An Analysis of Reordering Algorithms to Reduce the Computational Cost of the Jacobi-Preconditioned CG Solver Using High-Precision Arithmetic

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

[1] Universidade Federal de Lavras, Lavras, Minas Gerais, Brazil
sanderson@dcc.ufla.br, jrassis@posgrad.ufla.br
[2] Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, Brazil
guilherme.o.chagas@gmail.com

Abstract. Several heuristics for bandwidth and profile reductions have been proposed since the 1960s. In systematic reviews, 133 heuristics applied to these problems have been found. The results of these heuristics have been analyzed so that, among them, 13 were selected in a manner that no simulation or comparison showed that these algorithms could be outperformed by any other algorithm in the publications analyzed, in terms of bandwidth or profile reductions and also considering the computational costs of the heuristics. Therefore, these 13 heuristics were selected as the most promising low-cost methods to solve these problems. Based on this experience, this article reports that in certain cases no heuristic for bandwidth or profile reduction can reduce the computational cost of the Jacobi-preconditioned Conjugate Gradient Method when using high-precision numerical computations.

Keywords: Bandwidth reduction · Profile reduction · Conjugate Gradient Method · Graph labeling · Reordering algorithms · Sparse matrices · Graph algorithm · High-precision arithmetic · Ordering · Sparse symmetric positive-definite linear systems · Combinatorial optimization · Heuristics

1 Introduction

In several scientific and engineering fields, such as finite element analysis, computational fluid mechanics, and structural engineering, a fundamental task is the resolution of large sparse linear systems with the form $Ax = b$, where A is an $n \times n$ sparse, symmetric, and positive-definite matrix, b is a vector of length n, and x is an unknown vector (which is sought) of length n. Generally, the highest computational cost of the simulation is required in the resolution of these

O. Gervasi et al. (Eds.): ICCSA 2017, Part I, LNCS 10404, pp. 3–19, 2017.
DOI: 10.1007/978-3-319-62392-4_1

linear systems. A substantial amount of memory and a high processing cost are necessary to store and to solve these large-scale linear systems. For the low-cost solution of large and sparse linear systems, a heuristic for bandwidth or profile reduction is often used so that the corresponding coefficient matrix A will have narrow bandwidth and small profile. Thus, heuristics for bandwidth and profile reductions are used to achieve low processing and storage costs for solving large sparse linear systems [14,17]. In particular, profile reduction is employed to reduce storage costs of applications that employ the skyline data structure [9] to represent large-scale matrices.

Let $A = [a_{ij}]$ be a symmetric sparse $n \times n$ matrix. The bandwidth of line i is $\beta_i(A) = i - min(j : (1 \leq j < i)\ a_{ij} \neq 0)$. Bandwidth of A is defined as $\beta(A) = \max[(1 \leq i \leq n)\ \beta_i(A)] = \max[(1 \leq i \leq n)\ (i - \min[j : (1 \leq j < i)] \mid a_{ij} \neq 0)]$. The profile of A is defined as $profile(A) = \sum_{i=1}^{n} \beta_i(A)$. The bandwidth and profile minimization problems are NP-hard [28,31]. Since these problems have associations with an extensive variety of other problems in scientific and engineering disciplines, several heuristics for bandwidth and profile reductions have been proposed for reordering the rows and columns of sparse matrices to solve the bandwidth and profile reduction problems.

A prominent algorithm for solving large-scale sparse linear systems is the Conjugate Gradient Method (CGM) [21,26]. Duff and Meurant [8] showed that a local ordering of the vertices of the corresponding graph of A can improve cache hit rates so that a computational cost reduction of the CGM is reached. Moreover, Burgess and Giles [3] and Das et al. [6] showed that such local ordering can be attained by using a heuristic for bandwidth reduction. Moreover, one should employ an ordering which does not lead to an increase of the number of iterations of the CGM when a preconditioner is applied [15].

In this work, we analyze cases where selected heuristics for bandwidth or profile reduction may not reduce the computational times of the Jacobi-preconditioned CGM (JPCGM). In previous publications [14,16], we showed preceding results and based on this experience [2,5,15,17], 13 heuristics were selected as the most promising methods in this field. Thus, the main objective of this work is to analyze the results of 13 potential state-of-the-art low-cost heuristics for bandwidth and profile reductions (that were selected from systematic reviews [2,5,15,17]) when executed to reduce the computational cost of the JPCGM using high-precision floating-point arithmetic.

Section 2 describes the systematic reviews accomplished to identify the potential best low-cost heuristics for bandwidth and profile reductions. Section 3 describes how the numerical experiments were conducted in this study. Section 4 shows the results. Finally, Sect. 5 addresses the conclusions.

2 Systematic Reviews

As described, since the bandwidth and profile reduction problems have connections with a wide range of other problems in scientific and engineering disciplines, a large number of heuristics for bandwidth and profile reductions has been proposed. In systematic reviews, 133 heuristics for bandwidth and/or profile reductions were identified [2,5,15,17], published between the 1960s and the present day, including a recent proposed heuristic for bandwidth and profile reductions [13]. From the analysis performed, respectively, seven and six heuristics for bandwidth and profile reductions were selected to be evaluated in this computational experiment as potentially being the best low-cost heuristics for bandwidth (Burgess-Lai [4], FNCHC [27], GGPS [38], VNS-band [30], hGPHH [24], CSS-band [23]) or profile (Snay [35], Sloan [34], Medeiros-Pimenta-Goldenberg (MPG) [29], NSloan [25], Sloan-MGPS [32]) reduction. The Reverse Cuthill-McKee method with starting pseudo-peripheral vertex given by the George-Liu algorithm (RCM-GL) [10] was selected in both systematic reviews of heuristics for bandwidth and profile reductions. In particular, the RCM-GL method [10] is contained in the Matlab software [36]. Therefore, from the 133 identified heuristics for bandwidth and profile reduction, 12 were selected to be evaluated in this computational experiment because no other simulation or comparison showed that these 12 heuristics could be superseded by any other heuristics in the analyzed papers, concerning bandwidth or profile reduction, when the computation costs of the given heuristic were also considered. Thus, these 12 heuristics could be deemed as the most promising low-cost heuristics to solve the problems.

The GPS heuristic [12] was not selected in these systematic reviews. In spite of this, it was also implemented and its results were compared with these 12 heuristics in this computational experiment because it is one of the most classic low-cost heuristics evaluated in the field for both bandwidth and profile reductions. Thus, 13 heuristics were implemented and/or evaluated in this work.

3 Description of the Tests, Implementation of the Heuristics, Testing, and Calibration

A 64-bit executable program of the VNS-band heuristic (which was kindly provided by one of the heuristic's authors) was used. This executable only runs with instances up to 500,000 vertices.

The FNCHC-heuristic source code was also kindly provided by one of the heuristic's authors. With this, the source code was converted and implemented in this present work using the C++ programming language.

The 11 other heuristics' authors were requested for the sources and/or executables of their algorithms. Some authors informed that they no longer had the source code or executable, some authors did not answer, and other authors explained that the programs could not be provided. Then, the 11 other heuristics were also implemented using the C++ programming language so that the computational costs of the heuristics could be properly compared [15]. Specifically, the g++ version 4.8.2 compiler was used.

