INEXACT COORDINATE DESCENT

Rachael Tappenden (Joint work with Jacek Gondzio & Peter Richtárik)

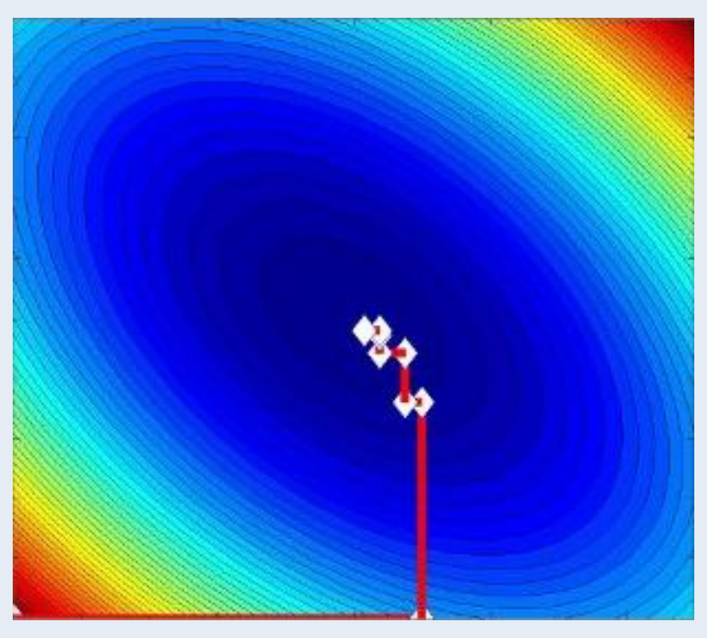
1. OVERVIEW

We extend the work of Richtárik and Takáč [1] and present a block coordinate descent method that employs **inexact** updates applied to the problem of minimizing the convex composite objective function

$$\min_{x\in\mathbf{R}^N}\{F(x) := f(x) + \Psi(x)\}, \tag{1}$$

where f is smooth and convex and Ψ is convex, (possibly) nonsmooth and (block) separable. In this work we:

- Introduce the Inexact Coordinate Descent (ICD) Method
- Provide convergence guarantees (iteration complexity results)
- Study the special case where f is quadratic and $\Psi = 0$:
  - Find the block update via conjugate gradients
  - Use preconditioning to accelerate the update step



Randomized coordinate descent in 2D.

Why is this work important?

We can compute inexact updates more quickly!
$\Rightarrow$ Reduction in the algorithm running time!

2. MOTIVATION & APPLICATIONS

Optimization problems are becoming increasingly large and new tools and algorithms are needed to solve them efficiently. Coordinate descent (CD) methods are a natural choice for very large-scale problems because:

- their memory requirements are low
- they have low per-iteration computational cost
- they take advantage of the underlying block structure

