# Iterative Solution of Large Sparse Linear Systems Arising from Application of Interior Point Method in Computational Geomechanics

Omid Kardani, Andrei V. Lyamin, and Kristian Krabbenhøft

***Abstract*— The efficiency of several preconditioned Conjugate Gradient (PCG) schemes for solving of large sparse linear systems arising from application of second order cone programming in computational plasticity problems is studied. Direct solvers fail to solve these linear systems in large sizes, such as three dimensional cases, due to their high storage and computational cost. This motivates using iterative methods. However, iterative solvers are not efficient without preconditioning techniques for difficult problems. In this paper, the effect of different incomplete factorization preconditioning techniques on the convergence behavior of the preconditioned Conjugate Gradient (PCG) method to solve these large sparse and usually ill-conditioned linear systems is investigated. Furthermore, numerical results of applying PCG to several sample systems are presented and discussed. Several suggestions are also made as potential research subjects in this field.**

***Index Terms*— incomplete factorization preconditioning, limit analysis, preconditioned conjugate gradient method, Interior Point Method**

## I. Introduction

The application of second order cone programming (SOCP) to solving optimization problems arising in Geomechanics has recently been of growing interest and significant advances have been made in this field. Some of the most important applications in Geomechanics include traditionally difficult problems in plasticity [12], [16], [17], limit and elastoplastic analysis [14] and most recently granular contact dynamics [13]. In this paper, we focus on the case of limit analysis. Upon formulating the original problem as SOCP, it can be solved by primal-dual interior point method (IPM) [2]. However, in each step of this method, a symmetric positive definite (SPD) linear system of equation needs to be solved. Due to their robustness and accuracy, the direct solvers have been traditionally used for this task [2], [26].

O. Kardani is with the Center for Geotechnical and Materials Modelling, University of Newcastle, NSW, Australia (phone: +612-4921-6292; e-mail: Omid.Kardani@ uon.edu.au).

A. V. Lyamin is with the Center for Geotechnical and Materials Modelling, University of Newcastle, NSW, Australia (e-mail: Andrei.Lyamin@newcastle.edu.au).

K. Krabbenhoft is with the Center for Geotechnical and Materials Modelling, University of Newcastle, NSW, Australia (e-mail: Kristian.Krabbenhoft@newcastle.edu.au).

However, for large three dimensional problems direct solvers require prohibitively high storage and computational efforts. Therefore, the use of iterative solvers becomes imperative. But iterative schemes are often far from being accurate for highly ill-conditioned systems arising in IPM iterations. This motivates using appropriate preconditioners to enhance the efficiency of the iterative solution schemes.

In our study, we use preconditioned Conjugate Gradient method (PCG) with several incomplete factorization preconditioning techniques and make a comparison of their effects on the robustness of PCG method.

The structure of the paper is as follows: in section 2, the SOCP as well as its application to limit analysis is introduced. In section 3, PCG method with various incomplete factorization preconditioners are briefly discussed. Then numerical results of applying the PCG method are presented and discussed in section 4. Finally, conclusions and future work are given in section5.

## II. Limit Analysis as SOCP Problem

Conic programming in the field of plasticity is concerned with the following standard form of problems:

$$\begin{aligned} &\text{minimize} && \alpha \\ &\text{subject to} && \mathbf{B}^T\boldsymbol{\sigma} = \alpha\mathbf{p} + \mathbf{p}_0\,, \\ &&& \boldsymbol{\sigma} \in \kappa \end{aligned} \tag{1}$$

in which constant and variable loads are given by $\mathbf{p}_0$ and $\mathbf{p}$, respectively. $\alpha$ denotes the load multiplier and $\mathbf{B}^T$ is the discrete equilibrium operator. Also, $\boldsymbol{\sigma}$ is the vector of the stresses and $\kappa$ denotes an admissible stress space.

