In many material science and condensed matter applications, the state of the system can be described entirely by a density matrix, which is a function of the system's Hamiltonian matrix,

$$\rho = f(H)$$

The Hamiltonian matrix is typically dense and structured (plane waves) or sparse (atomic orbitals, finite differencing). Here we focus entirely on the sparse case.

The traditional technique for finding the density matrix is to express the function in its spectral representation,

$$f(H) = \sum_i f(E_i)\psi_i \psi_i^t$$

Recently, there has been a lot of work on linear scaling methods*, some of which evaluate the matrix function directly in a polynomial form,

$$f(H) = \sum_i f_i H^i$$

but the polynomial degree needed for good accuracy is dependent on the function and the spectrum of the Hamiltonian and is in general prohibitively large. A more robust function representation involves approximating the contour integral definition of the function directly with a sum of matrix inverses,

$$f(H) = \frac{1}{2\pi i}\oint_\Gamma f(z)(zI - H)^{-1} dz \approx \sum_i w_i (z_i I - H)^{-1}$$

The number of terms needed for a given accuracy in this expansion is completely insensitive to the spectrum of the Hamiltonian. We only need diagonal and near diagonal terms of the density matrix, so <u>these material simulations are completely reduced to finding O(*N*) terms of the inverse of *N*x*N* sparse matrices</u>, as opposed to the usual eigenvalue formulation.

*S. Goedecker, *Linear Scaling Electronic Structure Methods*, Rev. Mod. Phys. **71**, 1085 (1999)

The motivation for working on this problem comes from numerous successes in related problems:

- In one dimension, local differential equations can be approximated by banded matrices with a bandwidth independent of problem size. An NxN banded matrix of bandwidth q can be exactly inverted in $O(qN)$ space and $O(q^2N)$ computation[1]. We can understand this easily in terms of the continuum problem – a qth degree differential equation has q homogeneous solutions, which we can patch together over the delta function that is used to define the inverse of the differential operator. Higher dimensions are problematic because there is a large space of homogeneous solutions. In a matrix representation, this translates to the bandwidth of the matrix being proportional to the system size. Any multidimension extension would require some compact approximate representation of all the homogeneous solutions.

- In the context of multigrid methods, there has been some success in quickly and compactly representing the inverse of elliptic differential operators by successively solving for details of the inverse on different resolutions/scales[2].

- Sparse approximate inverses to general sparse matrices have been investigated[3], in both factored and unfactored forms. They are constructed as preconditioners and as such they emphasis sparsity over accuracy. No efficient, compact high accuracy results have yet been reported.

1. G. Meurant., *A Review on the Inverse of Symmetric Tridiagonal and Block Tridiagonal Matrices*, SIAM J. Matrix Anal. Appl. **13**, 707 (1992)
2. M. Rozantsev, *Multigrid Computation of the FermionicInteractions in QCD Simulations*, PhD thesis, Weizmann Institute of Science (2000)
3. M. Benzi and M. Tuma. *A Comparative Study of Sparse Approximate Inverse Preconditioners* , Appl. Numer. Math. **30**, 305 (1999)

The motivation for working on this problem comes from numerous successes in related problems:

- Reduction techniques can successively remove unknowns from a system, by transforming to a similar but less sparse system of half the size, but they fail to be compact in the presence of oscillations[1].

- "Superfast" nested dissection can compactly factor a discretized differential operator by exploiting low-rank behavior of long range details (off-diagonal blocks of the factorization)[2]. Again, as yet no clear success in the presence of oscillations[3].

- Fast multipole methods[4] can compactly represent an analytically known inverse kernel by partitioning space and collectively representing far away groups of source points, carefully removing unnecessary detail by truncating appropriate function expansions. Success for the Helmholtz integral kernel demonstrates that oscillations aren't necessarily an Achilles heel of fast, compact solvers.

- Hierarchical matrices[5] are a framework for applying fast multipole-like principles to problems represented with matrices. Specifically, they demonstrate that inverses of elliptic operators have locally low rank, which leads to efficient compact storage. These methods have not yet been extended to non-elliptic operators/oscillatory integral kernels.

