SUPPLEMENTARY MATERIALS: AN ALGORITHM SOLVING COMPRESSIVE SENSING PROBLEMS BASED ON MAXIMAL MONOTONE OPERATORS*

YOHANN TENDERO[T], IGOR CIRIL[‡], JÉRÔME DARBON[§], AND SUSANA SERNA[¶]

SM1. Comparison of solutions of AISS, LARS, SPGL1, SeDuMi and algorithm 2.1 in terms of ℓ^1 norm. We wish to compare the ℓ^1-based methods considered in this paper, i.e., AISS [SM7], LARS [SM18], SPGL1 [SM40, SM41], SeDuMi [SM38] and algorithm 2.1 beyond the "probability of success" (4.1). Since the minimizer(s) to (P_{ℓ^1}) may not be unique we compare the ℓ^1 norms of the solutions produced by the above algorithms. To so do, let $\boldsymbol{b}^i$ denote the i-th observed vector and let $\boldsymbol{u}_k^i$ denote the i-th output of the algorithm k. We compare the algorithms by computing the solution with the smallest ℓ^1-norm among feasible solutions. This yields to the following formula to compute the score S_k of algorithm k with

$$S_k := \frac{1}{\#\text{ of tests}} \sum_i \mathbb{1}_{\{\|A\boldsymbol{u}_k^i - b\|_{\ell^2} \leqslant \varepsilon\}}(i) \mathbb{1}_{\{\|\boldsymbol{u}_k^i\|_{\ell^1} = \min_k \|\boldsymbol{u}_k^i\|_{\ell^1}\}}(i), \tag{SM1.1}$$

where, as usual, $\varepsilon := 1e-10$. The results of (SM1.1) applied to the above algorithms are depicted in Figure SM1. From this figure, we observed that algorithm 2.1 the one with highest score.

SM2. Extension for the noisy case. For applications where the observed data $\boldsymbol{b}$ is corrupted by a random Gaussian perturbation one is often interested in the following optimization problem

$$\begin{cases} \inf\limits_{\boldsymbol{u}\in\mathbb{R}^n} & \|\boldsymbol{u}\|_{\ell^1} \\ \text{s.t.} & \|A\boldsymbol{u} - \mathbf{b}\|_{\ell^2} \leqslant \epsilon \end{cases}$$

that can equivalently be formulated as follows for all $\epsilon \geqslant 0$

$$\begin{cases} \inf\limits_{\boldsymbol{e}\in\mathbb{R}^m} & G(\boldsymbol{e}) \\ \text{s.t.} & \|\boldsymbol{e}\|_{\ell^2} \leqslant \epsilon, \end{cases} \tag{SM2.1}$$

where

$$G(\boldsymbol{e}) := \inf_{\substack{\boldsymbol{u}\in\mathbb{R}^n \\ A\boldsymbol{u}-\boldsymbol{b}=\boldsymbol{e}}} \|\boldsymbol{u}\|_{\ell^1}. \tag{SM2.2}$$

A Lagragrian-based approach can be used for solving SM2.1. Consider the Lagrangian $\mathcal{L}_G : \mathbb{R}^m \times [0, +\infty) \to \mathbb{R} \cup \{+\infty\}$ defined as

$$\mathcal{L}_G(\boldsymbol{e}, \mu) = G(\boldsymbol{e}) + \mu(\|\boldsymbol{e}\|_{\ell^2} - \epsilon).$$

*Submitted to the editors DATE. A preliminary version of the work is [SM13].

Funding: Research of J. Darbon has been supported by the National Science Foundation under Grant No. NSF-1820821.

[T]DR2I, Institut Polytechnique des Sciences Avancées, 94200, Ivry-sur-Seine, France yohann.tendero@ipsa.fr.

[‡]DR2I, Institut Polytechnique des Sciences Avancées, 94200, Ivry-sur-Seine, France igor.cirl@ipsa.fr.

