Computing the Groebner basis of an ideal with respect to a term ordering is an essential step in solving systems of polynomials. Certain term orderings, such as the degree reverse lexicographical ordering (degrevlex), make the computation of the Groebner basis faster, while other orderings, such as the lexicographical ordering (lex), make it easier to interpret when used for polynomial system solving. Thus, the Groebner basis is often computed first using degrevlex ordering then convert to lex ordering.

One such algorithm is the sparse FGLM. Given a Groebner basis G of a zero dimensional ideal $I \subset \mathbb{K}[x_1, \ldots, x_n]$ in shape position and the multiplication matrices $T_1, \ldots, T_n \in \mathbb{K}^{D \times D}$, where D is the number of monomials in the canonical basis of G, it computes linear sequences of the form $s_1^{(i)} = (uT_1^i v_1)_{1 \leq i \leq 2D}$ and $s_j^{(i)} = (uT_1^i v_j)_{1 \leq i \leq D}$, $2 \leq j \leq n$, for some vectors $u, v_j \in \mathbb{K}^{D \times 1}$. One can view this as an application of the Wiedemann algorithm for sparse linear algebra The computation of these sequences is difficult to parallelize since previous terms are required to compute the next.

When the ideal is radical, we are developing an algorithm that uses small matrices instead of single vectors inspired by the block Wiedemann algorithm of Coppersmith. The block Wiedemann algorithm has been widely studied and have shown to be successful in problems such as integer factorization. More precisely, given the same input as above, let $u, v_j \in \mathbb{K}^{D \times M}$ be matrices with random entries; we compute the matrix sequences $s_1^{(i)} = (uT_1^i v_1)_{1 \leq i \leq \frac{2D}{M}}$ and $s_j^{(i)}(uT_1^i T_j v_1)_{1 \leq i \leq \frac{D}{M}}$. Then, we find $C_0, \ldots, C_D$ such that for any $t \geq 0$,

$$C_0 \cdot s_1^{(t)} + C_1 \cdot s_1^{(t+1)} + \cdots + C_{\frac{D}{M}} \cdot s_1^{(t+\frac{D}{M})} = 0$$