SUPPLEMENTARY MATERIALS: RANDOMIZED SUB-SAMPLED METHODS FOR MATRIX APPROXIMATION

ANDREW AZZAM*, BENJAMIN ONG†, AND ALLAN STRUTHERS‡

SM1. Additional Non-Accelerated Computational Results. The convergence test from subsection 4.1 was performed on the remaining matrices tested in [SM3]. As before, these figures show: BFGS($\diamond$) as specified by Eq. (2.5); DFP ($\diamond$) as specified by Eq. (2.4); NS ($\otimes$) as specified by Algorithm 2.1; SS1 ($\bullet$) as specified by Algorithm 2.2; SS2 ($\blacksquare$) as specified by Algorithm 2.3. All numerical experiments indicate that our non-accelerated sub-sampled algorithms converge predictably and consistently,

- Figure SM1 the LibSVM matrix Aloi of size $n = 128$;
- Figure SM2 the LibSVM matrix Protein of size $n = 357$;
- Figure SM3 the LibSVM matrix Real-Sim of size $n = 20958$;
- Figure SM4 the Sparse Suite matrix ND6K of size $n = 18000$;
- Figure SM5 the Sparse Suite matrix ex9 of size $n = 3363$;
- Figure SM6 the Sparse Suite matrix Chem97ZtZ of size $n = 2541$.
- Figure SM7 the Sparse Suite matrix Body of size $n = 17556$.
- Figure SM8 the Sparse Suite matrix bcsstk of size $n = 11948$.
- Figure SM9 the Sparse Suite matrix wathen of size $n = 30401$.

Plots in Figures SM3, SM4, and SM7 to SM9 indicate that a maximum running time is reached for the sub-sampled methods.

SM2. Additional Accelerated Computational Results. The convergence test from section 5 was performed on the remaining matrices tested in [SM3]. We illustrate the relative performance of the following algorithms:

- ($*$) BFGSA, Eq. (2.5) with adaptive sampling described in [SM3],
- ($\circ$) S1, Eq. (2.3) with $W = I_n$,
- ($\pentagon$) SS1A+, Algorithm 5.1
- ($\diamond$) BFGS, Eq. (2.5),
- ($\diamond$) DFP, Eq. (2.4)

on the following matrices:

- Figure SM10 the LibSVM matrix Aloi of size $n = 128$;
- Figure SM11 the LibSVM matrix Protein of size $n = 357$;
- Figure SM12 the LibSVM matrix Real-Sim of size $n = 20958$;
- Figure SM13 the Sparse Suite matrix ND6K of size $n = 18000$;
- Figure SM14 the Sparse Suite matrix ex9 of size $n = 3363$;
- Figure SM15 the Sparse Suite matrix Chem97ZtZ of size $n = 2541$.
- Figure SM16 the Sparse Suite matrix Body of size $n = 17556$.
- Figure SM17 the Sparse Suite matrix bcsstk of size $n = 11948$.
- Figure SM18 the Sparse Suite matrix wathen of size $n = 30401$.

The accelerated method Algorithm 5.1 performs well on all matrices including those with large $n \approx 10^4$ (see Figures SM12, SM13, and SM16 to SM18).

REFERENCES

*Department of Mathematical Sciences, Michigan Technological University (atazzam@mtu.edu).
†Department of Mathematical Sciences, Michigan Technological University (ongbw@mtu.edu).
‡Department of Mathematical Sciences, Michigan Technological University (struther@mtu.edu).

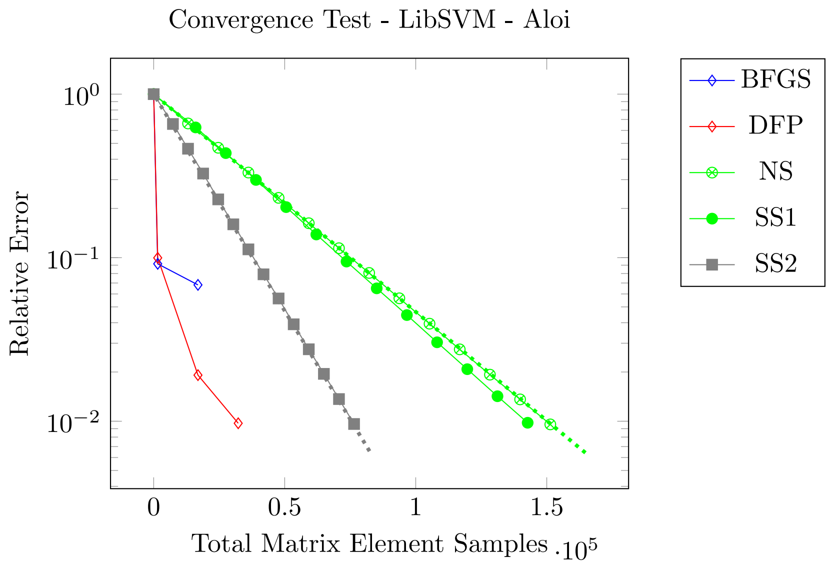

FIG. SM1. *Hessian approximation for the matrix from the LibSVM problem,* ***Aloi*** *(*$n = 128$*) [SM1] with* $s = 12 = \lceil\sqrt{128}\rceil$*. Dotted lines are theoretical convergence rates. Note, DFP and BFGS perform well.*

