**Advances in Computational Mathematics manuscript No.**
(will be inserted by the editor)

# Supplemental Material: Randomized Iterative Methods for Matrix Approximation

**Joy Azzam · Benjamin W. Ong · Allan A. Struthers**

the date of receipt and acceptance should be inserted later

## 1 Additional Non-Accelerated Computational Results

The convergence test from § 4.1 was performed on the remaining matrices tested in [3]. As before, these figures show: BFGS($\diamond$) as specified by eq. (9); DFP ($\diamond$) as specified by eq. (8); NS ($\otimes$) as specified by Algorithm 1; SS1 ($\bullet$) as specified by Algorithm 2; SS2 ($\blacksquare$) as specified by Algorithm 3. All numerical experiments indicate that our non-accelerated sub-sampled algorithms converge predictably and consistently,

– § 1 the LibSVM matrix Aloi of size $n = 128$;
– § 1 the LibSVM matrix Protein of size $n = 357$;
– § 1 the LibSVM matrix Real-Sim of size $n = 20958$;
– § 1 the Sparse Suite matrix ND6K of size $n = 18000$;
– § 1 the Sparse Suite matrix ex9 of size $n = 3363$;
– § 1 the Sparse Suite matrix Chem97ZtZ of size $n = 2541$.
– § 1 the Sparse Suite matrix Body of size $n = 17556$.
– § 1 the Sparse Suite matrix bcsstk of size $n = 11948$.
– § 1 the Sparse Suite matrix wathen of size $n = 30401$.

Plots in § 1 indicate that a maximum running time is reached for the sub-sampled methods.

## 2 Additional Accelerated Computational Results

The convergence test from § 5.1 was performed on the remaining matrices tested in [3]. We illustrate the relative performance of the following algorithms:

– ($*$) BFGSA, eq. (9) with adaptive sampling described in [3],
– ($\circ$) S1, eq. (6) with $W = I_n$,

Address(es) of author(s) should be given

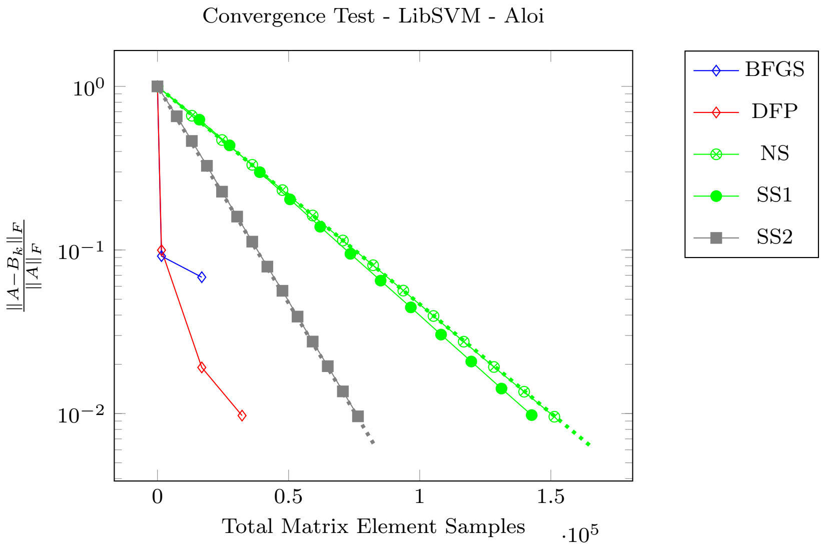


**Fig. 1** Hessian approximation for the matrix from the LibSVM problem, **Aloi** ($n = 128$) [1] with $s = 12 = \lceil\sqrt{128}\rceil$. Dotted lines are theoretical convergence rates. Note, DFP and BFGS perform well.

