



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Matrix-free interior point method for compressed sensing problems

Citation for published version:

Fountoulakis, K, Gondzio, J & Zhlobich, P 2014, 'Matrix-free interior point method for compressed sensing problems', *Mathematical Programming Computation*, vol. 6, no. 1, pp. 1-31. <https://doi.org/10.1007/s12532-013-0063-6>

Digital Object Identifier (DOI):

[10.1007/s12532-013-0063-6](https://doi.org/10.1007/s12532-013-0063-6)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Mathematical Programming Computation

Publisher Rights Statement:

The final publication is available at Springer via <http://dx.doi.org/10.1007/s12532-013-0063-6>.

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Matrix-free Interior Point Method for Compressed Sensing Problems

Kimón Fountoulakis · Jacek Gondzio* ·
Pavel Zhlobich

Technical Report ERGO 12-006

Abstract We consider a class of optimization problems for sparse signal reconstruction which arise in the field of Compressed Sensing (CS). A plethora of approaches and solvers exist for such problems, for example GPSR, FPC_AS, SPGL1, NestA, ℓ_1 - ℓ_s , PDCO to mention a few.

CS applications lead to very well conditioned optimization problems and therefore can be solved easily by simple first-order methods. Interior point methods (IPMs) rely on the Newton method hence they use the second-order information. They have numerous advantageous features and one clear drawback: being the second-order approach they need to solve linear equations and this operation has (in the general dense case) an $\mathcal{O}(n^3)$ computational complexity. Attempts have been made to specialize IPMs to sparse reconstruction problems and they have led to interesting developments implemented in ℓ_1 - ℓ_s and PDCO softwares. We go a few steps further. First, we use the matrix-free IPM, an approach which redesigns IPM to avoid the need to explicitly formulate (and store) the Newton equation systems. Secondly, we exploit the

* Supported by EPSRC Grant EP/I017127/1

Kimón Fountoulakis

School of Mathematics and Maxwell Institute, The University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom.

E-mail: K.Fountoulakis@sms.ed.ac.uk

Tel.: +44 131 650 5083, Fax: +44 131 650 6553

Jacek Gondzio

School of Mathematics and Maxwell Institute, The University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom.

E-mail: J.Gondzio@ed.ac.uk

Tel.: +44 131 650 8574, Fax: +44 131 650 6553

Pavel Zhlobich

School of Mathematics and Maxwell Institute, The University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom.

E-mail: P.Zhlobich@ed.ac.uk

Tel.: +44 131 650 5044, Fax: +44 131 650 6553

special features of the signal processing matrices within the matrix-free IPM. Two such features are of particular interest: an excellent conditioning of these matrices and the ability to perform inexpensive (low complexity) matrix-vector multiplications with them.

Computational experience with large scale one-dimensional signals confirms that the new approach is efficient and offers an attractive alternative to other state-of-the-art solvers.

Keywords Matrix-free Interior Point, Preconditioned Conjugate Gradient, Compressed Sensing, Compressive Sampling, ℓ_1 -regularization.

Mathematics Subject Classification (2000) 90C05, 90C06, 90C30, 90C25, 90C51

1 Introduction

We are concerned with the solution of the incomplete system, $m < n$, of linear equations

$$Ax = \hat{b}, \quad (1)$$

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $\hat{b} \in \mathbb{R}^m$. In particular, we are interested in the solution x with the smallest possible number of nonzero elements, otherwise known as the sparsest solution $\hat{x}$. Such problems arise in the fields of Statistics [32] and Signal processing [9].

The sparsest solution $\hat{x}$ of system (1) can be found by solving the following problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \|x\|_0 \\ \text{s.t.:} & Ax = \hat{b}, \end{aligned} \quad (2)$$

where $\|x\|_0 = \{\# \text{ of nonzero entries in } x\}$ and "s.t." stands for "subject to". The use of zero-norm makes the problem combinatorial and untractable in practice. Recent advances in the field of CS show that in certain situations [9] exact recovery of the sparsest solution $\hat{x}$ of (1) can be achieved with an overwhelming probability by solving the following Basis Pursuit [14] problem:

$$\begin{aligned} \text{BP:} \quad \min_{x \in \mathbb{R}^n} & \|x\|_1 \\ \text{s.t.:} & Ax = \hat{b}, \end{aligned} \quad (3)$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$. The problem (3) has a major advantage over (2). Unlike the zero-norm objective in (2), the ℓ_1 -norm objective in (3) can be reformulated as a linear function and therefore the problem (3) may be recast as a linear problem and becomes computationally tractable. Having a linear reformulation of (3), standard efficient optimization methods can be used to recover the sparsest solution $\hat{x}$.

In real-life applications the right hand side of (1) is often corrupted with noise and (1) is replaced with:

$$Ax = b = \hat{b} + e, \quad (4)$$

where $e \in \mathbb{R}^m$ denotes the error: we assume it has a normal distribution $e_i \sim \mathcal{N}(0, \sigma^2) \forall i = 1, 2, \dots, m$. For the noisy case (4) the sparsest solution $\hat{x}$ can be found by solving one of the following problems:

$$\text{BPDN :} \quad \min_{x \in \mathbb{R}^n} \tau \|x\|_1 + \|Ax - b\|_2^2 \quad (5a)$$

$$\text{LASSO:} \quad \begin{aligned} & \min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \\ & \text{s.t.: } \|x\|_1 \leq \epsilon_1 \end{aligned} \quad (5b)$$

$$\text{BP}_{\epsilon_2}: \quad \begin{aligned} & \min_{x \in \mathbb{R}^n} \|x\|_1 \\ & \text{s.t.: } \|Ax - b\|_2 \leq \epsilon_2 \end{aligned} \quad (5c)$$

where τ, ϵ_1 and ϵ_2 are positive scalars that regulate the sparsity and the upper bound on the noise error, respectively. Problem (5a) is the well-known *Basis Pursuit Denoising* introduced in [14], problem (5b) is the *Least Absolute Shrinkage and Selection Operator* (LASSO) used frequently in the field of computational statistics [32]. It can be shown using Theorem 27.4 from [34] that the problems in (5) are equivalent for specific values of scalars τ, ϵ_1 and ϵ_2 .

Practical problems have large dimensions and off-the-shelf approaches such as the simplex method or the (standard) IPM are often impractical. However, matrices A that appear in CS problems display several attractive features which may be exploited within an optimization algorithm. This has created an interest in developing specialized approaches to solving such problems.

There have been various first-order methods developed for the solution of (3) and (5). Let us mention the ones known to be the most efficient.

- *Gradient Projection Sparse Reconstruction* GPSR [20] defines new variables $u, v \in \mathbb{R}^n$ such that

$$|x_i| = u_i + v_i \quad \forall i = 1, 2, \dots, n, \quad (6)$$

where $u_i = \max(x_i, 0)$ and $v_i = \max(-x_i, 0)$. Then linearization of the ℓ_1 -norm is performed

$$\|x\|_1 = \mathbf{1}_n^T u + \mathbf{1}_n^T v, \quad (7)$$

with $u, v \geq 0$ and $\mathbf{1}_n \in \mathbb{R}^n$ being a column vector of all ones. Using the above linearization technique, GPSR solves the following constrained smooth reformulation of problem (5a)

$$\begin{aligned} & \min_{z \in \mathbb{R}^{2n}} \tau \mathbf{1}_{2n}^T z + \frac{1}{2} \|F^T z - b\|_2^2 \\ & \text{s.t.: } z \geq 0, \end{aligned} \quad (8)$$

where $z = [u ; v] \in \mathbb{R}^{2n}$, $F^\top = [A \ -A] \in \mathbb{R}^{m \times 2n}$. Once optimal values of variables u and v are found the solution x of the initial problem is retrieved by computing

$$x = u - v.$$

The price for the linearization is that comparing to the initial BPDN problem (5a) the dimension of the problem is doubled and $2n$ new non-negativity constraints are added. At each step of the algorithm a line search is performed along the negative gradient direction and the new iterate is projected to the feasible set defined by the imposed constraints $z \geq 0$.

- *Fixed Point Continuation Active Set* FPC_AS [39] solves problem (5a). FPC_AS is a two stage algorithm. At the first stage a shrinkage scheme is employed which aims to spot quickly the nonzero components of the sparse representation. Then the second stage is enabled to solve a smooth version of (5a) limited to the indexes of nonzero components found by the first stage of the algorithm.
- *Spectral Projected Gradient* SPGL1 [4] solves any of the problems (3), (5b) and (5c). The SPGL1 is a spectral projection gradient algorithm which iteratively solves (5b) for some values of ϵ_1 , each approximate solution of (5b) is used to build a root-finding problem, which is equivalent to (5c), and is solved by employing a Newton method.
- NestA [3] solves problem (5c) by using a variant of the Nesterov's smoothing gradient algorithm [33], which has been proved to have the optimal bound $\mathcal{O}(1/\epsilon)$ on the number of iterations, where ϵ is the required accuracy.

Independently there have been several attempts to design suitable IPM implementations. The most efficient among them, which can also handle large scale CS problems, are listed below.

- $\ell_1\text{-}\ell_s$ algorithm [29] solves a constrained smooth reformulation of problem (5a) which allows a straightforward preconditioning of the Newton equation system that is solved with a conjugate gradient method.
- PDCO algorithm [36] solves regularized constrained smooth reformulations of problems (3) and (5a). The Newton equation system is solved by applying an LSQR ("Least Squares QR factorization") method.

Both $\ell_1\text{-}\ell_s$ and PDCO have been demonstrated to be robust in comparison with other IPM implementations. However, they are not as accurate and as fast as state-of-the-art first-order methods.

In this paper we present a primal-dual feasible IPM specialized to CS problems. Primal-dual because it iterates simultaneously on primal and dual variables of a smooth reformulation of problem (5a) and feasible because the smooth reformulation of (5a) consists only of conic constraints which are always satisfied. Primal-dual methods have been shown to have the best theoretical convergence properties [30] among various IPMs, but they also enjoy the best practical convergence [24, 40]. Here we give a brief introduction of the structure of primal-dual IPM methods and we discuss important modifications

that result in the proposed approach. The actual implementation used in this paper is given in Subsection 5.1.

Primal-dual methods rely on Newton method to calculate primal-dual directions at each iteration. Newton method for primal-dual IPMs finds roots for linearized KKT (Karush-Kuhn-Tucker) systems or their reduced versions known as augmented and normal equations systems. These systems arise as first-order optimality conditions of log-barrier primal-dual pairs. The linearized KKT systems, referred as Newton linear systems, can be solved in two ways,

- by employing a direct linear solver, or
- by using an iterative solver, such as Krylov subspace methods [28].

The first option delivers a very robust primal-dual IPM where exact Newton directions are calculated. Despite its robustness this approach has the potential drawback of being computationally expensive. Especially in the case when the Newton linear system does not have an exploitable sparsity pattern and the computational effort per iteration reaches $\mathcal{O}(n^3)$.

The second option involves the use of approximate Newton directions. Although this might slightly increase the number of IPM iterations [26, 31], one hopes that the decreased computational effort per iteration should offset such a disadvantage. The performance of iterative methods depends on the spectral properties of the Newton linear system [28] and benefit from the use of appropriate preconditioning techniques which cluster the eigenvalues of the Newton linear system. If the Newton linear system is ill-conditioned and no low-cost preconditioner is applicable, then a direct approach might be more efficient. To conclude, a criterion to select between the two approaches of solving the Newton linear systems should take into account

1. the sparsity pattern of the systems,
2. the existence of a computationally inexpensive preconditioner,
3. the memory requirements of storing problem's data,
4. the existence of fast matrix-vector product implementations with the matrix of the linear system to be solved.

