Nonlinear Optimization

CMPT 419/983
Mo Chen
SFU Computing Science
25/09/2019

Outline

- Introduction to cvx
- Nonlinear optimization
- Sequential quadratic programming

References

- cvx user's guide: http://cvxr.com/cvx/doc/CVX.pdf
- S. Boyd and L. Vandenberghe, "Convex Optimization."
- M. Kochenderfer and T. A. Wheeler. "Algorithms for Optimization."
- Boggs, P. T., & Tolle, J. W. (1995). Sequential Quadratic Programming. *Acta Numerica*, *4*, 1. https://doi.org/10.1017/S0962492900002518

Introduction to cvx

- cvx: MATLAB software for disciplined convex programming
  - http://cvxr.com/cvx/download/
  - http://cvxr.com/cvx/doc/install.html

- User must make sure the program is convex

Coding example in cvx

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P x + q^\top x + r \\ \text{subject to} & -1 \le x \le 1 \end{array}$$

where $P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, q = \begin{bmatrix} -22 \\ -14.5 \\ 13 \end{bmatrix}, r = 1$

```
P = [13 12 -2; 12 17 6; -2 6 12];
q = [-22; -14.5; 13];
r = 1;
n = 3;
x_lower = -1;
x_upper = 1;

% Construct and solve the model
cvx_begin
    variable x(n)
    minimize ( (1/2)*quad_form(x,P) + q'*x + r )
    x >= x_lower;
    x <= x_upper;
cvx_end

fprintf('The computed optimal solution is (%.1f, %.1f, %.1f)\n', x(1), ...
  x(2), x(3))
```

```
Status: Solved
Optimal value (cvx_optval): -21.625

The computed optimal solution is (1.0, 0.5, -1.0)
```

Adapted from S. Boyd and L. Vandenberghe, “Convex Optimization,” *Optim. Methods Softw.*, vol. 25, no. 3, pp. 487–487, 2010.

Coding example in cvx

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P x + q^\top x + r \\ \text{subject to} & -1 \leq x \leq 1 \end{array}$$

$$\text{where} \quad P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, q = \begin{bmatrix} -22 \\ -14.5 \\ 13 \end{bmatrix}, r = 1$$

- What happens if

$$P = \begin{bmatrix} \mathbf{0} & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}?$$

Adapted from S. Boyd and L. Vandenberghe, "Convex Optimization," *Optim. Methods Softw.*, vol. 25, no. 3, pp. 487–487, 2010.

Coding example in cvx

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\top P x + q^\top x + r \\ \text{subject to} \quad & -1 \le x \le 1 \end{aligned}$$

$$\text{where} \quad P = \begin{bmatrix} \color{red}{\mathbf{0}} & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, q = \begin{bmatrix} -22 \\ -14.5 \\ 13 \end{bmatrix}, r = 1$$

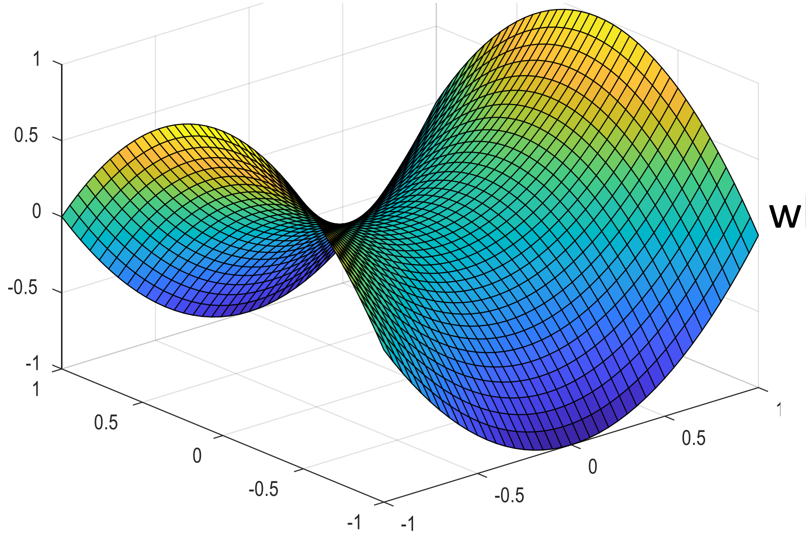
```
P = [0 12 -2; 12 17 6; -2 6 12];
q = [-22; -14.5; 13];
r = 1;
n = 3;
x_lower = -1;
x_upper = 1;

% Construct and solve the model
cvx_begin
    variable x(n)
    minimize ( (1/2)*quad_form(x,P) + q'*x + r )
    x >= x_lower;
    x <= x_upper;
cvx_end

fprintf('The computed optimal solution is (%.1f, %.1f, %.1f)\n', x(1), ...
  x(2), x(3))
```

