CARI'12

Some efficient methods for computing the determinants of large sparse matrices

E. KAMGNIA[(1)] — L. B. NGUENANG[(1)] — B. PHILIPPE[(2)]

(1) Department of Computer Science, University of Yaounde I, Yaounde, Cameroon.
Emails: erkamgnia@yahoo.fr, lnguenang@yahoo.fr

(2) INRIA, Rennes-Bretagne Atlantique, France.
Email: Bernard.Philippe@inria.fr

ABSTRACT. The computation of determinants intervenes in many scientific applications, as for example in the localization of eigenvalues of a given matrix A in a domain of the complex plane. When a procedure based on the application of the residual theorem is used, the integration process leads to the evaluation of the principal argument of the complex logarithm of the function $g(z) = \det((z+h)I - A)/\det(zI - A)$, and a large number of determinants is computed to insure that the same branch of the complex logarithm is followed during the integration. In this paper, we present some efficient methods for computing the determinant of a large sparse and block structured matrix. Tests conducted using randomly generated matrices show the efficiency and robustness of our methods.

RÉSUMÉ. Le calcul de déterminants intervient dans certaines applications scientifiques, comme par exemple dans le comptage du nombre de valeurs propres d'une matrice situées dans un domaine borné du plan complexe. Lorsqu'on utilise une approche fondée sur l'application du théorème des résidus, l'intégration nous ramène à l'évaluation de l'argument principal du logarithme complexe de la fonction $g(z) = \det((z+h)I - A)/\det(zI - A)$, en un grand nombre de points, pour ne pas sauter d'une branche à l'autre du logarithme complexe. Nous proposons dans cet article quelques méthodes efficaces pour le calcul du déterminant d'une matrice grande et creuse, et qui peut être transformée sous forme de blocs structurés. Les résultats numériques, issus de tests sur des matrices générées de façon aléatoire, confirment l'efficacité et la robustesse des méthodes proposées.

KEYWORDS : Determinant, eigenvalues, LU factorization, characteristic polynomial, Schur complement

MOTS-CLÉS : Déterminants, valeurs propres, polynôme caractéristique, factorisation LU, complément de Schur

1. Introduction

The computation of determinants is needed in many occasions, especially when one deals with eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$ since the characteristic polynomial $f(\lambda) = \det(\lambda I - A)$ is based on that function. For instance, for computing the number N_Γ of complex eigenvalues which are surrounded by a given curve (Γ), the procedure EIGENCNT [3] implies evaluating the characteristic polynomial at many points of (Γ) since the selected points must support the quadrature of

$$N_\Gamma = \frac{1}{2i\pi}\int_\Gamma \frac{f^{'}(z)}{f(z)}dz. \tag{1}$$

Let z and $z+h$, be two points of (Γ) and denoting the resolvent by $R(z) = (zI - A)^{-1}$, it then follows that

$$\begin{aligned}\int_z^{z+h} \frac{f'(z)}{f(z)}dz &= \log(\frac{f(z+h)}{f(z)}) \\ &= \log|\Phi_z(h)| + i\arg(\Phi_z(h)),\end{aligned}$$

where $\Phi_z(h) = \det(I + hR(z))$. The goal is therefore to insure that a branch corresponding to a given determination of the complex logarithm can be followed while avoiding any jump to another determination. In [3], a stepsize control is introduced for insuring this property. For large matrices, the number of necessary determinant evaluations may become very high especially when many eigenvalues lie near the boundary (Γ).

In this paper, the goal is to consider efficient techniques for computing the determinant of a large and sparse matrix which is structured with blocks as when obtained from the discretization of a Partial Differential Equation with a Domain Decomposition. Such a structure may also be obtained directly from any sparse matrix as shown in [2].

Computing a determinant always involves an LU-factorization of the matrix. When the LU-factorisation is performed with classical partial pivoting, the matrix is expressed as $A = P^T LU$, where P is the matrix of a permutation, and where L and U are respectively a lower-triangular matrix with unit main diagonal and an upper-triangular matrix. Therefore $\det(A) = \det(P)\det(U)$, where $\det(P) = \pm 1$ and $\det(U)$ is the product of all the diagonal entries of U. In this paper, we investigate ways to take advantage of the block structure of the matrix.

