Facilities Council

The Effects of Scalings on the Performance of a Sparse Symmetric Indefinite Solver

J. D. Hogg, J. A. Scott

March 4, 2008

RAL-TR-2008-007

Library and Information Services
SFTC Rutherford Appleton Laboratory
Harwell Science and Innovation Campus
Didcot
OX11 0QX
UK
Tel: +44 (0)1235 445384
Fax: +44(0)1235 446403
Email: library@rl.ac.uk

ISSN 1358-6254

RAL-TR-2008-007

The Effects of Scalings on the Performance of a Sparse Symmetric Indefinite Solver

J. D. Hogg[1], J. A. Scott[1]

ABSTRACT

Scaling is an important part of solving large sparse symmetric linear systems. In a direct method where the analysis is based only on strucuture it can help by reducing the number of delayed pivots and hence the memory required, the size of the computed factors, and total solution time.

In this paper, we examine the effects of scaling on the performance of a sparse symmetric indefinite solver than implements a multifrontal algorithm. We compare several scalings from the mathematical software library HSL, using a large test set of 367 problems from a wide range of practical applications.

Keywords: scaling, sparse symmetric matrices, sparse direct methods, indefinite factorizations, multifrontal approach

AMS(MOS) subject classifications: 65F35, 65F05, 65F50

---

[1] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, UK.
Email: j.d.hogg@rl.ac.uk & j.a.scott@rl.ac.uk
Work supported by EPSRC grant EP/F006535/1
Current reports available from `http://www.numerical.rl.ac.uk/reports/reports.html`.

March 4, 2008

1 Introduction

We are interested in solving linear systems

$$Ax = b, \tag{1}$$

where A is a large sparse symmetric matrix and the right-hand side b is a given dense vector. In the positive definite case, a direct method computes a Cholesky factorization of a permutation of A, that is, $PAP^T = LL^T$, where L is a lower triangular matrix and P is a permutation matrix. In the indefinite case, a factorization $PAP^T = LDL^T$ is computed, where L is now unit lower triangular and D is a block diagonal matrix with blocks of order 1 or 2. The solution process is completed by performing forward and back substitutions (that is, by first solving a lower triangular system and then an upper triangular system).

Most modern direct solvers have a number of distinct phases. The analyse phase optionally chooses a pivot sequence and, using only the sparsity pattern of A, predicts the non-zero pattern of the factors for the chosen pivot sequence. The factorize phase computes the numerical factorization using the data structures set up by the analyse phase. The forward and back substitutions are performed by the solve phase, which may be called repeatedly for different right-hand sides b. In general, to maintain numerical stability, it is necessary to modify the pivot sequence during the factorize phase, delaying small pivots until alternatives are available or they are safer to use. These delayed pivots cause additional fill-in of the factors beyond that predicted by the analyse phase and, in addition to providing a logistical overhead, lead to extra numerical work in both the factorize and solve phases.

A technique to recover from inaccuracies in the factorization is to use the computed matrix factorization as a preconditioner for an iterative method, such as iterative refinement or the Flexible Generalized Minimal Residual method (FGMRES) [3].

In this paper, we concentrate on reducing the number of delayed pivots. We can apply a (diagonal) scaling matrix S to (1) symmetrically

$$SASS^{-1}x = Sb.$$

This is equivalent to solving the following system

$$\begin{aligned} \hat{A} &= SAS \\ \hat{b} &= Sb \\ \hat{A}y &= \hat{b} \\ x &= Sy. \end{aligned}$$

If we are able to choose a good scaling, the matrix $\hat{A}$ will be numerically easier to factorize than A. That is, the number of delayed pivots is reduced enabling, in some cases, a factorization to be computed where previously memory (or time) limitations made it impossible.

How to find a good scaling S is still an open question, but a number of scaling heuristics have been proposed and are widely used. In this paper, we experiment with scalings available from the HSL library [14]. In particular, we use the sparse indefinite direct solver `MA57` [10] with the different scalings available in HSL, and use an iterative method to further improve accuracy. This study extends the work of Pralet [16]; our contribution is to carry out a systematic experimental study of these scalings applied to a large number of practical problems, including some highly challenging systems.

2 The Scalings

2.1 No Scaling

Many problems can be solved without the use of scaling and, as our numerical experiments will show, this often results in smaller residuals (before the application of refinement) than if a scaling is used and further, without the overhead of scaling, can lead to the fastest solution times.

2.2 MC30

MC30 is the oldest scaling in HSL and is described as a symmetric adaption of the method described in the paper of Curtis and Reid [6]. It scales $A = \{a_{ij}\}$ to the matrix $\hat{A} = \{\hat{a}_{ij}\}$ where

$$\hat{a}_{ij} = a_{ij} \exp(s_i + s_j),$$

with the aim of minimizing the sum of the squares of logarithms of the absolute values of the entries:

$$\begin{aligned} & \min_{\boldsymbol{s}} \quad \sum_{a_{ij} \neq 0} \left(\log |\hat{a}_{ij}|\right)^2 \\ = \quad & \min_{\boldsymbol{s}} \quad \sum_{a_{ij} \neq 0} \left(\log |a_{ij}| + s_i + s_j\right)^2 . \end{aligned}$$

This is achieved by a specialized conjugate gradient algorithm.

2.3 MC64

MC64 finds a maximum matching of an unsymmetric matrix such that the largest entries are moved on to the diagonal [11]; this leads to an unsymmetric scaling such that the scaled matrix has all ones on the diagonal and the off-diagonal entries are of modulus less than or equal to one. The approach can be symmetrized by the method of [12], which essentially amounts to initially ignoring the symmetry of the matrix and then averaging the relevant row and column scalings from the unsymmetric permutation.

In recent years, MC64 has been widely used in conjunction with both direct and iterative methods. It is used in the sparse direct solver SuperLU of Demmel and Li [8], where it is particularly advantageous to put large entries on the diagonal because SuperLU implements a static pivoting strategy that does not allow pivots to be delayed but rather adheres to the data structures established by the analyse phase. Benzi, Haws and Tuma [5] report on the beneficial effects of scalings to place large entries on the diagonal when computing incomplete factorization preconditioners for use with Krylov subspace methods. The symmetrized version was developed following the success of MC64 on unsymmetric systems; it is used by default within the HSL 2007 version of MA57.

2.4 MC77

MC77 uses an iterative procedure [18] to attempt to make all row and column norms of the matrix unity for a user-specified geometric norm $\|\cdot\|_p$. We shall consider the infinity and one norms in this paper. The infinity norm is the default within MC77 due to good convergence properties. It produces a matrix whose rows and columns have maximum entry of exactly one. The one norm produces a matrix whose row and column sums are exactly one (a doubly stochastic matrix) and is, in some sense, an optimal scaling [4].

We denote MC77 in the infinity norm and in the one norm by $\texttt{MC77}_\infty$ and $\texttt{MC77}_1$ respectively.

2.5 Hybrid Scalings

We observe that MC64 and $\texttt{MC77}_\infty$ are aiming for similar properties of the matrix — both will produce a matrix with row and column infinity norms close to 1. However, the MC64 scaling will further guarantee, if it is successful, that each row and column maximum will be in a unique row and column. It may be possible to use $\texttt{MC77}_\infty$ as a hot start, that is, to cheaply identify a partial transversal for MC64, but such an algorithmic combination is beyond the scope of this paper. We will however try each scaling as a simple prescaling to all other scalings.

3 Methodology

3.1 Codes used

In addition to the above scaling packages, we will use the following HSL codes.

`MA57` This package implements a multifrontal algorithm and is designed for the solution of large sparse symmetric indefinite systems. For efficiency, Level 3 BLAS are used. A large number of options are offered that can be used to tune the performance for a particular problem class or computer architecture. In our tests, we use the default settings except that (unless stated otherwise) we select the MeTiS ordering [15], we disable the internal call to the scaling routine `MC64`, and we treat all matrices as indefinite (that is, we allow numerical pivoting with a threshold parameter of 0.01).

`MA60` This subroutine uses the following iterative refinement scheme to improve the quality of the solution:

(a) compute the residual $r = Ax - b$;

(b) solve $Ad = r$ for d; and

(c) update $x = x - d$.

These three steps are repeated until a satisfactory error is obtained or the number of cycles of iterative refinement is greater than the maximum allowed. We use the default maximum limit of 16. The termination condition requires the error estimate to be less than machine precision; full details are provided in the user documentation for `MA60`.

`MI15` This routine uses the Flexible Generalized Minimal Residual method with restarts every m iterations, FGMRES(m), to solve the linear system $Ax = b$, optionally using preconditioning. If Gaussian elimination with static pivoting has been used to compute an approximate LU factorization of A, FGMRES(m) can be used to recover full backward error stability (see Arioli, Duff, Gratton and Parlet [3]). In this case, the left preconditioner is chosen to be the identity and the right preconditioner is chosen to be $P_R^{(i)} = P_R = (LU)^{-1}$. In our experiments, we choose the left preconditioner to be the identity and we right precondition by a solve with `MA57`, allowing a maximum of 20 iterations for convergence with a restart at $m = 10$ (These parameters were chosen following numerical experimentation). Our termination condition is the default, except we set the control parameter `CNTL(2)` (the absolute value of the residual) to 10^{-16}.

3.2 Test environment

Our test problems are all taken from the University of Florida Sparse Matrix Collection [7] and were chosen by selecting the real symmetric problems of dimension less than 100,000 that can be factorized by `MA57` on our test machine. This gives us 367 problems, of which 158 are positive semi-definite and 21 are singular; further details are given in Appendix A.

In our experiments, we generated the right hand side b by choosing x to be the vector of all ones and forming $b = Ax$.

All the tests were conducted on a 2.8 GHz Intel Pentium 4 CPU with 1GB of RAM, using the Goto BLAS [13]. The g95 Fortran compiler was used with the option -O2, all denormals were flushed to zero, and all computations were performed in double precision. All reported times are CPU timings in seconds.