[§]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

[¶]Departament de Matematiques, Universitat Autonoma de Barcelona, Bellaterra, Barcelona, Spain susana.serna@uab.cat

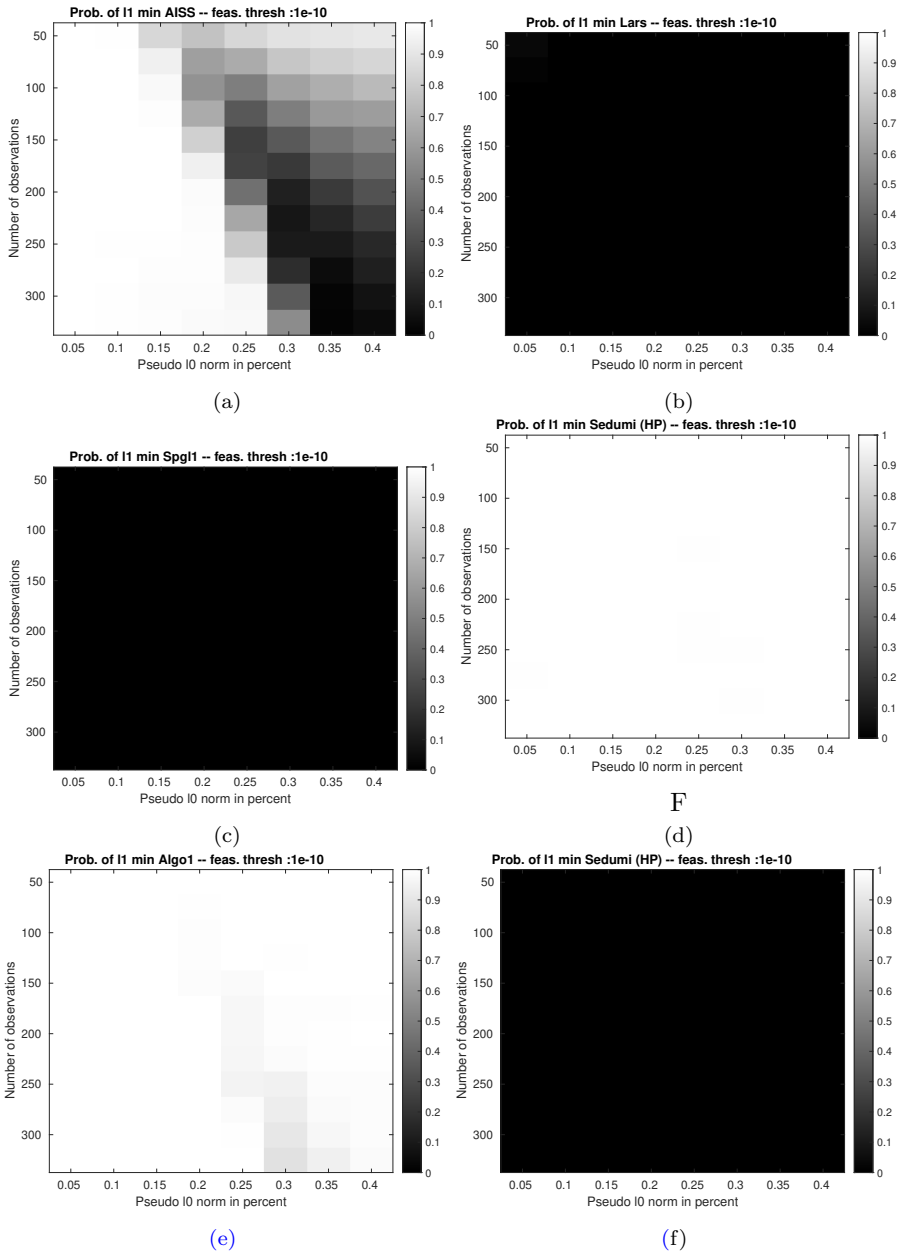


FIG. SM1. *Algorithms ranking: ℓ^1-minimization. This figures compares the quality of produced solutions by the considered algorithms using* (SM1.1). *A white color means a high score and a black color a low score. Panel (a): AISS [SM7], panel (b): LARS [SM18], panel (c): SPGL1 [SM40, SM41], panel (d): SeDuMi (SP) [SM38] panel (e): algorithm 2.1, panel (f): SeDuMi (HP) [SM38].*