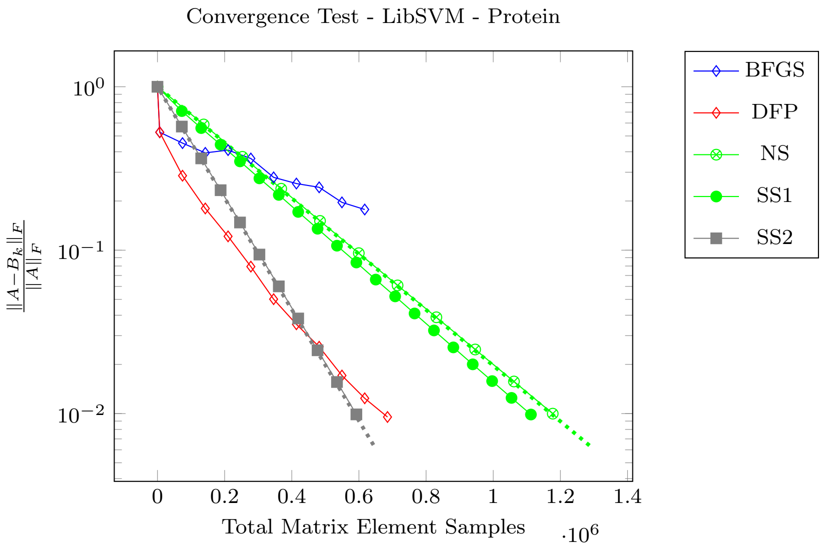


**Fig. 2** Hessian approximation for the matrix from the LibSVM problem, **Protein** ($n = 357$) [1] with $s = 19 = \lceil\sqrt{357}\rceil$. Dotted lines are theoretical convergence rates. Note, DFP performs well, BFGS performs poorly.

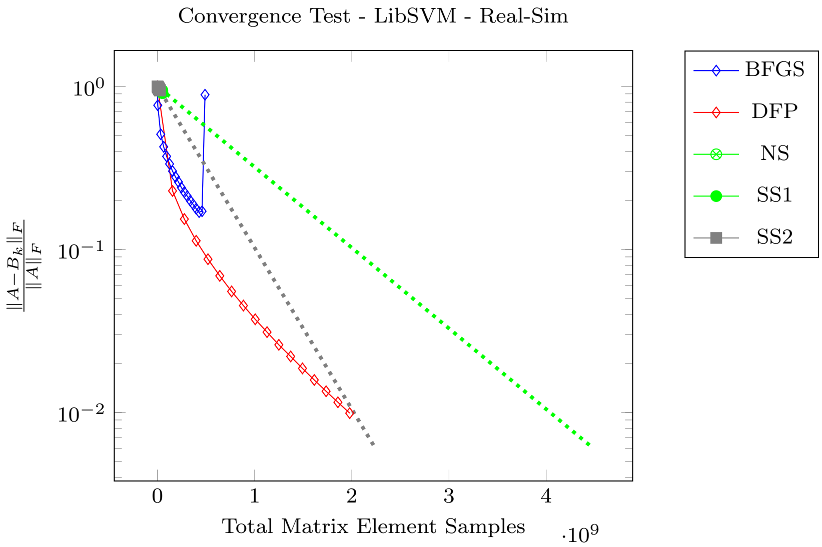


**Fig. 3** Hessian approximation for the matrix from the LibSVM problem, **Real-Sim** ($n = 20,958$) [1] with $s = 145 = \lceil\sqrt{20,958}\rceil$. Dotted lines are theoretical convergence rates for our algorithms. DFP performs well, BFGS does not converge.

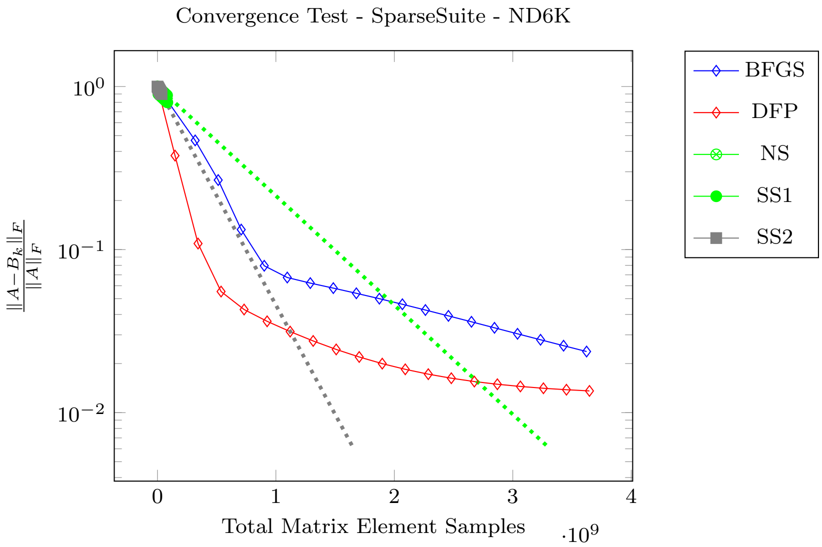


**Fig. 4** Hessian approximation for the matrix from the Sparse Suite Library, **ND6K** ($n = 18,000$) [2] with $s = 135 = \lceil\sqrt{18,000}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