In this paper, we focus on the situation where there is no particular sparsity pattern, the memory requirements can be high but conditions 2 and 4 are satisfied. For this reason, a preconditioned conjugate gradient method is more attractive than a direct method. Indeed, in the approach proposed in this paper, at each step of the primal-dual IPM the preconditioned conjugate gradient method is applied to compute an approximate Newton direction. Since we rely on an iterative method for linear algebra, the proposed primal-dual IPM is matrix-free [25], i.e. the explicit problem formulation is avoided and the measurement matrix A is used only as an operator to produce results of matrix-vector products Ax and $A^T y$. Although matrices A used in CS can be completely dense, i.e. Gaussian, partial Fourier, partial DCT (Discrete Cosine Transform), partial DST (Discrete Sine Transform), Haar wavelets etc, they do have interesting (exploitable) features. Arguments 3 and 4 are satisfied because for many measurement matrices that appear in sparse signal reconstruction

problems there are *super-fast algorithms* (e.g. $\mathcal{O}(n)$ or $\mathcal{O}(n \log n)$ complexity) for multiplication by a vector. For example, for Fourier, DCT and DST matrices there exists the FFTW implementation ("Fastest Fourier Transform in the West") [22] with complexity $\mathcal{O}(n \log n)$, for Haar wavelet and Noiselet matrices there exist algorithms of complexity $\mathcal{O}(n)$, see [1] and [15], respectively. Finally, to satisfy argument 2 we propose a preconditioner efficient on certain problems that is based on the fact that *sub-matrices of A with a given number of columns are uniformly well-conditioned* (this is called the Restricted Isometry Property, see the discussion in Section 2).

The objective of our developments is to design an IPM which preserves the main advantage of IPM, that is, it converges in merely a few iterations, and removes the main drawback of IPM, that is, avoids expensive computations of the Newton direction. Ideally, we would like to solve the CS problems in $\mathcal{O}(\log n)$ IPM iterations and keep the cost of a single IPM iteration as low as possible and not exceeding $\mathcal{O}(n \log n)$.

The paper is organized as follows. In Section 2 we discuss the particular features of CS matrices that are exploited in our approach. In Section 3 we reformulate sparse recovery optimization problems (3) and (5a) to make them suitable for matrix-free IPM. Section 4 concerns finding approximate Newton directions required at each step of the IPM. We calculate the normal equations system formulation of the above stated problem and analyze its properties. We propose an efficient preconditioner that can be used in the preconditioned conjugate gradient method. We prove that under certain conditions (that are satisfied in practice) eigenvalues of the preconditioned matrix are well clustered around 1. In Section 5 we compare the proposed matrix-free IPM with other state-of-the-art first and second-order solvers.

2 Properties of Compressed Sensing Matrices

Matrices which appear in sparse reconstruction problems originate from different bases in which signals are represented. What they all have in common are the conditions that guarantee recoverability of the sparsest solution of (1) by means of the ℓ_1 -norm minimization (3). The restricted isometry property (RIP) [13] is one of such conditions which shows how efficiently a measurement matrix captures information about sparse signals.

Definition 1 The restricted isometry constant δ_k of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as the smallest δ_k such that

$$(1 - \delta_k) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (9)$$

for all at most k -sparse $x \in \mathbb{R}^n$.

In words, for small δ_k , statement (9) requires that all column sub-matrices of A with at most k columns are well-conditioned. Informally, A is said to satisfy the RIP if δ_k is small for a reasonably large k . The next theorem

due to [21] establishes the relation between the RIP property and the sparse recovery.

Theorem 1 *Every k -sparse vector $x \in \mathbb{R}^n$ satisfying $Ax = \hat{b}$ is the unique solution of (3) if*

$$\delta_{2k} < \frac{3}{4 + \sqrt{6}} \approx 0.4652.$$

The restricted isometry property also implies stable recovery by ℓ_1 -norm minimization for vectors that can be well approximated by sparse ones, and it further implies robustness under noise on the measurements [13].

RIP is a very restrictive condition that depends on the size of the measurement matrix A . Clearly, the more columns n matrix A has (the larger the size of the vector x to recover) the larger δ_k in (9) is (the harder it is to guarantee sparse recovery). On the other hand, number of rows m of A is the number of measurements taken and, hence, the RIP constant δ_k decreases with m . Currently known measurement matrices satisfying RIP with small number of measurements fall into two categories [35]: (i) random matrices with i.i.d. sub-Gaussian variables, e.g., normalized i.i.d. Gaussian or Bernoulli matrices; (ii) random partial bounded orthogonal matrices obtained by choosing m rows uniformly at random from a normalized $n \times n$ Fourier or Walsh-Hadamard transform matrices. Number of measurements required to satisfy the RIP property for both classes of matrices is given in the table below.

Table 1 List of measurement matrices that have been proven to satisfy RIP

$m \times n$ measurement matrix	RIP regime	references
Gaussian	$m \geq Ck \log n$	[2, 35]
partial Fourier	$m \geq Ck \log^4 n$	[35]

Although it follows from Table 1 that Gaussian matrices are optimal for sparse recovery, they have limited use in practice because many applications impose structure on the matrix. Furthermore, recovery algorithms are significantly more efficient when the matrix admits a fast matrix-vector product. Due to the two former practical reasons, and since we are dealing with large-scale CS applications we limit ourselves to applications with measurement matrices A that

- are not stored explicitly,
- admit a low-cost matrix-vector product with A (e.g. $\mathcal{O}(n \log n)$ or $\mathcal{O}(n)$).

An important broad class of CS matrices comes from random sampling in bounded orthonormal systems. Partial Fourier matrix mentioned earlier is just one example of this type. Other examples are matrices related to systems of real trigonometric polynomials (partial discrete cosine (DCT) and discrete

sine (DST) matrices), Haar wavelets and noiselets. Quite often in applications a signal is sparse with respect to a basis different from the one in which measurements are made. Then it is said that a measurement/sparsity pair is given [9]. Assume that a vector z is sparse with respect to the basis of columns of a unitary matrix Ψ (*sparsity matrix*), i.e. $z = \Psi x$ for a k -sparse vector x . Further, assume that z is sampled with respect to the basis of columns of a unitary matrix Φ (*measurement matrix*): $y = R_m \Phi^\top z$, where R_m is a random sampling operator which satisfies $R_m R_m^\top = I$. Hence, matrix A in (1) is equal to $R_m \Phi^\top \Psi$ and its rows are orthonormal:

$$AA^\top = I_m. \quad (10)$$

The recoverability property of matrix A depends on the value of the so-called *mutual coherence* $\mu(\Phi, \Psi)$ of the measurement/sparsity pair (see [16]):

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{i,j} |\langle \phi_i, \psi_j \rangle|, \quad (11)$$

where ϕ_i, ψ_i are the i^{th} columns of matrices Φ, Ψ , respectively. Coherence simply measures the largest correlation between any two elements of Φ and Ψ . Next theorem due to [12] shows that the smaller the value of mutual coherence the better the recoverability property of matrix A .

Theorem 2 Fix $z \in \mathbb{R}^n$ and suppose that the coefficient sequence x of z in the unitary $n \times n$ basis Ψ is k -sparse. Select m measurements in the unitary $n \times n$ Φ domain uniformly at random. Then if

$$m \geq Ck\mu(\Phi, \Psi)^2 \log(n/p) \quad \text{and} \quad m \geq C' \log^2(n/p) \quad (12)$$

for some positive constants C, C' , then with overwhelming probability exceeding $1 - p$, the vector x is the unique solution to the ℓ_1 -minimization problem (3) with $A = R_m \Phi^\top \Psi$, where $R_m R_m^\top = I$ and A has orthonormal rows (10).

Let us note that condition (12) differs from those given in Table 1. Conditions in Table 1 ensure that once the random matrix is chosen, then with high probability all sparse signals can be recovered (*uniform recovery*). Although, (12) only guarantees that each fixed sparse signal can be recovered with high probability using a random draw of the matrix (*nonuniform recovery*).

To conclude, CS matrices have many useful properties that must be taken into account in the development of an efficient matrix-free IPM solver. In the current paper we make use only of the most general of them that are satisfied by every CS matrix. First, we weaken a little bit the condition of orthonormality (10) to include random matrices such as Gaussian and Bernoulli:

- **P1:** Rows of matrix A are close to orthonormal, i.e. there exists a small δ such that

$$\|AA^\top - I_m\|_2 \leq \delta. \quad (13)$$

Restricted isometry property (9) on the contrary assumes that columns of A are normalized. So, our interpretation of the RIP property that will be used throughout the paper is as follows.

- **P2**: Every k columns of A with $k \ll m$ are almost orthogonal and have similar norms, i.e. for every matrix B composed of arbitrary k columns of A

$$\left\| \frac{n}{m} B^\top B - I_k \right\|_2 \leq \delta_k. \quad (14)$$

By treating property **P2** as the chosen RIP, the bound for the RIP constant in Theorem 1 which relies on RIP in (9) will change. The following theorem is a modified version of Theorem 1 when property **P2** is used as a RIP.

Theorem 3 *Every k -sparse vector $x \in \mathbb{R}^n$ satisfying $Ax = \hat{b}$ is the unique solution of (3) if*

$$\delta_{2k} < \frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{6}},$$

where δ_{2k} is the minimum constant such that property **P2** holds for every $2k$ columns of matrix A , denoted by matrix B in **P2**.

Proof Let $x \in \mathbb{R}^n$ have k nonzero components and B in **P2** be any k column submatrix of A . Then from **P2** it follows that

$$\frac{m}{n} (1 - \delta_k) \|x\|_2^2 \leq \|Ax\|_2^2 \leq \frac{m}{n} (1 + \delta_k) \|x\|_2^2. \quad (15)$$

Proposition 2 in [21] gives bounds for δ_{2k} by using the RIP in (9). In our case, we replaced the RIP in (9) with (15). Therefore, the four modified conditions for δ_{2k} in Proposition 2 in [21] which guarantee that every k -sparse vector $x \in \mathbb{R}^n$ which satisfies $Ax = \hat{b}$ is the unique solution of (3), take the following forms:

- 1) $\delta_{2k} < \frac{1}{2} \frac{m}{n}$ when $k = 1$
- 2) $\delta_{2k} < \frac{3 \frac{m}{n}}{(1 + 3 \frac{m}{n} + \sqrt{(6k - 2r)/(k - 1)})}$ when $k = 3\omega + r$ and $1 \leq r \leq 3$
- 3) $\delta_{2k} < \frac{4 \frac{m}{n}}{(1 + 4 \frac{m}{n} + \sqrt{(12k - 3r)/(k - 1)})}$ when $k = 4\omega + r$ and $1 \leq r \leq 4$
- 4) $\delta_{2k} < \frac{2 \frac{m}{n}}{(1 + 2 \frac{m}{n} + \sqrt{1 + k/(8\omega + \lfloor 8r/5 \rfloor)})}$ when $k = 5\omega + r$ and $1 \leq r \leq 5$,

where $\omega = 0, 1, \dots$ is an integer variable. Table 2 shows with bold font which condition of the above four is the weakest for $2 \leq s \leq 8$. This table is equivalent of the table in proof of Theorem 1 in [21]. However, in [21] the table has exact values, where our Table 2 has functions depending on the ratio m/n instead. Using the same arguments as in proof of Theorem 1 in [21] and Table 2 we conclude that every k -sparse vector $x \in \mathbb{R}^n$ satisfying $Ax = \hat{b}$ is the unique solution of (3) if

$$\delta_{2k} < \frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{6}}.$$

This completes the proof.

