Code Verification for SOCP in the SCS Software

Angus Ramsay, Director (Ramsay Maunder Associates)

This technical note presents some basic code verification studies undertaken to demonstrate the Splitting Cone Solver (SCS) software, [1], and the FORTRAN interface written for this software, [2]. The SCS software solves mathematical programmes with a range of cone forms. In particular, the second-order cone form is of interest to engineers assessing structural strength for materials where the yield criterion is quadratic and may be mapped into a Lorentz cone.

Study Number 1: Reinforced Concrete Slab Example

The symmetric half of an isosceles triangular reinforced concrete slab is considered. The slab is fixed along the base and a 1kN point load is applied at the free vertex – 2kN for the full slab as shown in Figure 1. Isotropic reinforcement is adopted with a moment capacity of 100kNm/m for both top (hogging) and bottom (sagging) layers of steel.

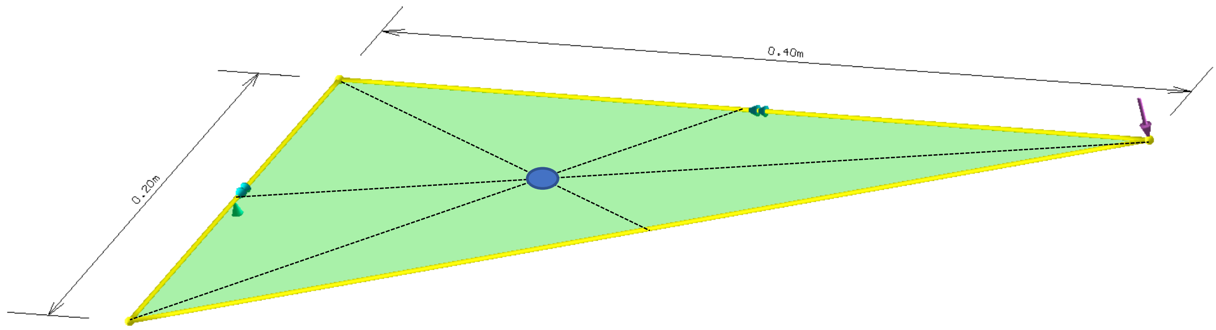

Figure 1: Reinforced concrete slab problem

The upper-bound (yield line) solution for this problem is shown as a single hogging yield line along the base with a load factor of 50, i.e., for the given reinforcement, the point load to cause the complete slab to collapse is 2 x 50 x 1kN = 100kN. The upper bound solution was obtained using the linear programme software Clp, [3].

The lower-bound solution for this problem also gives a load factor of 50. The lower bound solution was obtained using the SOC programming software MOSEK, [4].

With both upper and lower bound solutions leading to the same load factor indicates that this load factor is the exact value for this problem.

The SOCP for this problem is based on a particular solution derived from the yield line solution plus a single complementary or hyperstatic moment field which, in the case, is uniform. The total moment field (particular plus complementary) is monitored at the single centroidal sample point where it is constrained to lie within the bi-conic yield criterion of Nielsen, [5]. This yield criterion is mapped into two three-dimensional Lorentz cones, one for the hogging moments and one for the sagging moments. The objective for this SOCP is to maximise the load factor which is the variable that multiplies the particular solution. The second variable is the multiplier of the single hyperstatic moment field. The solution produced by MOSEK for the variables is [50, 0.5] and the input data and solution for this problem using the SCS solver are shown in Figure 2. Note that the input data was taken from the input to MOSEK and has been scaled. Thus, where as the moment capacity is 100kNm/m, unit values are specified in the {b} vector.

SCS						
{b}	{c}		[A]			{x}
1	-1		0.000	1.250		50
0	0		0.025	-2.500		0.5
0			-0.020	1.250		
1			0.000	1.250		
0			0.025	-2.500		
0			0.020	-1.250		

Figure 2: Input data and solution for SCS – Reinforced Concrete Problem

Study Number 2: Abstract Steel Example

This is an abstract problem using, as for the reinforced concrete slab example, a particular stress field plus a single hyperstatic stress field. The sample points are now considered where the three principal stresses are constrained by the L2 norm similar to, for example, the von Mises yield criterion, [*]. The Lorentz cones are now four-dimensional. The problem can be expressed in the following manner.

The total stress, $\{S\}$ is the linear combination of the particular stress field, $\{S_p\}$, and the hyperstatic stress field, $\{S_h\}$, where λ and φ are problem variables to be determined by maximising the load factor, λ.

$$\{S\} = \lambda\{S_p\} + \varphi\{S_h\}$$

The stress vectors comprise six terms, three principal stresses each at the two sample points. The two, four-dimensional cone constraints are:

$$\|S_{1,2,3}\|_2 \leq S_Y$$

$$\|S_{4,5,6}\|_2 \leq S_Y$$

This SCOP has been solved firstly in Excel using the GRG non-linear solver and, secondly, in SCS. Details of the input data and solutions are presented in Figure 3 where it is seen that the difference in the two solutions is small.