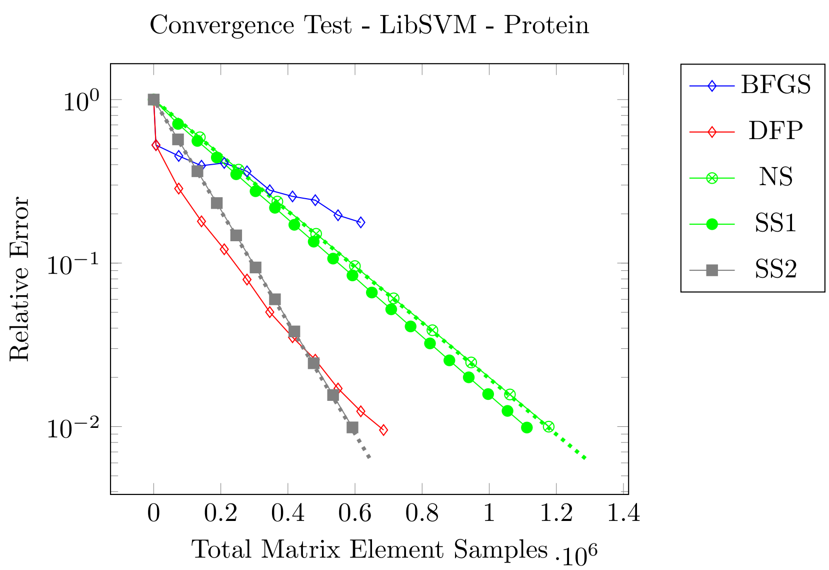

FIG. SM2. *Hessian approximation for the matrix from the LibSVM problem,* ***Protein*** *(*$n = 357$*) [SM1] with* $s = 19 = \lceil\sqrt{357}\rceil$*. Dotted lines are theoretical convergence rates. Note, DFP performs well, BFGS performs poorly.*

[SM1] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[SM2] T. A. Davis and Y. Hu, The university of florida sparse matrix collection, ACM Trans. Math. Softw., 38 (2011), pp. 1:1–1:25, https://doi.org/10.1145/2049662.2049663, http://doi.acm.org/10.1145/2049662.2049663.

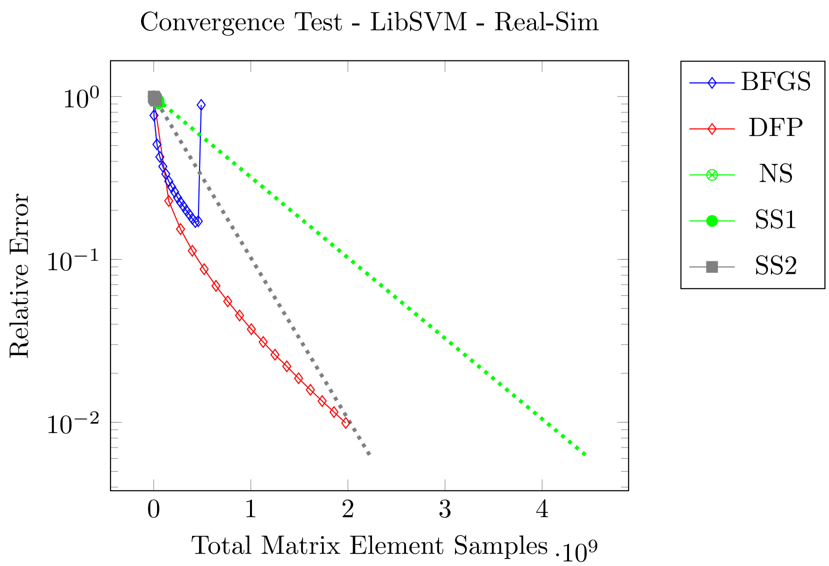

Fig. SM3. *Hessian approximation for the matrix from the LibSVM problem,* ***Real-Sim*** *($n = 20,958$) [SM1] with $s = 145 = \lceil\sqrt{20,958}\rceil$. Dotted lines are theoretical convergence rates for our algorithms. DFP performs well, BFGS does not converge.*