The IEEE 754 double-precision binary floating-point arithmetic is composed of 11 bits of exponent (ranging between 10^{-307} and 10^{307}) and a matissa comprised of 53 bits, which describes approximately 16 decimal digits. Nowadays, this double-precision floating-point arithmetic is adequately accurate for most scientific computing applications. Nonetheless, for some scientific applications, the 64-bit IEEE arithmetic is no longer suitable for today's large-scale numerical simulations. Thus, some relevant scientific applications require high-precision floating-point computations. High-precision floating-point arithmetic is used in applications where the execution time of arithmetic is not a limiting factor, or where accurate results with many digits in the mantissa are needed. Some of these applications demand a significand of 64 bits or more to reach numerically useful results. These applications derive from numerous scientific applications, such as climate modeling, computational fluid dynamics (CFD) problems (e.g. vortex roll-up simulations), computational geometry, mesh generation, computational number theory, Coulomb N-body atomic system simulations, experimental mathematics, large-scale physical simulations performed on highly parallel supercomputers (e.g. studies of the fine structure constant of physics), and quantum theory [1]. Particularly, mesh generation, contour mapping, and other computational geometry applications substantially trust on highly precise arithmetic, mostly when the domain is the unit cube. The reason is that small numerical errors can induce geometrically questionable results. Such troubles are latent in the mathematics of the formulas commonly used in such computations and cannot be repaired without a considerable effort [1]. Specifically, in the applications mentioned, portions of the code normally contain numerically sensitive computations. When using double-precision floating-point arithmetic, these applications may return results with questionable precision, depending on the stopping criteria used. These imprecise results may in turn cause larger errors. On the other hand, it is normally cheaper and more reliable to use high-precision floating-point arithmetic to overcome these troubles [1]. Specifically, in this computational experiment, we used instances derived from meshes generated in discretizations of partial differential equations (that govern CFD problems) by finite volumes [19,20]. Hence, our numerical experiments will focus on high-precision floating-point arithmetic. We used the *GNU Multiple Precision Floating-point Computations with Correct-Rounding* (MFR) library with 256-bit (when using instances originating from discretizations of the Laplace equation) and 512-bit (when using instances contained in the University of Florida sparse matrix collection) precisions.

Many heuristics evaluated here are highly dependent on the starting vertex. Since Koohestani and Poli [24] did not explain which pseudo-peripheral vertex finder was used, the George-Liu algorithm [11] for computing a pseudo-peripheral vertex was used in this computational experiment. Hence, we will refer this heuristic as hGPHH-GL.

It was not our objective that the results of the C++ programming language versions of the heuristics supersede all the results of the original implementations. Our objective was to code reasonably efficient implementations of the heuristics evaluated to make it possible an adequate comparison of the results of the 13 heuristics. However, we tested and calibrated the C++ programming language versions of the heuristics implemented to compare our implementations with the codes used by the original proposers of the heuristics to ensure the codes we implemented were comparable to the algorithms that were originally proposed. We compared the results of the C++ programming language versions of the heuristics with the results presented in the original publications.

In particular, a previous publication [15] shows how the heuristics were implemented, tested, and calibrated. The C++ programming language implementations of the heuristics obtained similar results in bandwidth or profile reductions to the results presented in the original publications (see [15]).

Table 1 shows the characteristics of the five workstations used to perform the simulations. Particularly, the Ubuntu 14.04 LTS 64-bit operating system was used.

Table 1. Characteristics of the machines used to perform the simulations.

Machine	Processor unit: Intel®	Cache memory	Main memory (DDR3)	Linux kernel
M1	Core™ i3-2120 CPU 3.3 GHz	3 MB	8 GB 1.333 GHz	3.13.0-39-generic
M2	Xeon™ E5620 CPU 2.4 GHz	12 MB	24 GB 1.333 GHz	3.13.0-44-generic
M3	Core™ i5-3570 CPU 3.4 GHz	6 MB	8 GB 1.333 GHz	3.13.0-37-generic
M4	Core™ i7-4510U CPU 2.0 GHz	4 MB	8 GB 1.6 GHz	3.16.0-23-generic
M5	Core™ i7-4790K CPU 4.0 GHz	8 MB	12 GB 1.6 GHz	3.19.0-31-generic

Three sequential runs, with both a reordering algorithm and with the JPCGM, were carried out with each instance. In addition, for this experimental analysis of 13 low-cost heuristics for bandwidth and profile reductions, we followed the suggestions given by Johnson [22], aiming at reducing the computational cost of the JPCGM.

4 Numerical Experiments and Analysis

This section shows the results obtained in simulations using the JPCGM, executed after applying heuristics for bandwidth and profile reductions. Section 4.1 shows the results of the resolutions of linear systems arising from the discretization of the Laplace equation by finite volumes [19]. Section 4.2 shows the results of the resolutions of linear systems contained in the University of Florida sparse matrix collection [7].

Tables in this section show the dimension n of the respective coefficients matrix of the linear system (or the number of vertices of the graph associated with the coefficient matrix on it or the name of the instance contained in the University of Florida sparse matrix collection), the name of the reordering algorithms applied, the results with respect to profile and bandwidth reductions, the average results of the heuristics in relation to the computational cost, in seconds (s), and the memory requirements, in mebibytes (MiB). In addition, these tables show the number of iterations and the total computational cost, in seconds, of the JPCGM. Furthermore, in spite of the small number of executions for each heuristic in each instance, these tables show the standard deviation (σ) and coefficient of variation (C_v), referring to the total computational cost of the JPCGM. Additionally, these tables show "–" in the first row of a set of simulations performed with each instance. This means that no reordering algorithm was used. With this result, one can verify the speed-down of the JPCGM attained when using a heuristic for bandwidth or profile reduction, shown in the last columns of these tables. In the tables below, numbers in bold face are the best results (up to two occurrences) in the β, *profile*, t(s), and m.(MiB) columns. Figures in this section are presented as line charts for clarity.

4.1 Instances Originating from the Discretization of the Laplace Equation by Finite Volumes

This section shows the results of the resolutions of linear systems arising from the discretization of the Laplace equation by finite volumes [19]. These linear systems are divided into two datasets: seven and eight linear systems ranging from 7,322 to 277,118 and from 16,922 to 1,115,004 unknowns comprised of matrices with random order [see Fig. 1 and Tables 2 and 3 (with executions performed on the M1 machine)] and originally ordered using a sequence given by the Sierpiński-like curve [18,37] [see Fig. 2 and Tables 4 and 5 (with executions performed on the M2 machine)], respectively.

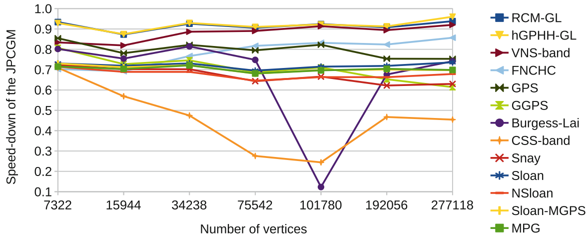

Fig. 1. Speed-downs of the JPCGM obtained using several heuristics for bandwidth and profile reductions applied to linear systems originating from the discretization of the Laplace equation by finite volumes and composed of matrices with random order (see Tables 2 and 3).