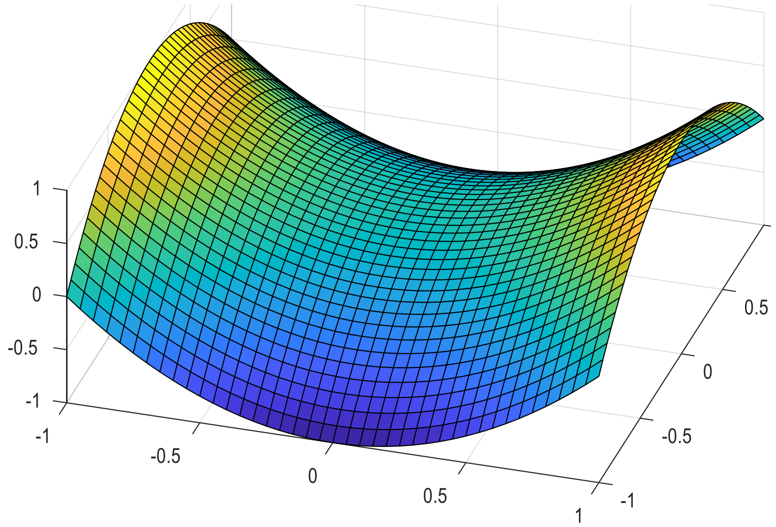
```
Error using cvx/quad_form (line 230)
The second argument must be positive or negative semidefinite.

Error in qp_cvx_example (line 19)
    minimize ( (1/2)*quad_form(x,P) + q'*x + r )
```

```
>> eig(P)

ans =

   -7.3059
   11.4985
   24.8074
```

Adapted from S. Boyd and L. Vandenberghe, "Convex Optimization," *Optim. Methods Softw.*, vol. 25, no. 3, pp. 487–487, 2010.

Coding example in cvx





$$\begin{array}{ll} \underset{x}{\text{minimize}} & x^\top P x + q^\top x + r \\ \text{subject to} & -1 \leq x \leq 1 \end{array}$$

where $P = \begin{bmatrix} \color{red}{\mathbf{0}} & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, q = \begin{bmatrix} -22 \\ -14.5 \\ 13 \end{bmatrix}, r = 1$

```
Error using cvx/quad_form (line 230)
The second argument must be positive or negative semidefinite.

Error in qp_cvx_example (line 19)
    minimize ( (1/2)*quad_form(x,P) + q'*x + r )
```

```
>> eig(P)

ans =

   -7.3059
   11.4985
   24.8074
```

Adapted from S. Boyd and L. Vandenberghe, “Convex Optimization,” *Optim. Methods Softw.*, vol. 25, no. 3, pp. 487–487, 2010.

Nonlinear Optimization Program

$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\
& h_j(x) = 0, j = 1, \dots, m
\end{aligned}
$$

- Easy cases:
  - Find global optimum for linear program: f, g_i, h_j are linear
  - Find global optimum for convex program: f, g_i are convex, h_j is linear
  - **Find local optimum for nonlinear program: $\boldsymbol{f, g_i, h_j}$ are differentiable**

Optimization in Robotic Decision Making

- Optimal control problem

$$\underset{u(\cdot)}{\text{minimize}}\ l\left(x\left(t_f\right), t_f\right) + \int_0^{t_f} c(x(t), u(t), t)dt$$

$$\text{subject to } \dot{x}(t) = f\left(x(t), u(t)\right)$$

- Observation: Discretize time → nonlinear optimization problem
  - Time discretization: $t = hk \coloneqq t^k$
  - Minimize over $\{u^k\}$, where $u^k \coloneqq u\left(t^k\right)$
  - State at time t^k is given by $x^k \coloneqq x\left(t^k\right)$

Nonlinear Optimization Program

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

- Strategy 1: write down the KKT conditions, and solve the resulting systems of equations

Karush-Kuhn-Tucker (KKT) Conditions

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

- Equations to solve: KKT conditions
  - Stationarity $\nabla_x L(x^*, \lambda^*, \mu^*) = 0$
  - Primal feasibility: $g_i(x^*) \leq 0,\; a_i^\top x^* - b_i = 0$
  - Dual feasibility: $\lambda^* \geq 0$
  - Complementary slackness: $\lambda_i^* g_i(x^*) = 0,\; i = 1, \dots, n$

- Use numerical equation solvers, or do it by hand (as much as possible)

- For convex problems, KKT conditions are necessary and sufficient
- For general nonlinear problems, KKT conditions are just necessary

Nonlinear Optimization Program

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

- Strategy 1: write down the KKT conditions, and solve the resulting systems of equations
- Strategy 2: Convert the nonlinear problem into a sequence of convex subproblems
  - Sequential convex programming
    - **Sequential quadratic programming**