Krabbenhoft et al. [14] proposed a practical form of SOCP for limit analysis by casting the Mohr-Coulomb criterion under plane strain conditions as quadratic cone. The resulting optimization problem then reads:

$$\begin{aligned} &\text{minimize} && \alpha \\ &\text{subject to} && \mathbf{B}^T\boldsymbol{\sigma} = \alpha\mathbf{p} + \mathbf{p}_0 \\ &&& \boldsymbol{\rho} = \mathbf{D}\boldsymbol{\sigma} + \mathbf{d} \quad ' \\ &&& \boldsymbol{\rho} \in \kappa_q \end{aligned} \tag{3}$$

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

where $\kappa_q$ is the following quadratic cone

$$\kappa_q = \left\{ \boldsymbol{\rho} \in \mathbb{R}^{m+1} \mid \rho_1 \geq \sqrt{{\rho_2}^2 + {\rho_3}^2} \right\}, \tag{4}$$

and

$$\mathbf{D} = \begin{pmatrix} \sin\phi & \sin\phi & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 2c\cos\phi \\ 0 \\ 0 \end{pmatrix},$$

with $\phi$ and $c$ denoting the friction angle and cohesion, respectively.

The problems of the form (3) can be solved using primal-dual interior point method [2]. In each step of this method, after some computationally cheap calculations, a system of linear equations of the general form

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{5}$$

in which $\mathbf{A}$ is a large sparse and symmetric positive definite (SPD) matrix, needs to be solved in order to find the search direction. These systems have been traditionally solved by performing a Cholesky factorization.

## III. Incomplete Factorization Preconditioning Techniques

As mentioned earlier, system (5) is problematic to solve by direct solvers for large three dimensional problems with millions of equation and unknowns involved. This necessitates exploiting efficient iterative schemes. Since the system is SPD, one of the most efficient iterative solvers can be PCG method preconditioned with incomplete Cholesky (IC) factorization techniques [3], [8], and [22]. IC factorization is done by the same procedure as the complete form. The only difference is that some of the fill-ins in the course of the factorization process are discarded. This leads to sparse factors which approximate exact Cholesky factors. Discarding new fill-ins is controlled by employing a dropping rule. In this way, a number of incomplete Cholesky factorization preconditioners can be constructed such as drop tolerance-based IC, IC with fixed fill-in and double threshold IC.

### *A. Drop tolerance-based incomplete Cholesky factorization*

One way to control the amount of fill-in allowed in the factorization process is to accept or discard new entries with regards to their absolute values. For this purpose, a drop tolerance $\tau > 0$, which is a positive real number, is used and fill-ins in step $k^{th}$ can be controlled in the following manner:

$$\begin{cases} a_{ij}^{(k)} \text{ is kept} & \left|a_{ij}^{(k)}\right| > \tau\sqrt{d_i^{(k)} d_j^{(k)}} \\ a_{ij}^{(k)} \text{ is dropped} & \textit{otherwise} \end{cases}, \tag{6}$$

in which $d_i^{(k)}$ and $d_j^{(k)}$ are the $i^{th}$ and $j^{th}$ diagonal elements of the matrix in step $k^{th}$, respectively [20]. This class of incomplete factorization methods are studied widely and shown to be very reliable preconditioners provided the suitable drop tolerance is chosen [3], [6], [19], [22].

### *B. Incomplete Cholesky factorization with fixed fill-in*

Incomplete factorization with fixed fill-in was first introduced by Jones and Plassmann [10]. In their proposed algorithm, the fill-in is controlled by keeping a limited number of elements which have the largest absolute values in each row of the Cholesky factor. They set this fixed number of fill-ins for each row to be the number of nonzero elements in the same row of the triangular part of the original matrix. A similar strategy was used by Lin and More [15]. However, in their method, they let a fixed number of additional elements to be accepted in each row of the Cholesky factor. Again, the acceptance of fill-ins is based on their absolute value. By denoting this fixed number by $\rho$, this preconditioner is known as FFIC($\rho$). Note that in the special case $\rho = 0$, the Jones and Plassmann's preconditioner [10] is obtained.

