Abstract—The minimum $\ell_1$-norm solution to an underdetermined system of linear equations $y = Ax$ is often, remarkably, also the sparsest solution to that system. This sparsity-seeking property is of interest in signal processing and information transmission. However, general-purpose optimizers are much too slow for $\ell_1$ minimization in many large-scale applications.

In this paper, the Homotopy method, originally proposed by Osborne et al. and Efron et al., is applied to the underdetermined $\ell_1$-minimization problem $\min \{ ||x||_1 : y = Ax \}$. Homotopy is shown to run much more rapidly than general-purpose LP solvers when sufficient sparsity is present. Indeed, the method often has the following $k$-step solution property: if the underlying solution has only $k$ nonzeros, the Homotopy method reaches that solution in only $k$ iterative steps. This $k$-step solution property is demonstrated for several ensembles of matrices, including incoherent matrices, uniform spherical matrices, and partial orthogonal matrices. These results imply that Homotopy may be used to rapidly decode error-correcting codes in a stylized communication system with a computational budget constraint. The approach also sheds light on the evident parallelism in results on $\ell_1$ minimization and Orthogonal Matching Pursuit (OMP), and aids in explaining the inherent relations between HOMOTOPY, Least Angle Regression (LARS), OMP, and polytope faces pursuit.

Index Terms—basis pursuit, $\ell_1$ minimization, HOMOTOPY methods, Least Angle Regression (LARS), LASSO, orthogonal matching pursuit, polytope faces pursuit, sparse representations, underdetermined systems of linear equations.

I. INTRODUCTION

RECENTLY, $\ell_1$-norm minimization has attracted attention in the signal processing community [5]–[9], [20], [28], [29], [33], [34], [45], as an effective technique for solving underdetermined systems of linear equations, of the following form.

A measurement vector $y \in \mathbb{R}^d$ is generated from an unknown signal of interest $x_0 \in \mathbb{R}^n$ by a linear transformation $y = Ax_0$, with a known matrix $A$ of size $d \times n$. An important factor in these applications is that $d < n$, so the system of equations $y = Ax$ is underdetermined. We solve the convex optimization problem

$$\left(P_1\right) \min_{x} \{ ||x||_1 : y = Ax \}$$

obtaining a vector $x_1$ which we consider an approximation to $x_0$.

Applications of $(P_1)$ have been proposed in the context of time–frequency representation [9], overcomplete signal representation [20], [28], [29], [33], [34], texture/geometry separation [47], [48], compressed sensing (CS) [14], [7], [55], rapid MR Imaging [38], [39], removal of impulsive noise [21], and decoding of error-correcting codes (ECC) [5], [45]. In such applications, the underlying problem is to obtain a solution to $y = Ax$ which is as sparse as possible, in the sense of having few nonzero entries. The above-cited literature shows that often, when the desired solution $x_0$ is sufficiently sparse, $(P_1)$ delivers either $x_0$ or a reasonable approximation.

Traditionally, the problem of finding sparse solutions has been catalogued as belonging to a class of combinatorial optimization problems, whereas $(P_1)$ can be solved by linear programming, and is thus dramatically more tractable in general. Nevertheless, general-purpose LP solvers ultimately involve solution of “full” $n \times n$ linear systems, and require many applications of such solvers, each application costing order $O(n^3)$ flops.

Many of the interesting problems where $(P_1)$ has been contemplated are very large in scale. For example, already 10 years ago, Basis Pursuit [9] tackled problems with $d = 8,192$ and $n = 262,144$, and problem sizes approached currently [54], [7], [55], [39] are even larger. Such large-scale problems are simply too large for general-purpose strategies to be used in routine processing.

A. Greedy Approaches

Many of the applications of $(P_1)$ can instead be attacked heuristically by fitting sparse models, using greedy stepwise least squares. This approach is often called Matching Pursuit or Orthogonal Matching Pursuit (OMP) [12] in the signal processing literature. Rather than minimizing an objective function, OMP constructs a sparse solution to a given problem by iteratively building up an approximation; the vector $y$ is approximated as a linear combination of a few columns of $A$, where the active set of columns to be used is built column by column, in a greedy fashion. At each iteration, a new column is added to the active set—the column that best correlates with the current residual.

Although OMP is a heuristic method, in some cases it works marvelously. In particular, there are examples where the data $y$ admit sparse synthesis using only $k$ columns of $A$, and greedy selection finds exactly those columns in just $k$ steps. Perhaps surprisingly, optimists can cite theoretical work supporting this notion; work by Tropp et al. [51], [53] and Donoho et al. [19] has shown that OMP can, in certain circumstances, succeed at finding the sparsest solution. Yet, theory provides comfort for
pessimists too; OMP fails to find the sparsest solution in certain scenarios where $\ell_1$ minimization succeeds [9], [15], [51].

Optimists can also rely on empirical evidence to support their hopeful attitude. By now several research groups have conducted studies of OMP on problems where $A$ is random with independent and identically distributed (i.i.d.) Gaussian elements and $x_0$ is sparse but random (we are aware of experiments by Tropp, by Petukhov, as well as ourselves) all with the conclusion that OMP—despite its greedy and heuristic nature—performs roughly as well as $(P_1)$ in certain problem settings. Later in this paper we document this in detail. However, to our knowledge, there is no theoretical basis supporting these empirical observations.

This has the air of mystery—why do $(P_1)$ and OMP behave so similarly, when one has a rigorous foundation and the other one apparently does not? Is there some deeper connection or similarity between the two ideas? If there is a similarity, why is OMP so fast while $\ell_1$ minimization is so slow?

In this paper we aim to shed light on all these questions.

B. A Continuation Algorithm to Solve $(P_1)$

In parallel with developments in the signal processing literature, there has also been interest in the statistical community in fitting regression models while imposing $\ell_1$-norm constraints on the regression coefficients. Tibshirani [50] proposed the so-called LASSO problem, which we state using our notation as follows:

$$
(L_q) \quad \min_x ||y - Ax||_2^2 \text{ subject to } ||x||_1 \leq q;
$$

in words: a least-squares fit subject to an $\ell_1$-norm constraint on the coefficients. In Tibshirani’s original proposal, $A_{d \times n}$ was assumed to have $d > n$, i.e., representing an overdetermined linear system.

It is convenient to consider instead the unconstrained optimization problem

$$
(D_\lambda) \quad \min_x ||y - Ax||_2^2/2 + \lambda ||x||_1
$$
i.e., a form of $\ell_1$-penalized least-squares. Indeed, problems $(L_q)$ and $(D_\lambda)$ are equivalent under an appropriate correspondence of parameters. To see that, associate with each problem $(D_\lambda): \lambda \in [0, \infty)$ a solution $x_\lambda$ (for simplicity assumed unique). The set $\{x_\lambda : \lambda \in [0, \infty)\}$ identifies a solution path, with $x_\lambda = 0$ for $\lambda$ large, and as $\lambda \to 0$, $x_\lambda$ converging to the solution of $(P_1)$. Similarly, $\{\hat{x}_q : q \in [0, \infty)\}$ traces out a solution path for $(L_q)$, with $\hat{x}_q = 0$ for $q = 0$ and, as $q$ increases, $\hat{x}_q$ converging to the solution of $(P_1)$. Thus, there is a reparametrization $q(\lambda)$ defined by $q(\lambda) = ||x_\lambda||_1$ so that the solution paths of $(D_\lambda)$ and $(L(q(\lambda)))$ coincide.

In the classic overdetermined setting, $d > n$, Osborne, Presnell, and Turlach [41], [42] made the useful observation that the solution path of $(D_\lambda)$, $\lambda \geq 0$ (or $(L_q)$, $q \geq 0$) is polygonal. Further, they characterized the changes in the solution $x_\lambda$ at vertices of the polygonal path. Vertices on this path correspond to solution vectors of $(D_\lambda)$, i.e., vectors having nonzero elements only on a subset of entries. As we move along the solution path, the subset is piecewise constant as a function of $\lambda$, changing only at critical values of $\lambda$, corresponding to the vertices on the polygonal path. This evolving subset is called the active set. Based on these observations, they presented the HOMOTOPY algorithm, which follows the solution path by jumping from vertex to vertex of this polygonal path. It starts at $x_\lambda = 0$ for $\lambda$ large, with an empty active set. At each vertex, the active set is updated through the addition and removal of variables. Thus, in a sequence of steps, it successively obtains the solutions $x_\lambda$ at a special problem-dependent sequence $\lambda$ associated to vertices of the polygonal path. The name HOMOTOPY refers to the fact that the objective function for $(D_\lambda)$ is undergoing a Homotopy from the $\ell_2$ constraint to the $\ell_1$ objective as $\lambda$ decreases.

Efron, Hastie, Johnstone, and Tibshirani [26] extended the work of Osborne et al., and formally showed that the HOMOTOPY algorithm solves the minimization problem $(D_\lambda)$ for all $\lambda > 0$, i.e., that it yields the entire solution path of $(D_\lambda)$. Malioutov et al. [40] were the first to apply the HOMOTOPY method to the formulation $(D_\lambda)$ in the underdetermined setting, when the data are noisy. In this paper, we also suggest using the HOMOTOPY method for solving $(P_1)$ in the underdetermined setting.

Homotopy Algorithm for Solving $(P_1)$: Apply the HOMOTOPY method: follow the solution path from $x_{\lambda_0} = 0$ to $\hat{x}_0$. Upon reaching the $\lambda = 0$ limit, $(P_1)$ is solved.

Traditionally, to solve $(P_1)$, one would apply the simplex algorithm or an interior-point method, which, in general, starts out with a dense solution and converges to the solution of $(P_1)$ through a sequence of iterations, each requiring the solution of a full linear system. In contrast, the HOMOTOPY method starts out at $x_{\lambda_0} = 0$, and successively builds a sparse solution by adding or removing elements from its active set. Clearly, in a sparse setting, this latter approach is much more favorable, since, as long as the solution has few nonzeros, HOMOTOPY will reach the solution in a few steps.

Numerically, each step of the algorithm involves the rank-one update of a linear system, and so if the whole procedure stops in $k$ steps, yielding a solution with $k$ nonzeros, its overall complexity is bounded by $k^3 + kdn$ flops. For $k \ll d$ and $d \ll n$, this is far better than the $d^3/3$ flops it would take to solve just one $d \times d$ linear system. Moreover, to solve $(D_\lambda)$ for all $\lambda \geq 0$ by a traditional approach, one would need to repeatedly solve a quadratic program for every $\lambda$ value of interest. For any problem size beyond the very smallest, that would be prohibitively time consuming. In contrast, HOMOTOPY delivers all the solutions $x_\lambda$ to $(D_\lambda)$, $\lambda \geq 0$.

C. Least Angle Regression (LARS)

In [26], Efron et al. suggested an alternate derivation of the HOMOTOPY algorithm, taking a geometric perspective, which is quite instructive. They defined a basic procedure, called Least Angle Regression (LARS), inspired by stagewise regression methods in statistics. The LARS algorithm maintains an active set of nonzero variables composing the current solution. At each step, the algorithm adds a new element to the active set, by taking a step along an equiangular direction, that is, a direction having equal angles with the vectors in the active set.

They further noted that the HOMOTOPY algorithm is very similar to LARS, with the exception that, when moving to a new
vertex of the solution polygon, HOMOTOPY may remove existing elements from the active set. Thus, they suggested a variant of LARS, named LARS with the LASSO modification, which is identical to the HOMOTOPY algorithm, and thus solves $(D_\lambda)$. In what follows, we use the common name HOMOTOPY to refer to both algorithms.

Efron et al. observed that, quite remarkably, experimental results suggest that the solution obtained by following the basic LARS procedure is often identical to the LASSO solution. This equality is very interesting in the present context, because LARS is so similar to OMP. Both algorithms build up a model a step at a time, adding a new variable to the active set at each step, and ensuring that the new variable is in some sense the most important among the potential candidate variables. The details of determining importance differ but in both cases involve the inner product between the candidate new variables and the current residual.

In short, a stepwise algorithm with a greedy flavor can sometimes produce the same result as full-blown $\ell_1$ minimization. This suggests a possibility which can be stated in two different ways:

- … that an algorithm for quasi-$\ell_1$ minimization runs just as rapidly as OMP;
- … that an algorithm visibly very similar to OMP can be just as effective as $\ell_1$ minimization.