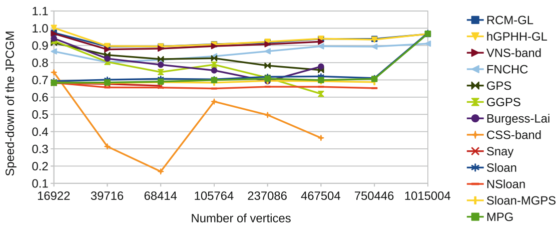

Fig. 2. Speed-downs of the JPCGM obtained using several heuristics for bandwidth and profile reductions applied to linear systems originating from the discretization of the Laplace equation by finite volumes and composed of matrices with a sequence given by the Sierpiński-like order (see Tables 4 and 5).

Tables 2, 3, 4 and 5 show that Sloan's heuristic almost always obtained the best profile results in these datasets. In addition, these tables show that the FNCHC heuristic achieved in general the best bandwidth results, but closely followed by the RCM-GL and hGPHH-GL heuristics, which presented much lower

Table 2. Resolution of three linear systems (derived from the discretization of the Laplace equation by finite volumes and composed of matrices with random order) using the JPCGM and vertices labeled by heuristics for bandwidth and profile reductions.

n	Heuristic	β	*profile*	Heuristic		JPCGM		σ	C_v (%)	Speed-down
				t(s)	m.(MiB)	iter.	t(s)			
7322	–	7248	16083808	–	–	498	**10**	0.02	0.16	–
	RCM-GL	80	396652	0.005	0.0	498	11	0.04	0.37	0.93
	hGPHH-GL	80	406461	0.006	0.0	498	11	0.03	0.29	0.93
	VNS-band	1599	966638	1.061	75.9	498	11	0.31	2.73	0.83
	FNCHC	**75**	444803	2.273	0.5	498	13	0.05	0.36	0.70
	GPS	78	404414	0.362	0.2	498	12	0.40	3.26	0.85
	GGPS	79	397534	0.415	1.3	498	13	0.04	0.30	0.81
	Burgess-Lai	152	407458	0.189	0.0	498	13	0.03	0.20	0.80
	CSS-band	7190	16103602	1.079	40.9	498	14	0.06	0.48	0.71
	Snay	1859	447914	0.240	0.3	498	14	0.10	0.03	0.72
	Sloan	391	375254	0.010	0.3	498	14	0.05	0.38	0.73
	NSloan	197	424476	**0.004**	0.3	498	15	0.01	0.07	0.72
	Sloan-MGPS	312	**374571**	0.030	0.2	498	14	0.02	0.14	0.73
	MPG	800	608671	0.010	0.2	498	15	0.03	0.17	0.72
15944	–	15902	76482022	–	–	745	**34**	0.21	0.63	–
	RCM-GL	121	1149442	0.020	0.0	745	39	0.04	0.10	0.87
	hGPHH-GL	124	1231692	0.020	0.0	745	39	0.29	0.76	0.88
	VNS-band	3916	5108940	1.190	196.7	745	40	0.49	1.23	0.82
	FNCHC	**113**	1321180	5.850	1.3	745	43	0.11	0.26	0.69
	GPS	118	1154030	1.580	0.5	745	42	1.38	3.31	0.78
	GGPS	118	1210195	3.550	2.8	745	43	0.07	0.17	0.73
	Burgess-Lai	212	1144254	1.090	0.0	745	44	0.07	0.16	0.75
	CSS-band	15749	77021429	8.560	192.7	745	51	0.33	0.65	0.57
	Snay	5862	1586436	1.020	0.5	745	47	0.06	0.13	0.70
	Sloan	484	**982693**	0.020	0.3	745	47	0.02	0.04	0.72
	NSloan	218	1222337	**0.010**	0.3	745	49	0.05	0.11	0.69
	Sloan-MGPS	481	1002661	0.100	0.3	745	47	0.14	0.30	0.71
	MPG	1277	1612396	0.030	0.3	745	48	0.20	0.41	0.70
34238	–	34059	357518296	–	–	1069	**105**	0.41	0.39	–
	RCM-GL	194	3411077	0.040	0.0	1069	114	0.90	0.79	0.93
	hGPHH-GL	192	3759478	0.040	0.0	1069	113	0.11	0.10	0.93
	VNS-band	2726	6767128	2.660	490.6	1069	116	1.23	1.06	0.89
	FNCHC	192	3913543	15.440	4.0	1069	122	0.38	0.31	0.77
	GPS	191	3545656	9.720	1.2	1069	118	0.59	0.50	0.82
	GGPS	**170**	3415253	19.690	5.2	1069	122	0.11	0.09	0.75
	Burgess-Lai	334	3282297	4.310	0.0	1069	125	0.29	0.23	0.81
	CSS-band	33923	359144453	67.280	910.7	1069	155	0.63	0.41	0.48
	Snay	21625	5150148	4.830	1.0	1069	145	0.53	0.36	0.70
	Sloan	917	**2578022**	0.060	0.8	1069	144	0.03	0.02	0.73
	NSloan	357	3608666	**0.030**	0.9	1069	153	0.14	0.09	0.69
	Sloan-MGPS	795	2671240	0.300	0.9	1069	146	0.07	0.05	0.72
	MPG	2322	3986576	0.060	0.9	1069	146	0.04	0.03	0.72

Table 3. Resolution of four linear systems (derived from the discretization of the Laplace equation by finite volumes and composed of matrices with random order) using the JPCGM and vertices labeled by heuristics for bandwidth and profile reductions.

