An Assessment of Reordering Algorithms to Speed Up the ICCG Method Applied to CFD Problems

Sanderson L. Gonzaga de Oliveira[1(✉)], Guilherme Oliveira Chagas[1], Júnior Assis Barreto Bernardes[1], Diogo T. Robaina[2], and Mauricio Kischinhevsky[2]

[1] Universidade Federal de Lavras, Lavras, Minas Gerais, Brazil
sanderson@dcc.ufla.br, guilherme.chagas@computacao.ufla.br, jrassis@posgrad.ufla.br

[2] Universidade Federal Fluminense, Niterói, Rio de Janeiro, Brazil
{drobaina,kisch}@ic.uff.br

Abstract. Previous publications analyzed a large number of heuristics for bandwidth and profile reductions, and 14 heuristics were selected as promising low-cost heuristics for these problems. Based on extensive numerical experiments, this paper evaluates these heuristics when applied to matrices contained in systems of linear equations arising from computational fluid dynamics problems and the most promising heuristics are identified when reducing the zero-fill incomplete Cholesky-preconditioned conjugate gradient method.

Keywords: Bandwidth reduction · Profile reduction
Sparse matrices · Graph labeling · Conjugate gradient method
Reordering algorithms

1 Introduction

An essential task in several scientific and engineering applications is the solution of systems of linear equations in the form $Ax = b$, where A is an $n \times n$ large-scale sparse matrix, x is the unknown n-vector solution which is sough, and b is a known n-vector. It is generally a part of the simulation that demands a high processing time [1].

Modern hierarchical memory architecture and paging policies favor programs that take locality of reference into consideration [1,2]. Thus, spatial locality (a sequence of recent memory references is clustered locally rather than randomly in the memory address space) is relevant when designing an algorithm in this context.

For the low-cost solution of large and sparse linear systems, an appropriate vertex labeling in a graph is desirable to ensure that the corresponding coefficient matrix A will have narrow bandwidth and small profile. Specifically, heuristics

O. Gervasi et al. (Eds.): ICCSA 2018, LNCS 10960, pp. 35–52, 2018.
https://doi.org/10.1007/978-3-319-95162-1_3

for bandwidth or profile reductions return a sequence of graph vertices with spatial locality. Thus, these heuristics are used to reach low computational costs when solving large sparse systems of linear equations [1,2].

The bandwidth reduction problem consists of labeling the vertices of a graph with positive integer labels aiming at minimizing the maximum absolute difference between labels of adjacent vertices. It is isomorphic to the problem of reordering the rows and columns of a symmetric matrix so that the objective is to locate non-null coefficients as close as possible along the main diagonal [3]. Let $A = [a_{ij}]$ be an $n \times n$ symmetric matrix associated with a connected undirected graph $G = (V, E)$ composed of a set of vertices V and a set of edges E. The bandwidth of row i is $\beta_i(A) = i - \min_{1 \leq j \leq i} [j : a_{ij} \neq 0]$, for $a_{ii} \neq 0$. Bandwidth $\beta(A)$ is the largest distance between the non-null coefficient of the lower triangular matrix and the main diagonal considering all rows of the matrix, that is, $\beta(A) = \max_{1 \leq i \leq n} [\beta_i(A)]$. 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 $s : V \rightarrow \{1, 2, \cdots, |V|\}$) is $\beta(G) = \max_{v \in V} [(\{v, u\} \in E) \ |s(v) - s(u)|]$, where $s(v)$ and $s(u)$ are labels of vertices v and u, respectively. The profile of A can be defined as $profile(A) = \sum_{i=1}^{n} \beta_i(A)$, or equivalently, $profile(G) = \sum_{v \in V} \max_{\{v,u\} \in E} [|s(v) - s(u)|]$. The bandwidth and profile minimization problems are NP-hard [4,5] and, since the mid-1960s, several heuristics have been proposed to solve the bandwidth and profile reduction problems [2,6–8] due to the existence of extensive connections between these problems and a large number of other scientific and engineering problems [1].

A prominent method for solving large sparse systems of linear equations is the preconditioned conjugate gradient method (CGM) when the matrix contained in them are symmetric and positive definite. The number of iterations of the CG method depends on the structure of the matrix A, its condition number, the accuracy to be achieved, and the preconditioner used [1]. Thus, the most important result of a reordering algorithm in conjunction with the preconditioner is to return a matrix A with a reduced condition number (or equivalently better cluster of eigenvalues). The reordering algorithm should provide spatial locality. Moreover, the application should use a data structure that depends neither on the bandwidth nor the profile of the matrix A, as is the case in the experiments presented in this paper. Consequently, the resulting linear system is much easier to solve than the original linear system, even when the linear system is composed of multiple right-hand side vectors [1].

A specific preconditioner is applied to speed up the conjugate gradient method depending on the problem in context. Among various preconditioners proposed (e.g., see [9]) for the conjugate gradient method, the incomplete Cholesky (IC) factorization is especially useful [1]. The IC(l) preconditioner with $l > 0$ will obtain a better approximation to A than IC(0) at the cost of increased memory requirements and processing times. Thus, we used the IC(0) preconditioner. The zero-fill incomplete Cholesky-preconditioned conjugate gradient method (ICCGM) has been employed promisingly in the solution of various problems (see [1] and references therein).

One can reduce computational costs using the conjugate gradient method by applying an adequate ordering of the vertices of the corresponding graph of A to improve cache hit rates. Such ordering can be reached using a heuristic for bandwidth reduction (see [1] and references therein).

Systematic reviews [2,6–8] identified almost 140 heuristics for bandwidth and profile reductions. This number is increasing each year [1,10,11]. Therefore, an important decision in this context is to choose the best reordering algorithm among several alternatives for the problem at hand. Furthermore, in previous publications [10–13], various heuristics for bandwidth and profile reductions were evaluated with the intention of reducing the execution cost of the Jacobi-preconditioned conjugate gradient method. Additionally, in a recent publication [1], several heuristics for bandwidth and profile reductions were evaluated. This publication [1] also identified the most promising heuristics for several application areas when reducing the computational cost of the ICCG method. Among 55 instances (arising from several application areas) used in this publication [1], it evaluated the heuristics for bandwidth and profile reductions when applied to only three linear systems originating from computational fluid dynamic (CFD) problems. Thus, this present work aims to evaluate the same 14 heuristics for bandwidth and profile reductions to speed up the ICCG method when applied to several instances that arise from this application area. To provide more specific details, systematic reviews [1,2,6–8] reported 14 promising low-cost heuristics for bandwidth (Burgess-Lai [14], FNCHC [15], GGPS [16], VNS-band [17], KP-band [3], CSS-band [18], and RBFS-GL [1]) and profile (Snay [19], Sloan [20], MPG [21], NSloan [22], and Sloan-MGPS [23]) reductions. Additionally, this paper [1] evaluated the reverse Cuthill-McKee method with pseudo-peripheral vertex given by the George-Liu algorithm (RCM-GL) [24] and GPS algorithm [25] in both contexts of bandwidth and profile reductions. Therefore, 14 heuristics were evaluated in this work. Thus, our work provides an extensive computational experiment employing heuristics for bandwidth and profile reductions aiming at reducing the computational time of the ICCG method applied to 23 instances originating from CFD problems. To be more precise, the main contribution of our work is the comparison of results obtained when using 14 heuristics for bandwidth and profile reductions to symmetric matrices contained in linear systems (with sizes nearly 300,000 unknowns) arising from CFD problems solved by the ICCG method.

The remainder of this manuscript is organized as follows. Section 2 describes how this paper conducted the simulations in this computational experiment. Section 3 presents and analyzes the results. Finally, Sect. 4 provides the conclusions.