Sequential Quadratic Programming

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, i = 1, \dots, n \\ & h_j(x) = 0, j = 1, \dots, m \end{aligned}$$

- Obtain a sequence x^k that converges to a local minimum
- Iteratively solve quadratic subproblems

- Objective: $\underset{d_x}{\text{minimize}} \ (\boldsymbol{r^k})^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$ where $d_x := x - x^k$,

  r^k, B_k
  - Depend on x^k
  - to be chosen

Sequential Quadratic Programming

$$\begin{aligned}\text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0 \\ & h(x) = 0\end{aligned}$$

- Obtain a sequence x^k that converges to a local minimum
- Iteratively solve quadratic subproblems
- Objective: $\underset{d_x}{\text{minimize}} \ (\boldsymbol{r^k})^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$ where $d_x := x - x^k$,

r^k, B_k

- Depend on x^k
- to be chosen

Sequential Quadratic Programming

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g(x) \leq 0 \\ & && h(x) = 0 \end{aligned}$$

- Objective: $\underset{d_x}{\text{minimize}}\ \left(\boldsymbol{r^k}\right)^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$ where $d_x := x - x^k$,
- Constraints: subject to $\nabla g\left(x^k\right)^\top d_x + g\left(x^k\right) \leq 0$

  $\nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0$

r^k, B_k

- Depend on x^k
- to be chosen

General SQP Algorithm

Given: x^0, B_0, merit function $\phi, k = 0$

1. Solve quadratic subproblem to obtain d_x
2. Choose step length α by comparing $\phi\left(x^k + \alpha d_x\right)$ and $\phi\left(x^k\right)$
3. Set $x^{k+1} = x^k + \alpha d_x$
4. Stop if converged
5. Compute B_{k+1}, r^{k+1} increment k, go to step 1

$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & g(x) \leq 0 \\
& h(x) = 0
\end{aligned}
$$

$$
\begin{aligned}
\underset{d_x}{\text{minimize}} \quad & \left(\boldsymbol{r^k}\right)^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x \\
\text{subject to} \quad & \nabla g\left(x^k\right)^\top d_x + g\left(x^k\right) \leq 0 \\
& \nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0
\end{aligned}
$$

where $d_x \coloneqq x - x^k$,

r^k, B_k

- Depend on x^k
- to be chosen

Degrees of freedom:

- r^k
- B_k
- ϕ
- α
- x^0

The Quadratic Subproblem: Naïve Version

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad g(x) \leq 0 \\ &\qquad\qquad\quad\;\; h(x) = 0 \end{aligned}$$

$\longrightarrow$

$$\underset{d_x}{\text{minimize}} \; \left(\boldsymbol{r^k}\right)^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$$

$$\text{subject to} \quad \nabla g\left(x^k\right)^\top d_x + g\left(x^k\right) \leq 0$$
$$\nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0$$

- The obvious choice: quadratize $f(x)$

where $d_x \coloneqq x - x^k$,

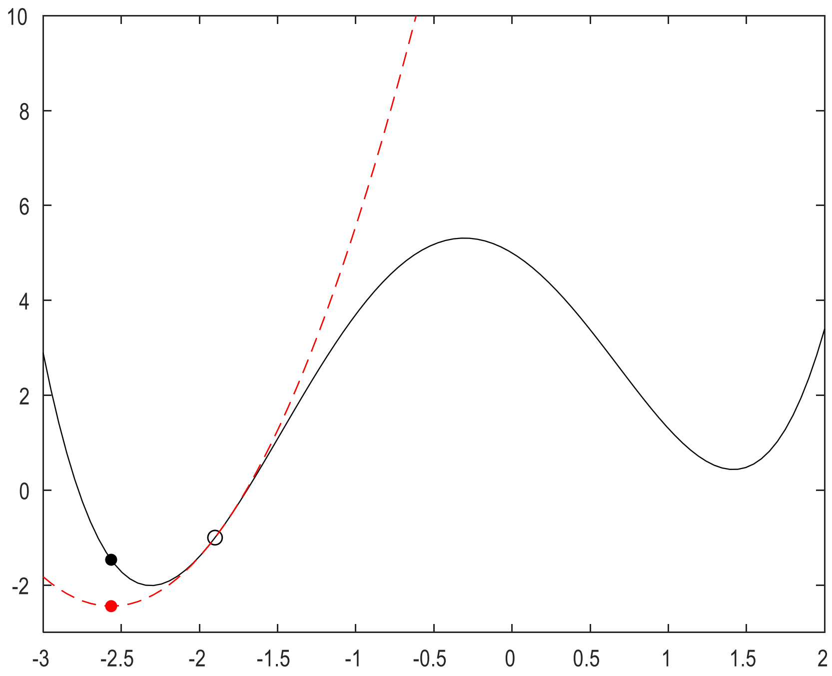
r^k, B_k
- Depend on x^k
- to be chosen

$r^k = \nabla f(x_k)$

$B_k = \mathrm{H} f(x_k)$

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n}\right)$$

$$\mathrm{H} f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$



The Quadratic Subproblem: Naïve Version

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad g(x) \le 0 \\ &\qquad\qquad\quad\; h(x) = 0 \end{aligned}$$

$$\underset{d_x}{\text{minimize}} \; \left(\boldsymbol{r^k}\right)^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$$

