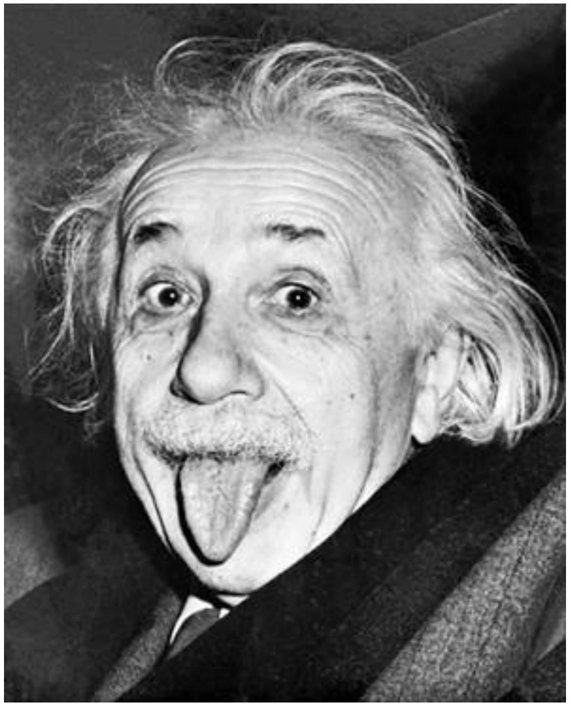
Sparse Solutions of Linear Systems of Equations and Sparse Modeling of Signals and Images

Alfredo Nava-Tudela

John J. Benedetto, advisor



Happy birthday Lucía!

Outline

- Problem: Find “sparse solutions” of **Ax = b**.
- Definitions of “sparse solution”.
- How do we find sparse solutions?

  The Orthogonal Matching Pursuit (OMP)
- Some theoretical results.
- Implementation and validation, some details.
- Validation results.
- Conclusions/Recapitulation.
- Project timeline, current status.
- References.

Problem

Let **A** be an *n* by *m* matrix, with $n < m$, and rank(**A**) = *n.* We want to solve

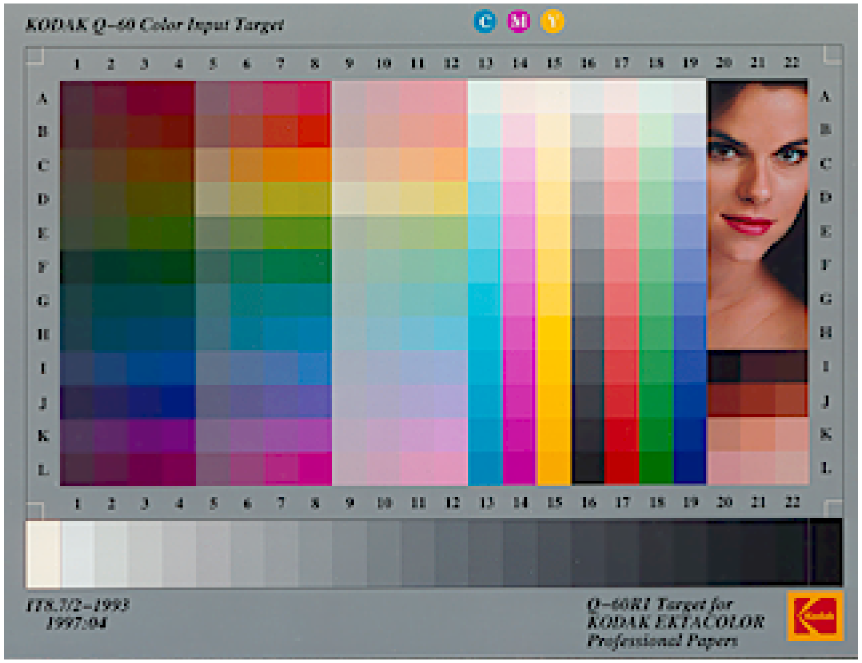
$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where **b** is a data or signal vector, and **x** is the solution with the fewest number of non-zero entries possible, that is, the "sparsest" one.

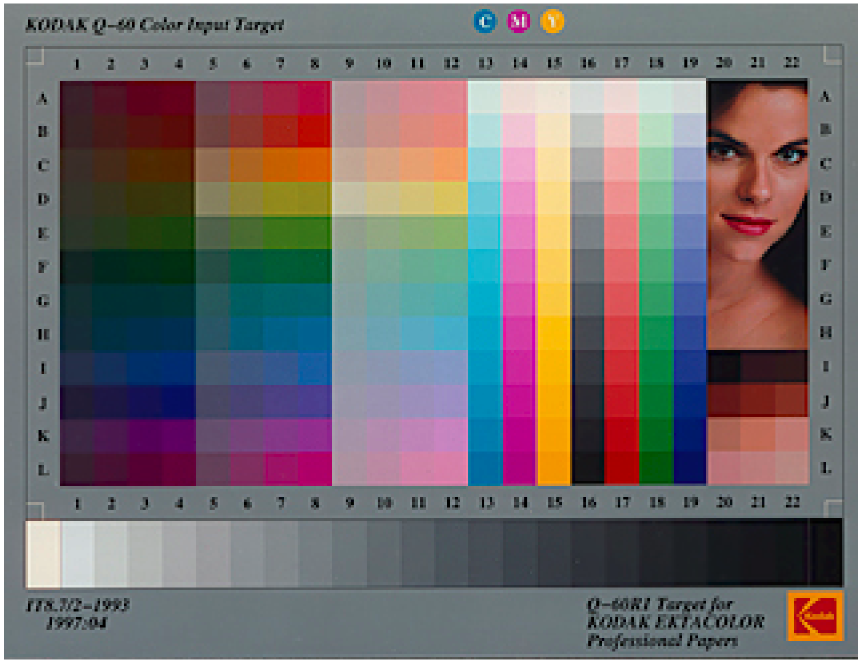
<u>Observations</u>:

- **A** is underdetermined and, since rank(**A**) = *n*, there is an infinite number of solutions. Good!
- How do we find the "sparsest" solution? What does this mean in practice? Is there a unique sparsest solution?

Why is this problem relevant?



231 kb, uncompressed,
320x240x3x8 bit



74 kb, compressed 3.24:1
JPEG

Why is this problem relevant?

512 x 512 Pixels,
24-Bit RGB,
Size 786 Kbyte

75:1, 10.6 Kbyte
JPEG2000

Why is this problem relevant?

“Sparsity” equals compression:

Assume **Ax** = **b**. If **x** is sparse, and **b** is dense, store **x**!

Image compression techniques, such as JPEG [6] or JPEG-2000 [5], are based in this idea, where a linear transformation provides a sparse representation within an error margin of the original image.

