SOLVING LARGE-SCALE SEMIDEFINITE PROGRAMS IN PARALLEL
M. V. NAYAKKANKUPPAM

AUTHOR'S RESPONSE TO REFEREE #3

We thank the referee for a thorough review and constructive comments, which we address below. (References pertain to this revised version.)

The errors in some of the formulas in Section 2.1 have been corrected by rewriting parts of that section.

The epsilon subdifferential formulas have been restated as suggested. In equation (10) in the revised version, it is indeed the same ϵ ocurring twice; this is a consequence of [15, Vol. II, Theorem X1.3.2.1].

Sections 5.1 and 5.2, along with Algorithms 5.1 and 5.2, of the previous version have been condensed into a brief overview as Section 5.1 in the present version. Textual explanations replacing Algorithms 5.1 and 5.2 now guide the reader through the Block Structured Lanczos Method. Where appropriate, we use the term *Ritz vector* instead of eigenvector. We fixed an error in the statement of Algorithm 5.1 that allowed prematurely exiting the `rst` loop if p eigenvalues had converged; this handles the referee's example of p converged eigenvalues immediately after the first spectral factorization. An eigenvalue is deemed unwanted only if the error bounds imply that it can never be one of the p largest; these are then used as implicit shifts. In particular, the maximal eigenvalue will never be one deemed unwanted, and the component of the starting vector in the direction of the maximal eigenvector will not be damped by the implicit restart. Unless the starting vector is orthogonal to the maximum eigenvalue's eigenspace (in which case any Krylov suspace method will fail), the maximum eigenpair will eventually emerge. Since this is not a block Lanczos method, the original claim about obtaining p maximal eigenpairs is true only if multiplicities are not counted; phrases have been added to reflect this. We have changed the starting vector to be random. Although this didn't seem to have much impact on the numerical results, we fully agree that the original choice of vector of all ones (normalized) could be a dangerous one.

The referee's suggestion for matrix-vector products could be faster for some types of problems (and not for others, such as low-rank, dense matrices), but it is more involved to implement block row partitioning for SDP's with multiple blocks, especially when these partitions straddle block boundaries.

Regarding accuracy, we also feel that $\delta = 0.01$ is very low, but $\delta = 0.001$ takes a really long time for some of the problems. Since the detailed numerical experiments require multiple runs, this becomes a very time consuming process overall. For most of the problems, the dimension of the bundle comes nowhere close to the dimension of the subdifferential at the optimum, so the method chokes when the accuracy requirement is increased.

We have added a paragraph in Section 9.5 to explain the anomalous entries in Table 7.

In Section 9.6, fewer eigenvalues were used to obtain fast solution times for the 0.01 accuracy level. It is indeed true that use of more eigenvalues improves overall performance for higher accuracy levels.