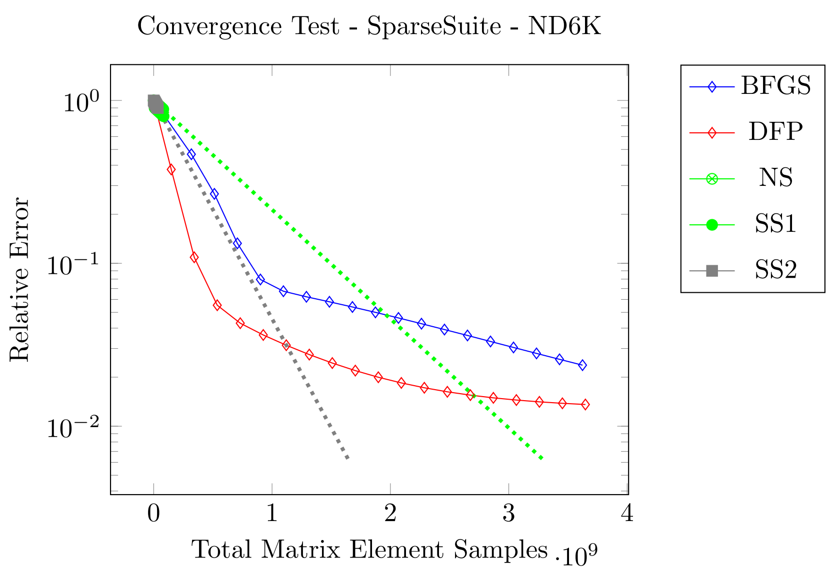

Fig. SM4. *Hessian approximation for the matrix from the Sparse Suite Library,* ***ND6K*** *($n = 18,000$) [SM2] with $s = 135 = \lceil\sqrt{18,000}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.*

[SM3] R. M. Gower and P. Richtárik, Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1380–1409, https://doi.org/10.1137/16M1062053, https://doi.org/10.1137/16M1062053.

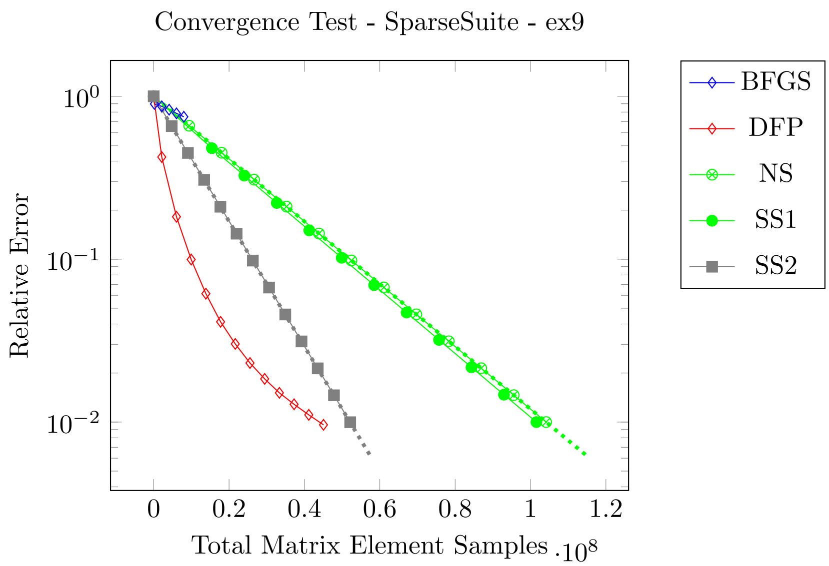

FIG. SM5. *Hessian approximation for the matrix from the Sparse Suite Library,* ***ex9*** *(*$n = 3363$*) [SM2] with* $s = 58 = \lceil\sqrt{3363}\rceil$*. Dotted lines are theoretical convergence rates for our algorithms.*