1. J. Schroder, U. Trottenberg, H. Reutersberg, *Reuktionsverfahren fuer differenzengleichungen bei randwertaufgaben II*, Numer. Math., **26** 429 (1976)
2. M. Gu, *Superfast Nested Dissection*, Bay Area Scientific Computing Day 2003.
3. M. Gu, private communication, November 2003.
4. L. Greengard and V. Rokhlin, *A Fast Algorithm for Particle Simulations*, J. Comp. Phys **73**, 325 (1987)
5. S. Börm, L. Grasedyck and W. Hackbusch, *An introduction to hierarchical matrices,* Math. Bohemica **127**, 229 (2002)

We start our analysis as generally as possible by saying that we have a multidimensional differential operator, $\hat{A}$, and it has an inverse, $A^{-1}(\vec{r},\vec{r}\,')$, that is continuous in both its variables and satisfies the equation

$$\hat{A}\,A^{-1}(\vec{r},\vec{r}\,')=\delta(\vec{r}-\vec{r}\,')$$

with the operator acting only on $\vec{r}$. For a fixed source point, $\vec{r}\,'$, this is just a single differential equation with a single source function.

Let us now imagine that we have an optimal solver and the only problem is compact storage. We have a discrete representation of the space at N points, and so the solution given a single source point is contained in N terms. If we want the diagonals of the inverse (ignoring that this diverges in the continuous problem), even with an optimal solver, we end up having to solve from N sources, inevitably doing $O(N^2)$ work intermediately. The trick to making use of an optimal solver is to solve an equation whose solution is known to decay rapidly, so that we can solve on a restricted domain whose size is independent of N. For some problems, the inverse decays exponentially away from the diagonal, but the more generic behavior is a power law decay, which isn't rapid enough.

Without any knowledge of the specific problem, all we can do to start is choose a source point, $\vec{r}_0$, and solve for the inverse from this point. In the next step, we can begin to attempt a more local calculation by examining the inverse at a point infinitesimally close to the first source point and expanding it as a Taylor series around the first point (keeping in mind its continuity),

$$A^{-1}(\vec{r},\vec{r}_0+d\vec{r})-A^{-1}(\vec{r},\vec{r}_0)=d\vec{r}\cdot\vec{\nabla}A^{-1}(\vec{r},\vec{r}_0)+O(|d\vec{r}|^2)$$

Given some restricted domain containing $\vec{r}_0$, we can find a $d\vec{r}$ small enough that the difference is negligible outside the restricted domain. This is the only useful piece of information we can get from the equation, since the Taylor series will diverge near the singularities at the two source points. We turn to our optimal solver to fill in the details, by solving $\hat{A} f_1(\vec{r}) = \delta(\vec{r} - \vec{r}_0 - d\vec{r}) - \delta(\vec{r} - \vec{r}_0)$ on the previously mentioned restricted domain. We now have the inverse from this second source point to some accuracy determined by the size of the restricted domain and infinitesimal step,

$$A^{-1}(\vec{r}, \vec{r}_0 + d\vec{r}) = A^{-1}(\vec{r}, \vec{r}_0) + f_1(\vec{r}) + O(\epsilon)$$

Now that we have the inverse from two source points, we can look for the inverse from a third infinitesimally close source point colinear with the previous two and this time use two Taylor series and cleverly combine equations,

$$A^{-1}(\vec{r}, \vec{r}_0 + d\vec{r}\,') - \frac{|d\vec{r}\,' - d\vec{r}| A^{-1}(\vec{r}, \vec{r}_0) + |d\vec{r}\,'| A^{-1}(\vec{r}, \vec{r}_0 + d\vec{r})}{|d\vec{r}\,' - d\vec{r}| + |d\vec{r}\,'|} = O(|d\vec{r}\,'|^2)$$

which cancels the first order error term of our extrapolation. Now, with a similarly sized restricted domain, we can choose a larger displacement and still have the difference be small on the boundary. We can then use the optimal solver again to find the difference within the domain. In general, with each step, we can cancel higher orders in a Talyor series and find better extrapolations and solve for increasingly farther source points in the form

$$A^{-1}(\vec{r}, \vec{r}\,') = c_0 A^{-1}(\vec{r}, \vec{r}_0) + \sum_{i=1} c_i f_i(\vec{r})$$

which serves to illustrate the notion of locally low rank: the farther away from $\vec{r}_0$, the fewer local detail patches f_i we'll intersect. However, we rely on our ability to extrapolate, which requires knowledge of a continuum underlying a discrete problem, and is held at the mercy of radii of convergence and analycity issues.