2. Avoiding overflows and underflows when computing a determinant

For any non singular matrix $A \in \mathbb{C}^{n\times n}$, let us consider its LU factorization $PA = LU$ where P is a permutation matrix of signature σ. Then $\det(A) = \sigma(\prod_{i=1}^n u_{ii})$ where $u_{ii} \in \mathbb{C}$ are the diagonal entries of U. When the matrix A is not correctly scaled, the product $(\prod_{i=1}^n u_{ii})$ may generate an overflow or underflow. To avoid this, we represent the determinant using the triplet (ρ, K, n) so that

$$\det(A) = \rho K^n \tag{2}$$

where:

$$\rho = \sigma \prod_{i=1}^{n} \frac{u_{ii}}{|u_{ii}|}, \quad (\rho \in \mathbb{C} \text{ with } |\rho| = 1), \text{ and}$$

$$K = \sqrt[n]{\prod_{i=1}^{n} |u_{ii}|} \quad (K > 0).$$

The quantity K is computed through its logarithm:

$$\log(K) = \frac{1}{n} \sum_{i=1}^{n} \log(|u_{ii}|).$$

Before raising to power n, to protect from under- or overflow, the positive constant K must be in the interval $[\frac{1}{\sqrt[n]{M_{fl}}}, \sqrt[n]{M_{fl}}]$ where M_{fl} is the largest floating point number.

3. Preliminary result

If A can be put in the form $A = I + UV$, where $U \in \mathbb{C}^{m\times n}$, $V \in \mathbb{C}^{n\times m}$, then the following proposition shows how to efficiently compute the determinant of A.

Proposition 3.1 *If* $U \in \mathbb{C}^{m\times n}$,$V \in \mathbb{C}^{n\times m}$ *then,* $\det(I_m + UV) = \det(I_n + VU)$.

Proof. Any eigenvalue of UV is either zero or is an eigenvalue of VU. Indeed, let λ be a nonzero eigenvalue of UV and $w \neq 0$, an associated eigenvector: $(UV)w = \lambda w$. Therefore $(VU)(Vw) = \lambda(Vw)$. Let us prove by contradiction that $Vw \neq 0$: if $Vw = 0$, and since $U(Vw) = \lambda w$, therefore $\lambda w = 0$, and finally, $\lambda = 0$.

Since $Vw \neq 0$, it follows that (λ, Vw) is an eigenpair of VU; therefore the nonzero eigenvalues of UV are nonzero eigenvalues of VU. Since U and V can be interchanged without affecting the result, the two matrices have the same nonzero eigenvalues. An argument of continuity proves that their algebraic multiplicities are the same. For any square matrix M, let us denote its spectrum by $\Lambda(M)$. It then follows that, $\Lambda(UV) \cup \{0\} = \Lambda(VU) \cup \{0\}$, which implies $\Lambda(I_m + UV) \cup \{1\} = \Lambda(I_n + VU) \cup \{1\}$ and therefore, $\det(I_m + UV) = \det(I_n + VU)$. ⋄

This proposition is especially useful when, either $m \ll n$, or $n \ll m$, since it may drastically reduce the order of the matrix for which determinant is sought. We use it twice in the next section.

4. Computing the Determinant of a 2x2-Block Matrix

We now consider the case of a 2x2 block matrix. Let the matrix $A \in \mathbb{C}^{n\times n}$ be defined by:

$$A = \begin{pmatrix} A_1 & A_{12} \\ A_{21} & A_2 \end{pmatrix}, \tag{3}$$

where $A_1 \in \mathbb{C}^{n_1\times n_1}$, $A_2 \in \mathbb{C}^{n_2\times n_2}$ ($n = n_1 + n_2$), and where A_{12} and A_{21} are corner matrices defined by:

$$A_{12} = \begin{pmatrix} 0 & 0 \\ C & 0 \end{pmatrix}, \text{ and } A_{21} = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix},$$