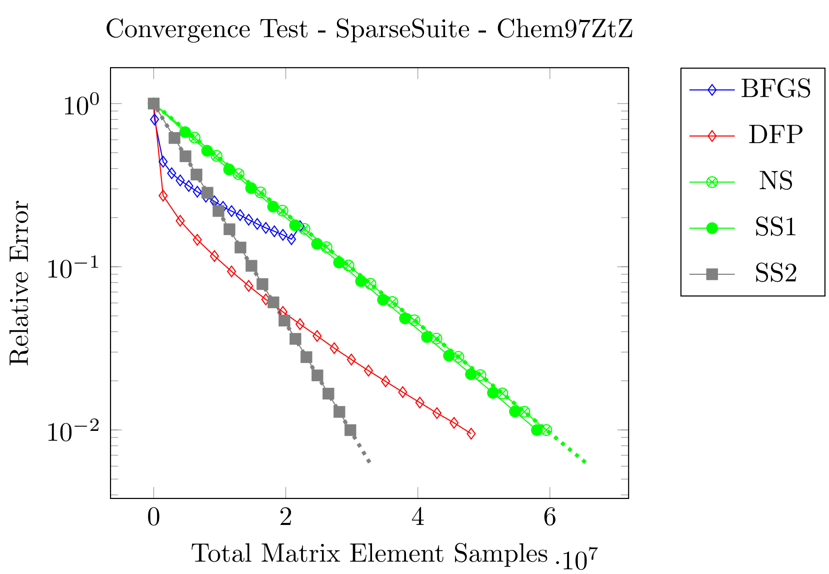

FIG. SM6. *Hessian approximation for the matrix from the Sparse Suite Library,* ***Chem97ZtZ*** *(*$n = 2541$*) [SM2] with* $s = 51 = \lceil\sqrt{2541}\rceil$*. Dotted lines are theoretical convergence rates for our algorithms.*

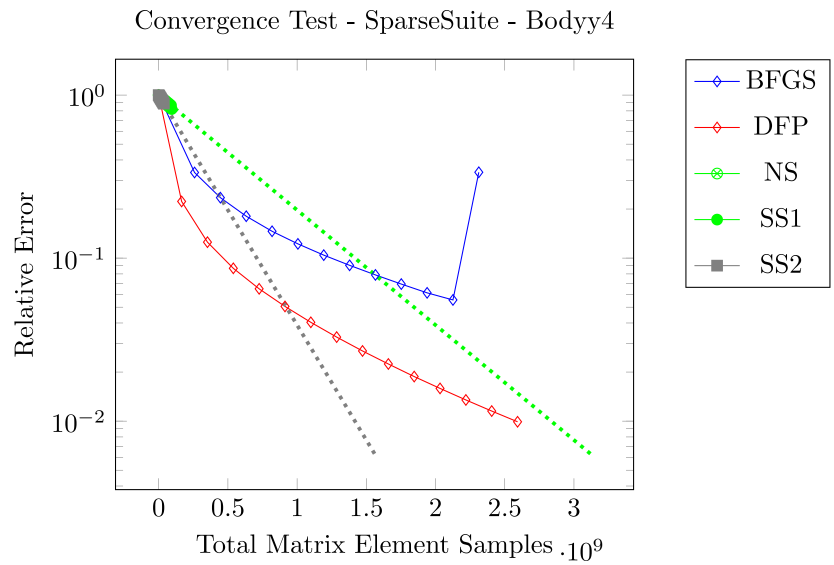

FIG. SM7. *Hessian approximation for the matrix from the Sparse Suite Library, **Body** ($n =$ 17,546) [SM2] with $s = 133 = \lceil\sqrt{17,546}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.*

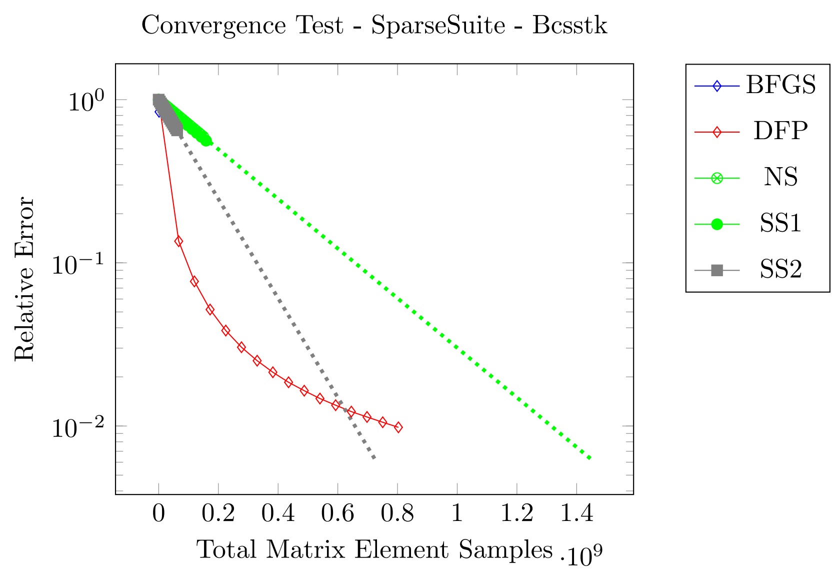

FIG. SM8. *Hessian approximation for the matrix from the Sparse Suite Library, **bcsstk** ($n =$ 11,948) [SM2] with $s = 110 = \lceil\sqrt{11,948}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.*