Definitions of "sparse"

- Convenient to introduce the l_0 "norm" [1]:

$$||\mathbf{x}||_0 = \# \{k : x_k \neq 0\}$$

- (P_0): $\min_{\mathbf{x}} ||\mathbf{x}||_0$ subject to $||\mathbf{Ax} - \mathbf{b}||_2 = 0$

- (P_0^{ε}): $\min_{\mathbf{x}} ||\mathbf{x}||_0$ subject to $||\mathbf{Ax} - \mathbf{b}||_2 < \varepsilon$

<u>Observations</u>: In practice, (P_0^{ε}) is the working definition of sparsity as it is the only one that is computationally practical. Solving (P_0^{ε}) is NP-hard [2].

Some theoretical results

Definition: The *spark* of a matrix **A** is the minimum number of linearly dependent columns of **A**. We write spark(**A**) to represent this number.

Theorem: If there is a solution **x** to **Ax** = **b**, and $||\mathbf{x}||_0 <$ spark(**A**) / 2, then **x** is the sparsest solution. That is, if $\mathbf{y} \neq \mathbf{x}$ also solves the equation, then $||\mathbf{x}||_0 < ||\mathbf{y}||_0$.

Observation: Computing spark(**A**) is combinatorial, therefore hard. Alternative?

Some theoretical results

Definition: The *mutual coherence* of a matrix **A** is the number

$$\mu(\mathbf{A}) = \max_{1 \le k,j \le m,\ k \ne j} \frac{|\mathbf{a}_k^T \mathbf{a}_j|}{||\mathbf{a}_k||_2 \cdot ||\mathbf{a}_j||_2}.$$

Lemma: spark(**A**) ≥ 1+1/mu(**A**).

Theorem: If **x** solves **Ax** = **b**, and $||\mathbf{x}||_0 < (1+\mu(\mathbf{A})^{-1})/2$, then **x** is the sparsest solution. That is, if **y** ≠ **x** also solves the equation, then $||\mathbf{x}||_0 < ||\mathbf{y}||_0$.

<u>Observation</u>: mu(**A**) is a lot easier and faster to compute, but 1+1/mu**(A)** far worse bound than spark**(A),** in general.

Finding sparse solutions: OMP

Orthogonal Matching Pursuit algorithm [1]:

Task: Approximate the solution of $(P_0) : \min_{\mathbf{x}} ||\mathbf{x}||_0$ subject to $\mathbf{Ax} = \mathbf{b}$.

Parameters: We are given the matrix $\mathbf{A}$, the vector $\mathbf{b}$, and the threshold ϵ_0.

Initialization: Initialize $k = 0$, and set

- The initial solution $\mathbf{x}^0 = \mathbf{0}$.
- The initial residual $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0 = \mathbf{b}$.
- The initial solution support $\mathcal{S}^0 = Support\{\mathbf{x}^0\} = \emptyset$.

Main Iteration: Increment k by 1 and perform the following steps:

- **Sweep:** Compute the errors $\epsilon(j) = \min_{z_j} ||z_j \mathbf{a}_j - \mathbf{r}^{k-1}||_2^2$ for all j using the optimal choice $z_j^* = \mathbf{a}_j^T \mathbf{r}^{k-1} / ||\mathbf{a}_j||_2^2$.
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall j \notin \mathcal{S}^{k-1}$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{Ax} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.
- **Update Residual:** Compute $\mathbf{r}^k = \mathbf{b} - \mathbf{Ax}^k$.
- **Stopping Rule:** If $||\mathbf{r}^k||_2 < \epsilon_0$, stop. Otherwise, apply another iteration.

Output: The proposed solution is $\mathbf{x}^k$ obtained after k iterations.

Some theoretical results

Definition: The *mutual coherence* of a matrix **A** is the number

$$\mu(\mathbf{A}) = \max_{1\le k,j\le m,\ k\neq j} \frac{|\mathbf{a}_k^T \mathbf{a}_j|}{||\mathbf{a}_k||_2 \cdot ||\mathbf{a}_j||_2}.$$

Theorem: If **x** solves **Ax** = **b**, and $||\mathbf{x}||_0 < (1+\mu(\mathbf{A})^{-1})/2$, then **x** is the sparsest solution. That is, if **y** ≠ **x** also solves the equation, then $||\mathbf{x}||_0 < ||\mathbf{y}||_0$.

Theorem: For a system of linear equations **Ax** = **b** (**A** an *n* by *m* matrix, *n* < *m*, and rank(**A**) = *n*), if a solution **x** exists obeying $||\mathbf{x}||_0 < (1+\mu(\mathbf{A})^{-1})/2$, then an OMP run with threshold parameter $\varepsilon_0 = 0$ is guaranteed to find **x** exactly.

Implementation and Validation

In light of these theoretical results, we can envision the following roadmap to validate an implementation of OMP.

- We have a simple theoretical criterion to guarantee both solution uniqueness and OMP convergence:

If $\mathbf{x}$ *is a solution to* $\mathbf{Ax} = \mathbf{b}$*, and* $||\mathbf{x}||_0 < (1+\mu(\mathbf{A})^{-1})/2$*,*
then $\mathbf{x}$ *is the unique sparsest solution to* $\mathbf{Ax} = \mathbf{b}$ *and OMP will find it.*

- Hence, given a full-rank n by m matrix $\mathbf{A}$ ($n < m$), compute $\mu(\mathbf{A})$, and find the largest integer k smaller than or equal to $(1+\mu(\mathbf{A})^{-1})/2$. That is, $k = \text{floor}((1+\mu(\mathbf{A})^{-1})/2)$.

Implementation and Validation

- Build a vector **x** with exactly *k* non-zero entries and produce a right hand side vector **b** = **Ax**. This way, you have a known sparsest solution **x** to which to compare the output of any OMP implementation.

- Pass **A**, **b**, and ε_0 to OMP to produce a solution vector $\mathbf{x}_{omp} = \mathrm{OMP}(\mathbf{A},\mathbf{b},\varepsilon_0)$.

- If OMP terminates after *k* iterations and $||\mathbf{A}\mathbf{x}_{omp} - \mathbf{b}|| < \varepsilon_0$, for all possible **x** and $\varepsilon_0 > 0$, then the OMP implementation would have been validated.

<u>Caveat</u>: The theoretical proofs assume infinite precision.

Implementation and Validation

- Some implementation details worth discussing:

The core of the algorithm is found in the following three steps. We will discuss in detail our implementation of the "Update Support" and "Update Provisional Solution" steps.