3.3 What makes a good scaling?

In order to evaluate different scalings, we have to determine what we are looking for. Ultimately, when solving linear systems, users are interested in a combination of three things:

Ability to solve As we have already observed, for some problems, scaling may be required to make the problem tractable, either in terms of memory, time or accuracy.

Speed In many applications, it is important that the problem is solved as fast as possible. If a direct method is used, solving for multiple right hand sides (or repeatedly solving for one or more right-hand sides) cheaply may also be an essential aspect — with the implication that refinement may be a significant additional cost.

Accuracy The problem needs to be solved to the required degree of accuracy. This will involve avoiding poor pivots during the numerical factorization, using an iterative procedure once the factorization is complete, or a combination of the two. We measure accuracy using the scaled residual

$$\frac{\|Ax - b\|}{\|A\|\|x\| + \|b\|}. \tag{2}$$

In this paper, we use the infinity norm $\|x\|_\infty = \max_i |x_i|$ and its induced matrix norm $\|A\|_\infty = \max_i \sum_j |a_{ij}|$.

3.4 Measures used in this paper

Delayed pivots We use the number of delayed pivots reported by `MA57` as a predictor of speed and, to some extent, numerical stability. Given an initial pivot order, the time and the memory required to factorize and then solve the linear system will depend on the number of delayed pivots. A large number of delayed pivots can also, in some cases, be indicative of a numerically difficult factorization that would be unstable without pivoting.

Scaling time Assuming the required accuracy is attained with and without scaling, a scaling that results in a faster factorization need not be advantageous if the combined time of scaling, factorizing and solving exceeds the unscaled solution time. We are also interested in the relative speeds of the different scaling algorithms.

Factorization time This is the time taken for the numerical factorization phase. For a given pivot order and threshold parameter u, this will depend on the scaling used.

Total time This is total time taken for the scaling followed by the analyse, factorize and solve phases of `MA57` followed by refinement.

MA57 residual This is the residual (2) evaluated using the original unscaled matrix directly after a single solve with `MA57`.

Iterative refinement residual This is the residual after `MA60` (iterative refinement).

FGMRES residual This is the residual after `MI15` (FGMRES(10)).

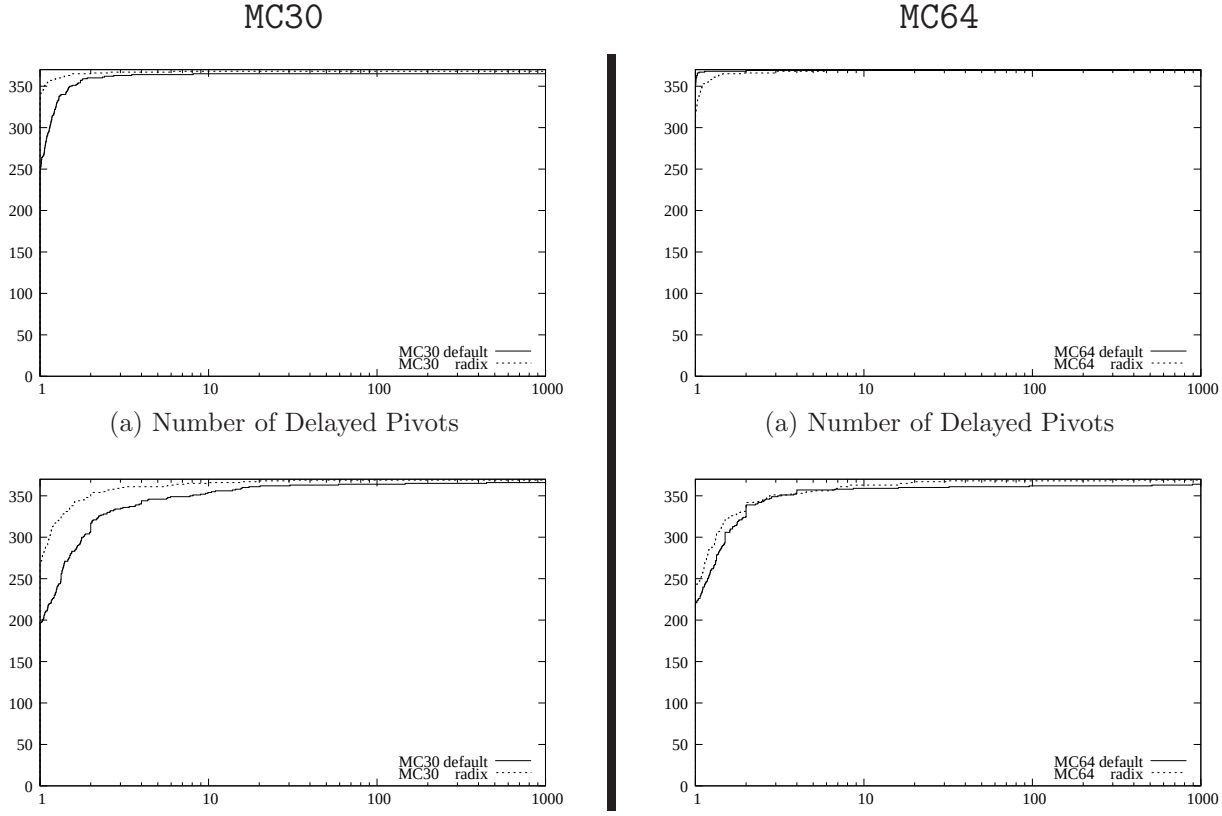
3.5 Performance Profiles

Because we have a large test set, many of our results are presented using performance profiles, as described in [9]. Let $\mathcal{S}$ represent the set of scalings that we wish to compare. Suppose that a given scaling $i \in \mathcal{S}$ reports a statistic $s_{ij} \geq 0$ when run on example j from the test set $\mathcal{T}$, and that the smaller this statistic the better the solver is considered to be. For example, s_{ij} might be the time for problem j using scaling i. For all problems $j \in \mathcal{T}$, we want to compare the performance of scaling i with the performance of the best scaling in the set $\mathcal{S}$.

For $j \in \mathcal{T}$, let $\hat{s}_j = \min\{s_{ij}; i \in \mathcal{S}\}$. Then for $\alpha \geq 1$ and each $i \in \mathcal{S}$ we define

$$k(s_{ij}, \hat{s}_j, \alpha) = \begin{cases} 1 & \text{if } s_{ij} \leq \alpha \hat{s}_j \\ 0 & \text{otherwise.} \end{cases}$$

Figure 1: Performance profile for radix and non-radix (default) scalings.



(a) Number of Delayed Pivots

(a) Number of Delayed Pivots

(b) Residual without iterative refinement

(b) Residual without iterative refinement

The performance profile of solver i is then given by the function

$$p_i(\alpha) = \frac{\sum_{j \in \mathcal{T}} k(s_{ij}, \hat{s}_j, \alpha)}{|\mathcal{T}|}, \quad \alpha \geq 1.$$

As already noted, in this study, the statistics used are timings, the number of delayed pivots and residual sizes. The range of α illustrated is chosen in each case to highlight the dominant trends in the data.

4 Numerical Experiments

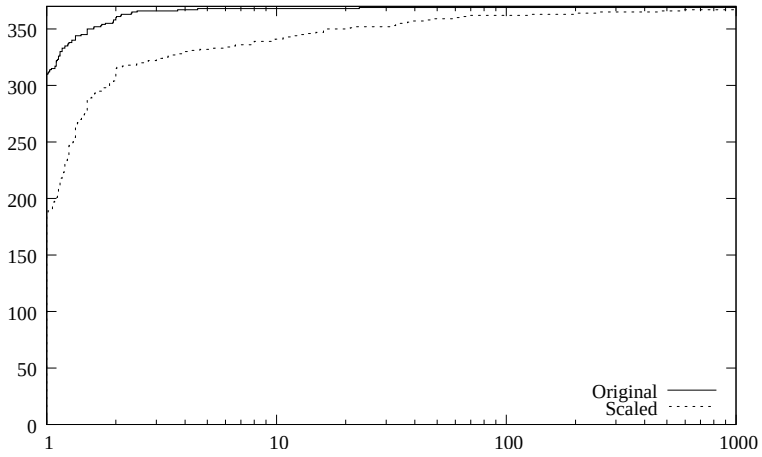
4.1 Scaling by the radix

It was common practice in the past [6] to scale by a power of the radix (on most modern machines, a power of two) in order to eliminate numerical errors due to scaling. This works because it acts as an integer addition on the exponent rather than a floating point calculation, with potential loss of precision. Further such a scaling was faster to apply due to this.

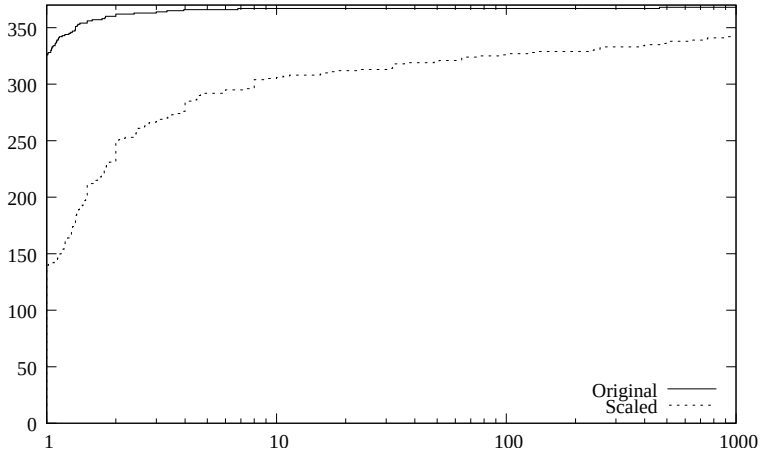
The procedure seems to have fallen out of favour with modern processors being able to apply any scaling cheaply, and as powers of sixteen (the radix on some older machines) were found to be rather too blunt an instrument when the threshold pivoting parameter used by partial pivoting within a direct solver was often chosen to be $u = 0.1$ (these days, $u = 0.01$ is more commonly used and is the default threshold in `MA57`).