Existence of a saddle point of this Lagrangian easily follows from the mini-max theorem [SM3, chap. 6] combined with our assumptions. We can proceed as follows to find a saddle-point of the Lagrangian $\mathcal{L}_G$. We can minimize $\mathcal{L}_G(\cdot, \mu)$ when μ is fixed using a black-box method as described in [SM1, section 8.3.1] (see below for a justification that the black-box can be computed using our algorithm). Then, we

can perform a bitonic-search on $\mathcal{L}_G(\boldsymbol{e},\cdot)$ to find the optimal Lagrange multiplier. We intend to program and test this approach and to provide details in our next paper.

We now briefly justify how to implement the black-box for G using our proposed algorithm. Following [SM1, section 8.3.1], a black-box for G, given $\boldsymbol{e}\in\mathbb{R}^m$, consists of computing $G(e)$ and $\boldsymbol{s}\in\partial G(\boldsymbol{e})$. We now explain how to compute s using our proposed algorithm. Define the Lagrangian $\mathcal{L}_{\boldsymbol{e}}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ given by

$$\mathcal{L}_{\boldsymbol{e}}(\boldsymbol{u},\boldsymbol{\lambda}) := J(\boldsymbol{u}) + \langle\boldsymbol{\lambda}, A\boldsymbol{u}\rangle + \langle\boldsymbol{\lambda}, -\boldsymbol{b}\rangle + \langle\boldsymbol{\lambda}, -\boldsymbol{e}\rangle\,, \tag{SM2.3}$$

we immediately obtain for all $\boldsymbol{e}\in\mathbb{R}^m$

$$G(\boldsymbol{e}) = -\inf_{\boldsymbol{\lambda}\in\mathbb{R}^m}\left\{-\inf_{\boldsymbol{u}\in\mathbb{R}^n}\mathcal{L}_e(\boldsymbol{u},\boldsymbol{\lambda})\right\} = -\inf_{\boldsymbol{\lambda}\in\mathbb{R}^m}\{g(\boldsymbol{\lambda})-\langle\boldsymbol{\lambda},-\boldsymbol{e}\rangle\} = g^*(-\boldsymbol{e}). \tag{SM2.4}$$

Consequently, combining (SM2.3)-(SM2.4) and [SM4, prop. 11.3 p. 476] we have

$$\partial G(\boldsymbol{e}) = \partial g^*(-\boldsymbol{e}) = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^m}\{g(\boldsymbol{\lambda})-\langle\boldsymbol{\lambda},-\boldsymbol{e}\rangle\}\,. \tag{SM2.5}$$

We immediately notice that the problem in (SM2.5) has the same form as the dual problem (D_{ℓ^1}) that we considered in this paper. Our algorithm allows us to efficiently compute an optimal $\bar{\boldsymbol{\lambda}}$ up to the machine precision by running our algorithm on the input vector $\boldsymbol{b}+\boldsymbol{e}$.

SM3. Extension to inequality constraints. Note that our proposed approach can be extended to handle inequality constraints. This is easily achieved using well-known equivalent reformulation of optimization problems (see e.g., [SM2, section 4.1.3]). Let us first consider the case of an inequality constraint that reads $\langle a,u\rangle - b\leqslant\epsilon$ where $\epsilon\in[0,+\infty)$. The above inequality is equivalent to the following equality constraint $\langle a,u\rangle - b+\gamma=\epsilon$ where $\gamma\geqslant 0$. Therefore any affine inequality constraints of the form can be mapped to an equality constraints using an extra variable. The above procedure trivially extends to any finite number of constraints that read $\langle a_i,u\rangle - b\leqslant\epsilon$ with $i=1,\ldots,M$ where $M\in\mathbb{N}$. As an application of the above procedure, we can for instance deal with constraints that read $\|Au-b\|_{\ell^\infty}\leqslant\epsilon$ or $\|Au-b\|_{\ell^1}\leqslant\epsilon$. Note that $\|\cdot\|_\infty$ or $\|\cdot\|_1$ norms can be replaced by any polyhedral norms (that can include weights) using the same procedure.