D. Properties of HOMOTOPY

As Efron et al. noted [26], the difference between HOMOTOPY and LARS lies in the provision by HOMOTOPY for terms to leave as well as enter the active set. From an algorithmic viewpoint, this difference translates into a few lines of code. And yet this small variation makes HOMOTOPY rigorously able to solve a global optimization problem. On the other hand, it implies that the number of steps required by HOMOTOPY can, in principle, be significantly greater than the number of steps required by LARS, as terms may enter and leave the active set numerous times. If so, we observe “model churning” which causes HOMOTOPY to run slowly. We present evidence that under favorable conditions, such churning does not occur. In such cases, the HOMOTOPY algorithm is roughly as fast as both LARS and OMP.

In this paper, we consider two settings for performance measurement: deterministic incoherent matrices and random matrices. We demonstrate that, when a $k$-sparse representation exists, $k \ll d$, the HOMOTOPY algorithm finds it in $k$ steps. Results in each setting parallel existing results about OMP in that setting. Moreover, the preceding discussion indicates that each step of HOMOTOPY is identical to a step of LARS and therefore very similar to a corresponding step of OMP. We interpret this parallelism as follows.

The HOMOTOPY algorithm, in addition to solving the $\ell_1$ minimization problem, runs just as rapidly as the heuristic algorithms OMP/LARS, in those problem suites where those algorithms have been shown to correctly solve the sparse representation problem.

Since there exists a wide range of cases where $\ell_1$-minimization is known to find the sparsest solution while OMP is known to fail [9], [15], [51], HOMOTOPY gives in some sense the best of both worlds by a single algorithm: speed where possible, sparsity where possible.

In addition, our viewpoint sheds new light on previously observed similarities between results about $\ell_1$ minimization and about OMP. Previous theoretical work [19], [51], [53] found that these two seemingly disparate methods had comparable theoretical results for finding sparse solutions. Our work exhibits inherent similarities between the two methods which motivate these parallel findings.

Fig. 1 summarizes the relation between $\ell_1$ minimization and OMP, by highlighting the intimate bonds between HOMOTOPY, LARS, and OMP. We elaborate on these inherent connections in a later section.

E. Main Results

Our results about the stopping behavior of HOMOTOPY require some terminology.

Definition 1: A problem suite $S(E, V; d, n, k)$ is a collection of problems defined by three components: a) an ensemble $E$ of matrices $A$ of size $d \times n$; b) an ensemble $V$ of $n$-vectors $\alpha$ with at most $k$ nonzeros; c) an induced ensemble of left-hand sides $y = A\alpha$.

Where the ensemble $V$ is omitted from the definition of a problem suite, we interpret that as saying that the relevant statements hold for any nonzero distribution. Otherwise, $V$ will take one of three values: (UNIFORM), when the nonzeros are i.i.d. uniform on $[0, 1]$; (GAUSS), when the nonzeros are i.i.d. standard normal; and (BERNOULLI), when the nonzeros are i.i.d. equiprobable Bernoulli distributed with values $\pm 1$. 

![Diagram of Bridging $\ell_1$ minimization and OMP. (1) HOMOTOPY provably solves $\ell_1$ minimization problems [26], (2) LARS is obtained from HOMOTOPY by removing the sign constraint check. (3) OMP and LARS are similar in structure, the only difference being that OMP solves a least-squares problem at each iteration, whereas LARS solves a linearly penalized least-squares problem. There are initially surprising similarities of results concerning the ability of OMP and $\ell_1$ minimization to find sparse solutions. We view those results as statements about the $k$-step solution property. We present evidence showing that for sufficiently small $k$, steps (2) and (3) do not affect the threshold for the $k$-sparse solution property. Since both algorithms have the $k$-step solution property under the given conditions, they both have the sparse solution property under those conditions.](image-url)
Definition 2: An algorithm \( A \) has the \( k \)-step solution property at a given problem instance \((A, y)\) drawn from a problem suite \( S(E, V; d, n, k) \) if, when applied to the data \((A, y)\), the algorithm terminates after at most \( k \) steps with the correct solution.

Our main results establish the \( k \)-step solution property in specific problem suites.

1) Incoherent Systems: The mutual coherence \( M(A) \) of a matrix \( A \) whose columns are normalized to unit length is the maximal off-diagonal entry of the Gram matrix \( A^T A \). We call the collection of matrices \( A \) with \( M(A) \leq \mu \) the incoherent ensemble with coherence bound \( \mu \) (denoted \( \text{IN}C_\mu \)). Let \( S(\text{IN}C_\mu; d, n, k) \) be the suite of problems with \( d \times n \) matrices drawn from the incoherent ensemble \( \text{IN}C_\mu \), with vectors \( \alpha_0 \) having \( \|\alpha_0\|_0 \leq k \). For the incoherent problem suite, we have the following result.

**Theorem 1:** Let \((A, y)\) be a problem instance drawn from \( S(\text{IN}C_\mu; d, n, k) \). Suppose that
\[
k \leq (\mu^{-1} + 1)/2.
\]
Then, the HOMOTOPY algorithm runs \( k \) steps and stops, delivering the solution \( \alpha_0 \).

The condition (1) has appeared elsewhere; work by Donoho et al. [20], [18], Gribonval and Nielsen [33], and Tropp [51] established that when (1) holds, the sparsest solution is unique and equal to \( \alpha_0 \), and both \( \ell_1 \) minimization and OMP recover \( \alpha_0 \). It was further shown that when condition (1) is satisfied, OMP recovers the sparsest solution in \( k \) steps or less. Theorem

2) Random Matrices: We first consider \( d \times n \) random matrices \( A \) from the Uniform Spherical Ensemble (USE). Such matrices have columns independently and uniformly distributed on the sphere \( S^{d-1} \). Consider the suite \( S(\text{USE}; d, n, k) \) of random problems where \( A \) is drawn from the USE and where \( \alpha_0 \) has at most \( k \) nonzeros at \( k \) randomly chosen sites, with the sites chosen independently of \( A \).

**Empirical Finding 1:** Fix \( \delta \in (0, 1) \). Let \( d = d_n = \lfloor \delta n \rfloor \). Let \((A, y)\) be a problem instance drawn from \( S(\text{USE}; d, n, k) \).
There is \( \varepsilon_n > 0 \) small, so that, with high probability, for
\[
k \leq \frac{d_n}{2\log(n)}(1 + \varepsilon_n)
\]
the HOMOTOPY algorithm runs \( k \) steps and stops, delivering the solution \( \alpha_0 \).

**Remarks:**
- We conjecture that corresponding to this empirical finding is a theorem, in which the conclusion is that as \( n \to \infty \), \( \varepsilon_n \to 0 \).
- The empirical result is in fact stronger. We demonstrate that for problem instances with \( k \) emphatically not satisfying (2), but rather
\[
k \geq \frac{d_n}{2\log(n)}(1 + \varepsilon_n)
\]
then, with high probability for large \( n \), the HOMOTOPY algorithm fails to terminate in \( k \) steps. Thus, (2) delineates a boundary between two regions in \( (k, d, n) \) space; one where the \( k \)-step property holds with high probability, and another where the chance of \( k \)-step termination is very low.
- This empirical finding also holds for matrices drawn from the Random Signs Ensemble (RSE). Matrices in this ensemble are constrained to have their entries \( \pm 1 \) (suitably normalized to obtain unit-norm columns), with signs drawn independently and with equal probability.

We next consider ensembles made of partial orthogonal transformations. In particular, we consider the following matrix ensembles.

- **Partial Fourier Ensemble (PFE).** Matrices in this ensemble are obtained by sampling at random \( d \) rows of an \( n \) by \( n \) Fourier matrix.
- **Partial Hadamard Ensemble (PHE).** Matrices in this ensemble are obtained by sampling at random \( d \) rows of an \( n \) by \( n \) Hadamard matrix.
- **Uniform Random Projection Ensemble (URPE).** Matrices in this ensemble are obtained by generating an \( n \) by \( n \) random orthogonal matrix and sampling \( d \) of its rows at random.

As it turns out, when applied to problems drawn from these ensembles, the HOMOTOPY algorithm maintains the \( k \)-step property at a substantially greater range of signal sparsities. In detail, we give evidence supporting the following conclusion.

**Empirical Finding 2:** Fix \( \delta \in (0, 1) \). Let \( d = d_n = \lfloor \delta n \rfloor \). Let \((A, y)\) be a problem instance drawn from \( S(E, V; d, n, k) \), with
\[
E \in \{\text{PFE}, \text{PHE}, \text{URPE}\}
\]
and
\[
V \in \{\text{UNIFORM}, \text{GAUSS}, \text{BERNOULLI}\}.
\]
The empirical function \( \hat{\rho}(\delta) \), depicted later in Fig. 7, marks a transition such that, for \( k/d_n \leq \hat{\rho}(\delta) \), the HOMOTOPY algorithm runs \( k \) steps and stops, delivering the solution \( \alpha_0 \).

A third, and equally important, finding we present involves the behavior of the HOMOTOPY algorithm when the \( k \)-step property does not hold. We establish the following.

**Empirical Finding 3:** Let \((A, y)\) be a problem instance drawn from \( S(E, V; d, n, k) \), with \( E \in \{\text{PFE}, \text{PHE}, \text{URPE}\} \), \( V \in \{\text{UNIFORM}, \text{GAUSS}, \text{BERNOULLI}\} \), and \( (d, n, k) \) chosen so that the \( k \)-step property does not hold. With high probability, HOMOTOPY, when applied to \((A, y)\), operates in either of two modes.

1) It recovers the sparse solution \( \alpha_0 \) in \( c_s \cdot d \) iterations, with \( c_s \) a constant depending on \( E, V, \) empirically found to be less than 1.6.
2) It does not recover $\alpha_0$, returning a solution to $(P_1)$ in $c_f \cdot d$ iterations, with $c_f$ a constant depending on $E$ and $V$, empirically found to be less than $4.85$.

We interpret this finding as saying that we may safely apply HOMOTOPY to problems whose solutions are not known a priori to be sparse, and still obtain a solution rapidly. In addition, if the underlying solution to the problem is indeed sparse, HOMOTOPY is particularly efficient in finding it. This finding is of great usefulness in practical applications of HOMOTOPY.

F. Error Correction in a Sparse Noisy Channel

The results presented above have far-reaching implications in practical scenarios. We demonstrate these with a stylized problem inspired by digital communication systems, of correcting gross errors in a digital transmission. Recently, researchers pointed out that $l_1$-minimization/sparsity ideas have a role to play in decoding linear error-correcting codes over $\mathbb{Z}$ [5, 45], a problem known, in general, to be NP-hard. Applying the HOMOTOPY algorithm in this scenario, we now demonstrate the following remarkable property, an implication of Empirical Finding 2.

**Corollary 1:** Consider a communications channel corrupting a fraction $\epsilon < 0.24$ of every $n$ transmitted integers, the corrupted sites chosen by a $\text{Bernoulli}(\epsilon)$ i.i.d. selection. For large $n$, there exists a rate-1/5 code, that, with high probability, can be decoded, without error, in at most $0.24n$ HOMOTOPY steps.

In other words, even when as many as a quarter of the transmitted entries are corrupted, HOMOTOPY operates at peak performance, recovering the encoded bits in a fixed number of steps, known in advance. This feature enables the design of real-time decoders obeying a hard computational budget constraint, with a fixed number of fops per decoding effort.

Let us describe the coding strategy in more detail. Assume $\theta$ is a digital signal of length $p$, with entries $\pm 1$, representing bits to be transmitted over a digital communication channel. Prior to transmission, we encode $\theta$ with a rate-1/5 code constructed in the following manner. Let $H$ be an $n \times n$ Hadamard matrix, $n \approx 5p$. Construct the "encoding matrix" $E$ by selecting $p$ rows of $H$ at random. The "decoding matrix" $D$ is then composed of the other $4p$ rows of $H$. The encoding stage amounts to computing $x = E^T \theta$, with $x$ the encoded signal, of length $n$. At the decoder, we receive $r = x + z$, where $z$ has nonzero in $k$ random positions, and the nonzero follow a specified distribution. At the decoder, we apply HOMOTOPY to solve

$$\min \| \alpha \|_1 \text{ subject to } D\alpha = D r;$$

call the solution $\hat{\alpha}$. The decoder output is then

$$\hat{\theta} = \text{sgn}(E^T(r - \hat{\alpha})).$$

The key property being exploited is the mutual orthogonality of $D$ and $E$. Specifically, note that $D r = D(E^T \theta + z) = D z$. Hence, $(EC_1)$ is essentially solving for the sparse error pattern.