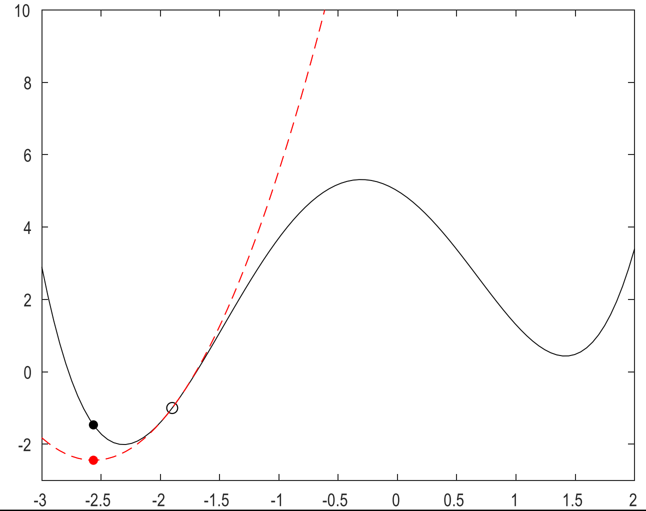
$$\text{subject to} \quad \nabla g\left(x^k\right)^\top d_x + g\left(x^k\right) \le 0$$
$$\nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0$$

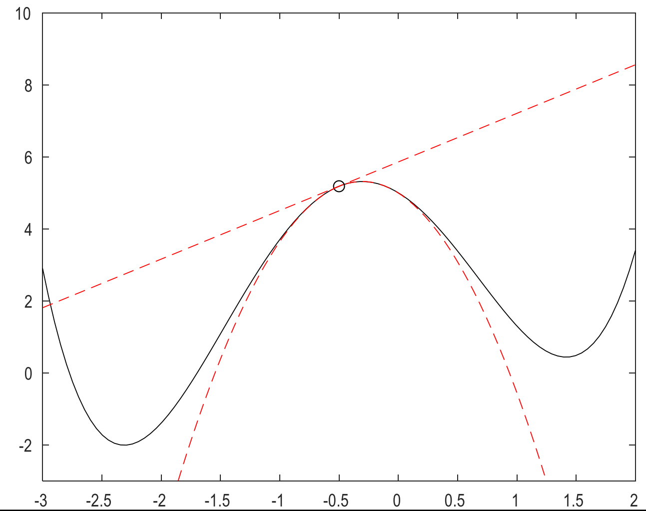
where $d_x \coloneqq x - x^k$,

- The obvious choice: quadratize $f(x)$

$$r^k = \nabla f(x_k)$$
$$B_k = \mathrm{H}f(x_k)$$

  - Convexify if needed, eg. by removing negative eigenvalues





Example

minimize $0.5x^4 + 0.8x^3 - 3x^2 - 2x + 5$
subject to $-3 \leq x \leq 2$

- Preliminaries (objective function)
  - $f(x) = 0.5x^4 + 0.8x^3 - 3x^2 - 2x + 5$
  - $\nabla f(x) = 4\times 0.5x^3 + 3\times 0.8x^2 - 2\times 3x - 2$
  - $\nabla^2 f(x) = 3\times 4\times 0.5x^2 + 2\times 3\times 0.8x - 6$

```
f = @(x) 0.5*x.^4 + 0.8*x.^3 - 3*x.^2 - 2*x + 5;
grad_f = @(x) 4*0.5*x.^3 + 3*0.8*x.^2 - 2*3*x - 2;
hess_f = @(x) 3*4*0.5*x.^2 + 2*3*0.8*x - 2*3;
```

- Preliminaries (constraints)
  - $g(x) = \begin{bmatrix} x-2 \\ -3-x \end{bmatrix}, \nabla g(x) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