n	Heuristic	β	*profile*	Heuristic		JPCGM		σ	C_v (%)	Speed-down
				t(s)	m.(MiB)	iter.	t(s)			
75542	–	75490	1744941733	–	–	1540	**328**	0.38	0.11	–
	RCM-GL	274	12086129	0.1	0	1540	362	0.62	0.17	0.91
	hGPHH-GL	277	13793938	0.1	0	1540	360	0.22	0.06	0.91
	VNS-band	21310	42564399	3.7	1198	1540	365	1.35	0.37	0.89
	FNCHC	**269**	13978910	40.8	7	1540	361	1.66	0.46	0.82
	GPS	272	12086603	41.5	4	1540	371	1.52	0.41	0.80
	GGPS	271	12405895	86.7	10	1540	391	0.44	0.11	0.69
	Burgess-Lai	460	11175444	42.5	0	1540	396	0.64	0.16	0.75
	CSS-band	74879	1747422045	692.6	3819	1540	497	1.33	0.27	0.28
	Snay	47789	28972039	41.7	2	1540	468	0.61	0.13	0.64
	Sloan	1521	**8981209**	0.2	1	1540	472	0.21	0.04	0.70
	NSloan	534	12805249	0.1	1	1540	508	0.04	0.01	0.65
	Sloan-MGPS	1236	9245713	1.0	1	1540	481	0.19	0.04	0.68
	MPG	4020	14107424	0.2	1	1540	481	0.06	0.01	0.68
101780	–	101583	3169282786	–	–	2173	**631**	0.36	0.06	–
	RCM-GL	405	21399542	0.1	0	2173	683	1.81	0.26	0.92
	hGPHH-GL	407	24041332	0.1	0	2173	684	0.30	0.04	0.92
	VNS-band	5207	25033097	5.6	1638	2173	685	2.07	0.30	0.91
	FNCHC	**391**	26974311	60.0	10	2173	699	1.41	0.20	0.83
	GPS	405	21399542	73.1	5	2173	694	4.57	0.66	0.82
	GGPS	400	21727818	153.2	16	2173	735	8.73	1.19	0.71
	Burgess-Lai	745	19394495	4385.7	0	2173	741	0.50	0.07	0.12
	CSS-band	101333	3160566736	1638.0	611	2173	944	2.03	0.22	0.24
	Snay	64553	45830895	61.1	3	2173	886	0.12	0.01	0.67
	Sloan	8845	**14909417**	0.3	2	2173	882	2.60	0.29	0.72
	NSloan	7602	21266761	0.1	2	2173	951	2.53	0.27	0.66
	Sloan-MGPS	8420	15400014	1.9	2	2173	902	2.78	0.31	0.70
	MPG	10502	24115880	0.2	2	2173	906	2.50	0.28	0.70
192056	–	191738	11329772559	–	–	2383	**1305**	0.72	0.06	–
	RCM-GL	360	42578191	**0.2**	0	2382	1437	3.39	0.24	0.91
	hGPHH-GL	364	48308977	0.3	0	2383	1429	0.12	0.01	0.91
	VNS-band	11142	99018771	16.5	3195	2383	1443	6.79	0.47	0.89
	FNCHC	**348**	48496246	114.8	21	2383	1469	1.65	0.11	0.82
	GPS	371	41541059	256.0	10	2383	1475	3.57	0.24	0.75
	GGPS	363	42925208	530.8	28	2383	1468	0.63	0.04	0.65
	Burgess-Lai	621	40149530	349.6	0	2383	1580	0.19	0.01	0.68
	CSS-band	191446	2737568773	793.5	1125	2383	1999	5.49	0.27	0.47
	Snay	112715	158031137	262.2	6	2384	1835	6.77	0.37	0.62
	Sloan	1963	**30916653**	0.7	4	2384	1815	0.49	0.03	0.72
	NSloan	750	44537494	**0.2**	4	2384	1968	0.25	0.01	0.66
	Sloan-MGPS	1759	31863871	4.2	4	2384	1858	1.06	0.06	0.70
	MPG	5366	47979879	0.5	4	2384	1853	0.37	0.02	0.70
277118	–	277019	23512579029	–	–	2771	**2236**	2.78	0.12	–
	RCM-GL	421	74726891	0.4	0	2771	2383	5.18	0.22	0.94
	hGPHH-GL	427	84714895	0.4	0	2771	2328	3.04	0.13	0.96
	VNS-band	12132	97666318	32.1	4618	2771	2397	6.11	0.26	0.92
	FNCHC	424	86076670	183.2	27	2771	2426	0.74	0.03	0.86
	GPS	**399**	72378558	510.3	16	2771	2459	12.28	0.50	0.75
	GGPS	420	75610158	1054.7	21	2771	2586	7.44	0.29	0.61
	Burgess-Lai	793	66880423	401.9	0	2771	2614	0.85	0.03	0.74
	CSS-band	276285	23509305627	1606.1	1680	2771	3314	26.97	0.81	0.45
	Snay	107539	310401674	516.6	10	2771	3032	0.75	0.02	0.63
	Sloan	2243	**55586226**	1.2	4	2771	3036	0.68	0.02	0.74
	NSloan	909	77343800	**0.3**	4	2771	3294	0.01	0.01	0.68
	Sloan-MGPS	2084	57032215	7.9	4	2771	3198	2.39	0.07	0.70
	MPG	7281	89227523	0.8	4	2771	3200	1.26	0.04	0.70

Table 4. Resolution of linear systems (ranging from 16,922 to 105,764 unknowns, derived from the discretization of the Laplace equation by finite volumes and composed of matrices ordered using a sequence determined by the Sierpiński-like curve) using the JPCGM and vertices labeled by heuristics for bandwidth and profile reductions.

