Lecture 8

Process Flowsheet Optimization

Part I: Motivation and Classification of Optimization Problems

Part II: Unconstrained Optimization

Part III: Constrained Nonlinear Optimization

Part IV: Solution Strategies: The KKT Conditions and SQP

Prepared jointly with Kyle Camarda, Associate Professor, University of Kansas

Introduction

People optimize. Investors seek to create portfolios that avoid excessive risk while achieving a high rate of return. Manufacturers aim for maximum efficiency in the design and operation of their production processes. Engineers adjust parameters to optimize the performance of their designs.

Nature optimizes. Physical systems tend to a state of minimum energy. The molecules in an isolated chemical system react with each other until the total potential energy of their electrons is minimized. Rays of light follow paths that minimize their travel time.

J. Nocedal, S. J. Wright, Numerical Optimization, 2006

Applications of Optimization in Chem E.

- Long-term planning and scheduling
  - facility location and sizing
  - transportation problems
- Process modeling
  - fitting data to a model
  - model selection for derivation of optimal operating conditions
- Process design
  - waste minimization
  - heat exchanger/reactor network synthesis
  - layout/piping
  - solvent selection, equipment selection

Goals and Challenges

- Goals:
  - Maximize profits
  - Minimize waste/environmental impact
  - Minimize energy usage
  - Minimize raw material usage

- Challenges:
  - Uncertainty in model parameters
  - Dynamic processes
  - Large problem sizes

Classification of Optimization Problems

- Assuming we have a model for our system which can be written mathematically, an optimization problem can be formulated which looks like:

$$\begin{array}{lll} \text{max/min } f(\bar{x}) & & \text{Objective function} \\ \text{subject to: } & \bar{g}(\bar{x}) \le 0 & \text{inequality constraints} \\ & \bar{h}(\bar{x}) = 0 & \text{equality constraints} \end{array}$$

- The functions $f, \bar{g}, \bar{h}$ can be either linear or nonlinear, and the variables $\bar{x}$ can be either continuous or <u>discrete</u>

Discrete Variables

We're used to dealing with continuous variables (temperatures, flowrates), but many decisions in chemical engineering are inherently discrete:

- How many batches of product *x* should be produced? (integer)
- Should we build a tank farm in Malaysia or not? (binary)
- How many reactors in parallel should be used for a given application? (integer)
- Which product should our batch plant produce on January 23rd? (binary)

Types of Optimization Problems

Based on these ideas, we categorize optimization problems in the following manner:

- Unconstrained, linear or nonlinear, all continuous (Use calculus to solve)
- Constrained, linear, all continuous: called a Linear Program or LP
- Constrained, nonlinear, all continuous: called a Nonlinear Program or NLP
- Constrained, linear, some integer/binary: called a Mixed-Integer Linear Program (MILP or MIP)
- Constrained, nonlinear, some integer/binary: called a Mixed-Integer Nonlinear Program or MINLP

Feasible Regions

Let's look at a 2-D linear program graphically:

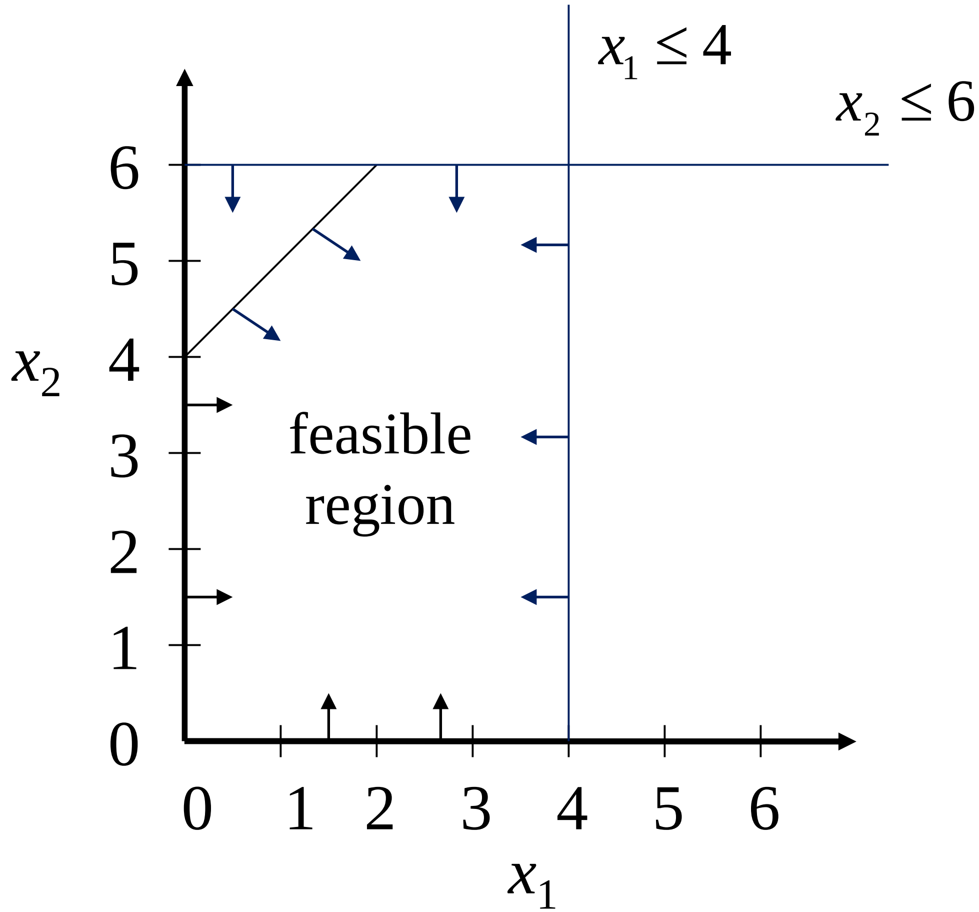
$$\max\ f(\bar{x}) = x_1 + x_2$$

$$\text{subject to:}\quad x_2 - x_1 \le 4$$

$$x_2 \le 6$$

$$x_1 \le 4$$

$$x_1, x_2 \ge 0$$



Any solution which lies within the feasible region is called a ***feasible solution***

Feasible Regions, continued

• Note the feasible region can also be called the <u>simplex</u>

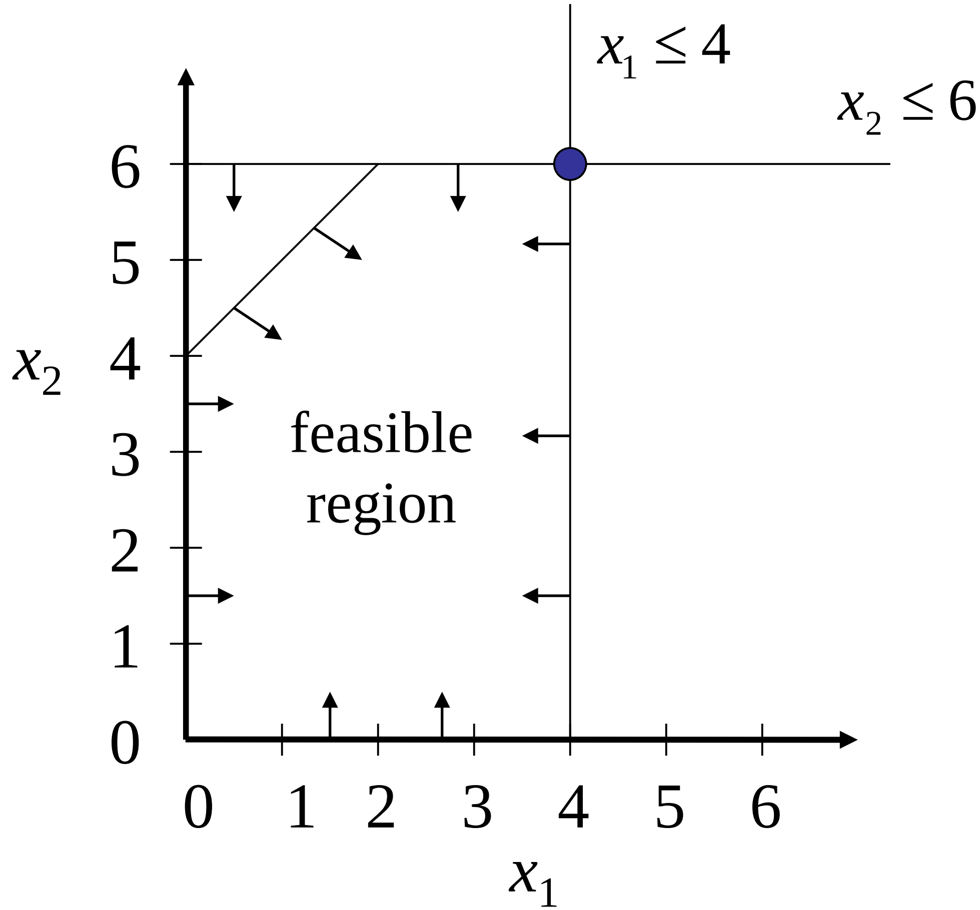
• The <u>optimal solution</u> lies at x_1=4, x_2=6

