Generalized Trust Region Subproblem

Ting Kei Pong
Combinatorics & Optimization, University of Waterloo
Waterloo

UW Optimization Seminar
Nov 2012
(Joint work with Henry Wolkowicz)

Outline

- Generalized trust region subproblem.
- Optimality conditions.
- Preprocessing, easy case and hard cases.
- (Extended) Rendl-Wolkowicz algorithm.
- Preliminary numerical results.

Generalized Trust Region Subproblem (GTRS)

The generalized trust region subproblem:

$$
\begin{aligned}
q^* := \min\ q(x) &:= x^T A x - 2a^T x \\
\text{s.t. } l &\leq \underbrace{x^T B x - 2b^T x}_{q_1(x)} \leq u,
\end{aligned}
$$

where $A, B \in \mathcal{S}^n$, $a, b \in \mathbb{R}^n$.

- Reduces to trust region subproblem (TRS) when $B = I$, $b = 0$, $0 = l < u$.
- Possible applications: TR methods for unconstr. min., subproblems for constrained optimization, regularization of ill-posed problems...

Assumptions on GTRS

- $B \neq 0$
- GTRS is feasible and bounded below.
- The following constraint qualification holds

$$\text{(CQ)} \qquad \operatorname{tr}(B\hat{X}) - 2b^T\hat{x} \in \operatorname{ri}([\ell, u]), \qquad \text{for some } \hat{X} \succ \hat{x}\hat{x}^T.$$

- The dual problem is feasible; i.e.,

$$\exists\lambda \text{ s.t. } a - \lambda b \in \mathcal{R}(A - \lambda B) \text{ and } A - \lambda B \succeq 0.$$

Optimality conditions I

Fact 1 (P, Wolkowicz '12): The optimal value of GTRS equals that of its SDP relaxation and its dual, where

$$\begin{array}{rll} p^*_{SDP} := & \inf & \mathrm{tr}(AX) - 2a^T x \\ & \text{s.t.} & \ell \leq \mathrm{tr}(BX) - 2b^T x \leq u, \\ & & X \succeq xx^T, \end{array}$$

$$\begin{array}{rll} d^*_{DSDP} := & \sup & \ell\lambda_+ - u\lambda_- - \gamma \\ & \text{s.t.} & \begin{bmatrix} \gamma & -(a - \lambda b)^T \\ -(a - \lambda b) & A - \lambda B \end{bmatrix} \succeq 0. \end{array}$$

Moreover, the dual optimal value is attained.

Optimality conditions II

Fact 2 (P, Wolkowicz '12): x^* optimal for GTRS iff $\exists\lambda^*$ s.t.

$$\left.\begin{array}{c}(A-\lambda^* B)x^* = a - \lambda^* b,\\ A-\lambda^* B \succeq 0,\end{array}\right\} \quad \text{dual feasibility}$$

$$\ell \le {x^*}^T B x^* - 2b^T x^* \le u, \quad \text{primal feasibility}$$

$$\left.\begin{array}{r}(\lambda^*)_+(\ell - {x^*}^T B x^* + 2b^T x^*) = 0,\\ (\lambda^*)_-({x^*}^T B x^* - 2b^T x^* - u) = 0.\end{array}\right\} \quad \text{complementary slackness}$$

Optimality conditions II

Fact 2 (P, Wolkowicz '12): x^* optimal for GTRS iff $\exists\lambda^*$ s.t.

$$\left.\begin{array}{c}(A-\lambda^* B)x^* = a - \lambda^* b,\\ A-\lambda^* B \succeq 0,\end{array}\right\}\quad \text{dual feasibility}$$

$$\ell \le {x^*}^T B x^* - 2b^T x^* \le u, \quad \text{primal feasibility}$$

$$\left.\begin{array}{r}(\lambda^*)_+(\ell - {x^*}^T B x^* + 2b^T x^*) = 0,\\ (\lambda^*)_-({x^*}^T B x^* - 2b^T x^* - u) = 0.\end{array}\right\}\quad \text{complementary slackness}$$

Fact 3 (Gay '81; Moré, Sorensen '83): x^* is optimal for TRS iff $\exists\lambda^*$ s.t.

$$\left.\begin{array}{c}(A-\lambda^* I)x^* = a,\\ A-\lambda^* I \succeq 0, \lambda^* \le 0,\end{array}\right\}\quad \text{dual feasibility}$$

$$\|x^*\|^2 \le u, \quad \text{primal feasibility}$$

$$\lambda^*(u - \|x^*\|^2) = 0. \quad \text{complementary slackness}$$

Further Assumptions on GTRS

Assume $\exists\hat{\lambda}$ s.t.

$$A - \hat{\lambda} B \succ 0.$$

Further Assumptions on GTRS

Assume $\exists \hat{\lambda}$ s.t.

$$A - \hat{\lambda} B \succ 0.$$

Then:

- GTRS has an optimal solution.
- There exists invertible matrix S s.t.

