LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

An experimental analysis of three heuristics for bandwidth reduction to speed up the incomplete Cholesky-conjugate gradient method

Sanderson L. Gonzaga de Oliveira
Universidade Federal de Lavras - UFLA
Lavras - MG
sanderson@ufla.br

ABSTRACT

A previous publications has reviewed a large number of heuristics for bandwidth reduction. Based on this experience, this work selects three heuristics as the most promising low-cost methods for solving the problem. Thus, this paper applies state-of-the-art low-cost heuristics for bandwidth reduction along with an incomplete Cholesky factorization and using this as a preconditioner for the conjugate gradient method. Numerical results delivered by these three heuristics for bandwidth reduction confirm the effectiveness of them for preprocessing linear systems arising from nine application areas.

KEYWORDS. Bandwidth reduction. Reordering algorithms. Graph labeling.

https://proceedings.science/p/106951?lang=pt-br

1. Introduction

A fundamental step in several applications in science and engineering is the resolution of large sparse linear systems in the form $Ax = b$ where A is a sparse $n \times n$ symmetric matrix, x is the unknown n-vector solution, and b is a known n-vector. It is typically the step of the simulation that demands a high computational cost.

The principal source of the problems with matrices arises from the discretization of elliptic or parabolic partial differential equations [Benzi, 2002]. The methods of finite elements, finite differences, and finite volumes are some of the most common numerical problem-solving methods related to physical phenomena governed by partial differential equations [Gonzaga de Oliveira et al., 2018]. These linear systems also originate from application fields that are not modeled by partial differential equations, such as optimization and economic problems. A large amount of memory and a high processing costs are required to store and solve these large-scale linear systems [Gonzaga de Oliveira et al., 2018].

Cache coherence is significant when designing an algorithm. An appropriate vertex labeling (of the graph associated with matrix A) is desirable for the low-cost solution of large and sparse linear systems. A proper vertex labeling generates the corresponding coefficient matrix A with a narrow bandwidth [Gonzaga de Oliveira et al., 2018], and this does not depend on the storage scheme chosen.

The use of a heuristic for bandwidth reduction is an appropriate alternative to achieve a sequence of graph vertices with cache coherence. Thereby, heuristics for bandwidth reduction are employed to attain low execution times for solving large sparse linear systems [Gonzaga de Oliveira et al., 2018].

Let $A = [a_{ij}]$ be an $n \times n$ symmetric matrix associated with an undirected graph $G = (V, E)$ composed of a set of vertices V and a set of edges E. The bandwidth of line i is $\beta_i(A) = i - \min_{1 \leq j < i}[j : a_{ij} \neq 0]$. The bandwidth of a matrix A is defined as $\beta(A) = \max_{1 \leq i \leq n}[\beta_i(A)]$. Equivalently, the bandwidth of G for a vertex labeling $S = \{s(v_1), s(v_2), \cdots, s(v_{|V|})\}$ (i.e. a bijective mapping from V to the set $\{1, 2, \cdots, |V|\}$) is $\beta(G) = \max_{v \in V}[(\{v, u\} \in E)\ |s(v) - s(u)|]$. The bandwidth minimization problem is NP-hard [Papadimitriou, 1976].

When a matrix is symmetric and strictly diagonally dominant, a prominent method for solving large sparse linear systems is the conjugate gradient method (CGM) [Hestenes e Stiefel, 1952; Lanczos, 1952]. The structure of matrix A, the accuracy to be attained, and the characteristics of the preconditioning applied influence the execution costs of the preconditioned conjugate gradient method.

The most relevant quality of a reordering algorithm in conjunction with the preconditioner is to provide a matrix A with a reduced condition number (or equivalently better cluster of eigenvalues), and that induces cache coherence when using a data structure that does not depend on the bandwidth of matrix A, as is the case in the experiments presented in this paper. Thereby, even when the linear system is composed of multiple right-hand side vectors, the resulting linear system is easier to solve than the original linear system is [Gonzaga de Oliveira et al., 2018].