To demonstrate our claim, we set $p = 256$, and considered error rates $k = \epsilon n$, with $\epsilon$ varying in [0.04, 0.4]. In each case, we generated a problem instance as described above, and measured the number of steps HOMOTOPY takes to reach the correct solution. Results are plotted in Fig. 2, showing four different noise distributions. Inspecting the results, we make two important observations. First, as claimed, for the noise distributions considered, as long as the fraction of corrupt entries is less that 0.24$n$, HOMOTOPY recovers the correct solution in $k$ steps, as

![Performance of HOMOTOPY in decoding a rate-1/5 partial Hadamard code](image)

Each panel shows the number of iterations, divided by $n$, that HOMOTOPY takes in order to recover the coded signal, versus $\epsilon$, the fraction of corrupt entries in the transmitted signal, with the noise distributed (a) $\mathcal{N}(0, 1)$. (b) $\pm 1$ with equal probabilities. (c) Cauchy. (d) Rayleigh. In each plot "o" indicates $k$-step recovery, and "x" implies recovery in more than $k$ steps.
implied by Empirical Finding 2. Second, as \( \epsilon \) increases, and \( k \) leaves the \( k \)-step region defined by (2), HOMOTOPY still does not take many more than \( k \) steps to reach a solution.

G. Relation to Earlier Work

Our results seem interesting in light of previous work by Osborne et al. [41] and Efron et al. [26]. In particular, our proposal for fast solution of \( \ell_1 \) minimization amounts simply to applying to \( (P_1) \) previously known algorithms (HOMOTOPY a.k.a LARS/LASSO) designed for fitting equations to noisy data. We make the following contributions.

• Few Steps. Efron et al. considered the overdetermined noisy setting, and remarked that while, in principle, HOMOTOPY could take many steps, they had encountered examples where it took a direct path to the solution, in which a term, once entered into the active set, stayed in the active set [26]. Our work in the noiseless underdetermined setting formally identifies a precise phenomenon, namely, the \( k \)-step solution property, and delineates a range of problem suites where this phenomenon occurs.

• Similarity of HOMOTOPY and LARS. Efron et al., in the overdetermined noisy setting, commented that, while in principle the solution paths of HOMOTOPY and LARS could be different, in practice they came across examples where HOMOTOPY and LARS yielded very similar results [26]. Our work in the noiseless case formally defines a property which implies that HOMOTOPY and LARS have the same solution path, and delineates a range of problem suites where this property holds. In addition, we present simulation studies showing that, over a region in parameter space, where \( \ell_1 \) minimization recovers the sparsest solution, LARS also recovers the sparsest solution.

• Similarity of HOMOTOPY and OMP. In the random setting, Tropp and Gilbert demonstrated that OMP recovers the sparsest solution under a condition similar to (2), although with a somewhat worse constant term; their result (and proof) can be interpreted in the light of our work as saying that OMP actually has the \( k \)-step solution property under their condition. In fact, it has the \( k \)-step solution property under the condition (2). Below, we elaborate on the connections between HOMOTOPY and OMP leading to these similar results.

• Similarity of HOMOTOPY and Polytope Faces Pursuit. Recently, Plumbley introduced a greedy algorithm to solve \( (P_1) \) in the dual space, which he named Polytope Faces Pursuit (PFP) [44], [43]. The algorithm bears strong resemblance to HOMOTOPY and OMP. Below, we elaborate on Plumbley’s work, to show that the affinities are not coincidental, and in fact, under certain conditions, PFP is equivalent to HOMOTOPY. We then conclude that PFP maintains the \( k \)-step solution property under the same conditions required for HOMOTOPY.

In short, we provide theoretical underpinnings, formal structure, and empirical findings. We also believe our viewpoint clarifies the connections between HOMOTOPY, LARS, and PFP.

H. Contents

The paper is organized as follows. Section II reviews the HOMOTOPY algorithm in detail. Section III presents running-time simulations alongside a formal complexity analysis of the algorithm, and discusses evidence leading to Empirical Finding 3. In Section IV, we prove Theorem 1, the \( k \)-step solution property for the sparse incoherent problem suite. In Section V, we demonstrate Empirical Finding 1, the \( k \)-step solution property for the sparse USE problem suite. Section VI follows, with evidence to support Empirical Finding 2. In Section VII, we elaborate on the relation between \( \ell_1 \) minimization and OMP hinted at in Fig. 1. This provides a natural segue to Section VIII, which discusses the connections between HOMOTOPY and PFP. Section IX then offers a comparison of the performance of HOMOTOPY, LARS, OMP, and PFP in recovering sparse solutions. Section X briefly discusses approximate methods for recovery of sparse solutions, and Section XI has concluding remarks. Finally, Section XII discusses the software accompanying this paper.

II. THE HOMOTOPY ALGORITHM

We begin with a description of the HOMOTOPY algorithm, following the derivation in [26]. Recall that the general principle undergirding homotopy methods is to trace a solution path, parametrized by one or more variables, while evolving the parameter vector from an initial value, for which the corresponding solution is known, to the desired value. For the LASSO problem \( (D) \), this implies following the solution \( x_\lambda \), starting at \( \lambda \) large and \( x_\lambda = 0 \), and terminating when \( \lambda \to 0 \) and \( x_\lambda \) converging to the solution of \( (P_1) \). The solution path is followed by maintaining the optimality conditions of \( (D) \) at each point along the path.

Specifically, let \( f_\lambda(x) \) denote the objective function of \( (D) \). By classical ideas in convex analysis, a necessary condition for \( x_\lambda \) to be a minimizer of \( f_\lambda(x) \) is that the zero vector is an element of the subdifferential of \( f_\lambda \) at \( x_\lambda \), denoted \( \partial_{x_\lambda} f_\lambda(x_\lambda) \) [2, Sec. 4.2]. We calculate

\[
\partial_{x_\lambda} f_\lambda(x_\lambda) = -A^T(y - Ax_\lambda) + \lambda \partial_{x_\lambda} \|x_\lambda\|_1 \tag{3}
\]

where \( \partial_{x_\lambda} \|x_\lambda\|_1 \) is the subdifferential of \( \|x_\lambda\|_1 \). Let \( I = \{ i : x_\lambda(i) \neq 0 \} \) denote the support of \( x_\lambda \), and call \( c = A^T(y - Ax_\lambda) \) the vector of residual correlations. Then, the condition \( 0 \in \partial_{x_\lambda} f_\lambda(x_\lambda) \) can be written equivalently as the two conditions

\[
c(I) = \lambda \cdot \text{sgn}(x_\lambda(I)) \tag{4}
\]

and

\[
|c(I')| \leq \lambda \tag{5}
\]

In words, residual correlations on the support \( I \) must all have magnitude equal to \( \lambda \), and signs that match the corresponding elements of \( x_\lambda \), whereas residual correlations off the support must have magnitude less than or equal to \( \lambda \). The HOMOTOPY
algorithm now follows from these two conditions, by tracing out the optimal path $x_\ell$ that maintains (4) and (5) for all $\lambda \geq 0$. The key to its successful operation is that the path $x_\lambda$ is a piecewise linear path, with a discrete number of vertices [26], [41].

The algorithm starts with an initial solution $x_0 = 0$, and operates in an iterative fashion, computing solution estimates $x_\ell$, $\ell = 1, 2, \ldots$. Throughout its operation, it maintains the active set $I$, which satisfies

$$I = \{ j : |c_j| = \|x\|_\infty = \lambda \}$$

as implied by conditions (4) and (5). At the $\ell$th stage, HOMOTOPY first computes an update direction $d_\ell$, by solving

$$A^T_A A d_\ell(I) = \text{sgn}(c(I))$$

with $d_\ell$ set to zero in coordinates not in $I$. This update direction ensures that the magnitudes of residual correlations on the active set all decline equally. The algorithm then computes the step size to the next breakpoint along the homotopy path. Two scenarios may lead to a breakpoint. First, that a nonactive element of $c_\ell$ would increase in magnitude beyond $\lambda$, violating (5). This first occurs when

$$\gamma_{\ell}^+ = \min_{i \in I^+} \left\{ \frac{\lambda - c_i}{1 - a_i^T v_\ell}, \frac{\lambda + c_i}{1 + a_i^T v_\ell} \right\}$$

where $v_\ell = A d_\ell(I)$, and the minimum is taken only over positive arguments. Call the minimizing index $i^+$. The second scenario leading to a breakpoint in the path occurs when an active coordinate crosses zero, violating the sign agreement in (4). This first occurs when

$$\gamma_{\ell}^- = \min_{i \in I^-} \{ -x_i / d_\ell(i) \}$$

where again the minimum is taken only over positive arguments. Call the minimizing index $i^-$. HOMOTOPY then marches to the next breakpoint, determined by

$$\gamma_{\ell} = \min\{ \gamma_{\ell}^+, \gamma_{\ell}^- \}$$

updates the active set, by either appending $I$ with $i^+$, or removing $i^-$, and computes a solution estimate

$$x_\ell = x_{\ell-1} + \gamma_{\ell}d_\ell.$$

The algorithm terminates when $\|x_\ell\|_\infty = 0$, and the solution of (P1) has been reached.

Remarks:

1) In the algorithm description above, we implicitly assume that at each breakpoint on the homotopy path, at most one new coordinate is considered as candidate to enter the active set (Efron et al. called this the “one at a time” condition [26]). If two or more vectors are candidates, more care must be taken in order to choose the correct subset of coordinates to enter the active set; see [26] for a discussion.

2) In the Introduction, we noted that the LARS scheme closely mimics HOMOTOPY, the main difference being that LARS does not allow removal of elements from the active set. Indeed, to obtain the LARS procedure, one simply follows the sequence of steps described above, omitting the computation of $i^-$ and $\gamma_{\ell}^-$, and replacing (10) with

$$\gamma_{\ell} = \gamma_{\ell}^+.$$

The resulting scheme adds a single element to the active set at each iteration, never removing active elements from the set.

3) The HOMOTOPY algorithm may be easily adapted to deal with noisy data. Assume that rather than observing $y_0 = A_0 x_0$, we observe a noisy version $y = A_0 x_0 + z$, with $\|z\|_2 \leq \epsilon_n$. Donoho et al. [19], [16] have shown that, for certain matrix ensembles, the solution $x_{\lambda}$ of (L) with $q = \epsilon_n$ has an error which is at worst proportional to the noise level. To solve for $x_{\lambda}$, we simply apply the HOMOTOPY algorithm as described above, terminating as soon as the residual satisfies $\|x\|_2 \leq \epsilon_n$. Since in most practical scenarios it is not sensible to assume that the measurements are perfectly noiseless, this attribute of HOMOTOPY makes it an apt choice for use in practice.

### III. Computational Cost

In earlier sections, we claimed that HOMOTOPY can solve the problem (P1) much faster than general-purpose LP solvers. In particular, when the $k$-step solution property holds, HOMOTOPY runs in a fraction of the time it takes to solve one full linear system, making it as efficient as fast stepwise algorithms such as OMP. To support these assertions, we now present the results of running-time simulations, complemented by a formal analysis of the asymptotic complexity of HOMOTOPY.

#### A. Running Times

To evaluate the performance of the HOMOTOPY algorithm applied to (P1), we compared its running times on instances of the problem suite $S(\text{USE, GAUS, d, n, k})$ with two state-of-the-art algorithms for solving Linear Programs. The first, LP_Solve, is a Mixed Integer Linear Programming solver, implementing a variant of the Simplex algorithm [1]. The other, PDCO, a Primal-Dual Convex Optimization solver, is a log-barrier interior point method, written by Michael Saunders of the Stanford Optimization Laboratory [46]. Table I shows the running times for HOMOTOPY, LP_Solve and PDCO, for various problem dimensions. The figures appearing in the table were measured on a 3-GHz Xeon workstation.