n	Heuristic	β	*profile*	Heuristic		JPCGM		σ	C_v (%)	Speed-down
				t(s)	m.(MiB)	iter.	t(s)			
16922	–	16921	1710910	–	–	767	51	0.02	0.04	–
	RCM-GL	115	1252527	0.02	0.0	767	53	0.73	1.40	0.974
	hGPHH-GL	119	1321688	0.02	0.0	767	51	0.07	0.14	1.002
	VNS-band	4756	2393029	1.29	144.7	767	52	0.19	0.37	0.969
	FNCHC	**114**	1372628	8.21	2.0	767	51	0.04	0.09	0.865
	GPS	115	1252527	3.25	0.5	767	53	0.24	0.47	0.915
	GGPS	115	1321081	2.31	4.1	767	53	0.13	0.25	0.927
	Burgess-Lai	224	1235707	0.70	0.0	767	54	0.30	0.56	0.941
	CSS-band	16746	85797563	13.45	140.9	767	55	1.06	1.91	0.744
	Snay	6212	1508415	1.56	0.5	767	74	0.11	0.15	0.682
	Sloan	571	**1074251**	0.02	0.5	767	74	0.02	0.03	0.692
	NSloan	229	1336588	**0.01**	0.3	767	75	0.01	0.02	0.684
	Sloan-MGPS	462	1093326	0.12	0.3	767	75	0.05	0.06	0.685
	MPG	1231	1750944	0.03	0.5	767	75	0.08	0.10	0.683
39716	–	39715	6309342	–	–	1144	**188**	0.04	0.02	–
	RCM-GL	195	4376986	0.05	0.0	1144	210	0.52	0.25	0.894
	hGPHH-GL	192	4770829	0.05	0.0	1144	209	0.31	0.15	0.897
	VNS-band	5863	9979067	2.32	327.2	1144	212	1.11	0.52	0.877
	FNCHC	189	5021600	22.89	3.4	1144	211	1.00	0.47	0.803
	GPS	**180**	4464634	9.05	0.8	1144	213	0.65	0.31	0.845
	GGPS	194	4391324	19.90	3.1	1144	214	4.54	2.13	0.804
	Burgess-Lai	335	4156848	6.63	0.0	1144	221	0.15	0.07	0.824
	CSS-band	39346	480512986	341.56	537.9	1144	257	1.69	0.66	0.314
	Snay	22597	6548607	8.67	1.2	1144	269	0.30	0.11	0.677
	Sloan	830	**3342149**	0.08	1.0	1144	268	0.23	0.09	0.701
	NSloan	372	4634523	**0.03**	1.0	1144	286	0.11	0.04	0.656
	Sloan-MGPS	831	3461255	0.43	1.0	1144	275	0.24	0.09	0.682
	MPG	2358	5222214	0.08	1.0	1144	274	0.17	0.06	0.685
68414	–	68413	14882117	–	–	1514	**430**	0.03	0.01	–
	RCM-GL	238	9598308	0.08	0.0	1514	481	1.09	0.23	0.894
	hGPHH-GL	236	10705920	0.08	0.0	1514	481	0.39	0.08	0.894
	VNS-band	516	17030717	4.59	557.4	1514	483	0.93	0.19	0.882
	FNCHC	233	11325816	42.02	5.1	1514	485	1.66	0.34	0.817
	GPS	**225**	9751463	28.90	1.5	1514	495	0.29	0.06	0.821
	GGPS	233	9781546	68.94	6.2	1514	508	0.07	0.01	0.746
	Burgess-Lai	440	8910920	30.69	0.0	1514	515	0.43	0.08	0.788
	CSS-band	67862	1432183654	1962.01	2043.0	1514	591	3.97	0.67	0.168
	Snay	43837	21823074	36.43	2.1	1514	609	0.11	0.02	0.666
	Sloan	1284	**7093207**	0.17	1.0	1514	609	0.16	0.03	0.706
	NSloan	442	10128234	**0.06**	1.0	1514	656	0.10	0.01	0.656
	Sloan-MGPS	1082	7326579	0.96	1.0	1514	625	0.06	0.01	0.687
	MPG	2811	11460778	0.16	1.0	1514	623	0.44	0.07	0.690
105764	–	105763	29560801	–	–	1846	**816**	0.04	0.04	–
	RCM-GL	311	18180951	0.13	0.0	1846	899	2.09	0.23	0.907
	hGPHH-GL	309	20753083	0.14	0.0	1846	901	1.75	0.20	0.905
	VNS-band	2809	33762228	9.38	857.2	1846	901	0.36	0.04	0.896
	FNCHC	**289**	21067109	69.00	9.4	1846	905	2.95	0.33	0.837
	GPS	299	18336159	60.95	2.1	1846	927	0.80	0.09	0.826
	GGPS	306	18163269	136.02	7.5	1846	899	0.29	0.03	0.789
	Burgess-Lai	483	16959146	112.25	0.0	1846	968	0.40	0.04	0.755
	CSS-band	105406	3418070351	305.06	494.5	1846	1115	3.78	0.34	0.575
	Sloan	1756	**13247695**	0.32	2.2	1846	1159	0.35	0.03	0.703
	NSloan	602	19321158	**0.10**	1.2	1846	1255	0.76	0.06	0.650
	Sloan-MGPS	1512	13685106	1.88	1.9	1846	1191	0.15	0.01	0.684
	MPG	4447	20523176	0.27	1.2	1846	1167	2.19	0.19	0.699

Table 5. Resolution of four linear systems (derived from the discretization of the Laplace equation by finite volumes and composed of matrices originally ordered using a sequence determined by the Sierpiński-like curve) using the JPCGM and vertices labeled by heuristics for bandwidth and profile reductions.

n	Heuristic	β	*profile*	Heuristic		JPCGM			C_v (%)	Speed-down
				t(s)	m.(MiB)	iter.	t(s)	σ		
237086	–	237085	115804392	–	–	2611	**2629**	8.69	0.33	–
	RCM-GL	391	56621430	0.3	0	2612	2866	7.98	0.28	0.92
	hGPHH-GL	393	64411087	0.3	0	2612	2852	3.88	0.14	0.92
	VNS-band	1783	92303199	40.8	1912	2612	2858	5.97	0.21	0.91
	FNCHC	**388**	64592246	177.0	20	2612	2859	1.31	0.05	0.87
	GPS	392	56476790	392.3	13	2612	2965	5.43	0.18	0.78
	GGPS	389	57030510	820.9	16	2612	2869	11.37	0.40	0.71
	Burgess-Lai	718	52953332	698.3	0	2612	3100	0.90	0.03	0.69
	CSS-band	236418	2682971255	1657.8	1107	2612	3637	25.22	0.69	0.50
	Sloan	2044	**41300807**	1.0	7	2612	3661	3.25	0.09	0.72
	NSloan	812	59082821	**0.2**	6	2612	3981	2.10	0.05	0.66
	Sloan-MGPS	1898	42561396	6.2	6	2612	3777	1.33	0.04	0.70
	MPG	5707	64716730	0.7	6	2612	3718	1.27	0.03	0.71
467504	–	467503	382386929	–	–	3446	**6972**	2.65	0.04	–
	RCM-GL	448	130166482	0.6	0	3446	7434	3.43	0.05	0.94
	hGPHH-GL	445	149299971	0.6	0	3446	7423	6.17	0.08	0.94
	VNS-band	5001	227019725	154.2	4904	3446	7409	2.46	0.03	0.92
	FNCHC	449	156037685	371.9	40	3446	7416	6.92	0.09	0.90
	GPS	455	129684974	1593.1	27	3446	7618	2.46	0.03	0.76
	GGPS	**438**	132748941	3228.4	29	3446	8033	7.73	0.10	0.62
	Burgess-Lai	862	119436630	895.2	0	3446	8071	3.74	0.04	0.78
	CSS-band	466181	2526782462	9693.1	3644	3446	9490	6.99	0.07	0.36
	Sloan	2498	**95551358**	2.4	10	3449	9676	13.96	0.14	0.72
	NSloan	911	135054695	**0.5**	10	3449	10563	35.21	0.33	0.66
	Sloan-MGPS	2251	98187318	14.2	10	3449	10053	49.93	0.50	0.69
	MPG	7842	152632760	1.7	11	3449	9974	66.00	0.66	0.70
750446	–	750445	911516500	–	–	4246	**13660**	5.64	0.07	–
	RCM-GL	**461**	224589050	0.9	**0**	4245	14563	4.39	0.03	0.94
	hGPHH-GL	471	257304543	1.0	**0**	4246	14629	7.03	0.05	0.93
	FNCHC	715	304070803	615.5	60	4245	14676	0.23	0.01	0.89
	Sloan	2441	**164952184**	4.0	21	4232	19223	36.40	0.19	0.71
	NSloan	946	232760320	**0.8**	21	4232	20942	9.65	0.05	0.65
	Sloan-MGPS	2264	169448464	24.4	21	4232	19857	38.68	0.19	0.69
	MPG	7986	265072969	2.8	20	4232	19352	22.96	0.12	0.71
1015004*	–	1015003	1580908606	–	–	4557	**20025**	14.55	0.07	–
	RCM-GL	462	316593383	**1.0**	**0**	4557	20726	84.12	0.41	0.97
	hGPHH-GL	465	363030399	1.1	**0**	4557	20709	33.82	0.16	0.97
	FNCHC	**455**	365943729	819.9	88	4569	21174	10.56	0.05	0.91
	Sloan	2484	**233117499**	4.1	27	4557	20587	37.33	0.18	0.97
	MPG	8107	375103029	2.4	27	4557	20682	38.37	0.19	0.97

*Executions performed on the M3 machine.

computational costs. Nevertheless, no gain was attained regarding the speed-up of the JPCGM when using these heuristics. In particular, the FNCHC heuristic presented a much higher computational cost than the RCM-GL, Sloan's, MPG, NSloan, Sloan-MGPS, and hGPHH-GL heuristics.

