Hybridizing Nested Dissection and Halo Approximate Minimum Degree for Efficient Sparse Matrix Ordering*

François Pellegrini[1], Jean Roman[1], and Patrick Amestoy[2]

[1] LaBRI, UMR CNRS 5800, Université Bordeaux I & ENSERB
351, cours de la Libération, F-33405 Talence, France
{pelegrin|roman}@labri.u-bordeaux.fr
[2] ENSEEIHT-IRIT
2, rue Charles Camichel, 31071 Toulouse Cédex 7, France
Patrick.Amestoy@enseeiht.fr

Abstract. Minimum Degree and Nested Dissection are the two most popular reordering schemes used to reduce fill-in and operation count when factoring and solving sparse matrices. Most of the state-of-the-art ordering packages hybridize these methods by performing incomplete Nested Dissection and ordering by Minimum Degree the subgraphs associated with the leaves of the separation tree, but to date only loose couplings have been achieved, resulting in poorer performance than could have been expected. This paper presents a tight coupling of the Nested Dissection and Halo Approximate Minimum Degree algorithms, which allows the Minimum Degree algorithm to use exact degrees on the boundaries of the subgraphs passed to it, and to yield back not only the ordering of the nodes of the subgraph, but also the amalgamated assembly subtrees, for efficient block computations.
Experimental results show the performance improvement, both in terms of fill-in reduction and concurrency during numerical factorization.

1 Introduction

When solving large sparse linear systems of the form $Ax = b$, it is common to precede the numerical factorization by a symmetric reordering. This reordering is chosen in such a way that pivoting down the diagonal in order on the resulting permuted matrix PAP^T produces much less fill-in and work than computing the factors of A by pivoting down the diagonal in the original order (the fill-in is the set of zero entries in A that become non-zero in the factored matrix).

The two most classically-used reordering methods are Minimum Degree and Nested Dissection. The Minimum Degree algorithm [19] is a local heuristic that performs its pivot selection by selecting from the graph a node of minimum degree. The Nested Dissection algorithm [8] is a global heuristic recursive algorithm which computes a vertex set S that separates the graph into two parts A and B, ordering S last. It then proceeds recursively on parts A and B until their

* This work is supported by the French *Commissariat à l'Énergie Atomique* CEA/CESTA under contract No. 7V1555AC, and by the GDR ARP of the CNRS.

sizes become smaller than some threshold value. This ordering guarantees that no non zero term can appear in the factorization process between unknowns of A and unknowns of B.

The Minimum Degree algorithm is known to be a very fast and general purpose algorithm, and has received much attention over the last three decades (see for example [1, 9, 16]). However, the algorithm is intrinsically sequential, and very little can be theoretically proven about its efficiency.

On the other hand, many theoretical results have been carried out on Nested Dissection ordering [5, 15], and its divide and conquer nature makes it easily parallelizable. In practice, Nested Dissection produces orderings which, both in terms of fill-in and operation count, compare well to the ones obtained with Minimum Degree [11, 13, 17]. Moreover, the elimination trees induced by nested dissection are broader, shorter, and better balanced, and therefore exhibit much more concurrency in the context of parallel Cholesky factorization [4, 6, 7, 11, 17, 18, and included references].

Due to their complementary nature, several schemes have been proposed to hybridize the two methods [12, 14, 17]. However, to our knowledge, only loose couplings have been achieved: incomplete Nested Dissection is performed on the graph to order, and the resulting subgraphs are passed to some Minimum Degree algorithm. This results in the fact that the Minimum Degree algorithm does not have exact degree values for all of the boundary vertices of the subgraphs, leading to a misbehavior of the vertex selection process.

In this paper, we propose a tight coupling of the Nested Dissection and Minimum Degree algorithms, that allows each of them to take advantage of the information computed by the other. First, the Nested Dissection algorithm provides exact degree values for the boundary vertices of the subgraphs passed to the Minimum Degree algorithm (called *Halo* Minimum Degree since it has a partial visibility of the neighborhood of the subgraph). Second, the Minimum Degree algorithm returns the assembly tree that it computes for each subgraph, thus allowing for supervariable amalgamation, in order to obtain column-blocks of a size suitable for BLAS3 block computations.