This paper analyzes experimentally three heuristics for bandwidth reduction with the objective of reducing execution times of the preconditioned conjugate gradient method. Symilar to the approach employed in a previous publication [Gonzaga de Oliveira et al., 2018], we apply the incomplete Cholesky factorization with zero fill-in as a preconditioner of the conjugate gradient method [Ajiz e Jennings, 1984].

The aforementioned publication [Gonzaga de Oliveira et al., 2018] has evaluated several heuristics for bandwidth reduction with the objective of reducing the zero-fill incomplete

https://proceedings.science/p/106951?lang=pt-br

Cholesky-preconditioned conjugate gradient method when applied to linear systems arising from various application areas. Among them, the previous publication [Gonzaga de Oliveira et al., 2018] has reported three heuristics as the most promising low-cost heuristics for bandwidth (the reverse Cuthill-McKee method [George, 1971], the resulting heuristic from a genetic programming approach proposed by Koohestani e Poli [2011] (KP-band), and the reverse breadth-first search procedure (RBFS). The George-Liu algorithm [George e Liu, 1979] provides the starting vertex for the three heuristics. This paper evaluates these three low-cost heuristics for bandwidth reduction when employed to reduce executions costs of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to 25 linear systems arising from nine application fields, including two application areas that the previous publication has not covered.

The remainder of this manuscript is structured as follows. Section 2 reports the heuristics for bandwidth reduction evaluated. Section 3 describes how this computational experiment was conducted. Section 4 discusses the results. Finally, Section 5 addresses the conclusions.

2. Reordering algorithms evaluated

The reverse Cuthill–McKee (RCM) method [George, 1971] labels the vertices of a graph $G(V, E)$ in order of increasing distance from a given pseudoperipheral vertex v. Specifically, it labels vertices with the same distance from the vertex v in order of increasing degree. Finally, the ordering is reversed. Therefore, the final label of vertex v is $|V|$.

Algorithm 1 shows a pseudocode of the Reverse Cuthill-McKee method. The method receives a graph $G = (V, E)$ and a pseudoperipheral vertex $v \in V$. The method labels the vertices of a graph (see line 8) with the same distance from the pseudoperipheral vertex v in order of increasing degree (see line 6). The method reverses the final ordering. Therefore, the final label of vertex v is $|V|$ at line 2. Thus, Algorithm 1 begins the numbering with the label $s(|V|)$ (see lines 2–4 in Algorithm 1). Algorithm 1 returns the new labeling at line 10.

Input: a connected graph $G = (V, E)$; a vertex $v \in V$;
Output: a labeling $S = \{s(1), s(2), ..., s(|V|)\}$;

```
1  begin
2      s(|V|) ← v;
3      i ← |V|;
4      j ← |V|;
5      while (i > 0) do
6          foreach (vertex w ∈ Adj(s(j)) − {s(|V|), ..., s(i)}, in order of increasing
           degree) do
7              i ← i − 1;
8              s(i) ← w;
9          j ← j − 1;
10     return S;
```

Algorithm 1: The Reverse Cuthill-McKee method [George, 1971].

Ordering the vertices in a graph such as the RCM ordering partitions them into *level sets* according to the distance from the pseudoperipheral vertex v. Given a vertex $v \in V$, the level structure $\mathscr{L}(v)$ rooted at vertex v, with depth $\ell(v)$, is the partitioning $\mathscr{L}(v) = \{L_0(v), L_1(v), \ldots, L_{\ell(v)}(v)\}$ where $L_0(v) = \{v\}$ and $L_i(v) = Adj(L_{i-1}(v)) - \bigcup_{j=0}^{i-1} L_j(v)$, for $i = 1, 2, 3, \ldots, \ell(v)$, $\ell(v)$ is the *eccentricity* of vertex v, and $Adj(U) = \{w \in V : (u \in U \subseteq V)\ \{u, w\} \in E\}$.