A slight speed-up of the JPCGM applied to the linear system composed of 16,922 when using the hGPHH-GL heuristic (see Table 4) was reached, but this gain is marginal. Moreover, a speed-down of the JPCGM was obtained when using the other heuristics for bandwidth and profile reductions applied to the other linear systems.

Tables 4 and 5 do not show results of Snay's heuristic [35] applied to linear systems larger than 68,414 unknowns. Snay's heuristic obtained better results (related to reduce the JPCGM computational cost) than the results of the CSS-band [23] and NSloan [25] heuristics when applied to the linear systems composed of 39,716 and 68,414 unknowns. However, Snay's heuristic performed less favorably than the other heuristics when applied to the linear system comprised of 16,922 unknowns (see Table 4).

The GPS [12], Burgess-Lai [4], GGPS [38], and CSS-band [23] presented higher computational costs than the other heuristics [see t(s)(Heuristic) column in Tables 3 and 5]. Consequently, Table 5 does not show the results of these four heuristics applied to the linear systems composed of 750,446 and 1,015,004 unknowns, keeping in mind that the VNS-band execution program runs with instances up to 500,000 unknowns. Furthermore, Table 5 does not show the results of the NSloan [25] and Sloan-MGPS [32] heuristics applied to the linear system composed of 1,015,004 unknowns because these two heuristics performed less favorably than the five other heuristics when applied to linear systems contained in this dataset.

4.2 Instances Contained in the University of Florida Sparse Matrix Collection

Table 6 provides the characteristics of 11 linear systems (composed of symmetric and positive-definite matrices) contained in the University of Florida sparse matrix collection [7]. Tables 2, 3, 4 and 5 show that the RCM-GL [10], Sloan's [34], MPG [29], NSloan [25], Sloan-MGPS [32], and hGPHH-GL [24] heuristics presented much lower computational costs than the other heuristics evaluated in this computational experiment. Then, these six low-cost heuristics for bandwidth or profile reduction evaluated in this study were applied to the dataset presented in Table 6.

Table 6. Eleven linear systems (composed of symmetric and positive-definite matrices) contained in the University of Florida sparse matrix collection.

Instance	Size	β	profile	Density (%)	Description
nasa1824	1824	239	205547	1.18	Structure from NASA Langley
nasa2910	2910	859	525745	2.06	Structure from NASA Langley
sts4098	4098	3323	5217389	0.43	Finite element structural engineering matrix
nasa4707	4704	423	917562	0.47	Structure from NASA Langley
Pres_Poisson	14822	12583	9789525	0.33	Computational fluid dynamics problem
olafu	16146	593	4951980	0.39	Structure from NASA Langley
raefsky4	19779	11786	19611188	0.34	Buckling problem for container model
nasasrb	54870	893	20311330	0.09	Structure from NASA Langley
thermal1	82654	80916	175625317	0.01	Unstructured finite element steady-state thermal problem
2cubes_sphere	101492	100407	483241271	0.02	Finite element electromagnetics 2 cubes in a sphere
offshore	259789	237738	3588201815	0.01	3D finite element transient electric field diffusion

Tables 7 and 8 and Fig. 3 show the results of the resolutions of 11 linear systems contained in the University of Florida sparse matrix collection using the JPCGM and vertices labeled using heuristics for bandwidth and profile reductions. The hGPHH-GL heuristic obtained the best speed-up of the JPCGM when applied to the *Pres_Poisson* instance (see Table 7). On the other hand, speed-downs of the JPCGM were obtained when using these six heuristics for bandwidth and profile reductions when applied to the 10 other linear systems contained in the University of Florida sparse matrix collection that were used here.

Among the heuristics evaluated, the Sloan-MGPS and Sloan's (RCM-GL) heuristics obtained (almost always) the best profile (bandwidth) results when applied to the instances composed in this dataset. Nevertheless, speed-downs of the JPCGM were obtained when using these heuristics (except the simulation using the *Pres_Poisson* instance).

5 Conclusions

The results of 13 heuristics for bandwidth and profile reductions applied to reduce the computational cost of solving three datasets of linear systems using the Jacobi-preconditioned Conjugate Gradient Method in high-precision floating-point arithmetic are described in this paper. These heuristics were selected from systematic reviews [2,5,15,17].

In experiments using three datasets composed of large-scale linear systems, the hGPHH-GL heuristic performed best when applied to one linear system aiming at reducing the computational cost of the JPCGM (see Table 7). On the other hand, speed-downs of the JPCGM were obtained when applying these 13 heuristics for bandwidth and profile reductions to the other linear systems that were used in this computational experiment. Thus, the attained results show that in certain cases no heuristic for bandwidth or profile reduction can reduce the computational cost of the Jacobi-preconditioned Conjugate Gradient Method when using high-precision numerical computations.

Concerning the set of linear systems arising from the discretization of the Laplace equation by finite volumes comprised of matrices with random order, each vertex has exactly three adjacencies [19]. Probably because of this, relabeling the vertices did not improve cache hit rates.

Regarding the set of linear systems originating from the discretization of the Laplace equation by finite volumes and comprised of matrices originally ordered using a sequence given by the Sierpiński-like curve [19], a large number of cache misses may be occurred after applying heuristics for bandwidth and profile reductions. Probably, the reason is that a space-filling curve already provides an adequate memory-data locality so that a reordering algorithm is not useful in such cases. We applied these 13 heuristics in large-scale linear systems and cache memory is a relevant factor in the execution times of these simulations. Evidence from the experiments described in this paper does allow the assertion that a linear system should be studied carefully before using a heuristic for bandwidth

Table 7. Resolution of seven linear systems contained in the University of Florida sparse matrix collection using the JPCGM and vertices labeled using heuristics for bandwidth and profile reductions