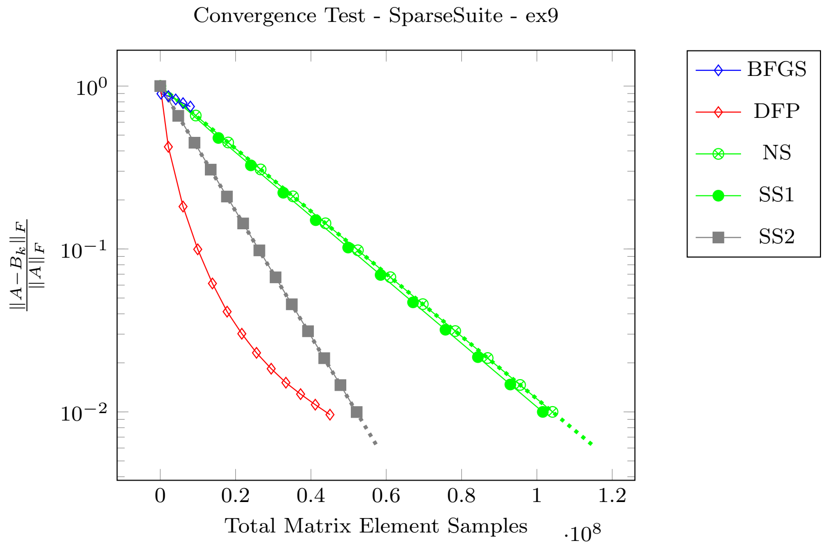


**Fig. 5** Hessian approximation for the matrix from the Sparse Suite Library, **ex9** ($n = 3363$) [2] with $s = 58 = \lceil\sqrt{3363}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

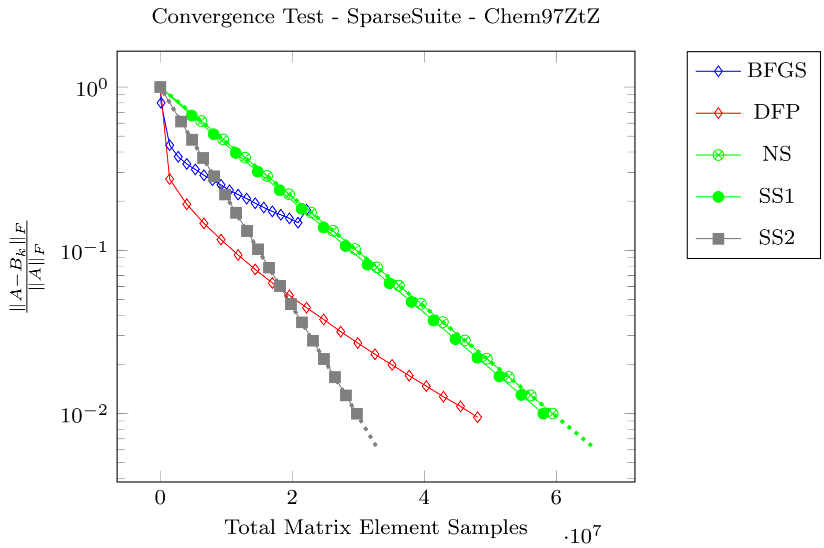


**Fig. 6** Hessian approximation for the matrix from the Sparse Suite Library, **Chem97ZtZ** ($n = 2541$) [2] with $s = 51 = \lceil\sqrt{2541}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