The rest of the paper is organized as follows. Section 2 describes the metrics we use to evaluate the quality of orderings. Section 3 presents the principle of our Halo Minimum Degree algorithm, and evaluates its efficiency. Section 4 outlines the amalgamation process that is carried out on the assembly trees, and validates its interest in the context of block computations. In the concluding section, we show that relatively high speed-ups can be achieved during numerical factorization by using our ordering scheme.

2 Metrics

The quality of orderings is evaluated with respect to several criteria. The first one, called NNZ, is the overall number of non zero terms to store in the factored reordered matrix, including extra logical zeros that must be stored when block

storage is used. The second one, OPC, is the operation count, that is, the number of arithmetic operations required to factor the matrix. To comply with existing RISC processor technology, the operation count that we consider in this paper accounts for all operations (additions, subtractions, multiplications, divisions) required by sequential Cholesky factorization, except square roots; it is equal to $\sum_c n_c^2$, where n_c is the number of non-zeros of column c of the factored matrix, diagonal included. The third criterion is the size, in integer words, of the secondary storage used to index the array of non-zeros. It is defined as SS_b when block storage is used, and SS_c when column compressed storage is used.

All of the algorithms described in this paper have been integrated into version 3.3 of the SCOTCH static mapping and sparse matrix ordering software package [17] developed at the LaBRI, which has been used for all of our tests. All the experiments were run on a 332MHz PowerPC 604E-based RS6000 machine with 512 Mb of main memory. The parallel experiments reported in the conclusion were run on an IBM-SP2 with 66 MHz Power2 thin nodes having 128 MBytes of physical memory and 512 MBytes of virtual memory each.

Table 1. Description of our test problems. NNZ_A is the number of extra-diagonal terms in the triangular matrix after it has been symmetrized, whenever necessary.

Name	Columns	NNZ_A	Description
144	144649	1074393	3D finite element mesh
598A	110971	741934	3D finite element mesh
AATKEN	42659	88237	
AUTO	448695	3314611	3D finite element mesh
BCSSTK29	13992	302748	3D stiffness matrix
BCSSTK30	28924	1007284	3D stiffness matrix
BCSSTK31	35588	572914	3D stiffness matrix
BCSSTK32	44609	985046	3D stiffness matrix
BMW3_2	227362	5530634	3D stiffness matrix
BMW7ST_1	141347	3599160	3D stiffness matrix
BRACK2	62631	366559	3D finite element mesh
CRANKSEG1	52804	5280703	3D stiffness matrix
CRANKSEG2	63838	7042510	3D stiffness matrix
M14B	214765	3358036	3D finite element mesh
OCEAN	143437	409593	3D finite element mesh
OILPAN	73752	1761718	3D stiffness matrix
ROTOR	99617	662431	3D finite element mesh
TOOTH	78136	452591	3D finite element mesh

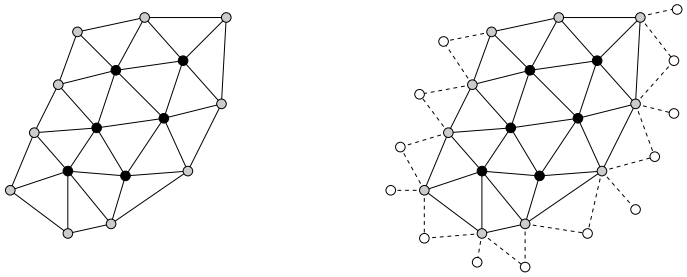
3 Halo Minimum Degree Ordering

In the classical hybridizations of Nested Dissection and Minimum Degree, incomplete Nested Dissection is performed on the original graph, and the subgraphs at the leaves of the separation tree are ordered using a minimum degree heuristic. The switching from Nested Dissection to Minimum Degree is controlled by a

criterion which is typically a threshold on the size of the subgraphs. It is for instance the case for ON-METIS 4.0 [14], which switches from Nested Dissection to Multiple Minimum Degree at a threshold varying between 100 and 200 vertices, depending on the structure of the original graphs.

The problem in this approach is that the degrees of the boundary vertices of the subgraphs passed to the Minimum Degree algorithm are much smaller than they would be if the original graph were ordered using Minimum Degree only. Consequently, the Minimum Degree algorithm will tend to order boundary vertices first, creating a "skin" of low-degree vertices that will favor fill-in between separator vertices and interior vertices, and across neighbors of the boundary vertices.