In Figure 1, we report on using `MC30`- and `MC64`-based radix and non-radix scalings. These results show that the radix scaling reduces the residual from the `MA57` solver in more cases than it does not, although

Figure 2: Performance profiles of the residuals after iterative refinement and after FGMRES for the original and the MC64 scaled matrices



(a) Iterative Refinement Residual



(b) FGMRES Residual

in most cases the improvement is less than one order of magnitude, and any advantage can be almost eliminated through iterative refinement.

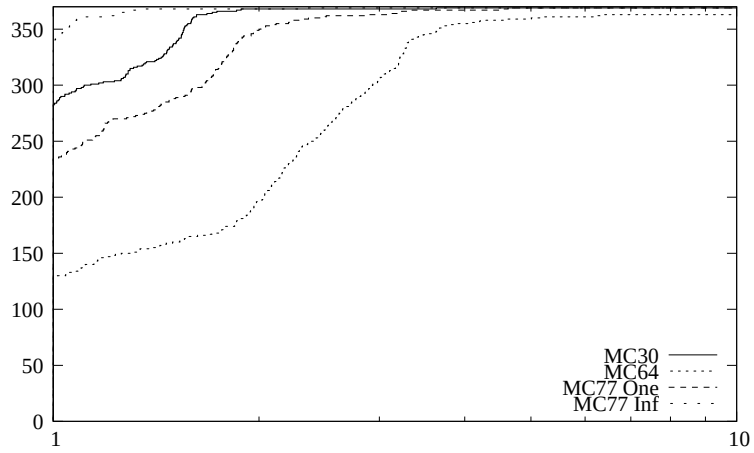
The number of delayed pivots shows an interesting property we have yet to explain. We would expect the non-radix scaling to produce fewer delayed pivots as it has improved the numerics of the problem; this is indeed the case with MC64, however with MC30 we see that the converse is true.

For the remainder of the experiments in this paper, we use a non-radix based scaling.

4.2 Use of unscaled matrix for refinement

It is worth noting that the unscaled matrix A should be retained for any iterative method used to refine the solution computed by the direct solver. Figure 2 demonstrates that using the scaled matrix will, in general, result in a larger residual (as measured with the original matrix). This is easily explained because we are then, in effect, solving a perturbed system.

Figure 3: Performance profile for time to find and apply scaling only



4.3 Numerical comparison of scalings

We assess each of our four main scalings (`MC30`, `MC64`, $\texttt{MC77}_1$, $\texttt{MC77}_\infty$) by itself and in combination with each other scaling (for example, `MC30` followed by `MC64`). We first consider what the 'best' prescaling for each scaling is (ie `MC30` by itself or after `MC64`) and then compare the performances of these best (combination) scalings before making our final recommendations. It is worth noting that, for each scaling, there are some problems on which it is the best, and if it is important to reduce factorization time and/or the amount of fill-in, users may want to try a variety of scalings on a representative sample of their problems before deciding which to use.

In Figure 3, we present a performance profile of the scaling times for the basic scalings. It is clear that $\texttt{MC77}_\infty$ is significantly faster than `MC64`.

4.3.1 `MC30`

We considered `MC30` by itself and prescaled by `MC64`, $\texttt{MC77}_\infty$ and $\texttt{MC77}_1$. The results showed little difference between these scalings, with $\texttt{MC77}_1$ producing marginally fewer delayed pivots and $\texttt{MC77}_\infty$ giving slightly better residuals for a little additional scaling time. However, for a very small minority of problems, $\texttt{MC77}_1$ is slow. We shall not use a prescaling as the overhead does not justify the gain.

4.3.2 `MC64`

Comparing `MC64` prescaled by each of the other scalings we again found that prescaling with `MC77` giving a small advantage for a small additional computational cost.

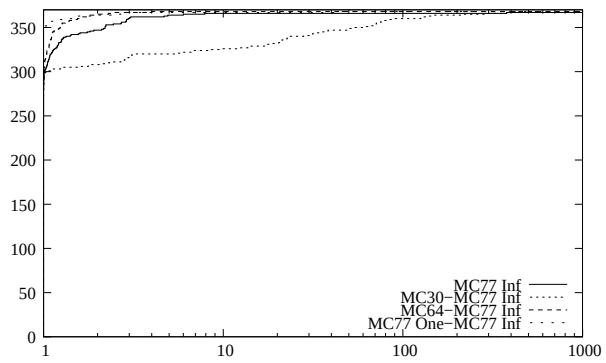
4.3.3 `MC77` Infinity Norm

Comparing the various prescalings for $\texttt{MC77}_\infty$ shows that we should avoid `MC30` as a prescaling, and that `MC64` and $\texttt{MC77}_1$ both reduce the number of delayed pivots. However, use of $\texttt{MC77}_\infty$ by itself gives the smallest `MA57` residuals, although this advantage is eliminated by the use of an iterative method. The performance profiles in Figure 4 illustrate these differences.

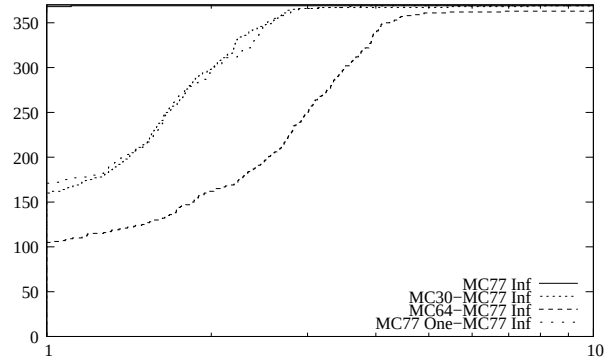
4.3.4 `MC77` One Norm

The use of any prescaling with $\texttt{MC77}_1$ generally impairs its performance, although prescaling with $\texttt{MC77}_\infty$ gives a minor reduction in the `MA57` residual.

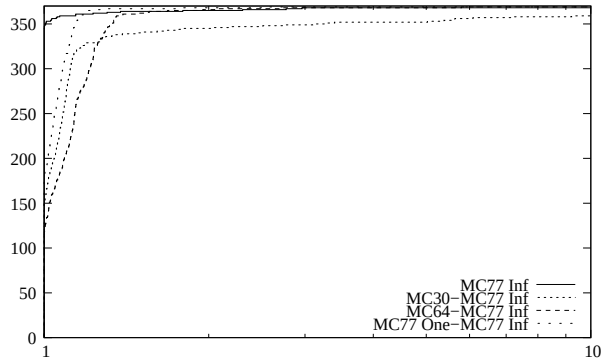
Figure 4: Performance profiles comparing various prescalings for $\texttt{MC77}_\infty$



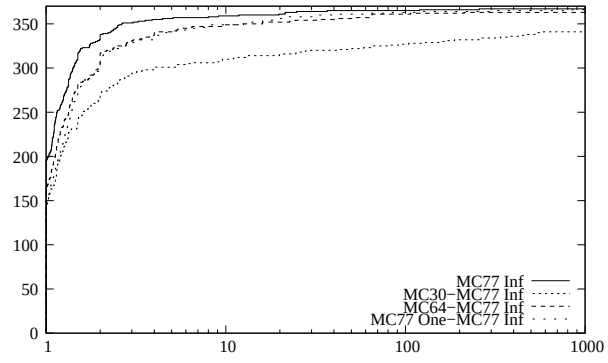
(a) Number of Delayed Pivots



(b) Time for scaling



(c) Total solution time



(d) `MA57` residuals

Figure 5: Number of delayed pivots for best variants

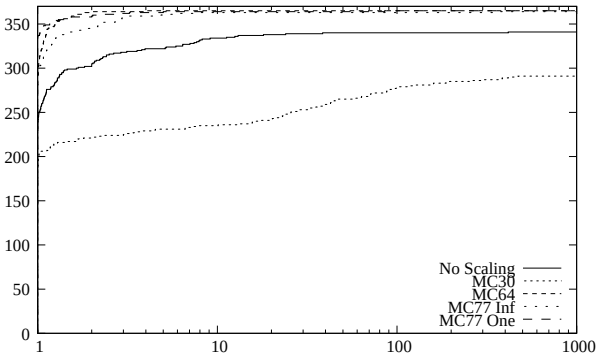


Figure 6: Scaling times for best variants

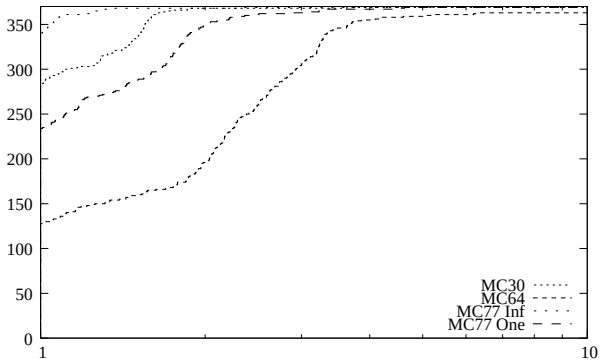


Figure 7: Total times for best variants

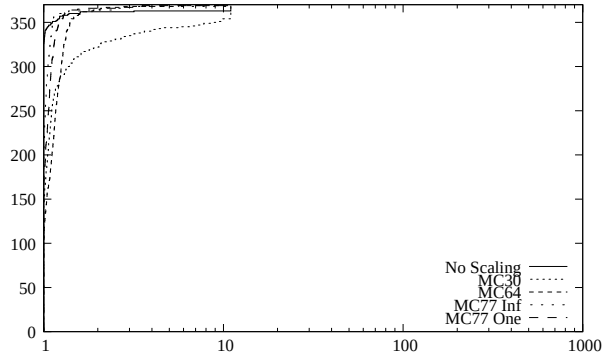
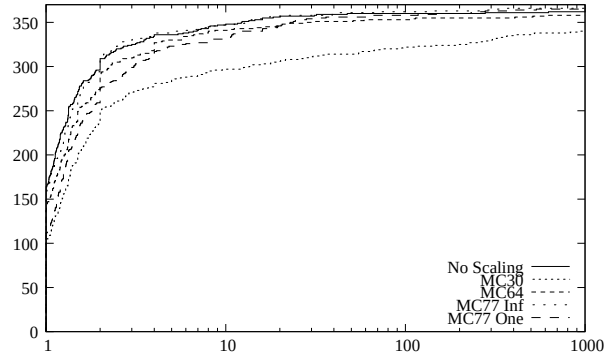


Figure 8: `MA57` residual for best variants



4.3.5 Overall results

In the remainder of this paper we compare the performance of no scaling, `MC30`, `MC64`, $\texttt{MC77}_1$, and $\texttt{MC77}_\infty$ (with no prescaling):

Results are presented in Figures 5–8. Note that, after refinement (with iterative refinement or FGMRES) the residuals are comparable in quality, and, for each problem, none of the scalings needed more than four iterations.