SM4. Additional Experiments. This section proposes an empirical evaluation of: OMP [SM35], CoSamp [SM33], GISS [SM32]. Note that OMP, CoSamp and GISS are greedy-based numerical algorithms while LARS, SPGL1, AISS and Algorithm 2.1 developed in the paper corresponding to this supplementary material are ℓ^1-based numerical algorithms. (Comparisons between LARS, SPGL1, AISS and Algorithm 2.1 are in section 4).

These methods are compared in terms of a "probability of success", as in section 4. The experimental setup is the same as in section 4. The implementation of CoSamp and OMP we used are due to S. Becker (code updated on Dec 12th, 2012) and can be found on Mathworks file exchange[1]. The implementation of GISS we used are the ones given by the authors of [SM7, SM32]. Note that the implementation of CoSamp requires an estimate of the sparsity. In all the experiments we used an oracle for CoSamp, i.e., provided the exact sparsity of the source element. Default parameters have been used for all methods. We now give the criteria used for the

[1] https://fr.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery

numerical comparisons of these numerical algorithms. We recall that a reconstruction is *a success* if the relative error satisfies $\frac{\|\boldsymbol{u}-\boldsymbol{u}_{est}\|_{\ell^2}}{\|\boldsymbol{u}\|_{\ell^2}} < \varepsilon$, where $\varepsilon = 10^{-10}$ or $\varepsilon = 10^{-4}$ as in section 4. As in section 4 equation (4.1), the empirical probability of success is given by

$$P_{(m,s)} := \frac{1}{\#\text{ of tests}} \sum_i \mathbb{1}_{\left\{\frac{\|\boldsymbol{u}^i-\boldsymbol{u}^i_{est}\|_{\ell^2}}{\|\boldsymbol{u}^i\|_{\ell^2}}<\varepsilon\right\}}(i).$$

Each method is tested on the same data by using the same random seed. We first consider $\varepsilon = 10^{-10}$. Figure SM2 depicts the empirical probability of success (4.1) for OMP, CoSamp, and GISS algorithms. As in section 4, we consider the difference of probability of success between Algorithm 2.1 and all other methods

$$D_{(m,s)} := P^{Algorithm\ 2.1}_{(m,s)} - P_{(m,s)},$$

where $m \in M,\ s \in S$, $P^{Algorithm\ 2.1}_{(m,s)}$ (resp. $P_{(m,s)}$) denotes the quantity (4.1) obtained with Algorithm 2.1 (resp. OMP, CoSamp or GISS). Note that a positive (negative) value in (4.2) means that Algorithm 2.1 achieves a higher (lower) probability of success than the compared algorithm. These differences of probability of success are depicted in figure SM3. It is easily seen that OMP and CoSamp succeed at retrieving the source signal with a probability of approximately 80% for a larger set of parameters than any other method. However, they produce correct results with a probability of nearly 100% for a much smaller set of parameters than AISS, GISS, and Algorithm 2.1 (see section 4). We also deduce from figure SM3 that Algorithm 2.1 always achieves a higher probability of success than GISS. An overview of the main similarities and differences between OMP [SM35], CoSamp [SM33], GISS [SM32] and Algorithm 2.1 are given in table SM1. In table SM1, we give the empirical probability that at least x% of signals are successfully reconstructed for all methods. This statistical indicator is defined as in section 4

$$P_{\geqslant x} = \frac{\#\left\{(m,s) \in M \times S \ :\ P_{(m,s)} \geqslant x\right\}}{\#M \cdot \#S}, \tag{SM4.1}$$

where $P_{(m,s)}$ is defined by (4.1) and # denotes the cardinality of a set. We now report results concerning the number of iterations needed to achieve convergence.