To avoid such behavior, the Minimum Degree algorithm must work on subgraphs that have real vertex degrees on their boundary vertices, to compute good orderings of all of the subgraph vertices. In order to do that, our Nested Dissection algorithm adds a topological description of the immediate surroundings of the subgraph with respect to the original graph (see figure 1). This additional information, called *halo*, is then exploited by our modified Approximate Minimum Degree [1] algorithm to reoder subgraph nodes only. The assembly tree (that is, the tree of supervariables) of both the subgraph and halo nodes is then built, for future use (see section 4). We refer to this modified approximate minimum degree as Halo-AMD, or HAMD.



a. Without halo. **b.** With halo.

Fig. 1. Knowledge of subgraph vertex degrees by the Minimum Degree algorithm without and with halo. Black vertices are subgraph internal vertices, grey vertices are subgraph boundary vertices, and white vertices are halo vertices, which are seen by the Halo Minimum Degree algorithm but do not belong to the subgraph.

Since it is likely that, in most cases (and especially for finite-element graphs), all of the vertices of a given separator will be linked by paths of smaller reordered numbers in the separated subsets, leading to completely filled diagonal blocks, we order separators using a breadth-first traversal of each separator subgraph from a pseudo-peripherial node [10], which proves to reduce the number of extra-diagonal blocks. Hendrickson and Rothberg report [12] that performing Minimum Degree to reorder last the subgraph induced by all of the separators found in the Nested Dissection process can lead to significant NNZ and OPC

improvement, especially for stiffness matrices. This hybridization type could also be investigated using our HAMD algorithm, but might be restricted to sequential solving, since breaking the separator hierarchy is likely to reduce concurrency in the elimination tree.

To experiment with the HAMD algorithm, we have run our orderer on several test graphs (see table 1) with three distinct strategies. In the first one, we perform graph compression [3, 12] whenever necessary (referred to as CP), then multi-level nested dissection (ND) with a subgraph size threshold of 120 to match ON-METIS's threshold, and finally the Approximate Minimum Degree of [1] (AMD, see table 3). In the second strategy, we replace the AMD algorithm by our halo counterpart, HAMD (table 2). In the third one, AMD is used to reorder the whole graph (see table 4).

Table 2. Ordering results with CP+ND+HAMD, with threshold 120. Bold means best over tables 2 to 5.

Name	NNZ	OPC	SSb
144	**4.714472e+07**	**5.663156e+10**	1.961818e+06
598A	**2.604285e+07**	**1.877590e+10**	1.491780e+06
AATKEN	3.742874e+07	1.891992e+11	**6.657630e+05**
AUTO	**2.214524e+08**	**4.557806e+11**	6.515154e+06
BCSSTK29	**1.580533e+06**	**3.347719e+08**	3.683400e+04
BCSSTK30	**4.339720e+06**	**1.185672e+09**	**6.619000e+04**
BCSSTK31	4.389614e+06	1.234546e+09	**1.893440e+05**
BCSSTK32	**5.484974e+06**	1.292770e+09	**1.251660e+05**
BMW3_2	**4.564966e+07**	3.205625e+10	5.046610e+05
BMW7ST_1	**2.471148e+07**	1.133335e+10	**2.777360e+05**
BRACK2	**5.869791e+06**	**1.803800e+09**	8.194700e+05
CRANKSEG1	**3.148010e+07**	**3.004003e+10**	**1.062940e+05**
CRANKSEG2	**4.196821e+07**	4.598694e+10	**1.231160e+05**
M14B	**6.262970e+07**	**6.113363e+10**	2.926878e+06
OCEAN	2.047171e+07	1.301932e+10	2.093677e+06
OILPAN	9.470356e+06	3.333825e+09	**1.016020e+05**
ROTOR	**1.569899e+07**	**9.277418e+09**	1.342506e+06
TOOTH	**1.041380e+07**	**6.288746e+09**	1.026880e+06

Using HAMD instead of AMD for ordering the subgraphs reduces both the NNZ and the OPC (see tables 2 and 3). For a threshold value of 120, gains for our test graphs range from 0 to 10 percent, with an average of about 5 percent.