It would certainly be useful to exploit generic properties of the inverses of differential operators, since they are similar to generic properties of sparse matrices, such as exponential decay of offdiagaonals of band matrix inverses*, but we must avoid any reference to a continuum in the solution method itself.

For Laplace's equation, for example, we can physically motivate a solution method that possesses the same locally low rank as the previous extrapolation concept without directly exploiting continuity. We again imagine that we have discretized our domain to a size N and that we have access to an optimal solver, and we will enumerate all the points of the domain as $\{\vec{r}_i\}$, order to be determined.

We again begin by solving for the inverse from the first source point, but this time, we'll attempt to approximate the whole inverse with this one solution,

$$A^{-1}(\vec{r},\vec{r}\,')\approx\frac{A^{-1}(\vec{r},\vec{r}_1)A^{-1}(\vec{r}_1,\vec{r}\,')}{A^{-1}(\vec{r}_1,\vec{r}_1)}$$

ignoring the possible singularity of the inverse on the diagonal and making use of the symmetry of the inverse. Now, this is a terrible approximation of the inverse unless either variable is at the initial source point, in which case it is exact. We again continue by using the inverse from one point to modify the calculation of the next point, this time making use of the one point both functions share,

$$\tilde{A}^{-1}(\vec{r},\vec{r}_2)\equiv A^{-1}(\vec{r},\vec{r}_2)-A^{-1}(\vec{r},\vec{r}_1)\frac{A^{-1}(\vec{r}_1,\vec{r}_2)}{A^{-1}(\vec{r}_1,\vec{r}_1)}\rightarrow\tilde{A}^{-1}(\vec{r}_1,\vec{r}_2)=0$$

defining what we'll call a screened inverse function, whose utility isn't immediately obvious. This screened inverse satisfies the differential equation with an added interior boundary condition,

$$\hat{A}\,\tilde{A}(\vec{r},\vec{r}_2)=\delta(\vec{r}-\vec{r}_2)\ \ for\ \ r\neq r_1\ \ else\ \ \tilde{A}(\vec{r}_1,\vec{r}_2)=0$$

*S. Demko, W. Moss, P. Smith, Decay Rates for Inverses of Band Matrices, Math. Comp. **43**, 491 (1984)

We can then update our approximate inverse to be correct also at this second point,

$$A^{-1}(\vec{r},\vec{r}\,')\approx\frac{A^{-1}(\vec{r},\vec{r}_1)A^{-1}(\vec{r}_{1,}\vec{r}\,')}{A^{-1}(\vec{r}_{1,}\vec{r}_1)}+\frac{\tilde{A}^{-1}(\vec{r},\vec{r}_2)\tilde{A}^{-1}(\vec{r}_{2,}\vec{r}\,')}{\tilde{A}^{-1}(\vec{r}_{2,}\vec{r}_2)}$$

To continue this process, we define a more general screened inverse function,

$$\tilde{A}^{-1}(\vec{r},\vec{r}_i)\equiv A^{-1}(\vec{r},\vec{r}_i)-\sum_{j=1}^{i-1}\tilde{A}^{-1}(\vec{r},\vec{r}_j)\frac{\tilde{A}^{-1}(\vec{r}_j,\vec{r}_i)}{\tilde{A}^{-1}(\vec{r}_j,\vec{r}_j)}$$

which again satisfies the differential equation with an increasing number of interior boundary conditions,

$$\hat{A}\,\tilde{A}(\vec{r},\vec{r}_i)=\delta(\vec{r}-\vec{r}_i)\ \ with\ \ \tilde{A}(\vec{r}_j,\vec{r}_i)=0\,\forall\,1\leq j<i$$

With all the screened terms taken together, we can exactly write the inverse as

$$A^{-1}(\vec{r},\vec{r}\,')=\sum_{i=1}^{N}\frac{\tilde{A}^{-1}(\vec{r},\vec{r}_i)\tilde{A}^{-1}(\vec{r}_i,\vec{r}\,')}{\tilde{A}^{-1}(\vec{r}_i,\vec{r}_i)}$$

which is in a form very analogous to a Cholesky factorization of the inverse. This connection will be exploited to solve these equations.