Pralet [16] reports that `MC30` performs poorly on many indefinite problems, and our results support this finding. In terms of the number of delayed pivots, all the other scalings perform far better than no scaling, with comparable results for $\texttt{MC77}_1$ and `MC64`. With respect to the scaling times, `MC64` is slower than the other scalings while $\texttt{MC77}_\infty$ is the fastest. The total time is lower for $\texttt{MC77}_1$ than for the other scalings but `MC64` is slightly more robust (there are a small number of problems on which both the `MC77` scalings do not perform well). If we look at the `MA57` residuals, $\texttt{MC77}_\infty$ performs almost as well as no scaling, while `MC30` generally performs worse (however the difference in the residual quality is eliminated once an iterative method is used).

To avoid distortions due to small problems (that are difficult to reliably time), in Table 1 we concentrate on problems that take at least 0.1 seconds to solve (whichever scaling is used). For each problem j and each scaling i, we record the total time t_{ij} and then compute

$$\alpha_{ij} = t_{ij}/t_j$$

where $t_j = \min\{t_{ij}\}$. A problem is regarded as 'not solved' if it runs out of memory or $\alpha_{ij} > 10$. In Table 1, we report the number of problems for which α_{ij} is at most 1, 1.1, 1.2, etc. For a given scaling,

Table 1: Performance data for the larger problems that require more than 0.1 seconds to solve. For each α interval the best results are highlighted in bold.

α	No scale	MC30	MC64	MC77_∞	MC77_1
1.0	**181**	5	5	30	21
1.1	**212**	110	64	184	133
1.2	214	153	133	**224**	204
1.3	220	165	180	**225**	218
1.4	223	169	208	**227**	**227**
1.5	223	177	220	**228**	**228**
1.6	223	179	224	228	**229**
1.7	225	183	226	228	**230**
1.8	225	185	226	228	**230**
1.9	225	187	228	228	**230**
2.0	225	187	228	228	**230**
Not Solved	6	15	**0**	1	**0**
$\alpha_{\max}$	-	-	**2.83**	-	3.16

(a) MeTiS

α	No scale	MC30	MC64	MC77_∞	MC77_1
1.0	**151**	17	15	37	27
1.1	**180**	58	45	129	93
1.2	**183**	98	81	166	143
1.3	**187**	115	108	181	169
1.4	187	131	131	**189**	185
1.5	188	137	147	**190**	**190**
1.6	189	141	164	192	**193**
1.7	191	148	171	192	**194**
1.8	191	151	185	192	**195**
1.9	191	153	190	192	**195**
2.0	191	153	194	194	**195**
Not Solved	4	15	**0**	**0**	**0**
$\alpha_{\max}$	-	-	3.80	8.94	**3.26**

(b) AMD

Table 2: Ten Interesting Problems

Name	m	Application	Properties
FIDAP/ex14	3973	Fluid Dynamics Finite Element	Indefinite
GHS_indef/bloweybl	30003	Materials Problem	Indefinite, Rank Deficient (rank 30002)
GHS_indef/copter2	55476	Fluid Dynamics Problem	Indefinite
GHS_indef/ncvxqp1	12111	Optimization Augmented System	Indefinite
GHS_indef/ncvxqp9	16554	Optimization Augmented System	Indefinite
Oberwolfach/LFAT5000	19994	Model Reduction	Positive Definite
Schenk_IBMNA/c-30	5321	Non-Linear Optimization	Indefinite
Schenk_IBMNA/c-52	23948	Non-Linear Optimization	Indefinite
Schenk_IBMNA/c-54	31793	Non-Linear Optimization	Indefinite
Schenk_IBMNA/c-62	41731	Non-Linear Optimization	Indefinite

$\alpha_{\max}$ denotes $\max_j \alpha_{ij}$. Data is shown for both Approximate Minimum Degree (AMD) [1, 2] and MeTiS orderings. The MeTiS ordering takes longer to compute and so there are more problems included for MeTiS.

The results show that only $\texttt{MC77}_1$ and `MC64` solve all the problems in a time that is within an order of magnitude of the best (although there is only one problem that $\texttt{MC77}_\infty$ does not succeed on). If we look only at $\alpha \leq 2.0$, $\texttt{MC77}_1$ slightly outperforms `MC64`. It is not clear whether this conclusion will hold if, in the future, we look at larger problems. We will see in the next section that slow total solution times for `MC64` are a result of the scaling taking a significant proportion of the total time, whereas the slowest times for $\texttt{MC77}_1$ are due to the production of a poor scaling that leads to a large number of delayed pivots.

5 Interesting Problems

We now examine more closely ten problems (see Table 2). These were chosen to illustrate where some of the scalings do poorly. We note that all the problems are part of our main test and the Schenk_IBMNA problems are from a larger set of such problems (see Appendix A).

Results for these problems are given in Table 3, and the relative timings for the different phases of `MA57` are shown in Graphs 9 and 10. We display results for both AMD and MeTiS orderings because, for our relatively small problems, the MeTiS time is a large proportion of the total solution time but this is not the case for AMD. Note that, because the analyse phase uses only the sparsity pattern, the analyse timings are independent of the scaling used.

These problems illustrate that, with the exception of `MC30`, there are problems for which each scaling is, by some measure, optimal. They also demonstrate each of the scalings can behave poorly. We note that, following iterative refinement, all residuals are comparable and hence are omitted. The optimization problems, which are characterized by having a mixture of extremely large and extremely small eigenvalues, provide the most challenging systems on which none of the scalings do consistently well.

Let us first consider using no scaling. This, in general, results in small `MA57` residuals (recall Figure 8) and, without the scaling overhead, the total solution time can be small (copter2, c-30, c-52). However there is a penalty in the large number of delayed pivots (LFAT5000, ncvxqp1) and, in the extreme case (bloweybl), this can lead to the problem not being solved because memory is exhausted.

While the results of the previous section tell us that $\texttt{MC77}_1$ is generally the best scaling in terms of delayed pivots, problems ncvxqp1 and c-62 illustrate that its behaviour is erratic and using `MC64` can lead to significantly fewer delayed pivots. On the worst case problem we found (c-62) the impact on the factorize time of the number of delayed pivots was almost a factor of four. On many of the problems,

Table 3: Results for Ten Interesting Problems

	Delayed Pivots					MA57 Residual				
Problem	none	MC30	MC64	MC77$_\infty$	MC77$_1$	none	MC30	MC64	MC77$_\infty$	MC77$_1$
ex14	4825	839	586	735	692	1.17e-16	1.92e-13	4.78e-14	3.69e-14	2.72e-14
bloweybl	-	9652	9652	10435	19559	-	1.37e-12	3.08e-12	7.28e-16	2.75e-13
copter2	120	108	110	118	60	1.27e-12	1.62e-12	1.27e-12	1.38e-12	1.39e-12
ncvxqp1	1.7e5	59697	13184	40234	38788	3.02e-19	8.13e-18	1.26e-13	2.13e-17	3.55e-17
ncvxqp9	6217	6147	2808	6234	3275	1.20e-16	1.20e-16	4.32e-19	6.01e-17	1.20e-16
LFAT5000	46309	13	13	13	13	1.16e-16	1.54e-16	2.19e-16	1.59e-16	1.75e-16
c-30	24	0	1	6	0	6.02e-17	6.96e-16	1.99e-16	3.51e-17	7.13e-16
c-52	950	17116	825	880	738	6.43e-17	1.29e-16	8.04e-18	2.01e-18	1.29e-16
c-54	7135	19012	1881	5395	2681	8.16e-17	1.53e-15	4.92e-14	1.63e-16	8.16e-17
c-62	5.5e5	5.5e5	1333	5.0e5	1.1e5	1.73e-16	1.73e-16	1.86e-14	4.09e-16	6.24e-15

	Scaling Time					Factorize Time				
Problem	none	MC30	MC64	MC77$_\infty$	MC77$_1$	none	MC30	MC64	MC77$_\infty$	MC77$_1$
ex14	0	0.025	0.088	0.016	0.023	0.273	0.033	0.023	0.025	0.023
bloweybl	-	0.040	0.099	0.033	0.038	-	0.069	0.069	0.070	0.142
copter2	0	0.252	0.876	0.213	0.188	4.38	4.60	4.46	4.35	4.37
ncvxqp1	0	0.019	0.636	0.019	0.021	102	14.5	2.34	8.75	8.09
ncvxqp9	0	0.017	0.532	0.020	0.019	0.161	0.160	0.552	0.158	0.063
LFAT5000	0	0.026	0.045	0.021	0.029	22.5	0.020	0.020	0.020	0.020
c-30	0	0.015	0.042	0.014	0.016	0.015	0.014	0.015	0.015	0.014
c-52	0	0.057	0.135	0.056	0.054	0.091	1.23	0.090	0.089	0.088
c-54	0	0.096	1.29	0.093	0.092	0.322	1.63	0.172	0.235	0.187
c-62	0	0.104	0.550	0.140	0.133	751	752	11.9	435	42.4

Figure 9: Showing relative timing dedicated to each part of the solve process, using MeTiS ordering.