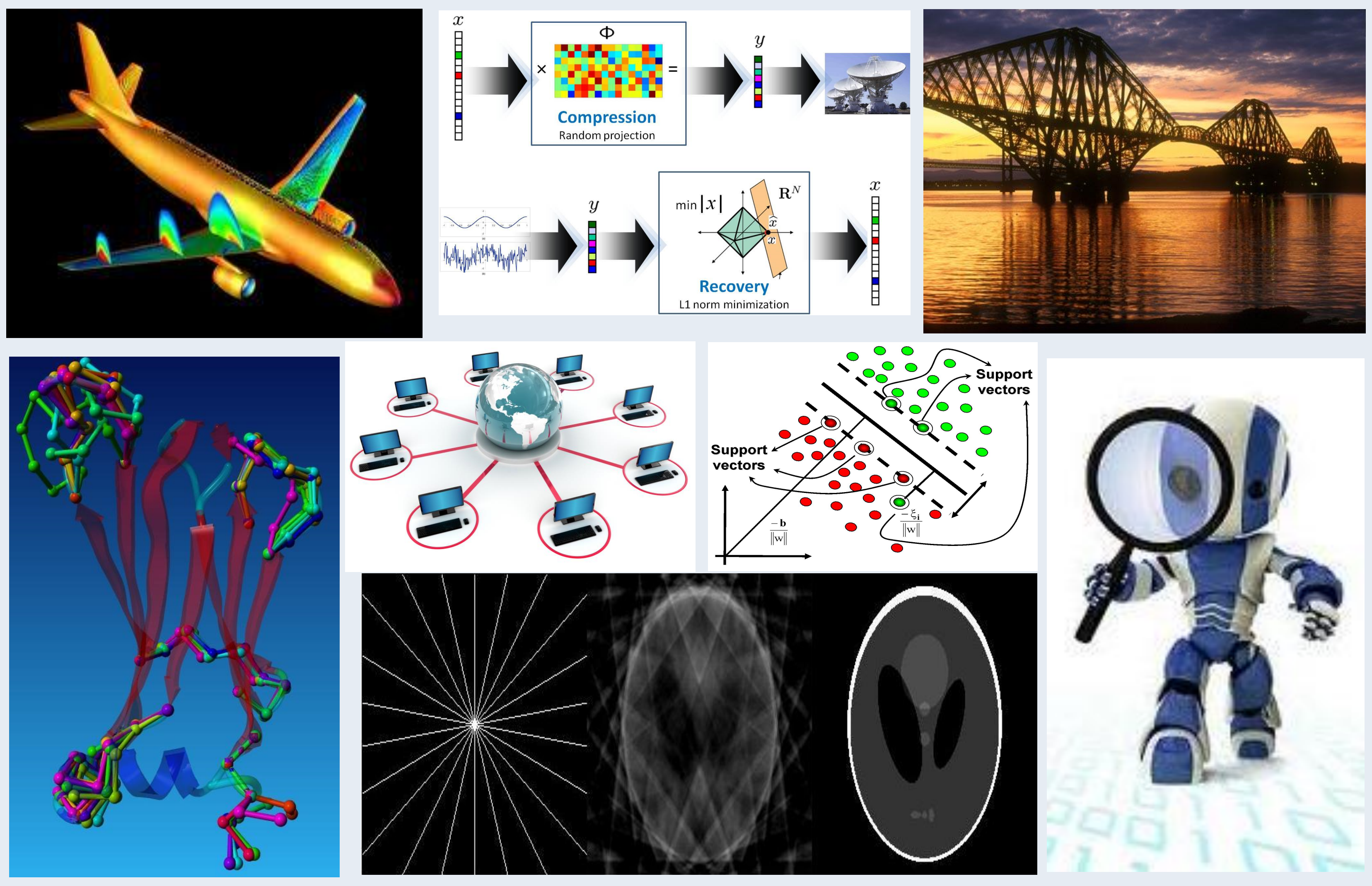


Figure 1. Applications for coordinate descent methods: Clockwise from top left: Aircraft design and timetabling, compressed sensing, truss topology design (Firth of Forth Railbridge), machine learning, image reconstruction, protein loop closure, distributed data, support vector machines.

3. THE ALGORITHM

Variants of coordinate descent methods differ in 2 key ways:

1. How they choose the block to update $\Rightarrow$ ICD chooses the block randomly
2. How they find the update $\Rightarrow$ ICD minimizes an overapproximation of F

Preliminaries: We take a permutation of the identity matrix U and partition it into n (different sized) blocks: $U = [U_1, \dots, U_n]$. We access the ith block of x via $x^{(i)} = U_i^T x$. We assume that the gradient of f is blockwise Lipschitz with constants $L_1, \dots, L_n$ and for block i we choose a positive definite matrix B_i.

