Science & Technology Facilities Council

Iterative methods within optimization problems

Tyrone Rees

STFC Rutherford Appleton Laboratory

The problem...

Consider the quadratic program:

$$\begin{aligned} \min \ & \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{f}^T\mathbf{x} \\ \text{s.t. } & B\mathbf{x} = \mathbf{g}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$H \in \mathbb{R}^{n\times n}$ s.p.d.
$B \in \mathbb{R}^{m\times n}$ full rank
$m < n$

Science & Technology Facilities Council

The problem...

Consider the quadratic program:

$$\min \ \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{f}^T\mathbf{x} \quad \text{s.t. } B\mathbf{x} = \mathbf{g}, \quad \mathbf{x} \geq \mathbf{0}$$

Primal

$$\max \ \mathbf{g}^T\mathbf{y} - \frac{1}{2}\mathbf{x}^T H\mathbf{x} \quad \text{s.t. } B^T\mathbf{y} + \mathbf{s} - H\mathbf{x} = \mathbf{f}, \quad \mathbf{s} \geq \mathbf{0}$$

Dual

Science & Technology Facilities Council

The problem...

Consider the quadratic program:

$$\min \ \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{f}^T\mathbf{x} \quad \text{s.t. } B\mathbf{x} = \mathbf{g}, \quad \mathbf{x} \geq \mathbf{0}$$

$$\max \ \mathbf{g}^T\mathbf{y} - \frac{1}{2}\mathbf{x}^T H\mathbf{x} \quad \text{s.t. } B^T\mathbf{y} + \mathbf{s} - H\mathbf{x} = \mathbf{f} \quad \mathbf{s} \geq \mathbf{0}$$

Primal

Dual

$$\mathbf{s} - H\mathbf{x} + B^T\mathbf{y} = \mathbf{f}$$
$$B\mathbf{x} = \mathbf{g}$$
$$XS\mathbf{e} = 0$$
$$\mathbf{x}_i \geq 0, \quad \mathbf{s}_i \geq 0$$

$X = \text{diag}(\mathbf{x})$

$S = \text{diag}(\mathbf{s})$

Science & Technology Facilities Council

Newton step

Need to solve a system of the form

$$\begin{bmatrix} -H & B^T & I \\ B & 0 & 0 \\ S_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{y}_k \\ \Delta \mathbf{s}_k \end{bmatrix} = \begin{bmatrix} \mathbf{r}_c \\ \mathbf{r}_b \\ -X_k S_k \mathbf{e} + \sigma_k \mu_k \mathbf{e} \end{bmatrix}$$

and take a partial step in the Newton direction

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{k+1} \\ \mathbf{s}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \\ \mathbf{s}_k \end{bmatrix} + \alpha_k \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{y}_k \\ \Delta \mathbf{s}_k \end{bmatrix}$$

Newton step

Need to solve a system of the form

$$\begin{bmatrix} H + S_k^{-1}X_k & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k \\ -\Delta \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} X_k^{-1}\mathbf{r}_\tau - \mathbf{r}_c \\ \mathbf{r}_b \end{bmatrix}$$

and take a partial step in the Newton direction

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{k+1} \\ \mathbf{s}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \\ \mathbf{s}_k \end{bmatrix} + \alpha_k \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{y}_k \\ \Delta \mathbf{s}_k \end{bmatrix}$$

Science & Technology Facilities Council

Newton step

ill-conditioned close to convergence

Need to solve a system of the form

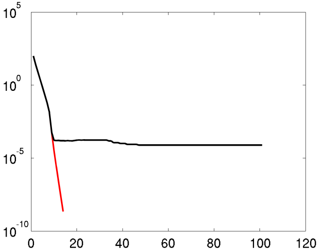
$$\begin{bmatrix} H + S_k^{-1}X_k & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k \\ -\Delta \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} X_k^{-1}\mathbf{r}_\tau - \mathbf{r}_c \\ \mathbf{r}_b \end{bmatrix}$$

and take a partial step in the Newton direction

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{k+1} \\ \mathbf{s}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \\ \mathbf{s}_k \end{bmatrix} + \alpha_k \begin{bmatrix} \Delta \mathbf{x}_k \\ \Delta \mathbf{y}_k \\ \Delta \mathbf{s}_k \end{bmatrix}$$

