QPBLUR: An active-set convex QP solver

Christopher Maes and Michael Saunders
iCME, Stanford University

RTRA STAE Workshop

Advanced methods and perspectives

in nonlinear optimisation and control

Toulouse, France

QPBLUR: An active-set convex QP solver based on regularized KKT systems

SNOPT obtains search directions from convex QP subproblems, currently solved by SQOPT. For problems with many degrees of freedom, the nullspace active-set method of SQOPT becomes inefficient.

QPBLUR: An active-set convex QP solver based on regularized KKT systems

SNOPT obtains search directions from convex QP subproblems, currently solved by SQOPT. For problems with many degrees of freedom, the nullspace active-set method of SQOPT becomes inefficient.

QPBLUR is an alternative QP solver intended for use within SNOPT. Primal and dual regularization ensures that the KKT system for any active set is nonsingular.

QPBLUR: An active-set convex QP solver based on regularized KKT systems

SNOPT obtains search directions from convex QP subproblems, currently solved by SQOPT. For problems with many degrees of freedom, the nullspace active-set method of SQOPT becomes inefficient.

QPBLUR is an alternative QP solver intended for use within SNOPT. Primal and dual regularization ensures that the KKT system for any active set is nonsingular.

A single-phase active-set method is possible. Warm starts can proceed from any active set. Block-LU updates of the KKT factors as in QPBLU (Hanh Huynh's PhD thesis 2008) allow use of packages such as LUSOL, MA57, PARDISO, SuperLU, or UMFPACK.

QPBLUR: An active-set convex QP solver based on regularized KKT systems

SNOPT obtains search directions from convex QP subproblems, currently solved by SQOPT. For problems with many degrees of freedom, the nullspace active-set method of SQOPT becomes inefficient.

QPBLUR is an alternative QP solver intended for use within SNOPT. Primal and dual regularization ensures that the KKT system for any active set is nonsingular.

A single-phase active-set method is possible. Warm starts can proceed from any active set. Block-LU updates of the KKT factors as in QPBLU (Hanh Huynh's PhD thesis 2008) allow use of packages such as LUSOL, MA57, PARDISO, SuperLU, or UMFPACK.

QPBLUR is effectively a penalty method with bounds. QPBCL (bound-constrained Lagrangian) includes a Lagrangian term to satisfy $Ax = b$ more accurately.

Supported by the Office of Naval Research and AHPCRC

Motivation

Why another QP solver?

We would like a sparse QP solver that

- can handle a large number of free variables
- can warm-start efficiently ($\Rightarrow$ active-set method)

Why another QP solver?

We would like a sparse QP solver that

- can handle a large number of free variables
- can warm-start efficiently ($\Rightarrow$ active-set method)

Such a solver would be useful

- inside SQP methods (like SNOPT)
- for related QPs (e.g. model predictive control)

SNOPT: Large-scale NLP

- Nonlinear objective & constraints, sparse Jacobian

SNOPT: Large-scale NLP

- Nonlinear objective & constraints, sparse Jacobian
- SQP method solves a sequence of QP subproblems:

$$\min_x \quad c^T x + \tfrac{1}{2} x^T H x$$
$$Ax = b, \qquad l \le x \le u,$$

where c, H, A, b change less and less

SNOPT: Large-scale NLP

- Nonlinear objective & constraints, sparse Jacobian
- SQP method solves a sequence of QP subproblems:

$$\min_x \quad c^T x + \tfrac{1}{2} x^T H x$$
$$Ax = b, \qquad l \le x \le u,$$

where c, H, A, b change less and less

- $H = G^T G$ is a limited-memory BFGS approximation

$$G = D(I + s_1 v_1^T)(I + s_2 v_2^T) \dots$$

SNOPT: Large-scale NLP

- Nonlinear objective & constraints, sparse Jacobian
- SQP method solves a sequence of QP subproblems:

$$\min_x \quad c^T x + \tfrac{1}{2} x^T H x$$
$$Ax = b, \qquad l \le x \le u,$$

where c, H, A, b change less and less

- $H = G^T G$ is a limited-memory BFGS approximation

$$G = D(I + s_1 v_1^T)(I + s_2 v_2^T) \dots$$

- Currently use SQOPT (active-set null-space method)

SQOPT: Large-scale convex QP

SQOPT

$$\min\ c^T x + \tfrac{1}{2} x^T H x \quad \text{st} \quad Ax = b, \quad l \le x \le u$$