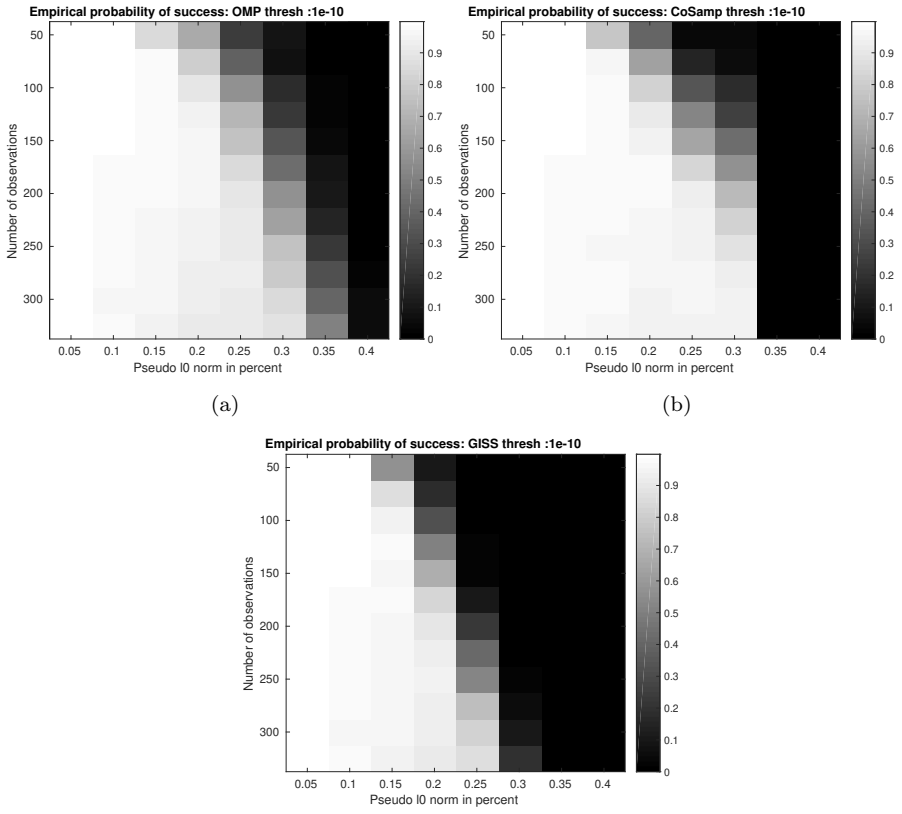
Figure SM4 gives the average number of iterations needed by the considered algorithms. OMP and GISS are much faster than Algorithm 2.1 with a maximal average number of iterations of (about) 250 and 300 respectively. The maximal average number of iterations needed by CoSamp fluctuates more depending on the sparsity. For CoSamp it remains below 300 for most experiments though and the maximal average number of iterations is bounded from above by $1,000$.

Figure SM5 depicts the empirical probability of success (4.1) for OMP, CoSamp, and GISS algorithms. Figure SM6 depicts the differences of probability of success.

REFERENCES

[SM1] J.-F. Bonnans, J. Gilbert, C. Lemarechal, and C. Sagastizabal, *Numerical Optimization*, Springer Berlin Heidelberg, 2006.

[SM2] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2014, https://doi.org/10.1017/CBO9780511804441, https://web.stanford.edu/%7Eboyd/cvxbook/.



(a) (b)

(c)

FIG. SM2. *Empirical probability of success* (4.1), *with* $\varepsilon = 10^{-10}$. *Panel (a): OMP [SM35], panel (b): CoSamp [SM33] and panel (c): GISS [SM32]. The non-zero entries of the source element* $\boldsymbol{u}$ *are drawn from a uniform distribution on* $[-1, 1]$. *The entries in* A *are drawn from i.i.d. realizations of a Gaussian distribution.*

TABLE SM1

Main assumption and statistical indicator of "success" for OMP, CoSamp and GISS methods. The numbers without parentheses correspond to $\varepsilon = 10^{-10}$ *and those between parentheses correspond to* $\varepsilon = 10^{-4}$. *Below, E.R.C. stands for exact recovery condition see, e.g., [SM35], R.I.C. stands for restricted isometry constant see, e.g., [SM33].*