<table>
<thead>
<tr>
<th>(d,n,k)</th>
<th>HOMOTOPY</th>
<th>LP_Solve</th>
<th>PDCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>(200,500,20)</td>
<td>0.03</td>
<td>2.04</td>
<td>0.90</td>
</tr>
<tr>
<td>(250,500,100)</td>
<td>0.94</td>
<td>9.27</td>
<td>1.47</td>
</tr>
<tr>
<td>(500,1000,100)</td>
<td>1.28</td>
<td>45.18</td>
<td>4.62</td>
</tr>
<tr>
<td>(750,1000,150)</td>
<td>2.15</td>
<td>85.78</td>
<td>7.68</td>
</tr>
<tr>
<td>(500,200,50)</td>
<td>0.34</td>
<td>59.52</td>
<td>6.86</td>
</tr>
<tr>
<td>(1000,2000,200)</td>
<td>7.32</td>
<td>407.99</td>
<td>22.90</td>
</tr>
<tr>
<td>(1600,4000,320)</td>
<td>31.70</td>
<td>2661.42</td>
<td>122.47</td>
</tr>
</tbody>
</table>

Examination of the running times in the table suggests two important observations. First, when the underlying solution to the problem (P1) is sparse, a tremendous saving in computation time is achieved using HOMOTOPY, compared to traditional LP solvers. For instance, when $A = 500 \times 2000$, and $y$ admits sparse synthesis with $k = 50$ nonzeros, HOMOTOPY terminates in about 0.34 s, 20 times faster than PDCO, and over 150 times faster than LP_Solve. Second, when the linear system is highly
underdetermined (i.e., \(d/n\) is small), even when the solution is not sparse, HOMOTOPY is more efficient than either LP_Solve or PDCCO. This latter observation is of particular importance for applications, as it implies that HOMOTOPY may be “safely” used to solve \(\ell_1\) minimization problems even when the underlying solution is not known to be sparse.

**B. Complexity Analysis**

The timing studies above are complemented by a detailed analysis of the complexity of the HOMOTOPY algorithm. We begin by noting that the bulk of computation is invested in the solution of the linear system (7) at each iteration. Thus, the key to an efficient implementation is maintaining a Cholesky factorization of the Gram minor \(A_I^T A_I\), updating it with the addition/removal of elements to/from the active set. This allows for fast solution for the update direction; \(O(d^3)\) operations are needed to solve (7), rather than the \(O(d^3)\) it would ordinarily take. In detail, let \(\ell = |I|\), where \(I\) denotes the current active set. The dominant calculations per iteration are the solution of (7) at a cost of \(2\ell^2\) flops, and computation of the step to the next vertex on the solution path, using \(n\delta + 6(n - \ell)\) flops. In addition, the Cholesky factor of \(A_I^T A_I\) is updated by appending a column to \(A_I\) at a cost of \(\ell^2 + \ell d + 2(\ell + d)\) flops, or “downdated” by removing a column of \(A_I\) using at most \(3\ell^2\) flops.

To conclude, without any sparsity constraints on the data, \(k\) HOMOTOPY steps would take at most \(4k\delta^2/3 + kd\ell + O(kn)\) flops. However, if the conditions for the \(k\)-step solution property are satisfied, a more favorable estimate holds.

**Lemma 1**: Let \((A, y)\) be a problem instance drawn from a suite \(S(E, V; d, n, k)\). Suppose the \(k\)-step solution property holds, and the HOMOTOPY algorithm performs \(k\) steps, each time adding a single element into the active set. The algorithm terminates in \(k^3 + kdn + 1/2k^2(d-1) + n(8k + d + 1) + 1/2k(5d-3)\) flops.

Notice that, for \(k \ll d\), the dominant term in the expression for the operation count is \(kdn\), the number of flops needed to carry out \(k\) matrix–vector multiplications. Thus, we may interpret Lemma 1 as saying that, under such favorable conditions, the computational workload of HOMOTOPY is roughly proportional to \(k\) applications of a \(d \times n\) matrix. For comparison, applying least-squares to solve the underdetermined linear system \(Ax = y\) would require \(2\delta^2n - 2d\delta^3/3\) operations [32]. Thus, for \(k \ll d\), HOMOTOPY runs in a fraction of the time it takes to solve one least-squares system.

To visualize this statement, Fig. 3 (a) displays the operation count of HOMOTOPY on a grid with varying sparsity and indeterminacy factors. In this simulation, we measured the total operation count of HOMOTOPY as a fraction of the solution of one \(d \times n\) least-squares system, for random instances of the problem suite \(S(USE, UNIFORM; d, n, k)\). We fixed \(n = 1000\), varied the indeterminacy of the system, indexed by \(\delta = d/n\), in the range \([0.1, 1]\), and the sparsity of the solution, indexed by \(\rho = k/d\), in the range \([0.05, 1]\). For reference, we superposed the theoretical bound \(\rho M\) below which the solution of \((P_\ell)\) is, with high probability, the sparsest solution (see Section IX for more details). Close inspection of this plot reveals that, below this curve, HOMOTOPY delivers the solution to \((P_\ell)\) rapidly, much faster than it takes to solve one \(d \times n\) least-squares problem.

**C. Number of Iterations**

Considering the analysis just presented, it is clear that the computational efficiency of HOMOTOPY greatly depends on the number of vertices on the polygonal HOMOTOPY path. Indeed, since it allows removal of elements from the active set, HOMOTOPY may, conceivably, require an arbitrarily large number of iterations to reach a solution. We note that this property is not shared by LARS or OMP; owing to the fact that these algorithms never remove elements from the active set, after \(d\) iterations they terminate with zero residual.

![Fig. 3. Computational cost of HOMOTOPY applied to \(S(USE, UNIFORM; d, n, k)\). Panel (a) shows the operation count as a fraction of one least-squares solution on a \(\rho - \delta\) grid, with \(n = 1000\). Panel (b) shows the number of iterations as a fraction of \(d = \delta \cdot n\). The superimposed dashed curve depicts the curve \(\rho M\) below which HOMOTOPY recovers the sparsest solution with high probability.](image-url)
HOMOTOPY takes, with high probability, no more than \(1.69 \cdot d\) iterations to obtain the minimum \(\ell_1\) solution.

### IV. Fast Solution With the Incoherent Ensemble

Let the \(d \times n\) matrix \(A\) have unit-length columns \(a_j, \|a_j\|_2 = 1\). The mutual coherence

\[
M(A) = \max_{i \neq j} \left| \langle a_i, a_j \rangle \right|
\]

measures the smallest angle between any pair of columns. As \(n > d\), this absolute inner product must be greater than zero: the columns cannot be mutually orthogonal; in fact, there is an established lower bound on the coherence of \(A\)

\[
M(A) \geq \sqrt{\frac{n - d}{d(n - 1)}},
\]

Matrices which satisfy this lower bound with equality are known as optimal Grassmannian frames [49]. There is by now much work establishing relations between the sparsity of a coefficient vector \(a_0\) and the coherence of a matrix \(A\) needed for successful recovery of \(a_0\) via \(\ell_1\) minimization [20], [18], [52], [33] or OMP [19], [51]. In detail, for a general matrix \(A\) with coherence \(M(A)\), both \(\ell_1\) minimization and OMP recover the solution \(a_0\) from data \(y = Aa_0\) whenever the number of nonzeros in \(a_0\) satisfies

\[
\|a_0\|_0 < \left( M(A)^{-1} + 1 \right)/2;
\]

see, for example, [18, Theorem 7] or [51, Corollary 3.6]. Comparing (11) with (1), we see that Theorem 1 essentially states that a parallel result holds for the HOMOTOPY algorithm.

Before proceeding to the proof of the theorem, we make a few introductory assumptions. As in the above, we assume that the HOMOTOPY algorithm operates with a problem instance \((A, y)\) as input, with \(y = Aa_0\) and \(\|a_0\|_0 = k\). To simplify notation, we assume, without loss of generality, that \(a_0\) has its nonzeros in the first \(k\) positions. Further, we operate under the convention that at each step of the algorithm, only one vector is introduced into the active set. If two or more vectors are candidates to enter the active set, assume the algorithm inserts them one at a time, on separate stages. Finally, to fix notation, let \(x_\ell\) denote the HOMOTOPY solution at the \(\ell\)th step, \(r_\ell = y - A x_\ell\) denote the residual at that step, and \(c_\ell = A^T r_\ell\) be the corresponding residual correlation vector. To prove Theorem 1, we introduce two useful notions.

**Definition 3:** We say that the HOMOTOPY algorithm has the **Correct Term Selection Property** at a given problem instance \((A, y)\), with \(y = Aa_0\), if at each iteration, the algorithm selects a new term to enter the active set from the support set of \(a_0\).

HOMOTOPY has the correct term selection property if, throughout its operation, it builds the solution using only correct terms. Thus, at termination, the support set of the solution is guaranteed to be a subset of the support set of \(a_0\).

**Definition 4:** We say that the HOMOTOPY algorithm has the **Sign Agreement Property** at a given problem instance \((A, y)\), with \(y = Aa_0\), if at every step \(\ell\), for all \(j \in I\)

\[
\text{sgn}(x_\ell(j)) = \text{sgn}(c_\ell(j)).
\]
In words, HOMOTOPY has the sign agreement property if, at every step of the algorithm, the residual correlations in the active set agree in sign with the corresponding solution coefficients. This ensures that the algorithm never removes elements from the active set.

These two properties are the fundamental building blocks needed to ensure that the $k$-step solution property holds. In particular, these two properties are necessary and sufficient conditions for successful $k$-step termination, as the following lemma shows.

**Lemma 2:** Let $(A, y)$ be a problem instance drawn from a suite $S(E, V; d, n, k)$. The HOMOTOPY algorithm, when applied to $(A, y)$, has the $k$-step solution property if and only if it has the correct term selection property and the sign agreement property.

**Proof:** To argue in the forward direction, we note that after $k$ steps, correct term selection implies that the active set is a subset of the support of $\alpha_0$, i.e., $I \subseteq \{1, \ldots, k\}$. In addition, the sign agreement property ensures that no variables leave the active set. Therefore, after $k$ steps, $I = \{1, \ldots, k\}$ and the HOMOTOPY algorithm recovers the correct sparsity pattern. To show that after $k$ steps, the algorithm terminates, we note that, at the $k$th step, the step-size $\gamma_k$ is chosen so that, for some $j \in I^c$,

$$c_k(j) = \gamma_k a_j^T A I d_k(I) = \lambda_k = \gamma_k$$  \hspace{1cm} (12)

with $\lambda_k = ||c_k(I)||_\infty$. In addition, for the $k$th update, we have

$$A I d_k(I) = A_I \left( A_I^T A_I \right)^{-1} A_I^T r_k = r_k$$

since $r_k$ is contained in the column space of $A_I$. Hence

$$c_k(I^c) = A_I^T A I d_k(I)$$

and $\gamma_k = \lambda_k$ is chosen to satisfy (12). Therefore, the solution at step $k$ has $A_I x_k = y$. Since $y$ has a unique representation in terms of the columns of $A_I$, we may conclude that $x_k = \alpha_0$.

The converse is straightforward. In order to terminate with the solution $\alpha_0$ after $k$ steps, the HOMOTOPY algorithm must select one term out of $\{1, \ldots, k\}$ at each step, never removing any elements. Thus, violation of either the correct term selection property or the sign agreement property would result in a number of steps greater than $k$ or an incorrect solution. \hfill \Box

Below, we will show that, when the solution $\alpha_0$ is sufficiently sparse, both these properties hold as the algorithm traces the solution path. Theorem 1 then follows naturally.

**A. Correct Term Selection**

We now show that, when sufficient sparsity is present, the HOMOTOPY solution path maintains the correct term selection property.

**Lemma 3:** Suppose that $y = A \alpha_0$ where $\alpha_0$ has only $k$ nonzeros, with $k$ satisfying

$$k \leq (\mu^{-1} + 1)/2.$$  \hspace{1cm} (13)

Assume that the residual at the $\ell$th step can be written as a linear combination of the first $k$ columns in $A$, i.e.,

$$r_\ell = \sum_{j=1}^k \beta(\ell) a_j.$$  \hspace{1cm} (14)

Then the next step of the HOMOTOPY algorithm selects an index from among the first $k$.

**Proof:** We will show that at the $\ell$th step

$$\max_{1 \leq i \leq k} |\langle r_\ell, a_i \rangle| > \max_{i \notin I} |\langle r_\ell, a_i \rangle|$$

and so at the end of the $\ell$th iteration, the active set is a subset of $\{1, \ldots, k\}$.

Let $G = A_I^T A_I$ denote the Gram matrix of $A$. Let $\hat{i} = \arg\max_{1 \leq i \leq k} |\beta(\ell)|$. The left-hand side of (14) is bounded below by

$$\max_{1 \leq i \leq k} |\langle r_\ell, a_i \rangle| \geq |\langle r_\ell, a_{\hat{i}} \rangle| \geq \left| \sum_{j=1}^k \beta(\ell) G_{i,j} \right|,$$

which can be further bounded as follows.

$$\geq |\beta(\hat{i})| - \sum_{j \neq \hat{i}} |G_{i,j}| |\beta(\ell)|$$

$$\geq |\beta(\hat{i})| - M(A) \sum_{j \neq \hat{i}} |\beta(j)|$$

$$\geq |\beta(\hat{i})| - M(A)(k - 1) |\beta(\hat{i})|,$$  \hspace{1cm} (15)

Here we used $||a_j||_2^2 = 1$ for all $j$ and $G_{i,j} \leq M(A)$ for $j \neq \hat{i}$.