Before we try to solve anything, we must examine the behavior of the screened inverse functions to see how they can help us. This sort of equation might arise in a physical system where you have an electric charge at the source point and grounded metal antennas at all the screened points. We know that in such a system, the potential generated by the charge will exponentially decay with a characteristic distance inversely proportional to the density of screening points (to the Dth root in D dimensions),

$$\tilde{A}^{-1}(\vec{r},\vec{r}_i)\propto\exp\left(-\left(\frac{i}{N}\right)^{1/D}|\vec{r}-\vec{r}_i|\right)$$

for a uniform/random/space filling distribution of screening points.

If we choose a fixed precision, ε, then we can restrict the screened inverse functions to a small active domain,

$$\varepsilon \propto \exp\left(-\left(\frac{i}{N}\right)^{1/D} R_i\right) \rightarrow R_i \propto \left(\frac{N}{i}\right)^{1/D} \ln(1/\varepsilon)$$

which leads to a representation of the total inverse that is within a log of optimal,

$$V_i = R_i^D \propto \left(\frac{N}{i}\right) \ln(1/\varepsilon)^D \rightarrow N_{total} = \sum_i V_i \propto N \ln(N) \ln(1/\varepsilon)^D$$

In terms of numerical linear algebra, we expect the Cholesky factorization of the inverse of a sparse matrix to be sparse to a finite numerical precision for certain orderings (such as a random ordering), at the very least for the case of a discretized Laplace equation.

We can now relax our condition of having an optimal solver because we can just solve the screened inverse functions in reverse, starting from the most screened, and use those answers to calculate the inverses of the successively less screened problems. In terms of matrices, this can be performed as a succession of rank one updates using the Sherman-Morrison formula or more simply by orthogonalizing a set of unit vectors with respect to the *A* matrix such as in the AINV algorithm*.

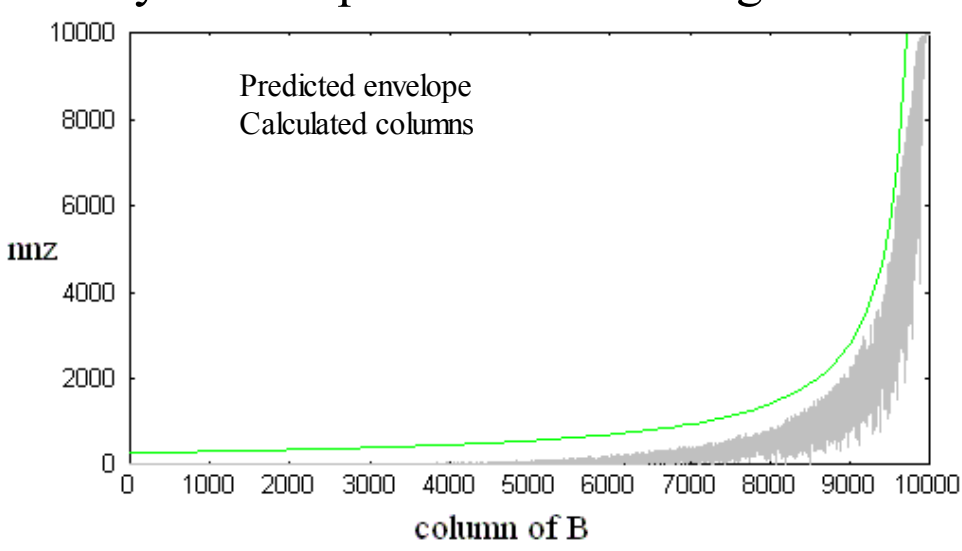
At this point, we'll stop with continuum arguments and mostly focus on the problem in terms of numerical linear algebra. *A* is no longer a differential operator, but rather a matrix approximating the operator using a finite difference representation.