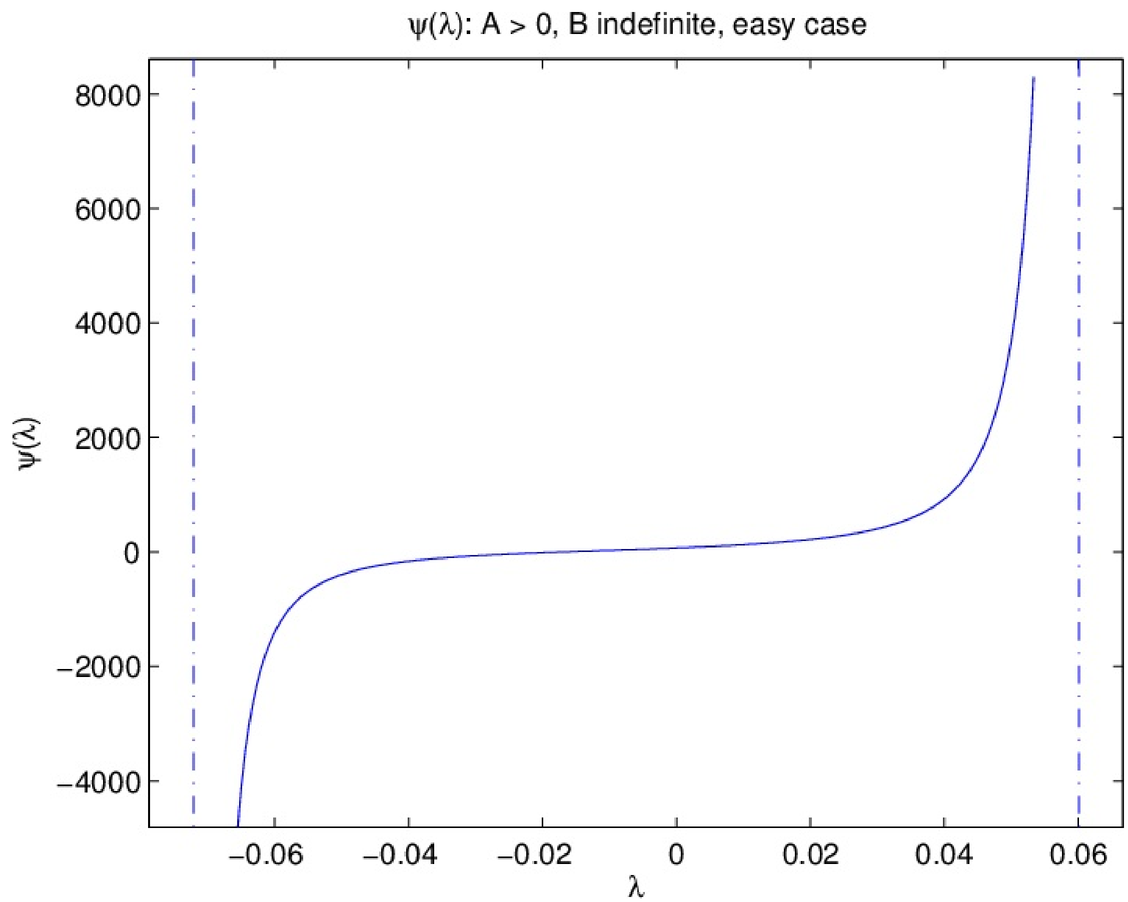
$$A = S\mathrm{diag}(\alpha)S^T \quad B = S\mathrm{diag}(\beta)S^T.$$

- $$A - \lambda B \succeq 0 \;\Leftrightarrow \underline{\lambda} := \max_{\{i:\beta_i<0\}} \frac{\alpha_i}{\beta_i} \leq \lambda \leq \min_{\{i:\beta_i>0\}} \frac{\alpha_i}{\beta_i} =: \overline{\lambda}.$$

- Denote $x(\lambda) := (A - \lambda B)^{-1}(a - \lambda b)$ on $(\underline{\lambda}, \overline{\lambda})$.

Preprocessing: reduce to equality case

Suppose $\underline{\lambda} < 0 < \overline{\lambda}$ and consider $\psi(\lambda) := q_1(x(\lambda))$.



Preprocessing: reduce to equality case

Since $\underline{\lambda} < 0 < \overline{\lambda}$, we have $A \succ 0$. Consider $q_1(x(0))$.

- If $\ell \leq q_1(x(0))) \leq u$, then $x(0)$ is an interior solution.
- If $q_1(x(0)) < \ell$, then $\lambda^* > 0$. Constraint reduces to $q_1(x) = \ell$.
- If $q_1(x(0)) > u$, then $\lambda^* < 0$. Constraint reduces to $q_1(x) = u$.

Preprocessing: reduce to equality case

Since $\underline{\lambda} < 0 < \overline{\lambda}$, we have $A \succ 0$. Consider $q_1(x(0))$.

- If $\ell \leq q_1(x(0))) \leq u$, then $x(0)$ is an interior solution.
- If $q_1(x(0)) < \ell$, then $\lambda^* > 0$. Constraint reduces to $q_1(x) = \ell$.
- If $q_1(x(0)) > u$, then $\lambda^* < 0$. Constraint reduces to $q_1(x) = u$.

Other cases ($\underline{\lambda} \geq 0$ or $\overline{\lambda} \leq 0$) considered similarly.

Denote the resulting equality constraint as $q_1(x) = s$.

Easy/Hard Cases for GTRS

Consider $\psi(\lambda) = s$ on $(\underline{\lambda}, \overline{\lambda})$.