When the threshold value increases (see figures 2.a and figures 2.b), gains tend to increase, as more space is left for HAMD to outperform AMD, and then eventually decrease to zero since AMD and HAMD are equivalent on the whole graph. However, for most graphs in our collection, the NNZ and OPC computed by CP+ND+HAMD increase along with the threshold. Therefore, the threshold should be kept small.

Table 3. Ordering results with CP+ND+AMD, with threshold 120, compared to CP+ND+HAMD (table 2). Bold means better than CP+ND+HAMD.

Name	NNZ	%	OPC	%	SSb
144	4.768610e+07	1.15	5.672072e+10	0.16	**1.830258e+06**
598A	2.641262e+07	1.42	1.883037e+10	0.29	**1.439660e+06**
AATKEN	3.760711e+07	0.48	1.892114e+11	0.01	9.088100e+05
AUTO	2.228161e+08	0.62	4.560077e+11	0.05	**5.899395e+06**
BCSSTK29	1.705363e+06	7.90	3.559825e+08	6.34	**2.372500e+04**
BCSSTK30	4.714555e+06	8.64	1.295689e+09	9.28	6.655200e+04
BCSSTK31	4.676407e+06	6.53	1.291574e+09	4.62	1.897200e+05
BCSSTK32	5.958303e+06	8.63	1.410264e+09	9.09	1.265890e+05
BMW3_2	4.896745e+07	7.27	3.307174e+10	3.17	**5.039480e+05**
BMW7ST_1	2.687398e+07	8.75	1.200949e+10	5.97	2.866670e+05
BRACK2	6.056172e+06	3.18	1.822081e+09	1.01	**7.729910e+05**
CRANKSEG1	3.271575e+07	3.93	3.114353e+10	3.67	1.109120e+05
CRANKSEG2	4.372140e+07	4.18	4.760978e+10	3.53	1.277210e+05
M14B	6.349348e+07	1.38	6.128062e+10	0.24	**2.709220e+06**
OCEAN	2.072061e+07	1.22	1.304039e+10	0.16	**1.918925e+06**
OILPAN	1.041493e+07	9.97	3.543230e+09	6.28	1.085550e+05
ROTOR	1.608042e+07	2.43	9.322391e+09	0.48	**1.338132e+06**
TOOTH	1.064755e+07	2.24	6.311166e+09	0.36	**9.697280e+05**

Table 4. Ordering results with (H)AMD alone, compared to CP+ND+HAMD (table 2). Bold means better than CP+ND+HAMD.

Name	NNZ	%	OPC	%	SSb
144	9.393420e+07	99.25	2.406120e+11	324.87	2.963011e+06
598A	4.592155e+07	76.33	6.224629e+10	231.52	2.293154e+06
AATKEN	**2.589400e+07**	-30.82	**1.202562e+11**	-36.44	9.855260e+05
AUTO	5.386292e+08	143.23	2.701255e+12	492.67	1.044489e+07
BCSSTK29	1.786234e+06	13.01	4.714500e+08	40.83	3.819000e+04
BCSSTK30	**3.582124e+06**	-11.24	**9.446337e+08**	-20.33	7.488900e+04
BCSSTK31	5.568769e+06	26.86	2.909362e+09	135.66	2.123520e+05
BCSSTK32	**4.989134e+06**	-9.04	**9.535496e+08**	-26.24	1.404360e+05
BMW3_2	5.071095e+07	11.09	4.721008e+10	47.27	5.406270e+05
BMW7ST_1	2.613003e+07	5.74	1.466054e+10	29.36	2.983190e+05
BRACK2	7.286548e+06	6.47	3.057435e+09	69.50	8.804010e+05
CRANKSEG1	4.036152e+07	24.14	5.256665e+10	74.99	1.396250e+05
CRANKSEG2	5.866411e+07	39.78	9.587950e+10	108.49	1.685880e+05
M14B	1.105011e+08	76.44	1.880238e+11	207.56	4.030059e+06
OCEAN	3.306994e+07	61.54	3.557219e+10	173.23	2.322347e+06
OILPAN	1.013450e+07	7.01	4.183709e+09	25.49	1.104820e+05
ROTOR	2.475696e+07	57.70	2.491224e+10	168.53	1.615861e+06
TOOTH	1.613290e+07	54.92	1.810155e+10	187.84	1.136381e+06

Table 5. Ordering results with ON-METIS 4.0, compared to CP+ND+HAMD (table 2). Bold means better than CP+ND+HAMD.