The Reverse Cuthill-McKee method labels the adjacent vertices to a vertex v by sorting them in order of ascending degree whereas the KP-band heuristic labels the vertices using the formula $0.179492928171 \cdot \varsigma^3 + 0.292849834929 \cdot \varsigma^2 - 0.208926175433 \cdot n - 0.736485142138 \cdot n \cdot \varsigma - 1.77524579882 \cdot \varsigma - 1.75681383404$ where n is the dimension of matrix A and ς is the sum of degrees of vertices connected to the vertex v. The reverse breadth-first search procedure is the reversed order of the BFS procedure (see Gonzaga de Oliveira et al. [2018]).

The George–Liu algorithm begins with an arbitrary vertex v and builds its rooted level structure $\mathscr{L}(v)$. Then, it builds the rooted level structure $\mathscr{L}(u)$ of the vertex $u \in L_{\ell(v)}(v)$ with minimum degree. If $\ell(u) > \ell(v)$, then u is attributed to v (i.e., $v \leftarrow u$) as well as $\mathscr{L}(v) \leftarrow \mathscr{L}(u)$ is computed, and the process is repeated; otherwise, the process stops and v is the pseudoperipheral vertex found.

Algorithm 2 shows the George-Liu algorithm. The algorithm begins with an arbitrary vertex v (see line 2 in Algorithm 2) and builds its rooted level structure $\mathscr{L}(v)$ (see line 3). Then, the algoritm builds the rooted level structure $\mathscr{L}(u)$ of the vertex $u \in L_{\ell(v)}(v)$ with minimum degree (see lines 5 and 6). If $\ell(u) > \ell(v)$ (see line 7), then u is attributed to v as well as $\mathscr{L}(v) \leftarrow \mathscr{L}(u)$ is computed (see lines 8 and 9), and the process is repeated; otherwise, the process stops and v is the pseudoperipheral vertex found (see line 11 in Algorithm 2).

Input: graph $G = (V, E)$;
Output: pseudo-peripheral vertex $v \in V$;
1 **begin**
2 $v \leftarrow ArbitraryVertex(V)$;
3 $\mathscr{L}(v) \leftarrow BuscaEmLargura(v)$; `// build a rooted level structure`
4 **repeat**
5 $u \leftarrow MinimumDegreeVertex(L_{\ell(v)}(v))$;
6 $\mathscr{L}(u) \leftarrow BFS(u)$; `// build a rooted level structure`
7 **if** $\ell(u) > \ell(v)$ **then**
8 $v \leftarrow u$;
9 $\mathscr{L}(v) \leftarrow \mathscr{L}(u)$;
10 **until** $(u \neq v)$;
11 **return** v;

Algorithm 2: The George-Liu algorithm [George e Liu, 1979].

3. Description of the tests

We wrote the codes of the heuristics for bandwidth reduction in the C++ programming language using the g++ version 5.4.0 compiler, with the optimization flag -O3. This paper uses a data structure based on the Compress Row Storage, Compress Column Storage, and Skyline Storage Scheme data structures within the implementation of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method.

The experiments employed a precision of 10^{-16} to the conjugate gradient method using a double-precision floating-point arithmetic. Additionally, the conjugate gradient method stops when its number of iterations is the size of the instance if the algorithm does not achieve the defined precision.

The vectors b and x_0 do not define if the preconditioned conjugate gradient method converge or not, but can influence the number of iterations and the CPU time is the principal method

https://proceedings.science/p/106951?lang=pt-br

for evaluating the results of algorithms in our experiments. When the collection does not supply the right-hand side of a linear system, we generated a vector b with random entries in the range $(-1, 1)$. We initialized the vector x_0 with zeros. We used CPU time as the primary metric in the experiments.

The experiments use 25 linear systems arising from nine application areas contained in the Suiteparse matrix collection [Davis e Hu, 2011] to analyze the speedup of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method after computing heuristics for bandwidth reduction. The original labeling given in the matrix was used to evaluate the speed-ups (or speed-downs) of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method.

The workstations used in the execution of the simulations featured an Intel® Core™ i7-4790K (CPU 4 GHz, 8 MB Cache, 12 GB of main memory DDR3 1.6 GHz, termed M1 machine) and an Intel® Core™ i5-3570 (CPU 3.4 GHz, 6 MB Cache, 8 GB of main memory DDR3 1.333 GHz, termed M2 machine) (Intel; Santa Clara, CA, United States). The linear systems that are part of the simulations carried out on each machine were chosen arbitrarily. The Ubuntu 16.04 LTS 64-bit operating system with Linux kernel-version 3.19.0-31-generic (3.13.0-37-generic) on the M1 (M2) machine was used.

