Penalty Techniques in SQP and Interior-Point Algorithms

Frank E. Curtis

INFORMS Annual Meeting

October 11, 2009

Outline

Outline

Nonlinear constrained optimization

- Consider optimization problems of the form

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & c(x) \leq 0 \end{aligned} \tag{1.1}$$

  where f and c are smooth (equality constraints OK, too)

- Algorithms for the solution of (1.1) are plentiful:
  - Penalty methods
  - Augmented Lagrangian methods
  - Sequential quadratic programming (SQP) methods
  - Interior-point (barrier) methods
  - ...

Open issues

All contemporary methods, however, suffer key disadvantages:

- Methods fail to solve all problems.
- Methods fail when constraint qualifications are not satisfied.
- Methods fail when problems are ill-conditioned/infeasible.
- Methods are too expensive for large problems.

These issues are especially important for particular problem classes:

- Mathematical Programs with Equilibrium Constraints (MPECs)
- Mixed-Integer Nonlinear Programs (MINLPs)

Open issues

All contemporary methods, however, suffer key disadvantages:

- Methods fail to solve all problems.
- Methods fail when constraint qualifications are not satisfied.
- Methods fail when problems are ill-conditioned/infeasible.
- Methods are too expensive for large problems.

These issues are especially important for particular problem classes:

- Mathematical Programs with Equilibrium Constraints (MPECs)
- Mixed-Integer Nonlinear Programs (MINLPs)

In this talk, we focus on ill-conditioned/infeasible problems that may be large.

The ℓ_1 penalty function

- We focus on a tool that is not new:

$$\phi(x;\rho) := \rho f(x) + v(x),$$

where

$$v(x) := \sum_{j=1}^{t} \max\{c^j(x), 0\}.$$

- This is an exact penalty function, and as for using it in an algorithm:

Theorem

Suppose x_* is a stationary point for $\phi(x;\rho)$ for all $\rho \in [0, \bar{\rho}]$ for some $\bar{\rho} > 0$. If x_* is feasible for (1.1), then it satisfies the KKT conditions for (1.1). If x_* is not feasible, then it is an infeasible stationary point.

Minimizing the ℓ_1 penalty function

- Consider the optimization problem

$$\min\ (x-1)^2,\ \text{s.t. } x=0$$

- The corresponding penalty function is

$$\phi(x;\pi) = \rho(x-1)^2 + |x|$$

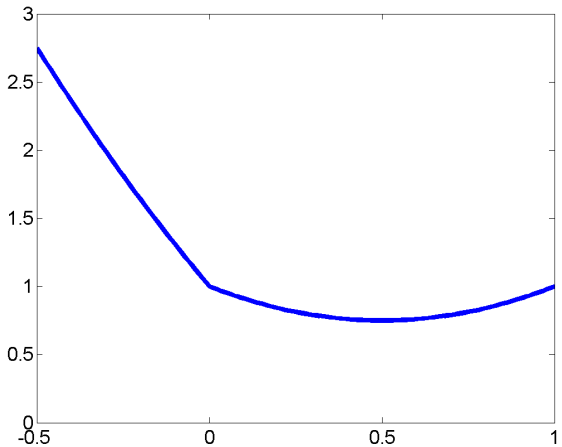


Figure: $\rho = 1$

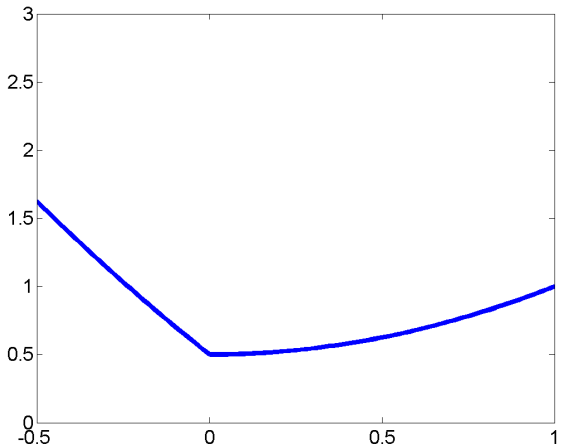


Figure: $\rho = \frac{1}{2}$

Outline

The search direction calculation

- Applying SQP methodology leads to the subproblem

$$\min_d \; f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T H_k d$$
$$\text{s.t. } c(x_k) + \nabla c(x_k)^T d \leq 0$$

(we'll discuss H_k more later...)