Name	NNZ	%	OPC	%	SSc
144	5.006260e+07	6.19	6.288269e+10	11.04	3.020012e+06
598A	2.670144e+07	2.53	1.945231e+10	3.60	2.119415e+06
AATKEN	**3.297272e+07**	-11.91	**1.545334e+11**	-18.32	5.789739e+06
AUTO	2.358388e+08	6.50	5.140633e+11	12.79	1.023613e+07
BCSSTK29	1.699841e+06	7.55	3.502626e+08	4.63	1.753670e+05
BCSSTK30	4.530240e+06	4.39	1.216158e+09	2.57	3.120820e+05
BCSSTK31	**4.386990e+06**	-0.06	**1.178024e+09**	-4.58	4.363570e+05
BCSSTK32	5.605579e+06	2.20	**1.230208e+09**	-4.84	5.073400e+05
BMW3_2	4.593307e+07	0.62	**2.815020e+10**	-12.18	2.619580e+06
BMW7ST_1	2.558782e+07	3.55	**1.097226e+10**	-3.19	1.600835e+06
BRACK2	6.025575e+06	2.65	1.851941e+09	2.67	9.413400e+05
CRANKSEG1	3.343857e+07	6.22	3.284621e+10	9.34	7.667890e+05
CRANKSEG2	4.320028e+07	2.94	**4.521963e+10**	-1.67	9.279760e+05
M14B	6.652286e+07	6.22	6.791640e+10	11.09	4.399879e+06
OCEAN	**1.950496e+07**	-4.72	**1.178200e+10**	-9.50	2.247596e+06
OILPAN	**9.064734e+06**	-4.28	**2.750608e+09**	-17.49	6.773740e+05
ROTOR	1.627034e+07	3.64	**9.460553e+09**	1.97	1.702922e+06
TOOTH	1.094157e+07	5.07	7.153447e+09	13.75	1.247352e+06

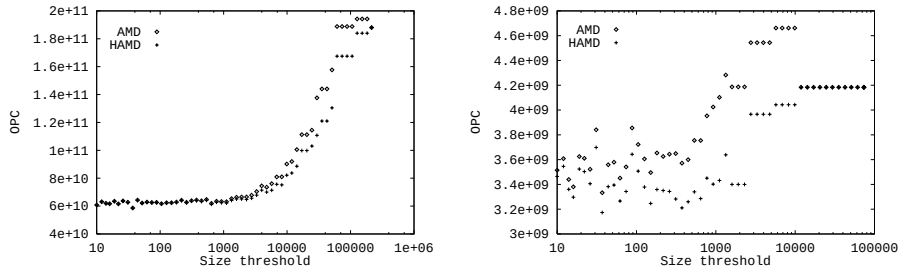
Two classes of graphs arise in all of these experiments. The first one, containing finite-element-like meshes, is very stable with respect to threshold (see figure 2.a). The second one, which includes stiffness matrices, has a very unstable behavior for small threshold values (see figure 2.b). However, for every given threshold, the ordering produced by HAMD always outperforms the ordering produced by AMD. Due to the small threshold used, the orderings computed by our CP+ND+HAMD algorithm are only slightly better on average than the ones computed by ON-METIS 4.0, although the latter does not use halo information (see table 5). This is especially true for stiffness matrices.

These experiments also confirm the current overall superiority of hybrid ordering methods against pure Minimum Degree methods (see table 4); concurring results have been obtained by other authors [11, 12, 13]. As a concluding remark, let us note that using block secondary storage almost always leads to much smaller secondary structures than when using column compressed storage (see tables 2 and 5).

4 Amalgamating supervariables

In order to perform efficient factorization, BLAS3-based block computations must be used with blocks of sufficient sizes. Therefore, it is necessary to amalgamate columns so as to obtain large enough blocks while keeping fill-in low.