4. Results and analysis

Tables 1-6 show the instances used, the reordering algorithms applied, the results with respect to bandwidth reduction, and execution times, in seconds (s). The same tables present the number of iterations and the total execution time, in seconds, of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method.

The first line of each instance in Tables 1-6 shows the results for linear systems solved using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method without reordering ("—") the rows and columns of the linear system. With this result, one can verify the speed-up (or speed-down) of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method provided by employing a heuristic for bandwidth reduction.

The last columns in Tables 1-6 show the speed-up (or speed-down) of the solver provided by employing a heuristic for bandwidth reduction. We calculate the speedup (or speed-down) as the time of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method without reordering divided by the total execution time of employing a heuristic for bandwidth reduction before executing the zero-fill incomplete Cholesky-preconditioned conjugate gradient method. Numbers in boldface are the best results.

Tables 1 and 2 show the average results reached by the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to nine linear systems arising from structural problems. This table shows that the RBFS heuristic yielded the best results in three of these instances (*nasa2910*, *olafu*, and *raefsky4*) with a significant speedup. Additionally, this table shows that the KP-band and Reverse Cuthill-McKee heuristics yielded the best results when applied to the *nasa4704* and *Serena* (the largest linear system used in this computational experiment) instances, respectively.

Tables 1 and 2 also show that the use of a heuristic for bandwidth reduction increased the execution times of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method when applied to four linear systems. In particular, the zero-fill incomplete Cholesky-preconditioned conjugate gradient method converges in zero iteration when applied to the instance *nasa2146*. The 2-norm condition number $\frac{\max[svd(A)]}{\min(svd(A))}$ of the original matrix is small (1724) compared with other matrices evaluated (*nasa1824*: 1.89599e+06, *sts4098*: 2.17087e+08, *nasa2910*: 5.95967e+06, *raefsky4*: 3.1301e+13, *Pres_Poisson*: 2.03665e+06). Similarly, the zero-fill incomplete Cholesky-

Tabela 1: Resolution of four linear systems (originating from structural problems) using the zero-fill incomplete Cholesky-preconditioned conjugate gradient (ICCG) method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M1 machine.

Linear system	Heuristic	β	Heuristic t(s)	CGM t(s)	IC0 t(s)	ICCGM t(s)	ICCGM iter.	Speed-up/down
nasa1824	—	239	—	0.2	0.1819	0.2	1824	—
	RBFS	274	**0.001**	0.2	0.0252	0.2	1824	0.99
	RCM	263	0.003	0.2	0.0251	0.2	1824	0.97
	KP-band	420	0.002	0.2	0.0253	0.2	1824	0.95
nasa2146	—	**81**	—	0.1	0.0002	0.1	0	—
	RBFS	149	**0.001**	0.1	0.0002	0.1	0	0.96
	KP-band	148	0.004	0.1	0.0002	0.1	0	0.94
	RCM	148	0.005	0.1	0.0002	0.1	0	0.90
nasa2910	—	859	—	3.2	0.4900	3.69	2910	—
	RBFS	867	**0.003**	1.1	0.1600	1.24	2910	**2.97**
	KP-band	1025	0.011	1.1	0.1600	**1.23**	2910	2.96
	RCM	876	0.012	1.1	0.1600	1.24	2910	2.94
Serena	—	**81578**	—	251205.1	73488.0000	324693.0	692102	—
	RCM	84947	6.200	**0.7**	74010.0000	74011.0	1	**4.39**
	KP-band	85694	8.200	203383.0	**73218.0000**	276601.0	547008	1.17
	RBFS	84746	**3.100**	511550.7	73945.0000	585496.0	1391349	0.55

preconditioned conjugate gradient method converges in one iteration when applied to the instances *nasasrb* (see Table 2), *olafu* (see Table 2), *nd3k* (see Table 4), and *ex15* (see Table 3).