2 Description of the Tests

This computational experiment uses a 64-bit executable program of the VNS-band heuristic. One of the heuristics' authors kindly provided this program. Additionally, a heuristics' author kindly provided the C programming language source code of the FNCHC heuristic. Then, we converted this source code to the C++ programming language. The 12 other heuristics were also implemented in the C++ programming language to compare the computational costs of the heuristics. Furthermore, we used a data structure based on the Compress Row Storage, Compress Column Storage, and Skyline Storage Scheme data structures to implement the ICCG method. Previous publications described details of our implementations, testing, and calibration of the heuristics [1,2].

The three datasets (composed of real symmetric instances) that are part of the simulations carried out on each machine are (Intel® Core™): (*i*) systems of linear equations (ranging from 7,332 to 277,118 unknowns) arising from finite-volume discretizations of the Laplace equation [26], originally using a random order, with executions performed on a workstation that contains an i3-2120 CPU 3.30 GHz, 3 MB of cache, 8 GB DDR3 1.333 GHz of main memory, Linux kernel 3.13.0-39-generic; (*ii*) linear systems (ranging from 4,846 to 232,052 unknowns) originating from finite-volume discretizations of the heat conduction equation with meshes generated using Voronoi diagrams (and Delaunay triangulations) [27], with executions performed on a workstation that contains an i3-550 CPU 3.20 GHz, 4 MB cache, 16 GB DDR3 1.333 GHz of main memory, Linux kernel 3.13.0-39-generic; (*iii*) 11 instances (ranging from 1,733 to 81,920 unknowns) contained in the SuiteSparse matrix (SSM) collection [28], with executions performed on a workstation that contains an i7-4790K CPU 4.00 GHz, 8 MB cache, 12 GB DDR3 1.6 GHz of main memory, Linux kernel 3.19.0-31-generic (Intel; Santa Clara, CA, United States). The Ubuntu 14.04 LTS 64-bit operating system was used in the simulations. A dataset used in each machine was chosen arbitrarily.

The convergence criterion used in the ICCG method was a reduction of the computed residual $|Ax = b|$ (i.e., a final backward error) to less than 10^{-16}. Thus, the final attainable accuracy in our numerical experiments is related to this precision. Three sequential runs were performed for each instance with both a reordering algorithm and with the ICCG method. This present work employs the GNU Multiple Precision Floating-point Computations with Correct-Rounding library to make it possible to obtain high precision in the computations.

3 Results and Analysis

This section presents and analyzes the results obtained in simulations using the ICCG method computed after executing heuristics for bandwidth and profile reductions. Sections 3.1 and 3.2 show the results obtained from the solutions of linear systems arising from finite-volume discretizations of the heat conduction (with instances ranging from 4,846 to 232,052 unknowns) and Laplace equations (with instances ranging from 7,322 to 277,188 unknowns), respectively.

Section 3.3 shows the results obtained from the solutions of 11 systems of linear equations (ranging from 1,733 to 81,920 unknowns) contained in the SuiteSparse matrix collection [28].

Tables contained in this section show the number of unknowns (n), the name of the reordering algorithm applied, the bandwidth and profile results, the results of the heuristics in relation to the computational times, in seconds (s), and the memory requirements, in mebibytes (MiB). These tables provide the computational times of the IC(0) preconditioner and the conjugate gradient method separately. It was decided to distinguish these costs because the renumbering produced using the reordering algorithms also affects the computational cost of the IC(0) preconditioner. These tables also present the number of iterations and the total computational times, in seconds, of the ICCG method. Moreover, in spite of the small number of executions for each heuristic in each instance, these tables provide the standard deviation σ and coefficient of variation $C_v = \sigma/\mu$ (where μ is the mean of the results analyzed), referring to the total execution cost of the ICCG method. Additionally, the first line of each instance presented in these tables shows results for systems of linear equations solved using the ICCG method without applying a reordering algorithm. These lines are indicated as "—" in these tables. With this result, one can check the speed-up (or speed-down) of the ICCG method provided by employing a reordering algorithm (i.e., the time of the ICCG method without applying a reordering algorithm divided by the time of the ICCG method executed in conjunction with a reordering algorithm), which is exhibited in the last columns of the tables below. Numbers in boldface are the best results. In addition, several figures in this section are presented as line charts for clarity.

3.1 Instances Arising from Finite-Volume Discretizations of the Heat Conduction Equation

Tables 1, 2 and Fig. 1, developed from an ample collection of heuristics for bandwidth and profile reductions that compose this work, show the average results obtained from the use of the ICCG method applied to instances derived from finite-volume discretizations of the heat conduction equation. These instances arise from the use of meshes generated by employing Voronoi diagrams (and Delaunay triangulations) [27].

Increasing the runtime of the VNS-band heuristic does not reduce the total cost of the whole simulation [10,11]. Table 2 and Fig. 1b show that several heuristics increased the processing time of the ICCG method or the speed-up is marginal when applied to various instances contained in this dataset (e.g., see the results related to the linear system composed of 50,592 unknowns in Table 2). Moreover, although Sloan's heuristic [20] reached the best results when applied to the smallest instances contained in this dataset (see Table 1), the RCM-GL method [24] yielded similar results to Sloan's [20] heuristic in these instances, and the RCM-GL method [24] obtained, in general, the best results in the largest linear system (see Table 2). Therefore, this method achieved on average the best

Table 1. Results from the solution of systems of linear equations (up to 23,367 unknowns and derived from finite-volume discretizations of the heat conduction equation [27]) using the ICCG method and vertices labeled by heuristics for bandwidth and profile reductions (continued on Table 2).