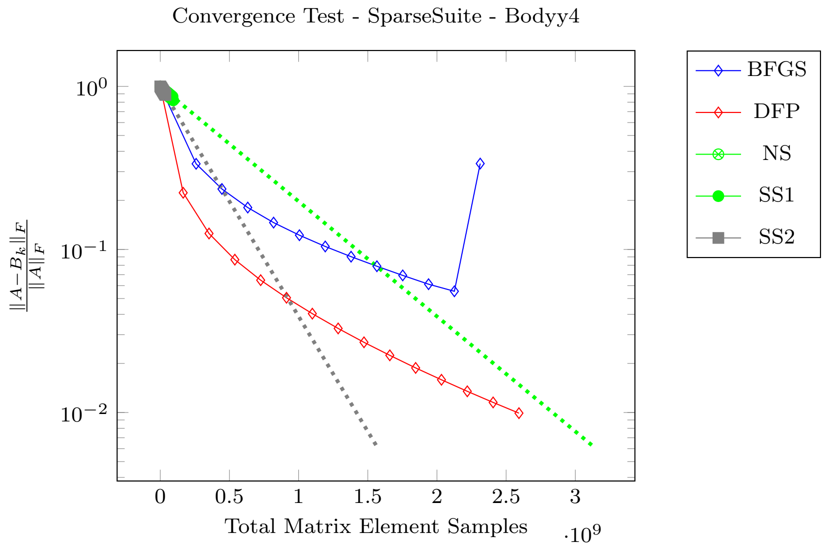


**Fig. 7** Hessian approximation for the matrix from the Sparse Suite Library, **Body** ($n = 17,546$) [2] with $s = 133 = \lceil\sqrt{17,546}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

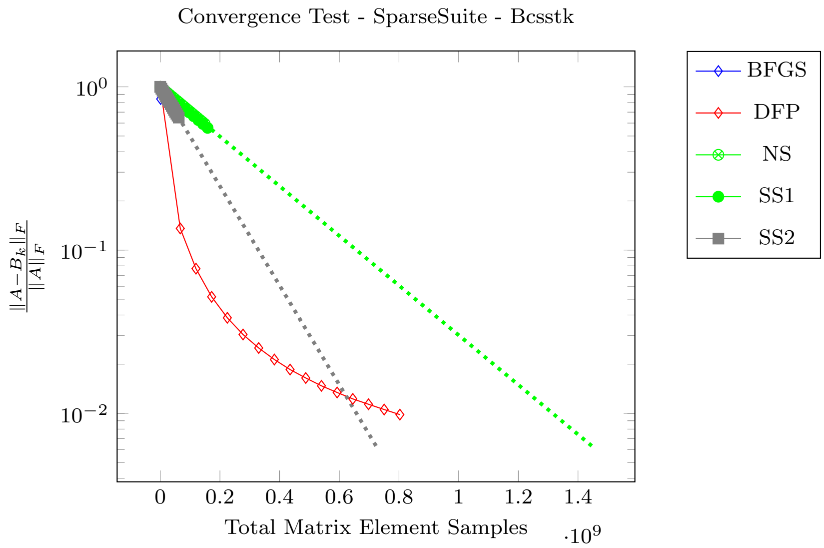


**Fig. 8** Hessian approximation for the matrix from the Sparse Suite Library, **bcsstk** ($n = 11,948$) [2] with $s = 110 = \lceil\sqrt{11,948}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

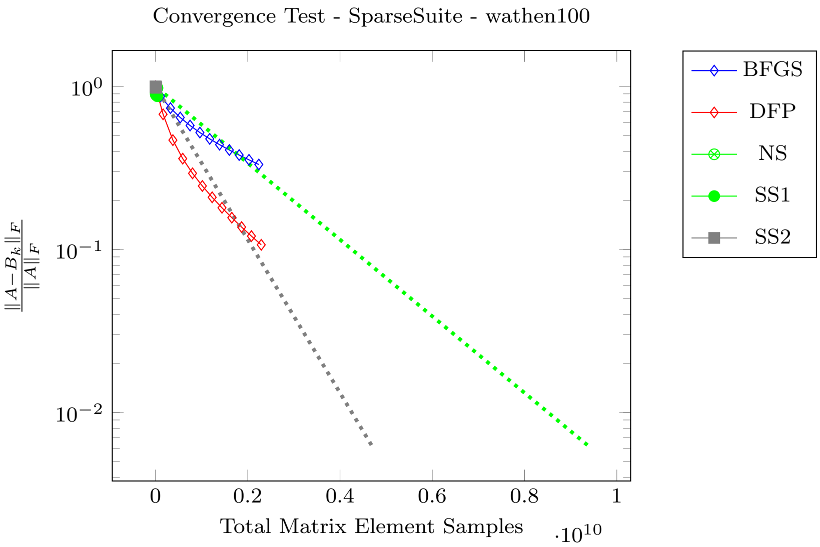


**Fig. 9** Hessian approximation for the matrix from the Sparse Suite Library, **wathen** ($n = 30,401$) [2] with $s = 175 = \lceil\sqrt{30,401}\rceil$. Dotted lines are theoretical convergence rates for our algorithms.