Table 3 shows the average results yielded by the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to linear systems arising from electromagnetics and computational fluid dynamics problems. This table confirms that the Reverse Cuthill-McKee and KP-band heuristics are promising methods to be applied to linear systems originating from computational fluid dynamics problems. The same table also shows that no heuristic for bandwidth reduction reduces the execution times of the iterative solver when applied to the instance *mhd3200b*.

Table 4 shows the average results delivered by the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to four linear systems arising from 2-D/3-D problems. This table confirms that a heuristic for bandwidth reduction does not favor the execution times of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method when applied to linear derived from this application area.

Table 5 shows the average results yielded by the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to linear systems arising from materials, computer graphics/vision, and thermal problems. This table confirms that the RBFS heuristic is a promising reordering heuristic to be applied to linear systems originating from materials problems. Additionally, the same table confirms that a heuristic for bandwidth reduction does not favor the execution times of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method when applied to linear systems derived from computer graphics/vision problems. Table 5 also shows that the Reverse Cuthill-McKEe and RBFS heuristics are promising methods to be applied to linear

https://proceedings.science/p/106951?lang=pt-br

Tabela 2: Resolution of five linear systems (originating from structural problems) using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M2 machine.

Linear system	Heuristic	β	Heuristic t(s)	CGM t(s)	IC0 t(s)	ICCGM t(s)	ICCGM iter.	Speed-up/down
sts4098	—	3323	—	**0.1**	**0.0553**	**0.2**	203	—
	RBFS	1343	**0.002**	0.2	0.0558	**0.2**	**199**	0.81
	RCM	**1165**	0.007	0.2	0.2352	0.4	843	0.43
	KP-band	1255	0.007	**0.1**	0.1678	1.3	4098	0.13
nasasrb	—	893	—	116.3	0.0464	116.4	1	—
	RBFS	**779**	**0.218**	**116.0**	0.0466	**116.1**	1	1.00
	KP-band	806	0.384	118.3	0.0458	118.4	1	0.98
	RCM	806	0.382	119.8	0.0461	119.8	1	0.97
olafu	—	593	—	9.1	0.0100	9.2	1	—
	RBFS	**537**	**0.065**	**8.7**	0.0100	**8.7**	1	**1.05**
	RCM	553	0.133	8.7	0.0200	8.7	1	1.04
	KP-band	573	0.138	9.2	0.0100	9.2	1	0.99
raefsky4	—	11786	—	139.3	15.5800	154.9	19779	—
	RBFS	**1028**	**0.083**	**15.1**	**3.4000**	**18.6**	**482**	**8.32**
	RCM	1070	0.158	49.9	15.9400	65.8	6940	2.35
	KP-band	1057	0.195	140.7	15.3100	156.1	19779	0.99
nasa4704	—	423	—	1.7	0.2200	1.9	4704	—
	KP-band	450	0.008	**1.6**	0.2200	**1.8**	4704	**1.05**
	RBFS	435	**0.003**	1.6	0.2200	**1.8**	4704	1.05
	RCM	**419**	0.011	1.6	0.2200	**1.8**	4704	1.02

systems arising from thermal problems.

Table 6 shows the average results obtained by the zero-fill incomplete Cholesky-preconditioned conjugate gradient method applied to linear systems arising from model reduction and combinatorial problems. This table shows that the Reverse Cuthill-McKee and KP-band (RBFS) heuristics are promising methods to be applied to linear systems arising from model reduction (combinatorial) problems.

Table 7 shows the characteristics (size, β, density, the most promising heuristic applied to the instance, application area, and the table that shows the results) of the linear systems used in the experiments. In particular, the RBFS heuristic yielded the best results when applied to the two linear systems arising from combinatorial problems.

Figure 1 shows the number of best results yielded by three heuristics for bandwidth reduction applied to 25 linear systems to reduce executions costs of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method. Original in the figure represents the original labeling of the linear system. The figure shows that heuristics for bandwidth reduction did not favor the execution time of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method when applied to ten linear systems used in this study.

https://proceedings.science/p/106951?lang=pt-br

Tabela 3: Resolution of four linear systems (originating from electromagnetics and computational fluid dynamics) using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M2 machine.