Table 2 Weakest RIP constant values for sparsity level $2 \leq k \leq 8$

	Case 2	Case 3	Case 4
$k = 2$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + 2\sqrt{2}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + 3\sqrt{2}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \sqrt{\frac{5}{3}}}$
$k = 3$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{6}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + 3\sqrt{\frac{3}{2}}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \frac{1}{2}\sqrt{7}}$
$k = 4$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{\frac{22}{3}}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + 2\sqrt{3}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \sqrt{\frac{5}{3}}}$
$k = 5$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{\frac{13}{2}}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + \frac{1}{2}\sqrt{57}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \frac{1}{2}\sqrt{\frac{13}{2}}}$
$k = 6$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + \sqrt{6}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + \sqrt{\frac{66}{5}}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \sqrt{\frac{5}{3}}}$
$k = 7$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + 2\sqrt{\frac{5}{3}}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + \frac{5}{\sqrt{2}}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + 3\sqrt{\frac{2}{11}}}$
$k = 8$	$\frac{3 \frac{m}{n}}{1 + 3 \frac{m}{n} + 2\sqrt{\frac{11}{7}}}$	$\frac{4 \frac{m}{n}}{1 + 4 \frac{m}{n} + 2\sqrt{3}}$	$\frac{2 \frac{m}{n}}{1 + 2 \frac{m}{n} + \sqrt{\frac{5}{3}}}$

Comparing the two bounds of the RIP constants in Theorems 1 and 3 we observe that the former is smaller, see Figure 1. For the purpose of the proposed preconditioner, discussed in Section 4, the smaller bound on δ_{2k} in Theorem 3 results in tighter bounds of the spectral properties of the preconditioned systems. The former is an advantage of property **P2** against RIP in (9), proved in Lemma 1. However, property **P2** and Theorem 3 result in a limitation of the maximum number of sparsity k for which problem (3) guarantees an exact recovery of the sparsest solution of $Ax = \hat{b}$. Fortunately, both results in Theorems 1 and 3 are rather pessimistic. It has been shown in [7] that RIP conditions of the form (9) and their scaled versions (**P2**) or (15) provide worst case scenarios of δ_{2k} and consequently of the sparsity level k such that problem (3) guarantees exact sparse recovery. To support the former argument, we refer the reader to [18], where it is shown that for Gaussian measurement matrices the average maximum sparsity level k that is guaranteed to be reconstructable by (3) is much greater than the one shown in Theorems 1 and 3. Moreover, it has been shown empirically in [18] that approximately the same result holds for various types of measurement matrices A , i.e. partial Fourier, partial Hadamard, Bernoulli etc. In Subsection 5.8 it is shown that the proposed algorithm satisfies approximately the average maximum sparsity k shown in [18]. Therefore, we conclude that by replacing RIP (9) with property **P2**:

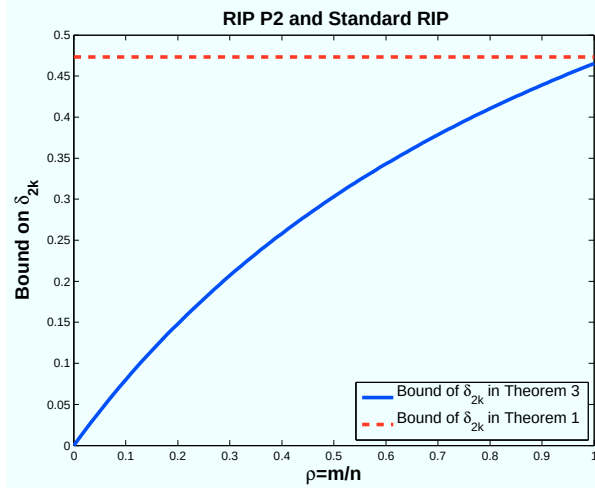


Fig. 1 Comparison on the bounds of δ_{2k} constants from Theorems 1 and 3

- improved bounds on the spectral properties of the preconditioned systems in Section 4 are obtained,
- a better approximation of matrix $B^T B$ with a scaled diagonal ρI by choosing appropriate constant ρ is possible and
- the empirical average reconstruction properties as shown in Subsection 5.8 are maintained.

3 Primal–Dual Problems in Matrix-free IPM

Non-smooth Basis Pursuit (3) and Basis Pursuit Denoising (5a) optimization problems can be reformulated into equivalent linear and convex quadratic problems, respectively. This is achieved via linearization of the non-smooth ℓ_1 -norm in the objective function.

After reformulating the BPDN problem (5a) to (8) as proposed in [20] for GPSR algorithm, we solve the latter using a primal–dual IPM. The reader interested in the theory of primal–dual IPMs is referred to the book of Wright [40]. Aspects of practical implementation have been addressed in a recent survey [24]. A description of the primal–dual IPM used in this paper is given in Section 5.1. For the primal problem (8) of interest the dual is

$$\begin{aligned} \text{Dual Sep.:} \quad & \max_{z, s \in \mathbb{R}^{2n}} -\frac{1}{2} z^T F F^T z \\ \text{s.t.:} \quad & c + F F^T z - s = 0 \\ & z, s \geq 0 \end{aligned} \tag{16}$$

where $c = \begin{bmatrix} \tau 1_n - A^T b \\ \tau 1_n + A^T b \end{bmatrix} \in \mathbb{R}^{2n}$.

At each step of the primal–dual IPM applied to the primal–dual pair (8) and (16) the corresponding Newton direction $(\Delta z, \Delta s)$ is computed by solving the following system of linear equations:

$$\begin{bmatrix} FF^\top - I_{2n} \\ S & Z \end{bmatrix} \times \begin{bmatrix} \Delta z \\ \Delta s \end{bmatrix} = \begin{bmatrix} f_z \\ f_s \end{bmatrix}, \quad (17)$$

where S and Z are diagonal matrices with vectors s and z on the diagonal, respectively, I_{2n} denotes an identity matrix of dimension $2n$ and

$$f_z = s - c - FF^\top z, \quad f_s = \sigma \mu e - ZS1_{2n}, \quad (18)$$

$\mu = z^\top s / (2n)$ is the barrier term of the IPM and $0 \leq \sigma \leq 1$ the centering parameter. In the matrix-free framework the dual variables Δs in (17) are eliminated to get:

$$(\Theta^{-1} + FF^\top) \Delta z = f_z + Z^{-1} f_s, \quad (19a)$$

$$\Delta s = Z^{-1} f_s - \Theta^{-1} \Delta z. \quad (19b)$$

where $\Theta = S^{-1}Z \in \mathbb{R}^{2n \times 2n}$. The reduced Newton system (19a), also known as augmented system, is solved by an appropriate preconditioned iterative method for which *only matrix-vector product with the constraint matrix F is allowed*. Thus, the matrix-free IPM approach has two major components:

- iterative solver for the augmented system,
- special-purpose preconditioner that exploits matrix structure.

The next section addresses these two issues.

4 Preconditioned Conjugate Gradient Method

The system (19a) has a symmetric positive definite matrix and the conjugate gradient (CG) method can be employed to solve it in a matrix-free regime. However, the convergence of the CG method can be too slow when a matrix is ill-conditioned and/or its eigenvalues are not clustered. In this section we discuss an efficient spectrally-equivalent diagonal matrix preconditioner for (19a). In particular, we give theoretical and practical justification of our approach to fast iterative solution of the system.

The proposed preconditioner for the system of equations (19a) is based on the exploitation of general properties of CS matrices and the behavior of the Θ matrix in (19a) close to optimality. Let us recall that in the notation of primal–dual pair (8)–(16), variable $s \in \mathbb{R}^{2n}$ is a Lagrange multiplier associated with the non-negativity constraint $z \geq 0$. Hence, at optimality $s_j z_j = 0 \forall j = 1, 2, \dots, 2n$. IPMs force the convergence to the optimal solution by perturbing this condition $s_j z_j = \mu \forall j$, where μ is the barrier term of the

IPM, and gradually reducing the perturbation μ to zero. At optimality indices $j \in \{1, 2, \dots, 2n\}$ are split into two disjoint sets:

$$\begin{aligned} \mathcal{B} &= \{j \mid z_j \rightarrow z_j^* > 0, s_j \rightarrow s_j^* = 0\} \\ &\text{and} \\ \mathcal{N} &= \{j \mid z_j \rightarrow z_j^* = 0, s_j \rightarrow s_j^* > 0\} \end{aligned} \quad (20)$$

that determine the activity of constraints. This partitioning has highly undesirable consequences for the diagonal scaling matrix $\Theta = S^{-1}Z$. Indeed, when μ approaches zero, for indices $j \in \mathcal{B}$, Θ_j goes to infinity and for indices $j \in \mathcal{N}$, Θ_j goes to zero.

Recall that $z = [u ; v]$, where u and v are the positive and negative components of vector x (see (6)), respectively. For sparse signals there are merely k ($k \ll 2n$) nonzero components in the optimal solution. The positive ones will contribute a nonzero element in u and the negative ones will contribute a nonzero element in v . At optimality the cardinality of set $\mathcal{B}$ is k . Hence, at later iterations of an IPM

$$\begin{aligned} \Theta_i &\gg 1 \quad \forall i \in \mathcal{B}, \quad \text{card } \mathcal{B} = k, \\ \Theta_i &\ll 1 \quad \forall i \in \mathcal{N}, \quad \text{card } \mathcal{N} = 2n - k. \end{aligned} \quad (21)$$

Let us now return to the question of preconditioning of the system of equations (19a). Its matrix is

$$H = \Theta^{-1} + FF^\top. \quad (22)$$

The behavior of matrix Θ near optimality is described by (21). It is clear that matrix Θ^{-1} has many large entries and only few small entries well before the IPM reaches the optimal solution. Let us introduce a number $C \gg 1$ that separates entries of Θ^{-1} of different magnitudes:

$$\#(\Theta_j^{-1} < C) = l. \quad (23)$$

Here l is just the number of small entries in Θ^{-1} and may be different from the sparsity k of the optimal solution. In the regime $l < m$, the second term $FF^\top$, whose rank is exactly m , works as a low-rank perturbation for the matrix Θ^{-1} in (22). Since, in Frobenius norm the first term Θ^{-1} dominates the second term $FF^\top$, we propose to replace $FF^\top$ in the preconditioner by a simple approximant. First, let us write system's matrix of (19a) in the block form by using the facts that $\Theta = \text{diag}(\Theta_u, \Theta_v)$ and $F^\top = [A \ -A]$:

$$H = \begin{bmatrix} \Theta_u^{-1} & \\ & \Theta_v^{-1} \end{bmatrix} + \begin{bmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{bmatrix}. \quad (24)$$

Our preconditioner is based on the approximation of $A^\top A$ by the closest (in Frobenius norm) scaled identity matrix ρI_n , $\rho = m/n$:

$$P = \begin{bmatrix} \Theta_u^{-1} + \rho I_n & -\rho I_n \\ -\rho I_n & \Theta_v^{-1} + \rho I_n \end{bmatrix}. \quad (25)$$

To simplify the analysis of the preconditioner, we first consider the case of $n \times n$ matrices H and P rather than block $2n \times 2n$ ones as defined by (24) and (25). The following lemma establishes spectral properties of the preconditioned matrix $P^{-1}H$ in the non-block case.

Lemma 1 *Define matrix H as*

$$H = \Theta^{-1} + A^\top A,$$

where $\Theta = \text{diag}(\Theta_1, \Theta_2, \dots, \Theta_n)$ — diagonal $n \times n$ matrix with $\Theta_j > 0$, and A — $m \times n$ matrix with $m \leq n/2$. Let C be any positive constant and l be defined as in (23), $\#(\Theta_j^{-1} < C) = l$. Additionally, let A satisfy property **P2** defined on page 9 for $k = l$ with some constant δ_l . If matrix A has orthonormal rows (10), then the eigenvalues of matrix H preconditioned by matrix P :

$$P = \Theta^{-1} + \rho I_n, \quad \rho = m/n$$

are clustered around 1, i.e.

$$|\lambda - 1| \leq \delta_l + \frac{1}{4} \frac{(3 - \rho)^2}{\rho \delta_l C} \quad \forall \lambda \in \text{spec}(P^{-1}H), \quad (26)$$

If matrix A has nearly orthonormal rows, i.e. satisfies **P1** defined on page 8, then

$$|\lambda - 1| \leq \delta_l + \frac{1}{4} \frac{(1 + \delta - \rho + 2\sqrt{1 + \delta})^2}{\rho \delta_l C} \quad \forall \lambda \in \text{spec}(P^{-1}H),$$

where δ has been defined in **P1**.

