Sparse Matrix Algebra for Quantum Modeling of Very Large Systems

Paweł Sałek and Emanuel Rubensson

Theoretical Chemistry, KTH
pawsa@theochem.kth.se,
WWW home page: http://www.theochem.kth.se/~pawsa/

Abstract. Matrices appearing in Hartree-Fock or density functional theory become sparse when separation between atoms exceeds system-dependent threshold value. The sparsity is block-wise where each block corresponds to an atom or a group of atoms. The sparsity is maintained by filtering out small elements below predetermined threshold. This filtering has to be done in a systematic fashion to rigorously control the error in the calculation. We describe a general method that provides a strict error control in blocked sparse matrix algebra and present its applications to matrix-matrix multiplication, the Trace Correcting Purification algorithm and the entire self-consistent field calculation.

1 Introduction

Large scale Hartree-Fock (HF) or density functional theory (DFT) in Kohn-Sham (KS) formalism consist of two time consuming steps: evaluation of the HF/KS matrix (F) and search for the corresponding density matrix. The former step involves evaluation of various kinds of integrals, out of which Coulomb, exchange and exchange-correlation are the most time-consuming ones. In this work, we focus on the latter one. The formation of the new density matrix is traditionally performed via diagonalization of the HF/KS matrix and use of the eigenvectors C^{occ} associated with the smallest eigenvalues to obtain the elements of the density matrix D:

$$FC^{\text{occ}} = \varepsilon SC^{\text{occ}} \qquad \rightarrow \qquad D = C^{\text{occ}}(C^{\text{occ}})^T \tag{1}$$

where S is the basis set overlap matrix. This operation scales cubically with the problem size and becomes the bottleneck for large systems. A method that takes advantage of the existing sparsity in the F matrix is needed. Several algorithms have been proposed, all of which compute the solution by iterative matrix-matrix multiplication. Their performance is therefore closely connected to the efficiency of the multiplications. The multiplications can scale linearly if multiplications by zeros which are present in sparse matrices are avoided. Care is needed to avoid fill-in. The fill-in can be prevented by filtering out small elements. A systematic filtering algorithm will control the error propagation and contain it under user-requested threshold.

2 Block Sparse matrices in KS method.

One feature that distinguishes matrices appearing in KS method is their rather limited sparsity. It is not extremely uncommon that 5000x5000 matrices have only 50% of elements below 10^{-8}. While sparsity is larger in many cases, one has to be able to handle semi-dense cases as well. We use for our work a multi-atom blocked sparse representation [1]. This representation provides a good performance by reordering the atoms and associated basis functions to move together matrix elements associated with atoms close in space.

2.1 Systematic Small-Submatrix Selection Algorithm

There exist several ways to prevent the fill-in by dropping small elements. One appealing way is to explicitly take advantage of the dependence of matrix element magnitude on distance between corresponding basis function centers [1, 2]. A submatrix is dropped when the shortest distance between the two atom groups is greater than a predefined cutoff radius. This approach – in spite of its simplicity and predictability – has some considerable deficiencies. It is rarely known which cutoff radius will correspond to a certain drop tolerance and an assumption has to be made about either the properties of the matrix or the molecule itself. In case of a density matrix, an exponential decay dependence of the matrix element on the distance can be used, with the exponent related to the band gap [3, 4]. The gap in the calculation depends mostly on the physical property of the system but also on chosen basis set or the Hamiltonian. Incorrect assumptions (eg. inaccurate band gap estimation) will cause severe difficulties with error control. In such a case, one can never be certain that all of the dropped submatrices were under the requested threshold. Such scheme may also unnecessarily include small blocks with a negligible contribution [5].

Another way to remove unnecessary elements is to look at the maximum absolute element in the submatrix. The entire submatrix is dropped if this element is smaller than a preselected threshold. One is able in this way to strictly control the error. Maximum absolute element in the matrix is one of many possible choices of the norm. This norm however can be quite inefficient, i.e. it can result in including elements contributing little to the overall result. The reason is that one has to assume that all elements in the matrix are as large as the largest element. A tighter norm that better estimates the magnitude of the submatrix can substantially improve performance.

Our method which we call a Systematic Small-Submatrix Selection Algorithm (SSSA) aims at improving the truncation efficiency by using an iterative process that is closely related to the definition of the norm [6]. The condition for removing submatrices is formulated in terms of entire rows if the infinity norm is used, entire columns if the 1-norm is used and entire matrices if the Frobenius norm is used. This contrasts with having a condition on each single submatrix. If, for example, the 1-norm is used one should repeatedly remove the smallest (the one with smallest norm) submatrix in the column as long as the sum of the removed submatrices' norms is smaller than the error limit ε. This process is

repeated for each column in the matrix. However, if one is to search for the smallest submatrix in each iteration this can be needlessly tedious. Therefore it is desired to keep the columns ordered by the 1-norm of each submatrix. The commonly used Compressed Sparse Column (CSC) representation does not require the columns to be ordered after row position in the matrix. The matrix-matrix multiplication implementation does not require the submatrices to be ordered, either. For this reason, efficient truncation can be achieved by keeping each column sorted in descending order after the 1-norm of each submatrix. Then submatrices are removed from the end as long as the sum of their 1-norms is smaller than ε. This approach is particularly efficient for large submatrices since only the submatrix norms need to be sorted, i.e. one value per submatrix. It can also be applied to other sparse matrix representations.