### *C. Double threshold incomplete Cholesky factorization*

The idea of using two different levels of dropping in the process of incomplete factorization is first proposed by Saad [21]. He designed a so called ILUT($\tau, \rho$) preconditioner with two thresholds $\tau$, which is a drop tolerance and $\rho$, which is in fact the maximum number of nonzero elements allowed in each row of the incomplete factors. This preconditioner was shown to be quite powerful for difficult problems [3], [21]. The same strategy can be employed for incomplete Cholesky factorization of SPD matrices to produce so-called ICT($\tau, \rho$) preconditioner.

### *D. Robust Incomplete Cholesky Factorization*

IC has been proved to exist for M-matrices [19] and also H-matrices with positive diagonals [18]. However, it can fail for general SPD matrices due to pivot breakdowns; that is, occurring a zero or negative pivot during the factorization process. There are several remedies for this problem.

One way is to apply a global shift to the diagonal of the matrix before starting the factorization. In this method which was proposed by Manteuffel [19], the original matrix $\mathbf{A}$ is replaced by

$$\mathbf{A} + \alpha\mathbf{D}, \tag{7}$$

where $\mathbf{D}$ is the diagonal of $\mathbf{A}$ and $\alpha$ is known as diagonal shifting parameter. Applying this diagonal shifting strategy with an appropriate shift parameter $\alpha$ to the diagonally scaled form the coefficient matrix which is $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ can be quite efficient and leads to very powerful preconditioners [7], [15], [23], and [24]. However, the process of choosing $\alpha$ is based on trial and error.

Another strategy to achieve a stable factorization without any pivot breakdowns for general SPD matrices is to design a

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

modified incomplete factorization without modifying the original matrix. The most famous and widely used strategy in this category is the robust incomplete factorization presented by Ajiz and Jennings [1]. Their method, which is abbreviated as AJRIC($\tau$), is in fact a modified form of drop tolerance-based IC factorization. It proceeds by adding the absolute value of each dropped element (or a factor of it [9]) to both corresponding diagonal elements of the matrix. This strategy leads to a breakdown-free IC factorization. Similar strategies can be found in [25].

In his renowned paper, Kaporin [11] presented a robust factorization for general SPD matrices which leads to a very efficient preconditioner. It is shown that the number of iterations of the CG solver can be significantly reduced. In this method, the SPD coefficient matrix is factorized as $\mathbf{U}^T\mathbf{U}+\mathbf{U}^T\mathbf{R}+\mathbf{R}^T\mathbf{U}$, where $\mathbf{U}$ is an upper triangular matrix approximating the exact Cholesky factor, while $\mathbf{R}$ is a strict upper triangular error matrix and contains very small elements. This strategy requires more computational effort compared to previously mentioned schemes. However the accuracy achieved in computed incomplete Cholesky factor has a dramatic effect on reducing the computational cost of preconditioned conjugate gradient method [4], [11].

In the next section, we present the numerical results of applying some of the discussed incomplete Cholesky factorization preconditioners to PCG method in an attempt to solve the three sample systems of the form (5) arising from solving problem (3) by primal-dual interior point method.

## IV. Numerical results

In this section numerical results of applying PCG method preconditioned with different variants of IC preconditioners are presented and discussed. Our three samples systems which will be referred to as T_small, T_mid and T_large have different sizes as followed from their names.

A summary of the features of the sample coefficient matrices, including the dimension of the matrix (DIM), the number of nonzero elements in the triangular part of the matrix (NNZ), the minimum (Min Eig) and maximum (Max Eig) magnitudes of eigenvalues and the condition number (CN) of the matrix (calculated by dividing the maximum eigenvalue by the minimum one) are presented in Table I.

As seen from the Table I, all three sample matrices are very ill-conditioned. This suggests a weak performance of the iterative solvers without any preconditioning.