Linear system	Heuristic	β	Heuristic	CGM	IC0	ICCGM		Speed-
			t(s)	t(s)	t(s)	iter.	t(s)	up
mhd3200b	—		—	0.03	0.001	8	0.03	—
	KP-band	8	0.0017	0.03	0.007	68	0.04	0.83
	RBFS	6	**0.0003**	0.03	0.014	146	0.04	0.73
	RCM	**5**	0.0016	0.03	0.061	620	0.09	0.35
ex9	—	85	—	1.00	0.142	3363	1.20	—
	KP-band	148	0.0070	**0.30**	**0.141**	**1122**	**0.50**	**2.52**
	RCM	148	0.0080	1.00	0.143	3363	1.10	1.04
	RBFS	**144**	0.0060	1.00	0.143	3363	1.20	1.01
ex15	-	67	—	0.30	0.002	1	0.30	—
	KP-band	**83**	0.0080	0.30	0.001	1	0.30	1.04
	RCM	**83**	0.0080	0.40	0.001	1	0.40	0.89
	RBFS	85	**0.0030**	0.40	0.001	1	0.40	0.89
Pres_Poisson	—	12583	—	5.10	1.501	349	6.60	—
	RCM	**289**	0.0810	**4.70**	**0.615**	**159**	**5.30**	**1.21**
	KP-band	365	0.0960	4.90	0.743	175	5.60	1.15
	RBFS	331	**0.0490**	4.90	0.785	186	5.70	1.14

Tabela 4: Resolution of three linear systems originating from 2-D/3-D problems using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M1 machine.

Linear system	Heuristic	β	Heuristic	CGM	IC0	ICCGM		Speed-
			t(s)	t(s)	t(s)	t(s)	iter.	down
nd3k	—	8342	—	**6.7**	0.02	**6.7**	1	—
	KP-band	**2491**	0.27	8.3	0.02	8.3	1	0.79
	RBFS	29	**0.06**	8.9	0.02	8.9	1	0.75
	RCM	2713	0.29	9.2	0.02	9.3	1	0.70
fv1	—	**99**	—	0.3	0.01	0.3	17	—
	KP-band	196	0.01	0.3	0.01	0.3	19	0.97
	RCM	195	0.01	0.3	0.01	0.3	17	0.96
	RBFS	196	0.01	0.3	0.01	0.3	17	0.95
Dubcova1	—	16052	—	**0.9**	0.15	**1.0**	14	—
	RCM	**500**	0.02	1.4	0.06	1.4	79	0.71
	KP-band	502	0.02	1.4	0.06	1.4	83	0.71
	RBFS	511	**0.01**	1.4	0.06	1.5	82	0.70

Tabela 5: Resolution of five linear systems (originating from materials, computer graphics/vision, and thermal problems) using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M1 (*bloweybq*, *crystm03*, *bundle1*) and M2 (*TED_B*, *TED_B_UNSCALED*) machines.

Linear system	Heuristic	β	Heuristic t(s)	CGM t(s)	IC0 t(s)	ICCGM t(s)	ICCGM iter.	Speed-up/down
bloweybq	—	10000	—	0.3	0.160	0.4	501	—
	RBFS	9999	**0.002**	0.3	**0.050**	0.4	**153**	**1.34**
	RCM	**9995**	0.005	0.3	0.100	0.4	316	1.15
	KP-band	**9995**	0.005	0.3	0.110	0.4	312	1.10
crystm03	—	570	—	5.1	0.020	5.1	9	—
	RCM	**338**	0.041	5.0	0.020	5.0	9	1.00
	KP-band	**338**	0.043	5.0	0.040	5.0	25	1.00
	RBFS	358	**0.019**	5.1	0.020	5.1	9	1.00
bundle1	—	10461	—	**0.3**	0.040	**0.3**	18	—
	KP-band	6298	0.053	1.5	0.040	1.5	17	0.21
	RBFS	**5833**	**0.013**	2.5	**0.030**	2.6	**16**	0.13
	RCM	6127	0.056	2.5	**0.030**	2.6	**16**	0.12
TED_B	—	48	—	0.9	0.003	0.9	3	—
	RBFS	37	**0.003**	**0.6**	0.002	**0.6**	1	**1.60**
	RCM	38	0.014	**0.6**	0.002	**0.6**	1	1.54
	KP-band	**31**	0.011	0.8	**0.002**	0.8	1	1.12
TED_B_UNSCALED	—	48	—	0.9	0.003	0.9	3	—
	RCM	34	0.011	**0.5**	0.002	**0.5**	1	**1.72**
	RBFS	30	**0.007**	0.6	0.002	0.7	1	1.44
	KP-band	**31**	0.011	0.8	**0.002**	0.8	1	1.11