Algorithm	**OMP** [SM35]	**CoSamp** [SM33]	**GISS** [SM32]
Assumption(s): $A/\boldsymbol{b}$	E.R.C.	R.I.C.	E.R.C.
$P_{\geqslant 0.9}$ (SM4.1)	0.5104 (0.5417)	**0.5521** (**0.5625**)	0.4063 (0.4167)
$P_{\geqslant 0.95}$ (SM4.1)	0.3854 (0.4479)	**0.4583** (**0.5208**)	0.3333 (0.3542)
$P_{\geqslant 0.99}$ (SM4.1)	0.1042 (0.1354)	**0.1250** (**0.1458**)	0.1042 (0.1354)
$P_{\geqslant 0.999}$ (SM4.1)	0 (0)	0 (0)	0 (0)
$P_{\geqslant 1}$ (SM4.1)	0 (0)	0 (0)	0 (0)

[SM3] I. EKELAND AND R. TEMAM, *Convex analysis and variational problems*, SIAM, 1999.
[SM4] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Variational Analysis*, Grundlehren der mathematischen Wissenschaften, Springer Berlin Heidelberg, 2009.

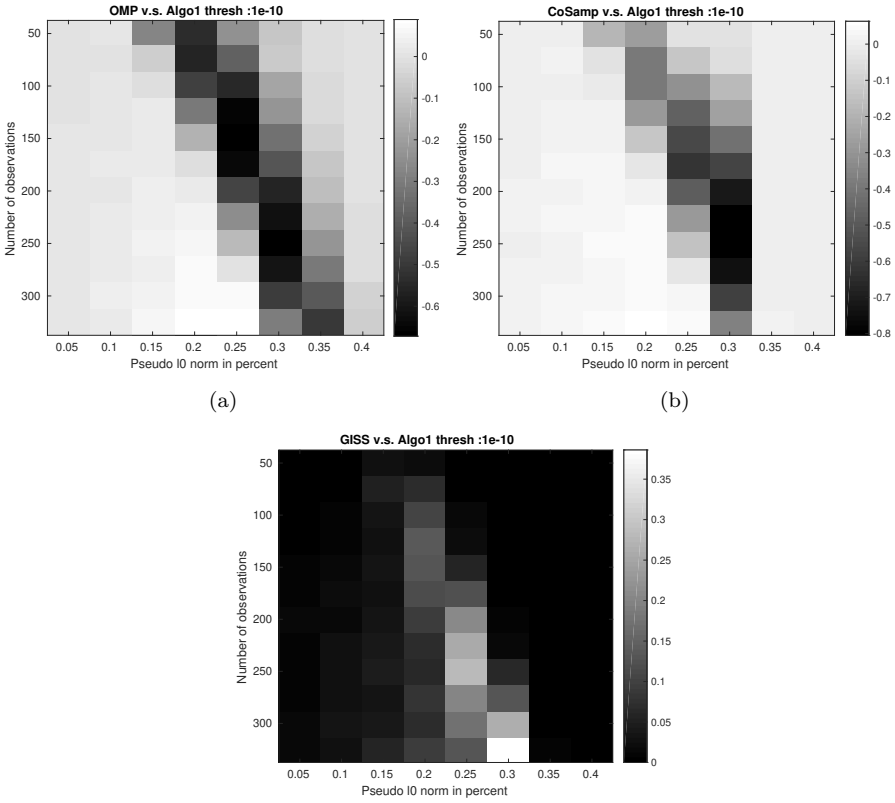


FIG. SM3. *Differences of probability of success* (4.2)*, with* $\varepsilon = 10^{-10}$*. Panel (a):* Algorithm 2.1*-OMP [SM35], panel (b):* Algorithm 2.1*-CoSamp [SM33] and panel (c):* Algorithm 2.1*-GISS [SM32]. A positive value indicates that* Algorithm 2.1 *achieves a higher probability of success than the considered method, a negative value the contrary.*