Reduced-Hessian method

SQOPT

$$\min \; c^T x + \tfrac{1}{2} x^T H x \quad \text{st} \quad Ax = b, \quad l \le x \le u$$

Reduced-Hessian method

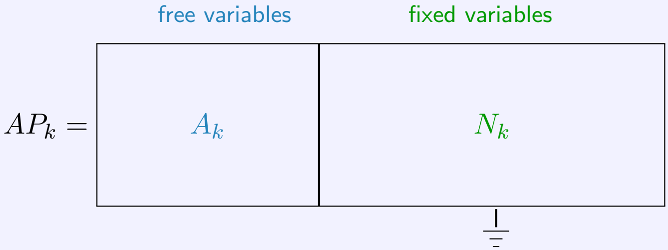
$$Z^T H Z \Delta x_S = -Z^T g$$

- $Z^T H Z = R^T R$ exactly
- $Z^T H Z \approx R^T R$ quasi-Newton
- CG if $dof > 2000$
- LUSOL provides reliable Basis Repair

SQOPT

$$\min \; c^T x + \tfrac{1}{2} x^T H x \quad \text{st} \quad Ax = b, \quad l \le x \le u$$

Reduced-Hessian method



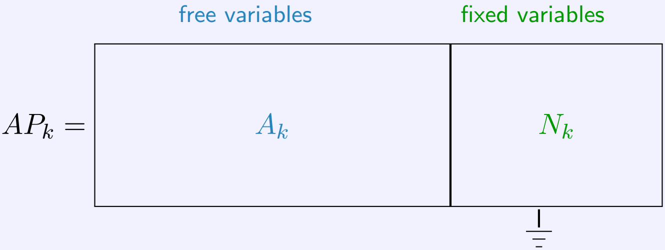
OK if A_k is nearly square:

$$10000 \times 12000 \quad \text{or} \quad 100000 \times 102000$$

SQOPT

$$\min \; c^T x + \tfrac{1}{2} x^T H x \quad \text{st} \quad Ax = b, \quad l \le x \le u$$

Reduced-Hessian method



For A_k 100000×400000, we need a
QP solver based on KKT systems like QPA in GALAHAD

KKT system for current active set

$$\text{Solve } K_0 \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

$$K_0 = \begin{bmatrix} H_k & A_k^T \\ A_k & \end{bmatrix}$$

QPBLU

QPBLU (Thesis of Hanh Huynh 2008)

F90 convex QP solver based on block-LU updates of K

QPBLU (Thesis of Hanh Huynh 2008)

F90 convex QP solver based on block-LU updates of K

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 D_0 L_0^T \quad \text{or} \quad L_0 U_0$$

QPBLU (Thesis of Hanh Huynh 2008)

F90 convex QP solver based on block-LU updates of K

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 D_0 L_0^T \quad \text{or} \quad L_0 U_0$$

- Uses black-box factorizer on K_0
  (LUSOL, MA57, PARDISO, SuperLU, UMFPACK)

QPBLU (Thesis of Hanh Huynh 2008)

F90 convex QP solver based on block-LU updates of K

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 D_0 L_0^T \quad \text{or} \quad L_0 U_0$$

- Uses black-box factorizer on K_0
  (LUSOL, MA57, PARDISO, SuperLU, UMFPACK)
- Active-set method keeps K_0 nonsingular in theory

QPBLU block-LU updates

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 U_0$$

QPBLU block-LU updates

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 U_0$$

To change active set (add/delete cols of A_k),
work with bordered system:

$$\begin{bmatrix} K_0 & V \\ V^T & E \end{bmatrix} = \begin{bmatrix} L_0 & \\ Z^T & I \end{bmatrix} \begin{bmatrix} U_0 & Y \\ & C \end{bmatrix}$$

QPBLU block-LU updates

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 U_0$$

To change active set (add/delete cols of A_k),
work with bordered system:

$$\left[\begin{array}{c|c} K_0 & V \\ \hline V^T & E \end{array}\right] = \begin{bmatrix} L_0 & \\ Z^T & I \end{bmatrix} \begin{bmatrix} U_0 & Y \\ & C \end{bmatrix}$$

- Y, Z sparse, C small

QPBLU block-LU updates

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 U_0$$

To change active set (add/delete cols of A_k),
work with bordered system:

$$\begin{pmatrix} K_0 & V \\ V^T & E \end{pmatrix} = \begin{bmatrix} L_0 & \\ Z^T & I \end{bmatrix} \begin{bmatrix} U_0 & Y \\ & C \end{bmatrix}$$

