# Numerical operator calculus in higher dimensions
# Beylkin and Mohlenkamp referred to in Udo's paper using AG to decompose tensors for MP2 calculation

## 1 introduction

When an algorithm is extended from d=1 to D in nearly every case its comp cost is taken to the power D. "curse of dimensionality".
propose the use of a representation for vectors and matrices that generalizes separation of variables while allowing controlled accuracy.

In physical problems computational complexity grows exponentially in the physical dimension. Present an approach that allos one dimensional algorithms to be extended to D dimensions without computational complexity growing exponentially.
In physics, separation of variables has been the most successful approach for avoiding the high cost of working in D dimensions. For example, instead of trying to find a D-dimensional function that solves the given equation (eg the multiparticle schrodinger equation) one only considers functions that can be product of one body functions. one produces a system of d weakly coupled one-dimensional equations for the functions. Iteratively solving the equation one can obtain an approximate solution to the original equation using only one-dimensional operation.
A natural extension, in hopes to improve accuracy, is the following

$$f(x_1, ..., x_D) = \sum_{l=1}^{r} s_l \phi_1^l(x_1) ... \phi_d^l(x_d)$$

r is called the separation rank.
by increasing r the approximate solution can be made as accurate as desired. This can be done by fixing the function and solving for the coefficients. The options include the use of a tensor product basis as well as configuration interaction methods. Another option is to sub the equation into the original equation. Unf the result equations are generally hard to deal with, since the equations are strongly coupled.
**see ref 2**
some analytic questions arise from this equation. What class of functions can be represented efficiently in this form? Some examples would be, one can characterize a class based on mixed derivatives of order k. Another example, "sparse-grids" method identify classes of functions compatible with certain representations and reduction of computational complexity can be achieved. Approach presented relies on properties of physically significant operators.
second question, how does one find the representation in the eqaution for a given function.

## 2 summary of the paper

The equation above is used as an approximation technique for functions and operators and the authors try to minimize the number of terms at each step of the algorithm. operators can be expressed in a similar form to the one above

$$A = \sum_{l=1}^{r} s_l B_1^l \otimes ... \otimes B_d^l$$

Linear algebra operations can be performed while keeping all objects in the form of the two equations. Can perform operation in d dimension using combinations of one-dimensional operators. Then solves the original equations directly while maintaining the intermediate functions and operators in the forms above, with adaptively chosen r.

Linear algebra operations can leat to objects that have large separation rank, r. At each step of the algorithm one seeks to minimize the separation rank by adaptively changing the coefficients and functios while maintaining accuracy.

# 3   The separated Representation

we require $A - \sum_{l=1}^{r} s_l V_1^l(j_1, j_1') V_2^l(j_2, j_2')...V_d^l(j_d, j_d')$ where s is a scalar s1 ¿= ... ¿=sr ¿0 and $V_i^l$ are matrices in dimension one with norm 1. and we require

$$\|A - \sum_{l=1}^{r} s_l V_1^l \otimes V_2^l \otimes ... \otimes V_d^l\| < \epsilon$$

sl separation values and r is separation rank. smallest r is the optimal separation rank.

In dimension 2 the separation rep in E4 reduces to a form similar to SVD and in fact can construct an optimal representation using an ordinary SVD algorithm but with an unusual pairing of indices. Instead of separating the vectors in dimension 2 in the input from the ouput coordinates, matrices are separated in dimension 1 in the j1 direction from matrices in dimension 1 in the j2 direction. Thus common matrix operations that have full rank operators still have low rank separation rank.

When d¿2 this representation is not unique even when r is the optimal separation rank. The optimal reps for different values of $\epsilon$ are not nested so one cannot add or delete terms from the sum equation in this section and retain a representation with optimal separation rank.

**look to ref 6 and 7**

# 4   The multiparticle schrodinger operator

In this section show that the separation rank of an appropriate approximation of H grows only logarithmically in d. Use the three dimensional operators as elementary building blocks.
$H = -\Delta + N + E$ where the laplacian is defined by

$$\Delta = (\Delta_1 \otimes I_2 \odot ...I_d) + ...(I_1 \otimes ... \otimes \Delta_d)$$

Where I is the identity
N is nulcear potential defined similarly to delta. The electron-electron interaction is defined as

$$E = \sum_{i=1}^{d-1} \sum_{m=i+1}^{d} W_{im}$$

Vi is the operator that multiplies by the nuclear potential function v(x) in the three dimensional variable $x_i$ and $W_{im}$ is multiplication by the electron-electron interaction (coulomb) potential w(xi-xm). In paper Theorem 2 provides bounds on the separation rank and proof provides a construction for their separated reps.

The theorems are confusing look into them again.

# 5   Basic Linear Algebra

The main point of the separated rep is that the elementary objects which we operate are one dimensional. Assume all objects have been discretized using N points in each direction, so a vector in dimension d has $N^d$ entries. IN d dimensions a dense matrix has $(N^2)^d$ entries whereas a banded matrix has $(bN)^d$ entries. In the separated representation we will need to store $d * r * N^2$ entries if the matrices are dense or $drbN$

entries if they are banded. Banded case demonstrates the effect of combing the separated representation with a fast one dimensional algorithm
it is possible to solve a linear system in the separated representation by posing the problem as a least squares problem and then doing a variant of the separation rank reduction algorithm described below.

# 6   finding "optimal" representations

The key to success is finding a separated representation with low separation rank for matrices of interest. Assuming A matrix is given in the separated representation but with larger separation rank r than necessary. The algorithm described is effective though is not gauranteed to find the optimal representation. Finding the optimal separation rank is not required for the approach.
The strategy in this method is to find the best approximation with rank is to begin with r=1 and then increase r and try again until the residual is less than $\epsilon$. Other initialization strategies exist

**see reference 11**

before starting reduce search space by rotating each direction into an efficient basis. for each fixed direction i, the matrix $\{V_i^l\}_{l=1}^r$ each have $N^2$ entries and span a vector space of dimension Mi ¡ min(r, $N^2$). The matrix V can be expressed in this basis as a vector V of length Mi. THe matrix A is expressed as the vector

$$A = \sum_{l=1}^r s_l V_1^l \otimes V_2^l \otimes ... \otimes V_d^l$$

The authors reduce the separation rank of A then undo the change of basis to recover the matrix A.

# 7   Alternating Least Squares

To find an r which minimizes the error is to solve the non-linear least squares problem. to make the problem do-able exploit the multilinearity of the problem and use an ALS approach. The key difference is that the input is a vector in the separated representation rather than a dense data vector in dimension d. The processes is the same as I have it now repeat the loop k iteratively minimizing the error between A and the CP decomposition until the norm residual is below some value.
The linear least squares problem naturally divides into separate problems for each coordinate. For fixed direction,k, form the matrix B with entries

$$B(\hat{l}, \tilde{l}) = \Pi_{i \neq k} < \tilde{V}_i^{\tilde{l}}, \tilde{V}_i^{\hat{l}} >$$

Then for a fixed coor jk , form the vector bjk with entries

$$b_{j_k}(\hat{l}) = \sum_{l=1}^r s_l V_k^{(}(j_k) P i_{i! = k} < \tilde{V}_i^{\tilde{l}}, \tilde{V}_i^{\hat{l}} >$$