# Mini-tutorial on *TAO-ADMM*

Hansol Suh

PETSc User Meeting 2023, Chicago, IL

# Alternating Direction Method of Multipliers

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{Ax} + \boldsymbol{Bz} = \boldsymbol{c} \end{aligned}$$

- $f$, $g$ convex
- $\mathcal{L}_\rho = f(\boldsymbol{x}) + g(\boldsymbol{z}) + y^T(\boldsymbol{Ax} + \boldsymbol{Bz} - \boldsymbol{c}) + (\rho/2)\|\boldsymbol{Ax} + \boldsymbol{Bz} - \boldsymbol{c}\|_2^2$
- ADMM:

$$x_{k+1} := \text{argmin}_x \mathcal{L}_\rho(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{y}_k)$$

$$z_{k+1} := \text{argmin}_z \mathcal{L}_\rho(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{y}_k)$$

$$y_{k+1} := y_k + \rho(\boldsymbol{Ax}_{k+1} + \boldsymbol{Bz}_{k+1} - \boldsymbol{c})$$

# Tomography Example

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$$

$$\Downarrow$$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 \\ \text{subject to} \quad & \boldsymbol{x} - \boldsymbol{z} = 0 \end{aligned}$$

Figure 1: Shepp-Logan phantom

# Tomography Example

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$$

$$\Downarrow$$

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 \\ \text{subject to} \quad & \boldsymbol{x} - \boldsymbol{z} = 0 \end{aligned}$$

Regularizer is not smooth, challenging for optimization.

Figure 1: Shepp-Logan phantom

# Tomography Example Notes:

- How to run:

```
make test search='tao_constrained_tutorials-tomographyADMM*'
```

- Other approaches:
  - Could use 2-norm-squared for regularizer
    - Poor performance! Relative error of 80%
  - Could smooth out the $\ell - 1$ term
    - `make test search='tau_leastsquares_tutorials-tomography*'`
  - Technically, using $\|Dx\|_1$, where $D$ is dictionary matrix (TV), gives better performance...

# Example in PETSc

## example.c

```
/* TAO init routines */
TaoCreate(comm, &tao);
TaoSetType(tao, TAOADMM);
TaoSetObjectiveAndGradientRoutine(tao, FullGradObj, AppCtx);
/* Continues... */
```

# Example in PETSc

Here, we are setting $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{c}$, and routines for $f(\boldsymbol{x})$, $g(\boldsymbol{z})$.

example.c

```
/* Misfit/Regularizer {M,R} part.  */
TaoADMMSet{M,R}ObjAndGradRoutine(tao, {f,g}_objgrad, AppCtx);
TaoADMMSet{M,R}HessRoutine(tao,H,H_pre,func_H_{f,g},AppCtx);
/* You can pass NULL, which means I for Misfit, and -I for Reg */
TaoADMMSet{M,R}ConstJac(tao,{A,B},{A,B}_pre,func_{A,B},AppCtx);
TaoADMMSetConstraintVectorRHS(tao, c);
```

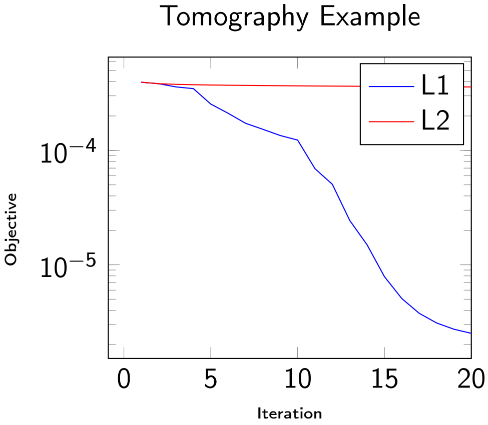
# Example in PETSc

You can specify solver types for $f(\boldsymbol{x})$ and $g(\boldsymbol{z})$.
You can also write your own custom solver via `TaoShell`.

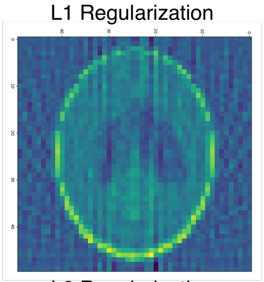
example.c

```
/* Set type for misfit  */
TaoADMMGetMisfitSubsolver(tao, &misfit);
TaoSetType(misfit, TAONLS);
/* Set type for L1 regularizer. */
TaoADMMGetRegSubsolver(tao, &reg);
TaoSetType(reg, TAOSHELL);
TaoShellSetContext(reg, shell_ctx);
TaoShellSetSolve(reg, SoftThreshold);
TaoADMMSetRegCoefficient(tao, lambda);
```

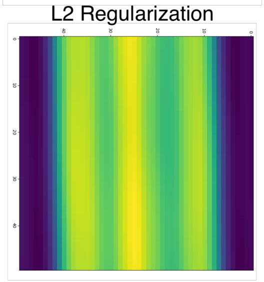
# Results



Tomography Example
L1
L2
Objective
$10^{-4}$
$10^{-5}$
0
5
10
15
20
Iteration

L1 Regularization

L2 Regularization

# Options

## options

```
/* Many options available!!  */
-tao_admm_regularizer_coefficient ...
-tao_admm_spectral_penalty ...
-tao_admm_relaxation_parameter ...
-tao_admm_tolerance_update_factor ...
-tao_admm_spectral_penalty_update_factor ...
-tao_admm_minimum_spectral_penalty ...
-tao_admm_dual_update {adaptive,adaptive_relaxed}
-tao_admm_regularizer_type {user,soft_threshold}
-misfit_tao_nls_pc_type ...
```

# End

Thanks! Any feedback would be greatly appreciated!

hsuh@anl.gov