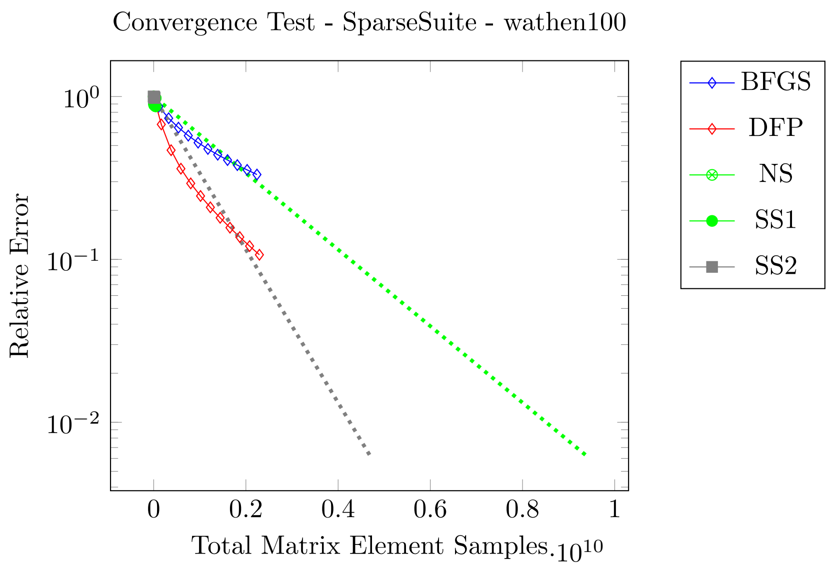

FIG. SM9. *Hessian approximation for the matrix from the Sparse Suite Library,* ***wathen*** *($n = 30,401$) [SM2] with $s = 175 = \lceil\sqrt{30,401}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.*

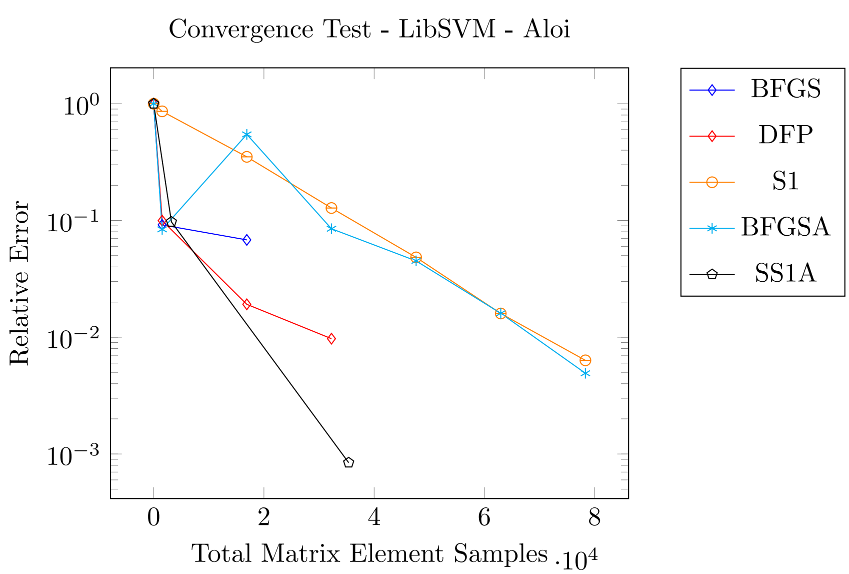

FIG. SM10. *Hessian approximation for the matrix from the LibSVM problem,* ***Aloi*** *($n = 128$) [SM1] with $s = 12 = \lceil\sqrt{128}\rceil$.*