Proof Let C be any positive constant, then the following two disjoint sets of indices can be defined:

$$\mathcal{B}_C = \{j \in \{1, 2, \dots, n\} : \Theta_j^{-1} < C\}, \quad \mathcal{N}_C = \{1, 2, \dots, n\} \setminus \mathcal{B}_C$$

Let B and N be matrices of columns of A with indices from $\mathcal{B}_C$ and $\mathcal{N}_C$, respectively. Without loss of generality we can assume that $\mathcal{B}_C$ are the first l indices, then

$$A = [B \ N], \quad B \in \mathbb{R}^{m \times l}, \quad N \in \mathbb{R}^{m \times (n-l)}.$$

Let λ be an eigenvalue of the preconditioned matrix $P^{-1}H$ corresponding to an eigenvector $v = [v_{\mathcal{B}_C} ; v_{\mathcal{N}_C}]$ of norm one, then

$$P^{-1}Hv = \lambda v \iff (H - P)v = \tau Pv, \quad \tau = \lambda - 1, \quad (27)$$

or, in the block form,

$$\left[\begin{array}{c|c} B^\top B - \rho I_l & B^\top N \\ \hline N^\top B & N^\top N - \rho I_{n-l} \end{array} \right] \begin{bmatrix} v_{\mathcal{B}_C} \\ v_{\mathcal{N}_C} \end{bmatrix} = \tau \left[\begin{array}{c|c} \Theta_{\mathcal{B}_C}^{-1} + \rho I_l & 0 \\ \hline 0 & \Theta_{\mathcal{N}_C}^{-1} + \rho I_{n-l} \end{array} \right] \begin{bmatrix} v_{\mathcal{B}_C} \\ v_{\mathcal{N}_C} \end{bmatrix} \quad (28)$$

Obviously, eigenvalues of $P^{-1}H$ are all real, hence τ is also real. Multiplication of (28) by $[v_{\mathcal{B}_C} ; v_{\mathcal{N}_C}]^T$ from the left gives

$$\begin{aligned} \tau \left[v_{\mathcal{B}_C}^T (\Theta_{\mathcal{B}_C}^{-1} + \rho I_l) v_{\mathcal{B}_C} + v_{\mathcal{N}_C}^T (\Theta_{\mathcal{N}_C}^{-1} + \rho I_{n-l}) v_{\mathcal{N}_C} \right] = \\ v_{\mathcal{B}_C}^T (B^T B - \rho I_l) v_{\mathcal{B}_C} + v_{\mathcal{N}_C}^T (N^T N - \rho I_{n-l}) v_{\mathcal{N}_C} + 2v_{\mathcal{B}_C}^T B^T N v_{\mathcal{N}_C}. \end{aligned} \quad (29)$$

Let us denote $\|v_{\mathcal{B}_C}\|_2^2$ by α , then $\|v_{\mathcal{N}_C}\|_2^2 = 1 - \alpha$ since $v = [v_{\mathcal{B}_C} ; v_{\mathcal{N}_C}]$ has unit norm. Bounding left hand side of (29) from below is trivial:

$$\left| \tau \left[v_{\mathcal{B}_C}^T (\Theta_{\mathcal{B}_C}^{-1} + \rho I_l) v_{\mathcal{B}_C} + v_{\mathcal{N}_C}^T (\Theta_{\mathcal{N}_C}^{-1} + \rho I_{n-l}) v_{\mathcal{N}_C} \right] \right| \geq |\tau| (\rho \alpha + C(1 - \alpha)). \quad (30)$$

Next, let us bound right hand side of (29) from above. We will distinguish two cases, orthonormal and nearly orthonormal rows of matrix A . First, we study the case of nearly orthonormal rows of matrix A . For this purpose we will use the SVD decompositions of matrices B and N :

$$B = U_B \Sigma_B V_B^T, \quad \Sigma_B = \begin{bmatrix} \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_l) \\ O_{m-l \times l} \end{bmatrix}$$

and

$$N = U_N \Sigma_N V_N^T, \quad \Sigma_N = \begin{bmatrix} \text{diag}(\varsigma_1, \varsigma_2, \dots, \varsigma_m) \\ O_{m \times (n-m-l)} \end{bmatrix}.$$

Restricted isometry property **P2** implies that

$$\sigma_1^2 \leq \rho(1 + \delta_l), \quad \sigma_l^2 \geq \rho(1 - \delta_l).$$

First, notice that

$$\left| v_{\mathcal{B}_C}^T (B^T B - \rho I_l) v_{\mathcal{B}_C} \right| \leq \rho \delta_l \alpha. \quad (31)$$

Using property **P1** we have

$$\begin{aligned} \|AA^T - I_m\|_2 &\leq \delta && \iff \\ \|AA^T\|_2 &\leq 1 + \delta && \iff \\ \|BB^T + NN^T\|_2 &\leq 1 + \delta && \implies \\ \|NN^T\|_2 &\leq 1 + \delta && \iff \\ \varsigma_1^2 &\leq 1 + \delta. && (32) \end{aligned}$$

Next, using (32) obtain

$$\|N^T N - \rho I_{n-l}\|_2 \leq \max\{\rho, 1 + \delta - \rho\} = 1 + \delta - \rho, \quad (\rho = m/n \leq 0.5)$$

and, hence,

$$\left| v_{\mathcal{N}_C}^T (N^T N - \rho I_{n-l}) v_{\mathcal{N}_C} \right| \leq (1 + \delta - \rho)(1 - \alpha). \quad (33)$$

Finally,

$$\|B^T N\|_2 \leq \|B\|_2 \|N\|_2 \leq \sigma_1 \varsigma_1 = \sqrt{1 + \delta} \sqrt{\rho(1 + \delta_l)} < \sqrt{(1 + \delta)}$$

because our assumptions $m \leq n/2$ and $\delta_l < 1$ imply $\sigma_i^2 \leq \sigma_1^2 \leq \rho(1 + \delta_l) < 1$. We conclude that

$$\left| 2v_{\mathcal{B}^C}^\top B^\top N v_{\mathcal{N}^C} \right| < 2\sqrt{1 + \delta} \sqrt{\alpha(1 - \alpha)}. \quad (34)$$

Bounds (33) and (34) are sharp and can be used to obtain very tight estimate on τ but we do not need them that sharp to obtain a sufficiently good estimate. So, we will release them a little bit to simplify the analysis:

$$\begin{aligned} \left| v_{\mathcal{N}^C}^\top (N^\top N - \rho I_{n-l}) v_{\mathcal{N}^C} \right| &\leq (1 + \delta - \rho)(1 - \alpha) \leq (1 + \delta - \rho)\sqrt{1 - \alpha}, \\ \left| 2v_{\mathcal{B}^C}^\top B^\top N v_{\mathcal{N}^C} \right| &< 2\sqrt{1 + \delta} \sqrt{\alpha(1 - \alpha)} \leq 2\sqrt{1 + \delta} \sqrt{1 - \alpha}. \end{aligned} \quad (35)$$

Using (30) and (31) and (35) we finally get

$$|\tau| \leq \frac{\rho\delta_l\alpha + (1 + \delta - \rho + 2\sqrt{1 + \delta})\sqrt{1 - \alpha}}{\rho\alpha + C(1 - \alpha)} \leq \delta_l(1 + \varepsilon). \quad (36)$$

Let us denote $\xi = (1 + \delta - \rho + 2\sqrt{1 + \delta})$ and show that ε is small for large values of C . Indeed (36) implies that

$$\xi\sqrt{1 - \alpha} \leq \delta_l(C + C\varepsilon - \rho\varepsilon)(1 - \alpha) + \rho\delta_l\varepsilon.$$

It can be checked by simple calculus, that $\sqrt{x} \leq C_1x + C_2$ on $[0, 1]$ whenever $C_1 \geq 1/(4C_2)$. In our case this implies

$$\frac{\delta_l}{\xi}(C + C\varepsilon - \rho\varepsilon) \geq \frac{\xi}{4\rho\delta_l\varepsilon}.$$

The largest solution of the quadratic equation in ε

$$\frac{4\rho\delta_l^2}{\xi^2}\varepsilon(C + C\varepsilon - \rho\varepsilon) = 1$$

is

$$\varepsilon_+ = \frac{1}{2} \cdot \frac{C}{C - \rho} \left(\sqrt{1 + \frac{\xi^2}{\rho\delta_l^2 C} \cdot \frac{C - \rho}{C}} - 1 \right) \leq \frac{\xi^2}{4\rho\delta_l^2 C}.$$

Hence, it is sufficient to take any $\varepsilon \geq \xi^2/(4\rho\delta_l^2 C)$ to satisfy the inequality (36):

$$\boxed{|\tau| \leq \delta_l + \frac{\xi^2}{4\rho\delta_l C} = \delta_l + \frac{1}{4} \frac{(1 + \delta - \rho + 2\sqrt{1 + \delta})^2}{\rho\delta_l C}}. \quad (37)$$

This completes the proof for matrix A which satisfies property **P1**. For the case of orthonormal rows of matrix A , i.e. $AA^\top = I_m$ simply set $\delta = 0$ in property **P1** to get

$$\boxed{|\tau| \leq \delta_l + \frac{1}{4} \frac{(3 - \rho)^2}{\rho\delta_l C}}. \quad (38)$$

This completes the proof.

For the result of the theorem to be useful we obviously need the bound in the right-hand side of inequalities in (37) and (38) to be sufficiently smaller than one. Let us take a closer look at the terms forming this bound. We are free to choose any value for the constant C we want, the larger the better. However, according to (23), l increases with the increase in C and, consequently, the restricted isometry constant δ_l also increases. Inequalities (37) and (38) hold for any value of C , hence we can replace it with

$$|\tau| \leq \min_C \left(\delta_l + \frac{1}{4} \frac{(1 + \delta - \rho + 2\sqrt{1 + \delta})^2}{\rho \delta_l C} \right) \quad (39)$$

and

$$|\tau| \leq \min_C \left(\delta_l + \frac{1}{4} \frac{(3 - \rho)^2}{\rho \delta_l C} \right) \quad (40)$$

and choose constant C that delivers the minimum.

For number of measurements m just a fraction $\rho = 1/4$ of the length n of the unknown signal, it is natural to assume the restricted isometry constant δ_{2l} to be less than $1/4$ (see Theorem 3), hence, according to [6], $\delta_{2l} < 1/4$, implies $\delta_l < 1/4$. Therefore, to have $|\tau| \leq 17/20$ we need $C = 20(0.75 + \delta + 2\sqrt{1 + \delta})^2/3$ in (23). For nearly orthonormal rows of matrix A we can assume that $\delta \leq 1$, which gives us $C \approx 139.74$ and certainly holds near optimality in the IPM. For orthonormal rows of matrix A we have $\delta = 0$, hence, $C \approx 50.41$.

The bounds in (37) and (38) are rather pessimistic. Computational experience suggests that eigenvalues of the preconditioned matrix get well clustered around 1 as long as $l = \#(\Theta_j^{-1} < 1)$ is such that the RIP constant $\delta_l < 1$. For example, for the discrete cosine (DCT) matrix with $n = 2^{10}$ and $m = 2^8$ the corresponding $l \leq 74$ (this number is obtained in a series of random tests).