Times are normalized by the longest time for each problem (given in parentheses)
From the bottom to the top: analyse, scaling, factorize, iterative refinement.

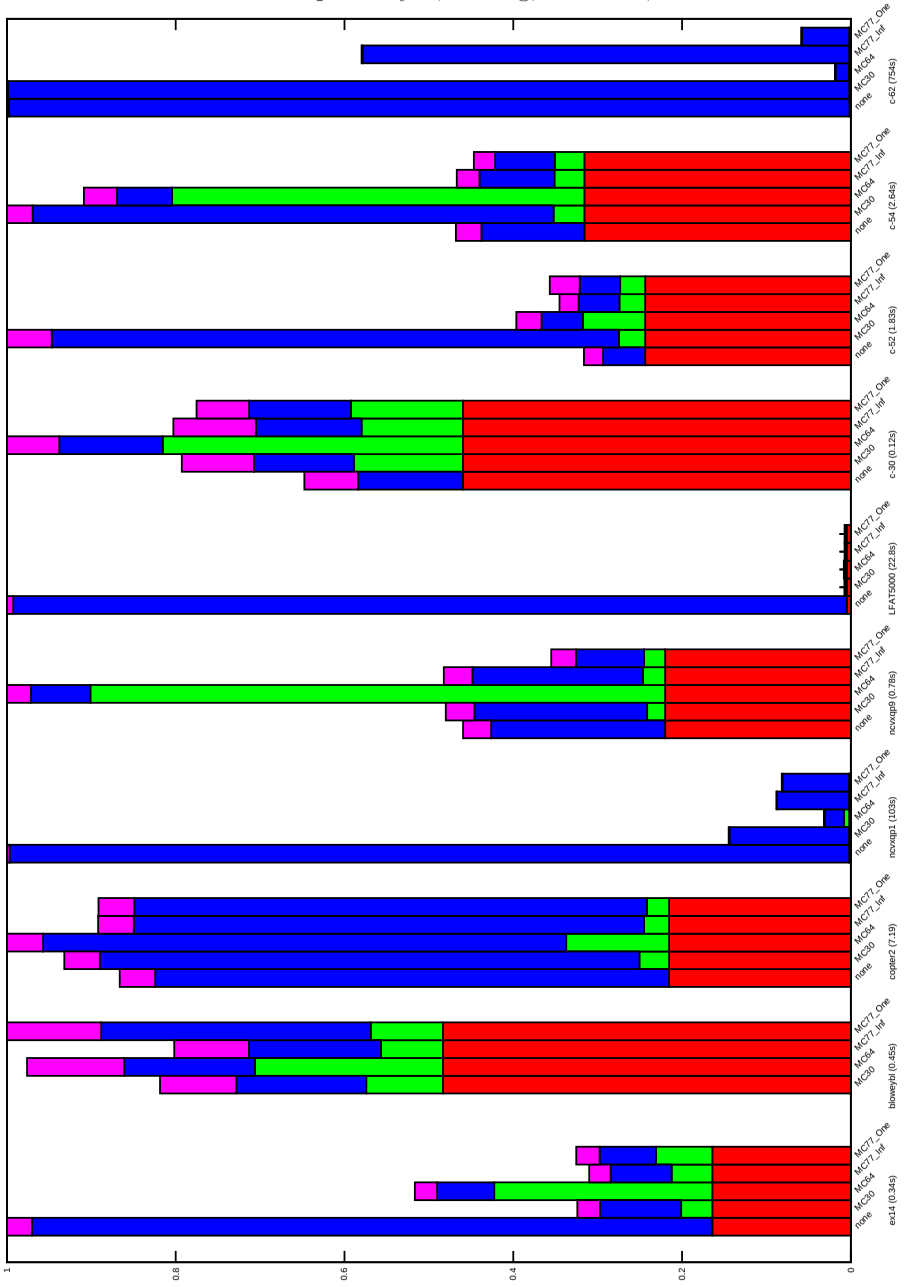
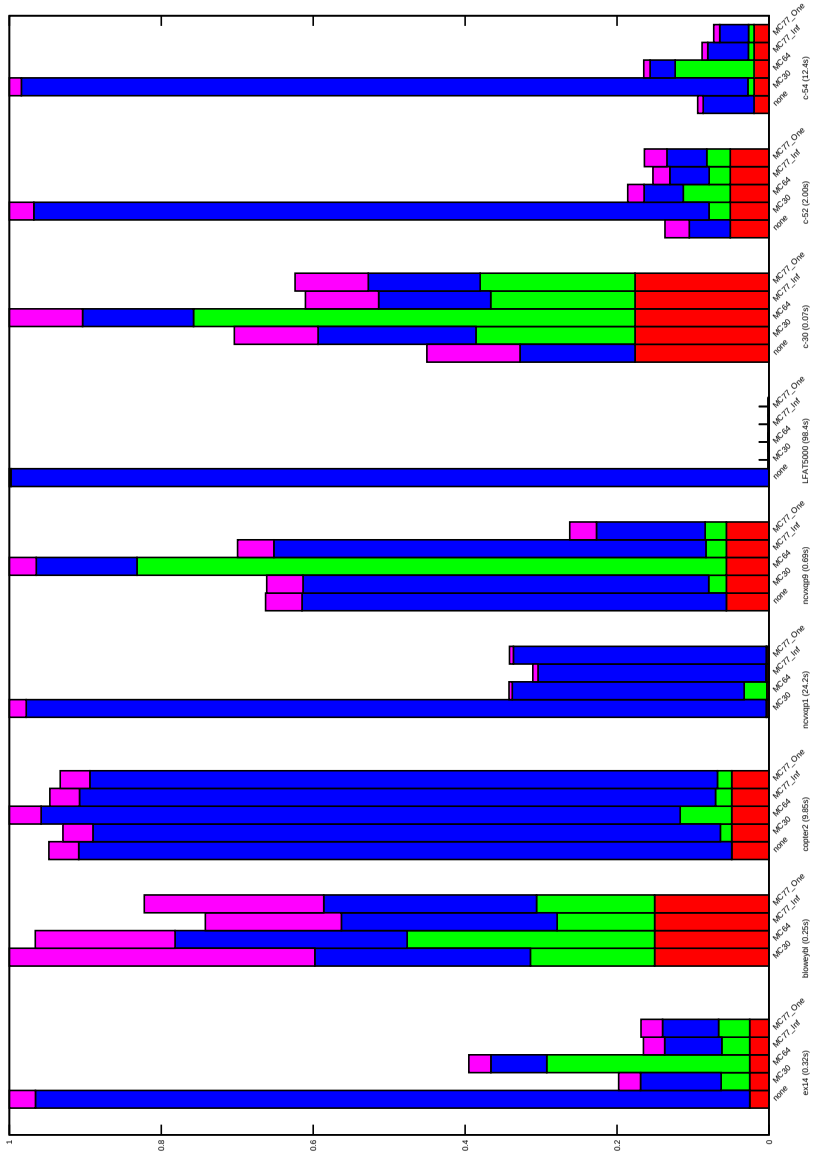


Figure 10: Showing relative timing dedicated to each part of the solve process, using AMD ordering.

Times are normalized by the longest time for each problem (given in parentheses).
From the bottom to the top: analyse, scaling, factorize, iterative refinement.
Note: c-62 with the AMD ordering failed to factorize due to lack of memory



MC64 results in slightly larger MA57 residuals and problems ncvxqp9, c-30 and c-54 illustrate that MC64 can be expensive, with the total solution time for these relatively small problems dominated by the scaling time. For larger problems, we anticipate that the time to scale will be a small fraction of the total time regardless of the scaling used.

$\texttt{MC77}_\infty$ is competitive when we compare the total solution times, and it leads to small MA57 residuals. Furthermore, it appears to be the most robust for singular problems (although in our tests we have not applied the MC64 modification suggested in [12] that is intended to cope better with singularity).

6 Conclusions

We conclude that, while the performance of MC64 and $\texttt{MC77}_1$ is equally good for most problems, MC64 is more consistent, with $\texttt{MC77}_1$ behaving poorly on a small minority of problems. This must, however, be set aside the amount of time required to compute the scaling — MC64 was by far the most expensive to compute and, for the size of problems tested, sometimes led to large total solution times. MC64 is also observed to occasionally produce MA57 residuals that are two or three orders of magnitude larger than $\texttt{MC77}_1$, but we found these could be reduced by using $\texttt{MC77}_\infty$ as a prescaling. Either MC64 (possibly prescaled by $\texttt{MC77}_\infty$) or $\texttt{MC77}_1$ would make a good default approach for a direct solver but we would also recommend that, if the user has a series of problems to solve, it may be worthwhile performing initial experiments before deciding which scaling to use. We note that for larger problems the scaling time may not be a significant proportion of the total solution time, so MC64 may prove better on this domain. MC30 is probably best avoided for indefinite problems.

A small number of steps of iterative refinement will, in most cases, remove any significant differences in the size of the residuals. For very large problems, it will be necessary to hold the matrix factors out of core. MA57 does not offer a facility for this but there is a new out-of-core multifrontal solver in HSL called HSL_MA77 [17]. For an out-of-core solver, the solve phase is expensive (since the cost of reading in the factor data once for the forward substitution and once for the back substitution adds a significant overhead). In this case, it is advantageous not to require the use of refinement (or, at least, to minimize the number of iterative steps). To this end, $\texttt{MC77}_\infty$ may be the best choice (although it may suffer a very large number of delayed pivots for some problems).