- Y, Z sparse, C small
- Quasi-Newton updates to H_k handled same way

QPBLU block-LU updates

$$K_0 = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} = L_0 U_0$$

To change active set (add/delete cols of A_k),
work with bordered system:

$$\begin{bmatrix} K_0 & V \\ V^T & E \end{bmatrix} = \begin{bmatrix} L_0 & \\ Z^T & I \end{bmatrix} \begin{bmatrix} U_0 & Y \\ & C \end{bmatrix}$$

- Y, Z sparse, C small
- Quasi-Newton updates to H_k handled same way
- Singular K_0 is a difficulty – need KKT repair

QPBLUR: Large-scale QP with Regularization

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^T x + \tfrac{1}{2}x^T H x + \tfrac{1}{2}\delta\|x\|_2^2 + \tfrac{1}{2}\mu\|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^Tx + \tfrac{1}{2}x^THx + \tfrac{1}{2}\delta\|x\|_2^2 + \tfrac{1}{2}\mu\|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \mu I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ 0 \end{pmatrix}$$

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \mu I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ 0 \end{pmatrix}$$

- Nonsingular for any active set (any active cols A_k)

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \mu I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ 0 \end{pmatrix}$$

- Nonsingular for any active set (any active cols A_k)
- Always feasible (no Phase 1)

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \mu I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ 0 \end{pmatrix}$$

- Nonsingular for any active set (any active cols A_k)
- Always feasible (no Phase 1)
- Can use LUSOL, MA57, UMFPACK, ... without change

QPBLUR (Thesis of Chris Maes 2010)

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$
$$Ax + \mu y = b, \qquad l \le x \le u$$

δ and μ small

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \mu I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ 0 \end{pmatrix}$$

- Nonsingular for any active set (any active cols A_k)
- Always feasible (no Phase 1)
- Can use LUSOL, MA57, UMFPACK, ... without change
- Can use Hanh's block-LU updates without change

QPBLUR strategy

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$
$$Ax + \mu y = b, \qquad l \le x \le u$$

MATLAB implementation

- Scale problem

QPBLUR strategy

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

MATLAB implementation

- Scale problem
- Solve with $\delta, \mu = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$

QPBLUR strategy

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$
$$Ax + \mu y = b, \qquad l \le x \le u$$

MATLAB implementation

- Scale problem
- Solve with $\delta, \mu = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$
- Unscale

QPBLUR strategy

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

$$Ax + \mu y = b, \qquad l \le x \le u$$

MATLAB implementation

- Scale problem
- Solve with $\delta, \mu = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$
- Unscale
- Solve with $\delta, \mu = \quad 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$

QPBLUR strategy

$$\min_{x,y} \quad c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2 + \tfrac{1}{2}\mu \|y\|_2^2$$

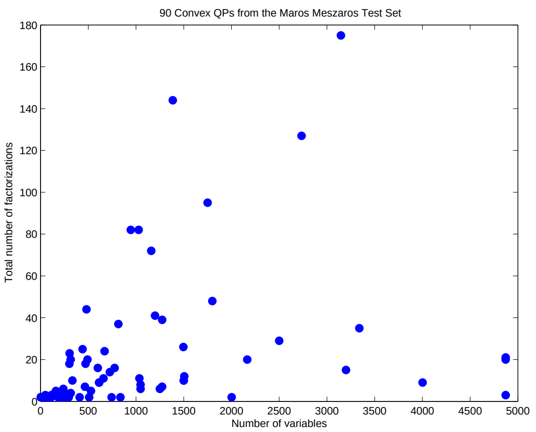
$$Ax + \mu y = b, \qquad l \le x \le u$$

MATLAB implementation

- Scale problem
- Solve with $\delta, \mu = 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$
- Unscale
- Solve with $\delta, \mu = 10^{-8}, 10^{-10}, 10^{-12}$ *opttol* $= \sqrt{\delta}$
- Exit if small relative change in obj