Instance	Machine	Heuristic	β	*profile*	Heuristic		JPCGM		σ	C_v (%)	Speed-up/down
					t(s)	m.(MiB)	iter.	t(s)			
nasa1824	M1	–	**239**	205547	–	–	1350	**24**	0.14	0.58	–
		RCM-GL	282	229770	0.004	0.0	1350	25	0.10	0.38	0.96
		hGPHH-GL	293	291203	0.003	0.0	1350	25	0.14	0.56	0.98
		Sloan	1303	**186725**	0.005	0.0	1350	25	0.17	0.67	0.96
		NSloan	415	284963	**0.002**	0.0	1351	26	0.14	0.54	0.94
		Sloan-MGPS	1102	190128	0.012	0.0	**1347**	25	0.18	0.71	0.97
		MPG	1519	516936	0.010	0.0	1350	26	0.16	0.61	0.94
nasa2910	M1	–	**859**	525745	–	–	1846	**133**	0.32	0.24	–
		RCM-GL	875	522223	0.018	0.0	1846	143	1.02	0.72	0.93
		hGPHH-GL	869	1288759	0.016	0.0	1846	140	1.08	0.77	0.95
		Sloan	2015	**456322**	0.018	0.0	**1839**	138	0.15	0.11	0.97
		NSloan	1327	955899	**0.010**	0.0	1844	145	0.49	0.34	0.92
		Sloan-MGPS	2010	460149	0.016	0.0	1842	139	0.16	0.12	0.96
		MPG	2708	2288760	0.038	0.0	1842	147	0.02	0.01	0.91
sts4098	M4	–	3323	5217389	–	–	590	**20**	0.01	0.03	–
		RCM-GL	**1165**	2084237	0.009	**0.0**	590	22	0.07	0.30	0.87
		hGPHH-GL	1171	2981815	0.008	**0.0**	**588**	22	0.39	1.80	0.90
		Sloan	3195	518163	0.023	0.2	589	21	0.01	0.01	0.95
		NSloan	3020	2505064	**0.007**	0.2	589	22	0.01	0.05	0.90
		Sloan-MGPS	3351	**461998**	0.073	0.2	590	21	0.01	0.03	0.95
		MPG	3729	961548	0.017	0.2	**588**	21	0.08	0.38	0.93
nasa4704	M4	–	423	917562	–	–	4248	**190**	0.08	0.04	–
		RCM-GL	**419**	918658	0.009	0.0	4245	201	0.81	0.40	0.94
		hGPHH-GL	450	1079926	0.009	0.0	4244	202	4.37	2.16	0.94
		Sloan	3084	834354	0.024	0.0	4244	204	0.62	0.30	0.93
		NSloan	678	1076453	**0.005**	0.0	4247	210	2.46	1.17	0.90
		Sloan-MGPS	2753	**808577**	0.056	0.0	4246	203	1.20	0.59	0.93
		MPG	3680	2716364	0.074	0.0	4244	212	0.04	0.02	0.90
Pres_Poiss.	M4	–	12583	9789525	–	–	1009	309	1.27	0.41	–
		RCM-GL	**326**	3009635	0.060	**0.0**	1012	297	0.99	0.33	1.04
		hGPHH-GL	364	3130744	0.059	**0.0**	1009	**293**	0.46	0.16	**1.06**
		Sloan	642	**2827171**	0.066	0.3	1012	295	0.39	0.13	1.05
		NSloan	594	3951006	**0.044**	0.3	1012	328	0.38	0.12	0.94
		Sloan-MGPS	582	2834035	0.156	0.3	1009	294	0.94	0.32	1.05
		MPG	14168	26556694	3.845	0.3	1009	297	0.02	0.01	1.03
olafu	M3	–	593	4951980	–	–	16146	**6833**	5.10	0.07	–
		RCM-GL	**553**	5029301	0.146	**0.0**	16146	7253	32.13	0.44	0.94
		hGPHH-GL	573	5165776	0.132	**0.0**	16146	7189	10.30	0.14	0.95
		Sloan	6173	4768547	0.146	0.3	16146	7154	2.24	0.03	0.96
		NSloan	4760	7334811	**0.105**	0.3	16146	7290	0.58	0.01	0.94
		Sloan-MGPS	7775	**4489770**	0.171	0.3	16146	7219	3.00	0.04	0.95
		MPG	14467	29376748	4.268	0.3	16146	7491	9.28	0.12	0.91
raefsky4	M4	–	11786	19611188	–	–	11245	**5862**	1.05	0.02	–
		RCM-GL	**991**	12553981	0.130	**0.0**	**11157**	6293	3.03	0.05	0.93
		hGPHH-GL	1141	13120923	0.110	**0.0**	11182	6313	397.88	6.30	0.93
		Sloan	6550	8587731	0.180	0.3	11180	6200	29.26	0.47	0.95
		NSloan	2242	15308534	**0.070**	0.3	11246	6786	21.77	0.32	0.86
		Sloan-MGPS	8378	**7841072**	0.340	0.3	11245	6411	4.27	0.07	0.91
		MPG	18201	74604715	6.790	0.3	11248	6803	1.01	0.01	0.86

or profile reduction aiming at reducing the computational cost of the JPCGM (and probably when using a preconditioned CGM or other iterative linear system solver).

Table 8. Resolution of four linear systems contained in the University of Florida sparse matrix collection using the JPCGM and vertices labeled using several heuristics for bandwidth and profile reductions.

Instance	Machine	Heuristic	β	*profile*	Heuristic		JPCGM		σ	C_v (%)	Speed-down
					t(s)	m.(MiB)	iter.	t(s)			
nasasrb	M1	–	893	20311330	–	–	25326	**27902**	2.28	0.01	–
		RCM-GL	**586**	19448635	0.3	**0**	25327	30242	14.94	0.05	0.92
		hGPHH-GL	806	20545002	0.2	**0**	25326	28461	54.27	0.19	0.98
		Sloan	5063	19055047	0.3	1	25320	29893	147.67	0.49	0.93
		NSloan	4865	23564619	**0.1**	1	25326	31307	24.46	0.08	0.89
		Sloan-MGPS	4932	**18682599**	0.6	1	25320	29713	60.37	0.20	0.94
		MPG	45896	346820836	13.7	1	25326	32795	61.17	0.19	0.85
thermal1	M5	–	80916	175625317	–	–	1885	**456**	0.75	0.17	–
		RCM-GL	**220**	12017373	0.1	**0**	1885	562	1.21	0.22	0.81
		hGPHH-GL	240	12997244	0.1	**0**	1885	556	0.26	0.05	0.82
		Sloan	889	**10487409**	0.2	3	1885	537	0.19	0.04	0.84
		NSloan	429	13393908	0.1	3	1885	594	0.12	0.02	0.77
		SloanMGPS	661	10677120	0.7	3	1885	559	0.09	0.02	0.81
		MPG	16857	10958622	0.2	3	1885	529	0.65	0.12	0.86
2cubes_sph.	M5	–	100407	483241271	–	–	33	**14**	0.05	0.38	–
		RCM-GL	4709	268149672	0.3	**0**	33	18	0.05	0.29	0.78
		hGPHH-GL	**4693**	345191689	0.3	**0**	33	18	0.07	0.37	0.78
		Sloan	11186	**186478091**	14.7	2	33	17	0.04	0.23	0.43
		NSloan	9203	346819754	**0.2**	2	33	18	0.12	0.68	0.75
		SloanMGPS	13446	200449820	5.8	2	33	17	0.04	0.21	0.60
		MPG	95371	460302437	21.5	2	33	18	0.02	0.13	0.35
offshore	M5	–	237738	3588201815	–	–	1226	**1952**	1.73	0.09	–
		RCM-GL	**21035**	2634951939	0.7	**0**	1226	2494	0.30	0.01	0.78
		hGPHH-GL	23859	3897866179	0.7	**0**	1228	2519	9.04	0.36	0.78
		Sloan	121957	1837918281	72.1	5	1237	2417	10.46	0.43	0.78
		NSloan	102633	3264868562	0.7	5	1226	2638	11.08	0.42	0.74
		SloanMGPS	124658	**1510670726**	141.6	5	1226	2489	9.35	0.38	0.74
		MPG	253828	4262147507	260.5	5	1230	2438	1.25	0.05	0.72

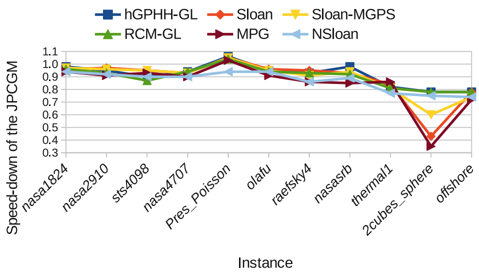

Fig. 3. Speed-downs of the JPCGM obtained using six heuristics for bandwidth and profile reductions applied to 11 linear systems contained in the University of Florida sparse matrix collection (see Tables 7 and 8).