	Easy case	Hard case 1	Hard case 2
$\underline{\lambda}, \overline{\lambda}$ finite	$a - \overline{\lambda} b \notin \mathcal{R}(A - \overline{\lambda} B)$ and $a - \underline{\lambda} b \notin \mathcal{R}(A - \underline{\lambda} B)$ $(\Rightarrow \underline{\lambda} < \lambda^* < \overline{\lambda})$	$a - \overline{\lambda} b \perp \mathcal{N}(A - \overline{\lambda} B)$ or $a - \underline{\lambda} b \perp \mathcal{N}(A - \underline{\lambda} B)$ $\underline{\lambda} < \lambda^* < \overline{\lambda}$	$a - \overline{\lambda} b \perp \mathcal{N}(A - \overline{\lambda} B)$ $\lambda^* = \overline{\lambda}$ or $a - \underline{\lambda} b \perp \mathcal{N}(A - \underline{\lambda} B)$ $\lambda^* = \underline{\lambda}$
$\underline{\lambda} = -\infty$	$a - \overline{\lambda} b \notin \mathcal{R}(A - \overline{\lambda} B)$ $(\Rightarrow -\infty < \lambda^* < \overline{\lambda})$	$a - \overline{\lambda} b \perp \mathcal{N}(A - \overline{\lambda} B)$ $-\infty < \lambda^* < \overline{\lambda}$	$a - \overline{\lambda} b \perp \mathcal{N}(A - \overline{\lambda} B)$ $\lambda^* = \overline{\lambda}$
$\overline{\lambda} = \infty$	$a - \underline{\lambda} b \notin \mathcal{R}(A - \underline{\lambda} B)$ $(\Rightarrow \underline{\lambda} < \lambda^* < \infty)$	$a - \underline{\lambda} b \perp \mathcal{N}(A - \underline{\lambda} B)$ $\underline{\lambda} < \lambda^* < \infty$	$a - \underline{\lambda} b \perp \mathcal{N}(A - \underline{\lambda} B)$ $\lambda^* = \underline{\lambda}$

Hard Case 2: Explicit Solution

Fact 4 (Moré '93): If $\overline{\lambda}$ is finite, then

$$v^T B v > 0, \;\; \forall v \in \mathcal{N}(A - \overline{\lambda} B) \backslash \{0\}.$$

Similarly, if $\underline{\lambda}$ is finite, then

$$v^T B v < 0, \;\; \forall v \in \mathcal{N}(A - \underline{\lambda} B) \backslash \{0\}.$$

Hard Case 2: Explicit Solution

Fact 4 (Moré ’93): If $\overline{\lambda}$ is finite, then

$$v^T B v > 0, \quad \forall v \in \mathcal{N}(A - \overline{\lambda} B) \backslash \{0\}.$$

Similarly, if $\underline{\lambda}$ is finite, then

$$v^T B v < 0, \quad \forall v \in \mathcal{N}(A - \underline{\lambda} B) \backslash \{0\}.$$

Since ψ is non-decreasing on $(\underline{\lambda}, \overline{\lambda})$,

- When $\lambda^* = \overline{\lambda}$, the solution is $x(\overline{\lambda}) + v$ for some $v \in \mathcal{N}(A - \overline{\lambda} B)$;
- When $\lambda^* = \underline{\lambda}$, the solution is $x(\underline{\lambda}) + v$ for some $v \in \mathcal{N}(A - \underline{\lambda} B)$.

Some Representative Algorithms for TRS

Newton-type method on the dual function/secular function:

- Moré and Sorensen algorithm (Moré, Sorensen ’83);
- Generalized Lanzcos Trust-Region algorithm (Gould et al. ’99, ’10).

Reformulation into parameterized eigenvalue problem:

- Rendl-Wolkowicz algorithm (Rendl, Wolkowicz ’97; Fortin, Wolkowicz ’03);
- Large Scale Trust-Region Subproblem algorithm (Rojas et al. ’00, ’09).

Extended Rendl-Wolkowicz Algorithm

Consider equality constrained problem $\mu^* = \min_{x^TBx - 2b^Tx = s} x^TAx - 2a^Tx$ and assume $s \neq -1$.

$$\begin{aligned}
\mu^* &= \min_{x^TBx - 2b^Tx = s,\ y_0^2 = 1} x^TAx - 2a^Txy_0 \\
&= \max_t \min_{x^TBx - 2b^Txy_0 = s,\ y_0^2 = 1} x^TAx - 2a^Txy_0 + ty_0^2 - t \\
&\geq \max_t \min_{x^TBx - 2b^Txy_0 + y_0^2 = s+1} x^TAx - 2a^Txy_0 + ty_0^2 - t \quad {**\ \text{eig prob}\ **} \\
&\geq \max_{t,\lambda} \min_{x,y_0} x^TAx - 2a^Txy_0 + ty_0^2 - t + \lambda(x^TBx - 2b^Txy_0 + y_0^2 - s - 1) \\
&= \max_{r,\lambda} \min_{x,y_0} x^TAx - 2a^Txy_0 + ry_0^2 - r + \lambda(x^TBx - 2b^Txy_0 - s) \\
&= \max_{\lambda} \min_{x,y_0^2=1} x^TAx - 2a^Txy_0 + \lambda(x^TBx - 2b^Txy_0 - s) = \mu^*.
\end{aligned}$$

Extended Rendl-Wolkowicz Algorithm

From the eig prob:

$$\mu^* = \max_t \underbrace{\min_{x^T Bx - 2b^T xy_0 + y_0^2 = s+1} x^T Ax - 2a^T xy_0 + ty_0^2}_{k_0(t)} - t$$

$$= \max_t k(t) := k_0(t) - t$$

where

$$k_0(t) := \inf_{x^T Bx - 2b^T xy_0 + y_0^2 = s+1} x^T Ax - 2a^T xy_0 + ty_0^2$$

$$= \inf \left\{ y^T \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix} y : \ y^T \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} y = s+1 \right\}.$$

$$k_0(t) = \inf \left\{ y^T \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix} y : \; y^T \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} y = s+1 \right\}.$$

- In TRS, $B = I$, $b = 0$ and $s > 0$, evaluating $k_0(t) \Leftrightarrow$ finding min. eig.;
- When $\begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} \succ 0$, evaluating $k_0(t) \Leftrightarrow$ finding min. gen. eig.;

$$k_0(t) = \inf\left\{ y^T \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix} y : \; y^T \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} y = s+1 \right\}.$$

- In TRS, $B = I$, $b = 0$ and $s > 0$, evaluating $k_0(t) \Leftrightarrow$ finding min. eig.;
- When $\begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} \succ 0$, evaluating $k_0(t) \Leftrightarrow$ finding min. gen. eig.;
- In general, using a $\hat{\lambda}$ with

$$\tilde{A} := \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix} - \hat{\lambda} \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} \succ 0,$$

evaluating $k_0(t) \Leftrightarrow$ finding min. gen. eig.