n	Heuristic	β	Profile	Heuristic		IC(0) t(s)	CGM t(s)	ICCGM		σ	C_v (%)	Speed-up
				t(s)	m.(MiB)			iter.	t(s)			
4846	—	4769	9116750	—	—	**2**	6	144	8	0.02	0.26	—
	Sloan	882	297009	0.006	0.2	3	3	85	7	0.02	0.29	**1.26**
	Sloan-MGPS	809	299210	0.025	0.0	3	3	86	7	0.02	0.37	1.24
	MPG	1084	305791	0.006	0.2	3	3	91	7	0.02	0.36	1.21
	Snay	1108	360746	0.285	0.1	3	3	**84**	6	0.02	0.35	1.21
	RCM-GL	152	425223	0.005	0.0	3	4	92	7	0.02	0.34	1.20
	GPS	140	406081	0.166	0.3	3	4	94	7	0.01	0.21	1.16
	RBFS-GL	165	448906	**0.002**	0.0	3	4	95	7	0.06	0.80	1.16
	GGPS	136	409134	0.397	0.6	3	4	94	7	0.01	0.07	1.11
	KP-band	152	471590	0.006	0.0	3	4	106	7	0.01	0.19	1.10
	NSloan	554	448633	0.004	0.0	3	4	106	7	0.03	0.35	1.10
	Burgess-Lai	233	417653	0.533	0.0	3	4	100	7	0.07	1.10	1.05
	VNS-band	154	499797	1.054	44.5	3	4	97	7	0.01	0.09	1.02
	CSS-band	4709	8470608	0.588	19.6	3	6	132	8	0.04	0.61	0.94
	FNCHC	**126**	460851	2.297	2.2	3	4	98	7	0.04	0.62	0.87
10728	—	10626	45314579	—	—	**12**	21	206	32	0.27	0.82	—
	Sloan	1415	**1041059**	0.020	0.3	16	11	124	27	0.28	1.03	**1.22**
	Sloan-MGPS	1521	1056012	0.100	0.3	16	11	124	27	0.16	0.60	1.21
	Snay	1418	1087614	0.880	0.3	16	**10**	**121**	**26**	0.12	0.44	1.19
	MPG	2087	1099888	0.020	0.2	16	11	133	27	0.12	0.43	1.18
	RBFS-GL	241	1455453	**0.007**	0.0	16	12	137	28	0.11	0.42	1.17
	RCM-GL	242	1504910	0.014	0.0	17	11	133	28	0.28	1.01	1.16
	KP-band	249	1645616	0.015	0.0	15	13	152	28	0.10	0.34	1.15
	GGPS	230	1540354	1.122	2.6	16	12	135	28	0.26	0.96	1.13
	GPS	**207**	1358676	0.658	1.5	17	11	133	28	0.11	0.38	1.13
	NSloan	1191	1525236	0.010	0.2	17	13	150	30	0.11	0.36	1.10
	VNS-band	552	1746660	1.148	135.6	17	12	145	29	0.51	1.75	1.07
	Burgess-Lai	398	1365197	5.895	0.0	16	12	141	28	0.06	0.22	0.96
	FNCHC	211	1646613	5.968	2.5	16	12	144	28	0.06	0.22	0.95
	CSS-band	10602	42658429	5.030	83.4	**12**	20	192	32	0.44	1.39	0.89
23367	—	23167	216212086	—	—	71	71	302	142	2.21	1.56	—
	Sloan	2973	3578074	0.080	0.6	95	33	177	129	1.60	1.25	**1.10**
	Sloan-MGPS	2713	3565599	0.334	0.3	96	33	178	129	0.90	0.70	1.09
	RCM-GL	367	4913698	0.050	0.0	94	37	192	130	0.45	0.35	1.09
	Snay	1823	**3413219**	2.942	0.7	95	33	**176**	**127**	0.76	0.60	1.09
	RBFS-GL	355	4786316	**0.020**	0.0	93	38	198	132	0.73	0.56	1.08
	MPG	3060	3493791	0.059	0.3	96	36	193	132	0.56	0.43	1.08
	KP-band	376	5521227	0.050	0.0	91	42	219	132	0.29	0.22	1.07
	GPS	**293**	4221479	3.650	3.3	93	37	194	131	0.45	0.35	1.06
	GGPS	317	5194437	5.220	3.5	93	39	202	132	0.11	0.08	1.04
	VNS-band	1564	8889127	1.500	371.1	94	43	223	137	0.58	0.42	1.02
	NSloan	1590	5012507	0.035	0.3	97	41	217	139	0.37	0.27	1.02
	FNCHC	309	5319151	15.320	3.0	93	40	208	133	0.97	0.73	0.96
	Burgess-Lai	465	4296542	11.190	0.0	96	42	213	137	0.69	0.51	0.96
	CSS-band	23127	203145287	38.630	549.1	**70**	67	276	137	2.80	2.05	0.81

Table 2. Results from the solution of systems of linear equations (ranging from 50,592 to 232,052 unknowns and derived from finite-volume discretizations of the heat conduction equation) using the ICCG method and vertices labeled by heuristics for bandwidth and profile reductions (continued from Table 1).

n	Heuristic	β	Profile	Heuristic		IC(0) t(s)	CGM t(s)	ICCGM		σ	C_v (%)	Speed-up
				t(s)	m.(MiB)			iter.	$t(s)$			
50592	—	50461	1020411959	—	—	**384**	220	431	604	0.8	0.14	—
	RCM-GL	553	16182346	0.1	0.0	479	116	278	595	4.0	0.67	1.02
	Snay	2928	**10727635**	9.4	1.4	482	**103**	257	**585**	1.0	0.17	1.02
	Sloan	4590	11711775	0.2	1.1	489	104	258	593	1.1	0.18	1.02
	RBFS-GL	541	16420191	0.1	0.0	478	118	283	597	3.0	0.50	1.01
	KP-band	547	18159371	0.1	0.0	471	128	314	600	0.2	0.03	1.01
	Sloan-MGPS	5150	11866029	1.2	1.1	490	105	**255**	594	2.8	0.48	1.01
	MPG	6604	11565237	0.1	1.1	492	111	275	603	2.4	0.39	1.00
	GPS	**466**	14149442	24.2	5.5	480	116	276	596	4.2	0.71	0.97
	VNS-band	7377	47268027	2.9	967.1	490	130	309	621	0.7	0.11	0.97
	NSloan	2568	16461468	0.1	1.1	499	132	316	631	3.2	0.50	0.96
	FNCHC	507	18166254	42.5	5.6	479	122	299	601	2.9	0.49	0.94
	GGPS	497	16985652	48.5	7.7	479	120	289	599	3.0	0.50	0.93
	CSS-band	50349	953617481	391.6	270.1	401	217	398	618	4.4	0.71	0.60
	Burgess-Lai	961	14447154	493.2	0.0	495	125	291	619	2.2	0.35	0.54
108683	—	108216	4725435534	—	—	**1963**	702	627	2665	3.6	0.13	—
	RBFS-GL	907	55649180	**0.1**	0.0	2166	357	408	2524	2.0	0.08	**1.06**
	RCM-GL	911	55350648	0.3	0.0	2185	352	400	2536	7.0	0.27	1.05
	KP-band	885	59355159	0.3	0.0	2177	400	452	2577	3.4	0.13	1.03
	Snay	3887	**33205366**	28.6	2.9	2248	**318**	**369**	2566	10.9	0.42	1.03
	VNS-band	16676	157588720	8.7	2293.7	2203	395	442	2599	4.2	0.16	1.02
	Sloan	8243	36198196	0.7	2.8	2306	321	370	2627	9.7	0.37	1.01
	MPG	14267	36552907	0.3	2.8	2298	344	396	2642	11.2	0.42	1.01
	Sloan-MGPS	7700	36496358	4.0	2.8	2309	328	371	2637	17.2	0.65	1.01
	FNCHC	756	59671449	116.2	11.6	2164	377	430	2541	7.3	0.29	1.00
	GPS	**642**	48744729	139.0	12.6	2167	355	402	2522	4.3	0.17	1.00
	NSloan	8680	52203136	0.2	2.8	2322	408	453	2730	21.8	0.80	0.98
	GGPS	743	54261170	226.5	23.3	2152	363	416	**2515**	2.1	0.08	0.97
	Burgess-Lai	1262	45836037	609.5	0.0	2259	389	428	2647	16.3	0.61	0.82
	CSS-band	108332	110396292	471.5	596.5	2179	700	575	2879	12.8	0.44	0.80
232052	—	231672	21652820640	—	—	**9339**	2167	908	11506	19.1	0.17	—
	RCM-GL	1231	168178362	0.6	0.0	9726	1081	573	**10807**	21.1	0.20	**1.07**
	RBFS-GL	1275	167101014	**0.3**	0.0	9731	1126	595	10857	8.9	0.08	1.06
	KP-band	1228	180118365	0.6	0.0	9811	1240	654	11050	3.7	0.03	1.04
	Snay	5818	**99995456**	92.3	6.3	10023	986	533	11009	36.1	0.33	1.04
	FNCHC	1148	185662203	286.3	22.4	9729	1171	618	10899	19.9	0.18	1.03
	VNS-band	17036	313594920	33.4	5048.3	9942	1221	636	11163	17.8	0.16	1.03
	Sloan	26898	121163861	2.6	5.2	10258	996	534	11254	23.3	0.21	1.02
	MPG	25623	117501685	0.8	5.2	10279	1070	574	11349	10.5	0.09	1.01
	Sloan-MGPS	26456	122023395	14.8	5.1	10424	1014	534	11438	50.5	0.44	1.01
	GPS	**1104**	148697458	584.6	25.4	9757	1108	580	10866	13.7	0.13	1.01
	NSloan	12010	166693574	0.5	5.2	10396	1275	657	11671	61.8	0.53	0.99
	GGPS	1190	172865013	1297.7	43.8	9694	1122	595	10816	11.5	0.11	0.95
	CSS-band	231315	2916471097	4486.6	841.4	11254	2207	830	13471	29.5	0.22	0.64
	Burgess-Lai	2048	153659093	47698.4	0.0	10287	1219	621	11472	6.5	0.06	0.19