- **Sweep:** Compute the errors $\epsilon(j) = \min_{z_j} ||z_j \mathbf{a}_j - \mathbf{r}^{k-1}||_2^2$ for all j using the optimal choice $z_j^* = \mathbf{a}_j^T \mathbf{r}^{k-1} / ||\mathbf{a}_j||_2^2$.
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall j \notin \mathcal{S}^{k-1}$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.

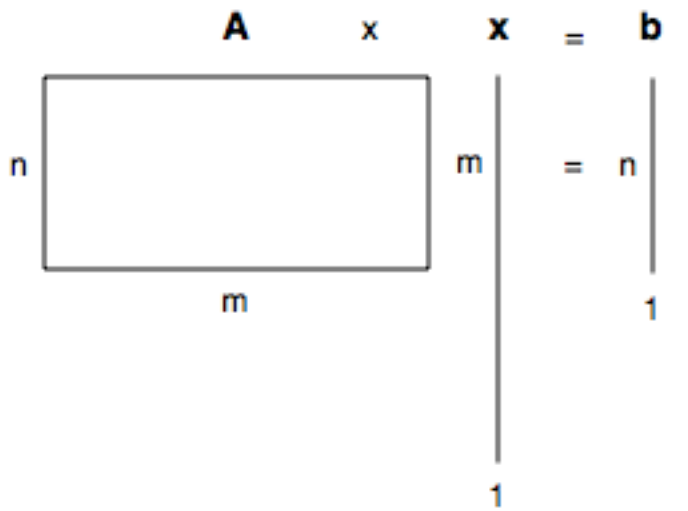
Implementation and Validation

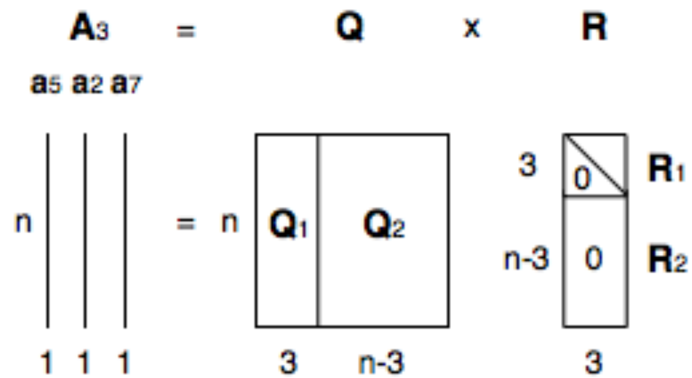
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall j \notin \mathcal{S}^{k-1}$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.

```
---Initialization---
k = 0;
activeCol = []; % will contain the indices of the active columns of A.
epsilon = zeros(m,1); % contains the errors epsilon(j) described above.
---Inside Main Loop---
k = k + 1;
% Sweep
for j = 1:m
        a_j = A(:,j);
        z_j = a_j'*r0/norm(a_j)^2;
        epsilon(j) = norm(z_j*a_j - r0)^2;
end
% Update Support
maxValueEpsilon = max(epsilon);
epsilon(activeCol) = maxValueEpsilon;
[minValueEpsilon, j_0] = min(epsilon); % j_0 is the new index to add.
activeCol(k) = j_0; % update the set of active columns of A.
```

Implementation and Validation

- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{Ax} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.





$\mathbf{A}_3 = \mathbf{QR} = \mathbf{Q}_1\mathbf{R}_1 + \mathbf{Q}_2\mathbf{R}_2 = \mathbf{Q}_1\mathbf{R}_1 + \mathbf{0} = \mathbf{Q}_1\mathbf{R}_1$

Solve the linear system $\mathbf{A}_3\mathbf{x}^* = \mathbf{b}$, with $\mathbf{x}^* \in R^3$. We have:

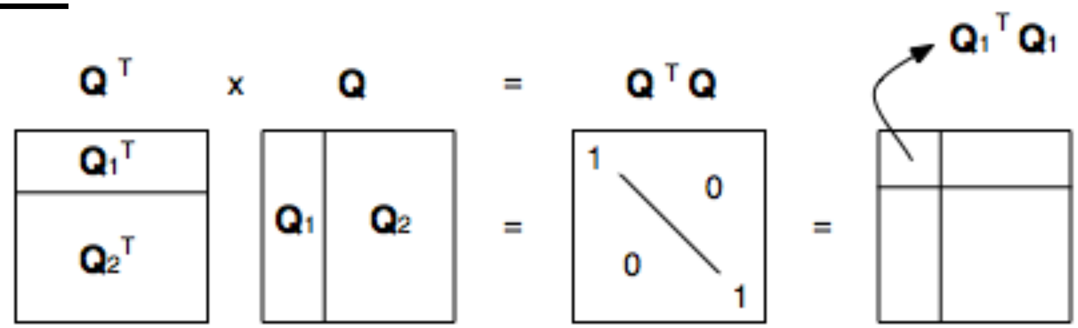
$$\mathbf{A}_3\mathbf{x}^* = \mathbf{QRx}^* = \mathbf{Q}_1\mathbf{R}_1\mathbf{x}^* = \mathbf{b} \Rightarrow \mathbf{Q}_1^T\mathbf{Q}_1\mathbf{R}_1\mathbf{x}^* = \mathbf{Q}_1^T\mathbf{b} \quad (1)$$

See [3] for more on the QR decomposition.

Implementation and Validation

- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.

Observation:



(1): $\mathbf{Q}_1{}^T\mathbf{Q}_1\mathbf{R}_1\mathbf{x}^* = \mathbf{Q}_1{}^T\mathbf{b} \Leftrightarrow \mathbf{R}_1\mathbf{x}^* = \mathbf{Q}_1{}^T\mathbf{b}$
$\Leftrightarrow \mathbf{x}^* = (\mathbf{R}_1)^{-1}\,\mathbf{Q}_1{}^T\mathbf{b},$

where we can obtain the last equation because **A** is a full rank matrix, and therefore $\mathbf{A}_3$ is too, implying $(\mathbf{R}_1)^{-1}$ exists.

Implementation and Validation

- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.

The minimizer $\mathbf{x}^{k=3}$ of $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2$, subject to support$\{\mathbf{x}\} = S^{k=3}$, is then obtained when we solve $\mathbf{A}_3\mathbf{x}^* = \mathbf{b}$, with $\mathbf{x}^* \in R^3$, and we set $\mathbf{x}^{k=3}$ equal to the "natural embedding" of $\mathbf{x}^*$ into the zero vector $\mathbf{0} \in R^m$.

```
---Initialization---
x0 = zeros(m,1);
---Inside Main Loop---
% Update the provisional solution by solving an equivalent unconstrained
% least squares problem.
A_k = A(:,activeCol);
[Q,R] = qr(A_k);
x0(activeCol) = R(1:k,:) \ Q(:,1:k)'*b;
```

Validation Results

We ran two experiments:

1) **A** $\in R^{100x200}$, with entries in N(0,1) i.i.d. for which μ(**A**) = 0.3713, corresponding to k = 1 ≤ K.
2) **A** $\in R^{200x400}$, with entries in N(0,1) i.i.d. for which μ(**A**) = 0.3064, corresponding to k = 2 ≤ K.