EXCEL										
	Sy		2							
			λ	φ						
	Variables		**1.748528**	**-0.911741**						
		{S}=λ{Sp}+φ{Sh}	λ{S_p}	φ{S_h}						
	S1	-0.4559	0.0000	-0.4559						
Point 1	S2	1.1022	0.8743	0.2279						
	S3	1.0647	1.7485	-0.6838						
	$\|\|S_{1,2,3}\|\|_2$	1.5988								
	S4	-0.4559	0.0000	-0.4559						
Point 2	S5	1.6305	1.3114	0.3191						
	S6	-1.0647	-1.7485	0.6838						
	$\|\|S_{4,5,6}\|\|_2$	2.0000								
SCS										
[A]=	0	0		{x}=	λ		SOC Constraints		{b}=	Sy
	0	0.5			φ		$\|\|S_{1,2,3}\|\|_2$<=S_y			0
	0.5	-0.25					$\|\|S_{4,5,6}\|\|_2$<=S_y			0
	1	0.75								0
	0	0		{c}=	1					Sy
	0	0.5			0					0
	0.75	-0.35								0
	-1	-0.75								0
		SCS Solution	**1.748528202**	**-0.91167**						
		Difference, %age	2.80629E-07	0.008205						

Figure 3: Excel and SCS solutions for abstract steel example

Closure

Two simple example SOCPs in the form required for the lower-bound limit analysis of engineering structures have been considered. The aim of this work was to elucidate the form of the input and to verify the solutions obtained by the SCS and the FORTRAN interface as this will form a key component in the Lamé Finite Element software currently being developed by Ramsay Maunder Associates.

In the first example SCS is compared with the MOSEK SOCP solver and the difference in the solution is minimal. In the second example SCS is compared with the GRG non-linear solver within Excel. These are different solution methods applied to the same problem and here too the difference in the solutions is small. For the first example, the solution obtained is a known theoretical solution. The exact solution is not, however, known for the second example. However, the probability that two independent codes are capable of producing incorrect but identical solutions is considered sufficiently unlikely that the work presented in this technical note will be taken as having satisfactorily verified the SCS and FORTRAN interface software being evaluated.

References

[1] Brendan O'Donoghue and Eric Chu and Neal Parikh and Stephen Boyd, {SCS}: Splitting Conic Solver, version 3.0.0,\url{https://github.com/cvxgrp/scs}, November 2021.

[2] David Bradly, SCS FORTRAN Interface, Bradly Associate Ltd, November 2021.

http://www.gino.co.uk/help/gino90/home.htm

[3] Coin-or *l*inear *p*rogramming, https://github.com/coin-or/Clp#readme

[4] MOSEK ApS, https://www.mosek.com/

[5] Maunder, E.A.W., & Ramsay, A.C.A., 'Equilibrium models for lower bound limit analyses of reinforced concrete slabs, Computers & Structures, 108-109, (2012), pp 100-109.

Appendix

The input data used for the two problems studied in this note (reverse order) as used in the SCS FORTRAN interface are shown below.

```
   sy=1d0

!    m=8; n=2
!    call SCS_Allocate_Data(Data,Sol,m=m,n=n)
!    Data%b(1:m) = (/sy,0d0,0d0,0d0,sy,0d0,0d0,0d0/)
!    Data%c(1:n) = (/-1d0,0d0/)
!    Data%A%x(1:m*n) = (/0d0,0d0,0.5d0,1d0,0d0,0d0,0.75d0,-1d0,0d0,0.5d0,-0.25d0,0.75d0,0d0,0.5d0,-0.35d0,-0.75d0/)
!    Data%A%i(1:m*n) = (/0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7/)
!    Data%A%p(1:n+1) = (/0,8,16/)
!    call SCS_Allocate_Cone(Cone,bsize=0,qsize=2,ssize=0,psize=0)
!    Cone%q = (/4,4/)

    m=6; n=2
    call SCS_Allocate_Data(Data,Sol,m=m,n=n)
    Data%b(1:m) = (/sy,0d0,0d0,sy,0d0,0d0/)
    Data%c(1:n) = (/-1d0,0d0/)
    Data%A%x(1:m*n) = (/0d0,0.025d0,-0.02d0,0d0,0.025d0,0.02d0,1.25d0,-2.5d0,1.25d0,1.25d0,-2.5d0,-1.25d0/)
    Data%A%i(1:m*n) = (/0,1,2,3,4,5,0,1,2,3,4,5/)
    Data%A%p(1:n+1) = (/0,6,12/)
    call SCS_Allocate_Cone(Cone,bsize=0,qsize=2,ssize=0,psize=0)
    Cone%q = (/3,3/)


    ![P]=[0] for RMA's problem:
   Data%P%m=0; Data%P%n=0 ; Data%p%p=0d0
```