The problem:

We'd like to be able to solve the augmented system using an iterative method



— exact (backslash)
— exact $+ 10^{-10}$ `randn`

$$\|b - Bx_k\|_2$$

GOULDQP2 from CUTEst

Constraint preconditioners

If the approximate solution satisfies the constraints exactly then interior point methods can be seen to converge

[Al-Jeiroudi, Gondzio, 2009]

Science & Technology Facilities Council

Constraint preconditioners

If the approximate solution satisfies the constraints exactly then interior point methods can be seen to converge

[Al-Jeiroudi, Gondzio, 2009]

Use a Krylov method with a constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}$$

then the constraints will be satisfied exactly* at every iteration

[Rozlozník, Simoncini, 2002] [de Sturler, Liesen, 2005]

Constraint preconditioners

If the approximate solution satisfies the constraints exactly then interior point methods can be seen to converge
[Al-Jeiroudi, Gondzio, 2009]

Use a Krylov method with a constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}$$

then the constraints will be satisfied exactly* at every iteration
[Rozložník, Simoncini, 2002] [de Sturler, Liesen, 2005]

* in exact arithmetic

Science & Technology Facilities Council

Issues with constraint preconditioners

$$\mathcal{A} = \begin{bmatrix} H + S^{-1}X & B^T \\ B & 0 \end{bmatrix} \qquad \mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}$$

- need to solve with B accurately
- need to use a ‘cheap’ G, but close to $H + S^{-1}X$

Issues with constraint preconditioners

$$\mathcal{A} = \begin{bmatrix} H + S^{-1}X & B^T \\ B & 0 \end{bmatrix} \qquad \mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}$$

We have studied the properties of the projected CG method for solving quadratic programming problems of the form (1.1)–(1.2). Due to the form of the preconditioners used by some nonlinear programming algorithms we opted for not computing a basis Z for the null space of the constraints, but instead projecting the CG iterates using a normal equations or augmented system approach. We have given examples showing that in either case significant roundoff errors can occur, and have presented an explanation for this.

We proposed several remedies. One is to use iterative refinement of the augmented system or normal equations approaches. An alternative is to update the residual at every iteration of the CG iteration, as described in §6. The latter can be implemented particularly efficiently when the preconditioner is given by $G = I$ in (2.5).

[Gould, Hribar, Nocedal, 2001]

Block- diagonal preconditioners

A popular preconditioning paradigm in other contexts has been

$$\mathcal{P} = \begin{bmatrix} G & 0 \\ 0 & W \end{bmatrix}$$

$$G \approx H + X^{-1}S$$
$$W \approx B(H + X^{-1}S)^{-1}B^T$$

[Murphy, Golub, Wathen, 2000]

MINRES: residual converges in the $\mathcal{P}^{-1}$-norm

- We will, in general, not have the constraints satisfied exactly

Science & Technology Facilities Council

Projections

Every (inner) iteration need not lie on the constraints, only the final approximation.

Idea: After getting an approximate solution $\begin{bmatrix} \widehat{\Delta \mathbf{x}_k} \\ \widehat{\Delta \mathbf{y}_k} \end{bmatrix}$ (by any means), project onto the constraints:

$$\begin{bmatrix} \Delta \bar{\mathbf{x}}_k \\ \Delta \bar{\mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} \widehat{\Delta \mathbf{x}_k} \\ \widehat{\Delta \mathbf{y}_k} \end{bmatrix} + \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} - \begin{bmatrix} H + S_k^{-1} X_k & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta \mathbf{x}_k} \\ \widehat{\Delta \mathbf{y}_k} \end{bmatrix} \right)$$

Projections

Every (inner) iteration need not lie on the constraints, only the final approximation.