with $B \in \mathbb{C}^{p\times r}$, and $C \in \mathbb{C}^{r\times p}$. The LU factorizations with partial pivoting of the blocks A_1 and A_2, provide the factorizations : $P_1A_1 = L_1U_1$ and $P_2A_2 = L_2U_2$, where $P_i, i = 1, 2$ are permutation matrices. Let us denote:

$$P = \begin{pmatrix} P_1 & 0 \\ 0 & P_2 \end{pmatrix},\ L = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix},\ U = \begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix}, \text{ and } \tilde{A} = PA = \hat{A}+\hat{C},$$

where

$$\hat{A} = L \begin{pmatrix} I_{n_1} & L_1^{-1}P_1A_{12}U_2^{-1} \\ 0 & I_{n_2} \end{pmatrix} U \text{ and } \hat{C} = \begin{pmatrix} 0 & 0 \\ P_2A_{21} & 0 \end{pmatrix}. \tag{4}$$

Since

$$\tilde{A} = \hat{A}(I_n + \hat{A}^{-1}\hat{C}), \tag{5}$$

the determinant of A can be computed from:

$$\det(A) = \det(P)\det(\hat{A})\det(I_n + \hat{A}^{-1}\hat{C}) \tag{6}$$

with $\det(P) = \det(P^T) = (-1)^k$, where k is the number of permutations used in generating P. From (4) and (5), the matrix $\tilde{A}$ can be factorized into

$$\tilde{A} = \hat{A}(I + U^{-1} \begin{pmatrix} I & -L_1^{-1}P_1A_{12}U_2^{-1} \\ 0 & I \end{pmatrix} L^{-1}\hat{C}) \tag{7}$$

To express $L^{-1}\hat{C}$, we can write it as

$$L^{-1}\hat{C} = \begin{pmatrix} L^{-1}\begin{pmatrix} 0 \\ P_2A_{21} \end{pmatrix} & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ L_2^{-1}P_2A_{21} & 0 \end{pmatrix} = \begin{bmatrix} 0 \\ L_2^{-1} \end{bmatrix} \begin{bmatrix} P_2A_{21} & 0 \end{bmatrix},$$

and

$$\det(I_n + \hat{A}^{-1}\hat{C}) = \det\left(I_{n_1+n_2} + U^{-1}\begin{bmatrix} -L_1^{-1}P_1A_{12}U_2^{-1}L_2^{-1} \\ L_2^{-1} \end{bmatrix}\begin{bmatrix} P_2A_{21} & 0 \end{bmatrix}\right).$$

Using Proposition (3.1) to reduce the order of the matrix from $n_1 + n_2$ to n_2, we obtain

$$\det(I + \hat{A}^{-1}\hat{C}) = \det\left(I_{n_2} - P_2\begin{bmatrix} C \\ 0 \end{bmatrix}\begin{bmatrix} 0 & I_p \end{bmatrix} U_1^{-1}L_1^{-1}P_1A_{12}U_2^{-1}L_2^{-1}\right). \tag{8}$$

By a similar technique, the block $A_{12}U_2^{-1}L_2^{-1}$, which appears in the previous expression, can be decomposed into

$$\begin{aligned} A_{12}U_2^{-1}L_2^{-1} &= \begin{pmatrix} 0 & 0 \\ B & 0 \end{pmatrix}\begin{pmatrix} (U_2^{-1})^{11} & (U_2^{-1})^{12} \\ 0 & (U_2^{-1})^{22} \end{pmatrix}\begin{pmatrix} (L_2^{-1})^{11} & 0 \\ (L_2^{-1})^{21} & (L_2^{-1})^{22} \end{pmatrix}, \\ &= \begin{bmatrix} 0 \\ I_p \end{bmatrix}\begin{bmatrix} X & Y \end{bmatrix}, \end{aligned}$$

where $X = B \begin{bmatrix} (U_2^{-1})^{11} & (U_2^{-1})^{12} \end{bmatrix} \begin{bmatrix} (L_2^{-1})^{11} \\ (L_2^{-1})^{21} \end{bmatrix}$ and $Y = B(U_2^{-1})^{12}(L_2^{-1})^{22}$, with the convention that, for $i, j = 1, 2$, the block $M^{ij} \in \mathbb{C}^{n_i \times n_j}$ of a matrix $M \in \mathbb{C}^{n \times n}$, is determined by the initial block structure (3). According to (8), we have