• At this point, the two constraints

$x_2 \le 6 \quad x_1 \le 4$

are called <u>active</u>

• The other constraints are inactive (oversatisfied)



Feasible Regions, continued

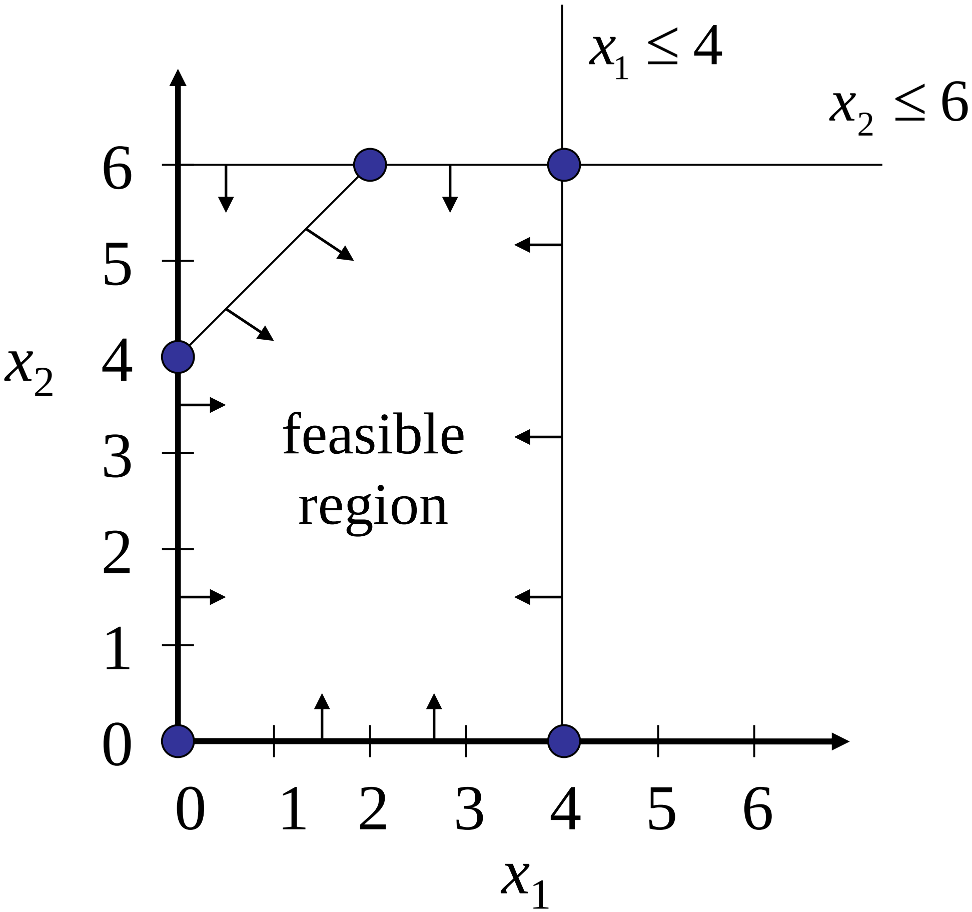
- How many feasible solutions are there?

Infinitely many!