Since, in order for the HAMD algorithm to behave well, the switching between the ND and HAMD methods must occur when subgraphs are large enough, the



a. For graph M14B.

b. For graph OILPAN.

Fig. 2. Values of OPC obtained with CP+ND+AMD and CP+ND+HAMD for various threshold values, for graphs M14B and OILPAN.

amalgamation process will only take place for the columns ordered by HAMD, as separators are of sufficient sizes. The amalgamation criterion that we have chosen is based on three parameters cmin, cmax, and frat: a leaf of the assembly tree is merged to its father either if it has less columns than some threshold value cmin or if it induces less fill-in than a user-defined fraction of the surface of the merged block frat, provided that the merged block has less columns than some threshold value cmax.

The partition of the original graph into supervariables is achieved by merging the partition of the separators and the partition of the supervariables amalgamated in each subgraph. One can therefore perform an efficient block symbolic factorization in quasi-linear space and time complexities (the linearity is proven in the case of complete Nested Dissection [5]).

To show the impact of supervariable amalgamation, we have run our orderer with strategy CP+ND+HAMD, for size thresholds of 120 and 1111 (chosen to be about ten times the first one), respectively with no amalgamation at all (cmax = 0) and with a minimum desired column block size of cmin = 16; OPC results are summarized in table 6.

The amalgamation process induces a slight increase of NNZ and OPC (less than 15 and 5 percent respectively on average for large graphs), with the benefit of a dramatic reduction of the size of the secondary storage (of more than 70 percent on average). This reduction is not important in terms of memory occupation, since the amount of secondary storage is more than one order of magnitude less than NNZ, but it shows well that efficient amalgamation can be performed only locally on the subgraphs, at a cost much lower than if global amalgamation were performed in a post-processing phase.

The impact of amalgamation on NNZ and OPC is equivalent for the two threshold values, i.e. for column block sizes of about 1 to 10 percent of the subgraph sizes. This leaves some freedom to set amalgamation parameters according to the desired BLAS effects, independently of the Nested Dissection threshold.

Table 6. OPC for CP+ND+HAMD, for thresholds 120 and 1111, and desired minimum column block widths of 0 and 16, compared to case $t = 120$ and $cmax = 0$ (table 3).

Name	t=120, cmin=16	%	t=1111, cmax=0	%	t=1111, cmin=16	%
144	5.697046e+10	0.60	5.861587e+10	3.50	5.898574e+10	4.16
598A	1.900982e+10	1.25	1.918044e+10	2.15	1.942980e+10	3.48
AATKEN	1.892588e+11	0.03	1.905714e+11	0.73	1.910815e+11	0.99
AUTO	4.569721e+11	0.36	**4.549330e+11**	-0.19	4.561968e+11	0.09
BCSSTK29	3.554720e+08	6.18	3.404095e+08	1.68	3.591577e+08	7.28
BCSSTK30	1.478795e+09	24.72	1.396119e+09	17.75	1.704818e+09	43.78
BCSSTK31	1.434832e+09	16.22	1.400022e+09	13.40	1.619635e+09	31.19
BCSSTK32	1.787947e+09	38.30	1.353213e+09	4.68	1.884575e+09	45.78
BMW3_2	3.639430e+10	13.53	3.110278e+10	-2.97	3.565307e+10	11.22
BMW7ST_1	1.407066e+10	24.15	1.201388e+10	6.00	1.495933e+10	31.99
BRACK2	1.888913e+09	4.72	1.931003e+09	7.05	2.024382e+09	12.23
CRANKSEG1	3.293365e+10	9.63	4.055242e+10	34.99	4.407524e+10	46.72
CRANKSEG2	5.046633e+10	9.74	6.102002e+10	32.69	6.636002e+10	44.30
M14B	6.164424e+10	0.84	6.214652e+10	1.66	6.270303e+10	2.57
OCEAN	1.321423e+10	1.50	**1.271274e+10**	-2.35	1.295883e+10	-0.46
OILPAN	5.130055e+09	53.88	3.431957e+09	2.94	5.380494e+09	61.39
ROTOR	9.440305e+09	1.76	9.893124e+09	6.64	1.006988e+10	8.54
TOOTH	6.388041e+09	1.58	6.339282e+09	0.80	6.461720e+09	2.75

5 Conclusion and perspectives

In this paper, we have presented a tight coupling of the Nested Dissection and Approximate Minimum Degree algorithms which always improves the quality of the produced orderings.