Finally, we remark that, for our test matrices, it was often not necessary to do any scaling at all. However, without scaling the worst case behaviour was more extreme than for any of the scalings we tried and, of course, we do not know if the test data was prescaled before being included in the University of Florida Sparse Matrix Collection.

7 Code availability

All the codes discussed in this report are included in the 2007 release of the mathematical software library HSL. All use of HSL requires a licence; details of how to obtain a licence and the packages are available at http://www.cse.scitech.ac.uk/nag/hsl/.

8 Acknowledgments

We are grateful to our colleagues John Reid, Iain Duff and Mario Arioli for useful discussions relating to this work.

References

[1] P. Amestoy, T. Davis, and I. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Analysis and Applications, 17 (1996), pp. 886–905.

[2] ——, *Algorithm 837: Amd, an approximate minimum degree ordering algorithm*, ACM Trans. Mathematical Software, 30 (2004), pp. 381–388.

[3] M. Arioli, I. S. Duff, S. Gratton, and S. Pralet, *A note on GMRES preconditioned by a perturbed LDL^T decomposition with static pivoting*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2024–2044.

[4] F. Bauer, *Optimally scaled matrices*, Numerische Mathematik, 5 (1963), pp. 73–87.

[5] M. Benzi, J. C. Haws, and M. Tuma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM Journal on Scientific Computing, 22 (2001), pp. 1333–1353.

[6] A. Curtis and J. Reid, *On the automatic scaling of matrices for gaussian elimination*, IMA Journal of Applied Mathematics, 10 (1972), pp. 118–124.

[7] T. Davis, *The University of Florida Sparse Matrix Collection.* http://www.cise.ufl.edu/research/sparse/matrices.

[8] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 720–755.

[9] E. D. Dolan and J. J. Moreé, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[10] I. Duff, *MA57– a new code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Mathematical Software, 30 (2004), pp. 118–154.

[11] I. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Analysis and Applications, 22 (2001), pp. 973–996.

[12] I. S. Duff and S. Pralet, *Strategies for scaling and pivoting sparse symmetric indefinite problems*, Tech. Rep. RAL-TR-2004-020, Rutherford Appleton Laboratory, 2004.

[13] K. Goto and R. van de Geijn, *High performance implementation of the level-3 BLAS*, ACM Transactions on Mathematical Software. to appear.

[14] HSL, *A collection of Fortran codes for large-scale scientific computation*, 2007. See http://www.cse.clrc.ac.uk/nag/hsl/.

[15] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing, 20 (1999), pp. 359–392.

[16] S. Pralet, *Constrained orderings and scheduling for parallel sparse linear algebra*, PhD thesis, CERFACS, 2004.

[17] J. K. Reid and J. A. Scott, *An out-of-core sparse Cholesky solver*, Tech. Rep. RAL-TR-2006-013, Rutherford Appleton Laboratory, 2006.

[18] D. Ruiz, *A scaling algorithm to equilibrate both rows and columns norms in matrices*, Tech. Rep. RAL-TR-2001-034, Rutherford Appleton Laboratory, 2001.

A Test Set

We list below the set of problems we tested out scalings on. The number of negative pivots was found through the use of `MA57` and the estimated condition number from `MA60`, in both cases using `MC64` as a scaling. Singular matrices are structurally singular as detected by `MC64` and hence no condition number was calculated.

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
ACUSIM/Pres_Poisson	14822	365313	0	2.0678E+05	
Alemdar/Alemdar	6245	24413	3583	1.5934E+05	
Andrews/Andrews	60000	410077	0	1.0000E+00	
Bai/bfwb398	398	1654	398	1.3583E+01	
Bai/bfwb62	62	202	62	1.0694E+01	
Bai/bfwb782	782	3382	782	1.3590E+01	
Bai/mhd3200b	3200	10758	0	1.5504E+05	
Bai/mhd4800b	4800	16160	0	2.3291E+05	
Bai/mhdb416	416	1364	0	1.9557E+04	
Bai/odepb400	400	399	0	-	Yes
Bates/Chem97ZtZ	2541	4951	0	1.9341E+01	
Bindel/ted_B	10605	77592	0	4.2811E+01	
Bindel/ted_B_unscaled	10605	77592	0	4.2811E+01	
Boeing/bcsstk34	588	11003	0	1.0029E+03	
Boeing/bcsstk35	30237	740200	3	3.7671E+12	
Boeing/bcsstk36	23052	583096	0	4.6990E+09	
Boeing/bcsstk37	25503	583240	3	2.6967E+13	
Boeing/bcsstk38	8032	181746	0	2.1529E+08	
Boeing/bcsstk39	46772	1068033	1	2.3058E+07	
Boeing/bcsstm34	588	12429	237	2.1995E+05	
Boeing/bcsstm35	30237	18211	20	-	Yes
Boeing/bcsstm36	23052	166389	0	-	Yes
Boeing/bcsstm37	25503	14765	40	-	Yes
Boeing/bcsstm38	8032	7842	47	-	Yes
Boeing/bcsstm39	46772	46772	0	2.0000E+00	
Boeing/crystk01	4875	160383	3	1.0171E+16	
Boeing/crystm01	4875	55107	0	6.0047E+01	
Boeing/crystm02	13965	168435	0	5.8600E+01	
Boeing/ct20stif	52329	1375396	0	1.6264E+09	
Boeing/msc00726	726	17622	0	2.2845E+03	
Boeing/msc01050	1050	15103	0	6.6758E+13	
Boeing/msc01440	1440	23855	0	2.6843E+05	
Boeing/msc04515	4515	51111	0	3.8341E+06	
Boeing/msc10848	10848	620313	0	1.6516E+07	
Boeing/msc23052	23052	588933	0	4.6849E+09	
Boeing/nasa1824	1824	20516	20	1.4485E+05	
Cannizzo/sts4098	4098	38227	0	8.8386E+04	
Cote/vibrobox	12328	177578	3	5.5486E+19	
Cunningham/m3plates	11107	6639	0	-	Yes
Cunningham/qa8fk	66127	863353	1	1.3002E+16	

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
Cunningham/qa8fm	66127	863353	0	5.4000E+01	
Cylshell/s1rmq4m1	5489	143300	0	6.1290E+05	
Cylshell/s1rmt3m1	5489	112505	0	9.7393E+05	
Cylshell/s2rmq4m1	5489	143300	0	3.8003E+07	
Cylshell/s2rmt3m1	5489	112505	0	6.7942E+07	
Cylshell/s3rmq4m1	5489	143300	0	3.8531E+09	
Cylshell/s3rmt3m1	5489	112505	0	6.7893E+09	
Cylshell/s3rmt3m3	5357	106526	0	6.7057E+09	
FIDAP/ex10hs	2548	29928	0	1.3659E+10	
FIDAP/ex10	2410	28625	0	1.6079E+10	
FIDAP/ex12	3973	42092	1133	2.6678E+17	
FIDAP/ex13	2568	39098	0	1.8463E+14	
FIDAP/ex14	3251	35013	900	2.4485E+10	
FIDAP/ex15	6867	52769	0	3.8524E+11	
FIDAP/ex2	441	13640	28	2.4914E+09	
FIDAP/ex32	1159	6251	295	9.3953E+17	
FIDAP/ex33	1733	11961	0	1.6386E+11	
FIDAP/ex3	1821	27253	0	1.4210E+10	
FIDAP/ex4	1601	16950	450	2.6840E+03	
FIDAP/ex5	27	153	0	6.7076E+07	
FIDAP/ex9	3363	51417	0	1.7961E+11	
Gaertner/nopoly	10774	40808	1	1.0000E+00	
GHS_indef/a0nsdsil	80016	200021	29318	1.5974E+06	
GHS_indef/a5esindl	60008	145004	21030	1.0516E+06	
GHS_indef/aug3dcqp	35543	77829	8000	4.3362E+08	
GHS_indef/blockqp1	60012	340022	40001	1.2799E+02	
GHS_indef/bloweya	30004	80005	10002	1.2391E+09	
GHS_indef/bloweybl	30003	70001	10001	-	Yes
GHS_indef/bloweybq	10001	39996	0	7.2846E+15	
GHS_indef/brainpc2	27607	96601	13800	7.1489E+04	
GHS_indef/bratu3d	27792	88627	12167	1.0818E+03	
GHS_indef/c-55	32780	218115	13659	3.4324E+04	
GHS_indef/c-58	37595	295076	15134	2.0567E+04	
GHS_indef/c-59	41282	260909	17469	5.0170E+03	
GHS_indef/c-62ghs	41731	300537	16573	1.1151E+04	
GHS_indef/c-63	44234	239469	18729	2.8262E+03	
GHS_indef/c-68	64810	315408	28264	2.7563E+05	
GHS_indef/c-69	67458	345714	29026	2.6966E+04	
GHS_indef/c-70	68924	363955	29622	1.8414E+04	
GHS_indef/c-72	84064	395811	36114	7.1888E+03	
GHS_indef/cont-201	80595	239596	40198	1.8451E+08	
GHS_indef/copter2	55476	407714	27726	6.8051E+05	
GHS_indef/dawson5	51537	531157	25794	8.3821E+05	
GHS_indef/dixmaanl	60000	179999	5340	2.1976E+05	
GHS_indef/dtoc	24993	34986	9997	-	Yes
GHS_indef/exdata_1	6001	1137751	3001	2.6611E+06	
GHS_indef/helm3d01	32226	230335	6771	1.6766E+06	
GHS_indef/k1_san	67759	303364	20804	-	Yes
GHS_indef/laser	3002	5000	1001	2.9973E+15	
GHS_indef/linverse	11999	53988	2838	7.2366E+03	