$$\det(I + \hat{A}^{-1}\hat{C}) \quad = \quad \det\left(I_{n_2} - P_2 \begin{bmatrix} C \\ 0 \end{bmatrix} \begin{bmatrix} 0 & I_p \end{bmatrix} U_1^{-1} L_1^{-1} P_1 \begin{bmatrix} 0 \\ I_p \end{bmatrix} \begin{bmatrix} X & Y \end{bmatrix} \right).$$

Applying proposition (3.1) once again to reduce the order of the matrix from n_2 to p leads to:

$$\det(I + \hat{A}^{-1}\hat{C}) = \det\left(I_p - \begin{bmatrix} X & Y \end{bmatrix} P_2 \begin{bmatrix} C \\ 0 \end{bmatrix} \begin{bmatrix} 0 & I_p \end{bmatrix} U_1^{-1} L_1^{-1} P_1 \begin{bmatrix} 0 \\ I_p \end{bmatrix} \right).$$

By denoting

$$\hat{H} = \left(I_p - \begin{bmatrix} X & Y \end{bmatrix} P_2 \begin{bmatrix} C \\ 0 \end{bmatrix} \begin{bmatrix} 0 & I_p \end{bmatrix} U_1^{-1} L_1^{-1} P_1 \begin{bmatrix} 0 \\ I_p \end{bmatrix} \right), \tag{9}$$

we obtain the following result:

Proposition 4.1 *If $A \in \mathbb{C}^{n \times n}$ has the following structure $A = \begin{pmatrix} A_1 & A_{12} \\ A_{21} & A_2 \end{pmatrix}$ where $A_1 \in \mathbb{C}^{n_1 \times n_1}$, $A_2 \in \mathbb{C}^{n_2 \times n_2}$,$n = n_1 + n_1$, $A_{12} = \begin{pmatrix} 0 & 0 \\ B & 0 \end{pmatrix}$, $A_{21} = \begin{pmatrix} 0 & C \\ 0 & 0 \end{pmatrix}$ with $B \in \mathbb{C}^{p \times r}$, $C \in \mathbb{C}^{r \times p}$. From the LU factorizations $P_i A_i = L_i U_i$, $i = 1, 2$, the determinant can be expressed by :*

$$\det(A) = \det(P_1)\det(P_2)\det(U_1)\det(U_2)\det(\hat{H}), \tag{10}$$

with $\hat{H}$ defined by the equation (9).

This decomposition allows an independent computation of the factorizations of A_1 and A_2.

Definition 4.1 *The method based on the formula (10) is called* `2x2`*-method.*

5. Sequential Method with k blocks

We now assume that the matrix A is block tridiagonal:

$$\begin{pmatrix} A_1 & A_{1,2} & 0 & \dots & 0 \\ A_{2,1} & A_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & A_{k-1,k} \\ 0 & \dots & 0 & A_{k,k-1} & A_k \end{pmatrix}, \tag{11}$$

where $A_{i+1,i}$ and $A_{i,i+1}$ are corner matrices defined by: $A_{i,i+1} = \begin{pmatrix} 0 & 0 \\ B_i & 0 \end{pmatrix}$ and $A_{i+1,i} = \begin{pmatrix} 0 & C_i \\ 0 & 0 \end{pmatrix}$, with $A_i \in \mathbb{C}^{n_i \times n_i}$, $B_i \in \mathbb{C}^{p_i \times r_i}$, $C_i \in \mathbb{C}^{r_i \times p_i}$.