Suppose $s + 1 \neq 0$.

Suppose $s + 1 \neq 0$.

Fact 5 (P, Wolkowicz '12): Suppose that B is either indefinite or positive semidefinite. Let

$$t_{-} := -\inf_{x^T Bx - 2b^T x = -1} x^T Ax - 2a^T x,$$
$$s_{-} := \inf x^T Bx - 2b^T x.$$

- If $s_{-} < -1$, then $\mathrm{dom}(k) = [t_{-}, \infty)$;
- If $s_{-} > -1$, then $k(t)$ is defined everywhere.

In either case, k is continuous and concave on its domain, and is differentiable in the interior of the domain except possibly at one point.

Suppose $s + 1 \neq 0$ and t^* be the global maximizer of $k(t)$.

Suppose $s+1 \neq 0$ and t^* be the global maximizer of $k(t)$.

Fact 6 (P, Wolkowicz '12): In the easy case/hard case 1, $t^* > t_-$ and there exists a generalized eigenvalue $\lambda^* \in (\underline{\lambda}, \overline{\lambda})$ of the matrix pair

$$\left(\begin{bmatrix} t^* & -a^T \\ -a & A \end{bmatrix}, \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} \right)$$

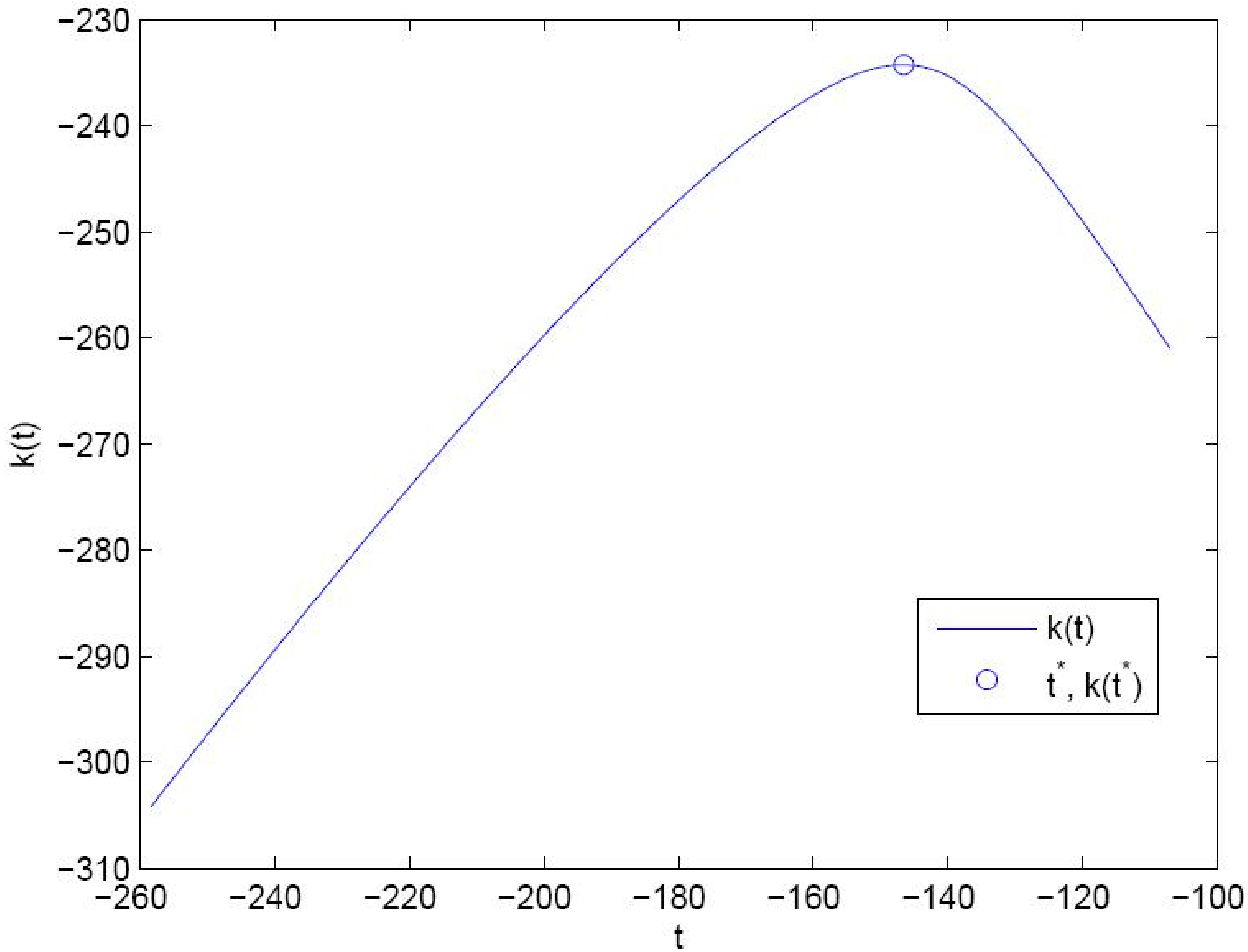
with a generalized eigenvector $y(t^*) = \begin{bmatrix} y_0(t^*) \\ w(t^*) \end{bmatrix}$ satisfying

$$y_0(t^*) \neq 0 \text{ and } y(t^*)^T \begin{bmatrix} 1 & -b^T \\ -b & B \end{bmatrix} y(t^*) = \operatorname{sign}(s+1).$$