In all PCG schemes designed, before applying the incomplete factorization, reverse Cuthill-McKee (RCM) reordering has been implemented to the coefficient matrix. This is because RCM has shown the best effect among all ordering schemes with regards to both computational cost and robustness of the incomplete Cholesky factorization preconditioners [5]. Also, the system has been scaled diagonally. It means the following system has been solved:

$$\mathbf{D}^{-1/2}\mathbf{P}\mathbf{A}\mathbf{P}^T\mathbf{D}^{-1/2}\mathbf{y}=\mathbf{D}^{-1/2}\mathbf{P}\mathbf{b},\qquad \mathbf{x}=\mathbf{P}^T\mathbf{D}^{-1/2}\mathbf{y}, \tag{8}$$

in which $\mathbf{P}$ is the RCM permutation matrix and $\mathbf{D}$ is the diagonal of $\mathbf{A}$. In addition, in cases where the incomplete factorization process was not robust and failed due to pivot breakdown, a global diagonal shifting strategy was applied to the matrix and recorded. Furthermore, the following stopping criterion is used for all PCG method runs:

$$\|\mathbf{r}_k\|_\infty \le 1.0e-7\times(\|\mathbf{b}\|_\infty+\|\mathbf{u}_k\|_\infty\|\mathbf{A}\|_\infty), \tag{9}$$

in which $\|\mathbf{b}\|_\infty$, $\|\mathbf{r}_k\|_\infty$ and $\|\mathbf{u}_k\|_\infty$ are the infinity norm of the right hand side vector, the current residual and solution vectors, respectively. Also, $\|\mathbf{A}\|_\infty$ denotes the infinity norm of the coefficient matrix, which is in fact the maximum of the row sums of the matrix. The maximum number of iterations is also set to be 5000 for all cases. In all following tables, the common notations used are as follows:

PCN: condition number of the preconditioned matrix;
P-Time: CPU time (in seconds) spent on building the preconditioner;
CG-Time: CPU time (in seconds) spent on CG process until a stopping criterion is met;
CG-Iter: the number of iterations performed by CG algorithm until convergence.

Finally, the computations are all carried out on a desktop computer with 2.8 GHz dual-core Processor and 4 GB of RAM.

### A. Incomplete Cholesky factorization with fixed fill-in

Tables II to IV show the convergence behavior of the PCG method preconditioned with FFIC($\rho$) for different values of $\rho$. In all cases of applying to our sample systems, the algorithm failed due to pivot breakdown. A global diagonal shifting strategy, therefore, has been employed. In other words, in a process of trial an error, a suitable shifting parameter as in (7) was used.

TABLE I
Properties of sample matrices

| Matrix | DIM | NNZ | Min Eig | Max Eig | CN |
|---|---|---|---|---|---|
| T_small | 5973 | 62214 | 1.3e-4 | 4.3e+5 | 3.3e+9 |
| T_mid | 165703 | 1757345 | 2.8e-6 | 9.7e+7 | 3.5e+13 |
| T_large | 329119 | 3584324 | 1.8e-6 | 4.5e+6 | 2.5e+12 |

TABLE II
PCG preconditioned with FFIC applied to T_small

| Fixed Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 0 | 5.3e+4 | 3.906e-2 | 2.016 | 2367 |
| 5 | 5.4e+4 | 4.812e-2 | 2.343 | 2323 |
| 10 | 5.1e+4 | 5.214e-2 | 2.619 | 2330 |
| 20 | 5.2e+4 | 5.985e-2 | 2.995 | 2314 |
| 40 | 4.9e+4 | 7.182e-2 | 3.556 | 2351 |

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE III
PCG PRECONDITIONED WITH FFIC APPLIED TO T_MID

| Fixed Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 0 | 7.3e+4 | 4.082 | 82.406 | 2834 |
| 5 | 7.2e+4 | 4.004 | 87.443 | 2805 |
| 10 | 7.3e+4 | 5.111 | 100.121 | 2904 |
| 20 | 7.1e+4 | 6.365 | 118.482 | 2909 |
| 40 | 7.2e+4 | 8.591 | 152.290 | 2826 |