- If this problem is infeasible, then we must relax the constraints:

$$\min_{d,s} \; \rho(f(x_k) + \nabla f(x_k)^T d) + \sum s^j + \tfrac{1}{2} d^T H_k d$$
$$\text{s.t. } c(x_k) + \nabla c(x_k)^T d \leq s, \; s \geq 0$$

(or perform feasibility restoration...)

The search direction calculation

- Alternatively, we can define a local model of $\phi(x;\rho)$ at x_k:

$$q_k(d;\rho) := \rho(f(x_k)+\nabla f(x_k)^T d)+\sum \max\{c^j(x_k)+\nabla c^j(x_k)^T d, 0\}+\tfrac{1}{2} d^T H_k d$$

and solve

$$\min_d \; q_k(d;\rho). \tag{2.1}$$

- Problem (2.1) has the smooth reformulation

$$\begin{aligned} \min_{d,s} \; & \rho(f(x_k) + \nabla f(x_k)^T d) + \sum s^j + \tfrac{1}{2} d^T H_k d \\ \text{s.t. } & c(x_k) + \nabla c(x_k)^T d \le s, \; s \ge 0 \end{aligned}$$

(The same problem as on the previous slide!)

Penalty-SQP algorithm outline

- Determine an appropriate value ρ_k for the penalty parameter
- Compute (d_k, s_k) by solving the penalty-SQP subproblem
- Perform a backtracking line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \rho_k) \leq \phi(x_k; \rho_k) - \eta\alpha_k(q_k(0; \rho_k) - q_k(d_k; \rho_k))$$

- Update $x_{k+1} \leftarrow x_k + \alpha d_k$

Penalty-SQP algorithm outline

- Determine an appropriate value ρ_k for the penalty parameter
- Compute (d_k, s_k) by solving the penalty-SQP subproblem
- Perform a backtracking line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \rho_k) \leq \phi(x_k; \rho_k) - \eta \alpha_k \Delta q_k(d_k; \rho_k)$$

- Update $x_{k+1} \leftarrow x_k + \alpha d_k$

Convergence (in "nice" cases)

- Under certain assumptions (bounded gradients, full rank Jacobians, strict complementarity, sufficiently positive definite Hessians), SQP will identify the correct "active set" near the solution (Robinson, 1974)
- Thus, convergence becomes quadratic à la Newton's method

Open issues

- How do we update ρ so that we converge globally?
- How do we update ρ so that we converge locally at a fast rate, even if the problem is infeasible?

(Oftentimes, ρ is updated only if progress toward feasibility is not made over a number of iterations, and nothing special may be done if the problem is infeasible!)

Steering rules (Byrd, Nocedal, and Waltz, 2008)

Philosophically, we aim to:

- Ensure that during each iteration, progress toward attaining feasibility is proportional to the progress that would be made if we chose $\rho_k = 0$
- Ensure that ρ does not settle on a value such that minimizers of $\phi(x;\rho)$ do not correspond to solutions of the nonlinear program

Mathematically, we aim to:

- Choose ρ_k so that

$$\Delta l_k(d_k) \geq \epsilon_1 \Delta l_k(\overline{d}_k)$$

- Choose ρ_k so that

$$\Delta q_k(d_k;\rho_k) \geq \epsilon_2 \Delta l_k(\overline{d}_k)$$

Here, $\overline{d}_k$ is the solution when $\rho = 0$ and l_k is the linear part of q_k:

$$l_k(d) := \rho(f(x_k) + \nabla f(x_k)^T d) + \sum \max\{c^j(x_k) + \nabla c^j(x_k)^T d, 0\}$$

Fast convergence on infeasible problems

- We choose a Hessian that depends on ρ_k:

$$H(x_k, \lambda_k; \rho_k) := \rho_k \nabla^2 f(x_k) + \sum \lambda_k^j \nabla^2 c^j(x_k)$$