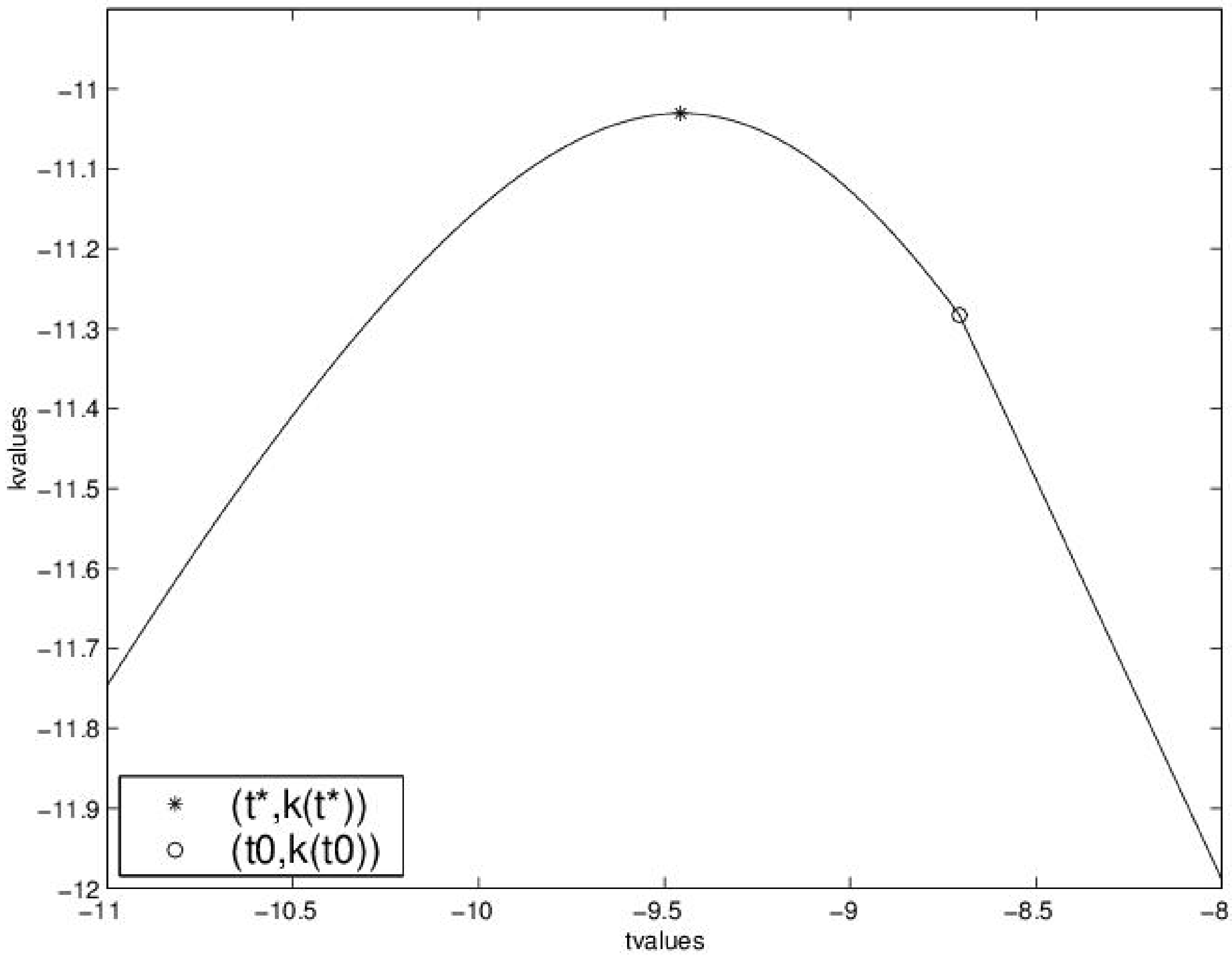
Moreover, $x^* := \frac{w(t^*)}{y_0(t^*)}$ solves GTRS, i.e.,