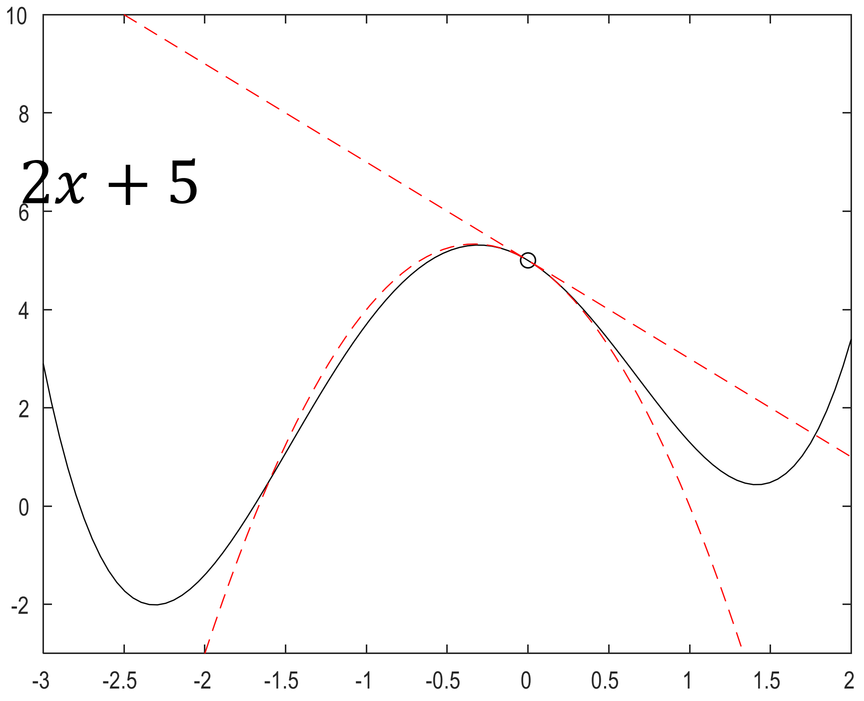
```
g =      {@(x) -x-3;    @(x) x-2};
grad_g = {@(x) -1;      @(x) 1};
```

- Preliminaries (initial approximation)
  - $x_0 = 0$

```
x_0 = 0;
x_k = x_0;
```

Example

$$\text{minimize } 0.5x^4 + 0.8x^3 - 3x^2 - 2x + 5$$
$$\text{subject to } -3 \leq x \leq 2$$



- Quadratize objective

$$\underset{d_x}{\text{minimize}} \ \left(r^k\right)^\top d_x + \frac{1}{2} d_x^\top B_k d_x$$

  - $r^k = \nabla f(x_k)$
  - $B_k = \mathrm{H}f(x_k)$
    - But make sure to convexify!

```
small = 0;
BK = hess_f(x_k);

[V, D] = eig(BK);
D(D<small) = small;
BK = V*D*V^-1;
```

```
q_approx = @(x) f(x_k) + grad_f(x_k)*(x-x_k) + 0.5*BK*(x-x_k).^2;
```

- Solve the quadratic subproblem

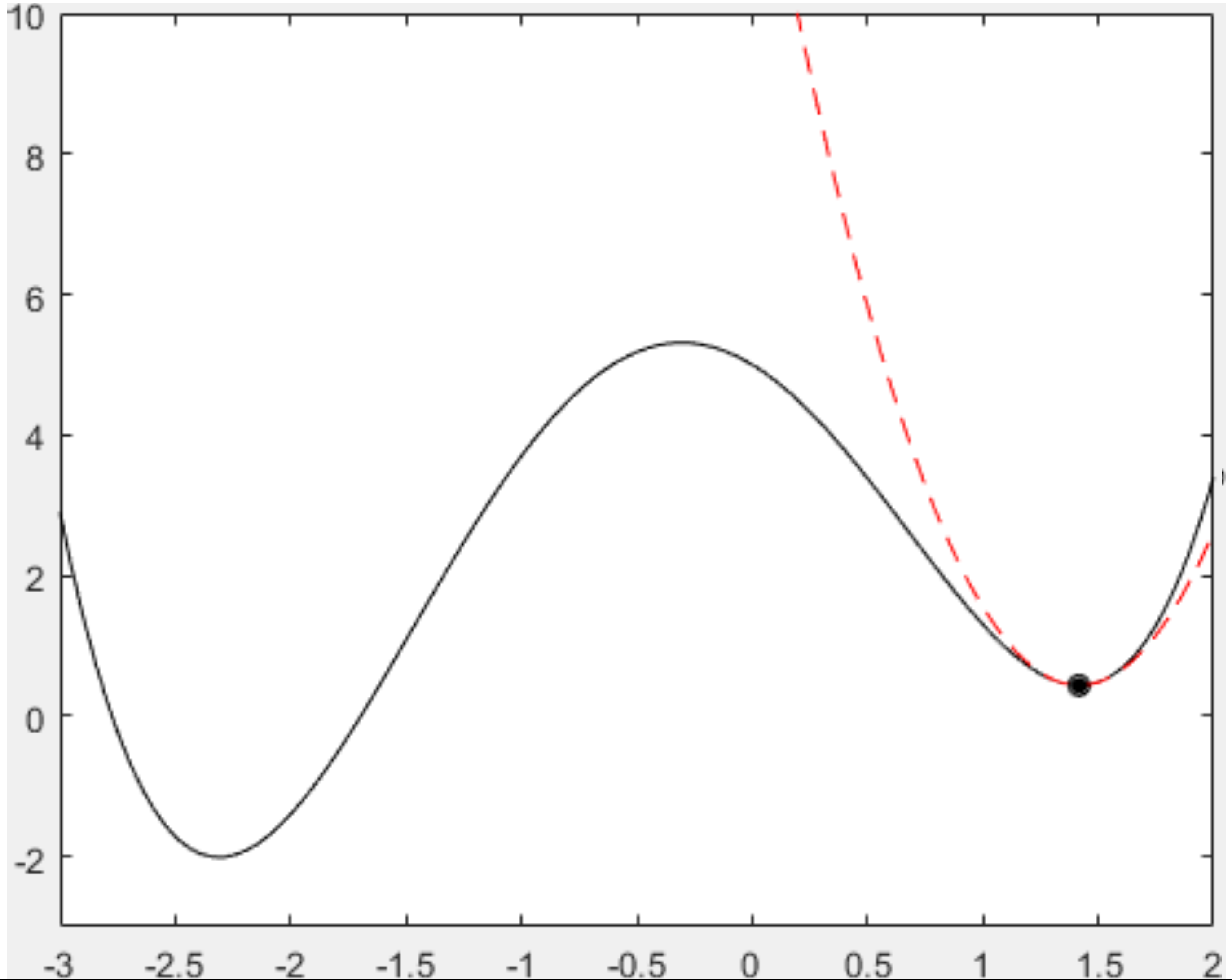
```
% Solve quadratic subproblem
d_k = qp_general(BK, grad_f(x_k), 0, h_qp, gradh_qp, g_qp, gradg_qp, n);
```