- We can handle this by the realization that the optimal solution for <u>any</u> linear objective function must lie at an <u>extreme point</u>

- Example: max x_1-x_2

  Solution is at (4,0)

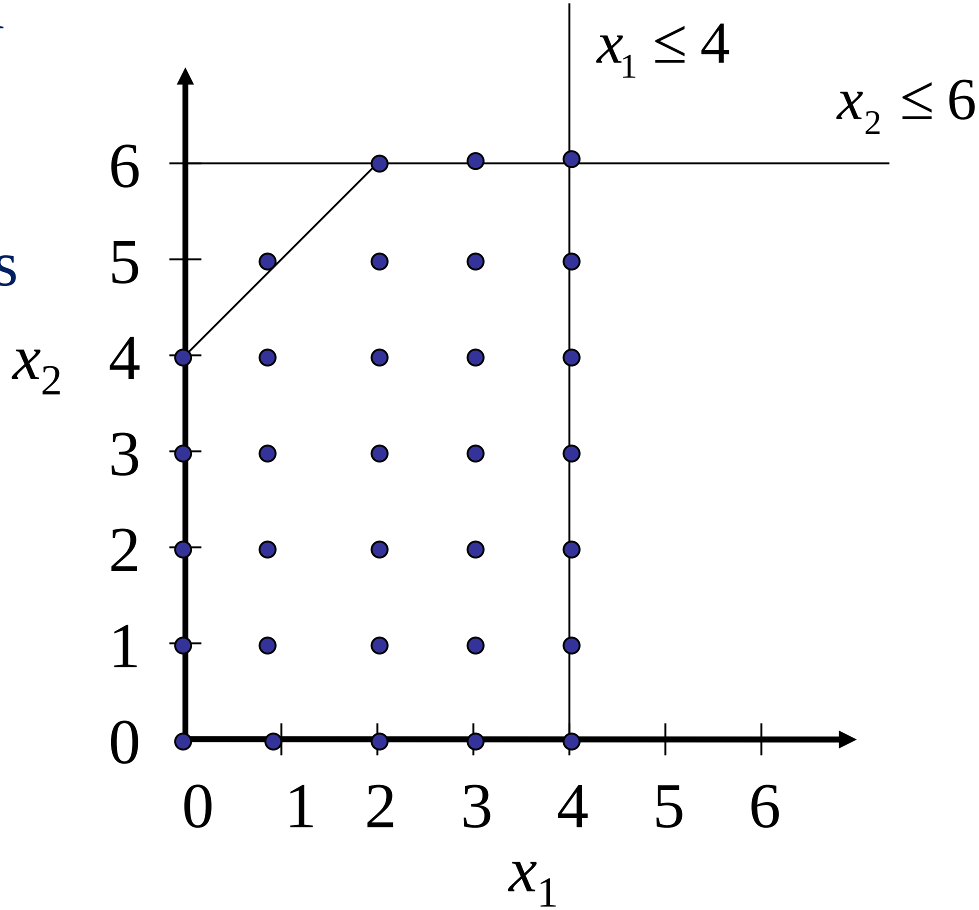


Feasible Regions, continued

- Now, what happens if x_1 and x_2 are forced to be integers?

- The feasible region becomes a set of discrete points

- If all the equations in the integer problem are linear (MILP), we still have one solution, but it may not lie at an extreme point of the simplex