Finding the update t: The function F might be complicated, so we work with the overapproximation:

$$F(x + U_i t) \le f(x) + \underbrace{\langle \nabla_i f(x), t\rangle + \tfrac{L_i}{2}\langle B_i t, t\rangle + \Psi_i(x^{(i)} + t)}_{V_i(x,t)} + \sum_{j\neq i} \Psi_j(x^{(j)}). \tag{2}$$

- **Exact CD $\Rightarrow$ exactly minimize (2), i.e., the update is: $\mathbf{t}^* = \arg\min_{\mathbf{t}} \mathbf{V_i}(\mathbf{x}, \mathbf{t})$.**
- **ICD $\Rightarrow$ inexactly minimize (2). i.e., for $\delta \ge 0$ the update is $\mathbf{t}_{\delta^{(\mathbf{i})}} = \arg\min_{\mathbf{t}} \mathbf{V_i}(\mathbf{x}, \mathbf{t}) + \delta^{(\mathbf{i})}$.**

The error δ can't be too big! We restrict $\delta^{(i)} \le \alpha(F(x) - F^*) + \beta$ for $\alpha, \beta \ge 0$.

Algorithm: Inexact Coordinate Descent

```
1. Choose an initial point $x_0 \in \mathbf{R}^N$
2. for $k = 0, 1, 2, \dots$ do
3.     Choose block $i \in \{1, 2, \dots, n\}$ with probability $p_i > 0$
4.     Choose error level $\delta_k^{(i)}$ for block $i$.
5.     Compute the inexact update $t_{\delta_k^{(i)}}$ to block $i$ of $x_k$
6.     Update block $i$ of $x_k$: $x_{k+1} = x_k + U_i t_{\delta_k^{(i)}}$
7. endfor
```

4. ITERATION COMPLEXITY

We hope to guarantee that our algorithm will converge. For ICD, if we take at least K iterations where:

$$K \ge \frac{c}{\epsilon - \alpha c}\log\left(\frac{\epsilon - \frac{\beta c}{\epsilon - \alpha c}}{\epsilon\rho - \frac{\beta c}{\epsilon - \alpha c}}\right) + \frac{c}{\epsilon} + 2,$$

then $\mathbf{P}(F(x_K) - F^* \le \epsilon) \ge 1 - \rho$. i.e., we are within ϵ of the minimum F^* with probability exceeding $1 - \rho$, where: ϵ is the accuracy, ρ is the confidence, $\alpha, \beta \ge 0$ are measures of the inaccuracy in the update and c is a constant depending on the number of blocks n and a (weighted) measure of the initial residual.

- These results generalise those for exact CD. If $\alpha = \beta = 0$ then the exact complexity results are recovered!
- This result can be simplified in the strongly convex and smooth cases.

5. PRECONDITIONING

For $f = \frac{1}{2}\|Ax - b\|_2^2$ and $\Psi = 0$, (2) simplifies, so at every iteration of ICD, the update t is found by solving

$$A_i^T A_i t = -A_i^T (Ax - b), \tag{3}$$

where $A_i = U_i^T A$. We assume that $A_i^T A_i \succ 0$.

- **Exact CD $\Rightarrow$ Solve (3) using direct methods (matrix factorizations)**
- **ICD $\Rightarrow$ Solve (3) using iterative methods (Conjugate Gradients CG) FASTER!**

Question: How can we solve (3) even faster? Use preconditioning!
Rather than finding the update via (3), for $\mathcal{P}_i \approx A_i^T A_i$ we solve

$$\mathcal{P}_i^{-1} A_i^T A_i t = \mathcal{P}_i^{-1} A_i^T (Ax - b). \tag{4}$$

The preconditioned matrix $\mathcal{P}_i^{-1} A_i^T A_i$ should have better spectral properties than $A_i^T A_i$, (i.e., eigenvalues clustered around one) which significantly speeds up the convergence of CG. (See Figure 2.)

6. NUMERICAL EXPERIMENTS

In these numerical experiments we let $f = \frac{1}{2}\|Ax - b\|_2^2$, $\Psi = 0$ and assume that A has block angular structure:

$$A = \begin{bmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_n \\ D_1 & D_2 & \cdots & D_n \end{bmatrix}, \qquad A_i = \begin{bmatrix} C_i \\ \\ D_i \end{bmatrix}$$

- From (3): $A_i^T A_i = C_i^T C_i + D_i^T D_i$.
- Choose the preconditioner

$$\mathcal{P}_i = \begin{cases} C_i^T C_i & \text{if } C_i \text{ is tall} \\ C_i^T C_i + \rho I & \text{if } C_i \text{ is wide} \end{cases} \tag{5}$$

Experiment 1: In the first experiment we study the effect of preconditioning on the clustering of eigenvalues. Figure 2 shows the distribution of eigenvalues of the original matrix $A_i^T A_i$ and the preconditioned matrix $\mathcal{P}_i^{-1} A_i^T A_i$ where $\mathcal{P}_i$ is defined in (5), for both C_i tall and C_i wide.