- Update x_k

```
% Update iterate
x_k = x_k + alpha * d_k;
```

Example

minimize $0.5x^4 + 0.8x^3 - 3x^2 - 2x + 5$
subject to $-3 \le x \le 2$



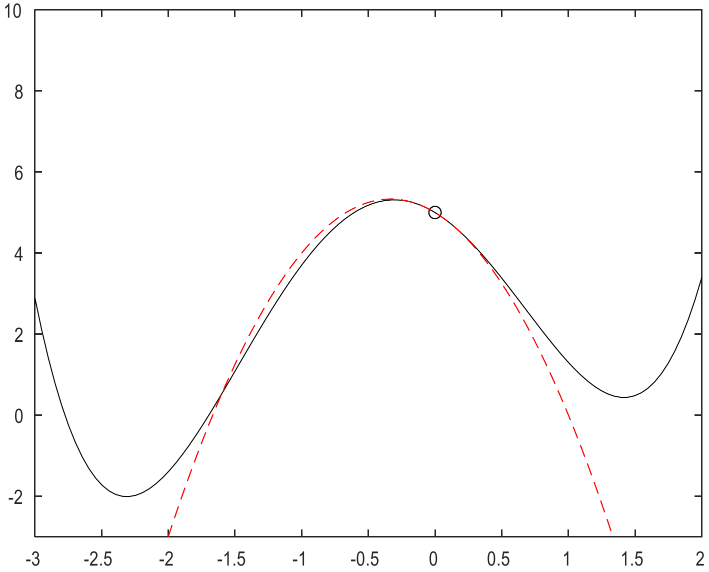
Example

$$\text{minimize } 0.5x^4 + 0.8x^3 - 3x^2 - 2x + 5$$
$$\text{subject to } -3 \le x \le 2$$

- Quadratize objective

$$\underset{d_x}{\text{minimize}}\ \left(r^k\right)^\top d_x + \frac{1}{2} d_x^\top B_k d_x$$

  - $r^k = \nabla f(x_k)$
  - $B_k = \mathrm{H}f(x_k) = \nabla^2 f(x_k)$
    - What if we didn't convexify?