As for the right-hand side of (14), we note that for $i > k$ we have

$$|\langle r_\ell, a_i \rangle| \leq \sum_{j=1}^k |\beta(\ell)| |G_{i,j}| \leq M(A) \sum_{j=1}^k |\beta(j)|$$

$$\leq k M(A) |\beta(\hat{i})|.$$  \hspace{1cm} (16)

Combining (15) and (16), we get that for (14) to hold, we need

$$1 - (k - 1) M(A) > k M(A).$$  \hspace{1cm} (17)

Since $k$ was selected to exactly satisfy this bound, relation (14) follows. \hfill \Box

Thus, when $k$ satisfies (13), the HOMOTOPY algorithm only ever considers indices among the first $k$ as candidates to enter the active set.

**B. Sign Agreement**

Recall that an index is removed from the active set at step $\ell$ only if condition (4) is violated, i.e., if the signs of the residual correlations $c_k(\ell)$, $i \in I$ do not match the signs of the corresponding elements of the solution $x_k$. The following lemma shows that, when $\alpha_0$ is sufficiently sparse, an even stronger result holds: at each stage of the algorithm, the residual correlations in the active set agree in sign with the direction of change of the corresponding terms in the solution. In other words, the solution moves in the right direction at each step. In particular, it implies that throughout the HOMOTOPY solution path, the sign agreement property is maintained.

**Lemma 4:** Suppose that $y = A \alpha_0$, where $\alpha_0$ has only $k$ nonzeros, with $k$ satisfying

$$k \leq (\mu^{-1} + 1)/2.$$
For \( k \in \{1, \ldots, k\} \), let \( c_k = A^T r_k \), and let the active set \( I \) be defined as in (6). Then the update direction \( d_\ell \) defined by (7) satisfies
\[
\text{sgn}(d_\ell(I)) = \text{sgn}(c_\ell(I)). 
\] (18)

**Proof:** Let \( \lambda_\ell = ||c_\ell||_\infty \). We will show that for \( i \in I \),
\[
|d_\ell(i) - \text{sgn}(c_\ell(i))| < 1,
\]
implying that (18) holds. To do so, we note that (17) can be rewritten as
\[
\lambda_\ell (A^T_\ell A_\ell - I_d) d_\ell(I) = -\lambda_\ell d_\ell(I) + c_\ell(I)
\]
where \( I_d \) is the identity matrix of appropriate dimension. This yields
\[
\begin{align*}
||\lambda_\ell d_\ell(I) - c_\ell(I)||_\infty \\
\leq ||A^T_\ell A_\ell - I_d||_{(\infty, \infty)} \cdot ||\lambda_\ell d_\ell(I)||_\infty \\
\leq \frac{1 - M(A)}{2} ||\lambda_\ell d_\ell(I)||_\infty \\
\leq \frac{1 - M(A)}{2} (||c_\ell(I)||_\infty + ||\lambda_\ell d_\ell(I) - c_\ell(I)||_\infty) \\
\leq \frac{1 - M(A)}{2} (\lambda_\ell + ||\lambda_\ell d_\ell(I) - c_\ell(I)||_\infty)
\end{align*}
\]
where \( || \cdot ||_{(\infty, \infty)} \) denotes the induced \( \ell_\infty \) operator norm. Rearranging terms, we get
\[
||\lambda_\ell d_\ell(I) - c_\ell(I)||_\infty \leq \lambda_\ell \cdot \frac{1 - M}{1 + M} < \lambda_\ell
\]
thus
\[
||d_\ell(I) - \text{sgn}(c_\ell(I))||_\infty < 1.
\]
Relation (18) follows. \( \square \)

**C. Proof of Theorem 1**

Lemmas 3 and 4 establish the correct term selection property and the sign agreement property at a single iteration of the algorithm. We will now give an inductive argument showing that the two properties hold at every step \( \ell \in \{1, \ldots, k\} \).

In detail, the algorithm starts with \( x_0 = 0 \); by the sparsity assumption on \( y = A_0 h_0 \), we may apply Lemma 3 with \( r_0 = y \), to get that at the first step, a column among the first \( k \) is selected. Moreover, at the end of the first step we have \( x_1 = \gamma_1 d_1 \), and so, by Lemma 4
\[
\text{sgn}(x_1(I)) = \text{sgn}(c_1(I)).
\]
By induction, assume that at step \( \ell \), only indices among the first \( k \) are in the active set, i.e.,
\[
r_\ell = \sum_{j=1}^{k} \beta_\ell(j) a_j,
\]
and the sign condition (4) holds, i.e.,
\[
\text{sgn}(x_\ell(I)) = \text{sgn}(c_\ell(I)).
\]
Applying Lemma 3 to \( r_\ell \), we get that at the \( (\ell + 1) \)th step, the term to enter the active set will be selected from among the first \( k \) indices. Moreover, for the updated solution we have
\[
\text{sgn}(x_\ell(I + 1)) = \text{sgn}(x_\ell(I) + \gamma_{\ell + 1} d_{\ell + 1}(I)).
\]
By Lemma 4, we have
\[
\text{sgn}(d_{\ell + 1}(I)) = \sigma_{\ell + 1}(I).
\]
We observe that
\[
c_{\ell + 1} = c_\ell - \gamma_{\ell + 1} A^T_\ell A_\ell d_{\ell + 1}
\]
whence, on the active set,
\[
|c_{\ell + 1}(I)| = |c_\ell(I)|(1 - \gamma_{\ell + 1})
\]
and so
\[
\sigma_{\ell + 1}(I) = \sigma_\ell(I).
\]
In words, once a vector enters the active set, its residual correlation maintains the same sign. We conclude that
\[
\text{sgn}(x_{\ell + 1}(I)) = \sigma_{\ell + 1}(I). \quad (19)
\]
Hence, no variables leave the active set. To summarize, both the correct term selection property and the sign agreement property are maintained throughout the execution of the HOMOTOPY algorithm.

We may now invoke Lemma 2 to conclude that the \( k \)-step solution property holds. This completes the proof of Theorem 1. \( \square \)

**V. Fast Solution With the Uniform Spherical Ensemble (USE)**

Suppose now that \( A \) is a random matrix drawn from the USE. It is not hard to show that such matrices \( A \) are naturally incoherent; in fact, for \( \epsilon > 0 \), with high probability for large \( n \)
\[
M(A) \leq \sqrt{\frac{4 \log(n)}{d}} \cdot (1 + \epsilon).
\]
Thus, applying Theorem 1, we get an immediate corollary that if \( k \) satisfies
\[
k \leq \sqrt{d/\log(n)} \cdot (1/4 - \epsilon') \quad (20)
\]
then HOMOTOPY is highly likely to recover any sparse vector \( h_0 \) with at most \( k \) nonzeros. However, as it turns out, HOMOTOPY operates much more efficiently when applied to instances from the random suite \( S(\text{USE}; d, n, k) \) than what is implied by incoherence. We now discuss evidence leading to Empirical Finding 1. We demonstrate, through a comprehensive suite of simulations, that the formula \( d/(2 \log(n)) \) accurately describes the breakdown of the \( k \)-step solution property. Before doing so, we introduce two important tools that will be used repeatedly in the empirical analysis below.
A. Phase Diagrams and Phase Transitions

In statistical physics, a phase transition is the transformation of a thermodynamic system from one phase to another. The distinguishing characteristic of a phase transition is an abrupt change in one or more of its physical properties, as underlying parameters cross a region of parameter space. Borrowing terminology, we say that property $\mathcal{P}$ exhibits a phase transition at a sequence of problem suites $S(E, V; d, n, k)$, if there is a threshold function $\text{THRESH}(d, n)$ on the parameter space $(k, d, n)$, such that problem instances with $k \leq \text{THRESH}(d, n)$ have property $\mathcal{P}$ with high probability, and problem instances above this threshold do not have property $\mathcal{P}$ with high probability. As we show later, the $k$-step solution property of HOMOTOPY, applied to problem instances drawn from $S(\text{USE}, d, n, k)$, exhibits a phase transition at around $d/(2 \log(n))$. Thus, there is a sequence $(\epsilon_n)$ with each term small and positive, so that for $k < (1 - \epsilon_n) d/(2 \log(n))$, HOMOTOPY delivers the sparsest solution in $k$ steps with high probability; on the other hand, if $k > (1 + \epsilon_n) d/(2 \log(n))$, then with high probability, HOMOTOPY fails to terminate in $k$ steps.

To visualize phase transitions, we utilize phase diagrams. In statistical physics, a phase diagram is a graphical depiction of a parameter space decorated by boundaries of regions where certain properties hold. Here we use this term in an analogous fashion, to mean a graphical depiction of a subset of the parameter space $(d, n, k)$, illustrating the region where an algorithm has a given property $\mathcal{P}$, when applied to a problem suite $S(E, V; d, n, k)$. It will typically take the form of a two-dimensional (2D) grid, with the threshold function associated with the corresponding phase transition dividing the grid into two regions: A region where property $\mathcal{P}$ occurs with high probability, and a region where $\mathcal{P}$ occurs with low probability.

While phase transitions are sometimes discovered by formal mathematical analysis, more commonly, the existence of phase transitions is unearthed by computer simulations. We now describe an objective framework for estimating phase transitions from simulation data. For each problem instance $(A, y)$ drawn from a suite $S(E, V; d, n, k)$, we associate a binary outcome variable $Y_{k,n}^{d}$, with $Y_{k,n}^{d} = 1$ when property $\mathcal{P}$ is satisfied on that realization, and $Y_{k,n}^{d} = 0$ otherwise. Within our framework, $Y_{k,n}^{d}$ is modeled as a Bernoulli random variable with probability $\pi_{k,n}$. Thus

$$P\left(Y_{k,n}^{d} = 1\right) = \pi_{k,n} \quad \text{and} \quad P\left(Y_{k,n}^{d} = 0\right) = 1 - \pi_{k,n}.$$ 

Our goal is to estimate a value $\hat{k}_{\delta,n}$ where the transition between these two states occurs. To do so, we employ a logistic regression model on the mean response $E\left\{Y_{k,n}^{d}\right\}$ with predictor variable $k$:

$$E\left(Y_{k,n}^{d}\right) = \frac{\exp(\beta_0 + \beta_k k)}{1 + \exp(\beta_0 + \beta_k k)}.$$

Logistic response functions are often used to model threshold phenomena in statistical data analysis [36]. Let $\hat{\pi}_{k,n}$ denote the value of the fitted response function. We may then compute a value $\hat{k}_{\eta,n}$ indicating that, with probability exceeding $1 - \eta$, a problem instance drawn from $S(E, V; d, n, k)$ with $k < \hat{k}_{\eta,n}$, property $\mathcal{P}$ holds. $\hat{k}$ is given by

$$\hat{\pi}_{k} = 1 - \eta.$$

Thus, computing $\hat{k}_{\eta,n}$ for a range of $(d, n)$ values essentially maps out an empirical phase transition of property $\mathcal{P}$. The value $\eta$ fixes a location on the transition band below which we define the region of success. In our work, we set $\eta = 0.25$.

B. The $k$-Step Solution Property

We now apply the method described in the previous section to estimate an empirical phase transition for the $k$-step solution property of the HOMOTOPY algorithm. Cursory empirical results about the $k$-step behavior of HOMOTOPY were also described in [24]. The experimental setup is as follows. For $(d, n, k)$, we generated a problem instance $(A, y)$ drawn from the problem suite $S(\text{USE}, \text{UNIFORM}, d, n, k)$. To this instance we applied the HOMOTOPY algorithm, and recorded the outcome of the response variable $Y_{k,n}^{d}$. We did so in two cases: first, fixing $d = 200$, varying $n$ through 40 log-equispaced points in $[200, 20000]$ and $k$ through 40 equispaced points in $[1, 40]$; second, keeping $n$ fixed at 1000, varying $d$ through 40 equispaced points in $[100, 1000]$ and $k$ through 40 equispaced points in $[2, 100]$. For each $(d, n, k)$ instance we ran 100 independent trials. To estimate the regression coefficients $\beta_0, \beta_1$ in (21), we used maximum-likelihood estimation [36].