Table 7. Elapsed time (in seconds) for the numerical factorization phase of the MUMPS code for matrix OILPAN. Uniprocessor timings are based on CPU time.

Procs	AMD	CP+ND+AMD	CP+ND+HAMD
1	41	40	37
8	9.9	8.1	6.9
16	7.6	5.5	4.9
24	7.2	4.9	4.4

This work is still in progress, and finding exactly the right criterion to switch between the two methods is still an open question. As a matter of fact, we have evidenced that performing Nested Dissection too far limits the benefits of the Halo-AMD algorithm. On the opposite, switching too early limits the potentiality of concurrency in the elimination trees.

As a concluding remark, we present in table 7 the influence of the ordering scheme on the performance of the numerical factorization. The multifrontal code MUMPS (see [2], partially supported by the PARASOL project, EU ESPRIT IV LTR project 20160) was used to obtain these results on an IBM-SP2. Our

ordering scheme provides well balanced assembly trees (the maximum speed-ups obtained with AMD only and with CP+ND+HAMD are respectively 5.69 and 8.41).

References

[1] P. Amestoy, T. Davis, and I. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. and Appl.*, 17:886–905, 1996.

[2] P. Amestoy, I. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *to appear in special issue of Comput. Methods in Appl. Mech. Eng. on domain decomposition and parallel computing*, 1998.

[3] C. Ashcraft. Compressed graphs and the minimum degree algorithm. *SIAM J. Sci. Comput.*, 16(6):1404–1411, 1995.

[4] C. Ashcraft, S. Eisenstat, J. W.-H. Liu, and A. Sherman. A comparison of three column based distributed sparse factorization schemes. In *Proc. Fifth SIAM Conf. on Parallel Processing for Scientific Computing*, 1991.

[5] P. Charrier and J. Roman. Algorithmique et calculs de complexité pour un solveur de type dissections emboîtées. *Numerische Mathematik*, 55:463–476, 1989.

[6] G. A. Geist and E. G.-Y. Ng. Task scheduling for parallel sparse Cholesky factorization. *International Journal of Parallel Programming*, 18(4):291–314, 1989.

[7] A. George, M. T. Heath, J. W.-H. Liu, and E. G.-Y. Ng. Sparse Cholesky factorization on a local memory multiprocessor. *SIAM Journal on Scientific and Statistical Computing*, 9:327–340, 1988.

[8] A. George and J. W.-H. Liu. *Computer solution of large sparse positive definite systems.* Prentice Hall, 1981.

[9] A. George and J. W.-H. Liu. The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31:1–19, 1989.

[10] N. E. Gibbs, W. G. Poole, and P. K. Stockmeyer. A comparison of several bandwidth and profile reduction algorithms. *ACM Trans. Math. Soft.*, 2:322–330, 1976.

[11] A. Gupta, G. Karypis, and V. Kumar. Scalable parallel algorithms for sparse linear systems. In *Proc. Stratagem'96, Sophia-Antipolis*, pages 97–110, July 1996.

[12] B. Hendrickson and E. Rothberg. Improving the runtime and quality of nested dissection ordering. *SIAM J. Sci. Comput.*, 20(2):468–489, 1998.

[13] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. TR 95-035, University of Minnesota, June 1995.

[14] G. Karypis and V. Kumar. METIS – *A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices – Version 4.0.* University of Minnesota, September 1998.

[15] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM Journal of Numerical Analysis*, 16(2):346–358, April 1979.

[16] J. W.-H. Liu. Modification of the minimum-degree algorithm by multiple elimination. *ACM Trans. Math. Software*, 11(2):141–153, 1985.

[17] F. Pellegrini and J. Roman. Sparse matrix ordering with SCOTCH. In *Proceedings of HPCN'97, Vienna, LNCS 1225*, pages 370–378, April 1997.

[18] R. Schreiber. Scalability of sparse direct solvers. Technical Report TR 92.13, RIACS, NASA Ames Research Center, May 1992.

[19] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *J. Proc. IEEE*, 55:1801–1809, 1967.