– (⬠) SS1A+, Algorithm 4
– (◇) BFGS, eq. (9),
– (◇) DFP, eq. (8)

on the following matrices:

– § 2 the LibSVM matrix Aloi of size $n = 128$;
– § 2 the LibSVM matrix Protein of size $n = 357$;
– § 2 the LibSVM matrix Real-Sim of size $n = 20958$;
– § 2 the Sparse Suite matrix ND6K of size $n = 18000$;
– § 2 the Sparse Suite matrix ex9 of size $n = 3363$;
– § 2 the Sparse Suite matrix Chem97ZtZ of size $n = 2541$.
– § 2 the Sparse Suite matrix Body of size $n = 17556$.
– § 2 the Sparse Suite matrix bcsstk of size $n = 11948$.
– § 2 the Sparse Suite matrix wathen of size $n = 30401$.

The accelerated method algorithm 4 performs well on all matrices including those with large $n \approx 10^4$ (see § 2).

## References

1. 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`.
2. 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`.

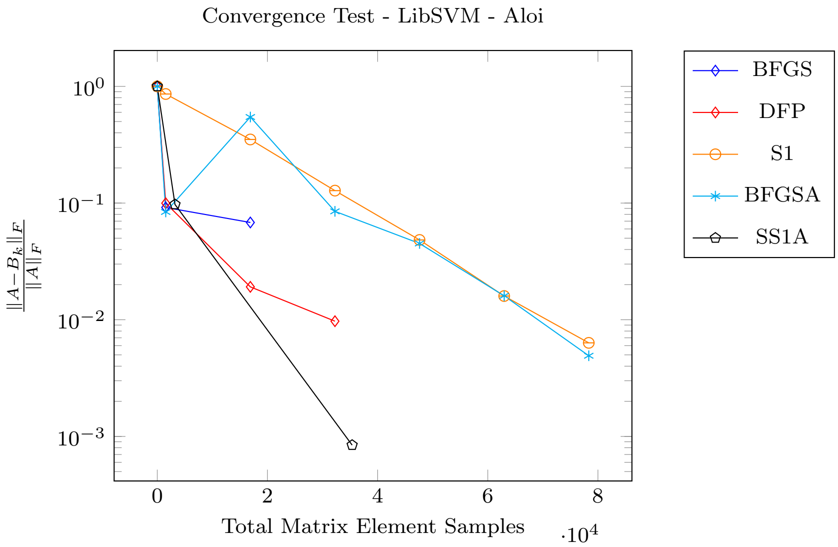


**Fig. 10** Hessian approximation for the matrix from the LibSVM problem, **Aloi** ($n = 128$) [1] with $s = 12 = \lceil\sqrt{128}\rceil$.