<u>Observations</u>:

- **A** will be full-rank with probability 1 [1].
- For full-rank matrices **A** of size n x m, the mutual coherence satisfies μ(**A**) ≥ √{(m - n)/(n·(m - 1))} [4]. That is, the upper bound of K = (1 + μ(**A**)$^{-1}$)/2 can be made as big as needed, provided n and m are big enough.

Validation Results

For each matrix **A**, we chose 100 vectors with *k* non-zero entries whose positions were chosen at random, and whose entries were in N(0,1).

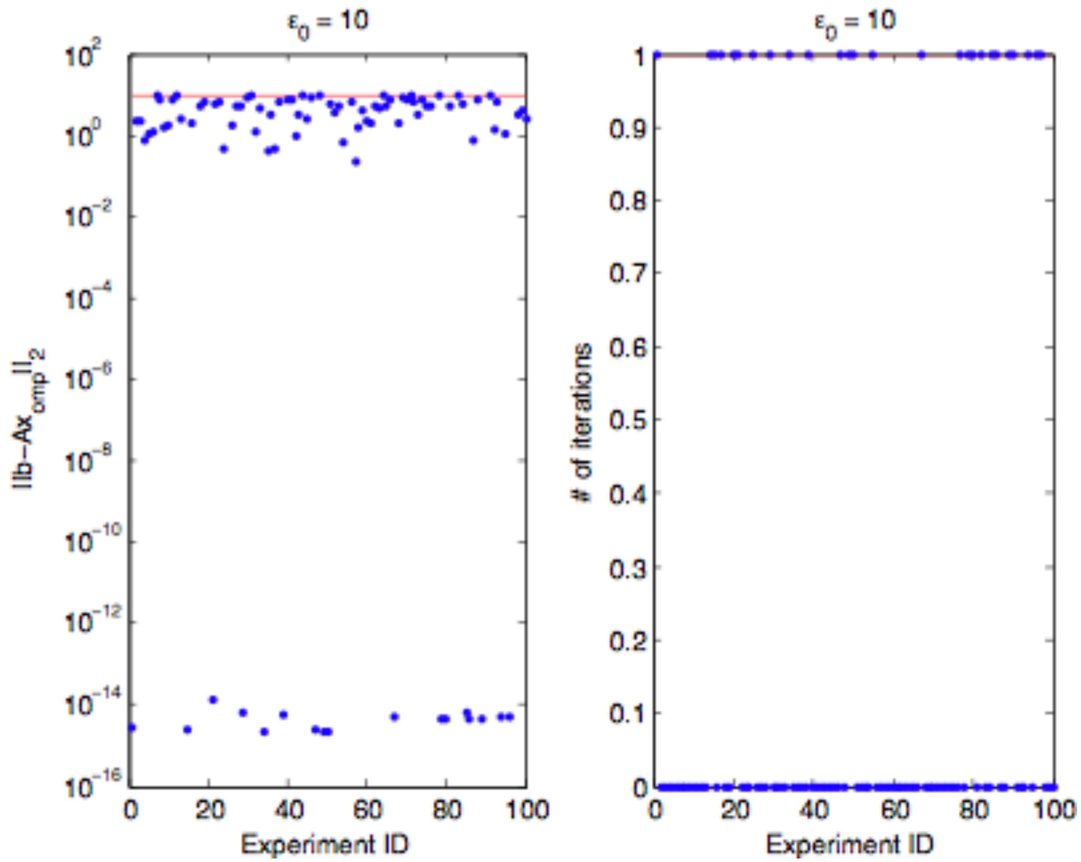
Then, for each such vector **x**, we built a corresponding right hand side vector **b** = **Ax**.

Each of these vectors would then be the unique sparsest solution to **Ax** = **b**, and OMP should be able to find them.

Finally, given $\varepsilon_0 > 0$, if our implementation of OMP were correct, it should stop after *k* steps (or less), and if $\mathbf{x}_{OMP} = OMP(\mathbf{A},\mathbf{b},\varepsilon_0)$, then $||\mathbf{b} - \mathbf{A}\mathbf{x}_{OMP}|| < \varepsilon_0$.