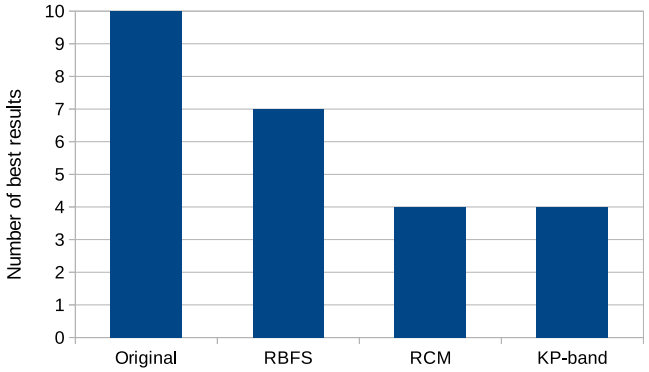


Figura 1: Number of best results obtained by heuristics for bandwidth reduction applied to 25 linear systems to reduce executions costs of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method. Original represents the original labeling of the linear system.

Tabela 6: Resolution of four linear systems (originating from model reduction and combinatorial problems) using the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using heuristics for bandwidth reduction. Executions performed on the M2 machine.

Linear system	Heuristic	β	Heuristic	CGM	IC0	ICCGM		Speed-
			t(s)	t(s)	t(s)	iter.	t(s)	up
LF10000	—	3	—	16	1.18	19998	17	—
	RCM	3	0.02	**15**	**1.10**	19998	**16**	**1.09**
	KP-band	3	0.02	16	1.12	19998	17	1.02
	RBFS	3	**0.01**	17	1.16	19998	18	0.97
LFAT5000	—	5	—	14	1.01	19994	15	—
	KP-band	4	0.01	**13**	1.02	19994	**14**	**1.05**
	RBFS	4	**0.01**	15	1.02	19994	16	0.91
	RCM	3	0.01	15	1.02	19994	16	0.91
Trefethen_20000b	—	—	—	6	0.05	10	6	—
	RBFS	**7278**	**0.04**	5	0.04	**9**	**5**	**1.08**
	KP-band	7376	0.07	**5**	0.06	10	**5**	1.07
	RCM	7578	0.07	5	0.04	10	5	1.04
Trefethen_20000	—	—	—	5	0.05	11	5	—
	RBFS	7379	**0.04**	**5**	**0.04**	11	**5**	**1.05**
	RCM	7373	0.06	5	0.05	11	5	0.98
	KP-band	**7344**	0.07	6	0.06	11	6	0.91

5. Conclusions

A previous review [Gonzaga de Oliveira et al., 2018] identified three methods as the most promising heuristics for bandwidth reduction to be used as a preprocessing step to reduce execution times of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method. In experiments using 25 linear systems (arising from nine application areas) composed of real symmetric and positive-definite matrices, the RBFS (in 7 instances), Reverse Cuthill-McKee [George e Liu, 1981] (in 4 instances), and KP-band [Koohestani e Poli, 2011] (in 4 instances) heuristics showed the best results in reducing the execution times of the zero-fill incomplete Cholesky-preconditioned conjugate gradient method. However, reordering the vertices did not favor the computing time of this linear system solver when applied to ten linear systems used (arising from structural, 2-D/3-D, materials, computer graphics/vision, and electromagnetics problems). The experiments conducted in this study confirm the results of a previous publication [Gonzaga de Oliveira et al., 2018].