Idea: After getting an approximate solution $\begin{bmatrix} \widehat{\Delta \mathbf{x}_k} \\ \widehat{\Delta \mathbf{y}_k} \end{bmatrix}$ (by any means), project onto the constraints:

$$\begin{bmatrix} \bar{\Delta \mathbf{x}}_k \\ \bar{\Delta \mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} \widehat{\Delta \mathbf{x}_k} \\ \widehat{\Delta \mathbf{y}_k} \end{bmatrix} + \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \phi \\ \mathbf{g} - B\widehat{\Delta \mathbf{x}_k} \end{bmatrix}$$

Science & Technology Facilities Council

Pick G and ϕ

$\phi = 0$:

$$\left\| \begin{bmatrix} \Delta\mathbf{x}_k - \Delta\bar{\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \Delta\bar{\mathbf{y}}_k \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}_k} \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}_k} \end{bmatrix} - \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{g} - B\widehat{\Delta\mathbf{x}_k} \end{bmatrix} \right\|$$

Pick G and ϕ

$\phi = 0$:

$$\left\| \begin{bmatrix} \Delta\mathbf{x}_k - \Delta\bar{\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \Delta\bar{\mathbf{y}}_k \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}_k} \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}_k} \end{bmatrix} - \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{g} - B\widehat{\Delta\mathbf{x}_k} \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}_k} \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}_k} \end{bmatrix} - \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ B & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}_k} \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}_k} \end{bmatrix} \right\|$$

Pick G and ϕ

$\phi = 0$:

$$\left\| \begin{bmatrix} \Delta\mathbf{x}_k - \Delta\bar{\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \Delta\bar{\mathbf{y}}_k \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}}_k \end{bmatrix} - \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{g} - B\widehat{\Delta\mathbf{x}}_k \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ B & 0 \end{bmatrix} \right) \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}}_k \end{bmatrix} \right\|$$

Pick G and ϕ

$\phi = 0$:

$$\left\| \begin{bmatrix} \Delta \mathbf{x}_k - \Delta \bar{\mathbf{x}}_k \\ \Delta \mathbf{y}_k - \Delta \bar{\mathbf{y}}_k \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Delta \mathbf{x}_k - \widehat{\Delta \mathbf{x}_k} \\ \Delta \mathbf{y}_k - \widehat{\Delta \mathbf{y}_k} \end{bmatrix} - \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{g} - B\widehat{\Delta \mathbf{x}_k} \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} G & B^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k - \widehat{\Delta \mathbf{x}_k} \\ \Delta \mathbf{y}_k - \widehat{\Delta \mathbf{y}_k} \end{bmatrix} \right\|$$

$$\leq \left\| \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} G & B^T \\ 0 & 0 \end{bmatrix} \right\| \left\| \begin{bmatrix} \Delta \mathbf{x}_k - \widehat{\Delta \mathbf{x}_k} \\ \Delta \mathbf{y}_k - \widehat{\Delta \mathbf{y}_k} \end{bmatrix} \right\|$$

$G = \beta I$

Singular values of $\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \beta I & B^T \\ 0 & 0 \end{bmatrix}$ are the square roots of the eigenvalues of

$$\begin{bmatrix} I - (1-\beta^2)B^T(BB^T)^{-1}B & \beta B^T(BB^T)^{-1} \\ \beta(BB^T)^{-1}B & I \end{bmatrix}.$$

Tend towards $0, 1$ as $\beta \to 0$.

Science & Technology Facilities Council

$G = \beta I$

Singular values of $\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \beta I & B^T \\ 0 & 0 \end{bmatrix}$ are the square roots of the eigenvalues of

$$\begin{bmatrix} I - (1-\beta^2)B^T(BB^T)^{-1}B & \beta B^T(BB^T)^{-1} \\ \beta(BB^T)^{-1}B & I \end{bmatrix}.$$

Tend towards $0, 1$ as $\beta \to 0$.

- $\beta = 0$: projecting the error onto the constraint

$$\begin{bmatrix} \Delta\mathbf{x}_k - \Delta\bar{\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \Delta\bar{\mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} I - B^T(BB^T)^{-1}B & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_k - \widehat{\Delta\mathbf{x}}_k \\ \Delta\mathbf{y}_k - \widehat{\Delta\mathbf{y}}_k \end{bmatrix}$$

$G = \beta I$

Singular values of $\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \beta I & B^T \\ 0 & 0 \end{bmatrix}$ are the square roots of the eigenvalues of

$$\begin{bmatrix} I - (1-\beta^2)B^T(BB^T)^{-1}B & \beta B^T(BB^T)^{-1} \\ \beta(BB^T)^{-1}B & I \end{bmatrix}.$$

Tend towards $0, 1$ as $\beta \to 0$.