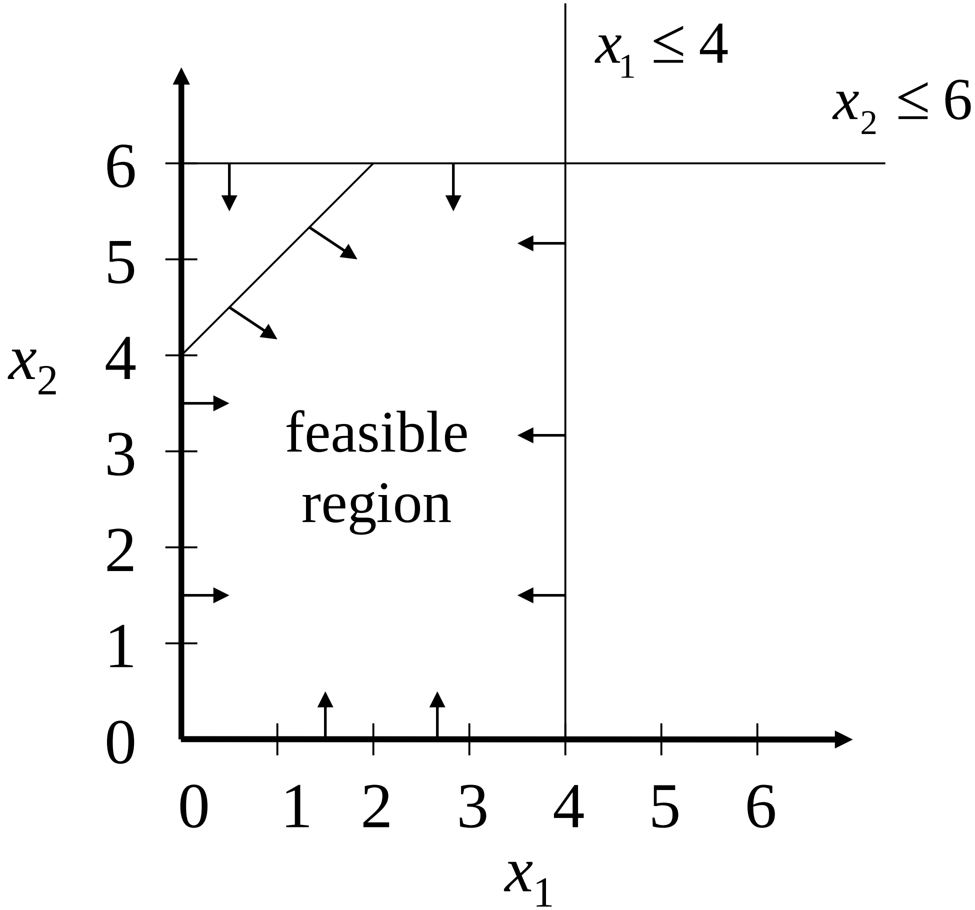


Feasible Regions, continued

Back to the LP, one final point:

$$\max \; f(\bar{x}) = x_1 + x_2$$

$$\text{subject to}: \quad x_2 - x_1 \le 4$$
$$x_2 \le 6$$
$$x_1 \le 4$$
$$x_1, x_2 \ge 0$$



Note we have 6 equations, 2 unknowns. How many optimal solutions are there?

1 in this case, could be 0

This means degrees of freedom analysis ***cannot*** be used directly here

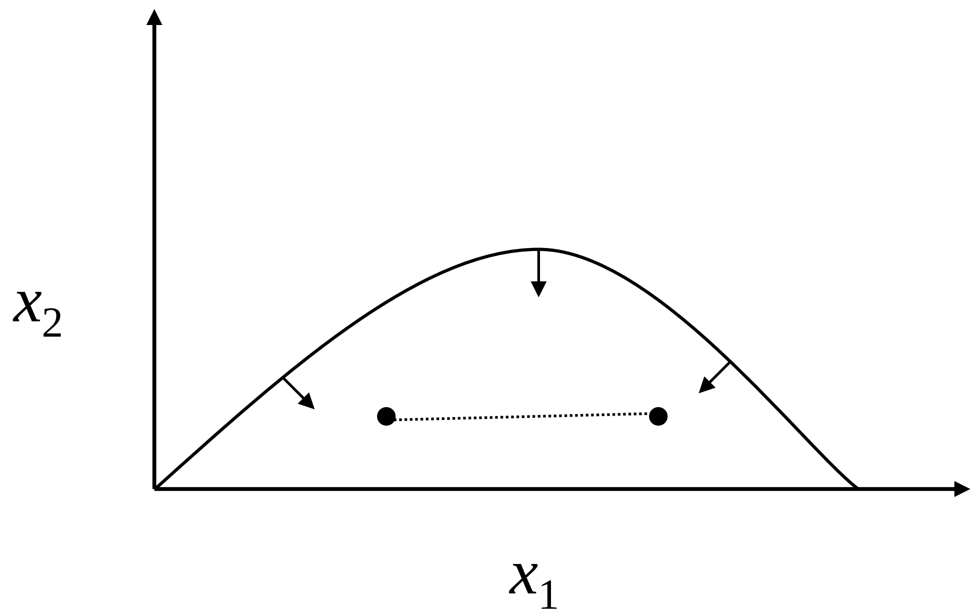
Convex Functions

- Let’s start by looking at one constraint of an NLP:

$$x_2 \le -ax_1^2 + bx_1 \qquad x_1, x_2 \ge 0$$

- Rearrange to standard form:

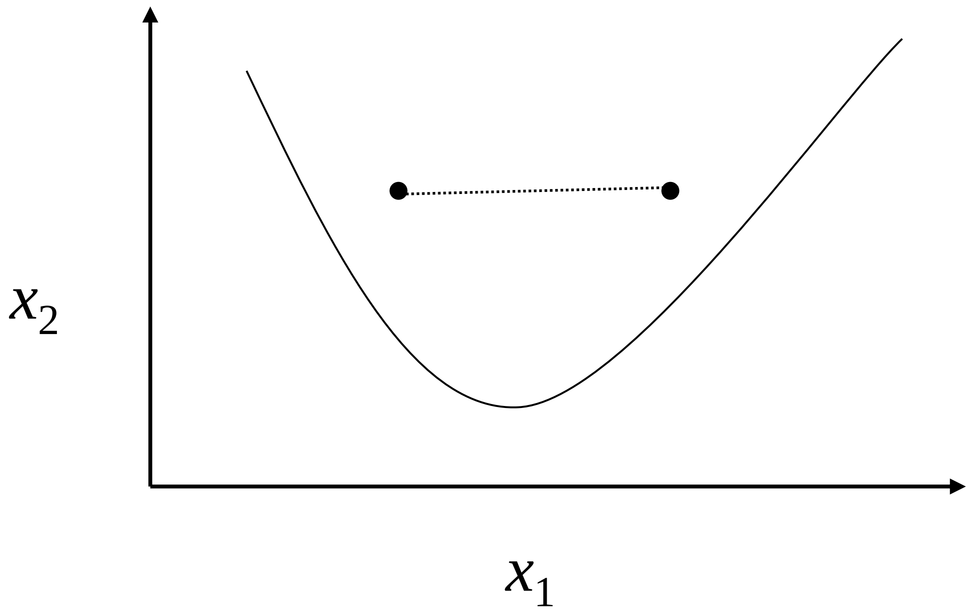
$$g(x) = ax_1^2 - bx_1 + x_2 \le 0$$



- $g(x)$ is a <u>concave function</u>, since a line drawn between any two points on or below the curve stays on or below the curve