- If ρ is decreased by the steering rules, then in addition we decrease ρ so that it is proportional to the squared norm of the KKT conditions of problem (1.1) for $\rho = 0$, i.e., the square of

$$\left(\left\| \sum \lambda_k^j \nabla c^j(x_k) \right\| + \sum_{j \in \mathcal{S}_k} |c^j(x_k) \lambda_k^j| + \sum_{j \in \mathcal{V}_k} |c^j(x_k)(1 - \lambda_k^j)| \right)^2 \qquad (2.2)$$

Penalty-SQP algorithm

- Solve the SQP subproblem for $\rho = \rho_{k-1}$
- If d_k is linearly feasible, then continue; otherwise, set $\rho_k \leq \rho_{k-1}$ so that the steering rules are satisfied
- If $\rho_k < \rho_{k-1}$, then decrease ρ_k, if necessary, so that $\rho_k \propto (2.2)$
- Perform a backtracking line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \rho_k) \leq \phi(x_k; \rho_k) - \eta \alpha_k \Delta q_k(d_k; \rho_k)$$

- Update $x_{k+1} \leftarrow x_k + \alpha d_k$

Convergence (Byrd, Curtis, and Nocedal 2009)

Assumption

Suppose $x_\times$ is an infeasible stationary point and $\lambda_\times$ are Lagrange multipliers so that $(x_\times, \lambda_\times)$ is a KKT point for $\rho = 0$. Then,

- *(Smoothness) f and c are twice differentiable about $x_\times$*
- *(Regularity) The Jacobian of active constraints $\nabla c_{\mathcal{A}}(x_\times)$ has full row rank*
- *(Strict Complementarity) $\lambda_\times^j \in (0, 1)$ for $j \in \mathcal{A}$*
- *(2nd-order Sufficiency) $d^T H(x_\times, \lambda_\times; 0)d > 0$ for all $d \neq 0$ with $\nabla c_{\mathcal{A}}^T(x_\times)d = 0$.*

Theorem

Let $x_\times$ be an infeasible stationary point and suppose $\rho_k = O(\|(x_k, \lambda_k) - (x_\times, \lambda_\times)\|^2)$ for all large k. Then, if (x_k, λ_k) is sufficiently close to $(x_\times, \lambda_\times)$, the sequence $\{(x_k, \lambda_k)\}$ converges quadratically to $(x_\times, \lambda_\times)$.

Numerical results: Feasible instance (batch)

k	f_k	v_k	$E_k(\rho_{k-1})$	$E_k(0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+6.00e+02	4.70e+02	2.43e+02	2.27e+02	1.00e+00	2.48e+01	1.00e+00
1	+2.36e+02	2.45e+02	6.35e+01	8.24e+01	1.00e+00	2.89e+00	1.00e+00
2	+1.04e+02	2.24e+02	1.03e+01	2.76e+01	1.00e-01	7.69e+00	5.00e-01
3	+4.42e+02	1.58e+02	3.21e+00	1.63e+01	1.00e-02	1.10e+01	5.00e-01
4	+2.86e+03	8.21e+01	2.64e+00	1.00e+01	1.00e-03	5.64e+00	1.00e+00
5	+2.97e+04	1.05e+01	3.42e+00	4.83e+00	1.00e-04	3.05e+00	1.00e+00
6	+5.69e+04	3.08e+00	4.90e-01	2.53e+00	1.00e-04	1.66e+00	1.00e+00
7	+6.28e+04	1.89e+00	4.16e-02	1.65e+00	1.00e-05	1.50e+00	1.00e+00
8	+9.33e+04	2.88e-01	2.50e-01	3.44e-01	1.00e-05	9.55e-01	1.00e+00
9	+1.03e+05	2.07e-02	1.86e-02	2.44e-02	1.00e-05	2.17e-01	1.00e+00
10	+1.04e+05	4.24e-04	3.59e-04	4.99e-04	1.00e-05	1.54e-03	1.00e+00
11	+1.04e+05	1.31e-07	7.42e-08	--------	--------	--------	--------