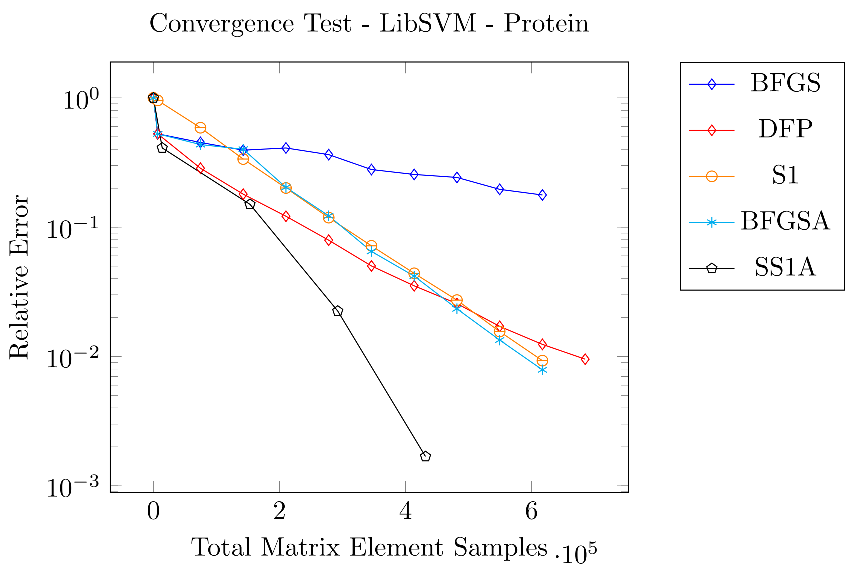

FIG. SM11. *Hessian approximation for the matrix from the LibSVM problem,* ***Protein*** *($n = 357$) [SM1] with $s = 19 = \lceil\sqrt{357}\rceil$.*

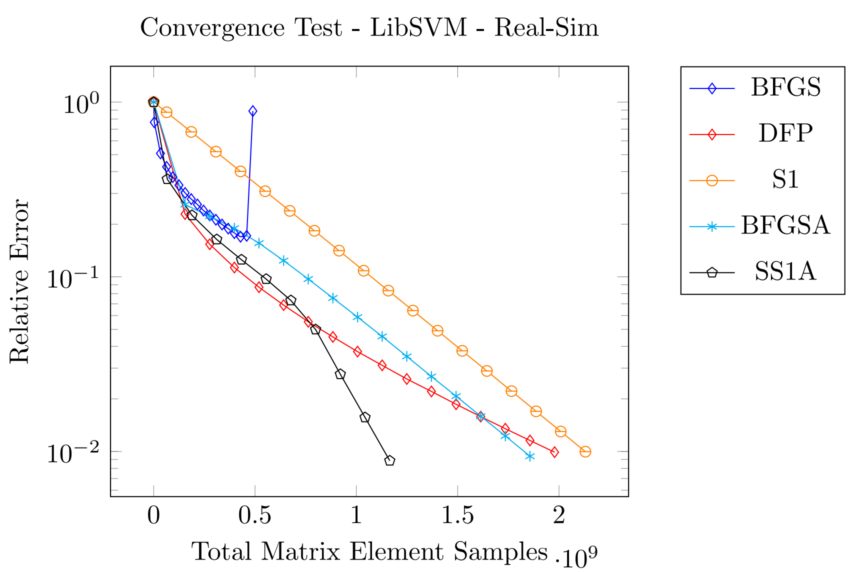

FIG. SM12. *Hessian approximation for the matrix from the LibSVM problem,* ***Real-Sim*** *($n = 20,958$) [SM1] with $s = 145 = \lceil\sqrt{20,958}\rceil$.*

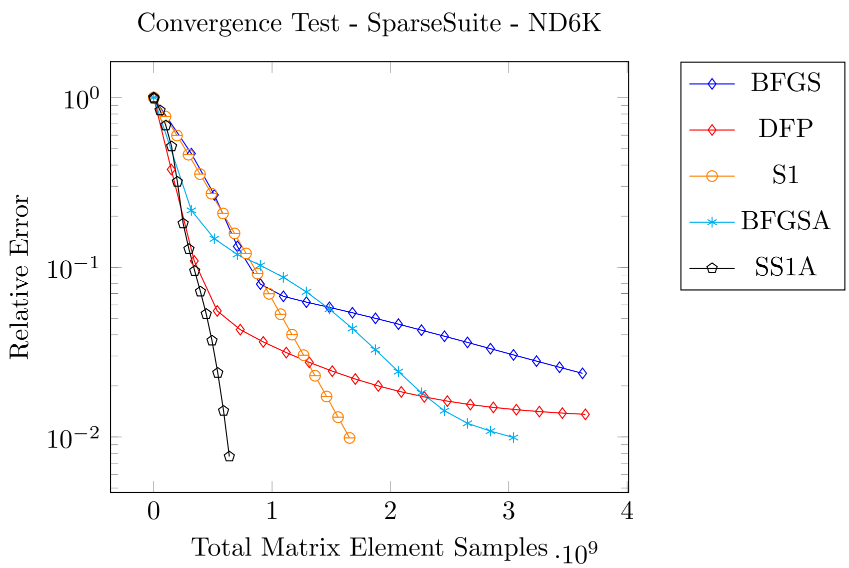

FIG. SM13. *Hessian approximation for the matrix from the Sparse Suite Library,* ***ND6K*** *(*$n = 18,000$*) [SM2] with* $s = 135 = \lceil\sqrt{18,000}\rceil$.