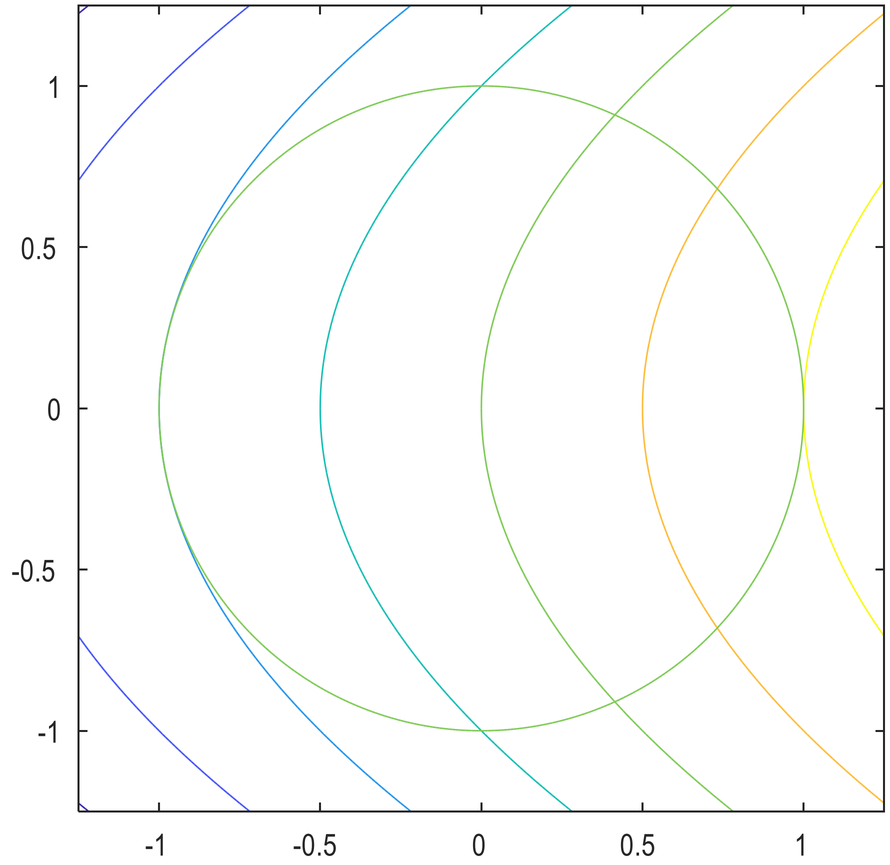


```
small = 0;
BK = hess_f(x_k);

[V, D] = eig(BK);
D(D<small) = small;
BK = V*D*V^-1;
```

```
q_approx = @(x) f(x_k) + grad_f(x_k)*(x-x_k) + 0.5*BK*(x-x_k).^2;
```

```
Error using cvxprob/newobj (line 57)
Disciplined convex programming error:
   Cannot minimize a(n) concave expression.

Error in minimize (line 21)
    newobj( prob, 'minimize', x );

Error in qp_general (line 20)
  minimize ( (1/2)*quad_form(x,P) + q'*x + r) % objective

Error in sqp_convexify (line 113)
  d_k = qp_general(BK, grad_f(x_k), 0, h_qp, gradh_qp, g_qp, gradg_qp, n);
```

Naïve Quadratization Issues



- Nonlinear constraints

$$\begin{aligned}\underset{x}{\text{minimize}}\ & x_1 - \frac{1}{2}x_2^2 \\ \text{subject to}\ & x_1^2 + x_2^2 - 1 = 0\end{aligned}$$

  - Optimal solution: $(-1,0)$
  - At point $\left(x_1^k, x_2^k\right) = (-(1+\epsilon), \epsilon)$

$$\begin{aligned}\underset{d_x}{\text{minimize}}\ & d_{x,1} - \frac{1}{2}d_{x,2}^2 \\ \text{subject to}\ & d_{x,2} = \frac{1+\epsilon}{\epsilon}d_{x,1} - \epsilon - 1\end{aligned}$$

  - Numerically unstable due to ϵ in denominator

$$\begin{bmatrix}2x_1^k & 2x_2^k\end{bmatrix}\begin{bmatrix}d_{x,1}\\ d_{x,2}\end{bmatrix} = -2(1+\epsilon)d_{x,1} + 2\epsilon d_{x,2}$$

$$\nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0$$

$$(1+\epsilon)^2 + \epsilon^2 - 1 = 2\epsilon^2 + 2\epsilon$$

Quadratize Lagrangian

$$\text{minimize} \quad f(x)$$
$$\text{subject to} \quad g(x) \leq 0$$
$$h(x) = 0$$

$\longrightarrow$

$$\underset{d_x}{\text{minimize}} \ \left(\boldsymbol{r^k}\right)^\top d_x + \frac{1}{2} d_x^\top \boldsymbol{B_k} d_x$$

subject to $\nabla g\left(x^k\right)^\top d_x + g\left(x^k\right) \leq 0$

$\nabla h\left(x^k\right)^\top d_x + h\left(x^k\right) = 0$

where $d_x \coloneqq x - x^k$

- Alternative objective:

$$r^k = \nabla f(x_k)$$
$$B^k = \mathrm{H}L(x_k, u_k, v_k), \qquad L(x, u, v) = f(x) + u^\top g(x) + v^\top h(x)$$