* M. Benzi and M. Tuma. *A Comparative Study of Sparse Approximate Inverse Preconditioners* , Appl. Numer. Math. **30**, 305 (1999)

Our goal now is to find a matrix factorization of the form $B^T A B = D$ which will allow us to represent the inverse of A in a factored form, $A^{-1} = B D^{-1} B^T$. If we consider the columns of B as separate vectors, b_i , then we can think of this process as orthogonalizing a set of vectors with respect to A. Given an initial set of N vectors, $\{x_i\}$, we can perform an incomplete classical Gram-Schmidt orthogonalization procedure

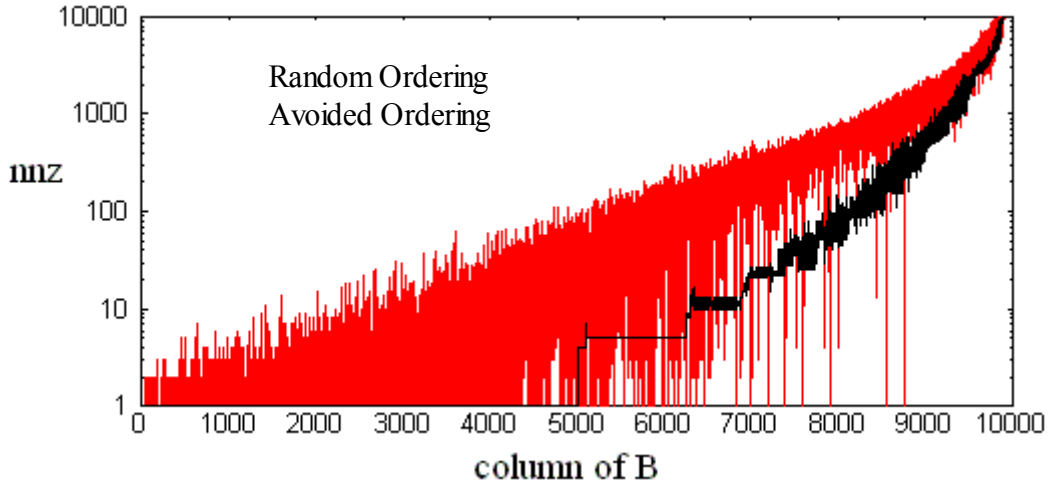
$$
\begin{aligned}
&for\ i = 1..N: \\
&\quad 1.\ for\ j < i \rightarrow c_{ij} = \frac{1}{d_j} b_j^T A x_i \\
&\quad 2.\ b_i = x_i - \sum_{j<i} c_{ij} b_j \\
&\quad 3.\ d_i = b_i^T A b_i \\
&\quad 4.\ truncate\ terms\ of\ b_i\ smaller\ than\ \varepsilon / \sqrt{|d_i|}
\end{aligned}
$$

We must be wary of the diagonal terms becoming too small, which would signify a bad choice of initial vectors, and also of the instabilities of the classical Gram-Schmidt procedure if ε is set close to the machine precision. To perform this process efficiently, we make use of a sparse vector data structure. For the Laplace equation on a 2D unit square represented by 100x100 5-point finite differencing grid and using randomly ordered basis vectors, we clearly see the predicted screening behavior:



$N = 10000$
$nnz(A) = 49600$
$nnz(B) = 5821598$
$fillin = 117$
$\varepsilon = 10^{-5}$
$\|ABB^T - I\|_2 = 1.2\,x\,10^{-3}$