Now we are ready to state the spectral properties of the preconditioned matrix $P^{-1}H$ for the system of equations (19a). We leave the theorem without a proof as it is a straightforward corollary of Lemma 1.

Theorem 4 *Let H and P be block matrices defined in (24) and (25), respectively. Then the preconditioned matrix $P^{-1}H$ has*

1. *the eigenvalue 1 of multiplicity n ;*
2. *remaining n eigenvalues defined in Lemma 1 with $\Theta = \Theta_u + \Theta_v$.*

Theorem 4 establishes the clustering of eigenvalues of $P^{-1}H$ around 1. Hence, iterative method such as conjugate gradient applied to the system of equations (19a) is expected to converge in just a few iterations if the preconditioner P in (25) is used. The latter theoretical results are also confirmed in practical experiments. Figure 3 demonstrates clustering of eigenvalues $\lambda(H)$ and $\lambda(P^{-1}H)$ in the case that the A matrix in H (24) is a Discrete Cosine Transform (DCT) matrix with normalized rows, $AA^T = I$. The parameters for the size of the problem are set to $m = 2^{10}$, $n = 2^{12}$ and the sparsity level is fixed to $k = 51$. In the left sub-Figure 2a the clustering of the eigenvalues $\lambda(H)$ is shown. Every vertical line presents the spreading of $\lambda(H)$ at a particular

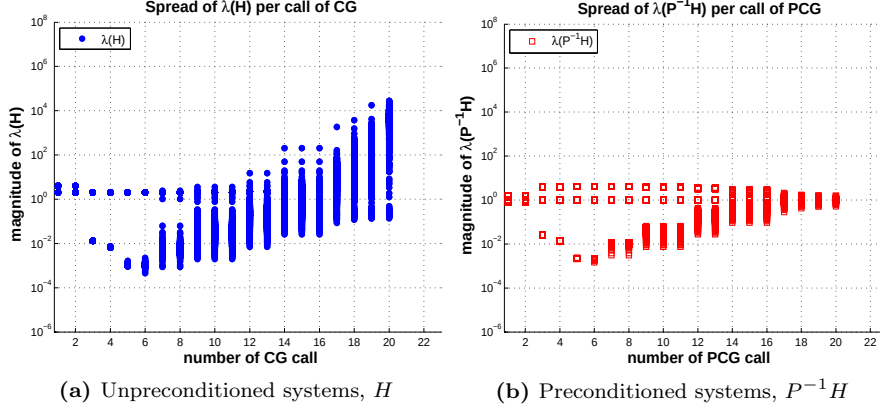


Fig. 2 Clustering of the eigenvalues for the matrices H and $P^{-1}H$ as the matrix-free IPM approaches optimality. The matrix A in H (24) is a DCT matrix with normalized rows. The parameters of the problem set to $m = 2^{10}$, $n = 2^{12}$ and $k = 51$. Twenty systems for the matrices H and $P^{-1}H$ are solved in total

CG call as the matrix-free IPM progresses. One can observe that the clustering worsens as the matrix-free IPM approaches optimality. On the contrary, eigenvalues of the preconditioned matrices $P^{-1}H$ show the opposite behavior. In particular, as the matrix-free IPM progresses eigenvalues $\lambda(P^{-1}H)$ start to cluster around one. The latter is depicted with the vertical columns in the right sub-Figure 2b.

5 Computational Experience

We illustrate our developments by comparing the matrix-free IPM's efficiency with those of the state-of-the-art first-order methods, FPC_AS and SPGL1 and with two other interior point based solvers, ℓ_1 - ℓ_s and PDCO. The experiments are made on Sparco test suite [5].

We use the FPC_AS CG version of FPC_AS algorithm, where “CG” stands for the conjugate gradient method. The FPC_AS CG has been shown in [39] to be considerably faster than other versions of FPC and FPC_AS software packages. The FPC_AS CG solves problem (5a). The code of FPC_AS CG package can be found at http://www.caam.rice.edu/~optimization/L1/FPC_AS/. We use the SPGL1_bp version of SPGL1 software package for noiseless signals and the SPGL1_bpdn version for noisy signals, where “bp” stands for basis pursuit and “bpdn” for basis pursuit denoising, respectively. The SPGL1_bp solves problem (3) and the “bpdn” version solves problem (5c). The code of SPGL1 package can be found at <http://www.cs.ubc.ca/labs/scl/spgl1>. Those versions of the FPC_AS and SPGL1 software packages were found to be faster and more accurate than other first-order methods mentioned in Subsection 5.2. Therefore, GPSR and NestA solvers are excluded from the

Table 3 Symbols and abbreviations used in tables and figures in Section "Computational Experience" 5

m, n, k	number of rows and columns of the matrix A and the number of nonzero elements in the optimal sparsest signal representation
$\hat{x}$	optimal sparse representation
x_W	$x_{W_i} = x_i$ if $i \in W$, otherwise $x_{W_i} = 0$, where $W := \{i = 1, 2, \dots, n \mid \hat{x}_i \neq 0\}$
$r.e(x_W)$	relative error $\ x_W - \hat{x}\ _2 / \ \hat{x}\ _2$
$res(x_W)$	residual $\ Ax_W - b\ _2$, where b should be replaced with $\hat{b}$ in case of noiseless signals
$nId(x_W)$	distance from the optimal value of ℓ_1 -norm, $ \ x_W\ _1 - \ \hat{x}\ _1 $
$obj(x_W)$	objective value of BPDN problem, $\tau\ x_W\ _1 + \ Ax_W - b\ _2^2$
$nMat$	total number of matrix vector products Ax and $A^\top y$

comparison. The $\ell_1\text{-}\ell_s$ solver implements problem (5a), it can be found at http://www.stanford.edu/~boyd/l1_ls/. The PDCO solver is used through the file SolveFasBP.m of SparseLab software package. The PDCO solver can be found at <http://www.stanford.edu/group/SOL/software/pdco.html> and the SparseLab software package at <http://sparselab.stanford.edu/>. The PDCO solver implements problems (3) and (5a).

In addition, three more experiments are performed. The first one tests the robustness of solvers matrix-free IPM, SPGL1_bpdn, FPC_AS CG and $\ell_1\text{-}\ell_s$, on problems of Sparco test suite, given a fixed level of noise. The second, replaces the core of matrix-free IPM, which is the preconditioned CG with a direct solver and shows how the CPU time required for reconstruction scales for each case. The third, demonstrates that the empirical phase transition properties of matrix-free IPM fit the theoretical average phase transition properties shown in [18].

All solvers used in this section, including the matrix-free IPM are MATLAB implementations. All experiments were performed using MATLAB version R2012b (8.0.0.783) 64-bit on a Dual 8 Core Intel Xeon (Sandybridge) running Redhat Enterprise Linux in 64-bit mode. Finally, the RICE Wavelet toolbox, included in Sparco test suite, was compiled using gcc compiler version 4.4.6 20120305 (Red Hat 4.4.6-4). The matrix-free IPM, the data files and the MATLAB scripts used to generate the results in this section can be downloaded from <http://www.maths.ed.ac.uk/ERGO/mfipmcs/>.

Before proceeding to the following subsections it would be convenient for the reader to be familiarized with symbols and abbreviations used in the subsequent figures and comparison tables explained in Table 3.

5.1 Single centrality corrector primal-dual matrix-free IPM

The implementation used in this paper is a single-corrector primal-dual IPM [23]. The original version proposed in [23] makes use of multiple centrality correctors, however, after computational experimentation it was observed that a single corrector was enough for the fast convergence of the IPM in few iterations. In a standard multiple-corrector variant at every iteration multiple centrality corrector directions are calculated, which are combined with a predictor direction in order to produce the final primal-dual direction [23]. To compute the corrector and predictor directions one needs to solve multiple linear systems (19) where only the right hand side varies. In case that a direct solver is used to solve the linear systems, the extra cost of solving several equations instead of one is negligible, because the dominating cost is the decomposition of the matrix $(\Theta^{-1} + FF^T)$. However, this is not the case when iterative method (PCG) is used to solve systems (19). In particular, the cost of calculating every term in composite direction is approximately the same. In order to avoid the high cost of computing extra corrector directions at every iteration in our single-corrector matrix-free IPM we slightly bias the predictor direction to point to the central path and perform corrector directions only when necessary. Like a long-step variant of primal-dual IPM [40] this guarantees that at every iteration the objective function is decreased rapidly while the algorithm maintains the small distance to the central path. As proposed in [23], the criterion to decide whether a corrector direction is calculated is the value of the primal and dual step sizes. When many biased predictor directions are performed the primal-dual iterates tend to approach the boundary of the feasible region. This results in small step sizes of the subsequent iterations. When this happens a strong re-centering corrector is employed which pushes the next iteration to the vicinity of central path such that next step sizes are more likely to have large values. Ideally, the values of the step sizes of the primal and dual directions should be bounded away from zero while global convergence of the method is guaranteed. This would allow fast practical convergence of matrix-free IPM, which translates into few iterations. Indeed, one can observe from the computational experience reported in Section 5.7 that 10 to 20 iterations of the matrix-free IPM is enough for convergence. This behaviour has been observed also in all computational experiments discussed in Section 5.5. We make our software available to the research community so that the interested reader can reproduce any numerical experiments from this paper. The pseudo-code of the implemented single-corrector primal-dual matrix-free IPM follows.

Single-Corrector Primal-Dual Matrix-Free IPM

```

1: Input Choose  $z^0, s^0 > 0$ ,  $0 < \sigma_1 < \sigma_2 < \sigma_3 \leq 1$  and  $0 < \bar{\alpha} < \tilde{\alpha} < 1$ . For  $k = 1, 2, \dots$ 
   generate  $z^{k+1}$  from  $z^k$  and  $s^{k+1}$  from  $s^k$  according to the iteration:

2: while Duality Gap of (8) and (16)  $\geq \epsilon$  do

3:   if  $k \neq 1$  and  $(\alpha_P^{k-1} \leq \bar{\alpha}$  or  $\alpha_D^{k-1} \leq \tilde{\alpha})$  then
4:      $\sigma = \sigma_2$ 
5:   else
6:      $\sigma = \sigma_1$ 
7:   end if

   (* predictor step *)

8:   solve (19) using PCG with  $\sigma$  and  $z^k, s^k$  in (18) to obtain  $(\Delta \bar{z}^k, \Delta \bar{s}^k)$ 
   choose primal and dual step sizes  $\alpha_P^k, \alpha_D^k$  in  $[0, 1]$  as the largest values of  $\alpha_P, \alpha_D$  such
   that


$$z^k(\alpha_P^k) = z^k + \alpha_P \Delta \bar{z}^k > 0$$


$$s^k(\alpha_D^k) = s^k + \alpha_D \Delta \bar{s}^k > 0$$


   (* corrector step *)

9:   if  $\alpha_P^k \leq \tilde{\alpha}$  or  $\alpha_D^k \leq \tilde{\alpha}$  then
10:    solve (19) using PCG with  $\sigma = \sigma_3$  and  $z^k(\alpha_P^k), s^k(\alpha_D^k)$  in (18) to obtain
     $(\Delta \bar{z}^k, \Delta \bar{s}^k)$ 
    set  $(\Delta z^k, \Delta s^k) = (\Delta \bar{z}^k, \Delta \bar{s}^k) + (\Delta \bar{z}^k, \Delta \bar{s}^k)$ 
    choose primal and dual step sizes  $\alpha_P^k, \alpha_D^k$  in  $[0, 1]$  as the largest values of  $\alpha_P, \alpha_D$ 
    such that


$$z^k(\alpha_P^k) = z^k + \alpha_P \Delta z^k > 0$$


$$s^k(\alpha_D^k) = s^k + \alpha_D \Delta s^k > 0$$


11:   end if

12:   set  $(z^{k+1}, s^{k+1}) = (z^k(\alpha_P^k), s^k(\alpha_D^k))$ 
13: end while

```

The input parameters σ_1, σ_2 are user-defined and control the centering bias of the predictor directions, while σ_3 parameter controls the strong centering in the corrector directions. For all experiments they have been set to $\sigma_1 = 0.1$, $\sigma_2 = 0.5$ and $\sigma_3 = 0.8$. The input parameters $\bar{\alpha}$ and $\tilde{\alpha}$ are user-defined, $\bar{\alpha}$ controls whether σ_1 or σ_2 will be used as a centering parameter for the predictor directions and $\tilde{\alpha}$ controls the frequency of the corrector updates. For all experiments they have been set to $\bar{\alpha} = 0.5$ and $\tilde{\alpha} = 0.1$.