Fig. 4 displays two phase diagrams, giving a succinct depiction of the simulation results. Fig. 4(a) displays a grid of $(k, n)$ values, $k \in [1, 40], n \in [200, 20000]$. Each point on this grid takes a value between 0 and 1, representing the ratio of successful outcomes to overall trials; these are precisely the observed $E\left\{Y_{k,n}^{d}\right\}$, on a grid of $(k, n)$ values, with $d$ fixed. Overlaid are curves for the estimated $\hat{k}_{\eta,n}$ for each $n$ with $\eta = 0.25$, and the theoretical bound $d/(2 \log(n))$. Fig. 4(b) has a similar configuration on a $(k, d)$ grid, with $n$ fixed at 1000. Careful inspection of these phase diagrams reveals that the estimated $\hat{k}_{\eta,n}$ closely follows the curve $d/(2 \log(n))$; Fig. 4(a) shows the logarithmic behavior of the phase transition as $n$ varies, and Fig. 4(b) shows its linear behavior as $d$ varies.

Table III offers a summary of the results of this experiment, for the three nonzero ensembles used in our simulations: UNIFORM, GAUSS, and BERNOULLI. In each case, two measures for statistical goodness-of-fit are reported: the mean squared error (MSE) and the $R^2$ statistic. The MSE measures the squared deviation of the estimated $k$ values from the proposed formula $d/(2 \log(n))$. The $R^2$ statistic measures how well the observed phase transition fits the theoretical model; a value close to 1 indicates a good fit. These statistical measures give another indication that the empirical behavior is in line with theoretical predictions. Moreover, they support the theoretical assertion that the $k$-step solution property of the HOMOTOPY algorithm is independent of the coefficient distribution.

The results in Fig. 4 and Table III all demonstrate consistency between the empirical phase transition and the bound $d/(2 \log(n))$, even for very modest problem sizes. We now turn to examine the sharpness of the phase transition as the problem dimensions increase. In other words, we ask whether for large
versus $\epsilon_n$ in (2) becomes smaller as $n$ increases. We note that the regression coefficient $\beta_1$ in (21) associated with the predictor variable $k$ dictates how sharp the transition from $\pi_k^{d,n} = 1$ to $\pi_k^{d,n} = 0$ is; a large negative value for $\beta_1$ implies a "step function" behavior. Hence, we expect $\beta_1$ to grow in magnitude as the problem size increases. This is indeed verified in Fig. 5(a) which displays the behavior of $\beta_1$ for increasing $n$, and $d$ fixed at 200. For illustration, Fig. 5(b) and (c) shows the observed mean response $\gamma_k^{d,n}$ versus $k$ for $n = 200$ with $\beta_1 \approx -0.35$, and $n = 20000$ with $\beta_1 \approx -1.1$, respectively. Clearly, the phase transition in panel (c) is much sharper.

C. Correct Term Selection and Sign Agreement

In Section IV, we have identified two fundamental properties of the HOMOTOPY solution path, namely, correct term selection and sign agreement, that constitute necessary and sufficient conditions for the $k$-step solution property to hold. We now present the results of a simulation study identifying phase transitions in these cases.

In detail, we repeated the experiment described in the previous section, generating problem instances from the suite $S(U, \text{UNIFORM}; d, n, k)$ and applying the HOMOTOPY algorithm. As the outcome of the response variable $\gamma_k^{d,n}$, we recorded first the property $\{ k \}$ The HOMOTOPY algorithm selected only terms from among $\{1, \ldots, k\}$ as entries to the active set $\{\}$, and next, the property $\{ k \}$ At each iteration of the HOMOTOPY algorithm, the signs of the residual correlations on the active set match the signs of the corresponding solution terms $\{\}$. Finally, we estimated phase transitions for each of these properties. Results are depicted in Fig. 6(a) and (b). Fig. 6(a) displays a phase diagram for the correct term selection property, and Fig. 6(b) has a similar phase diagram for the sign agreement property.

The results are quite instructive. The phase transition for the correct term selection property agrees well with the formula $d/(2 \log(n))$. Rather surprisingly, the phase transition for the sign agreement property seems to be at a significantly higher threshold, $d/(\sqrt{2} \cdot \log(n))$. 

Fig. 4. $k$-step solution property of HOMOTOPY for the problem suite $S(U, \text{UNIFORM}; d, n, k)$. Shaded attribute is the proportion of successful terminations, out of 100 trials, in the case: (a) $k = 1, \ldots, 40, d = 200$, and $n = 200, \ldots, 20000$; (b) $k = 2, \ldots, 100, d = 100, \ldots, 1000$, and $n = 1000$. Overlaid are curves for $d/(2 \log(n))$ (solid), $k_{\pi_0^{d,n}}^{d,n}$ (dashed), and the confidence bound for $k_{\pi_0^{d,n}}^{d,n}$ with 95% confidence level (dotted).

![Image](image-url)

Fig. 5. Sharpness of phase transition. The top panel shows the behavior of the regression coefficient $\beta_1$ vs. $k$. The bottom panels show two instances of the fitted transition model, for (b) $d = 200$, $n = 200$ and (c) $d = 200$, $n = 20000$.
Consequently, the “weak link” for the HOMOTOPY $k$-step solution property, i.e., the attribute that dictates the sparsity bound for the $k$-step solution property to hold, is the addition of incorrect terms into the active set. We interpret this finding as saying that the HOMOTOPY algorithm would typically err first “on the safe side,” adding terms off the support of $\Theta_0$ into the active set (a.k.a. false discoveries), before removing correct terms from the active set (a.k.a. missed detections).

D. Random Signs Ensemble (RSE)

Matrices in the RSE are constrained to have elements of constant amplitude, with signs independently drawn from an equiprobable Bernoulli distribution. This ensemble is known to be effective in various geometric problems associated with underdetermined systems, through work of Kashin [37], followed by Garnaev and Gluskin [30]. Previous work [53]–[55] gave theoretical and empirical evidence to the fact that many results known to be important for various extremal problems in experimental design [35].

To show this, we conducted a series of simulations, parallelizing the study described in Section V-B, with the problem suite $S(\text{RSE}; d, n, k)$ replacing $S(\text{USE}; d, n, k)$. The results of the simulation are summarized in Table IV, which has MSE and $R^2$ measures for the deviation of the empirical phase transition from the formula $d/(2 \log(n))$, for different coefficient ensembles. The results indeed suggest that the HOMOTOPY algorithm has the $k$-step solution property when applied to the RSE ensemble $S(\text{RSE}; d, n, k)$, with $k < d/(2 \log(n))$. An important implication of this conclusion is that in practice, we may use matrices drawn from the RSE to replace matrices from the USE without loss of performance. This may result in reduced complexity and increased efficiency, as matrices composed of random signs can be generated and applied much more rapidly than general matrices composed of real numbers.

VI. Fast Solution With Partial Orthogonal Ensembles

We now turn our attention to a class of matrices composed of partial orthogonal matrices, i.e., matrices constructed by sampling a subset of the rows of an orthogonal matrix. In particular, we will be interested in the following three ensembles.

• Partial Fourier Ensemble (PFE). Matrices in this ensemble are obtained by taking an $n_t$ by $n$ Fourier matrix and sampling $d$ rows at random. The PFE is of utmost importance in medical imaging and spectroscopy, as it can be used to model reconstruction of image data from partial information in the frequency domain [7], [6], [38], [39].

• Partial Hadamard Ensemble (PHE). Matrices in this ensemble are obtained by taking an $n_t$ by $n$ Hadamard matrix and sampling $d$ of its rows at random. This ensemble is known to be important for various extremal problems in experimental design [35].

• Uniform Random Projections (URPE). Matrices in this ensemble are obtained by taking an $n_t$ by $n$ random orthogonal matrix and sampling $d$ of its rows at random. It serves as a general model for the PFE and PHE.

As we now demonstrate, the $k$-step solution property of HOMOTOPY holds at a significantly greater range of signal sparsities than observed in Section V.

We first consider the PFE and PHE. Partial Fourier matrices and partial Hadamard matrices have drawn considerable attention in the sparse approximation community [7], [6], [55], [54], [31]. We mention two incentives.
Fig. 7. $k$-step solution property of HOMOTOPY for the suite $S$($PFE$, $V$: $d$, $n$, $k$). Each phase diagram presents success rates on a $\rho = \delta$ grid, with $\rho = k/d$ in $[0.05, 1]$, $\delta = d/n$ in $[0.1, 1]$, and $n = 1024$, for: (a) $V = U_{\text{NORMAL}}$; (b) $V = G_{\text{ABEISS}}$; (c) $V = B\text{ERNOULLI}$. Overlaid are curves for $\tilde{f}_{0.25}(\delta)$ (dash-dotted), with the confidence bounds for 95% confidence level (dotted), and $\rho = 1/(2\log(n))$ (solid). The range of sparsities in which the $k$-step solution property holds is much more extensive than implied by the bound $k < d/(2\log(n))$.

- **Modeling of Physical Phenomena.** Fourier operators play a key role in numerous engineering applications modeling physical phenomena. For instance, acquisition of two- and three-dimensional magnetic resonance images is modeled as applying a Fourier operator to spatial data. Thus, application of a partial Fourier operator would correspond to acquisition of partial frequency data. Likewise, the Hadamard operator, albeit not as common, is often used in Hadamard transform spectroscopy.

- **Fast Implementations.** Both the Fourier transform and the Hadamard transform have special structure allowing for rapid computation. They can be computed by algorithms which run much faster than the naïve implementation from definition. In detail, the fast Fourier transform (FFT) can be applied to an $n$-vector using $O(n\log(n))$ operations, rather than the $n^2$ operations needed by direct computation [3]. Similarly, a fast Hadamard transform (FHT) exists, which applies a Hadamard operator using $O(n\log(n))$ operations. With these fast algorithms, rapid computation of partial Fourier or partial Hadamard products is straightforward. Specifically, note that we may write a $d$ by $n$ matrix $A$ from the PFE as $A = R \cdot F$, with $F$ an $n$ by $n$ Fourier matrix, and $R$ a $d$ by $n$ random sampling matrix. Translating to linear operations, application of $A$ amounts to applying $F$, using the FFT, followed by sampling $d$ entries of the resulting vector. Similarly for partial Hadamard matrices.

The experimental setup leading to Empirical Finding 2 utilizes a more extensive phase diagram. Expressly, we fix $n$ and generate a grid of points indexed by variables $\delta = d/n$ and $\rho = k/d$, with $\delta$ in $(0, 1]$ and $\rho$ in $(0, 1]$. Thus, at a fixed $\delta, d = \lfloor \delta n \rfloor$, and $k = \lfloor \rho d \rfloor$ ranges between 1 and $d$. In a sense, this $\rho-\delta$
grid gives a complete picture of all sparse underdetermined settings of interest. Data collection and estimation of an empirical phase transition were done in the same manner described in Section V-B. Figs. 7 and 8 summarize the simulation results, each depicting three phase diagrams, for three different nonzero distributions.

The results are indeed surprising; for problems from the suite $\mathcal{S}(\text{PFE}; d, n, k)$, HOMOTOPY maintains the $k$-step solution property at a much broader range of sparsities $k$ than observed for problems from $\mathcal{S}(\text{USE}; d, n, k)$, particularly at high values of the indeterminacy ratio $\delta$. For instance, at $\delta = 3/4$, the empirical results indicate that when the nonzeros are uniformly distributed, the empirical phase transition is at $k = \hat{\rho} d \approx 174$, whereas the formula $d/(2 \log(n)) \approx 55$, implying that the range of sparsities for which HOMOTOPY terminates in $k$ steps has grown threefold.

We conducted parallel simulation studies for the PHE and the URPE, and we summarize the empirical findings in Fig. 9.

It displays the empirical phase transitions $\hat{\rho}(\delta)$ for the three partial orthogonal ensembles with different nonzero distributions. Careful inspection of the curves reveals that the $k$-step property of HOMOTOPY holds at roughly the same region of $(\delta, \rho)$ space for all combinations of matrix ensembles and coefficient distributions considered, with mild variations in the location and shape of the phase transition for different nonzero distributions.

The empirical phase transition $\hat{\rho}(\delta)$ does not yield itself to a simple representation in functional form. It does, however, have several important characteristics, which we now highlight.