$$x^* = \underset{x^T B x - 2b^T x = s}{\arg\min} \; x^T A x - 2a^T x.$$

$k(t)$ in Easy Case, $s_- > -1$



$k(t)$ in Hard Case 1, $s_- > -1$



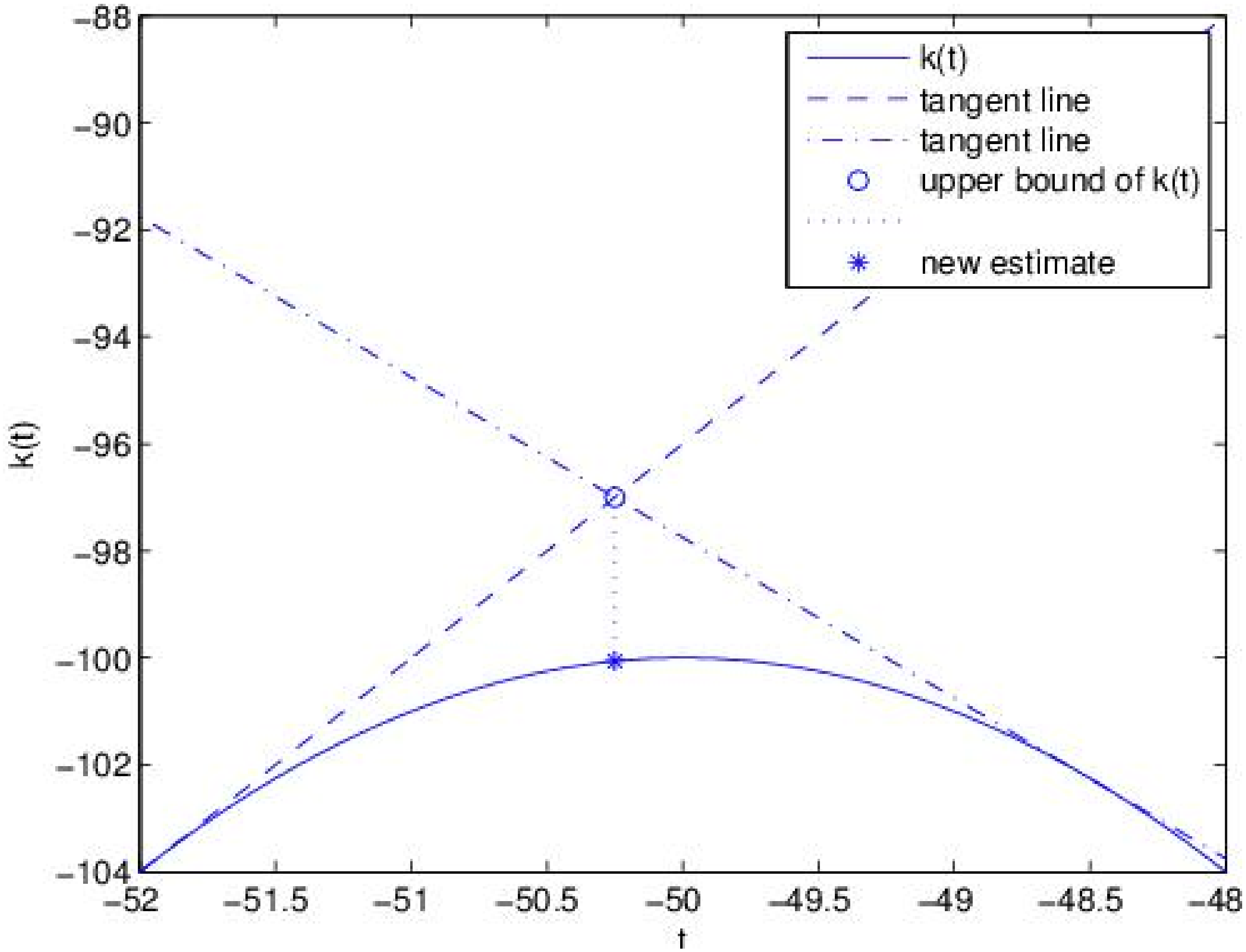
Maximizing $k(t)$

- Triangle Interpolation.
- Vertical Cut.
- Inverse Linear Interpolation: consider

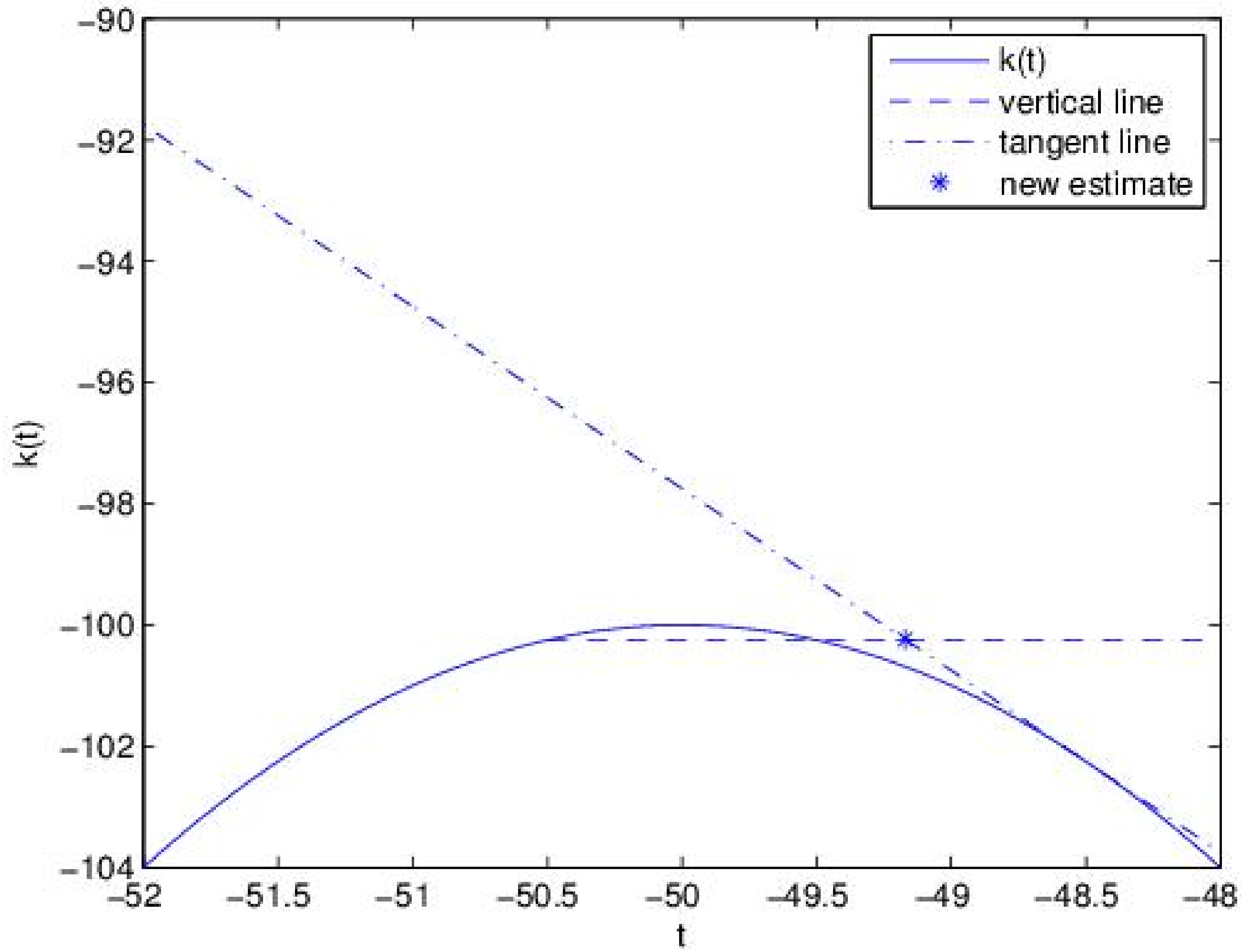
$$\Psi(t) := \sqrt{|s+1|} - \frac{1}{y_0(t)}.$$