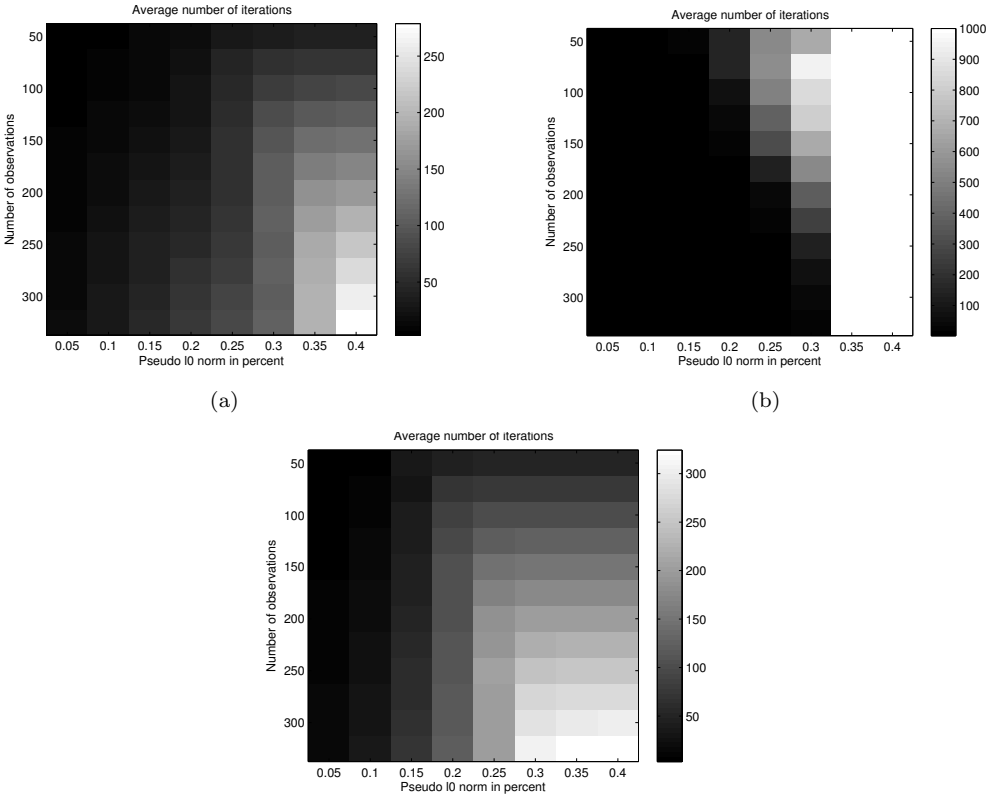
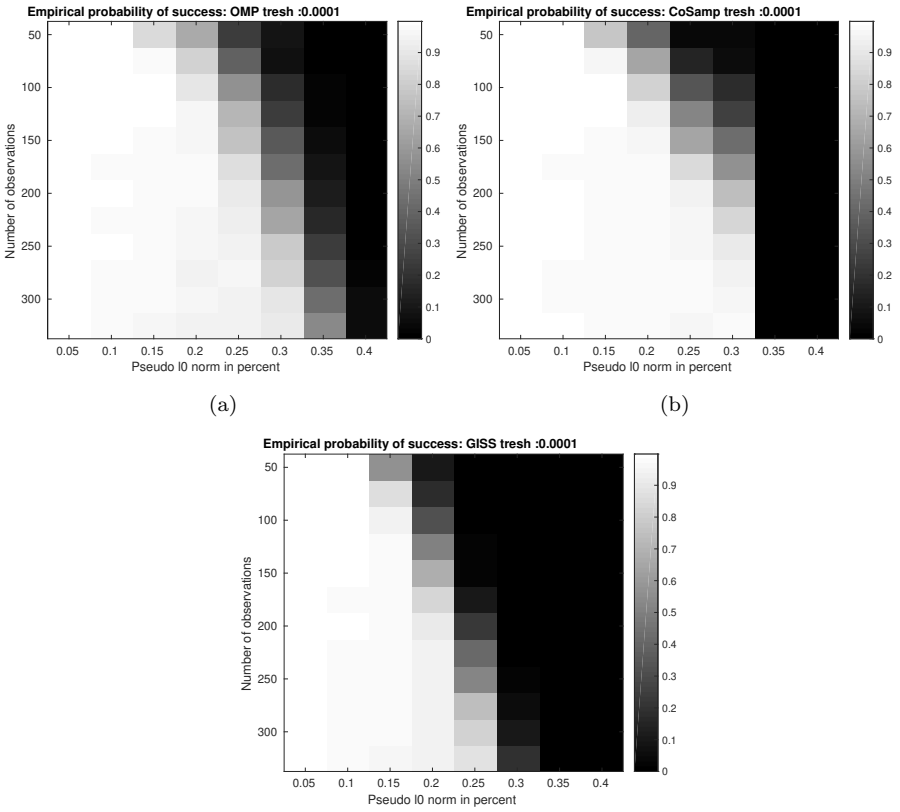