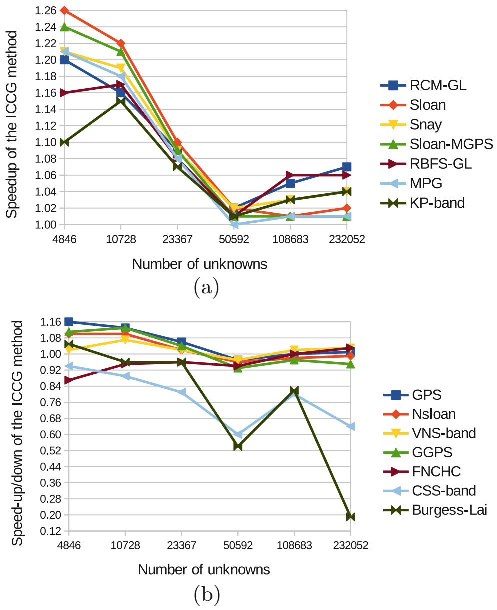

Fig. 1. Speed-ups/downs of the ICCG method obtained when using 14 heuristics for bandwidth and profile reductions applied to matrices contained in systems of linear equations derived from finite-volume discretizations of the heat conduction equation (see Tables 1 and 2).

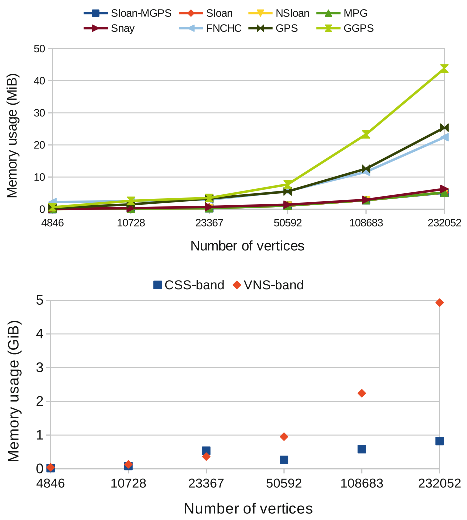

Fig. 2. Memory requirements of 10 heuristics for bandwidth and profile reductions applied to matrices contained in systems of linear equations derived from finite-volume discretizations of the heat conduction equation (see Tables 1 and 2).

results in reducing the computational time of the ICCG method when applied to systems of linear equations contained in this dataset.

Figure 2 shows memory requirements of 10 heuristics for bandwidth and profile reductions. The RCM-GL [24], Burgess-Lai [14], KP-band [3], and RBFS-GL [1] heuristics are in-place algorithms (if one implements the label as an attribute of a vertex).

3.2 Instances Originating from Finite-Volume Discretizations of the Laplace Equation

This section shows simulations using seven linear systems ranging from 7,322 to 277,118 unknowns. In particular, these instances (arising from finite-volume discretizations of the Laplace equation) were ordered randomly because irregular triangulations were employed to generate them [26].

Tables 3, 4 and Fig. 3 show that the RCM-GL method [24] dominated the other heuristics when considering the speedup of the ICCG method applied to these instances. In particular, Fig. 3 does not show the results obtained from the use of the CSS-band and Burgess-Lai heuristics because these two heuristics performed less favorably than the other heuristics applied to the instances contained in this dataset (see Tables 3 and 4). Snay's heuristic [19] was dominated by other heuristics so that we did not apply this algorithm to instances larger than 34,238 unknowns contained in this dataset.

Figure 4 shows memory requirements of eight heuristics for bandwidth and profile reductions applied to matrices contained in linear systems arising from finite-volume discretizations of the Laplace equation. This figure does not show the results from the use of the VNS-band and CSS-band heuristics because these two heuristics showed larger storage costs than the other heuristics when applied to instances contained in this dataset (see Tables 3 and 4).

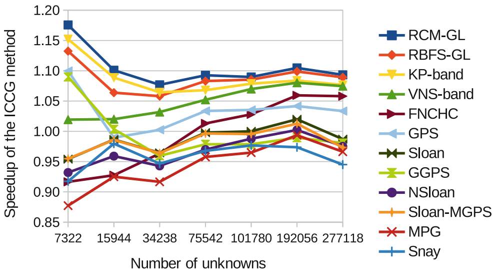

Fig. 3. Speed-ups/downs of the ICCG method obtained when applying 12 heuristics for bandwidth and profile reductions (see Tables 3 and 4) to matrices contained in linear systems with random order originating from finite-volume discretizations of the Laplace equation.

Table 3. Results from the solution of systems of linear equations (up to 34,238 unknowns, derived from finite-volume discretizations of the Laplace equation and composed of matrices with random order) using the ICCG method and vertices labeled by heuristics for bandwidth and profile reductions (continued on Table 4).

n	Heuristic	β	Profile	Heuristic		IC(0) t(s)	CGM t(s)	ICCGM		σ	C_v (%)	Speed-up
				t(s)	mem.			iter.	$t(s)$			
7322	—	7248	16083808	—	—	**3**	8	271	11	0.01	0.05	—
	RCM-GL	80	397878	0.005	0.0	4	5	201	**9**	0.01	0.12	**1.18**
	RBFS-GL	80	398377	**0.003**	0.0	4	6	202	10	0.01	0.06	1.13
	KP-band	80	406461	0.006	0.0	4	5	208	10	0.02	0.24	1.15
	Burgess-Lai	152	407458	0.151	0.0	4	6	201	10	0.01	0.11	1.12
	GPS	78	404414	0.362	0.2	4	5	203	10	0.11	1.12	1.10
	GGPS	80	403391	0.323	1.3	4	6	204	10	0.02	0.27	1.09
	VNS-band	1599	966638	1.061	75.9	4	6	214	10	0.07	0.67	1.02
	Sloan-MGPS	296	**374434**	0.031	0.2	5	7	**197**	12	0.08	0.71	0.96
	Sloan	407	377055	0.010	0.2	5	7	199	12	0.02	0.19	0.95
	NSloan	178	425289	0.004	0.2	5	7	204	12	0.01	0.12	0.93
	CSS-band	7185	15979344	0.870	40.9	**3**	8	271	11	0.10	0.87	0.92
	Snay	1853	448728	0.240	0.3	5	7	199	12	0.28	2.39	0.92
	FNCHC	**71**	423722	2.222	0.5	4	6	205	10	0.02	0.18	0.92
	MPG	846	610304	0.010	0.2	4	8	228	13	0.09	0.75	0.88
15944	—	15902	76482022	—	—	**19**	26	396	45	0.08	0.17	—
	RCM-GL	120	1149124	0.020	0.0	24	17	290	**41**	0.04	0.11	**1.10**
	KP-band	121	1219929	0.020	0.0	24	17	304	41	0.07	0.16	1.09
	RBFS-GL	124	1148009	**0.010**	0.0	25	18	291	42	0.04	0.09	1.06
	Burgess-Lai	212	1144254	1.090	0.0	25	18	291	43	0.14	0.33	1.03
	VNS-band	3916	5108940	1.180	196.7	25	19	319	43	0.28	0.64	1.02
	GGPS	117	1210422	1.790	2.8	25	19	302	43	0.21	0.48	1.00
	GPS	118	1154030	3.610	0.5	25	17	290	42	0.59	1.40	0.99
	Sloan	477	**982724**	0.026	0.3	24	22	286	46	0.24	0.53	0.99
	Sloan-MGPS	473	1003462	0.097	0.3	24	22	284	46	0.11	0.23	0.99
	Snay	5862	1586436	1.027	0.4	23	22	**281**	45	0.24	0.52	0.98
	NSloan	218	1222337	0.014	0.3	24	23	298	47	0.15	0.31	0.96
	MPG	1222	1587797	0.028	0.3	23	25	323	49	0.37	0.76	0.93
	FNCHC	**113**	1315311	5.730	1.3	25	18	301	43	0.13	0.31	0.93
	CSS-band	15724	76687757	8.060	192.7	**19**	26	395	44	0.56	1.26	0.86
34238	—	34609	357518296	—	—	108	82	565	190	0.60	0.32	—
	RCM-GL	192	3413888	0.040	0.0	126	**50**	411	**176**	0.47	0.27	**1.08**
	RBFS-GL	191	3416318	**0.002**	0.0	127	52	412	179	0.75	0.42	1.06
	KP-band	193	3756415	0.040	0.0	125	53	434	178	0.03	0.02	1.06
	VNS-band	2726	6767128	1.670	490.5	127	55	444	182	0.31	0.17	1.03
	Burgess-Lai	334	3282297	4.300	0.0	130	54	419	184	0.01	0.09	1.01
	GPS	191	3545656	9.610	1.6	127	53	418	179	0.80	0.45	1.00
	Sloan	917	**2578022**	0.060	0.8	129	68	395	196	0.94	0.48	0.96
	Sloan-MGPS	845	2672048	0.299	0.9	129	67	**388**	196	1.24	0.63	0.96
	FNCHC	**170**	3910584	15.030	4.0	126	55	436	182	0.76	0.42	0.96
	GGPS	192	3415253	19.630	5.2	126	52	411	178	0.36	0.21	0.96
	Snay	21625	5150148	5.496	1.0	128	67	391	195	1.36	0.70	0.95
	NSloan	366	3608451	**0.033**	0.9	128	73	419	201	0.70	0.35	0.94
	MPG	2111	3965061	0.059	0.9	129	77	448	207	0.42	0.20	0.92
	CSS-band	33953	359290792	68.630	910.7	**107**	81	562	189	2.80	1.48	0.74