Numerical results: Infeasible instance (batchmod)

k	f_k	v_k	$E_k(\rho_{k-1})$	$E_k(0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+6.00e+02	4.75e+02	2.43e+02	2.25e+02	1.00e+00	2.85e+01	1.00e+00
1	+2.36e+02	2.46e+02	6.25e+01	8.13e+01	1.00e+00	3.38e+00	1.00e+00
2	+1.04e+02	2.25e+02	1.03e+01	2.84e+01	1.00e-01	1.03e+01	5.00e-01
3	+4.35e+02	1.61e+02	3.35e+00	1.78e+01	1.00e-02	1.33e+01	5.00e-01
4	+2.85e+03	8.52e+01	2.75e+00	1.09e+01	1.00e-03	6.41e+00	1.00e+00
5	+3.01e+04	1.39e+01	3.55e+00	3.58e+00	1.00e-04	3.30e+00	1.00e+00
6	+5.72e+04	6.13e+00	5.00e-01	1.78e+00	1.00e-04	1.25e+00	1.00e+00
7	+6.35e+04	4.94e+00	3.71e-02	1.29e+00	1.00e-05	1.45e+00	1.00e+00
8	+9.40e+04	3.68e+00	1.70e-01	5.17e-01	1.00e-05	6.01e-01	1.00e+00
9	+1.09e+05	3.45e+00	1.58e-02	4.72e-01	1.00e-06	1.68e+00	1.00e+00
10	+1.63e+05	3.24e+00	1.22e-01	1.55e-01	1.00e-06	1.65e+00	1.00e+00
11	+2.27e+05	3.09e+00	3.92e-02	3.41e-02	1.00e-06	3.68e-01	1.00e+00
12	+2.27e+05	3.07e+00	3.87e-04	4.32e-02	1.00e-07	7.74e-01	1.00e+00
13	+2.66e+05	3.04e+00	6.76e-03	6.79e-03	1.00e-07	2.24e-01	1.00e+00
14	+2.66e+05	3.04e+00	1.19e-05	1.20e-05	1.44e-10	1.30e-01	1.00e+00
15	+2.66e+05	3.04e+00	7.82e-09	6.02e-09	--------	--------	--------

(Same problem as previous slide, but with an additional bound on a variable.)

Outline

Penalty-Barrier

- "But I want to solve large problems! The size that I normally solve with interior-point methods."
- We think you should be able to!
- We should be cautious, though. This has been tried before....

Previous/related work

- Modified barrier methods (Jittorntrum and Osborne, 1980)
- Exterior-point methods (Polyak, 2008)
- Interior-point penalty methods (Chen and Goldfarb, 2006)

Quote (Conn, Gould, and Toint, 2000): These types of methods "suffer from a high sensitivity of their performance to the choice of parameters," which, they go on to state, "restricts their use as general-purpose methods."

Previous/related work

- Modified barrier methods (Jittorntrum and Osborne, 1980)
- Exterior-point methods (Polyak, 2008)
- Interior-point penalty methods (Chen and Goldfarb, 2006)

Quote (Conn, Gould, and Toint, 2000): These types of methods "suffer from a high sensitivity of their performance to the choice of parameters," which, they go on to state, "restricts their use as general-purpose methods."

We attack this issue head-on by focusing closely on updates for the parameters

Penalty methods

- Take the original nonlinear program:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & c(x) \leq 0 \end{aligned}$$

- Replace it with

$$\begin{aligned} \min\ & \rho f(x) + \sum s^j \\ \text{s.t. } & c(x) \leq s,\ s \geq 0 \end{aligned}$$

- Regularizes the constraints nicely
- Solution of LPs/QPs expensive for large problems

Barrier methods

- Take the original nonlinear program:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & c(x) \leq 0 \end{aligned}$$

- Replace it with

$$\begin{aligned} \min\ & f(x) - \mu \sum \ln r^j \\ \text{s.t. } & c(x) + r = 0,\ r \geq 0 \end{aligned}$$

- Step calculations as linear systems is good for large problems
- Not sufficiently regularized for ill-conditioned problems

Penalty-barrier methods

- Take the original nonlinear program:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & c(x) \leq 0 \end{aligned}$$

- Replace it with

$$\begin{aligned} \min\ & \rho f(x) - \mu \sum (\ln r^j + \ln s^j) + \sum s^j \\ \text{s.t. } & c(x) + r - s = 0,\ \ r \geq 0, s \geq 0 \end{aligned}$$