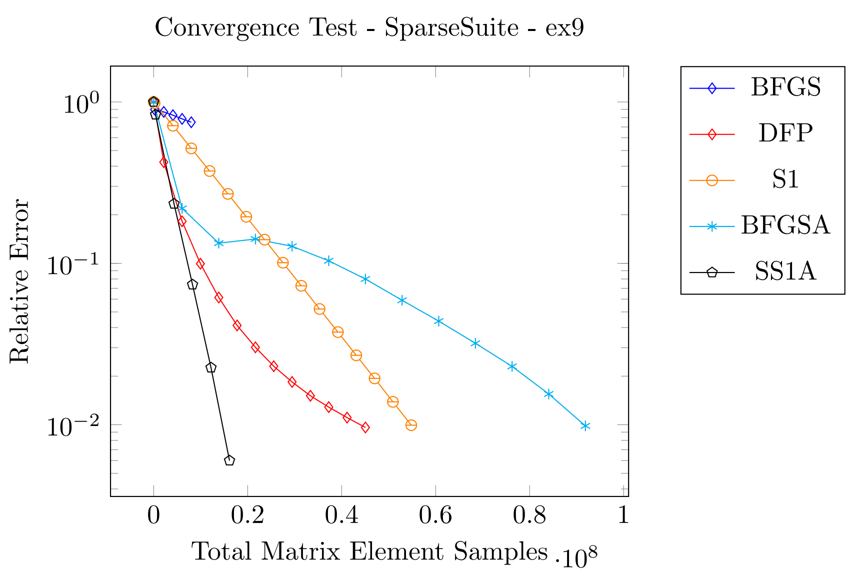

FIG. SM14. *Hessian approximation for the matrix from the Sparse Suite Library,* ***ex9*** *(*$n = 3363$*) [SM2] with* $s = 58 = \lceil\sqrt{3363}\rceil$.

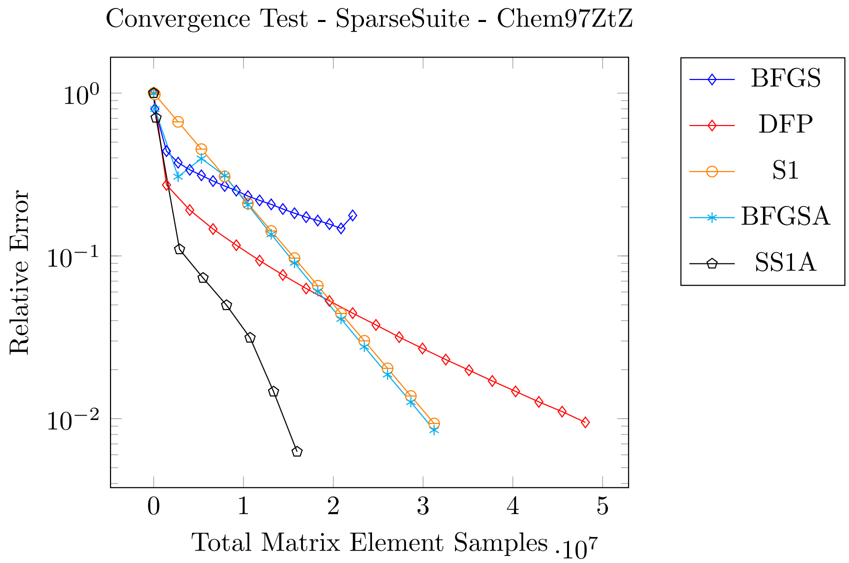

FIG. SM15. *Hessian approximation for the matrix from the Sparse Suite Library,* ***Chem97ZtZ*** *($n = 2541$) [SM2] with $s = 51 = \lceil\sqrt{2541}\rceil$.*