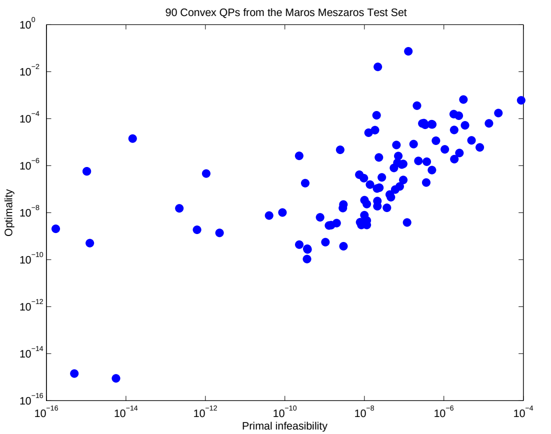
Numerical Results on 90 QP problems (Meszaros set)

KKT factorizations



No. of factorizations on 90 Meszaros QP test problems

Accuracy of solutions



Residuals for 90 Meszaros QP test problems

QPBLUR pros and cons

$$\min_{x,y}\ c^Tx + \tfrac{1}{2}x^THx + \tfrac{1}{2}\delta x^Tx + \tfrac{1}{2}\mu y^Ty \qquad Ax + \mu y = b,\ l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)

QPBLUR pros and cons

$$\min_{x,y}\; c^T x + \tfrac{1}{2}x^T H x + \tfrac{1}{2}\delta x^T x + \tfrac{1}{2}\mu y^T y \qquad Ax + \mu y = b,\; l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)
- Any black-box LU or LDL^T solver
  (preferably separate L/U solves and no refinement)

QPBLUR pros and cons

$$\min_{x,y}\ c^T x + \tfrac{1}{2}x^T H x + \tfrac{1}{2}\delta x^T x + \tfrac{1}{2}\mu y^T y \qquad Ax + \mu y = b,\ l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)
- Any black-box LU or LDL$^{\mathsf{T}}$ solver
  (preferably separate L/U solves and no refinement)
- LU should never report singularity (no KKT repair)

QPBLUR pros and cons

$$\min_{x,y} \; c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta x^T x + \tfrac{1}{2}\mu y^T y \qquad Ax + \mu y = b,\; l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)
- Any black-box LU or LDL$^\mathsf{T}$ solver
  (preferably separate L/U solves and no refinement)
- LU should never report singularity (no KKT repair)
- Simple algorithm (strict convexity, always feasible, no primal degeneracy)

QPBLUR pros and cons

$$\min_{x,y} \; c^Tx + \tfrac{1}{2}x^THx + \tfrac{1}{2}\delta x^Tx + \tfrac{1}{2}\mu y^Ty \qquad Ax + \mu y = b,\; l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)
- Any black-box LU or LDL^T solver
  (preferably separate L/U solves and no refinement)
- LU should never report singularity (no KKT repair)
- Simple algorithm (strict convexity, always feasible, no primal degeneracy)

Disadvantages

- Large regularization: many itns from cold start

QPBLUR pros and cons

$$\min_{x,y}\ c^Tx + \tfrac{1}{2}x^THx + \tfrac{1}{2}\delta x^Tx + \tfrac{1}{2}\mu y^Ty \qquad Ax + \mu y = b,\ l \le x \le u$$

Advantages

- Warm starts (any x_0, any working set)
- Any black-box LU or LDL^T solver (preferably separate L/U solves and no refinement)
- LU should never report singularity (no KKT repair)
- Simple algorithm (strict convexity, always feasible, no primal degeneracy)

Disadvantages

- Large regularization: many itns from cold start
- Tiny regularization: risks ill-conditioned KKT (but so far so good)

Penalty vs Augmented Lagrangian

Rethink QPBLUR

$$\phi(x) \equiv c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2} \delta \|x\|_2^2$$

Rethink QPBLUR

$$\phi(x) \equiv c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2} \delta \|x\|_2^2$$

Regularized QP

$$\min_{x,y} \ \phi(x) + \tfrac{1}{2}\mu\|y\|_2^2 \qquad Ax + \mu y = b, \quad l \le x \le u$$

Rethink QPBLUR

$$\phi(x) \equiv c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2$$

Regularized QP

$$\min_{x,y} \; \phi(x) + \tfrac{1}{2}\mu \|y\|_2^2 \qquad Ax + \mu y = b, \quad l \le x \le u$$

$\equiv$ quadratic penalty function:

$$\min_{\ell \le x \le u} \; \phi(x) + \tfrac{1}{2}\rho \|b - Ax\|_2^2 \qquad (\rho = 1/\mu)$$

Rethink QPBLUR

$$\phi(x) \equiv c^T x + \tfrac{1}{2} x^T H x + \tfrac{1}{2}\delta \|x\|_2^2$$

Regularized QP

$$\min_{x,y} \ \phi(x) + \tfrac{1}{2}\mu \|y\|_2^2 \qquad Ax + \mu y = b, \quad l \le x \le u$$