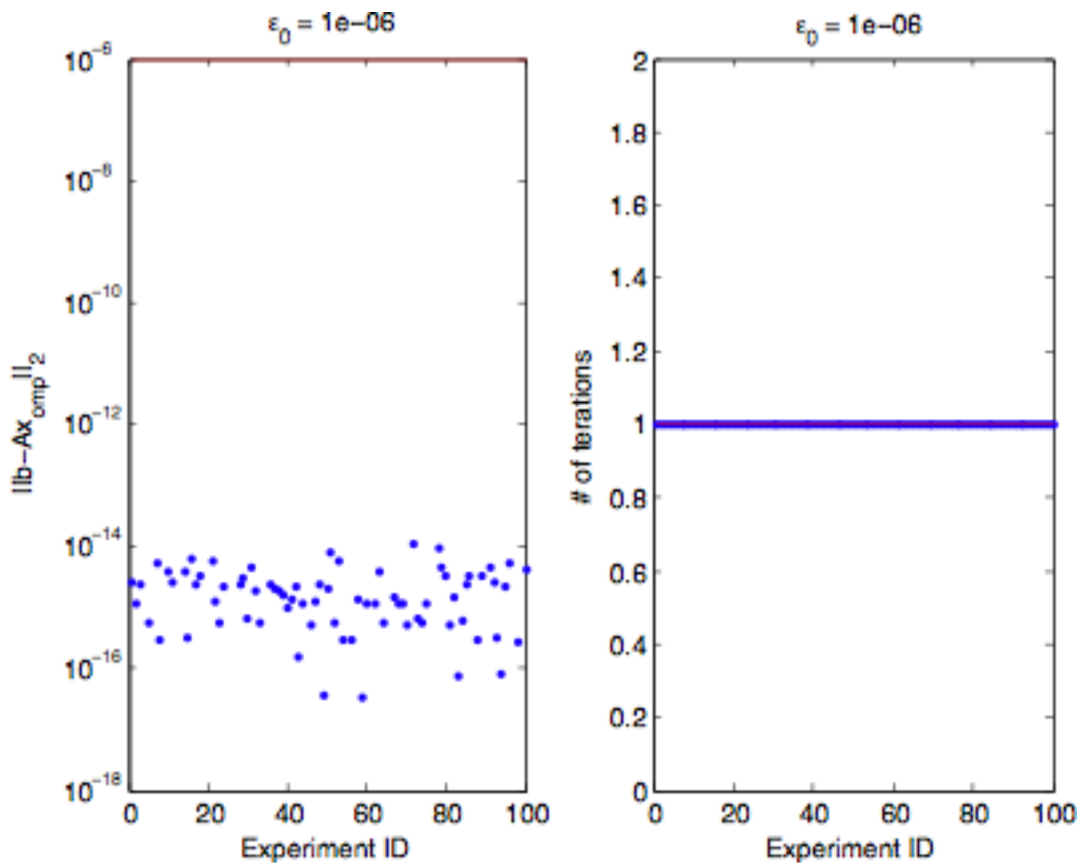
Validation Results

k = 1



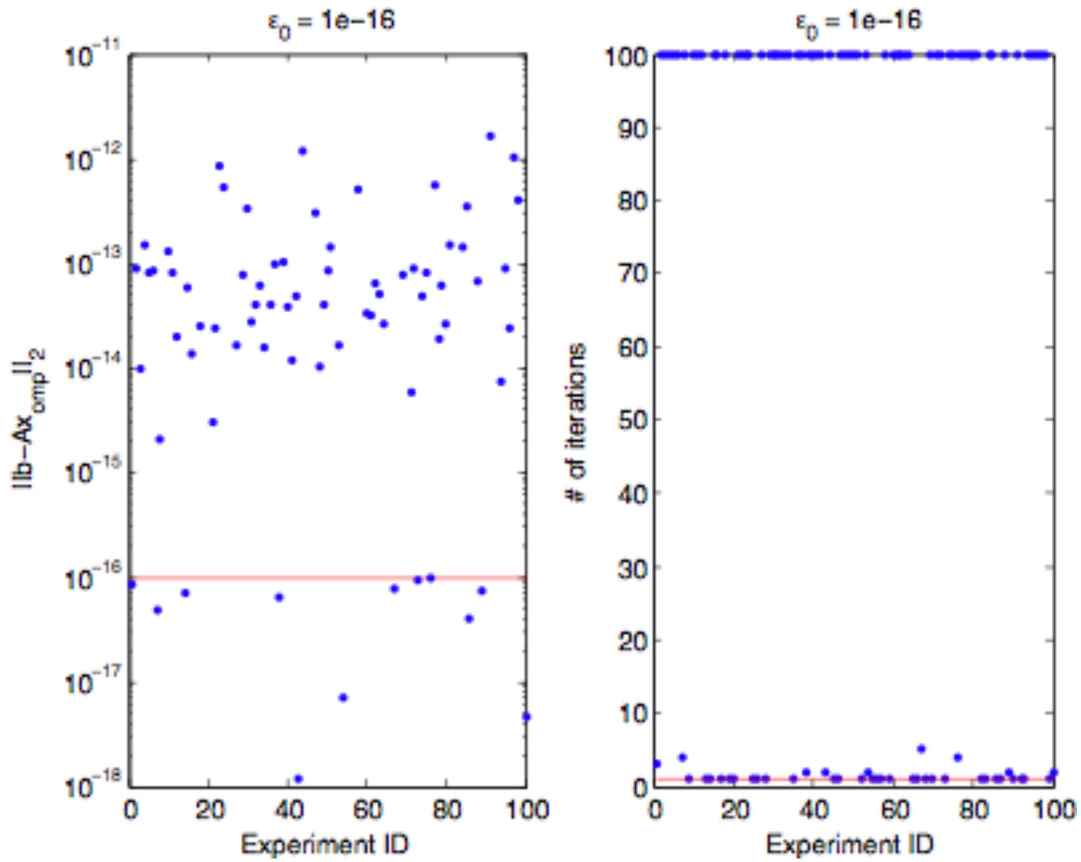
Validation Results

k = 1



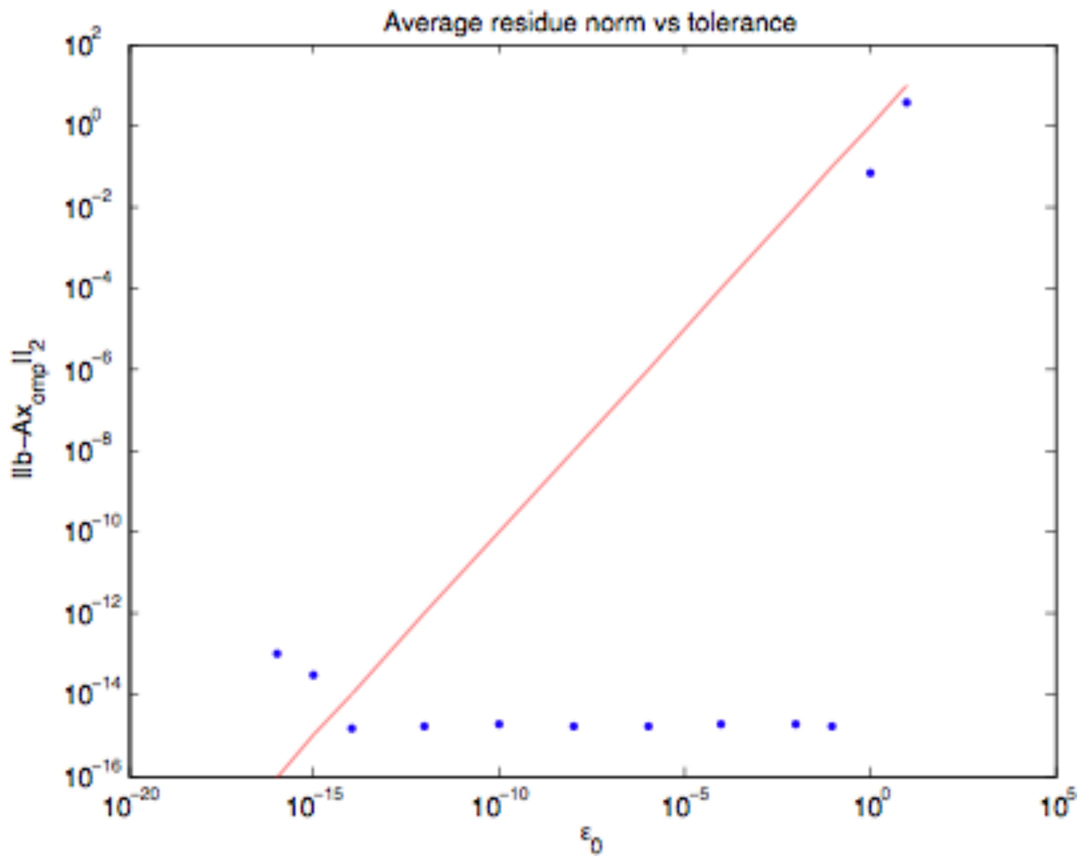
Validation Results

k = 1



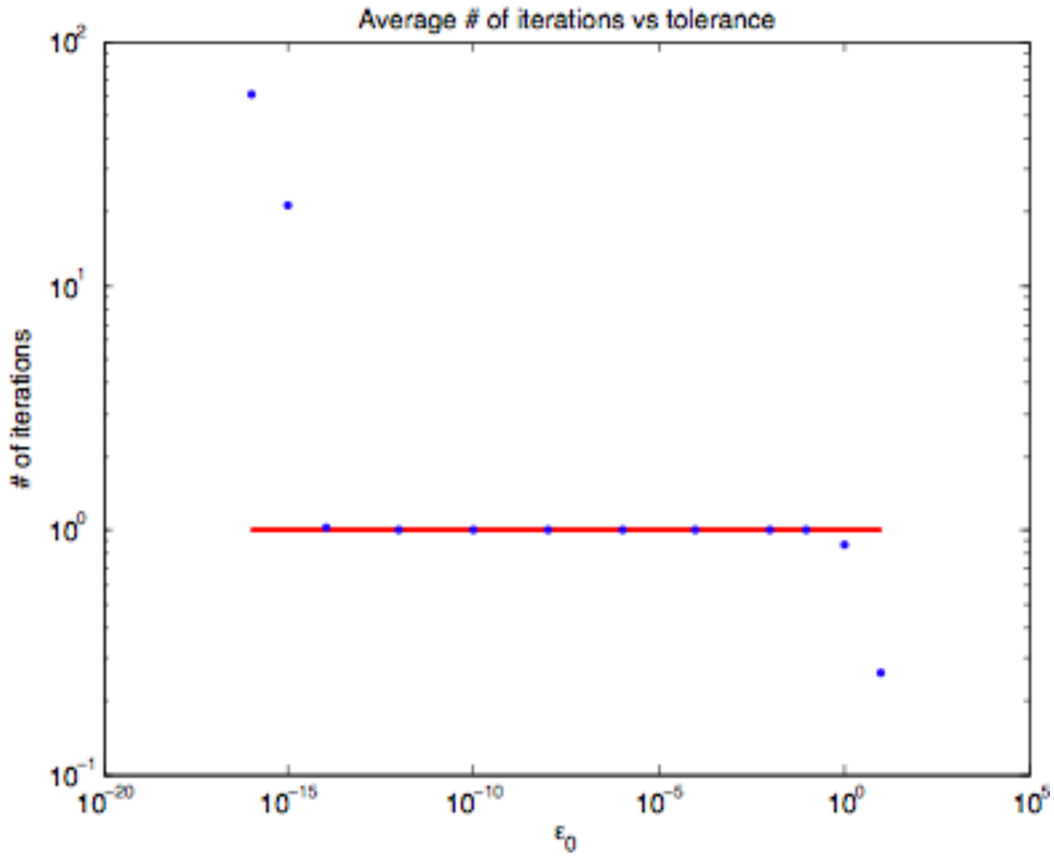
Validation Results

k = 1



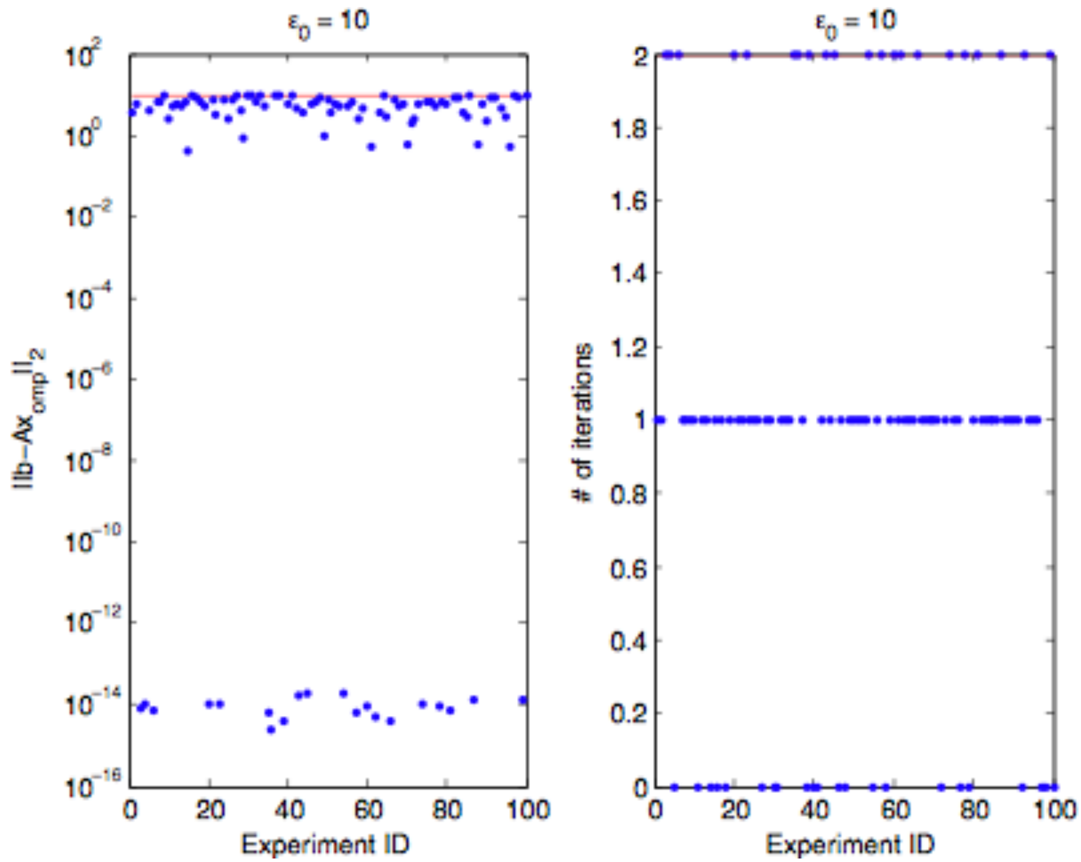
Validation Results

k = 1



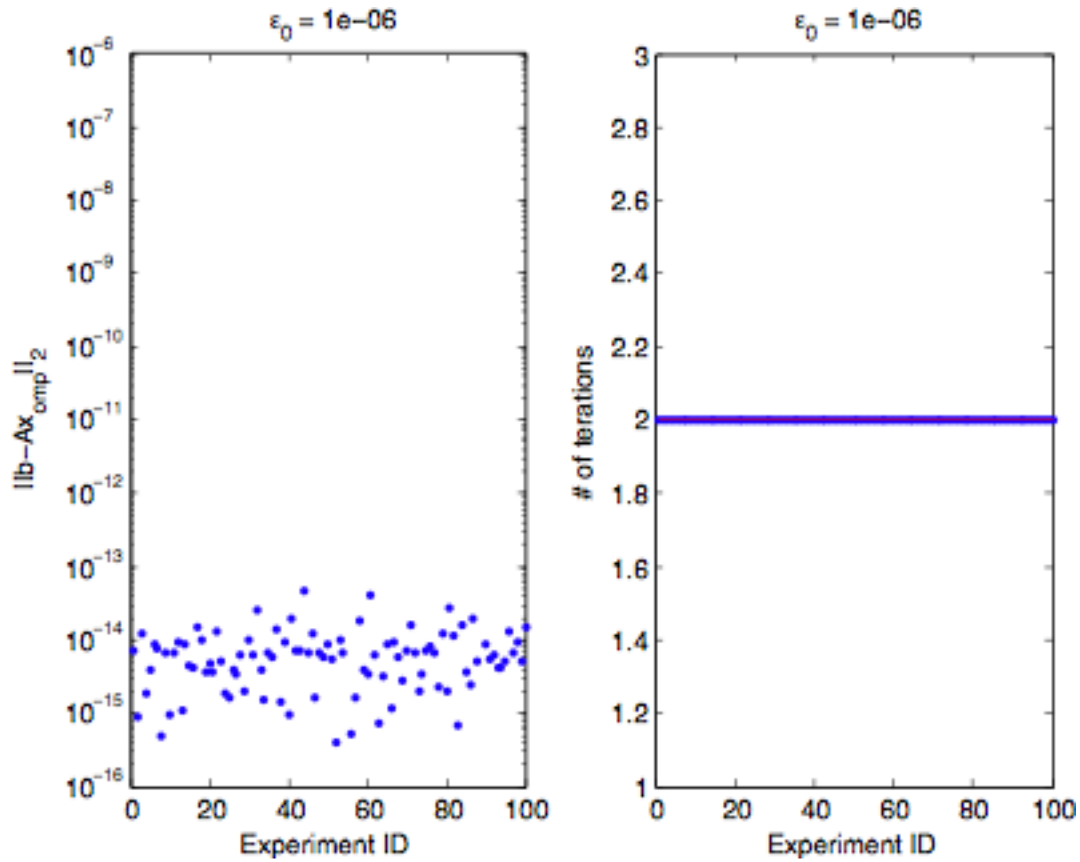
Validation Results

k = 2



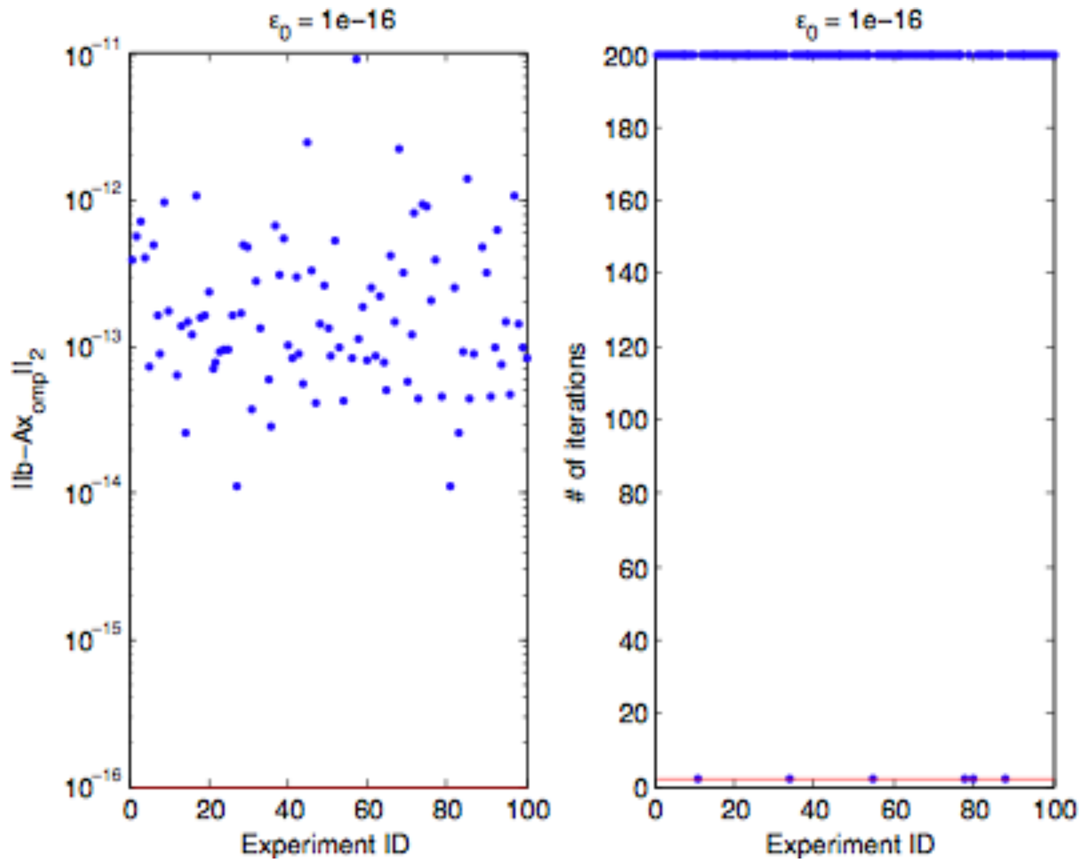
Validation Results

k = 2