5.2 Benchmarks

In order to have a base of comparison we choose to show the efficiency of the matrix-free IPM on already existing benchmarks, which have been used by several researchers including [4, 39]. Experiments are performed on 18 real

valued sparse reconstruction problems, see Table 4, from the Sparco collection [5]. In total, Sparco collection consists of 26 problems, out of which 6 are complex valued and 20 real valued. For the experiments in this section, the complex valued problems with IDs 1, 4, 8, 501 to 503, are ignored since the matrix-free IPM manipulates only real data. Moreover, 2 out of the 20 real valued problems, with IDs 703 and 901, are also ignored because of their difficulty to be generated on any machine in a stand-alone approach, since, they require external packages such as CurveLab [8] and FFTW [22]. For problems in Table 4 with IDs 401 to 403, 601 to 603, 701 and 702, the optimal representation $\hat{x}$ is not given by Sparco toolbox. Therefore, the SPGL1_bp solver is used to obtain $\hat{x}$ with required high accuracy. In particular to obtain $\hat{x}$, the parameters of SPGL1_bp are set to

$$\text{bpTol} = 1.0\text{e-}15, \quad \text{optTol} = 1.0\text{e-}15, \quad \text{decTol} = 20\,000. \quad (41)$$

where bpTol controls the tolerance for identifying a basis pursuit solution, optTol controls the optimality tolerance and decTol controls the frequency of Newton updates. Some of the components of the obtained solution from the SPGL1_bp might be nearly but not exactly zero, hence, as nonzero components are considered the ones in the set $\text{nnz}(x) := \{k = 1, 2, \dots, n \mid \sum_{i=1}^k |\bar{x}_i| \leq 0.999\|x\|_1\}$, where $\bar{x}$ is the vector x sorted in decreasing order of absolute values of its components. Then we set $\hat{x}_i = x_i$ if $i \in W$ otherwise $\hat{x}_i = 0$, where $W := \{i = 1, 2, \dots, n \mid i \in \text{nnz}(x)\}$.

Noise is introduced to the noiseless measurements $\hat{b}$ using the following command in MATLAB:

$$b = \text{awgn}(\hat{b}, \text{SNR}, \text{'measured'}), \quad (42)$$

The function `awgn` is a MATLAB function from Communications Systems Toolbox which adds white Gaussian noise to signal $\hat{b}$. The SNR is the signal to noise ratio, measured in dB. The 'measured' option specifies that the power of the signal is calculated first before the addition of the noise.

5.3 Equivalence of BP_{ϵ_2} and BPDN

It has already being stated in Section 1 that problems BP_{ϵ_2} in (5c) and BPDN in (5a) are equivalent given particular parameters ϵ_2 and τ . In this paper the tested solvers implement problem BP_{ϵ_2} , i.e. SPGL1_bpdn, or problem BPDN, i.e. matrix-free IPM, FPC_AS CG $\ell_1\text{-}\ell_s$ and PDCO. In order to perform a fair comparison among these solvers it has to be made certain that all codes solve equivalent problems. Otherwise, different optimal solutions will be obtained, therefore, a straightforward and clear comparison would be impossible. Unfortunately, exact values of ϵ_2 and τ which make problems BP_{ϵ_2} and BPDN equivalent are not known a priori, except for the case of orthogonal matrix A . However, given ϵ_2 an approximate τ can be computed such that an approximate equivalence holds.

Table 4 18 out of 20 real valued problems of Sparco collection

Problem	ID	m, n	Operator	$\ \hat{x}\ _1$
[16,17] blocksig	2	1 024, 1 024	wavelet	4.5e+02
[16,17] blkheavi	9	128, 128	heaviside	4.1e+01
[16,17] blknheavi	10	1 024, 1 024	normal. heaviside	9.8e+02
[20] blurrycam	701	65 536, 65 536	blurring, wavelet	1.0e+04
[20] blurspike	702	16 384, 16 384	blurring	3.4e+02
cosspike	3	1 024, 2 048	DCT	2.2e+02
gausspike	11	256, 1 024	Gaussian ens.	2.4e+01
gcosspike	5	300, 2 048	Gaussian ens., DCT	1.8e+02
[27] jitter	902	200, 1 000	DCT	1.7e+00
[10] p3poly	6	600, 2 048	Gaussian ens., wavelet	1.7e+03
[11] sgnsipike	7	600, 2 560	Gaussian ens.	2.0e+01
[19] spiketrn	903	1 024, 1 024	1D convolution	1.3e+01
[37] soccer1	601	3 200, 4 096	binary, wavelet	4.2e+02
[37] soccer2	602	3 200, 4 096	binary, Haar wavelet	7.4e+02
srcsep1	401	29 166, 57 344	windowed DCT	1.0e+03
srcsep2	402	29 166, 86 016	windowed DCT	7.7e+03
srcsep3	403	196 608, 196 608	blurring, wavelet	1.0e+03
[20] yinyang	603	1 024, 4 096	wavelet	2.6e+02

According to [4] given ϵ_2 the parameter τ which makes problems BP_{ϵ_2} and BPDN equivalent, is the optimal Lagrange multiplier of the dual problem of BP_{ϵ_2} . Since, SPGL1.bpdn outputs both the primal iterates and the optimal Lagrange multiplier of BP_{ϵ_2} , it can be used to approximately find τ . Having such a parameter τ the BPDN solvers, matrix-free IPM, FPC_AS CG ℓ_1 - ℓ_s , PDCO and the BP_{ϵ_2} solver SPGL1.bpdn can be legitimately compared.

Moreover, in order to be able to compare the quality of the reconstructed representations for each solver when solving equivalent problems, the optimal sparsest representation for a particular level of noise needs to be known in advance. This is definitely not the case when noise is added manually by the user to a noiseless signal $\hat{b}$ using (42). Due to manual corruption of signal $\hat{b}$, the energy of the added noise $\epsilon_2 = \|e\|_2$ is known in advance. Hence, solving BP_{ϵ_2} will give the optimal sparsest representation for this particular level of noise, ϵ_2 . This solution is obtained by first calling SPGL1.bpdn solver to solve BP_{ϵ_2} by setting $\epsilon_2 = \|e\|_2$ with required high accuracy, see (41). During this process the approximate τ which makes problems BP_{ϵ_2} and BPDN equivalent is obtained from SPGL1.bpdn as has been described before. Hence, it is concluded that approximate τ and optimal sparse representations can be calculated such that a fair comparison can be conducted.

Finally, for noiseless signals $\hat{b}$, the problem is easier. Problems BP (3) and BPDN (5a) are almost equivalent for sufficiently small τ , i.e. 1.0e-10. However, such a small τ can make the ℓ_1 -norm in BPDN numerically negligible, see Figure 6.2 in [4] for numerical examples. For the former reason, if such a case is observed for BPDN solvers, parameter τ is set experimentally to a larger value, their values are given in Table 6.

5.4 Termination Criteria and Parameter Tuning

Termination of the compared solvers is forced when a solution of similar quality to the one of matrix-free IPM is obtained. In order to do so, the termination criteria of the compared solvers are changed. In particular, SPGL1 solver is terminated when the following criteria are satisfied

$$n1d(x_W^k) \leq n1d(x_W^m), \quad r.e(x_W^k) \leq r.e(x_W^m), \quad res = (x_W^k) \leq res(x_W^m),$$

where x_W^k is the projected representation at the k^{th} iteration of SPGL1 and x_W^m is the projected representation obtained by matrix-free IPM. Solvers FPC_AS, $\ell_1\text{-}\ell_s$ and PDCO are terminated when the following conditions are satisfied

$$obj(x_W^k) \leq obj(x_W^m), \quad r.e(x_W^k) \leq r.e(x_W^m).$$

Using these criteria for the compared solvers it is made certain that the reconstructed representations have approximately the same ℓ_1 -norm, ℓ_2 -norm of residual $Ax_W - b$ and number of non zero elements in x_W . The differentiation of the termination criteria for solver SPGL1 is done because SPGL1 solves problem BP_{e2}, unlike all other codes which solve the BPDN problem. Hence, it is more natural and fair for SPGL1 to be compared with other solvers using termination criteria in SPGL1 way.

Occasionally, certain solvers required too many matrix-vector products without achieving a solution of similar quality to the one delivered by the matrix-free IPM. In this case the solvers were terminated when $nMat > 40\,000$.

Regarding the parameter tuning of the compared solvers, all their parameters are set to their default values. For the matrix-free IPM the following parameters need to be set.

- tol: Relative duality gap of primal-dual pair (8) and (16). For noisy problems, this parameter varies between 1.0e-6 and 1.0e-10. For noiseless problems it varies between 1.0e-7 and 1.0e-14.
- maxiters: Maximum number of iterations. For all problems this parameter is set to 100.
- tolpcg: Tolerance of preconditioned CG method. For noisy problems this parameter varies between 1.0e-1 and 1.0e-2 and for noiseless ones it varies between 1.0e-1 and 1.0e-6.
- mxiterpcg: Maximum number of iterations of preconditioned CG method. For all problems this parameter is set to 200.

Since a large number of experiments has been performed, the exact parameter tuning of matrix-free IPM is not given here. However, it can be found in the MATLAB scripts which reproduce the results in this section, see <http://www.maths.ed.ac.uk/ERG0/mfipmcs/>.

Finally, the parameters ϵ_2 and τ in problems (5c) and (5a), respectively, for noisy problems are set as described in Subsection 5.3 for $\epsilon_2 = \|e\|_2$. For noiseless problems τ is set to arbitrarily small values given in Table 6.

5.5 Comparison

In this section we present computational results obtained for the Sparco collection problems discussed in the Benchmarks section. Both noisy and noiseless measurements are considered. Noise is added to measurements using (42) by fixing the SNR = 60 dB. A comparison among the previously mentioned solvers is made in terms of the quality of reconstruction and computational effort. The results of experiments are shown in Table 5. The first column in Table 5 shows the IDs of the Sparco problems. For each ID the first and second sub-rows give results for noisy and noiseless measurements, respectively. The second column reports the ℓ_1 -norm of the projected reconstructed representation for matrix-free IPM. The third column shows the relative error *r.e.*, see Table 3, of the projected reconstructed representation that was achieved by matrix-free IPM. The forth column shows the ℓ_2 -norm of the residual, denoted by *res* in Table 3, for matrix-free IPM. The rest of the table shows the number of matrix-vector products, *nMat*, that were needed by each solver to reconstruct a solution of similar quality to the one of matrix-free IPM. In cases when number of matrix-vector products required by a solver exceeded 40 000, the solver was terminated with a failure status. To be precise, it is a failure to converge to a solution similar to the one obtained by matrix-free IPM. Problems for which the matrix-free IPM converged with the lowest number of matrix-vector products among all solvers compared are denoted in bold. In Table 6 are shown the regularization parameters τ for noiseless signals that were used for BPDN for solvers matrix-free IPM, FPC_AS, ℓ_1 - ℓ_s and PDCO. Finally, for noiseless signals the version SPGL1_bp of SPGL1 solver is called.