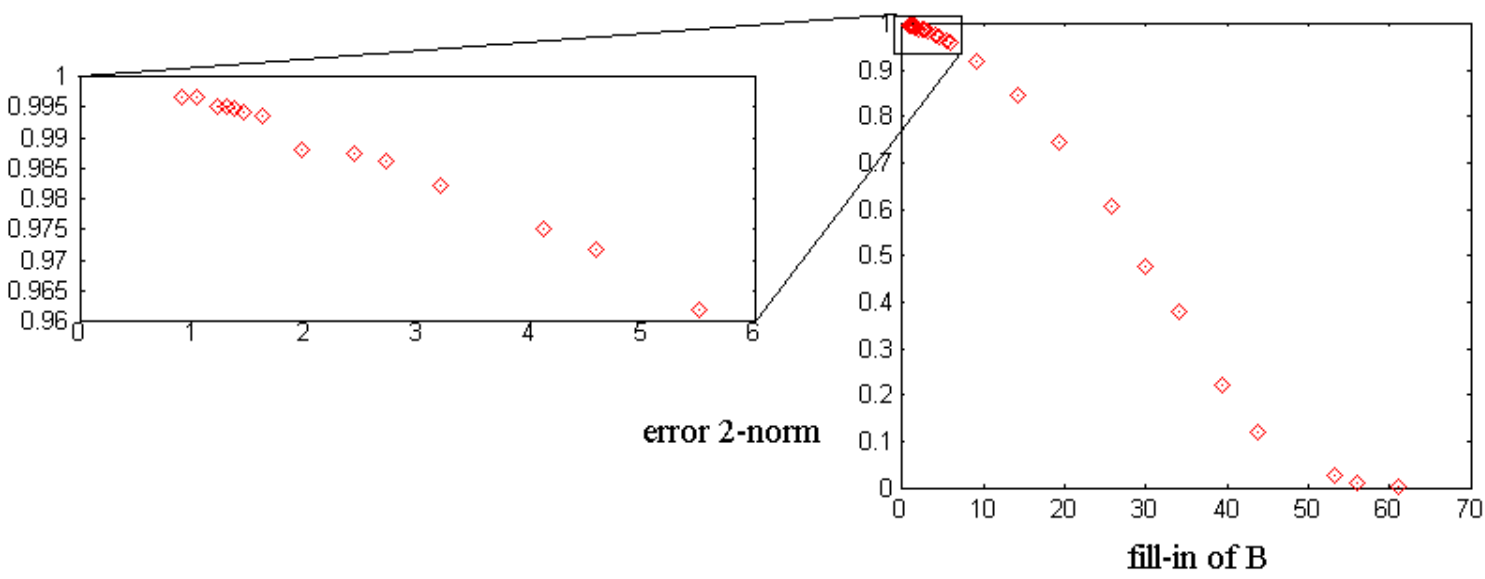
Random ordering is useful because it avoids intersections with previously orthogonalized vectors on average. This is only true when applied to a matrix with roughly the same number of nonzero elements per column. A more general ordering is to just choose a new vector at each step of the orthogonalization that intersects as few previous vectors as possible. This avoided ordering exibits the same behavior as the random ordering, with some improvement:



$N = 10000$
$\varepsilon = 10^{-5}$
Random :
$nnz(B) = 5774149$
$\|ABB^T - I\|_2 = 1.2\,x\,10^{-3}$
Avoided :
$nnz(B) = 3491661$
$\|ABB^T - I\|_2 = 1.8\,x\,10^{-4}$