Approximate the inverse of Ψ by a linear function $t(\Psi) = a\Psi + b$, and set $t_+ = t(0)$.

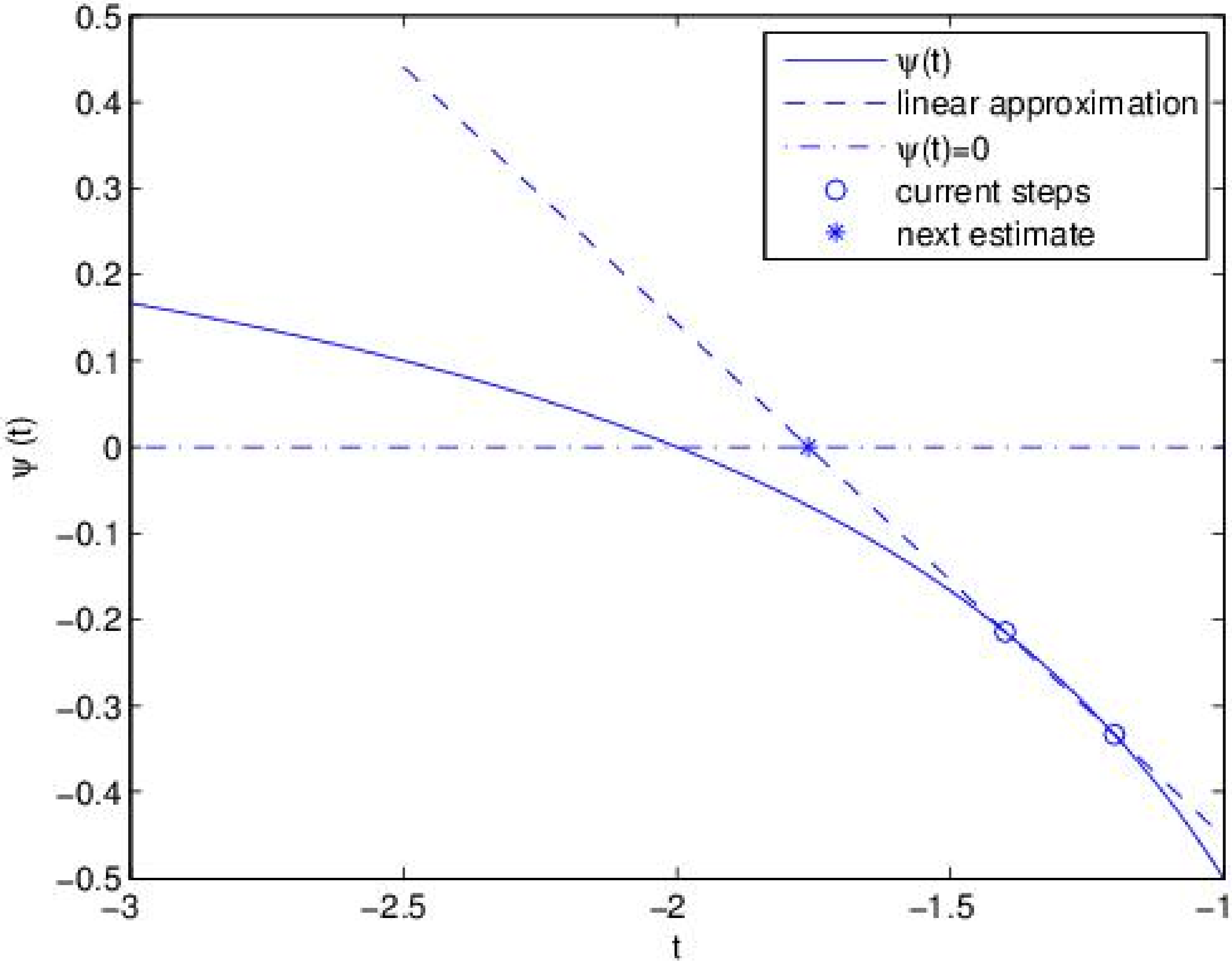
Triangle Interpolation



Vertical Cut



Inverse Linear Interpolation



Simulations: $b = 0$

- Use eigifp (Golub, Ye '02) to compute generalized eigenvalues.
- Termination for RW algorithm:

$$\max\left\{\frac{|k'(t)|^2}{|s+1|^2}, \frac{|q_0(x)-k(t)|}{|{q_0}^{\text{best}}|+1}, \frac{|q_1(x)-s|}{|s|+1}, \frac{\|Ax-\lambda Bx-a\|^2}{(\|A\|_2+\|a\|+1)^2}\right\} < 10^{-13}$$

- Compare with Newton + Armijo on dual function, termination:

$$\max\left\{\frac{|q_0(x)-d(\lambda)|}{|q_0(x)|+1}, \frac{|q_1(x)-s|}{|s|+1}, \frac{\|Ax-\lambda Bx-a\|^2}{(\|A\|_2+\|a\|+1)^2}\right\} < 10^{-12}.$$

- Two 2.4 GHz quad-core Intel E5620 Xeon 64-bit CPUs, 48 GB RAM, Matlab 7.14 (R2012a).