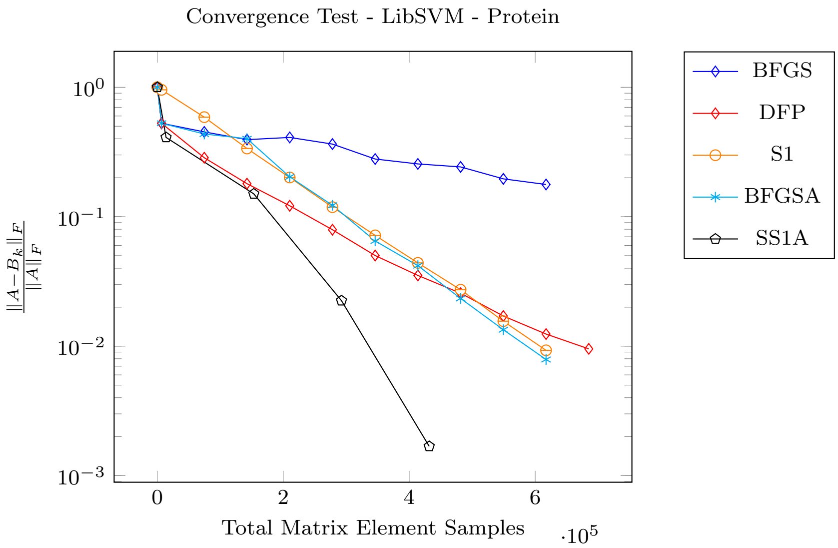


**Fig. 11** Hessian approximation for the matrix from the LibSVM problem, **Protein** ($n = 357$) [1] with $s = 19 = \lceil\sqrt{357}\rceil$.

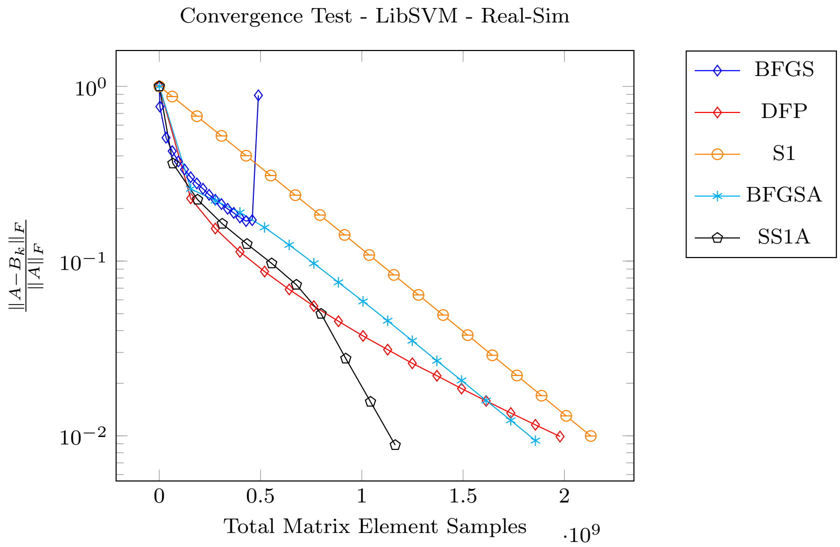


**Fig. 12** Hessian approximation for the matrix from the LibSVM problem, **Real-Sim** ($n = 20,958$) [1] with $s = 145 = \lceil\sqrt{20,958}\rceil$.

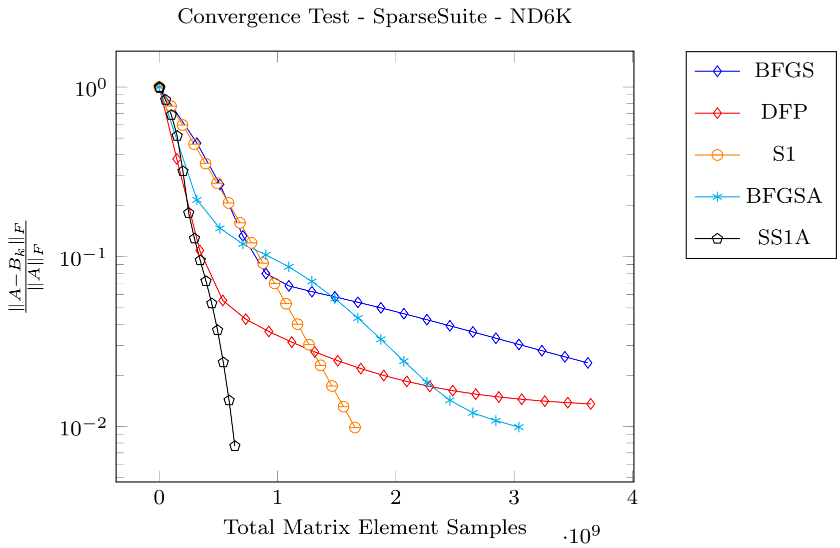


**Fig. 13** Hessian approximation for the matrix from the Sparse Suite Library, **ND6K** ($n = 18,000$) [2] with $s = 135 = \lceil\sqrt{18,000}\rceil$.