- Step calculations as linear systems is good for large problems
- Regularized for ill-conditioned problems

Penalty-barrier methods

- Take the original nonlinear program:

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } c(x) \leq 0 \end{aligned}$$

- Replace it with

$$\begin{aligned} &\min\ \rho f(x) - \mu \sum (\ln r^j + \ln s^j) + \sum s^j \\ &\text{s.t. } c(x) + r - s = 0,\ \ r \geq 0, s \geq 0 \end{aligned}$$

- Step calculations as linear systems is good for large problems
- Regularized for ill-conditioned problems

Search direction calculation

- Applying Newton's method to the optimality conditions:

$$\begin{bmatrix} H_k & 0 & 0 & \nabla c(x_k) \\ 0 & \Omega_k & 0 & I \\ 0 & 0 & \Gamma_k & -I \\ \nabla c(x_k)^T & I & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta r_k \\ \Delta s_k \\ \Delta \lambda_k \end{bmatrix} = - \begin{bmatrix} \rho \nabla f(x_k) + \nabla c(x_k) \lambda_k \\ \lambda_k - \mu R_k^{-1} e \\ e - \lambda_k - \mu S_k^{-1} e \\ c(x_k) + r_k - s_k \end{bmatrix}$$

where

$$\Omega_k := R_k^{-1} \Lambda_k \qquad \Gamma_k := S_k^{-1} (I - \Lambda_k)$$

Search direction calculation

- Applying Newton's method to the optimality conditions:

$$\begin{bmatrix} H_k & 0 & 0 & \nabla c(x_k) \\ 0 & \Omega_k & 0 & I \\ 0 & 0 & \Gamma_k & -I \\ \nabla c(x_k)^T & I & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta r_k \\ \Delta s_k \\ \Delta \lambda_k \end{bmatrix} = - \begin{bmatrix} \rho \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ \lambda_k - \mu R_k^{-1} e \\ e - \lambda_k - \mu S_k^{-1} e \\ c(x_k) + r_k - s_k \end{bmatrix}$$

where

$$\Omega_k := R_k^{-1}\Lambda_k \qquad \Gamma_k := S_k^{-1}(I - \Lambda_k)$$

- The right-hand side splits up nicely:

$$\begin{bmatrix} \rho \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ \lambda_k - \mu R_k^{-1} e \\ e - \lambda_k - \mu S_k^{-1} e \\ c(x_k) + r_k - s_k \end{bmatrix} = \rho \begin{bmatrix} \nabla f(x_k) \\ 0 \\ 0 \\ 0 \end{bmatrix} + \mu \begin{bmatrix} 0 \\ -R_k^{-1} e \\ -S_k^{-1} e \\ 0 \end{bmatrix} + \begin{bmatrix} \nabla c(x_k)\lambda_k \\ \lambda_k \\ e - \lambda_k \\ c(x_k) + r_k - s_k \end{bmatrix}$$

Open issues

- How do we update ρ and μ?
- How do we judge progress?
- Do we need a third parameter?

Slack reset

- Recall the penalty-barrier subproblem

$$\begin{aligned} \min\ & \rho f(x) - \mu \sum (\ln r^j + \ln s^j) + \sum s^j \\ \text{s.t. } & c(x) + r - s = 0,\ \ r \geq 0, s \geq 0 \end{aligned}$$

- For a fixed x_k, the following problem can be solved uniquely for (r_k, s_k)

$$\begin{aligned} \min\ & -\mu \sum (\ln r^j + \ln s^j) + \sum s^j \\ \text{s.t. } & c(x_k) + r - s = 0,\ \ r \geq 0, s \geq 0 \end{aligned}$$

- We think of $r_k := r(x_k)$ and $s_k := s(x_k)$

Merit function

- We can show that if the slacks are reset, the computed search direction will be a descent direction for the penalty-barrier objective!
- Therefore, our merit function becomes

$$\phi(x;\rho,\mu) := \rho f(x) - \mu \sum (\ln r^j(x) + \ln s^j(x)) + \sum s^j(x)$$

Updating ρ and μ (conservatively)

- As a conservative approach, we can think about a fixed ρ
- Run a Fiacco-McCormick approach to update $\mu \longrightarrow 0$
- Once the penalty-barrier problem is solved, if point is feasible, we are done; otherwise, we can reduce ρ and start Fiacco-McCormick again
- (This is a good back-up strategy, but we want something more aggressive!)