$\equiv$ quadratic penalty function:

$$\min_{\ell \le x \le u} \ \phi(x) + \tfrac{1}{2}\rho \|b - Ax\|_2^2 \qquad (\rho = 1/\mu)$$

Why not Augmented Lagrangian?:

$$\min_{\ell \le x \le u} \ \phi(x) - \hat{y}^T(b - Ax) + \tfrac{1}{2}\rho \|b - Ax\|_2^2$$

BCL method as in LANCELOT

BCL Method (LANCELOT)

$$\min\ \phi(x)\ \ \text{st}\ \ c(x) = 0,\quad \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

BCL Method (LANCELOT)

$$\min\ \phi(x) \ \text{ st }\ c(x) = 0, \quad \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

- Subproblem: $\min_x\ L(x, \hat{y}, \rho)\ \text{ st }\ \ell \le x \le u$

BCL Method (LANCELOT)

$$\min \ \phi(x) \ \ \text{st} \ \ c(x) = 0, \ \ \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

- Subproblem: $\min_x \ L(x, \hat{y}, \rho) \ \ \text{st} \ \ \ell \le x \le u$
- Solve to get $\hat{x}$ (optimality tol $\omega \to 0$)

BCL Method (LANCELOT)

$$\min \ \phi(x) \ \text{ st } \ c(x) = 0, \quad \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

- Subproblem: $\min_x \ L(x, \hat{y}, \rho)$ st $\ell \le x \le u$
- Solve to get $\hat{x}$ (optimality tol $\omega \to 0$)
- If $\|c(\hat{x})\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)

BCL Method (LANCELOT)

$$\min \ \phi(x) \ \text{ st } \ c(x) = 0, \ \ \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

- Subproblem: $\min_x \ L(x, \hat{y}, \rho)$ st $\ell \le x \le u$
- Solve to get $\hat{x}$ (optimality tol $\omega \to 0$)
- If $\|c(\hat{x})\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)
- Otherwise, increase ρ

BCL Method (LANCELOT)

$$\min\ \phi(x)\ \text{ st }\ c(x) = 0,\quad \ell \le x \le u$$

Bound-constrained augmented Lagrangian method

$$L(x, \hat{y}, \rho) = \phi(x) - \hat{y}^T c(x) + \tfrac{1}{2}\rho \|c(x)\|^2$$

- Subproblem: $\min_x\ L(x, \hat{y}, \rho)$ st $\ell \le x \le u$
- Solve to get $\hat{x}$ (optimality tol $\omega \to 0$)
- If $\|c(\hat{x})\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)
- Otherwise, increase ρ
- Repeat

BCL Method for QP

$$\min_{x} \ \phi(x) \ \ \text{st} \ \ Ax = b, \ \ \ell \le x \le u$$

BCL subproblem

$$\begin{aligned} \min_{x,\,r} \ L(x, \hat{y}, \rho) \quad &= \quad \phi(x) + \hat{y}^T r + \tfrac{1}{2}\rho\|r\|^2 \\ & \quad\ \ Ax + r = b, \qquad \ell \le x \le u \end{aligned}$$

BCL Method for QP

$$\min_{x} \ \phi(x) \ \text{ st } \ Ax = b, \ \ \ell \le x \le u$$

BCL subproblem

$$\begin{aligned} \min_{x,\,r} \ L(x, \hat{y}, \rho) \quad &= \quad \phi(x) + \hat{y}^T r + \tfrac{1}{2}\rho\|r\|^2 \\ & \quad\quad Ax + r = b, \qquad \ell \le x \le u \end{aligned}$$

- Solve subproblem to get $\hat{x}$, $\hat{r}$ (optimality tol $\omega \to 0$)

BCL Method for QP

$$\min_{x} \ \phi(x) \ \text{ st } \ Ax = b, \ \ \ell \le x \le u$$

BCL subproblem

$$\begin{aligned} \min_{x,\,r} \ L(x, \hat{y}, \rho) \ &= \ \phi(x) + \hat{y}^T r + \tfrac{1}{2}\rho\|r\|^2 \\ & \quad Ax + r = b, \qquad \ell \le x \le u \end{aligned}$$

- Solve subproblem to get $\hat{x}$, $\hat{r}$ (optimality tol $\omega \to 0$)
- If $\|\hat{r}\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)