TABLE IV
PCG PRECONDITIONED WITH FFIC APPLIED TO T_LARGE

| Fixed Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 0 | 6.7e+4 | 8.648 | 172.538 | 2869 |
| 5 | 6.6e+4 | 13.367 | 207.789 | 3178 |
| 10 | 6.4e+4 | 14.273 | 204.274 | 2852 |
| 20 | 6.4e+4 | 17.976 | 257.976 | 3027 |
| 40 | 6.4e+4 | 22.738 | 307.403 | 2743 |

Tables II to IV show that in some cases additional fill-ins lead to fewer number of CG iterations. This can be interpreted as the result of improvement in the condition number of the coefficient matrix. On the other hand, by allowing more fill-ins, the preconditioner becomes less sparse and consequently the time of even fewer number of CG iterations grows. With regards to total time taken for both preconditioning and solving process, it seems beneficial to allow some fill-ins.

*B. Ajiz-Jennings' Robust Incomplete Cholesky factorization*

This is one of the most popular versions of incomplete Cholesky factorization which is widely used in different engineering applications [3]. As mentioned in the previous section, it is a breakdown-free version of the incomplete Cholesky factorization with drop tolerance and for general SPD matrices. The convergence analysis of the PCG method preconditioned with AJRIC( $\tau$ ) for different values of the drop tolerance $\tau$ applied to our three sample matrices are stated in Tables V to VII.

TABLE V
PCG PRECONDITIONED WITH AJRIC APPLIED TO T_SMALL

| Drop Tolerance | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-2 | 6.2e+4 | 4.296e-2 | 2.095 | 1500 |
| 1.0e-3 | 6.5e+4 | 8.203e-2 | 2.305 | 1412 |
| 1.0e-4 | 6.5e+4 | 0.117 | 2.005 | 1390 |
| 1.0e-5 | 6.4e+4 | 0.152 | 2.676 | 1348 |

TABLE VI
PCG PRECONDITIONED WITH AJRIC APPLIED TO T_MID

| Drop Tolerance | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-2 | 7.5e+4 | 2.339 | 60.975 | 1383 |
| 1.0e-3 | 7.6e+4 | 4.332 | 51.542 | 1109 |
| 1.0e-4 | 7.6e+4 | 5.968 | 79.004 | 1133 |
| 1.0e-5 | 7.6e+4 | 7.628 | 126.327 | 1301 |

TABLE VII
PCG PRECONDITIONED WITH AJRIC APPLIED TO T_LARGE

| Drop Tolerance | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-2 | 7.0e+4 | 5.968 | 133.704 | 1497 |
| 1.0e-3 | 7.0e+4 | 11.699 | 136.302 | 1394 |
| 1.0e-4 | 6.9e+4 | 18.023 | 187.150 | 1381 |
| 1.0e-5 | 6.9e+4 | 18.496 | 290.094 | 1402 |

The last three tables show that by choosing a smaller value for the drop tolerance, the expense of constructing the preconditioner increases since more fill-ins allowed in the incomplete factor. However, in most cases the number of iterations of CG method decreases for smaller drop tolerances, yet the preconditioner is less sparse and as a result the CG solver is more time consuming. Looking for a balance between these two features, one can suggest the values in the interval $[1.0e-2, 1.0e-3]$ to be more appropriate in our application.

*C. Incomplete Cholesky factorization with double threshold*

In Tables VIII to X, the convergence behavior of the PCG method preconditioned with ICT( $\tau$ , $\rho$ ) is given for different values of $\tau$ and $\rho$ applied to our three different matrices. Again, the factorization breakdowns were encountered, so, like the case of FFIC( $\rho$ ), a global diagonal shifting strategy has been utilized.