3 Applications

3.1 Accumulated Error in Trace Correcting Purification

Density matrix purification has been proposed in a multitude of variants [7–11]. The purification algorithms rely on the fact that Fock and density matrices share a common set of eigenvectors but have different eigenvalues. One therefore applies a series of transforms to the Fock matrix so that the eigenvalues corresponding to occupied eigenvectors converge to 1 and the remaining eigenvalues converge to 0.

The simplest purification algorithm, developed by Niklasson, called trace-correcting purification (TCP) [8], is not only the easiest one to implement but also very competitive when it comes to performance measured in number of matrix-matrix multiplications [8, 10]. The TCP algorithm assumes orthogonal basis set, i.e. the overlap matrix $S = I$. Therefore, the generalized eigenvalue problem in Eq. 1 has to be transformed to standard form. Once this has been done, one proceeds as follows:

```
compute P = (lmax I-F)/(lmax-lmin)
while abs(trace(P)-N)>threshold
    if(trace(P)>N) then
        P := P*P
     else
        P := 2*P-P*P
end while
```

where `lmax` and `lmin` are upper boundary of the largest and lower boundary of the lowest eigenvalues, respectively. The final result after n steps is contained in P_n. TCP algorithm accumulates the truncation error over the iterations. It is therefore crucial to use a systematic filtering in order to control the accumulation error [6].

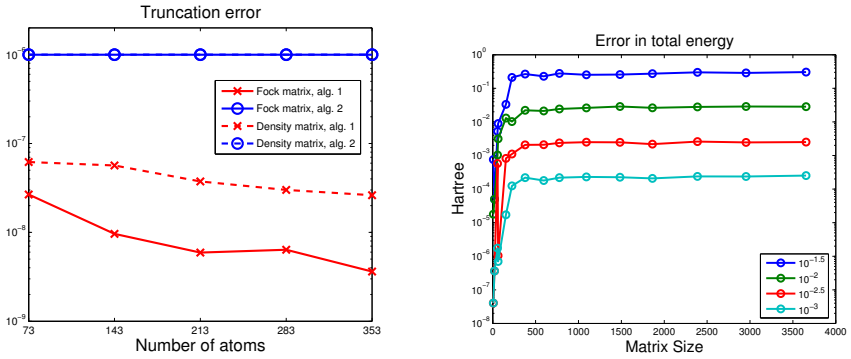

Fig. 1. Left panel: Truncation error introduced by filtering algorithms: traditional threshold-based filtering (alg.1) and SSSA (alg.2). Right panel: Error in the final SCF energy as a function of selected SSSA threshold and the system size. For systems affected by truncation errors, SSSA algorithm keeps their impact at a strictly controlled level.

3.2 Impact of filtering on SCF energy

Choice of the filtering algorithm has fundamental impact on the outcome of the calculation. An inadequate filtering algorithm may cause that multiplications by negligibly small blocks are performed — which will damage the performance of the algorithm. Figure 1 shows that SSSA provides exactly requested accuracy of the calculation, while a simple-minded filtering based only on the block norm may cause that more and more small blocks get included in the calculation, making it more expensive than necessary.

References

1. C. Saravanan, Y. Shao, R. Baer, P.N. Ross, and M. Head-Gordon, J. Comp. Chem. **24**, 618 (2003).
2. X.-P. Li, R.W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
3. E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).
4. P.E. Maslen, C. Ochsenfeld, C.A. White, M.S. Lee, and M. Head-Gordon, J. Phys. Chem. A **102**, 2215 (1998).
5. M. Challacombe, Comp. Phys. Comm. **128**, 93 (2000).
6. E.H. Rubensson and P. Sałek, J. Comp. Chem. 26, 1628-1637 (2005).
7. A.H.R. Palser and D.E. Manolopoulos, Phys. Rev. B **58**, 12704 (1998).
8. A.M.N. Niklasson, Phys. Rev. B **66**, 155115 (2002).
9. A.M.N. Niklasson, C.J. Tymczak, and M. Challacombe, J. Chem. Phys. **118**, 8611 (2003).
10. D.A. Mazziotti, Phys. Rev. E **68**, 066701 (2003).
11. A. Holas, Chem. Phys. Lett. **340**, 552 (2001).