Updating ρ and μ (the ρ part)

- We adapt the steering rules to penalty-barrier methods
- We need a fixed quantity that we can "hold on to"
- Note: for $\rho = 0$ and μ set to the values used to compute r_k and s_k, the search direction will be one of descent for $\phi(x; 0, \mu)$!
- Our conditions may appear messy:

$$\begin{aligned} \Delta\tilde{l}(\Delta x_k^{\rho,\mu}; 0, \mu, x_k) &\geq \epsilon_1 \Delta l(\Delta z_k^{0,\mu}; 0, \mu, z_k) \\ \Delta\tilde{q}(\Delta z_k^{\rho,\mu}; \rho, \mu, x_k) &\geq \epsilon_2 \Delta l(\Delta z_k^{0,\mu}; 0, \mu, z_k) \end{aligned}$$

but they have a similar meaning as before!

Updating ρ and μ (the ρ part)

- We adapt the steering rules to penalty-barrier methods
- We need a fixed quantity that we can "hold on to"
- Note: for $\rho = 0$ and μ set to the values used to compute r_k and s_k, the search direction will be one of descent for $\phi(x; 0, \mu)$!
- Our conditions may appear messy:

$$\begin{aligned} \Delta\tilde{l}(\Delta x_k^{\rho,\mu}; 0, \mu, x_k) &\geq \epsilon_1 \Delta l(\Delta z_k^{0,\mu}; 0, \mu, z_k) \\ \Delta\tilde{q}(\Delta z_k^{\rho,\mu}; \rho, \mu, x_k) &\geq \epsilon_2 \Delta l(\Delta z_k^{0,\mu}; 0, \mu, z_k) \end{aligned}$$

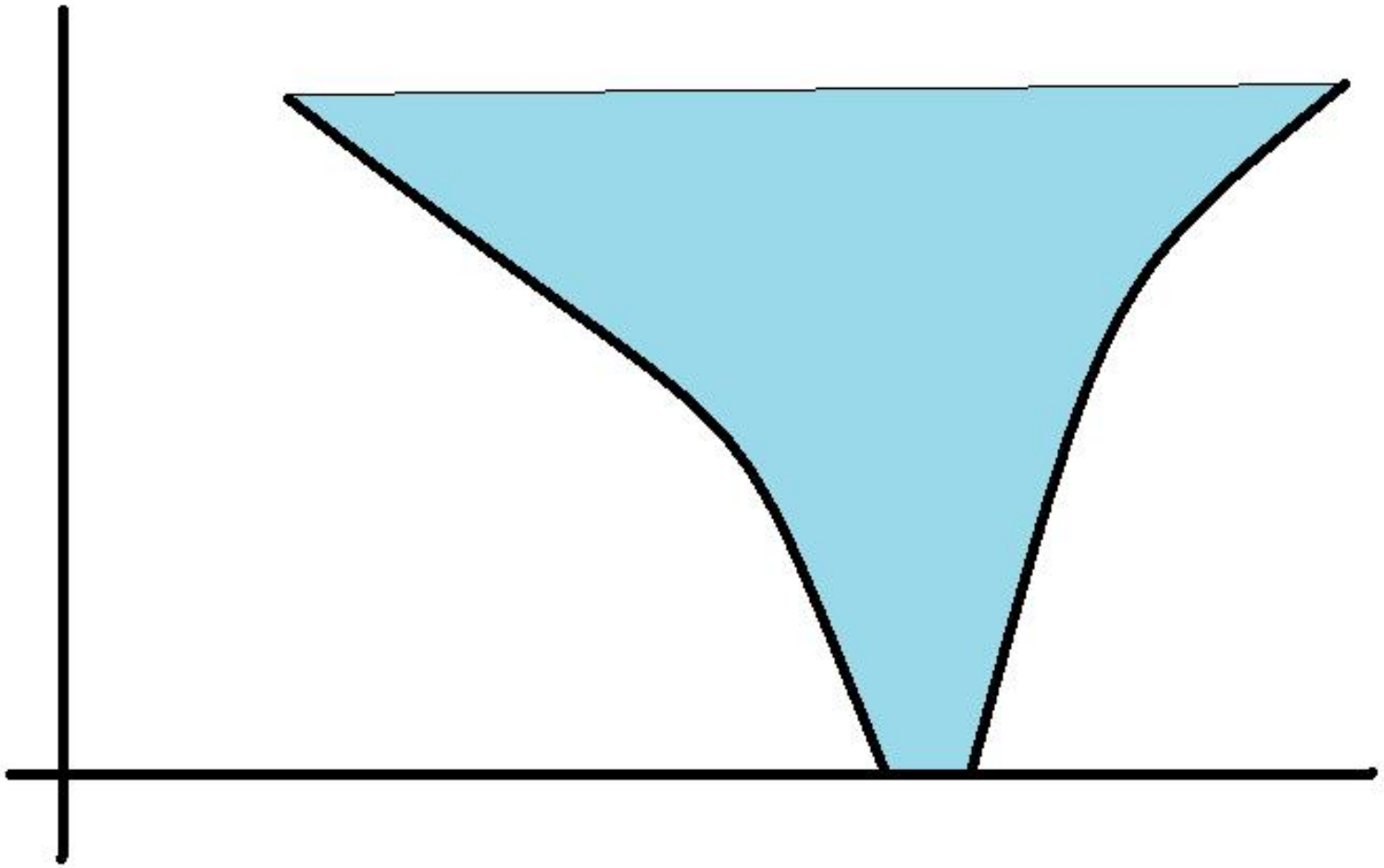
but they have a similar meaning as before!