It appears that the computation time taken by CG to solve the preconditioned system is far more dependent on the number of fixed fill-ins rather than on the drop tolerance since the fixed fill-in parameter determines the density of the preconditioner. The same comment can be given on the storage requirements of the preconditioner.

TABLE VIII
PCG PRECONDITIONED BY ICT APPLIED TO T_SMALL

| Drop Tolerance | Max Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|---|
| 1.0e-3 | 5 | 7.0e+4 | 1.562e-2 | 1.869 | 2418 |
| | 10 | 6.4e+4 | 3.125e-2 | 1.743 | 2365 |
| | 20 | 6.4e+4 | 3.125e-2 | 2.169 | 2317 |
| 1.0e-4 | 5 | 7.0e+4 | 1.562e-2 | 1.630 | 2415 |
| | 10 | 6.4e+4 | 2.734e-2 | 1.743 | 2355 |
| | 20 | 6.3e+4 | 3.125e-2 | 2.169 | 2316 |
| 1.0e-5 | 5 | 7.0e+4 | 1.562e-2 | 1.688 | 2420 |
| | 10 | 6.4e+4 | 3.125e-2 | 1.738 | 2360 |
| | 20 | 6.3e+4 | 4.687e-2 | 2.132 | 2313 |
| 1.0e-6 | 5 | 7.0e+4 | 3.125e-2 | 1.611 | 2408 |
| | 10 | 6.4e+4 | 3.125e-2 | 1.776 | 2373 |
| | 20 | 6.3e+4 | 4.687e-2 | 2.132 | 2309 |

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE X
PCG PRECONDITIONED BY ICT APPLIED TO T_LARGE

| Drop Tolerance | Max Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|---|
| 1.0e-3 | 5 | 8.2e+4 | 7.281 | 144.710 | 2882 |
| | 10 | 8.0e+4 | 13.562 | 163.734 | 2764 |
| | 20 | 7.5e+4 | 21.218 | 263.500 | 3414 |
| 1.0e-4 | 5 | 8.2e+4 | 7.281 | 144.816 | 2891 |
| | 10 | 8.0e+4 | 13.515 | 161.460 | 2748 |
| | 20 | 7.5e+4 | 21.796 | 225.824 | 2940 |
| 1.0e-5 | 5 | 8.2e+4 | 7.281 | 143.488 | 2871 |
| | 10 | 8.0e+4 | 14.218 | 162.238 | 2756 |
| | 20 | 7.5e+4 | 22.250 | 223.714 | 2931 |
| 1.0e-6 | 5 | 8.2e+4 | 8.140 | 144.716 | 2891 |
| | 10 | 8.0e+4 | 14.703 | 170.507 | 2848 |
| | 20 | 7.5e+4 | 23.156 | 228.273 | 2963 |

TABLE IX
PCG PRECONDITIONED BY ICT APPLIED TO T_MID

| Drop Tolerance | Max Fill-in Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|---|
| 1.0e-3 | 5 | 8.8e+4 | 3.027 | 68.752 | 2922 |
| | 10 | 8.4e+4 | 4.074 | 78.208 | 2940 |
| | 20 | 8.5e+4 | 6.464 | 99.131 | 2928 |
| 1.0e-4 | 5 | 8.8e+4 | 2.996 | 70.502 | 2932 |
| | 10 | 8.4e+4 | 4.011 | 77.551 | 2914 |
| | 20 | 8.5e+4 | 7.297 | 98.890 | 2919 |
| 1.0e-5 | 5 | 8.8e+4 | 3.011 | 66.376 | 2902 |
| | 10 | 8.4e+4 | 5.089 | 77.382 | 2910 |
| | 20 | 8.5e+4 | 7.308 | 99.574 | 2922 |
| 1.0e-6 | 5 | 8.8e+4 | 3.027 | 67.842 | 2910 |
| | 10 | 8.4e+4 | 5.042 | 77.367 | 2910 |
| | 20 | 8.5e+4 | 6.308 | 97.487 | 2915 |

However, in most cases, the drop tolerance has an obvious effect on improvement of the number of iterations of the CG method. According to these observations, the drop tolerance can be interpreted as a parameter responsible for the accuracy of the solution and the fixed fill-in number as a parameter to control the storage requirement and computational expense of the solver.