FIG. SM4. *Average number of iterations. Panel (a): OMP [SM35], panel (b): CoSamp [SM33] and panel (c): GISS [SM32].*



(a) (b)

(c)

FIG. SM5. *Empirical probability of success* (4.1), *with* $\varepsilon = 10^{-4}$. *Panel (a): OMP [SM35], panel (b): CoSamp [SM33] and panel (c): GISS [SM32]. The non-zero entries of the source element* $\boldsymbol{u}$ *are drawn from a uniform distribution on* $[-1, 1]$. *The entries in* A *are drawn from i.i.d. realizations of a Gaussian distribution.*

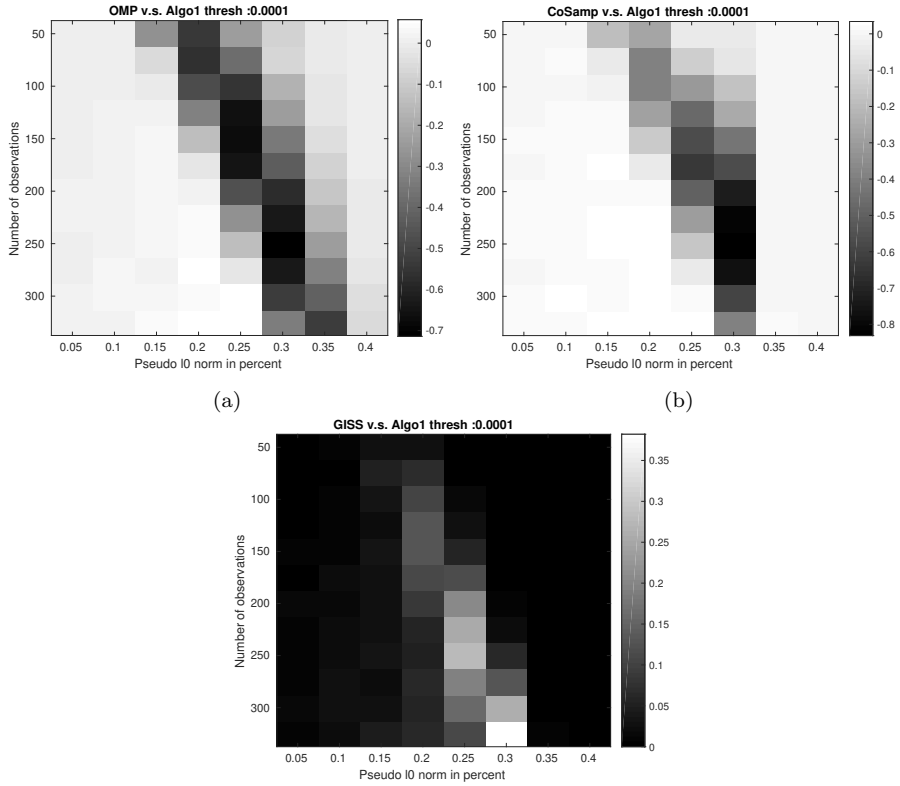


FIG. SM6. *Differences of probability of success (4.2), with $\varepsilon = 10^{-4}$. Panel (a): Algorithm 2.1-OMP [SM35], panel (b): Algorithm 2.1-CoSamp [SM33] and panel (c): Algorithm 2.1-GISS [SM32]. A positive value indicates that Algorithm 2.1 achieves a higher probability of success than the considered method, a negative value the contrary.*