- First, notice that at the right end of the phase diagram, we have $\hat{\rho}(\delta = 1) = 1$. In other words, when the matrix $A$ is square, HOMOTOPY always recovers the sparsest solution in $k$ steps. This is hardly surprising, since, at $\delta = 1$, the matrix $A$ is orthogonal, and the data $y$ is orthogonal to columns of $A$ off the support of $\alpha_0$, ensuring correct term selection. In addition, the update step $d(I)$ in (7) is equal
Fig. 9. Estimated phase transition curves. Each panel displays empirical phase transition curves for the PFE (solid), PHE (dashed), URPE (dash-dotted), with (a) uniform nonzero distribution; (b) Gaussian nonzero distribution; (c) Bernoulli nonzero distribution.

to $\text{sgn}(c_y(I))$, implying that the sign agreement property holds. Thus, $k$-step termination is guaranteed.

- Second, at the left end of the phase diagram, as $\delta \to 0$, we may rely on recent theoretical developments to gauge the behavior of the phase transition curve. In their work on randomly projected polytopes, Donoho and Tanner showed that an upper bound for the value $\rho(\delta)$ below which $\ell_1$ minimization recovers the sparsest solution with high probability, asymptotically behaves like $1/(2\log(1/\delta))$ as $\delta \to 0$ [22]. In particular, this same upper bound holds in our case, as failure to recover the sparsest solution necessarily implies failure of the $k$-step property.

- Finally, and most importantly, we note that the $k$-step phase transition for the partial orthogonal ensembles is stable with increasing $n$, in the following sense. For $\delta \in (0,1]$ fixed and $n$ sufficiently large, the range of $\rho$ in which the $k$-step property holds does not change with $n$. Evidence to the stability of the phase transition is given in Fig. 8, which demonstrates that for $\delta$ fixed at 0.5, the phase transition occurs at an approximately constant value of $\rho$. The implication of this property is that the empirical phase transitions computed here may be used to estimate the $k$-step behavior of HOMOTOPY for problem instances of any size.

VII. BRIDGING $\ell_1$ MINIMIZATION AND OMP

In the Introduction, we alluded to the fact that there is undeniable parallelism in results about the ability of $\ell_1$ minimization and OMP to recover sparse solutions to systems of underdetermined linear equations. Indeed, earlier reports [9], [19], [15], [51]–[53] documented instances where, under similar conditions, both $\ell_1$ minimization and OMP recover the sparsest solution. Yet, these works did not offer any insights as to why the two seemingly disparate techniques should offer similar performance in such cases. This section develops a linkage between $\ell_1$ minimization and OMP.
As a first step, we clear up some confusion in terminology. Previous work studying the performance of OMP set conditions on the sparsity $k$ under which the algorithm “correctly recovers the sparest solution” [19], [51], [53]. Yet, in effect, what was proven is that OMP “selects a correct term at each iteration, and terminates with the correct solution after $k$ iterations”; we now recognize this as the $k$-step solution property. We note that the $k$-step solution property is not, in general, a necessary condition for OMP to correctly recover the sparsest solution. Indeed, as long as $k \leq d$ and the active set $I$ contains all the entries in $\text{supp}(\alpha_0)$, the active set matrix $A_I$ has full column rank, hence, the solution of the least-squares system $(A_I^T A_I)^{-1} A_I^T y$ is unique and equal to $\alpha_0$. However, the $k$-step solution property is clearly a sufficient condition for recovery of the sparsest solution. Its usefulness lies in the simplicity of $k$-step solutions; when the $k$-step property holds, the solution path is the simplest possible, consisting of $k$ correct steps. Thus, it yields itself naturally to focused analysis.

In particular, the $k$-step solution property is useful in explaining the parallelism between $\ell_1$ minimization and OMP. As depicted schematically in Fig. 1, when the $k$-step solution property holds, $\ell_1$ minimization can be transformed into OMP through a series of modifications. We elaborate on each of these steps in Sections VII-A–C.

### A. $\ell_1$ Minimization $\longrightarrow$ Homotopy

For the phrase “algorithm A has the $k$-step solution property” to make sense, the algorithm must have a stepwise structure, building a solution approximation term by term. This is where the HOMOTOPY algorithm fits in, bridging the high-level notion of $\ell_1$ minimization with the lower level stepwise structure of OMP. Earlier work [26], [41] has shown that HOMOTOPY is a correct algorithm for solving the $\ell_1$ minimization problem (P1).

### B. Homotopy $\longrightarrow$ LARS

As noted earlier, the LARS procedure is a simplification of the HOMOTOPY algorithm, achieved by removing the condition for sign agreement of the current solution and the residual correlations. This brings us one step closer to OMP; while HOMOTOPY allows for removal of terms from the active set, both LARS and OMP insert terms, one by one, into the active set, never removing any active elements.

Interestingly, when the $k$-step property holds, this simplification changes nothing in the resulting solution. To see that, recall that HOMOTOPY has the $k$-step solution property if and only if it has the correct term selection property and the sign agreement property. Correct term selection is all that is needed by LARS to reach the solution in $k$ steps; after $k$ correct terms are inserted into the active set, the algorithm would terminate with zero residual. Therefore, the assertion of Theorem 1 holds for LARS as well.

**Corollary 2:** Let $(A, y)$ be a problem instance drawn from $S(\text{NCG}, d, n, k)$, with $k \leq (\mu^{-1} + 1)/2$. Then, LARS runs $k$ steps and stops, delivering the solution $\alpha_0$.

Similar conclusions can be made about the assertions of Empirical Findings 1 and 2.

### C. LARS $\longrightarrow$ OMP

We already hinted at the fact that HOMOTOPY and OMP share an inherent affinity. We now demonstrate that the two methods are based on the same fundamental procedure; solving a sequence of least-squares problems on an increasingly large subspace, defined by the active columns of $A$.

More precisely, assume we are at the $\ell$th iteration, with a solution estimate $\hat{x}_{\ell-1}$ and an active set $I$ consisting of $\ell-1$ terms. Recall the procedure underlying OMP: It appends a new term to the active set by selecting the vector $\alpha_j$ that maximizes the absolute residual correlation: $j = \arg \max_{j \in I} |a_j^T (y - A \hat{x}_{\ell-1})|$, and $I := I \cup \{j\}$. It then projects $y$ onto $\mathcal{R}(A_I)$ by solving

$$A_I^T A_I x_{\ell}(I) = A_I^T y$$

thus guaranteeing that the residual is orthogonal to the subspace $\mathcal{R}(A_I)$. Once $y$ is spanned by the active columns, the algorithm terminates with zero residual.

While LARS was derived from a seemingly disparate perspective, it in fact follows the same basic procedure, replacing (22) with a penalized least-squares problem. Recalling the discussion in Section II, we note that LARS also selects a new term according to the maximum correlation criterion, and then solves

$$A_I^T A_I x_{\ell}(I) = A_I^T y - \lambda_{\ell} \cdot s$$

where $s$ is a vector of length $\ell$, recording the signs of residual correlations of each term at the point it entered the active set, and $\lambda_{\ell}$ is the correlation magnitude at the breakpoint on the LARS path. Indeed, (22) and (23) are identical if not for the penalization term on the right-hand side of (23).

In previous sections, we presented theoretical and empirical evidence demonstrating that in particular cases, this modification does not have an effect on the $k$-step solution property of the procedure. We offer two examples. First, for the incoherent problem suite $S(\text{INC}, d, n, k)$: the statement of Theorem 1 holds for OMP as well (see, e.g., [51, Corollary 3.6]). Second, for the random problem suite $S(\text{USE}, d, n, k)$, Tropp et al. proved results which can now be reinterpreted as saying that, when $k \leq d/((\log(n)))$, then with high probability for large $n$, OMP has the $k$-step solution property, paralleling the statement of Empirical Finding 1. We conjecture, in fact, that the bound $k \leq d/((2\log(n)))$ holds for OMP as well. Finally, we note that connections between the $k$-step property of LARS and OMP in other settings have not been explored here, yet the suggestive evidence presented in this paper indicates that this may be an exciting direction for future research.

To summarize, we exhibited a series of transformations, starting with $\ell_1$ minimization, and ending with greedy pursuit. Each transformation is characterized clearly, and maintains the $k$-step property of the solution path. We believe this sequence clarifies initially surprising similarities between results about $\ell_1$ minimization and OMP.

### VIII. HOMOTOPY AND PFP

PFP was introduced as a greedy algorithm to solve (P1) in the dual space. As pointed out in [43], [44], although the algorithm...
is derived from a seemingly viewpoint, it exhibits interesting similarities to HOMOTOPY. To study these in more detail, we first briefly describe the algorithm.

Define the standard-form linear program equivalent to \((P_1)\)

\[
(P_1) \quad \min \mathbf{1}^T \mathbf{x} \quad \text{subject to} \quad \mathbf{y} = \overline{\mathbf{A}} \mathbf{x}, \quad \mathbf{x} \geq 0
\]

where \(\overline{\mathbf{A}} = [\mathbf{A} - \mathbf{A}].\) The equivalence of \((P_1)\) and \((P_1)\) is well established; the solutions \(\overline{\mathbf{x}}\) of \((P_1)\) and \(\mathbf{x}\) of \((P_1)\) have the correspondence \(\overline{\mathbf{x}}(j) = \max\{x(j), 0\}\) for \(1 \leq j \leq n\), and \(\overline{\mathbf{x}}(j) = \max\{-x(j - n), 0\}\) for \(n + 1 \leq j \leq 2n\). The Lagrangian dual of the linear program \((P_1)\) is given by

\[
(D_1) \quad \max \mathbf{y}^T \mathbf{v} \quad \text{subject to} \quad \overline{\mathbf{A}}^T \mathbf{v} \leq 1.
\]

By the strong duality theorem, an optimal solution \(\overline{\mathbf{x}}\) of the primal problem \((P_1)\) is associated with an optimal solution \(\mathbf{v}\) to the dual problem \((D_1)\), such that \(\mathbf{1}^T \overline{\mathbf{x}} = \mathbf{y}^T \mathbf{v}\). PFP operates in the dual space, tracing a solution to the dual problem \((D_1)\), which then yields a solution to the primal problem \((P_1)\) (or, equivalently, \((P_1)\)). To do so, the algorithm maintains an active set \(I\) of nonzero coordinates, and carries out a familiar procedure at each iteration: Selection of a new term to enter the active set, removal of violating terms from the active set (if any), and computation of a new solution estimate. Specifically, at the \(\ell\)th stage, PFP first selects a new term to enter the active set, via the criterion

\[
\ell = \arg \max_{\mathbf{v} \neq \mathbf{0}} \left\{ \frac{\pi_{\ell}^T r_{\ell}}{1 - \pi_{\ell}^T v_{\ell}} \mid \pi_{\ell}^T r_{\ell} > 0 \right\}
\]

(24)

with \(r_{\ell} = \mathbf{y} - \overline{\mathbf{A}} \overline{\mathbf{x}}_{\ell}\) the current residual. It then updates the active set \(I \leftarrow I \cup \{\pi_{\ell}\}\), and computes a primal solution estimate, the projection of \(\mathbf{y}\) onto the subspace spanned by the active vectors

\[
\overline{\mathbf{x}}_{\ell+1} = (\overline{\mathbf{A}}^T \overline{\mathbf{A}})^{-1} \overline{\mathbf{A}}^T \mathbf{y}
\]

(25)

and a dual solution estimate

\[
v_{\ell+1} = \overline{\mathbf{A}} (\overline{\mathbf{A}}^T \overline{\mathbf{A}})^{-1} \mathbf{1}.
\]

(26)
Finally, the algorithm verifies that all the active coefficients $\bar{x}_{t+1}(1)$ are nonnegative; coordinates with negative coefficients are removed from the active set.