Table 4. Results from the solution of systems of linear equations (ranging from 75,542 to 277,118 unknowns, derived from finite-volume discretizations of the Laplace equation and composed of matrices with random order) using the ICCG method and vertices labeled by heuristics for bandwidth and profile reductions (continued from Table 3).

n	Heuristic	β	Profile	Heuristic		IC(0) t(s)	CGM t(s)	ICCGM		σ	C_v (%)	Speed-up
				t(s)	m.(MiB)			iter.	t(s)			
75542	—	75490	1744941733	—	—	**591**	268	815	859	1.2	0.13	—
	RCM-GL	275	12096902	0.1	0.0	622	**164**	607	**786**	0.7	0.08	**1.09**
	RBFS-GL	275	12128940	0.1	0.0	624	169	616	793	1.4	0.18	1.08
	KP-band	277	13793938	0.1	0.0	625	179	663	804	0.1	0.02	1.07
	VNS-band	21310	42564399	3.7	1198.1	637	176	642	813	2.2	0.26	1.05
	GPS	272	12086603	41.4	3.6	624	166	607	790	0.4	0.05	1.03
	FNCHC	**266**	13848181	41.5	7.2	630	177	644	806	2.7	0.33	1.01
	Burgess-Lai	460	11175444	42.8	0.0	642	171	612	814	0.8	0.10	1.00
	Sloan	1521	**8981209**	0.2	1.1	647	214	572	861	5.5	0.63	1.00
	Sloan-MGPS	1236	9245713	1.1	1.1	646	215	568	861	2.0	0.23	1.00
	GGPS	271	12405895	86.2	10.3	625	166	604	792	1.5	0.18	0.98
	NSloan	533	12805775	0.7	1.1	646	250	628	886	2.4	0.27	0.97
	MPG	4020	14107424	0.2	1.1	645	247	653	897	5.4	0.60	0.96
	CSS-band	75031	1747217131	692	3819.2	**591**	269	819	859	7.6	0.88	0.55
101780	—	101583	3169282786	—	—	1094	405	907	1499	0.1	0.01	—
	RCM-GL	407	21336387	0.1	0.0	1126	**249**	670	1375	0.6	0.05	**1.09**
	RBFS-GL	406	21346316	0.1	0.0	1127	254	676	1381	0.7	0.05	**1.09**
	KP-band	408	23967214	0.1	0.0	1124	265	716	1389	1.8	0.13	1.08
	VNS-band	5207	25033097	5.6	1638.4	1140	256	686	1395	0.7	0.05	1.07
	GPS	405	21399542	73.4	4.9	1125	250	668	**1374**	3.5	0.25	1.04
	FNCHC	**399**	25629738	59.9	9.8	1133	265	705	1398	1.0	0.07	1.03
	Sloan	8624	**15003839**	0.3	2.6	1177	322	640	1498	0.1	0.01	1.00
	Sloan-MGPS	8420	15400014	2.0	2.6	1180	323	635	1503	7.4	0.49	1.00
	NSloan	7730	21163363	0.1	2.6	1165	352	685	1517	8.3	0.54	0.99
	GGPS	402	21624122	152.7	15.5	1128	251	665	1379	2.7	0.19	0.98
	MPG	10570	24189624	0.2	2.6	1185	368	723	1553	0.6	0.04	0.97
	CSS-band	101300	3161769330	227.1	611.4	**1093**	405	906	1498	9.2	0.61	0.87
	Burgess-Lai	745	19394495	4383.7	0.0	1152	262	683	1414	4.3	0.31	0.26
192056	—	191738	11329772559	—	—	4115	1045	1215	5160	11.0	0.21	—
	RCM-GL	360	42577946	0.2	0.0	**4030**	**642**	909	**4671**	3.9	0.08	**1.11**
	RBFS-GL	360	42627911	**0.1**	0.0	4037	659	927	4696	4.2	0.09	1.10
	KP-band	364	48325681	0.3	0.0	4057	704	1000	4761	4.0	0.08	1.08
	VNS-band	11142	99018771	16.6	3195.0	4056	705	992	4761	8.2	0.17	1.08
	FNCHC	**344**	48743268	115.2	21.0	4056	700	982	4756	7.5	0.16	1.06
	GPS	371	41541059	257.4	10.1	4048	649	911	4696	7.8	0.17	1.04
	Sloan	1963	**30916661**	0.8	4.6	4219	840	866	5059	18.9	0.37	1.02
	Sloan-MGPS	1733	31860210	4.3	4.6	4249	842	857	5091	33.2	0.65	1.01
	NSloan	753	44591269	0.2	4.6	4223	925	932	5148	8.2	0.16	1.00
	Burgess-Lai	621	40149530	349.5	0.0	4142	682	934	4824	17.2	0.36	1.00
	MPG	5366	47979879	0.5	4.6	4240	953	971	5193	29.0	0.56	0.99
	GGPS	363	42925208	530.0	27.8	4040	645	904	4685	1.5	0.03	0.99
	CSS-band	191322	2731669814	1345.0	1124.9	4101	1047	1211	5148	15.0	0.29	0.80

(*continued*)