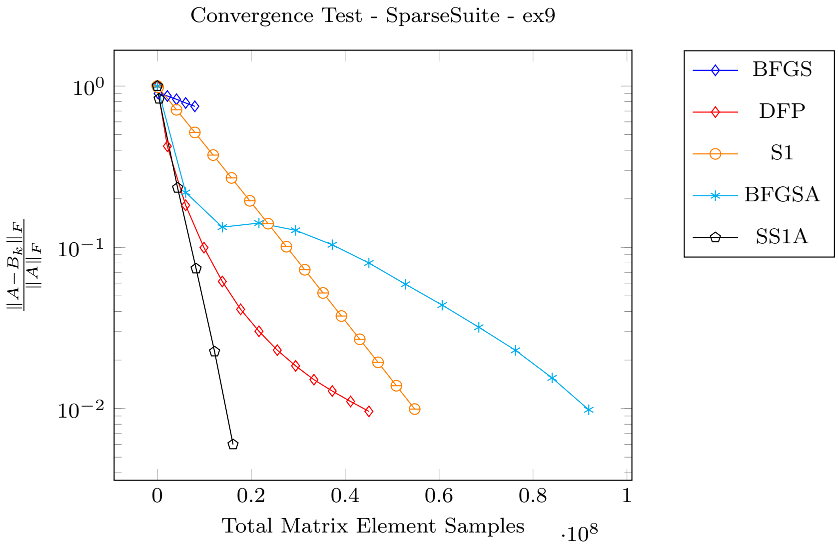


**Fig. 14** Hessian approximation for the matrix from the Sparse Suite Library, **ex9** ($n = 3363$) [2] with $s = 58 = \lceil\sqrt{3363}\rceil$.

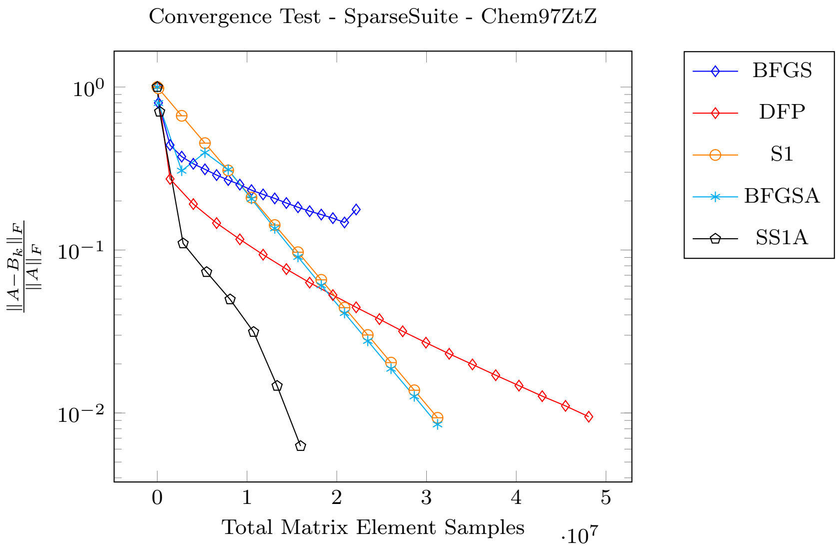


**Fig. 15** Hessian approximation for the matrix from the Sparse Suite Library, **Chem97ZtZ** ($n = 2541$) [2] with $s = 51 = \lceil\sqrt{2541}\rceil$.