BCL Method for QP

$$\min_x \ \phi(x) \ \text{ st } \ Ax = b, \ \ \ell \le x \le u$$

BCL subproblem

$$\begin{aligned}\min_{x,\,r} \ L(x,\hat{y},\rho) \ &= \ \phi(x) + \hat{y}^T r + \tfrac{1}{2}\rho\|r\|^2 \\ & \quad\ Ax + r = b, \qquad \ell \le x \le u\end{aligned}$$

- Solve subproblem to get $\hat{x}$, $\hat{r}$ (optimality tol $\omega \to 0$)
- If $\|\hat{r}\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)
- Otherwise, increase ρ

BCL Method for QP

$$\min_x \ \phi(x) \ \text{ st } \ Ax = b, \ \ \ell \le x \le u$$

BCL subproblem

$$\begin{aligned} \min_{x,\,r} \ L(x, \hat{y}, \rho) \ &= \ \phi(x) + \hat{y}^T r + \tfrac{1}{2}\rho\|r\|^2 \\ & \quad Ax + r = b, \qquad \ell \le x \le u \end{aligned}$$

- Solve subproblem to get $\hat{x}$, $\hat{r}$ (optimality tol $\omega \to 0$)
- If $\|\hat{r}\| < \eta$, update $\hat{y}$ (feasibility tol $\eta \to 0$)
- Otherwise, increase ρ
- Repeat

QPBCL

QPBCL: Active-set method for convex QP

$$AP = \begin{pmatrix} A_k & N_k \end{pmatrix}$$

QPBCL: Active-set method for convex QP

$$AP = \begin{pmatrix} A_k & N_k \end{pmatrix}$$

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \frac{1}{\rho} I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ \frac{1}{\rho}(\hat{y} - y) + r \end{pmatrix}$$

$$\Delta r = \frac{1}{\rho}(y + \Delta y - \hat{y}) - r$$

QPBCL: Active-set method for convex QP

$$AP = \begin{pmatrix} A_k & N_k \end{pmatrix}$$

$$\begin{pmatrix} -(H_k + \delta I) & A_k^T \\ A_k & \frac{1}{\rho} I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_k - A_k^T y \\ \frac{1}{\rho}(\hat{y} - y) + r \end{pmatrix}$$

$$\Delta r = \frac{1}{\rho}(y + \Delta y - \hat{y}) - r$$

- Same system as QPBLUR – just different rhs
- Still single phase
- Still use LUSOL, MA57, UMFPACK, ...
- Still use Hanh's block-LU updates

Numerical Results

90 Meszaros QP problems

QPBCL solves with smaller penalty ρ (hence larger $\mu \equiv 1/\rho$) compared to QPBLUR

COPS 3.0 problems with many degrees of freedom

	m	n	*dof*
dirichlet	42	8981	5355
henon	82	10801	9410
lane_emden	82	19240	5414
minsurf	0	5000	4782

- SNOPT/AMPL
- Major iterations generate QPs, solved by SQOPT
- QP data loaded into MATLAB, solved by QPBCL

COPS 3.0 problems with many degrees of freedom

	m	n	*dof*
dirichlet	42	8981	5355
henon	82	10801	9410
lane_emden	82	19240	5414
minsurf	0	5000	4782

- SNOPT/AMPL
- Major iterations generate QPs, solved by SQOPT
- QP data loaded into MATLAB, solved by QPBCL

Compare SQOPT iterations and QPBCL iterations

COPS 3.0 problems with many degrees of freedom

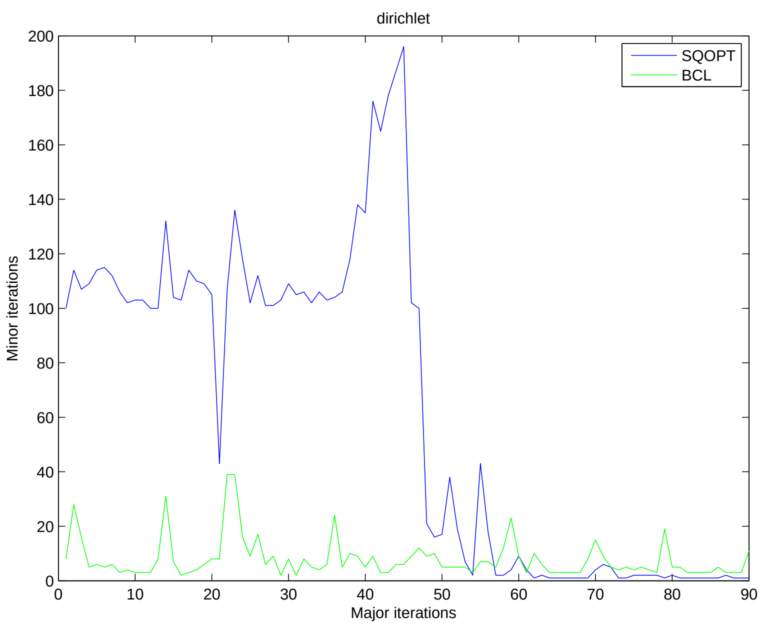
	m	n	*dof*
dirichlet	42	8981	5355
henon	82	10801	9410
lane_emden	82	19240	5414
minsurf	0	5000	4782