From the LU factorization $P_1A_1 = L_1U_1$ and by partitioning $\tilde{A} = PA$ where $P = \begin{pmatrix} P_1 & 0 \\ 0 & I \end{pmatrix}$ and $A = \begin{pmatrix} A_1 & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, then $\tilde{A}$ admits the following LU decomposition:

$$\tilde{A} = \begin{pmatrix} I & 0 \\ A_{21}(P_1A_1)^{-1} & I \end{pmatrix} \begin{pmatrix} P_1A_1 & P_1A_{12} \\ 0 & S_1 \end{pmatrix}$$

with the Schur complement $S_1 = A_{22} - A_{21}U_1^{-1}L_1^{-1}P_1A_{12}$. Thus

$$\det(A) = \det(P_1)\det(A_1)\det(S_1). \tag{12}$$

The Schur complement S_1 can be easily built by

$$\begin{aligned} S_1 &= A_{22} - \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} 0 & X \end{bmatrix} L_1^{-1}P_1 \begin{bmatrix} 0 \\ B \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \\ &= A_{22} - \begin{bmatrix} I \\ 0 \end{bmatrix} \begin{bmatrix} 0 & X \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} \\ &= A_{22} - \begin{pmatrix} XW_2 & 0 \\ 0 & 0 \end{pmatrix} \end{aligned}$$

with $X = C(U_1^{-1})^{22}$ and $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = L_1^{-1}P_1 \begin{bmatrix} 0 \\ B \end{bmatrix}$, with $XW_2 \in \mathbb{C}^{r_1 \times r_1}$. For sufficiently small r_1, only the leading block of order r_1 of A_2 is affected when A_{22} is transformed into S_1.

This approach is then recursively applied to the tailing matrix

$$A_{22} = \begin{pmatrix} A_2 & A_{2,3} & 0 & \dots & 0 \\ A_{3,2} & A_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & A_{k-1,k} \\ 0 & \dots & 0 & A_{k,k-1} & A_k \end{pmatrix},$$

after A_{22} has been corrected. Repeating this process $(k-1)$ times we deduce

$$\det(A) = \prod_{i=1}^{k} \det(P_i)\det(U_i). \tag{13}$$

Definition 5.1 *The method based on the formula (13) is called* `kxk`*-method.*

REMARK. — For the special case $k = 2$, we emphasize on the fact that, methods `2x2` and `kxk` are not equivalent.

6. Numerical Results

Numerical tests were conducted on complex matrices with dense blocks generated randomly. The factorization of dense matrices were obtained using LAPACK procedure [1].

6.1. Efficiency of the `2x2`-method

We considered two methods: (1) the 2x2-method developed in section 4, with dense LU factorization of the dense blocks, and (2) the full factorization of the whole matrix considered as dense. The corresponding running times are denoted t_1 and t_2. We begin by setting $n_1 = n_2 = 1000$ and varying r from 5 to 130, while keeping the relation $r = p$. The ratio $f_{n_1}(r) = \frac{t_2}{t_1}$ is plotted in Figure 1. In the second experiment, the block size n_1 varies and the connecting block size is maintained at $r = 50$. The corresponding speed-up $f_{n_1}(50)$ is plotted in Figure 2.

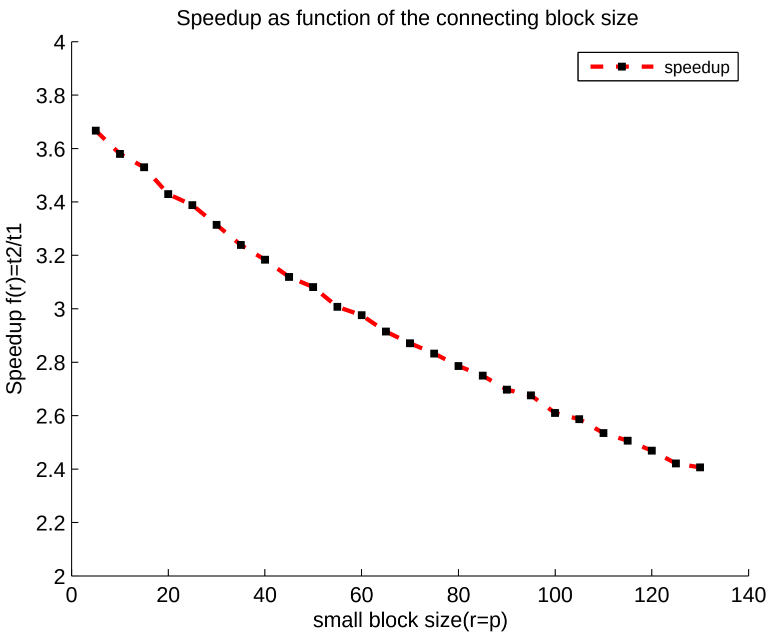

Figure 1. *Speedup* $f_{1000}(r)$.