The affinities between HOMOTOPY and PFP are not merely cosmetic. In fact, the two algorithms, derived from seemingly distinct viewpoints, carry out nearly identical procedures. To highlight these similarities, assume that both algorithms are applied to the standard form linear program $P_1$. We begin by examining the criteria for selection of a new term to enter the active set. In particular, assume we are at the $t$th stage of the HOMOTOPY algorithm. Recalling the discussion in Section II above, the HOMOTOPY term selection criterion (when applied to $P_1$) is

$$i^+ = \arg \min_{i \in I^c} \left\{ \frac{\lambda - c y_i(\hat{y})}{1 - \bar{a}_i^T v_y} \right\}$$

where $v_y = \bar{A}_I d(I) - \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} \mathbf{1}$ as in (26) above, and $\min^+$ indicates that the minimum is taken over positive arguments. Comparing (27) with (24), we note that the main difference lies in the residual; since PFP projects the data $y$ onto the space spanned by the active vectors, denoted $\mathcal{R}_I$, its residual is necessarily orthogonal to $\mathcal{R}_I$. HOMOTOPY, in contrast, does not enforce orthogonality, and its residual can be expressed as the sum of a component in $\mathcal{R}_I$ and a component in the orthogonal complement $\mathcal{R}_I^\perp$. Formally

$$r_t = P_I r_t + r_{t}^\perp$$

where $P_I = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} \bar{A}_I^T$ is the orthogonal projector onto $\mathcal{R}_I$, and $r_{t}^\perp = (I - P_I) r_{t}$ is the component of $r_t$ in $\mathcal{R}_I^\perp$. The component of $r_t$ in $\mathcal{R}_I$ obeys

$$P_I r_t = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} c_t = \lambda v_y.$$

Plugging this into (27), we get

$$i^+ = \arg \min_{i \in I^c} \left\{ \frac{\lambda - \bar{a}_i^T r_{t}^\perp + \lambda v_y}{1 - \bar{a}_i^T v_y} \right\}$$

$$= \arg \min_{i \in I^c} \left\{ \lambda - \bar{a}_i^T r_{t}^\perp \left( \frac{1}{1 - \bar{a}_i^T v_y} \right) \right\}$$

$$= \arg \max_{i \in I^c} \left\{ \bar{a}_i^T r_{t}^\perp \left( \frac{\bar{a}_i^T r_{t}^\perp}{1 - \bar{a}_i^T v_y} \right) > 0 \right\}$$

where the last equality holds because $\bar{A}_I^T v_y < 1$ throughout the HOMOTOPY solution path. Thus, conditional on $r_{t}^{\text{PFP}} = r_{t}^\perp$, the HOMOTOPY selection criterion is equivalent to PFP’s selection criterion. Now, since $r_{t}^\perp = y - P_I y$ as long as no elements are removed from the active set, $r_{t}^{\text{PFP}} = r_{t}^\perp$ by definition. In summary, the discussion above establishes the following.

**Corollary 3:** As long as no elements are removed from the active set, the solution path of PFP is identical to the HOMOTOPY solution path. In particular, with the sign condition removed, PFP is equivalent to LARS.

This finding is interesting in light of the earlier discussion; it implies that when the $k$-step solution property holds for HOMOTOPY, it also holds for PFP. In particular, the statement of Theorem 1 holds for PFP as well. Formally, we have the following.

**Corollary 4:** Let $(A, y)$ be a problem instance drawn from $\mathcal{S}(\text{INC}_p; d, n, k)$, with $k \leq (p^{-1} + 1)/2$. Then, PFP runs $k$ steps and stops, delivering the solution $\hat{x}_0$.

Similarly, Empirical Findings 1 and 2 hold for PFP as well. Fig. 10 displays phase diagrams for the $k$-step solution property (Fig. 10(a)), correct term selection property (Fig. 10(b)), and sign agreement property (Fig. 10(c)), when PFP is applied to instances of the problem suite $\mathcal{S}(\text{USE}, \text{UNIFORM}; d, n, k)$. The three properties exhibit clear phase transition at around $d/(2 \log(n))$. Analogously, the statements in Empirical Findings 2 can be verified to hold for PFP as well; we do not bring such evidence here for brevity of space.

**IX. RECOVERY OF SPARSE SOLUTIONS**

Previous sections examined various properties of HOMOTOPY, all centered around the $k$-step solution property. The motivation is clear; the HOMOTOPY algorithm is at peak efficiency when it satisfies the $k$-step solution property. Yet, we noted that failure to terminate in $k$ steps need not imply failure to recover the sparsest solution. We now turn to study this latter property in more detail.

**Definition 5:** An algorithm $\mathcal{A}$ has the $k$-sparse solution property at a given problem instance $(A, y)$ drawn from $\mathcal{S}(\text{USE}; V; d, n, k)$ if, when applied to the data $(A, y)$, the algorithm terminates with the solution $\hat{x}_0$.

Earlier papers used a different term for the same concept [18], [15], [54]; they would use the term $l_0$ equivalence to indicate recovery of the sparsest solution, i.e., equivalence to the solution of the combinatorial optimization problem

$$(P_0) \min_{x} ||x||_0 \text{ subject to } y = Ax.$$  

For the HOMOTOPY algorithm, the $k$-sparse solution property has been well-studied. Indeed, since the HOMOTOPY algorithm is guaranteed to solve $(P_0)$, it will have the $k$-sparse solution property whenever the solutions to $(P_0)$ and $(P_1)$ coincide. In a series of papers [15], [13], [17], Donoho has shown that for problem instances from the suite $\mathcal{S}(\text{USE}; d, n, k)$, the equivalence between solutions of $(P_0)$ and $(P_1)$ exhibits a clear phase transition, denoted $\rho_N$, that can be computed numerically. This phase transition provides a theoretical prediction for the conditions under which HOMOTOPY satisfies the $k$-sparse solution property. Yet, at present, no such predictions exist for LARS, OMP, or PFP.

We now present the results of an extensive simulation study, examining the validity of the $k$-sparse solution property for HOMOTOPY, LARS, OMP, and PFP, comparing the performance of these four closely related algorithms. We measure the $k$-sparse solution property for the three algorithms, closely paralleling the experimental framework described in earlier sections. The resulting phase diagrams appear in Fig. 11. Parallel results summarizing a similar experiment done for the URPE are presented in Fig. 12. Preliminary inspection of these phase diagrams reveals that the suggestive similarities between the three algorithms indeed translate into comparable performance in recovering the sparsest solution. More careful examination commands several important observations.
• **LARS approximates** HOMOTOPY. Throughout all the displayed phase diagrams, the empirical phase transition of LARS closely follows the transition of HOMOTOPY. In other words, almost anytime $\ell_1$ minimization successfully recovers the sparsest solution, LARS succeeds in finding it as well. This is a rather surprising find, as, at least in the USE case, for $k > d/(2\log(n))$, LARS is bound to err and add elements not in the support of $\Omega_0$ into its active set. The evidence we present here suggests that inside the equivalence phase, LARS still correctly recovers the support of $\Omega_0$ and obtains the sparsest solution.

• **Dependence on the Distribution of Nonzeros.** Examining Fig. 11, we note that for the suite $S$ (USE, $V: d, n, k$), the performance of HOMOTOPY and LARS is independent of the underlying distribution of the nonzero coefficients. This is not true for OMP, whose performance is affected by the underlying distribution. In particular, OMP’s worst performance is recorded when the nonzeros have constant amplitude and a random sign pattern; this has been noted previously in the analysis of OMP [15], [51].

• **Better Recovery with Partial Orthogonal Ensembles.** Comparison of Fig. 12 with Fig. 11 reveals that, in general, all four algorithms perform better when applied to problem instances from the suite $S$ (URPE, $d, n, k$). In particular, HOMOTOPY and LARS show a marked increase in performance when applied to the suite $S$ (URPE, UNIFORM, $d, n, k$). In our experiments, we observed similar phenomena when applying the three algorithms to problems drawn from the suites $S$ (PFE, $d, n, k$) and $S$ (PHE, $d, n, k$) as well. This is indeed an important finding, as these suites find extensive uses in applications.

X. **ON APPROXIMATE $\ell_1$ MINIMIZATION ALGORITHMS**

In the belief that true $\ell_1$ minimization is far too slow for large-scale applications, numerous authors have explored approximate algorithms based on iterative thresholding or similar
Fig. 12. The $k$-sparse solution property for the suite $S(URPE, V; d, n, k)$. Each panel has plots of the empirical phase transition curves of HOMOTOPY (dashed), LARS (dash-dotted), OMP (dotted), and PFP (solid) alongside the theoretical curve $\rho_W$, on a $p - \delta$ grid, with $n = 1000$, for: (a) $V = \text{UNIFORM}$; (b) $V = \text{GAUSS}$; (c) $V = \text{BERNOULI}$. 

Ideas; examples include [23], [11], [47], [48], [27]; related ideas include projection onto convex sets [6]. The reader should keep in mind that there are applications where such methods indeed operate faster than the methods discussed here, while still producing a faithful approximate solution; see, for example, [23, Sec. 9]. On the other hand, the present methods for $\ell_1$ minimization are much faster than general-purpose optimizers, so much of the apparent advantage claimed for approximate $\ell_1$ schemes has been neutralized by the algorithms studied here. Furthermore, approximate methods inevitably trade off aptitude for speed; in general, the range of sparsities in which such methods reliably recover the sparsest solution is considerably limited, compared to $\ell_1$ minimization. A further advantage of the HOMOTOPY viewpoint is the stimulus it provides for future research: the empirical phenomena identified here may lead to deep research in polytope theory, perhaps comparable to the kinds of results in [22].

XI. Conclusion

In this paper, we study the HOMOTOPY algorithm, introduced by Osborne et al. [41] and Efron et al. [26]. It was originally proposed in the context of overdetermined noisy systems. We propose here to use HOMOTOPY to obtain sparse solutions of underdetermined systems of linear equations in a noise-free setting. HOMOTOPY follows a piecewise linear solution path, starting from the zero solution, and terminating at the solution of $P_1$. It maintains an active set of column vectors of the matrix $A$, adding or removing a vector at each step, and updating the solution accordingly. It is a particularly effective algorithm for solving $\ell_1$ minimization problems when the underlying solution is sparse, and the underlying system is incoherent or random. In such scenarios, it significantly outperforms state-of-the-art general LP solvers. Moreover, its computational complexity is then comparable to greedy stepwise algorithms such as OMP.

Such claims we made concrete by showing that for two matrix classes, deterministic incoherent matrices and random uniform spherical matrices, HOMOTOPY has the following $k$-step solution property: If the data admit sparse representation using $k$ columns of $A$, HOMOTOPY finds it in exactly $k$ steps. We proved that for problems drawn from the suite $S(INC_H; d, n, k)$, HOMOTOPY has the $k$-step solution property whenever $k < (\mu^{-1} + 1)/2$. We also observed empirically that
for instances of the problem suite \(\mathcal{S}(\text{USE}; d, n, k)\), homotopy has the \(k\)-step solution property with high probability when \(k\) is less than a threshold, empirically \(k < d/(2 \log(n)) \cdot (1 - \epsilon_n)\).

Using the notions of phase diagrams and phase transitions, we presented empirical evidence supporting these assertions. We showed that for problem instances drawn from \(\mathcal{S}(\text{USE}; d, n, k)\), the \(k\)-step solution property of homotopy exhibits an empirical phase transition at \(d/(2 \log(n))\), regardless of the distribution of the nonzeros in the solution. Simulations with the suite \(\mathcal{S}(\text{RSE}; d, n, k)\) resulted in similar findings. For the PFE and PHE, we noted that homotopy has the \(k\)-step solution property at a much wider range of sparsities, compared with the USE. Thus, in many applications involving PFE and PHE matrices, homotopy would be particularly efficient in finding the sparsest solution. Similar conclusions were drawn for the problem suite \(\mathcal{S}(\text{URPE}; d, n, k)\).

The \(k\)-step analysis we presented sheds light on the relationship between \(f_1\) minimization and OMP. We argued that the two approaches can be bridged through a series of relaxations, each step maintaining the \(k\)-step property. In addition, we demonstrated that, conditional on the sign agreement property, PFP and LARS are algebraically equivalent. These insights illuminated the parallelism evident in the performance of homotopy, LARS, OMP, and PFP in recovering sparse solutions to underdetermined linear systems.

### XII. Accompanying Software

SparseLab is a library of Matlab functions we make available on the web at http://sparselab.stanford.edu. It has been developed in the spirit of reproducible research [4], [10], namely, the idea that publication of computational research demands also the publication of the complete software environment needed for reproducing that research. SparseLab has been used by the authors to create the figures and tables used in this paper, and the toolbox contains scripts which will reproduce all the calculations of this paper. The current release includes about 400 Matlab files, datasets, and demonstration scripts.

In addition, we note that Efron et al. have made available a software package in R and Splus that includes an implementation of the LARS algorithm, as well as the LASSO variant. The package is available on the web, see [25]. It also includes the data used in the examples appearing in [26].

### Acknowledgment

Yaakov Tsaig would like to thank Michael Saunders for helpful discussions and references.

### References

[34] R. Gribonval and M. Nielsen, “Highly sparse representations from dictionaries are unique and independent of the sparseness measure,” 2006, to be published.