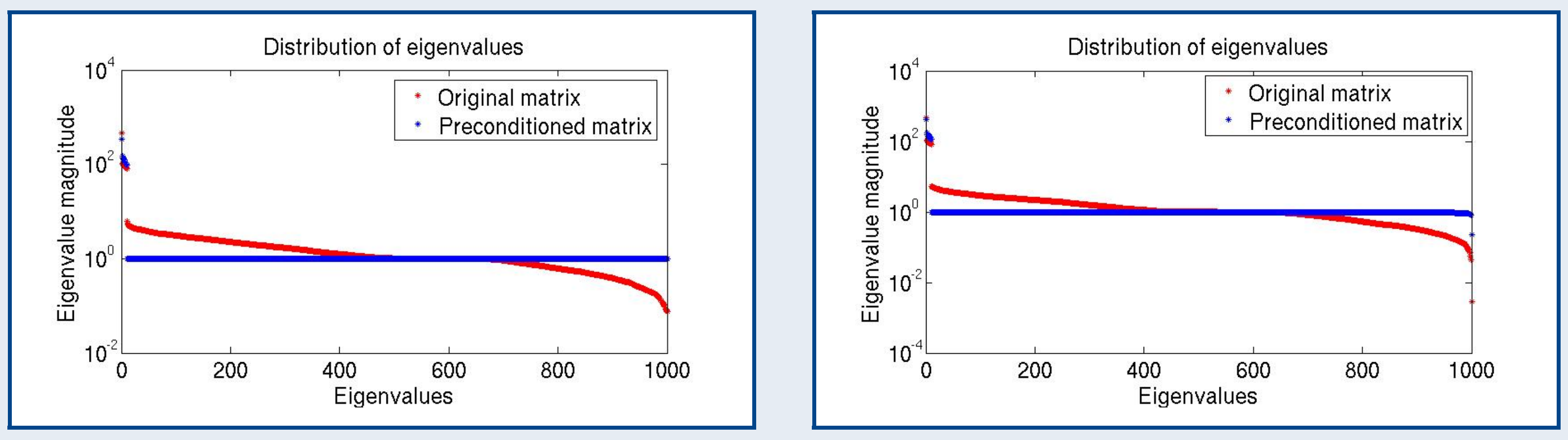


Figure 2: The distribution of eigenvalues before and after preconditioning where A has block angular structure (for a random block A_i). In the left plot C_i is tall and in the right plot C_i is wide ($\rho = 10^{-2}$). In both cases the clustering of eigenvalues is greatly improved after the application of the preconditioner. Most of the eigenvalues are 1 (C_i tall) or $\sim$1 (C_i wide), and the extremal eigenvalues are pulled towards 1.

Experiment 2: In the second experiment we compare the algorithm runtime using either an exact or an inexact update. For ICD we use CG to compute the update and present results with and without preconditioning (solving (4) or (3) respectively). There are $n = 10$ blocks and noise is added to the vector b.