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
GHS_indef/mario001	38434	114643	15304	4.2336E+04	
GHS_indef/ncvxbqp1	50000	199984	37398	8.2330E+08	
GHS_indef/ncvxqp1	12111	40537	5000	1.0271E+17	
GHS_indef/ncvxqp9	16554	31547	7500	6.9614E+08	
GHS_indef/olesnik0	88263	402623	27233	6.1771E+17	
GHS_indef/qpband	20000	30000	5000	1.2571E+01	
GHS_indef/sit100	10262	34094	3120	4.1956E+17	
GHS_indef/spmsrtls	29995	129971	15979	7.8561E+04	
GHS_indef/stokes128	49666	295938	16386	3.9189E+16	
GHS_indef/stokes64s	12546	74242	4098	8.1436E+14	
GHS_indef/stokes64	12546	74242	4098	2.9892E+19	
GHS_indef/tuma1	22967	50560	9607	1.0547E+03	
GHS_indef/tuma2	12992	28440	5477	8.6113E+02	
GHS_psdef/apache1	80800	311492	0	1.0668E+06	
GHS_psdef/crankseg_1	52804	5333507	0	1.9459E+05	
GHS_psdef/crankseg_2	63838	7106348	0	2.1824E+05	
GHS_psdef/cvxbqp1	50000	199984	0	6.3666E+06	
GHS_psdef/gridgena	48962	280523	0	3.7154E+05	
GHS_psdef/jnlbrng1	40000	119600	0	1.7775E+02	
GHS_psdef/minsurfo	40806	122214	0	8.2000E+01	
GHS_psdef/obstclae	40000	118804	0	4.2000E+01	
GHS_psdef/oilpan	73752	1835470	0	3.0539E+08	
GHS_psdef/s3dkq4m2	90449	2455670	0	6.9891E+10	
GHS_psdef/s3dkt3m2	90449	1921955	0	1.2010E+11	
GHS_psdef/torsion1	40000	118804	0	4.2000E+01	
GHS_psdef/vanbody	47072	1191985	0	2.4857E+13	
GHS_psdef/wathen100	30401	251001	0	5.1716E+01	
GHS_psdef/wathen120	36441	301101	0	7.2056E+01	
Grund/meg4	5860	26324	54	7.8880E+05	
Gset/G10	800	19176	400	4.3357E+04	
Gset/G11	800	1600	400	2.3278E+04	
Gset/G12	800	1600	400	3.3889E+03	
Gset/G13	800	1600	399	3.3819E+04	
Gset/G18	800	4694	401	2.6108E+04	
Gset/G19	800	4661	401	7.2313E+03	
Gset/G20	800	4672	402	1.0880E+06	
Gset/G21	800	4667	401	2.5327E+04	
Gset/G27	2000	19990	1000	4.6682E+04	
Gset/G28	2000	19990	1000	5.3688E+04	
Gset/G29	2000	19990	1001	5.7464E+04	
Gset/G30	2000	19990	1001	2.8521E+04	
Gset/G31	2000	19990	1000	2.0240E+04	
Gset/G32	2000	4000	1000	1.6704E+04	
Gset/G33	2000	4000	1001	3.1158E+04	
Gset/G34	2000	4000	1000	2.5711E+04	
Gset/G39	2000	11778	1001	5.4037E+04	
Gset/G40	2000	11766	997	1.7115E+04	
Gset/G41	2000	11785	999	8.4174E+04	
Gset/G42	2000	11779	997	3.5004E+04	
Gset/G56	5000	12498	2483	-	Yes

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
Gset/G57	5000	10000	2500	4.6896E+04	
Gset/G59	5000	29570	2503	4.5648E+04	
Gset/G61	7000	17148	3480	-	Yes
Gset/G62	7000	14000	3500	3.6970E+04	
Gset/G64	7000	41459	3496	1.7626E+05	
Gset/G65	8000	16000	4000	2.6700E+05	
Gset/G66	9000	18000	4500	1.3318E+05	
Gset/G67	10000	20000	5000	1.4857E+06	
Gset/G6	800	19176	399	3.3607E+04	
Gset/G7	800	19176	399	1.5595E+05	
Gset/G8	800	19176	399	1.4547E+05	
Gset/G9	800	19176	400	4.5451E+04	
HB/1138_bus	1138	2596	0	5.1165E+05	
HB/494_bus	494	1080	0	8.9041E+04	
HB/662_bus	662	1568	0	4.7383E+04	
HB/685_bus	685	1967	0	9.4624E+03	
HB/bcsstk01	48	224	0	1.1364E+04	
HB/bcsstk02	66	2211	0	4.3841E+03	
HB/bcsstk03	112	376	0	3.8495E+05	
HB/bcsstk04	132	1890	0	1.8571E+04	
HB/bcsstk05	153	1288	0	8.1638E+03	
HB/bcsstk06	420	4140	0	1.8221E+05	
HB/bcsstk07	420	4140	0	1.8221E+05	
HB/bcsstk08	1074	7017	0	1.0831E+05	
HB/bcsstk09	1083	9760	0	2.4960E+04	
HB/bcsstk10	1086	11578	0	3.3123E+04	
HB/bcsstk11	1473	17857	0	2.0555E+07	
HB/bcsstk12	1473	17857	0	2.0555E+07	
HB/bcsstk13	2003	42943	0	1.8037E+06	
HB/bcsstk14	1806	32630	0	2.8497E+04	
HB/bcsstk15	3948	60882	0	3.7169E+05	
HB/bcsstk16	4884	147631	0	2.5142E+03	
HB/bcsstk17	10974	219812	0	4.3774E+06	
HB/bcsstk18	11948	80519	0	1.3946E+06	
HB/bcsstk19	817	3835	0	3.6154E+09	
HB/bcsstk20	485	1810	0	4.4715E+09	
HB/bcsstk21	3600	15100	0	6.8144E+04	
HB/bcsstk22	138	417	0	1.5573E+04	
HB/bcsstk23	3134	24156	0	5.2545E+09	
HB/bcsstk24	3562	81736	0	1.1094E+09	
HB/bcsstk25	15439	133840	0	5.9001E+07	
HB/bcsstk26	1922	16129	0	8.1232E+05	
HB/bcsstk27	1224	28675	0	1.3541E+04	
HB/bcsstk28	4410	111717	0	7.6861E+07	
HB/bcsstm02	66	66	0	2.0000E+00	
HB/bcsstm04	132	132	0	-	Yes
HB/bcsstm05	153	153	0	2.0000E+00	
HB/bcsstm06	420	420	0	2.0000E+00	
HB/bcsstm07	420	3836	0	4.7976E+02	
HB/bcsstm08	1074	1074	0	2.0000E+00	

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
HB/bcsstm09	1083	1083	0	2.0000E+00	
HB/bcsstm10	1086	11589	54	1.3585E+05	
HB/bcsstm11	1473	1473	0	2.0000E+00	
HB/bcsstm12	1473	10566	0	1.5065E+04	
HB/bcsstm13	2003	11973	0	-	Yes
HB/bcsstm19	817	817	0	2.0000E+00	
HB/bcsstm20	485	485	0	2.0000E+00	
HB/bcsstm21	3600	3600	0	2.0000E+00	
HB/bcsstm22	138	138	0	2.0000E+00	
HB/bcsstm23	3134	3134	0	2.0000E+00	
HB/bcsstm24	3562	3562	0	2.0000E+00	
HB/bcsstm25	15439	15439	0	2.0000E+00	
HB/bcsstm26	1922	1922	0	2.0000E+00	
HB/bcsstm27	1224	28675	31	4.8041E+09	
HB/lund_a	147	1298	0	2.7603E+05	
HB/lund_b	147	1294	0	2.8441E+03	
HB/nos1	237	627	0	5.3599E+06	
HB/nos2	957	2547	0	1.3253E+09	
HB/nos3	960	8402	0	5.9591E+04	
HB/nos4	100	347	0	1.8046E+03	
HB/nos5	468	2820	0	8.0483E+03	
HB/nos6	675	1965	0	3.4852E+06	
HB/nos7	729	2673	0	1.2934E+08	
HB/plat1919	1919	17159	1	7.9845E+18	
HB/plat362	362	3074	0	4.2775E+11	
HB/saylr3	1000	2375	1000	7.7267E+03	
HB/saylr4	3564	12940	3564	5.8993E+06	
HB/sherman1	1000	2375	1000	7.7267E+03	
HB/zenios	2873	15032	169	-	Yes
HB/bcsstm01	48	48	0	-	Yes
HB/gr_30_30	900	4322	0	3.7723E+02	
Lourakis/bundle1	10581	390741	0	2.4218E+02	
MathWorks/Kuu	7102	173651	0	2.2410E+04	
MathWorks/Muu	7102	88618	0	4.2811E+01	
Mulvey/finan512	74752	335872	0	1.5714E+01	
Nasa/nasa1824	1824	20516	0	1.4588E+05	
Nasa/nasa2146	2146	37198	0	1.5156E+03	
Nasa/nasa2910	2910	88603	0	2.6965E+05	
Nasa/nasa4704	4704	54730	0	8.1746E+06	
ND/nd3k	9000	1644345	0	3.3513E+07	
ND/nd6k	18000	3457658	0	3.7355E+07	
Nemeth/nemeth01	9506	367280	6004	2.8731E+02	
Nemeth/nemeth02	9506	202157	9506	2.1843E+00	
Nemeth/nemeth03	9506	202157	9506	2.2625E+00	
Nemeth/nemeth04	9506	202157	9506	2.4168E+00	
Nemeth/nemeth05	9506	202157	9506	2.5939E+00	
Nemeth/nemeth06	9506	202157	9506	2.8593E+00	
Nemeth/nemeth07	9506	202159	9506	3.2327E+00	
Nemeth/nemeth08	9506	202161	9506	3.7485E+00	
Nemeth/nemeth09	9506	202506	9506	4.6004E+00	