Future studies will reveal whether these results may be extended to the preconditioned conjugate gradient method or even to other preconditioned iterative solvers. Also as a continuation of this work, we intend to implement and evaluate the following preconditioners: Algebraic Multigrid, ILU and variations, Successive Over-Relaxation (SOR), Symmetric SOR, and Gauss-Seidel. We plan to examine the effectiveness of the schemes with the computation of incomplete or approximate factorization-based preconditioners as well as approximate inverse preconditioners. These techniques shall be used as preconditioners of the (serial and parallel) conjugate gradient method to evaluate their computational performance along with (serial and parallel) heuristics for bandwidth reduction. Specifically, in parallel implementations, the choice of a preconditioner may also de-

Tabela 7: Twenty-five linear systems composed of symmetric and positive-definite matrices.

Instance	Size	β	Density	Best speedup	Kind	Table
nasa1824	1824	239	1.178%	—	structural problem	1
nasa2146	2146	81	1.569%			
nasa2910	2910	859	2.058%	RBFS		
Serena	1391349	81578	0.003%	RCM		2
sts4098	4098	3323	0.431%	—		
nasasrb	54870	893	0.089%			
olafu	16146	593	0.389%	RBFS		
raefsky4	19779	11786	0.337%			
nasa4704	4704	423	0.473%	KP-band		
mhd3200b	3200	18	0.179%	—	electromagnetics problem	3
ex9	3363	85	0.880%	KP-band	computational fluid dynamics	
ex15	6867	67	0.209%			
Pres_Poisson	14822	12583	0.326%	RCM		
nd3k	9000	8342	4.049%	—	2-D/3-D problem	4
fv1	9064	99	0.092%			
Dubcova1	16129	16052	0.097%			
bloweybq	10001	10000	0.050%	RBFS	materials problem	5
crystm03	24696	570	0.096%	—		
bundle1	10581	10461	0.688%		computer graphics/vision problem	
TED_B	10605	48	0.129%	RBFS	thermal problem	
TED_B_UNSCALED				RCM		
LF10000	19998	3	0.025%		model reduction problem	6
LFAT5000	19994	5	0.020%	KP-band		
Trefethen_20000b	19999	16384	0.139%	RBFS	combinatorial problem	
Trefethen_20000	20000	16384				

pend on the architecture (e.g. multicore environments). Concerning massively parallel computing, we also plan to evaluate heuristics for bandwidth reduction implemented within the Intel® Math Kernel Library running on Intel® Many Integrated Core architecture.

Acknowledgments

We thank J. Bernardes for sending us the results of experiments performed on his machine.

Referências

Ajiz, M. A. e Jennings, A. (1984). A robust incomplete Choleski-conjugate gradient algorithm. *International Journal for Numerical Methods in Engineering*, 20:949–966.

Benzi, M. (2002). Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182:418–477.

Davis, T. A. e Hu, Y. (2011). The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38(1):1–25.

George, A. (1971). *Computer implementation of the finite element method*. PhD thesis, Stanford University, Stanford.

George, A. e Liu, J. W. (1981). *Computer solution of large sparse positive definite systems*. Prentice-Hall, Englewood Cliffs.

George, A. e Liu, J. W. H. (1979). An implementation of a pseudoperipheral node finder. *ACM Transactions on Mathematical Software*, 5(3):284–295.

Gonzaga de Oliveira, S. L., Bernardes, J. A. B., e Chagas, G. O. (2018). An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. *Computational & Applied Mathematics*, 37(3):2965–3004.

Hestenes, M. R. e Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(36):409–436.

Koohestani, B. e Poli, R. (2011). A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In *Research and Development in Intelligent Systems XXVIII*, p. 93–106, London, UK. Springer London.

Lanczos, C. (1952). Solutions of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49(1):33–53.

Papadimitriou, C. H. (1976). The NP-completeness of bandwidth minimization problem. *Computing Journal*, 16:177–192.