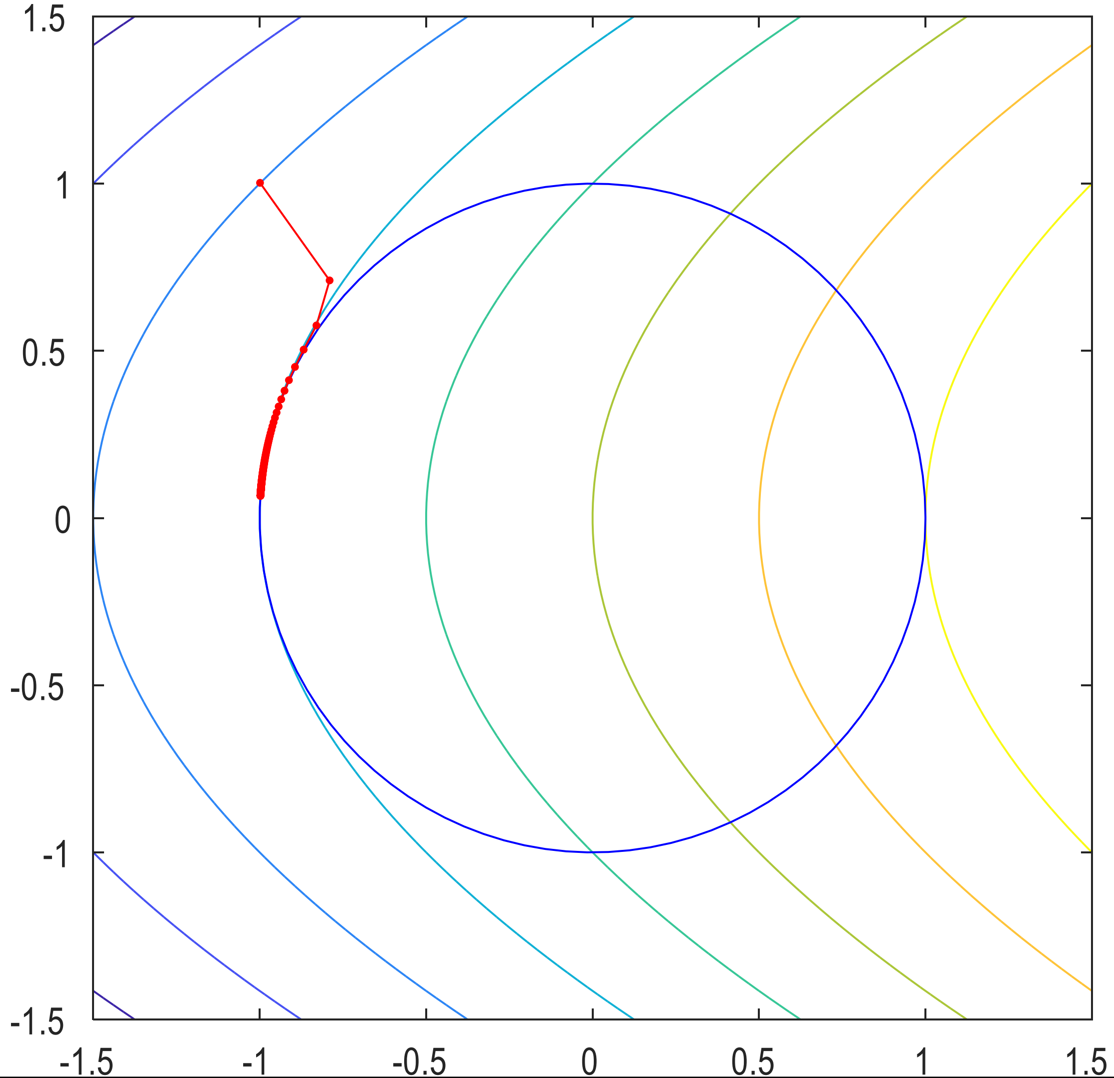
  - Captures nonlinearly in constraints
  - Local minima x^* of original nonlinear program is a local minima of

$$\underset{x}{\text{minimize}} \ L(x, u^*, v^*)$$

subject to $h(x) = 0$

$g(x) \leq 0$

Quadratize Lagrangian



Sequential Quadratic Programming

- Many other variants
  - Boggs, P. T., & Tolle, J. W. (1995). Sequential Quadratic Programming. *Acta Numerica*, *4*, 1. https://doi.org/10.1017/S0962492900002518

- Only a local method
  - Faster convergence than gradient methods

- Not discussed today: convergence analysis

- Requires initialization

Tricks

- Sparsity in constraints
- Good initializations
- Slackening constraints
- Step sizes
- Merit functions
- …
- Use software packages
    - SNOPT, KNITRO, TOMLAB, IPOPT, …