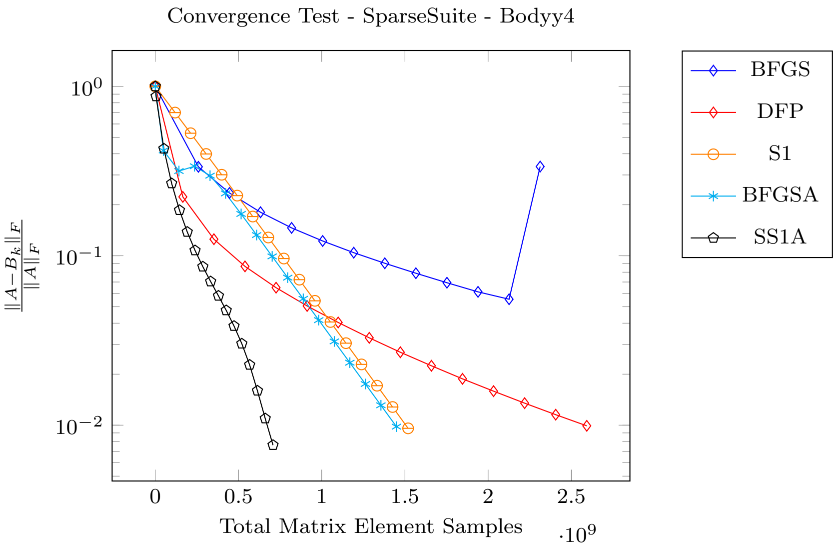


**Fig. 16** Hessian approximation for the matrix from the Sparse Suite Library, **Body** ($n = 17{,}546$) [2] with $s = 133 = \lceil\sqrt{17{,}546}\rceil$.

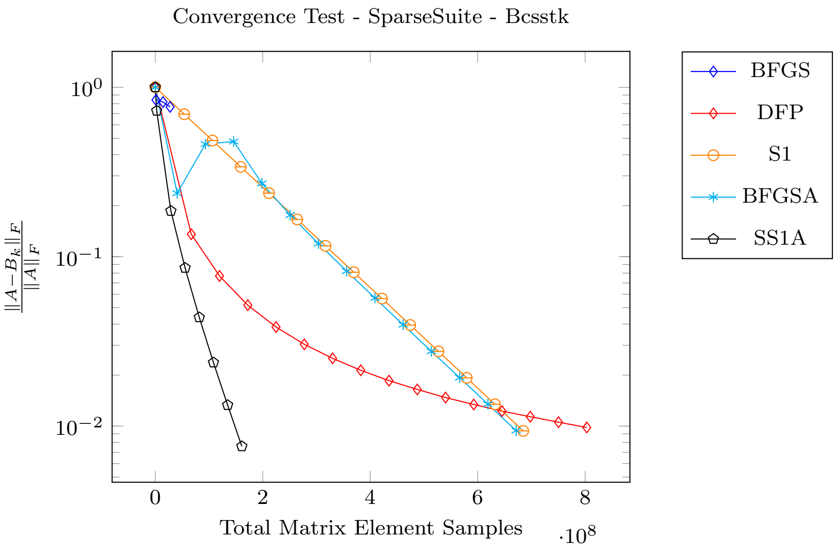


**Fig. 17** Hessian approximation for the matrix from the Sparse Suite Library, **bcsstk** ($n = 11{,}948$) [2] with $s = 110 = \lceil\sqrt{11{,}948}\rceil$.

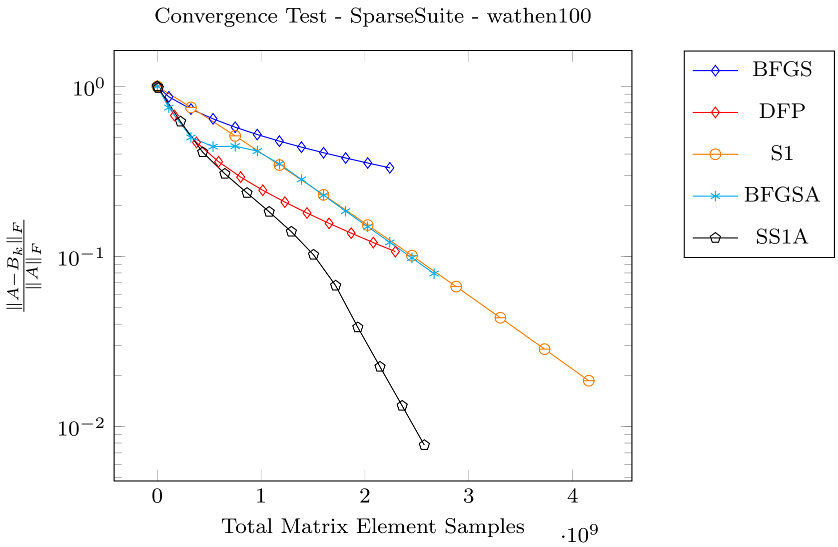


**Fig. 18** Hessian approximation for the matrix from the Sparse Suite Library, **wathen** ($n = 30,401$) [2] with $s = 175 = \lceil\sqrt{30,401}\rceil$.

3. 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.