Updating ρ and μ (the μ part)

- We adapt the adaptive barrier rules (Nocedal, Wächter, and Waltz, 2009)
- Choose ρ as large as possible so that the steering rules hold for some μ
- Then, choose μ so as to minimize a model of the KKT error:

$$m(\Delta z, \Delta\lambda; \rho, \mu, z) := \left\| \begin{bmatrix} \rho\nabla f(x) + \nabla c(x)(\lambda + \Delta\lambda) \\ (R + \Delta R)(\lambda + \Delta\lambda) \\ (S + \Delta S)(e - \lambda - \Delta\lambda) \end{bmatrix} \right\|^2$$

Typical allowable values of (μ, ρ)



Numerical results

- Preliminary results in Matlab are promising
- For example, for a small complementarity problem:

$$\begin{aligned} &\min \ \tfrac{1}{2}(x_1 - 1)^2 + (x_2 - 1)^2 \\ &\text{s.t. } x_1 \geq 0, \ x_2 \geq 0, \ x_1 x_2 \leq 0 \quad (\text{i.e., } x_1 \perp x_2) \end{aligned}$$

k	f	v	KKT
0	+1.50e+00	0.00e+00	2.62e+00
1	+1.50e+00	0.00e+00	8.01e-01
2	+1.50e+00	0.00e+00	2.70e-01
3	+1.50e+00	0.00e+00	2.65e-01
4	+1.50e+00	0.00e+00	2.65e-01
5	+1.50e+00	1.60e-11	2.64e-01
6	+1.41e+00	3.13e-10	2.43e-01
7	+1.18e+00	8.49e-11	1.77e-01
8	+9.50e-01	1.08e-10	4.35e-01
9	+5.00e-01	2.31e-07	3.41e-02
10	+5.00e-01	2.31e-07	7.37e-07

Numerical results

- Implementation in `KNITRO` coming soon ;-)

Outline

Conclusion

- We have discussed the use of penalty functions to regularize algorithms for nonlinear constrained optimization
- In the context of SQP, penalties effectively regularize the constraints, and we have shown that they can allow for rapid infeasibility detection (convergence is superlinear for feasible *and* infeasible instances)
- In the context of interior-point methods, we are developing techniques so that the usual disadvantages of penalty-barrier methods are not a problem for us
- In the end, we hope to have an approach that solves large (feasible and infeasible) problems quickly

Thanks!!