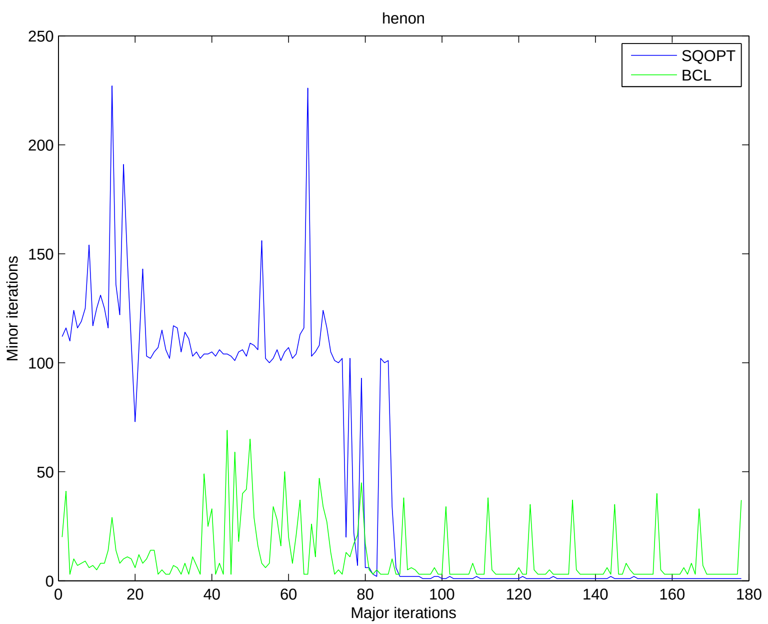
- SNOPT/AMPL
- Major iterations generate QPs, solved by SQOPT
- QP data loaded into MATLAB, solved by QPBCL

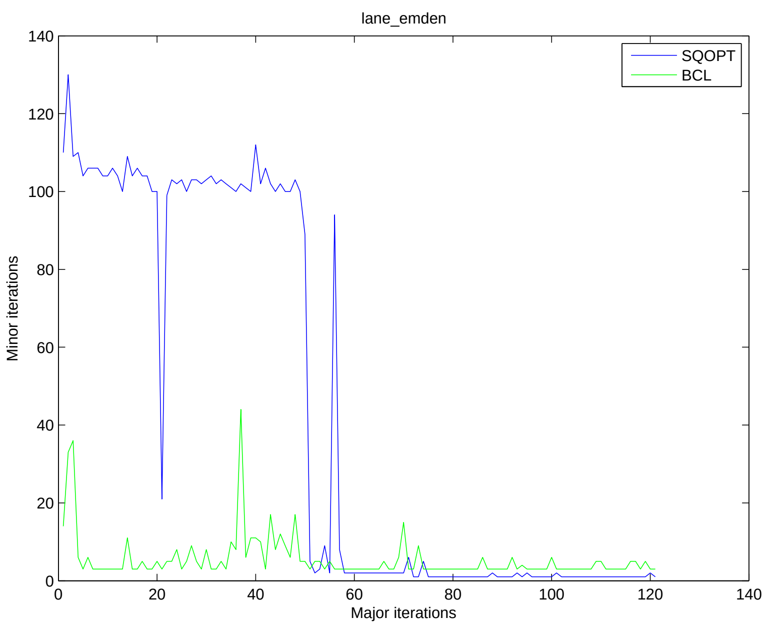
Compare SQOPT iterations and QPBCL iterations