6.2. Efficiency of the `kxk`-method

Since for $k = 2$, the 2x2-method and kxk-method are not equivalent, we compare them using the following test: $n_1 = n_2 = 1000$, $r = p = 50$. Figures 3 and 4 show the speedup as function of respectively the connecting block size and $A1$ and $A2$ block sizes. The gain with the kxk-method is higher than that with the 2x2-method. However, the latter method allows independent LU-factorizations, as opposed with the kxk-method.

The kxk-method is now compared to the full factorization method, using a set of matrices with a varying number of blocks: from $k = 2$ to $k = 10$, with $n_1 = 1000$, $r_1 = p_1 = 50$, for all the corresponding blocks. Figure 5 displays the results. The speedup behaves quadratically as expected: the full factorization is of complexity $O(k^3 n_1^3)$ whereas the kxk-method complexity is $O(kn_1^3)$.

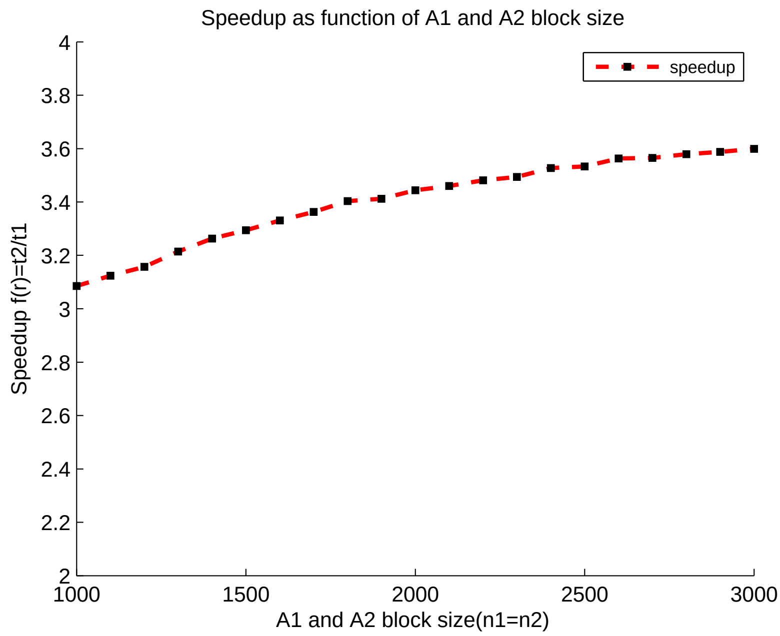

Figure 2. *Speedup* $f_{n_1}(50)$.

7. Conclusion and future works

In this paper, we have proposed efficient procedures for computing determinant of structured sparse matrices. The structure of matrices can be obtained using partitioning methods such as METIS [4]. For the special case of the 1D-partitioning (a block diagonal

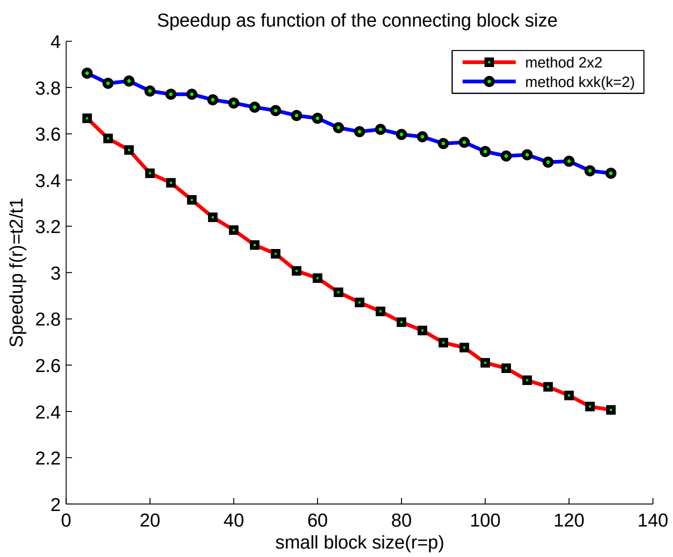

Figure 3. *Speedup* $f_{1000}(r)$.

