Homework Assignment 2: Sparse Matrix Multiplication in Fortran

Carlos Rodriguez-Arias Santiago, Nuria Vides García

January 10, 2025

Method	**Size**	
	25x25	125x125
Sparse	1%: 2.7400E-07	1%: 1.5230E-05
	5%: 1.3650E-06	2%: 1.4164E-03
	10%: 3.2560E-06	5%: 4.5052E-04
	25%: 1.7921E-05	10%: 1.9142E-03
	50%: 7.5588E-05	25%: 1.0634E-02
		50%: 4.0582E-02
Dense (hand-written)	1%: 1.0817E-04	1%: 1.2637E-02
	5%: 1.0886E-04	2%: 1.2636E-02
	10%: 1.0792E-04	5%: 1.2638E-02
	25%: 1.0952E-04	10%: 1.2637E-02
	50%: 1.0876E-04	25%: 1.2653E-02
		50%: 1.2654E-02
Dense (BLAS)	1%: 1.6156E-05	1%: 1.4148E-03
	5%: 1.6233E-05	2%: 1.4164E-03
	10%: 1.6145E-05	5%: 1.4157E-03
	25%: 1.6106E-05	10%: 1.4160E-03
	50%: 1.5738E-05	25%: 1.4158E-03
		50%: 1.4095E-03

Table 1: Execution time table based on matrix size and fill percentage.

Discussion

Analysis by Matrix Size

Matrix Size 25x25: For sparse matrices, the execution time is significantly lower compared to dense matrices (both hand-written and BLAS methods), especially as the percentage of non-zero elements increases. The dense meth-

ods (hand-written and BLAS) show similar execution times, though BLAS is generally faster than the hand-written method, particularly for larger matrices.
Matrix Size 125x125: As the matrix size increases to 125x125, the execution time for all methods increases, as expected. However, the performance difference between the methods becomes more noticeable. Sparse matrices remain significantly faster than dense matrices. Among the dense methods, BLAS continues to show a slight advantage over the hand-written method.

Analysis by Fill Percentage (Non-Zero Elements)

Low Fill Percentage (1% and 2%): For matrices with low fill percentages (1% and 2%), the time differences between the three methods are relatively small. The sparse method remains the fastest, with noticeable differences compared to the dense methods, but the advantage is not as pronounced.
Moderate Fill Percentage (5% - 10%): As the percentage of non-zero elements increases (from 5% to 10%), the execution time for sparse matrices starts to increase slightly, though it remains significantly faster than the dense methods. There is not much variation in execution times between the two dense methods.
High Fill Percentage (25% - 50%): When the fill percentage increases further to 25% and 50%, the execution time for sparse matrices approaches that of the dense methods, but it is still more efficient. In these cases, the performance of the dense matrices (both hand-written and BLAS) is more homogeneous. The execution time for dense matrices (both BLAS and hand-written) remains fairly stable as the fill percentage increases, with BLAS being slightly faster than the hand-written method.

Summary of Efficiency and Scaling Behavior

Sparse: Sparse matrices show much better performance as the matrix size or fill percentage increases, compared to dense methods. As the percentage of non-zero elements increases, the execution time for the sparse method grows, but it remains more efficient than the dense methods in most cases.
Dense (hand-written): The hand-written dense method shows relatively constant execution times that do not depend much on the matrix size or fill percentage. However, it is generally slower than both the BLAS method and the sparse method, especially as the matrix size grows.
Dense (BLAS): The BLAS method is generally more efficient than the hand-written method, especially for larger matrices (such as 125x125), where BLAS optimization makes a noticeable difference. However, even though BLAS improves speed, it does not match the efficiency of sparse matrices when dealing with matrices that have a low number of non-zero elements.

Conclusions

- **Sparse method** is the most efficient for small to medium matrices with a low fill percentage (1%-10%).
- **Dense (BLAS) method** is the most efficient for large dense matrices and is slightly better than the hand-written method.
- **Dense (hand-written) method** is slower and becomes less efficient compared to the other two methods as the matrix size and fill percentage increase.