### *D.Kaporin's robust incomplete factorization*

The last incomplete factorization technique which was selected for numerical tests is the robust incomplete Cholesky second order stabilized factorization, denoted by RIC2S($\theta$), with $\theta > 0$ being a truncation parameter, introduced by Kaporin in his celebrated paper [11]. Tables XI to XIII present the convergence analysis of PCG method preconditioned by RIC2S($\theta$) for three different truncation thresholds applied on our three sample systems.

Tables XI to XIII reveal that the number of iterations of the CG method dramatically falls by decreasing of the truncation parameter $\theta$. It is also noticeable that even though the density of the preconditioner increases with a smaller $\theta$ and as a result the computational cost of each CG iteration grows, the reduction in CG iterations are so significant that the total time taken by CG solver drops in all cases. On the other hand, by exploiting a smaller truncation parameter, the cost of constructing the preconditioner goes up.

TABLE XI
PCG PRECONDITIONED WITH RIC2S APPLIED TO T_SMALL

| Truncation Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-1 | 8.7e+4 | 0.574 | 1.183 | 1317 |
| 5.0e-2 | 7.3e+4 | 0.840 | 1.100 | 1111 |
| 1.0e-2 | 7.1e+4 | 1.354 | 1.028 | 912 |

TABLE XII
PCG PRECONDITIONED WITH RIC2S APPLIED TO T_SMALL

| Truncation Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-1 | 1.0e+5 | 13.511 | 144.664 | 4586 |
| 5.0e-2 | 8.7e+4 | 18.179 | 44.089 | 1349 |
| 1.0e-2 | 8.1e+4 | 24.539 | 36.925 | 984 |

TABLE XIII
PCG PRECONDITIONED WITH RIC2S APPLIED TO T_SMALL

| Truncation Parameter | PCN | P-Time | CG-Time | CG-Iter |
|---|---|---|---|---|
| 1.0e-1 | 8.4e+4 | 21.769 | 106.394 | 1622 |
| 5.0e-2 | 7.2e+4 | 33.304 | 93.710 | 1402 |
| 1.0e-2 | 6.6e+4 | 45.976 | 88.773 | 1132 |

## V.CONCLUSION

The effect of several most famous incomplete Cholesky factorization preconditioners on the PCG method employed to solve large sparse linear systems of equations arising in implementation of second order conic programming in limit analysis were investigated and compared in this paper. In most cases, allowing more fill-ins leads to a more accurate preconditioner and hence fewer number of CG iterations. However, as the preconditioner becomes denser, the time spent on each CG iteration grows. Therefore, in most cases, more fill-ins are not effective for CG in terms of computational time. Overall, although the Kaporin's RIC2S is the most expensive preconditioner to built, it has the best effect on reducing the number of CG iterations and CG time and in our application it turns out to be the most efficient preconditioner for PCG solver providing the appropriate truncation parameter is chosen.

Future work in this field would be devoted to the implementation of the optimized embedding of PCG in the primal-dual interior point solver for limit analysis. Ultimately, parallelization of the whole solution scheme in order to obtain better performance for large sparse linear systems using multi processor computers and graphics processing units (GPU) will be considered in the future.

## REFERENCES

[1] M. A. Ajiz and A. Jennings, "A robust incomplete Choleski-conjugate gradient algorithm," in *International Journal of Numerical Methods in Engineering*, vol. 20, 1984, pp. 949–966.