As a continuation of this work, we intend to implement and evaluate the following preconditioners: Algebraic Multigrid, incomplete Cholesky factorization, threshold-based incomplete LU (ILUT), Successive Over-Relaxation (SOR), Symmetric SOR, and Gauss-Seidel. To provide more specific detail, we intend to study the effectiveness of the strategies when using incomplete or approximate factorization based preconditioners as well approximate inverse preconditioners. These techniques shall be used as preconditioners of the Conjugate Gradient Method and the Generalized Minimal Residual (GMRES) method [33] to evaluate their computational performance in conjunction with heuristics for bandwidth and profile reductions. Parallel strategies of the above algorithms will also be studied.

Extended (256-bit and 512-bit) precision was employed in this work. This reduces rounding errors. However, it increases the execution times by a large factor and it may not be performed when solving certain real problems. We intend to examine what occurs in double-precision arithmetic in future studies.

Acknowledgments. This work was undertaken with the support of the Fapemig - Fundação de Amparo à Pesquisa do Estado de Minas Gerais. The authors would like to thank respectively Prof. Dr. Dragan Urosevic, from the Mathematical Institute SANU, and Prof. Dr. Fei Xiao, from Beepi, for sending us the VNS-band executable programs, and the source code of the FNCHC heuristic. In addition, we would like to thank the reviewers for their valuable comments and suggestions.

References

1. Bailey, D.H.: High-precision floating-point arithmetic in scientific computation. Comput. Sci. Eng. **7**(3), 54–61 (2005)
2. Bernardes, J.A.B., Gonzaga de Oliveira, S.L.: A systematic review of heuristics for profile reduction of symmetric matrices. Procedia Comput. Sci. **51**, 221–230 (2015). (International Conference on Computational Science, ICCS)
3. Burgess, D.A., Giles, M.: Renumbering unstructured grids to improve the performance of codes on hierarchial memory machines. Adv. Eng. Softw. **28**(3), 189–201 (1997)
4. Burgess, I.W., Lai, P.K.F.: A new node renumbering algorithm for bandwidth reduction. Int. J. Numer. Methods Eng. **23**, 1693–1704 (1986)
5. Chagas, G.O., Gonzaga de Oliveira, S.L.: Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: a systematic review. Procedia Comput. Sci. (ICCS) **51**, 211–220 (2015)
6. Das, R., Mavriplis, D.J., Saltz, J.H., Gupta, S.K., Ponnusamy, R.: Design and implementation of a parallel unstructured Euler solver using software primitives. AIAA J. **32**(3), 489–496 (1994)
7. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1:1–1:25 (2011)
8. Duff, I.S., Meurant, G.A.: The effect of ordering on preconditioned conjugate gradients. BIT Numer. Math. **29**(4), 635–657 (1989)
9. Felippa, C.A.: Solution of linear equations with skyline-stored symmetric matrix. Comput. Struct. **5**(1), 13–29 (1975)
10. George, A., Liu, J.W.: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall, Englewood Cliffs (1981)
11. George, A., Liu, J.W.H.: An implementation of a pseudoperipheral node finder. ACM Trans. Math. Softw. **5**(3), 284–295 (1979)
12. 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)

13. Gonzaga de Oliveira, S.L., 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). doi:10.1007/978-3-319-42085-1_5
14. 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. Data Min. **12**(2), 190–209 (2017). http://dx.doi.org/10.1504/IJBIDM.2017.10004158
15. 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. (2016). doi:10.1007/s40314-016-0394-9
16. 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: The XLVIII of the Brazilian Symposium of Operations Research (SBPO), Vitória, Brazil, September 2016
17. 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, August 2015. Sobrapo
18. Gonzaga de Oliveira, S.L., Kischinhevsky, M.: Sierpiński curve for total ordering of a graph-based adaptive simplicial-mesh refinement for finite volume discretizations. In: Proceedings of the Brazilian National Conference on Computational and Applied Mathematics (CNMAC), pp. 581–585, Belém, Brazil (2008)
19. 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)
20. Gonzaga de Oliveira, S.L., 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). doi:10.1007/978-3-319-21404-7_37
21. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**(36), 409–436 (1952)
22. Johnson, D.: A theoretician's guide to the experimental analysis of algorithms. In: Goldwasser, M., Johnson, D.S., McGeoch, C.C., (eds.) Proceedings of the 5th and 6th DIMACS Implementation Challenges, Providence (2002)
23. Kaveh, A., Sharafi, P.: Ordering for bandwidth and profile minimization problems via charged system search algorithm. IJST-T Civ. Eng. **36**(2), 39–52 (2012)
24. 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). doi:10.1007/978-1-4471-2318-7_7
25. Kumfert, G., Pothen, A.: Two improved algorithms for envelope and wavefront reduction. BIT Numer. Math. **37**(3), 559–590 (1997)
26. Lanczos, C.: Solutions of systems of linear equations by minimized iterations. J. Res. Natl. Bur. Stand. **49**(1), 33–53 (1952)
27. Lim, A., Rodrigues, B., Xiao, F.: A fast algorithm for bandwidth minimization. Int. J. Artif. Intell. Tools **3**, 537–544 (2007)
28. Lin, Y.X., Yuan, J.J.: Profile minimization problem for matrices and graphs. Acta Mathematicae Applicatae Sinica **10**(1), 107–122 (1994)

29. 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)
30. 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)
31. Papadimitriou, C.H.: The NP-completeness of bandwidth minimization problem. Comput. J. **16**, 177–192 (1976)
32. Reid, J.K., Scott, J.A.: Ordering symmetric sparse matrices for small profile and wavefront. Int. J. Numer. Methods Eng. **45**(12), 1737–1755 (1999)
33. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Comput. **7**, 856–869 (1986)
34. Sloan, S.W.: A Fortran program for profile and wavefront reduction. Int. J. Numer. Methods Eng. **28**(11), 2651–2679 (1989)
35. Snay, R.A.: Reducing the profile of sparse symmetric matrices. Bulletin Geodésique **50**(4), 341–352 (1976)
36. The MathWorks, Inc.: MATLAB, 1994–2015. http://www.mathworks.com/products/matlab
37. Velho, L., Figueiredo, L.H., Gomes, J.: Hierarchical generalized triangle strips. Vis. Comput. **15**(1), 21–35 (1999)
38. 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. **90**, 121–136 (2009)