One can observe in Table 5 that the matrix-free IPM was the fastest solver in 11 out of 36 noisy and noiseless problems, while it was the second fastest for another 14 problems, denoted by italic font. It is important to be mentioned that the performance of the compared solvers crucially depends on the condition number of matrices build of subsets of columns of matrix A with cardinality κ , less than m , i.e. full-rank sub-matrices of A . Unfortunately, it is a computational demanding task to check the condition number of every full-rank sub-matrix for the problems shown in Table 4. Nevertheless, by experimenting with a few sub-matrices one can get a picture of how well-conditioned sub-matrices of A might be.

Based on the previous criterion we observed that on problems that the matrix-free IPM was first or second, matrix A had relatively ill-conditioned

Table 5 Results for noisy and noiseless Sparco problems

				mfIPM	$\ell_1\text{-}\ell_s$	PDCO	FPC_AS	SPGL1
ID	$\ x\ _1$	$r.e$	res	$nMat$				
2	4.5e+02	5.3e-04	8.2e-02	61	726	6 611	9	40 000
	4.5e+02	1.0e-11	8.4e-10	65	644	40 011	40 002	21
3	2.2e+02	9.9e-04	1.3e-01	195	446	5 115	119	70
	2.2e+02	1.8e-08	1.8e-06	387	1 540	40 005	192	146
5	1.8e+02	3.0e-03	2.3e-01	1 367	5 042	28 369	630	510
	1.8e+02	2.4e-05	1.8e-03	6 239	20 758	41 479	636	40 000
6	1.7e+03	2.2e-02	1.7e+01	2 507	2 838	42 125	720	40 000
	1.7e+03	4.6e-02	3.6e+01	7 193	40 011	18 685	573	40 000
7	2.0e+01	5.4e-04	2.3e-03	165	452	955	78	63
	2.0e+01	5.6e-07	1.1e-06	259	952	709	78	87
9	4.1e+01	1.0e-03	1.7e-01	377	574	579	446	8 855
	4.1e+01	5.2e-12	1.6e-10	661	3 860	7 113	40 002	40 000
10	9.0e+02	9.3e-02	3.3e+00	2 431	11 421	1 043	40 001	40 000
	9.8e+02	1.0e-09	8.9e-08	4 519	8 192	42 647	40 001	40 000
11	2.4e+01	1.4e-03	1.3e-01	767	2 186	3 291	217	143
	2.4e+01	6.8e-05	5.2e-03	1 241	4 542	4 299	219	189
401	1.0e+03	8.9e-02	1.2e-01	2 747	42 622	61 327	40 076	882
	1.0e+03	7.7e-02	9.7e-02	3 193	43 512	48 511	40 076	814
402	1.0e+03	1.0e-01	1.9e-01	4 393	46 458	44 169	40 078	517
	1.0e+03	8.1e-02	2.0e-01	4 991	49 122	43 845	40 078	617
403	7.6e+03	1.2e-02	7.1e-01	2 841	6 136	40 495	2 305	699
	7.7e+03	4.1e-03	9.2e-02	6 031	43 278	69 913	40 046	932
601	3.3e+02	6.1e-02	5.7e+01	1 179	14 684	40 153	40 080	40 000
	4.0e+02	3.9e-02	4.8e+00	4 409	9 664	43 369	40 076	1 116
602	5.9e+02	1.0e-01	4.8e+01	1 199	17 097	40 631	40 023	898
	6.4e+02	1.1e-01	3.2e+00	4 669	22 392	42 139	40 043	40 000
603	2.6e+02	4.1e-03	4.2e-02	1 777	40 693	50 369	40 002	443
	2.5e+02	4.6e-02	5.9e-01	3 545	2 350	40 181	338	95
701	9.1e+03	4.6e-02	1.5e-01	1 217	33 160	91 147	40 044	1 658
	1.0e+04	2.4e-07	4.1e-03	1 907	4 722	49 093	40 001	40 000
702	3.4e+02	4.8e-03	3.4e-03	711	1 600	5 525	40 001	40 000
	3.4e+02	6.4e-08	2.4e-03	1 913	3 030	49 009	40 037	12 388
902	1.7e+00	5.3e-04	5.2e-04	143	498	237	40	49
	1.7e+00	2.0e-06	9.6e-07	239	675	279	42	59
903	1.3e+01	2.4e-03	1.4e-01	3 105	8 466	4 775	8 237	6 735
	1.3e+01	3.5e-06	1.9e-04	4 163	25 128	30 979	33 529	40 000

sub-matrices, at least for the ones that we experimented with. The previous implies that the proposed preconditioner was not as efficient as predicted in Section 4. However, the ill-conditioning also adversely affected the performance of SPGL1 and FPC_AS, as shown in Table 5. On the contrary, on problems that matrix A seemed to have well-conditioned sub-matrices, the preconditioner was very efficient, which resulted in a very fast matrix-free IPM. However, SPGL1 and FPC_AS were faster. For example, see problems with IDs 2, 3, 7 and 902.

Table 6 Regularization parameters τ for problem BPDN and noiseless measurements $\hat{b}$ for the experiments performed in Table 5

τ	Problems
1.0e-10	2, 9, 10, 701, 702
1.0e-08	401, 402, 603
1.0e-07	3, 7, 902
1.0e-05	903
1.0e-04	5, 403, 601, 602
1.0e-03	6
1.0e-02	11

5.6 Robustness to Noise

In this subsection we compare the matrix-free IPM with SPGL1, FPC_AS CG, $\ell_1\text{-}\ell_s$, in terms of their reconstruction capabilities for different levels of noise. The results collected in Table 5 and analysed in Section 5.5 reveal that PDCO and $\ell_1\text{-}\ell_s$ demonstrate comparable efficiency but the latter is usually faster. Therefore, solver PDCO will not be used in our further experiment.

For this experiment, the level of noise is varied from SNR = 10 dB to SNR = 120 dB with a step of 10 dB. The quality of reconstruction for all solvers is measured using the amplitude criterion [38]

$$\text{amp}(x_W) = \frac{\sqrt{\frac{1}{n}\|x_W - \hat{x}\|_2^2}}{\sqrt{\frac{1}{m}\|e\|_2^2}}.$$

The main purpose of using the *amp* criterion, instead of *r.e*, is that the former amplifies the *r.e*, the nominator of *amp*, as $\|e\|_2 \rightarrow 0$. Hence, less accurate representations will be emphasized.

As in Section 5.5 when the optimal representation $\hat{x}$ of BP is unknown it is calculated approximately using solver SPGL1_bp with required high accuracy (41). In order to have a fair comparison it is necessary to know at least approximately the parameter τ which makes problems BP_{ϵ_2} and BPDN equivalent and moreover, the optimal sparse representation of BP_{ϵ_2} for $\epsilon_2 = \|e\|_2$. The former issues are solved as described in Subsection 5.3.

To compare the solvers the following criterion is defined

$$\text{rampd}(x_W) = \frac{\max(\text{amp}(x_W^*) - \text{amp}(x_W^s), 0)}{\text{amp}(x_W^s)}, \quad (43)$$

where *rampd* stands for relative amplitude difference, x_W^* is the reconstructed projected representation by solvers matrix-free IPM, FPC_AS CG, $\ell_1\text{-}\ell_s$, and x_W^s is the reconstructed projected representation of solver SPGL1_bpdn. Notice that if *rampd* equals zero, then the representation x_W^* is of better quality than x_W^s , otherwise the inverse is true.

Table 7 Average quality reconstruction results over SNR from 10 dB to 120 dB for solvers matrix-free IPM, FPC_AS and $\ell_1\text{-}\ell_s$ on Sparco problems in Table 4

ID	Avg. <i>rampd</i> for SNR from 10 dB to 120 dB			Avg. <i>rampd</i> for SNR from 10 dB to 60 dB		
	mfIPM	FPC_AS	$\ell_1\text{-}\ell_s$	mfIPM	FPC_AS	$\ell_1\text{-}\ell_s$
2	6.1e-09	2.5e-10	0.0e+00	2.8e-13	2.6e-13	0.0e+00
3	1.3e-04	8.7e-05	0.0e+00	4.0e-09	6.4e-14	0.0e+00
5	5.1e-06	5.0e-07	1.7e-01	7.1e-11	9.8e-07	0.0e+00
6	1.2e-07	2.4e-10	1.1e+00	2.5e-07	4.8e-10	0.0e+00
7	1.5e-02	5.7e-08	0.0e+00	4.3e-06	3.5e-15	0.0e+00
9	1.1e-08	1.1e-01	4.9e-06	2.1e-08	2.1e-01	1.0e-08
10	7.3e-04	1.6e-01	0.0e+00	1.5e-03	2.5e-01	0.0e+00
11	4.2e-05	1.8e-05	0.0e+00	1.4e-10	3.7e-12	0.0e+00
401	8.6e+00	1.2e+01	8.5e+00	1.8e-01	1.9e-01	1.7e-01
402	8.0e+00	2.2e+01	8.0e+00	2.0e-01	1.9e+01	2.1e-01
403	1.9e+00	3.8e+00	1.2e+00	8.1e-03	6.4e-12	1.4e-02
601	3.8e+05	4.0e+03	1.5e+01	4.8e-11	8.1e+03	1.9e-01
602	1.6e+00	2.9e+03	7.4e+00	1.1e-10	5.7e+03	1.3e-01
603	8.1e-01	6.7e+00	7.7e-01	2.2e-08	3.7e-01	1.8e-03
701	6.4e-08	1.9e+00	3.2e-03	0.0e+00	3.8e+00	6.4e-03
702	7.9e-02	2.5e+01	6.2e-03	0.0e+00	5.1e+01	1.3e-03
902	9.1e-02	9.7e-09	0.0e+00	2.1e-07	1.3e-08	0.0e+00
903	1.5e-04	3.8e+00	1.5e-04	1.0e-11	7.5e+00	0.0e+00

In Table 7 is shown the average value of *rampd* over all SNR for each solver. The first column of Table 7 reports the ID of every Sparco problem. From the second to the forth column the average *rampd* over all SNR for each solver is shown. The last three columns report the average *rampd* for SNRs from 10 dB to 60 dB for each solver. Notice in Table 7 that matrix-free IPM for problems with IDs 2 to 11 and 701 to 903 was consistently recovering a high quality solution. For problems with IDs 401 to 603 for SNR > 60 dB all BPDN solvers, matrix-free IPM, FPC_AS CG and $\ell_1\text{-}\ell_s$, were unable to reconstruct an adequate representation and this is in contrast to SPGL1. A similar observation has been reported in [4]. In this work the authors mentioned that this issue of BPDN solvers might be due to very small regularisation parameter τ , obtained from SPGL1 solver as the energy of noise is decreased. In this case, the regularization effect of the ℓ_1 -norm starts to be negligible and the solvers face considerable numerical difficulties. However, in our experiments we observed for these problems that not always the τ parameter was small and additionally, there were other problems where τ was even smaller but successful reconstruction was possible. Therefore, we conclude that this failure of BPDN solvers might be problem dependent.

5.7 Preconditioned Conjugate Gradient Method Against Direct Linear Solver

In this subsection we replace PCG in steps 8 and 10 of matrix-free IPM with a direct linear solver. It has been mentioned in Section 1 that direct linear solvers are efficient when the system to be solved is sufficiently sparse. However, for CS the systems (19) to be solved are completely dense due to density of matrix A . For this reason, large scale problems are not storable in a moderate computer with 8 giga byte of random access memory. Even worse, matrix A might be an algorithmic operator, i.e. DCT, therefore, direct solvers cannot be employed. Hence, direct linear solvers for CS inside an IPM are only applicable when the measurement matrix A is explicitly available, i.e. Gaussian matrix, and only for small scale problems, i.e. $n = 2^{12}$ or smaller. In addition to the former disadvantages of a direct solver for CS problems, its computational complexity for systems (19) will be of order $\mathcal{O}(n^3)$. This is a well known result, for completely dense linear systems. Therefore, it is expected that for very small instances the two approaches might require similar CPU time to converge, while as dimensions grow the CPU time of the IPM version with the direct linear solver will increase rapidly. Indeed, this is confirmed by Figure 3a.