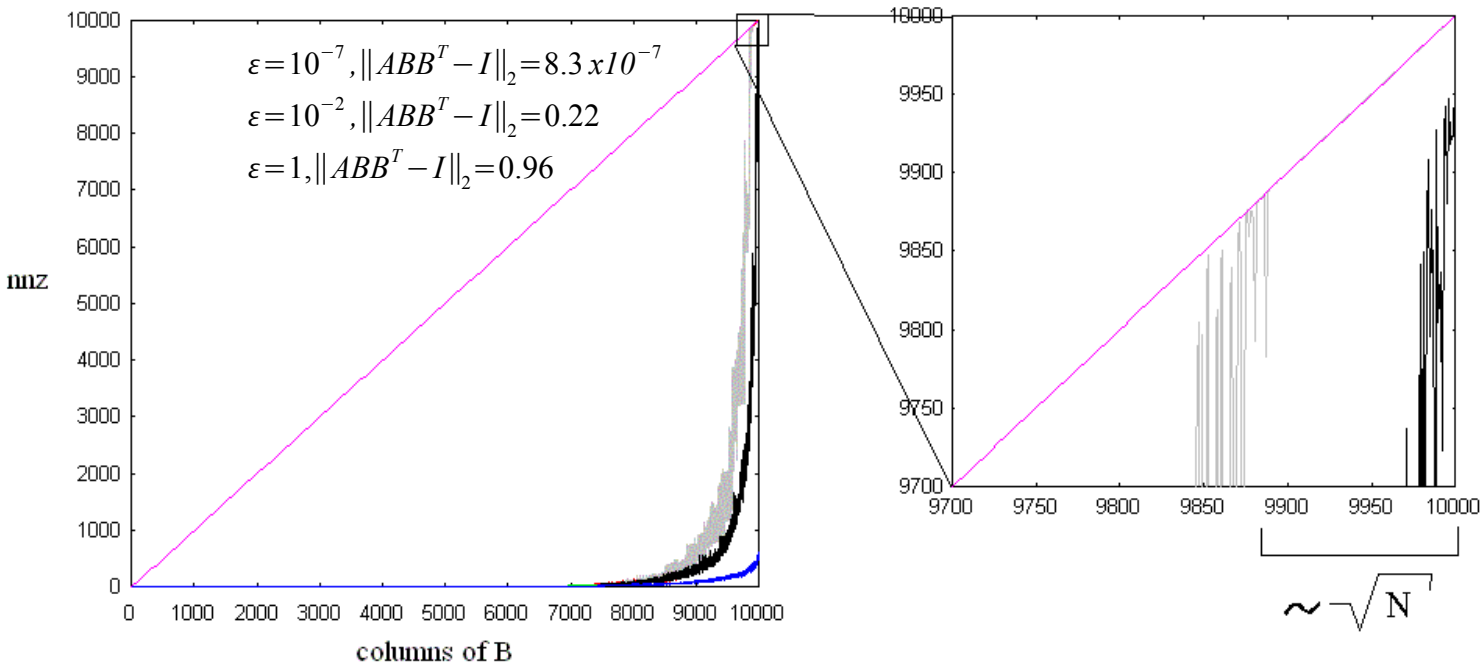
We will now just use the avoided ordering, since it performs better.

Since previous work on factored sparse approximate inverses has focused on very low fill and/or very low drop tolerance, it is useful to study the 2-norm of the error in this limit,



The typical case of forcing B to be approximately as sparse as A leads to a very poor inverse, which can be substantially improved in this example with a factor of 10 increase in the fill.

This simple, physically motivated method unfortunately has a simple, physically motivated difficulty: for a given domain and tolerance, as we successively solve for columns of B and "undo the screening", it can very well happen that the localized active region can grow beyond the extent of our domain. For the rest of the calculation, the vectors of B will be dense and if the number of these vectors has some polynomial dependence on the size of the system, then this method is no longer optimal. This now sets a limit on how accurate we are allowed to be within this framework. We can clearly see this transition in a plot of column sparsities,



The maximally accurate but still "screened" choice above isn't accurate enough to serve directly as an inverse, but it is accurate enough to make for a good preconditioner – a lesser but still very desirable quality.

This method as currently formulated isn't numerically competative with any existing methods or representations of inverses, such as Hierarchical matrices or Total Reduction, in terms of either speed or accuracy. However, it suggests paths for improvement that might eventually be competative:

- The B matrix we are solving for doesn't need to be a triangular matrix, since we are avoiding matrix solves. Related to this, we can start the inversion process from an arbitrary set of vectors. If we choose anything other that the basis vectors, we will inevitably have more non-zero overlaps between the vectors we are orthogonalizing, but we might reduce the overall weight of these couplings, which can lead to a stronger decay of terms and more opportunity for truncations at later stages of the calculation.

- The B matrix starts out very sparse and only the later columns fill in very densely. To avoid the calculation of the dense columns, we can stop the orthogonalization process midway and transform the intermediate matrix,

$$\begin{bmatrix} I & 0 \\ -C\tilde{D^{-1}} & I \end{bmatrix} \begin{bmatrix} \tilde{D} & C^T \\ C & \tilde{A} \end{bmatrix} \begin{bmatrix} I & -\tilde{D^{-1}}C^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} \tilde{D} & 0 \\ 0 & \tilde{A} - C\tilde{D^{-1}}C^T \end{bmatrix}$$

  which allows us to restart the inversion process on a matrix of smaller size but increased density.

<u>WORK IN PROGRESS...</u>