Simulations: $B \succ 0$, $b = 0$

- Notice that $s_- = 0 > -1$.

- One can show that (P, Wolkowicz '12)

$$\bar{\lambda} - \sqrt{\frac{a^T B^{-1} a}{s}} \leq t^* \leq \bar{\lambda} + \sqrt{s a^T B^{-1} a}.$$

  Use these end points to initialize RW algorithm.

- Newton method initialized at $\lambda^0 = \bar{\lambda} - 1$.

- Consider two cases: A indefinite or $A \succ 0$.

Simulations: $B \succ 0$, $b = 0$, indef. A

For each $n = 10000, 15000, 20000$, generate 10 "easy instances":

```
B = sprandsym(n,1e-2,0.1,2); A = sprandsym(n,1e-2);
lambda = eigifp(A,B,1,opts) - 5 - 5*rand(1);
x = randn(n,1)/10; a = (A - lambda*B)*x; s = x'*B*x;
u = 1.2*s; l = 0.8*s;
```

Apply Newton to dual function with $s = u$.

Table 1: Indef. A and $B \succ 0$.

n	RW iter/cpu/fval/feas$_{\text{eq}}$	Newton+Armijo iter/cpu/fval/feas$_{\text{eq}}$
10000	6/4.92/-4.106880221e+4/4.1e-13	9/6.89/-4.106880222e+4/5.5e-07
15000	6/12.89/-9.910679264e+4/5.5e-13	9/17.56/-9.910679264e+4/1.0e-07
20000	6/19.26/-1.443904856e+5/1.6e-12	9/29.85/-1.443904856e+5/8.7e-07

Simulations: $B \succ 0$, $b = 0$, $A \succ 0$

For each $n = 10000, 15000, 20000$, generate 10 "easy instances":

```
B = sprandsym(n,1e-2,0.1,2);
A = sprandsym(n,1e-2,0.1,2) + 10*B;
lambda = eigifp(A,B,1,opts) - 5 - 5*rand(1);
x = randn(n,1)/10; a = (A - lambda*B)*x; s = x'*B*x;
```

Apply Newton to dual function with $s = \ell$, with u, ℓ similarly defined.

Table 2: $A \succ 0$ and $B \succ 0$.

n	RW iter/cpu/fval/feas$_{\text{eq}}$	Newton+Armijo iter/cpu/fval/feas$_{\text{eq}}$
10000	6/3.11/-1.122480624e+4/1.8e-13	10/5.73/-1.122480624e+4/1.0e-08
15000	6/7.91/-2.160566627e+4/5.9e-13	9/13.64/-2.160566627e+4/1.2e-08
20000	6/14.16/-3.175205009e+4/4.5e-13	9/25.67/-3.175205009e+4/1.6e-07

Simulations: indef. B, $b = 0$

- Assume $A \succ 0$; then

$$\overline{\lambda} = \frac{-1}{\texttt{eigifp(-B,A,1,opts)}}, \quad \underline{\lambda} = \frac{1}{\texttt{eigifp(B,A,1,opts)}}.$$

- No efficient way to initialize RW algorithm. Have to artificially generate t_- and the corresponding λ_-.

- Newton method initialized at $\lambda^0 = \frac{\overline{\lambda}+\underline{\lambda}}{2}$.

- First apply RW algorithm. Then apply Newton to dual with $s = u$ or $s = \ell$, determined from the preprocessing.

Simulations: indef. B, $b = 0$, $A \succ 0$

For each $n = 10000, 15000, 20000$, generate 10 "easy instances":

```
A = sprandsym(n,1e-2,0.1,2); B1 = sprandsym(n,1e-2);
lambda = (lambdaup + lambdalow)/2;
x = randn(n,1)/10; a = (A - lambda*B1)*x; s1 = x'*B1*x;
lambda0 = lambda*(-abs(s1)); t0 = 2*a'*x - x'*A*x;
B = B1/(-abs(s1)); s = randn(1)*sqrt(n); l = s; u = s + 1
```

Table 3: $A \succ 0$ and indef. B.

n	RW iter/cpu/fval/feas$_{\text{eq}}$	Newton+Armijo iter/cpu/fval/feas$_{\text{eq}}$
10000	6/9.49/-2.034919097e+3/1.1e-14	6/9.64/-2.034919097e+3/1.7e-08
15000	7/23.66/-3.385643481e+3/1.4e-14	6/23.85/-3.385643481e+3/8.9e-12
20000	7/43.75/-7.195068138e+3/1.1e-14	6/46.96/-7.195068132e+3/1.7e-07

Conclusion & Future work

- GTRS can be analyzed similarly as TRS, under suitable assumptions.
- The RW algorithm can be extended to solve GTRS.
- More numerical tests. Develop TRM with GTRS as subproblem.
- Estimating/bounding t_- when it is finite.

Thanks for coming!