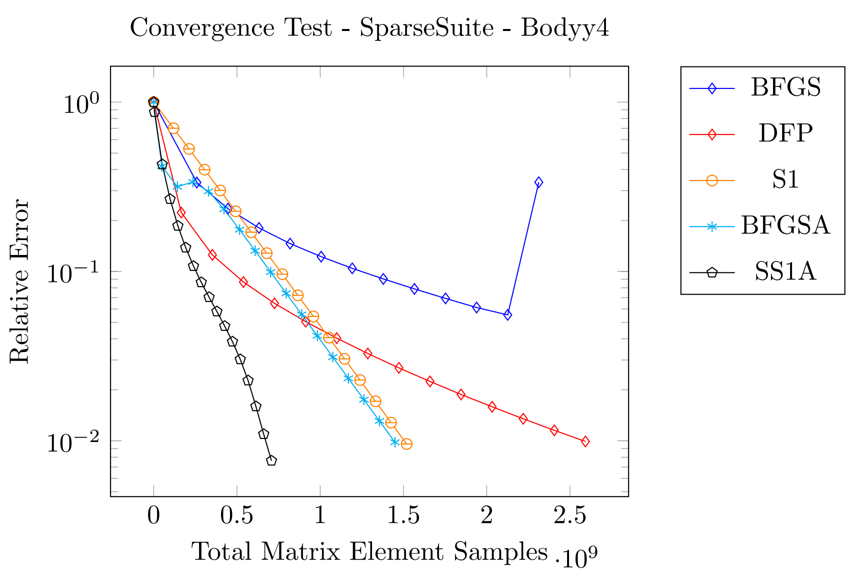

FIG. SM16. *Hessian approximation for the matrix from the Sparse Suite Library,* ***Body*** *($n = 17,546$) [SM2] with $s = 133 = \lceil\sqrt{17,546}\rceil$.*

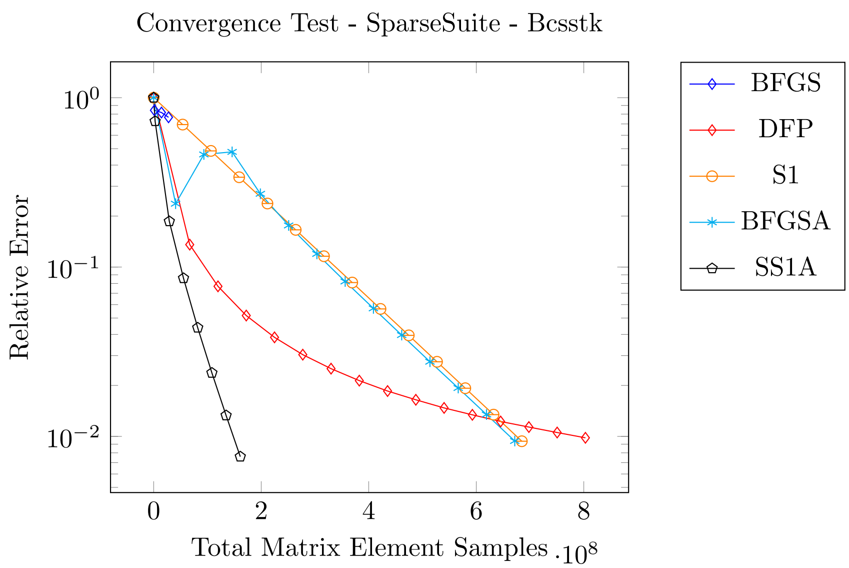

FIG. SM17. *Hessian approximation for the matrix from the Sparse Suite Library,* ***bcsstk*** *(*$n = 11,948$*) [SM2] with* $s = 110 = \lceil\sqrt{11,948}\rceil$.

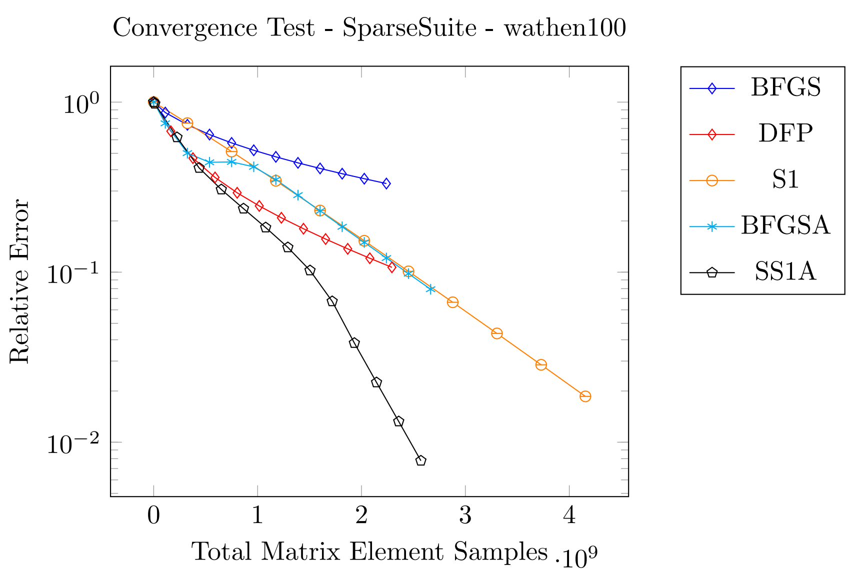

FIG. SM18. *Hessian approximation for the matrix from the Sparse Suite Library,* ***wathen*** *(*$n = 30,401$*) [SM2] with* $s = 175 = \lceil\sqrt{30,401}\rceil$.