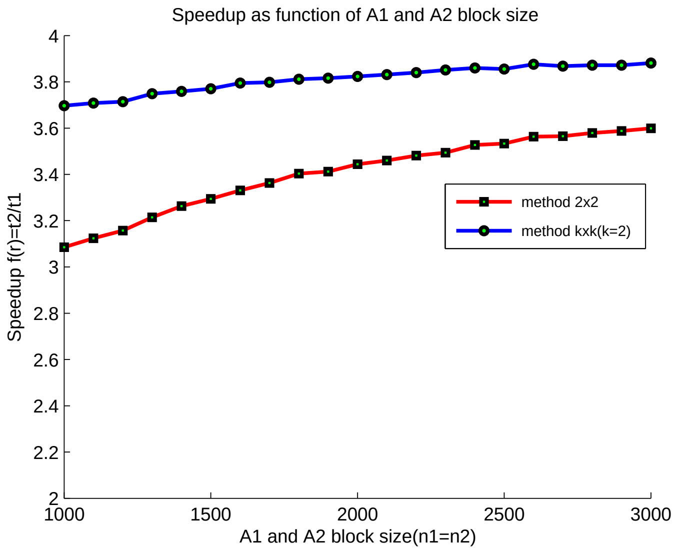

Figure 4. *Speedup* $f_{n_1}(50)$.

structure with overlapping), the Atenekeng-Grigori-Sosonkina procedure [2] can be used. Tests have been conducted on random matrices with dense blocks.

In future works, we will investigate methods which can involve parallel factorizations for k blocks, as proposed for $k = 2$ in this paper. At least, one possibility is to reorder the matrix into an arrowhead structure, which insures that the LU factors follows the same pattern.

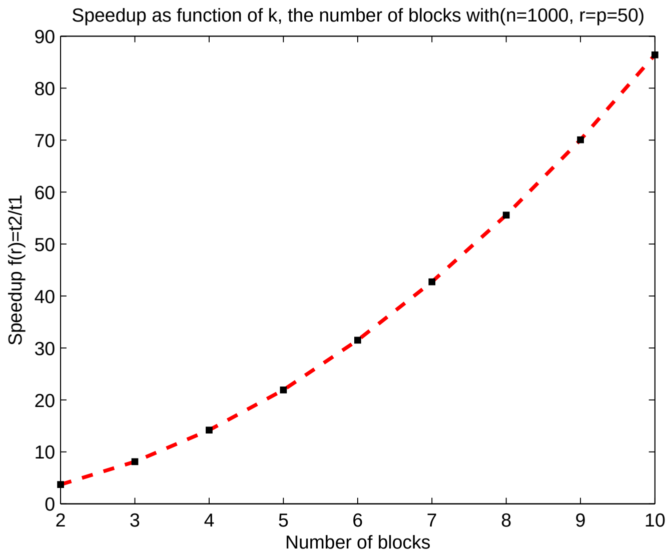

Figure 5. *Speedup* $f(r) = \frac{t_2}{t_1}$

Acknowledgement

This work was supported in part by CORUS II and in part by the projet LIRIMA, during the first and second authors visits within the research team SAGE at the INRIA Rennes-Bretagne Atlantique center.

8. References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, third edition, 1999.

[2] G. A. Atenekeng-Kahou, L. Grigori, and M. Sosonkina. A partitioning algorithm for block-diagonal matrices with overlap. *Parallel Computing*, 34:332 – 344, 2008.

[3] E. Kamgnia and B. Philippe. Counting eigenvalues in domains of the complex field. Technical Report RR-7770, INRIA, Oct. 2011.

[4] G. Karypis and V. Kumar. METIS - unstructured graph partitioning and sparse matrix ordering system, version 2.0. University of Minnesota, CS Dept, Technical report, 1995.