Table 4. (*continued*)

n	Heuristic	β	Profile	Heuristic		IC(0) t(s)	CGM t(s)	ICCGM		σ	C_v (%)	Speed-up
				t(s)	m.(MiB)			iter.	t(s)			
277118	—	277019	23512579029	—	—	8563	1766	1430	10329	3.2	0.03	—
	RCM-GL	420	74697812	0.3	0.0	**8371**	1076	1066	**9447**	7.0	0.07	**1.09**
	RBFS-GL	419	74759303	**0.2**	0.0	8387	1096	1078	9483	2.2	0.02	**1.09**
	KP-band	425	84682829	0.4	0.0	8416	1178	1168	9593	5.4	0.06	1.08
	VNS-band	12132	97666318	32.3	4618.2	8430	1149	1134	9579	11.6	0.12	1.08
	FNCHC	412	85849403	182.4	26.7	8419	1161	1144	9580	8.7	0.09	1.06
	GPS	**399**	72378558	511.5	16.4	8395	1087	1070	9483	23.4	0.25	1.03
	Burgess-Lai	793	66880423	397.9	0.0	8640	1138	1093	9778	2.9	0.03	1.02
	Sloan	2243	**55586225**	1.2	5.6	9070	1392	1012	10463	14.9	0.14	0.99
	GGPS	422	75150257	1051.7	20.7	8412	**1069**	1047	9480	6.1	0.06	0.98
	NSloan	908	77425832	0.3	5.6	9055	1535	1086	10590	66.4	0.63	0.98
	Sloan-MGPS	2094	57033429	7.4	5.6	9206	1399	**1001**	10605	68.3	0.64	0.97
	MPG	6920	89231836	0.8	5.6	9113	1576	1128	10689	44.0	0.41	0.97
	CSS-band	276240	2054061697	3994.8	1679.6	8579	1765	1421	10344	46.6	0.45	0.72

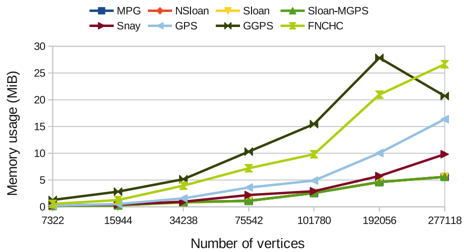

Fig. 4. Memory requirements of eight reordering algorithms applied to matrices contained in systems of linear equations with a random order (see Tables 3 and 4) originating from finite-volume discretizations of the Laplace equation.

3.3 Experiments Using Instances Contained in the SSM Collection

Tables 1, 2, 3, 4 and Figs. 1, 2, 3, 4 show that several heuristics dominated Snay's [19], NSloan [22], and CSS-band [18] heuristics. Moreover, these tables and figures show that the GPS [25], Burgess-Lai [14], FNCHC [15], GGPS [16], and VNS-band [17] algorithms were dominated by the RCM-GL [24], KP-band [3], and RBFS-GL [1] orderings. Hence, we applied six heuristics for bandwidth and profile reductions to a dataset composed of 11 systems of linear equations contained in the SSM collection: the RCM-GL [24], Sloan's [20], MPG [21], Sloan-MGPS [23], KP-band [3], and RBFS-GL [1] heuristics.

Tables 5 and 6 show the instance's name and density (d.). Moreover, these tables and Fig. 5 show the average results obtained by the ICCG method applied to 11 systems of linear equations contained in the SSM collection [28]. Tables 5 and 6 show the 2-norm (a 1-norm) condition number (estimation) $\kappa(A)$ of the instances. Specifically, Tables 5 and 6 show that the ICCG method converges in n iterations, where n is the size of the linear system when applied to ill-conditioned

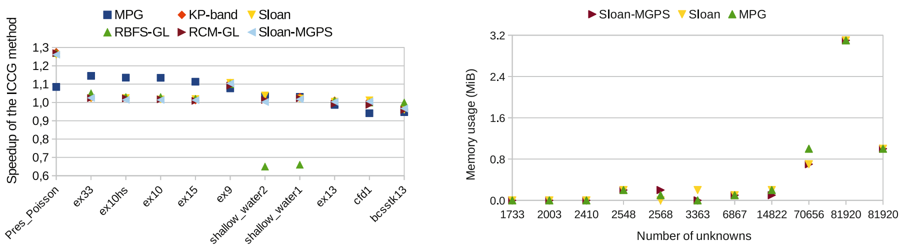

Fig. 5. Speedups of the ICCG method and memory usage obtained when using heuristics for bandwidth and profile reductions (see Tables 5 and 6) applied to 11 systems of linear equations arising from computational fluid dynamics problems contained in the SSM collection [28].

Table 5. Results from the solution of eight systems of linear equations (arising from CFD problems) contained in the SuiteSparse matrix collection [28] employing the zero-fill incomplete Cholesky-preconditioned conjugate gradient method and vertices labeled using reordering algorithms (continued on Table 6).

LS	n	d.	Heuristic	β	Profile	(2-norm) $\kappa(A)$	Heuristic		CGM t(s)	IC(0) t(s)	ICCGM		Speed-up/down
							t(s)	mem.			t(s)	iter.	
ex33	1733	0.74%	—	157	125884	7.0e+12	—	—	18.18	0.4	19	1733	—
			MPG	977	232924	3.1e+11	0.002	0.0	**15.81**	0.4	**16**	1733	**1.14**
			RBFS-GL	97	71487	3.1e+11	**0.001**	0.0	17.28	0.4	18	1733	1.05
			Sloan-MGPS	569	57790	3.7e+11	0.002	0.0	17.67	0.5	18	1733	1.02
			RCM-GL	97	70259	3.1e+11	0.002	0.0	17.76	0.4	18	1733	1.02
			Sloan	506	**55817**	3.7e+11	**0.001**	0.0	17.70	0.5	18	1733	1.02
			KP-band	**96**	76081	**2.3e+11**	0.001	0.0	17.78	0.4	18	1733	1.02
bcsstk13	2003	2.09%	—	1250	434798	1.1e+10	—	—	64.12	1.8	**66**	2003	—
			RBFS-GL	521	524853	2.2e+07	**0.001**	0.0	**64.08**	**1.5**	66	2003	**1.00**
			KP-band	537	644513	**1.8e+07**	0.005	0.0	66.50	1.6	68	2003	0.97
			Sloan-MGPS	1699	522268	1.9e+07	0.022	0.0	66.20	1.9	67	2003	0.97
			Sloan	1790	**512622**	2.5e+08	0.009	0.0	66.72	1.9	69	2003	0.96
			RCM-GL	**502**	519475	1.1e+08	0.006	0.0	67.19	**1.5**	69	2003	0.96
			MPG	1454	729001	1.5e+08	0.012	0.0	67.59	1.9	70	2003	0.95
ex10	2410	0.94%	—	113	170956	9.1e+11	—	—	55.41	**1.3**	57	2410	—
			MPG	1310	437834	2.7e+12	0.004	0.0	**48.62**	1.4	**50**	2410	**1.13**
			RBFS-GL	112	72752	1.5e+11	**0.001**	0.0	53.82	1.5	55	2410	1.03
			KP-band	**105**	132506	1.2e+11	0.003	0.0	54.24	**1.3**	56	2410	1.02
			RCM-GL	509	247099	2.0e+11	0.002	0.0	54.40	1.4	56	2410	1.02
			Sloan	215	**70891**	1.9e+11	0.004	0.0	54.32	1.5	56	2410	1.02
			Sloan-MGPS	174	71513	**9.5e+10**	0.003	0.0	54.47	1.5	56	2410	1.01
ex10hs	2548	0.88%	—	117	150904	5.5e+11	—	—	61.63	1.5	63	2548	—
			MPG	1236	460152	7.0e+12	0.006	0.2	**54.20**	1.4	**56**	2548	**1.14**
			RBFS-GL	109	78699	2.1e+11	**0.001**	0.0	59.86	1.5	61	2548	1.03
			KP-band	105	141132	4.5e+11	0.005	0.0	60.21	1.4	62	2548	1.02
			Sloan	199	77868	**1.4e+11**	0.003	0.2	60.33	1.3	62	2548	1.02
			RCM-GL	**105**	**77226**	5.2e+11	0.005	0.0	60.44	1.3	62	2548	1.02
			Sloan-MGPS	176	78605	1.6e+11	0.004	0.2	60.93	1.3	62	2548	1.01