- $\beta = 0$: projecting the error onto the constraint
- $B\widehat{\Delta \mathbf{x}_k}$ should be *small*: better behaved than worst case

$$\begin{bmatrix} \Delta \mathbf{x}_k - \Delta \bar{\mathbf{x}}_k \\ \Delta \mathbf{y}_k - \Delta \bar{\mathbf{y}}_k \end{bmatrix} = \begin{bmatrix} I - (1-\beta^2)B^T(BB^T)^{-1}B & \beta B^T(BB^T)^{-1} \\ \beta(BB^T)^{-1}B & I \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_k - \widehat{\Delta \mathbf{x}_k} \\ \Delta \mathbf{y}_k - \widehat{\Delta \mathbf{y}_k} \end{bmatrix}$$

Science & Technology Facilities Council

$G = \beta I$

Singular values of $\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \beta I & B^T \\ 0 & 0 \end{bmatrix}$ are the square roots of the eigenvalues of

$$\begin{bmatrix} I - (1-\beta^2)B^T(BB^T)^{-1}B & \beta B^T(BB^T)^{-1} \\ \beta(BB^T)^{-1}B & I \end{bmatrix}.$$

Tend towards $0, 1$ as $\beta \to 0$.

- $\beta = 0$: projecting the error onto the constraint
- $B\widehat{\Delta \mathbf{x}_k}$ should be *small*: better behaved than worst case
- Independent of the (1,1) block

Science & Technology Facilities Council

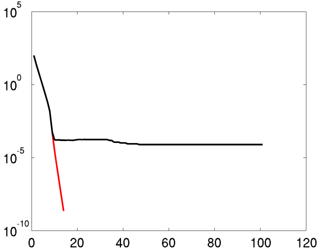
Applying post-processor

Need to solve with $\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix}$ once per outer iteration.

- Factorize once?
- Use a weighted least squares solver? [MA75]
- Use

$$\begin{bmatrix} \beta I & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ (1/\beta)B & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -BB^T \end{bmatrix} \begin{bmatrix} \beta I & B^T \\ 0 & (1/\beta)I \end{bmatrix}?$$

Numerical results

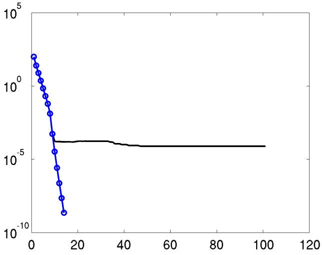


$$\|b - Bx_k\|_2$$

— exact (backslash)
— exact + 10^{-10}`randn`

GOULDQP2 from CUTEst

Numerical results

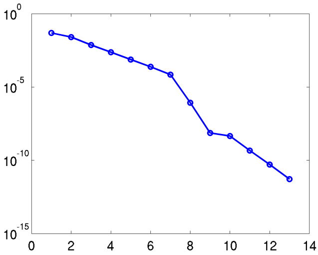


$\|b - Bx_k\|_2$

— exact (backslash)
— exact $+ 10^{-10}$randn
— exact $+ 10^{-10}$randn $+$ post-processing ($\beta = 10^{-3}$)

GOULDQP2 from CUTEst

Numerical results

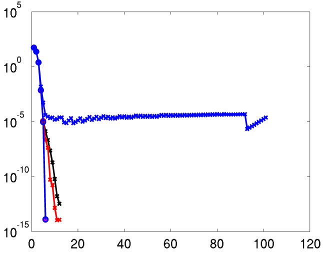


— exact (backslash)
— exact + 10^{-10}randn
— exact + 10^{-10}randn + post-processing ($\beta = 10^{-3}$)

$$\mu_k = \frac{\mathbf{x}^T \mathbf{s}}{n}$$

GOULDQP2 from CUTEst

Numerical results



$\|b - Bx_k\|_2$

— MINRES, Augmented Lagrangian prec.
— MINRES, Block Diagonal prec.
— BiCG-Stab, Block Lower triangular prec.
$\times \to$ no post-proc.
$\circ \to$ with post-proc.

AUG3DCQP from CUTEst

Science & Technology Facilities Council

Conclusions:

- If we employ a simple post-processing projection, any iterative method can be used to safely approximately solve the augmented system in primal dual interior point methods

Outlook:

- How to pick β?
- Is there a better choice of G and/or ϕ?
- Freed of the need to satisfy the constraints, what's a good preconditioner for matrices with this structure?