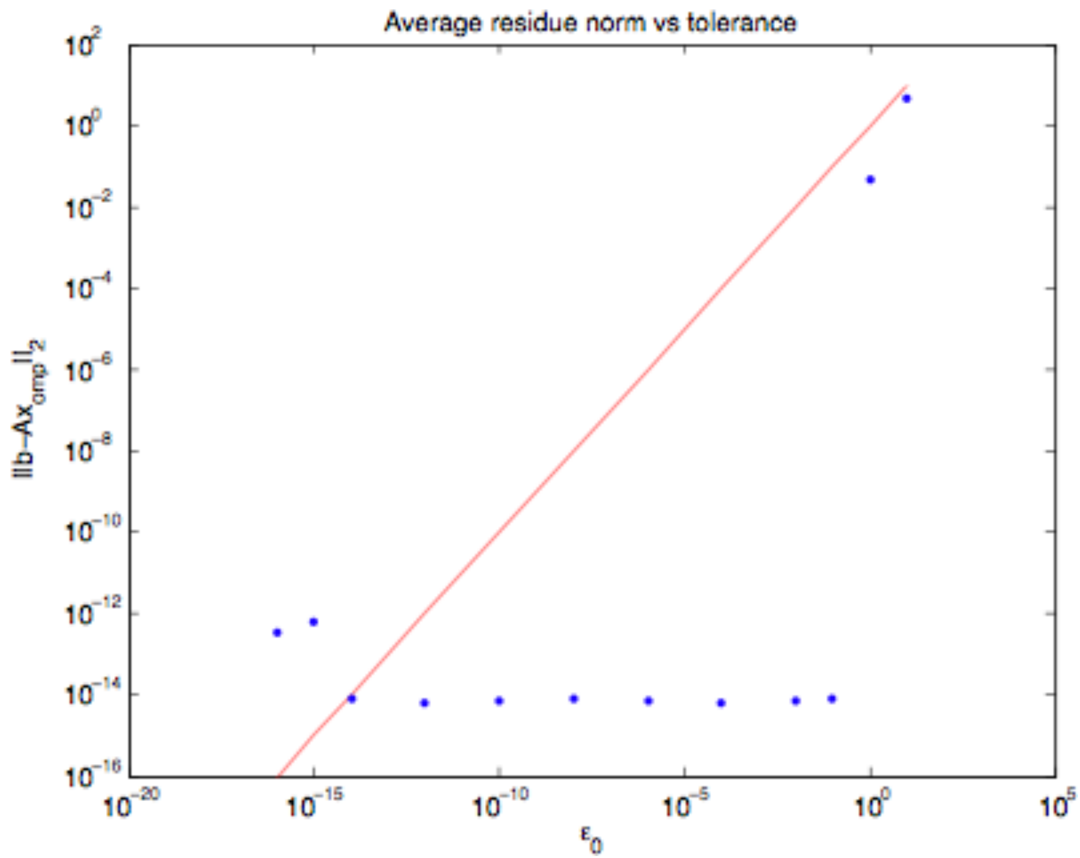
Validation Results

k = 2



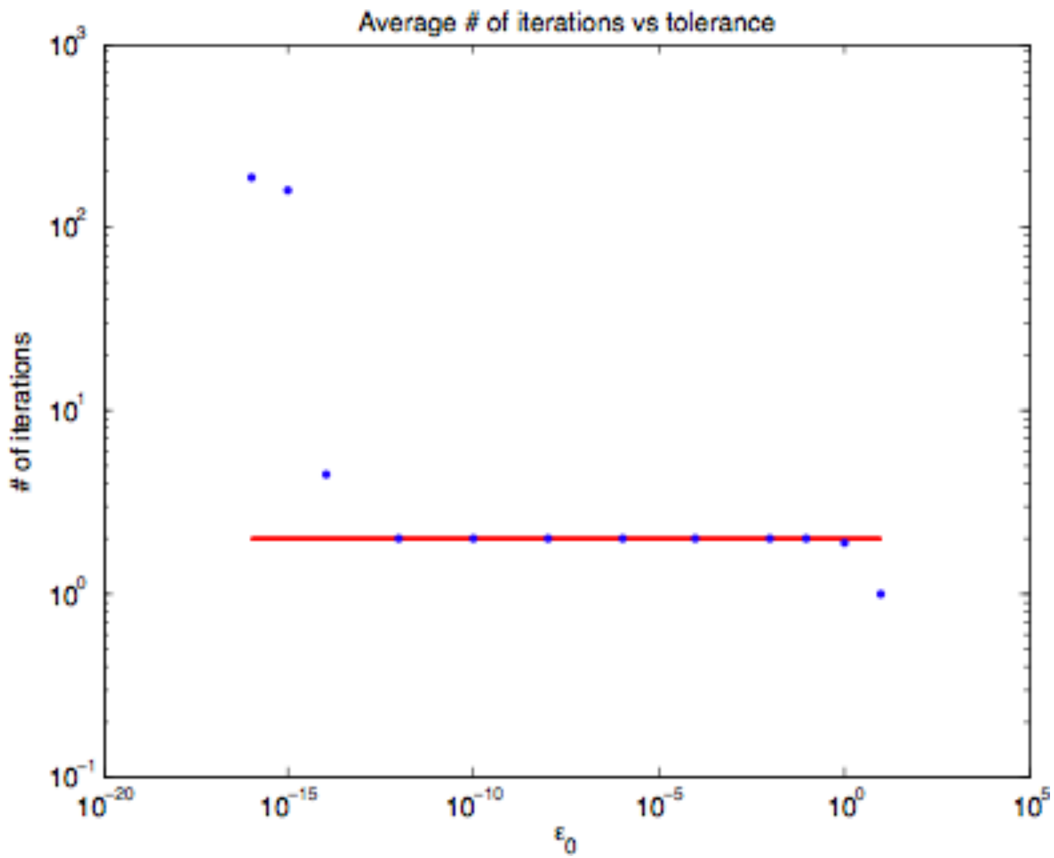
Validation Results

$k = 2$



Validation Results

k = 2



Conclusions/Recapitulation

- There are simple criteria to test the uniqueness of a given sparse solution.

- There are algorithms that find sparse solutions, e.g., OMP; and their convergence can be guaranteed when there are “sufficiently sparse” solutions.

- Our implementation of OMP is successful up to machine precision as predicted by current theoretical bounds.

Future Work

Revisiting Compression: Propose to study the compression properties of the matrix

$$\mathbf{A} = [DCT, DWT]$$

and compare it with the compression properties of DCT or DWT alone.

Study the behavior of OMP for this problem.

Interested in compression vs error graph characteristics.

References

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Review, 51 (2009), pp. 34–81.

[2] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24 (1995), pp. 227-234.

[3] G. W. Stewart, **Introduction to Matrix Computations**, Academic Press, 1973.

[4] T. Strohmer and R. W. Heath, *Grassmanian frames with applications to coding and communication*, Appl. Comput. Harmon. Anal., 14 (2004), pp. 257-275.

[5] D. S. Taubman and M. W. Mercellin, **JPEG 2000: Image Compression Fundamentals, Standards and Practice**, Kluwer Academic Publishers, 2001.

[6] G. K. Wallace, *The JPEG still picture compression standard*, Communications of the ACM, 34 (1991), pp. 30-44.