[2] E. D. Anderson, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," in *Mathematical Programming*, vol. 95, 2003, pp. 249–277.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[3] M. Benzi, "Preconditioning techniques for large linear systems: A survey," in *Journal of Computational Physics*, vol. 182, 2002, pp. 418–477.
[4] M. Benzi, J. K. Cullum, and M. Tuma, "Robust approximate inverse preconditioning for the conjugate gradient method," in *SIAM Journal on Scientific Computing*, vol. 22, 2000, pp. 1318–1332.
[5] M. Benzi, D. B. Szyld, and A. van Duin, "Orderings for incomplete factorization preconditioning of nonsymmetric problems," in *SIAM Journal on Scientific Computing*, vol. 20, 1999, pp. 1652–1670.
[6] K. Chan, *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005.
[7] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic Press, 1981.
[8] G. Gambolati, M. Ferronato, and C. Janna, "Preconditioners in computational Geomechanics: A survey," in *International Journal of Numerical and Analytical Methods in Engineering*, vol. 35, 2011, pp. 980–996.
[9] I. Hladik, M. B. Reed, and G. Swoboda, "Robust preconditioners for linear elasticity FEM analyses," in *International Journal of Numerical Methods in Engineering*, vol. 40, 1997, pp. 2109–2127.
[10] M. T. Jones and P. E. Plassmann, "An improved incomplete Cholesky factorization," in *ACM Transactions on Mathematical Software*, vol. 21, 1995, pp. 5–17.
[11] I. E. Kaporin, "High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition," in *Numerical Linear Algebra with Applications*, vol. 5, 1998, pp. 483–509.
[12] K. Krabbenhoft and A. V. Lyamin, "Computational Cam clay plasticity using second-order cone programming," in *Computer Methods in Applied Mechanics and Engineering*, submitted for publication.
[13] K. Krabbenhoft and A. V. Lyamin, and J. Huang, "Granular contact dynamics using mathematical programming methods," in *Computer Methods in Applied Mechanics and Engineering*, submitted for publication.
[14] K. Krabbenhoft and A. V. Lyamin, and S. W. Sloan, "Formulation and solution of some plasticity problems as conic programs," in *International Journal of Solids and Structures*, vol. 44, 2007, pp. 1533–1549.
[15] C.-J. Lin and J. J. More, "Incomplete Cholesky factorizations with limited memory," in *SIAM Journal on Scientific Computing*, vol. 21, 1999, pp. 24–45.
[16] A. Makrodimopoulos and C. M. Martin, "A novel formulation of upper bound limit analysis as a second-order programming problem," in *Proc. COMPLAS Conf.*, E. Onate and D. Owen, Eds. Barcelona, 2005.
[17] A. Makrodimopoulos and C. M. Martin, "Limit analysis using large-scale SOCP optimization," in *Proc. 13th Nat. Conf. UK Association for Computational Mechanics in Engineering*, Sheffield, 2005, pp. 21-24.
[18] T. A. Manteuffel, "An incomplete factorization technique for positive definite linear systems," in *Mathematics of Computation*, vol. 34, 1980, pp. 473–497.
[19] J. A. Meijerink and H. A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix," in *Mathematics of Computation*, vol. 31, 1977, pp. 148–162.
[20] N. Munksgaard, "Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients," in *ACM Transactions on Mathematical Software*, vol. 6, 1980, pp. 206–219.
[21] Y. Saad, "ILUT: A dual threshold incomplete LU factorization," in *Numerical Linear Algebra with Applications*, vol. 1, 1994, pp. 387–402.
[22] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd ed., 2003.
[23] T. Schlick, "Modified Cholesky factorizations for sparse preconditioners," in *SIAM Journal on Scientific Computing*, vol. 14, 1993, pp. 424–445.
[24] R. B. Schnabel and E. Eskow, "A new modified Cholesky factorization," in *SIAM Journal on Scientific Computing*, vol. 11, 1990, pp. 1136–1158.
[25] M. Tismenetsky, "A new preconditioning technique for solving large sparse linear systems," in *Linear Algebra with Applications*, vol. 154–156, 1991, pp. 331–353.
[26] S. J. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM, 1997.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)