(*continued*)

Table 5. (*continued*)

LS	n	d.	Heuristic	β	Profile	(2-norm) $\kappa(A)$	Heuristic t(s)	Heuristic mem.	CGM t(s)	IC(0) t(s)	ICCGM t(s)	ICCGM iter.	Speed-up/down
ex13	2568	1.15%	—	165	127134	1.2e+15	—	—	77.48	1.9	79	2568	—
			KP-band	167	231828	1.5e+14	0.004	0.0	**76.61**	1.9	79	2568	**1.02**
			RBFS-GL	185	125553	1.5e+14	**0.001**	0.0	76.94	1.9	79	2568	**1.01**
			Sloan	652	125050	1.5e+14	0.003	0.0	78.11	1.9	79	2568	1.00
			Sloan-MGPS	546	**122369**	1.5e+14	0.003	0.2	77.20	1.9	79	2568	1.00
			RCM-GL	**182**	125123	1.5e+14	0.005	0.0	78.11	1.9	80	2568	0.99
			MPG	1906	1061077	1.5e+14	0.010	0.1	78.60	1.9	80	2568	0.99
ex9	3363	0.88%	—	**85**	**156224**	1.2e+13	—	—	148.30	3.3	152	3363	—
			KP-band	148	288037	2.4e+12	0.005	0.0	**133.51**	3.3	137	3363	**1.11**
			Sloan	604	158613	2.4e+12	0.003	0.2	133.65	3.3	137	3363	1.11
			RBFS-GL	153	159760	2.4e+12	**0.002**	0.0	133.71	4.0	138	3363	1.10
			Sloan-MGPS	406	162327	2.4e+12	0.004	0.0	134.02	3.3	137	3363	1.10
			RCM-GL	147	157476	6.1e+12	0.006	0.0	135.60	3.3	139	3363	1.09
			MPG	2296	1744453	2.4e+12	0.015	0.0	137.55	3.3	141	3363	1.08
ex15	6867	0.21%	—	**67**	450556	8.6e+12	—	—	326.82	7.0	334	6867	—
			MPG	1338	1320684	**1.7e+13**	0.029	0.1	**293.21**	6.6	**300**	6867	**1.11**
			RBFS-GL	83	262534	4.6e+13	**0.001**	0.0	321.70	6.8	328	6867	1.02
			Sloan	139	**250857**	2.0e+13	0.005	0.1	321.55	**6.2**	328	6867	1.02
			Sloan-MGPS	146	251009	6.4e+13	0.008	0.1	323.40	**6.2**	330	6867	1.01
			KP-band	84	267003	2.6e+14	0.006	0.0	323.54	6.8	330	6867	1.01
			RCM-GL	83	261901	3.2e+13	0.005	0.0	324.18	6.8	331	6867	1.01
Pres_Poisson	14822	0.33%	—	12583	9789525	2.0e+06	—	—	77.40	107.8	186	269	—
			KP-band	364	3130744	3.5e+05	0.046	0.0	37.82	107.5	**145**	133	**1.28**
			RBFS-GL	**331**	3024736	1.1e+06	**0.019**	0.0	38.66	106.9	146	137	1.27
			RCM-GL	346	3051085	3.0e+06	0.048	0.0	**37.52**	108.6	146	**132**	1.27
			Sloan-MGPS	640	2832467	2.7e+06	0.114	0.1	39.15	107.6	147	138	1.26
			Sloan	548	**2830658**	**3.2e+05**	0.049	0.2	39.25	107.8	147	138	1.26
			MPG	14168	26556694	7.7e+05	2.399	0.2	64.05	**104.8**	169	220	1.08

instances (i.e., with large condition numbers). Among the information presented in Table 5, in particular, this table shows a small number of iterations for the ICCG method when applied to the *Pres_Poisson* instance, whose condition number is smaller than the other seven linear systems contained in this table. Furthermore, Table 6 shows that the execution times of the ICCG method to solve the *shallow_water1* and *shallow_water2* instances (composed of 81,920 unknowns and with small condition numbers) are approximately four times lower than the execution times to compute the *cfd1* instance (composed of 70,656 unknowns and with a large condition number).

The MPG [21] (KP-band [3]) heuristic obtained the best results when applied to the *ex33*, *ex10*, *ex10hs*, and *ex15* (*ex9* and *Pres_Poisson*) instances. The KP-band [3] (Sloan [20]) heuristic obtained the best results when applied to the *ex13* and *shallow_water1* (*cfd1* and *shallow_water2*) instances, but these gains are marginal.

When setting the same precision in exploratory investigations using standard double-precision floating-point format, the ICCG method converged in a similar

Table 6. Results from the solution of three systems of linear equations (arising from CFD problems) contained in the SSM collection [28] employing the ICCG method and vertices labeled using reordering algorithms (continued from Table 5).

LS	n	d.	Heuristic	β	Profile	(1-norm) $\kappa(A)$ est.	Heuristic		CGM t(s)	IC(0) t(s)	ICCGM		Speed-up/down
							t(s)	mem.			t(s)	iter.	
cfd1	70656	0.04%	—	6229	1.0+08	**1.3e+06**	—	—	443	1441	1884	542	—
			Sloan	19834	6.0e+07	1.8e+06	1.0	0.7	**413**	1449	**1862**	**499**	**1.01**
			KP-band	3322	1.3e+08	4.1e+06	0.2	0.0	443	1423	1866	539	1.01
			Sloan-MGPS	19186	**5.7e+07**	3.6e+06	4.1	0.7	416	1453	1869	501	1.01
			RBFS-GL	3289	1.2e+08	1.3e+06	**0.1**	0.0	428	1448	1876	524	1.00
			RCM-GL	**3182**	1.1e+08	3.5e+06	0.2	0.0	450	1459	1909	542	0.99
			MPG	56114	4.7e+08	2.8e+06	17.9	1.0	576	**1408**	1984	682	0.94
shallow_water1	81920	0.01%	—	40959	**3.5e+07**	3.6	—	—	20	470	473	7	—
			KP-band	1029	6.8e+07	1.3	0.2	0.0	2	**456**	**458**	6	**1.03**
			MPG	3619	7.0e+07	1.3	0.8	3.1	2	**456**	**458**	7	1.03
			RCM-GL	**966**	6.6e+07	1.5	0.2	0.0	2	460	462	6	1.02
			Sloan	3070	6.8e+07	1.3	0.7	3.1	2	460	462	6	1.02
			Sloan-MGPS	2578	6.6e+07	1.3	4.0	3.1	2	459	460	6	1.02
			RBFS-GL	322	2.2e+07	2.2	**0.1**	0.0	2	713	715	6	0.66
shallow_water2	81920	0.01%	—	40959	3.5e+07	11.3	—	—	3	464	467	10	—
			Sloan	1225	2.2e+07	1.6	0.2	1.0	3	**447**	**450**	10	**1.04**
			MPG	1226	2.4e+07	1.6	0.3	1.0	3	449	452	10	1.03
			RCM-GL	329	2.2e+07	4.2	**0.1**	0.0	3	458	461	10	1.01
			KP-band	**327**	2.2e+07	1.6	**0.1**	0.0	3	458	461	10	1.01
			Sloan-MGPS	896	2.3e+07	1.6	1.4	1.0	3	462	465	10	1.00
			RBFS-GL	337	3.3e+07	4.2	**0.1**	0.0	3	714	717	**9**	0.65

number of iterations to using high-precision floating-point arithmetic. Thus, in these experiments, we observed that high-precision floating-point arithmetic does not minimize delay in the convergence of the ICCG method. Figure 5 also shows the memory usage obtained when using three heuristics for bandwidth and profile reductions applied to 11 instances arising from computational fluid dynamics problems contained in the SSM collection [28].

