
for i=4 to 5 do
 for j=1 to n-ib(j)+1 do y(ib(i)+j-1):=y(ib(i)+j-1)+ab(i,j)*x(j)
```

The function `dgbmv` *is much slower, because one stores all the zero entries between the extreme nonzero bands and of course time is lost while multiplying with zero numbers!*