Despite the higher computational effort required by direct solvers for CS problems, such an approach will produce exact Newton directions, hence, one would expect that IPM iterations will be the minimum possible. Suprisingly, in Figure 3b we show, that matrix-free IPM with PCG requires as few iterations as its IPM version with a direct linear solver. Indeed, recent analysis of [26] indicates that allowing the use of inexact Newton directions in an IPM does not adversely affect the worst-case complexity result of this method.

In the experiments reported in Figures 3a and 3b matrix A is Gaussian, the sparsity pattern of the optimal representation $\hat{x}$ is chosen at random, while the nonzero components follow a standard normal distribution. The noiseless measurements are produced by $\hat{b} = A\hat{x}$. The size of problem n , is varied from 2^5 to 2^{12} with a step of times 2, the measurements m are varied from 2^3 to 2^{10} with a step of times 2 and the sparsity level k is set to $\lceil m/20 \rceil$. Finally, the τ parameter in BPDN problem (5a) is set to $\tau = 1.0\text{e-}3$. To solve the linear systems we use the `mldivide` function of MATLAB, which in case of symmetric real matrices with positive diagonal, i.e. (19), performs Cholesky factorisation. For details of the `mldivide` function we refer the reader to <http://www.mathworks.co.uk/help/matlab/math/systems-of-linear-equations.html>.

5.8 Average Phase Transition

Recently, it has been shown in [18] that for any problem instance (A, b) , where A is Gaussian, there is a maximum ratio $\bar{\nu}_\rho = k/m$ given $\rho = m/n$ that below of it the problems (3) or (5a) guarantee on average reconstruction of the optimal sparse representation. The latter, has been introduced as the notion of average phase transition for Gaussian matrices. Moreover, it has been shown

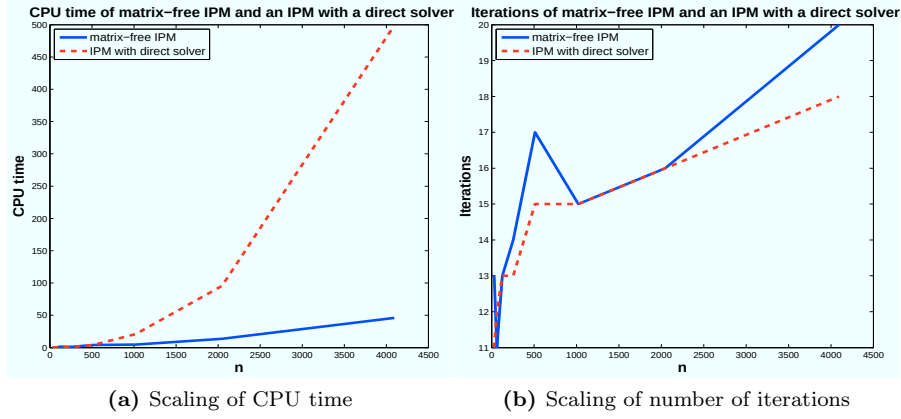


Fig. 3 Scaling of CPU time and number of iterations as the size of problem n grows for matrix-free IPM and an IPM in which the PCG is replaced with a direct solver

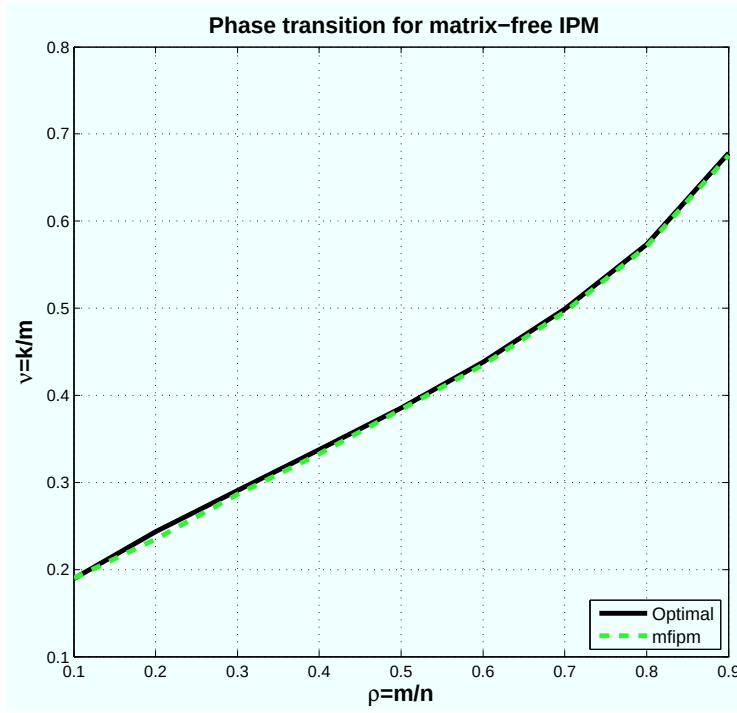


Fig. 4 Empirical phase transition for matrix-free IPM. The solid curve denotes the theoretically optimal phase transition. The dashed curve denotes the empirical phase transition for 50% success rate of matrix-free IPM

empirically that other measurement matrices such as partial Fourier, partial Hadamard, Bernoulli etc, have the same average phase transition properties. Ideally, an efficient ℓ_1 -regularization solver should have empirical average phase transition at the same level $\bar{\nu}_\rho$.

In this section we show that the empirical phase transition properties of matrix-free IPM fit the average Gaussian phase transition properties by reproducing a similar experiment as that in Section 2 of [18]. Let us now explain the experiment. The parameter n is fixed to $n = 1000$. The measurements m are varied from $m = 100$ to $m = 900$ with a step of 100. For each of the nine measurements m the sparsity of the optimal representation is varied from $k = 1$ to $k = m$ with a step of one and for each k , 100 trials are conducted. The sensing matrix A is chosen by taking randomly m rows from an $n \times n$ normalized discrete cosine transform matrix. Each nonzero coefficient of the sparse representation is set to ± 1 with equal probability, while the sparsity pattern is chosen at random. All the generated problems are solved using the matrix-free IPM solver, the reconstruction is considered successful when $r.e \leq 1.0e-5$. For each ratio ν_ρ we compute the success ratio $p(\nu_\rho) = S/100$, where S is the number of trials for which the $r.e \leq 1.0e-5$. It has been demonstrated empirically in [18] that for any problem instance (A, b) , where A is a partial DCT matrix a solver with average phase transition properties has $\max\{\nu_\rho \mid p(\nu_\rho) \geq 0.5\} \approx \bar{\nu}_\rho$. The latter means that the empirical average phase transition for 50% success rate overlaps with the theoretical average phase transition for Gaussian matrices. In Figure 4, we plot the empirical phase transition for 50% success rate of matrix-free IPM and the theoretical average phase transition. The two curves overlap.

6 Conclusions

We propose and implement a computationally inexpensive matrix-free primal-dual interior point method, based on [25] and [23], for the ℓ_1 -regularized problems arising in the field of Compressed Sensing. At every iteration of the proposed primal-dual interior point method the direction is obtained by solving the linear system (19a) using the conjugate gradient method. Unfortunately, the matrices $\Theta^{-1} + FF^\top$ in these systems tend to be ill-conditioned as the algorithm converges, hence, the conjugate gradient method might get slow. To remedy this ill-conditioning we propose a low-cost preconditioner for the conjugate gradient method. The proposed preconditioning technique exploits features of Compressed Sensing matrices as well as interior point methods. Its efficiency is justified theoretically and confirmed in numerical experiments.

Computational experience presented in this paper shows that although the Compressed Sensing research community seems to favor first-order methods, a specialized (matrix-free) interior point method is very competitive and offers a viable alternative.

References

1. N. Ahmed and K. R. Rao. *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag, Berlin, 1975.
2. R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.
3. S. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1):1–39, 2011.
4. E. Van Den Berg and M. P. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal of Scientific Computing*, 31(2):890–912, 2008.
5. E. Van Den Berg, M. P. Friedlander, G. Hennenfent, F. J. Herrman, R. Saab, and Ö. Yilmaz. Sparco: A testing framework for sparse reconstruction. *ACM Trans. Math. Software*, 35(4):1–16, 2009.
6. J. D. Blanchard, C. Cartis, and J. Tanner. Decay properties of restricted isometry constants. *Computational Optimization and Applications*, 16(7):572–575, 2009.
7. J. D. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: How sharp is the restricted isometry property? *SIAM Rev.*, 53(1):105–125, February 2011.
8. E. Candès, L. Demanet, D. Donoho, and L. Ying. Curvelab. <http://www.curvelet.org/software.html>.
9. E. J. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, 2006.
10. E. J. Candès and J. Romberg. Practical signal recovery from random projections. In *Wavelet Applications in Signal and Image Processing XI, Proc. SPIE Conf. 5914.*, 2004.
11. E. J. Candès and J. Romberg. ℓ_1 -magic. *Technical Report, Caltech*, 2007. <http://users.ece.gatech.edu/~justin/l1magic/>.
12. E. J. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007.
13. E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006.
14. S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.
15. R. Coifman, F. Geshwind, and Y. Meyer. Noiselets. *Appl. Comput. Harmon. Anal.*, 10(1):27–44, 2001.
16. D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inf. Theory*, 47(7):2845–2862, 2001.
17. D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
18. D. L. Donoho and J. Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98(6):913–924, 2010.
19. C. Dossal and S. Mallat. Sparse spike deconvolution with minimum scale. In *Proceedings of Signal Processing with Adaptive Sparse Structured Representations*, 81(3):123–126, 1994.
20. M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse Problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.
21. S. Foucart. A note on guaranteed sparse recovery via ℓ_1 -minimization. *Applied and Computational Harmonic Analysis*, 29(1):97–103, 2010.
22. M. Frigo and S. G. Johnston. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.
23. J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.
24. J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.
25. J. Gondzio. Matrix-free interior point method. *Computational Optimization and Applications*, 51(2):457–480, 2012.

26. J. Gondzio. Convergence analysis of an inexact feasible interior point method for convex quadratic programming. *SIAM Journal on Optimization*, 23(3):1510–1527, 2013.
27. G. Hennenfent and F. J. Herrmann. Random sampling: new insights into the reconstruction of coarsely-sampled wavefields. In *SEG International Exposition and 77th Annual Meeting*, 2007.
28. C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*, volume 16 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1995.
29. S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1 -regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.
30. M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Mathematical Programming*, 61:263–280, 1993.
31. Z. Lu, R. D. S. Monteiro, and J. W. O’Neal. An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming. *SIAM Journal on optimization*, 17:287–310, 2006.
32. A. J. Miller. *Subset selection in regression*. Chapman & Hall/CRC, London, 2002.
33. Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.
34. R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics, Princeton University Press, 1970.
35. M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.
36. M. Saunders and B. Kim. PDCO: Primal-dual interior method for convex objectives. *Technical Report, Stanford University*, 2002. <http://www.stanford.edu/group/SOL/software/pdco.html>.
37. D. Takhar, J. N. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. K. Kelly, and R. G. Baraniuk. A new camera architecture based on optical-domain compression. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Computational Imaging*, 6065, 2006.
38. A. Thomson. Compressive single-pixel imaging. *Proceedings of the 2nd IMA Conference on Mathematics in Defence, Defence Academy, Shrivenham, UK*, 2011.
39. Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation. *SIAM J. Sci. Comput.*, 32(4):1809–1831, 2010.
40. S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.