4 Conclusions

Systematic reviews [1,2,6–8] reported the most promising low-cost heuristics for bandwidth and profile reductions. Thus, our computational experiment compared the results of the implementations of 14 heuristics for bandwidth and profile reductions when applied to 23 instances arising from CFD problems.

Table 5 shows six promising heuristics for bandwidth and profile reductions to reduce computational times of the ICCG method. In particular, in experiments using instances (with sizes to almost 300,000 unknowns) from three datasets, three out of 14 heuristics for bandwidth and profile reductions evaluated in this computational experiment showed the best overall results in reducing the processing cost of the ICCG method. Specifically, the RCM-GL [24], MPG [21], and

KP-band [3] obtained the most promising results when used to reduce the processing cost of the ICCG method when applied to instances arising from computational fluid dynamics problems. Future studies will reveal whether this may be extended to any preconditioned conjugate gradient method or even to other preconditioned iterative solvers. In particular, the RCM-GL method [24] achieved the best results when applied to instances arising from finite-volume discretizations of the heat conduction and Laplace equations [26,27] (see Tables 2, 3, 4 and Figs. 1a and 3). Thus, the in-place low-cost RCM-GL method [24] remains in the state of the practice in bandwidth reduction when applied to instances originating from CFD problems. On the other hand, the MPG [21] (in four linear systems) and KP-band [3] (in two linear systems) heuristics reached the largest number of best results in simulations with 11 instances contained in the SuiteSparse matrix collection [28].

As a continuation of this work, we intend to implement and evaluate other preconditioners in conjunction with the conjugate gradient method. We also plan to implement parallel approaches of the above algorithms in future investigations.

Acknowledgments. We would like to thank Prof. Dr. Dragan Urosevic, from the Mathematical Institute SANU, for sending us the VNS-band executable program. We would also like to thank Prof. Dr. Fei Xiao for sending us the source code of the FNCHC heuristic.

References

1. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of reordering algorithms to reduce the computational cost of the incomplete Cholesky-conjugate gradient method. Comput. Appl. Math. (2017). https://doi.org/10.1007/s40314-017-0490-5
2. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. Comput. Appl. Math. **37**(2), 1412–1471 (2018). https://doi.org/10.1007/s40314-016-0394-9
3. Koohestani, B., Poli, R.: A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In: Bramer, M., Petridis, M., Nolle, L. (eds.) Research and Development in Intelligent Systems XXVIII, pp. 93–106. Springer, London (2011). https://doi.org/10.1007/978-1-4471-2318-7_7
4. Papadimitriou, C.H.: The NP-completeness of bandwidth minimization problem. Comput. J. **16**, 177–192 (1976)
5. Lin, Y.X., Yuan, J.J.: Profile minimization problem for matrices and graphs. Acta Mathematicae Applicatae Sinica **10**(1), 107–122 (1994)
6. Chagas, G.O., Gonzaga de Oliveira, S.L.: Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. Proc. Comput. Sci. (Proc. Int. Conf. Comput. Sci. (ICCS)) **51**, 211–220 (2015)
7. Bernardes, J.A.B., Gonzaga de Oliveira, S.L.: A systematic review of heuristics for profile reduction of symmetric matrices. Proc. Comput. Sci. (Proc. Int. Conf. Comput. Sci. (ICCS)) **51**, 221–230 (2015)

8. Gonzaga de Oliveira, S.L., Chagas, G.O.: A systematic review of heuristics for symmetric-matrix bandwidth reduction: methods not based on metaheuristics. In: The XLVII Brazilian Symposium of Operational Research (SBPO), Ipojuca-PE, Brazil, Sobrapo, August 2015
9. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
10. Gonzaga de Oliveira, S.L., de Abreu, A.A.A.M., Robaina, D., Kischinhevsky, M.: A new heuristic for bandwidth and profile reductions of matrices using a self-organizing map. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9786, pp. 54–70. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42085-1_5
11. Gonzaga de Oliveira, S.L., Abreu, A.A.A.M., Robaina, D.T., Kischnhevsky, M.: An evaluation of four reordering algorithms to reduce the computational cost of the Jacobi-preconditioned conjugate gradient method using high-precision arithmetic. Int. J. Bus. Intell. Min. **12**(2), 190–209 (2017)
12. Gonzaga de Oliveira, S.L., Bernardes, J.A.B., Chagas, G.O.: An evaluation of several heuristics for bandwidth and profile reductions to reduce the computational cost of the preconditioned conjugate gradient method. In: Proceedings of the Brazilian Symposium on Operations Research (SBPO 2016), Vitória, Brazil, Sobrapo, September 2016
13. Gonzaga de Oliveira, S.L., Chagas, G.O., Bernardes, J.A.B.: An analysis of reordering algorithms to reduce the computational cost of the Jacobi-preconditioned CG solver using high-precision arithmetic. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10404, pp. 3–19. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62392-4_1
14. Burgess, I.W., Lai, P.K.F.: A new node renumbering algorithm for bandwidth reduction. Int. J. Numer. Meth. Eng. **23**, 1693–1704 (1986)
15. Lim, A., Rodrigues, B., Xiao, F.: A fast algorithm for bandwidth minimization. Int. J. Artif. Intell. Tools **3**, 537–544 (2007)
16. Wang, Q., Guo, Y.C., Shi, X.W.: A generalized GPS algorithm for reducing the bandwidth and profile of a sparse matrix. Prog. Electromagn. Res. J. **90**, 121–136 (2009)
17. Mladenovic, N., Urosevic, D., Pérez-Brito, D., García-González, C.G.: Variable neighbourhood search for bandwidth reduction. Eur. J. Oper. Res. **200**, 14–27 (2010)
18. Kaveh, A., Sharafi, P.: Ordering for bandwidth and profile minimization problems via charged system search algorithm. Iranian J. Sci. Technol.-Trans. Civil Eng. **36**(2), 39–52 (2012)
19. Snay, R.A.: Reducing the profile of sparse symmetric matrices. Bull. Geodésique **50**(4), 341–352 (1976)
20. Sloan, S.W.: A Fortran program for profile and wavefront reduction. Int. J. Numer. Meth. Eng. **28**(11), 2651–2679 (1989)
21. Medeiros, S.R.P., Pimenta, P.M., Goldenberg, P.: Algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. Eng. Comput. **10**(3), 257–266 (1993)
22. Kumfert, G., Pothen, A.: Two improved algorithms for envelope and wavefront reduction. BIT Numer. Math. **37**(3), 559–590 (1997)
23. Reid, J.K., Scott, J.A.: Ordering symmetric sparse matrices for small profile and wavefront. Int. J. Numer. Meth. Eng. **45**(12), 1737–1755 (1999)
24. George, A., Liu, J.W.: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs (1981)

25. Gibbs, N.E., Poole, W.G., Stockmeyer, P.K.: An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM J. Numer. Anal. **13**(2), 236–250 (1976)
26. Gonzaga de Oliveira, S.L., Kischinhevsky, M., Tavares, J.M.R.S.: Novel graph-based adaptive triangular mesh refinement for finite-volume discretizations. Comput. Model. Eng. Sci. **95**(2), 119–141 (2013)
27. Gonzaga de Oliveira, S.L., de Oliveira, F.S., Chagas, G.O.: A novel approach to the weighted Laplacian formulation applied to 2D Delaunay triangulations. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2015. LNCS, vol. 9155, pp. 502–515. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21404-7_37
28. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1:1–1:25 (2011)