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
Nemeth/nemeth10	9506	205477	9506	5.8754E+00	
Nemeth/nemeth11	9506	208885	9506	8.0118E+00	
Nemeth/nemeth12	9506	228162	9506	1.2416E+01	
Nemeth/nemeth13	9506	241989	9506	2.4083E+01	
Nemeth/nemeth14	9506	252825	8505	4.1514E+02	
Nemeth/nemeth15	9506	274654	6004	2.6359E+02	
Nemeth/nemeth16	9506	298259	5504	3.5618E+01	
Nemeth/nemeth17	9506	319563	5504	2.4461E+01	
Nemeth/nemeth18	9506	352370	5504	2.0767E+01	
Nemeth/nemeth19	9506	413904	5504	1.7810E+01	
Nemeth/nemeth20	9506	490688	5504	1.5768E+01	
Nemeth/nemeth21	9506	591626	5504	1.2439E+01	
Nemeth/nemeth22	9506	684169	5504	1.2522E+01	
Nemeth/nemeth23	9506	758158	5504	1.2673E+01	
Nemeth/nemeth24	9506	758028	5504	1.2697E+01	
Nemeth/nemeth25	9506	760632	5504	1.2699E+01	
Nemeth/nemeth26	9506	760633	5504	1.2699E+01	
Norris/fv1	9604	47434	0	1.3761E+01	
Norris/fv2	9801	48413	0	1.3764E+01	
Norris/fv3	9801	48413	0	4.4205E+03	
Oberwolfach/filter2D	1668	6209	1668	2.6357E+04	
Oberwolfach/flowmeter0	9669	38530	9669	1.2928E+04	
Oberwolfach/gas_sensor	66917	885141	66917	3.4571E+06	
Oberwolfach/gyro_k	17361	519260	0	2.2871E+08	
Oberwolfach/gyro_m	17361	178896	0	7.3184E+05	
Oberwolfach/gyro	17361	519260	0	2.2871E+08	
Oberwolfach/LF10	18	50	0	1.4387E+05	
Oberwolfach/LFAT5000	19994	49980	0	1.2189E+16	
Oberwolfach/LFAT5	14	30	0	6.3750E+03	
Oberwolfach/rail_1357	1357	5171	1357	1.3611E+04	
Oberwolfach/rail_20209	20209	79721	20209	2.1805E+05	
Oberwolfach/rail_5177	5177	20181	5177	5.4493E+04	
Oberwolfach/rail_79841	79841	316881	79841	8.7231E+05	
Oberwolfach/spiral	1434	9831	1434	8.5213E+04	
Oberwolfach/t2dah_a	11445	93781	11445	1.1868E+15	
Oberwolfach/t2dah_e	11445	93781	0	1.9647E+02	
Oberwolfach/t2dah	11445	93781	11445	1.1868E+15	
Oberwolfach/t2dal_a	4257	20861	4257	3.8046E+14	
Oberwolfach/t2dal_bci	4257	20861	4257	3.2133E+17	
Oberwolfach/t2dal_e	4257	4257	0	2.0000E+00	
Oberwolfach/t2dal	4257	20861	4257	3.8046E+14	
Oberwolfach/t3dh_a	79171	2215638	79171	3.3620E+14	
Oberwolfach/t3dh_e	79171	2215638	4	3.9912E+19	
Oberwolfach/t3dh	79171	2215638	79171	3.3620E+14	
Oberwolfach/t3dl_a	20360	265113	20360	3.7941E+13	
Oberwolfach/t3dl_e	20360	20360	0	2.0000E+00	
Oberwolfach/t3dl	20360	265113	20360	3.7941E+13	
Okunbor/aft01	8205	66886	0	2.2281E+05	
Pajek/GD97_b	47	132	23	-	Yes
Pajek/USAir97	332	2126	161	-	Yes

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
Pajek/GD99_b	64	127	24	-	Yes
Pajek/geom	7343	11898	3278	-	Yes
Pajek/Journals	124	6096	0	3.4925E+01	
Pajek/Reuters911	13332	148038	6651	-	Yes
Pajek/Sandi_authors	86	124	41	-	Yes
Pajek/Stranke94	10	45	8	8.5660E+01	
PARSEC/benzene	8219	125444	2	4.2025E+03	
PARSEC/Na5	5832	155731	4	3.3141E+03	
PARSEC/Si10H16	17077	446500	41	1.3175E+04	
PARSEC/Si2	769	9285	1	7.1481E+02	
PARSEC/SiH4	5041	88472	4	2.9581E+03	
PARSEC/SiNa	5743	102265	5	2.2848E+03	
Pothen/bodyy4	17546	69742	0	9.4636E+02	
Pothen/bodyy5	18589	73935	0	5.7504E+03	
Pothen/bodyy6	19366	77057	0	3.7715E+04	
Pothen/mesh1e1	48	177	0	1.0634E+01	
Pothen/mesh1em1	48	177	0	2.3109E+01	
Pothen/mesh1em6	48	177	0	1.2390E+01	
Pothen/mesh2e1	306	1162	0	2.5451E+02	
Pothen/mesh2em5	306	1162	0	2.7325E+02	
Pothen/mesh3e1	289	1089	0	1.7991E+01	
Pothen/mesh3em5	289	1089	0	9.9978E+00	
Rothberg/cfd1	70656	949510	0	7.4141E+05	
Schenk_IBMNA/c-18	2169	8657	901	4.0978E+03	
Schenk_IBMNA/c-19	2327	12072	865	3.5318E+03	
Schenk_IBMNA/c-20	2921	12803	1300	2.9179E+05	
Schenk_IBMNA/c-21	3509	17833	1490	7.4566E+03	
Schenk_IBMNA/c-22	3792	16332	1662	1.1931E+04	
Schenk_IBMNA/c-23	3969	17524	1649	7.2092E+02	
Schenk_IBMNA/c-24	4119	19909	1792	3.6510E+04	
Schenk_IBMNA/c-25	3797	26716	1407	1.9780E+03	
Schenk_IBMNA/c-26	4307	19422	1787	1.8403E+04	
Schenk_IBMNA/c-27	4563	17969	1942	8.2028E+03	
Schenk_IBMNA/c-28	4598	17594	1900	1.1770E+04	
Schenk_IBMNA/c-29	5033	24382	2211	4.7931E+04	
Schenk_IBMNA/c-30	5321	35507	2498	1.8460E+07	
Schenk_IBMNA/c-31	5339	41955	2220	3.0800E+04	
Schenk_IBMNA/c-32	5975	30223	2724	1.8066E+04	
Schenk_IBMNA/c-33	6317	31220	2791	8.7714E+03	
Schenk_IBMNA/c-34	6611	35472	2583	2.7082E+03	
Schenk_IBMNA/c-35	6537	34714	2423	1.2384E+04	
Schenk_IBMNA/c-36	7479	36710	3355	2.1399E+04	
Schenk_IBMNA/c-37	8204	41440	3454	9.0401E+03	
Schenk_IBMNA/c-38	8127	42908	3307	4.5595E+02	
Schenk_IBMNA/c-39	9271	73041	4034	2.2643E+04	
Schenk_IBMNA/c-40	9941	45721	4464	9.2783E+03	
Schenk_IBMNA/c-41	9769	55757	4320	2.5864E+09	
Schenk_IBMNA/c-42	10471	60378	4541	2.0615E+04	
Schenk_IBMNA/c-43	11125	67400	5194	1.0467E+05	
Schenk_IBMNA/c-44	10728	47864	4474	1.7307E+04	

Name	m	nnz(A)	Neg Pivots	Est Cond	Singular
Schenk_IBMNA/c-45	13206	93829	6314	1.0803E+06	
Schenk_IBMNA/c-46	14913	72655	6830	3.7738E+03	
Schenk_IBMNA/c-47	15343	113372	6641	2.8708E+04	
Schenk_IBMNA/c-48	18354	92217	8137	2.5624E+04	
Schenk_IBMNA/c-49	21132	89087	8956	2.2409E+04	
Schenk_IBMNA/c-50	22401	108013	9799	2.7038E+05	
Schenk_IBMNA/c-51	23196	113123	9875	3.7426E+03	
Schenk_IBMNA/c-52	23948	113332	10288	1.4863E+09	
Schenk_IBMNA/c-53	30235	201224	13943	3.1408E+05	
Schenk_IBMNA/c-54	31793	211743	14129	2.5497E+09	
Schenk_IBMNA/c-56	35910	208405	15987	5.0943E+08	
Schenk_IBMNA/c-57	37833	221515	17836	2.4236E+04	
Schenk_IBMNA/c-60	43640	171109	19221	7.3556E+03	
Schenk_IBMNA/c-61	43618	176817	18575	1.0667E+04	
Schenk_IBMNA/c-62	41731	300537	16573	8.1087E+07	
Schenk_IBMNA/c-64b	51035	384438	23881	3.2770E+04	
Schenk_IBMNA/c-64	51035	384438	23881	7.7320E+08	
Schenk_IBMNA/c-65	48066	204297	20408	8.8270E+03	
Schenk_IBMNA/c-66b	49989	274498	22293	4.9722E+04	
Schenk_IBMNA/c-66	49989	274498	22293	2.9569E+04	
Schenk_IBMNA/c-67b	57975	294955	26718	1.9744E+04	
Schenk_IBMNA/c-67	57975	294955	26718	1.5961E+09	
Schmid/thermal1	82654	328556	0	3.9160E+05	
Simon/olafu	16146	515651	0	1.8654E+08	
Simon/raefsky4	19779	674195	0	1.8508E+11	
UTEP/Dubcova1	16129	134569	0	2.6115E+03	
UTEP/Dubcova2	65025	547625	0	1.0397E+04	