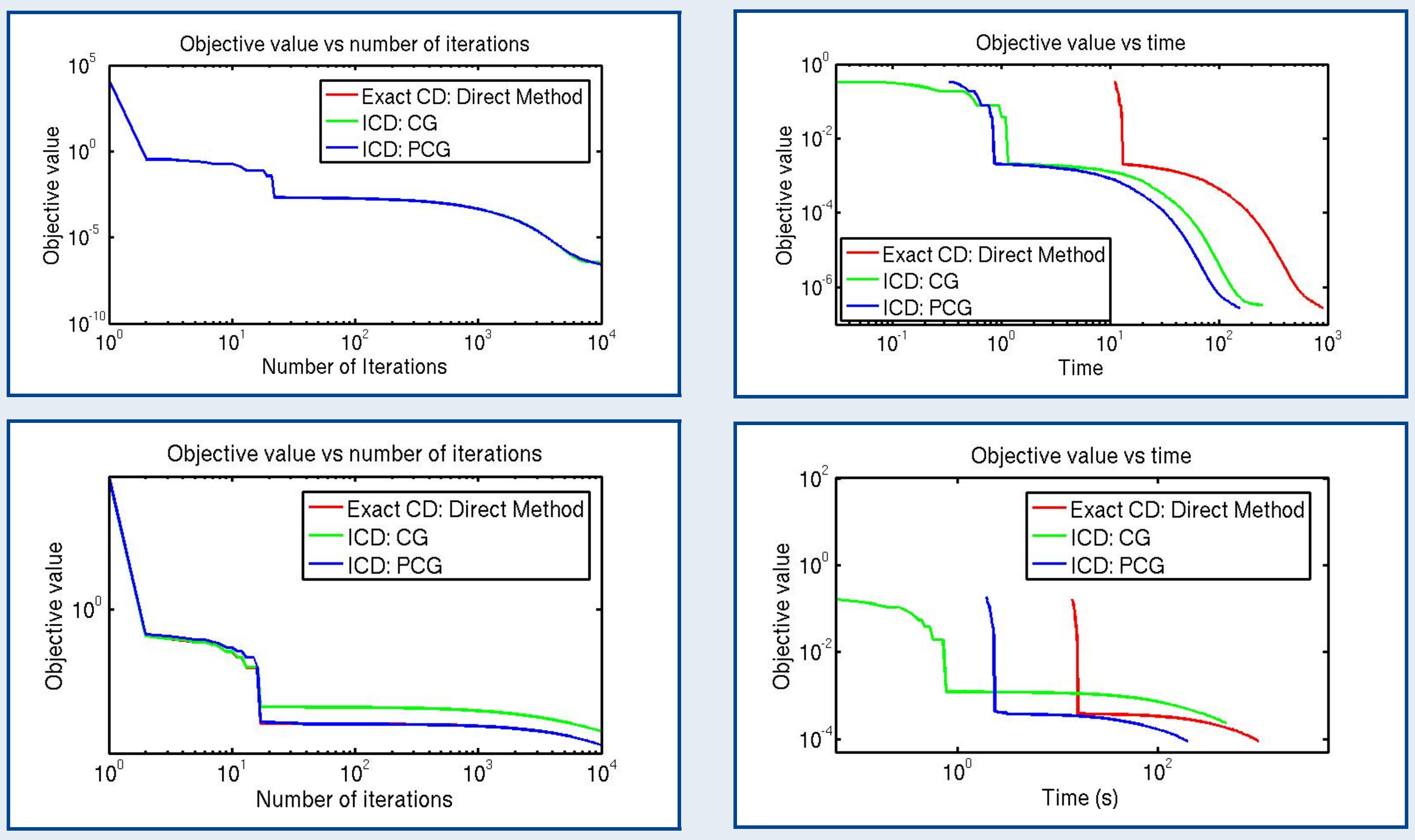


Figure 3: Investigating the effects of inexact updates. In the top row the size of C_i is 1250×1000 and in the bottom row the size of C_i is 990×1000. In both cases D_i is 10×1000. The same number of iterations are required for both an exact and inexact update. However, an inexact update leads to a decrease in the overall algorithm running time, and preconditioning helps reduce this even further.

7. REFERENCES

1. P. Richtárik and M. Takac, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function", *Mathematical Programming Series A*, (2012).
2. R.Tappenden, P. Richtárik and J. Gondzio, "Inexact coordinate descent", *arXiv*, (2013). (See the QR code ⇒)

{j.gondzio,peter.richtarik,r.tappenden}@ed.ac.uk