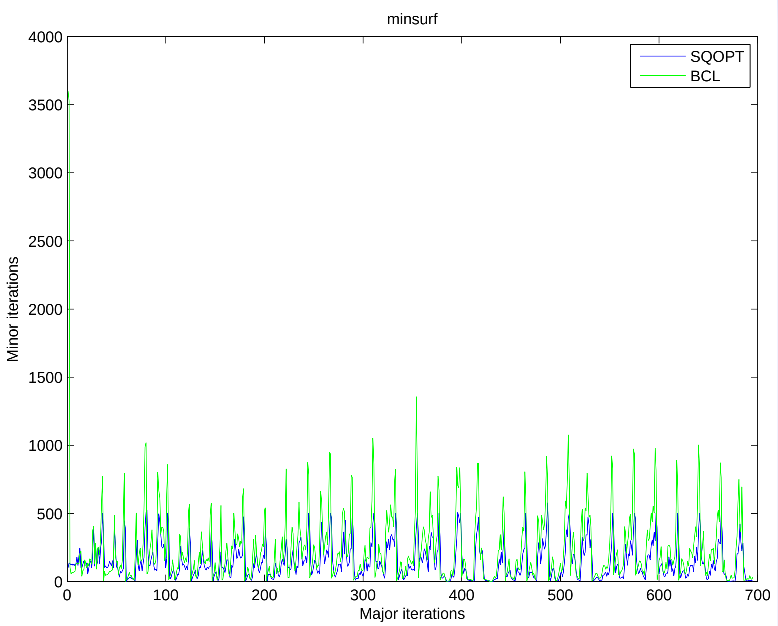
There are 3 kinds of people: Those who can count, and those who cannot.

— George Carlin


dirichlet
Minor iterations
Major iterations
SQOPT
BCL
0
20
40
60
80
100
120
140
160
180
200
0
10
20
30
40
50
60
70
80
90


henon
SQOPT
BCL
Minor iterations
Major iterations
0
50
100
150
200
250
0
20
40
60
80
100
120
140
160
180


lane_emden
SQOPT
BCL
Minor iterations
Major iterations
0
20
40
60
80
100
120
140


minsurf
4000
3500
3000
2500
2000
1500
1000
500
0
Minor iterations
SQOPT
BCL
0
100
200
300
400
500
600
700
Major iterations

Augmented Lagrangian methods for QP

- Conn, Gould, and Toint (1992)
  LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer-Verlag

Augmented Lagrangian methods for QP

- Conn, Gould, and Toint (1992)
  LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer-Verlag

- Nocedal and Wright (2006)
  Chapter 17 of Numerical Optimization, Springer

Augmented Lagrangian methods for QP

- Conn, Gould, and Toint (1992)
  LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer-Verlag

- Nocedal and Wright (2006)
  Chapter 17 of Numerical Optimization, Springer

- M. P. Friedlander and S. Leyffer (2008)
  Global and finite termination of a two-phase augmented Lagrangian filter method for general QP, SISC 30

Augmented Lagrangian methods for QP

- Conn, Gould, and Toint (1992)
  LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer-Verlag

- Nocedal and Wright (2006)
  Chapter 17 of Numerical Optimization, Springer

- M. P. Friedlander and S. Leyffer (2008)
  Global and finite termination of a two-phase augmented Lagrangian filter method for general QP, SISC 30

- Philip Gill and Elizabeth Wong (2009–$\cdots$)
  Continuing Hanh's f90 QPBLU for use in SNOPT
  BCL approach for indefinite QP (2nd derivatives in SNOPT)

Augmented Lagrangian methods for QP

- Conn, Gould, and Toint (1992)
  LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Springer-Verlag
- Nocedal and Wright (2006)
  Chapter 17 of Numerical Optimization, Springer
- M. P. Friedlander and S. Leyffer (2008)
  Global and finite termination of a two-phase augmented Lagrangian filter method for general QP, SISC 30
- Philip Gill and Elizabeth Wong (2009–$\cdots$)
  Continuing Hanh's f90 QPBLU for use in SNOPT
  BCL approach for indefinite QP (2nd derivatives in SNOPT)
- C. Kirches, H. G. Bock, J. P. Schlöder and S. Sager (2009)
  A factorization with update procedures for a KKT matrix arising in direct optimal control
  Active-set method with block-structured KKT factorization

Even convex QP isn't easy

but conferences help us make progress

Even convex QP isn't easy

but conferences help us make progress

Merci beaucoup a tous

Iain Duff
Serge Gratton
Xavier Vasseur
Brigitte Yzel

STAE
RTRA
CERFACS