Convex Functions, continued

- If the function were $g(x) = ax_1^2 - bx_1 + x_2 + C \leq 0$

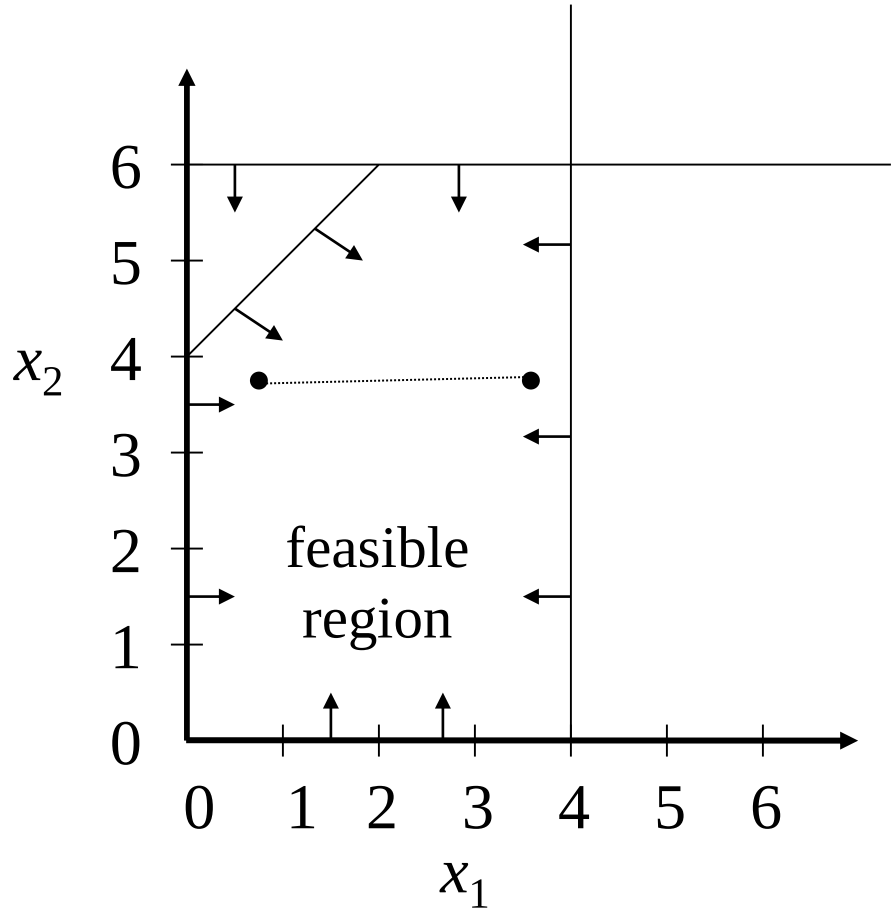


$$g(\alpha x_1 + (1-\alpha)x_2) \leq \alpha g(x_1) + (1-\alpha)g(x_2)$$

- $g(x)$ is now a convex function, since a line drawn between any two points on or above the curve stays on or above the curve

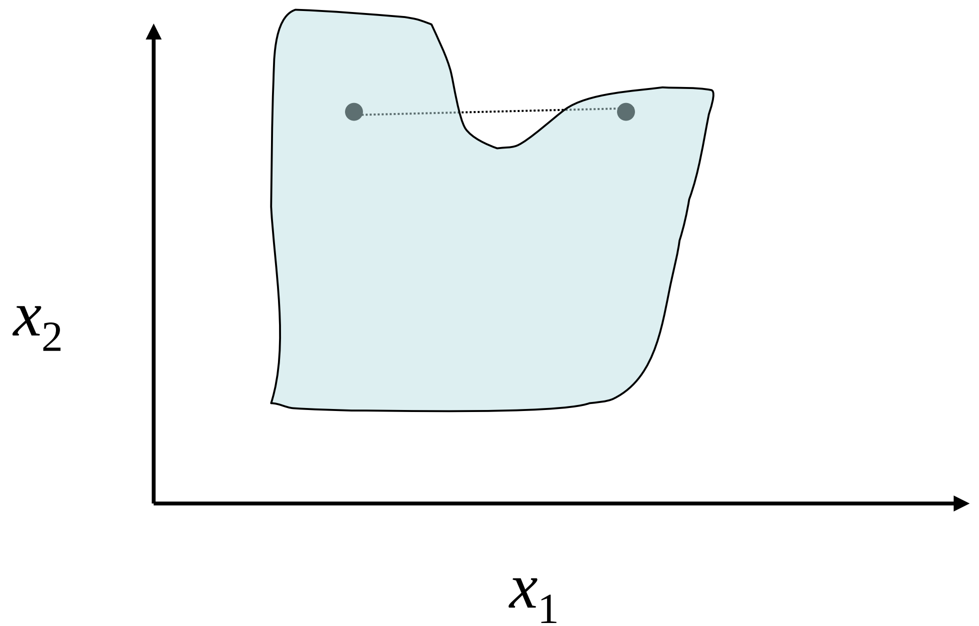
Convex Functions, continued

This feasible region to an LP is convex since a line drawn between any two feasible points remains completely within the region.



A Nonconvex Region

If the feasible region looks like



then the constraint set forms a nonconvex region. The problem is clear – even for a simple objective function like $f(x)=x_2$, we see one local and one global maximum

The Use of Convexity

Take the following optimization problem:

$$\begin{aligned} \min \quad & f(\bar{x}) \\ \text{s.t.} \quad & \bar{g}(\bar{x}) \le 0 \end{aligned}$$

- Regardless of problem type, if all the $\bar{g}(\bar{x})$ are convex, then the feasible region will be a convex set.
- If the objective function $f(x)$ is also convex, then we have a <u>convex programming problem</u>
- This guarantees us that the problem has only one minimum, the global one
- This also works for maxima – just multiply the objective function and constraints by -1

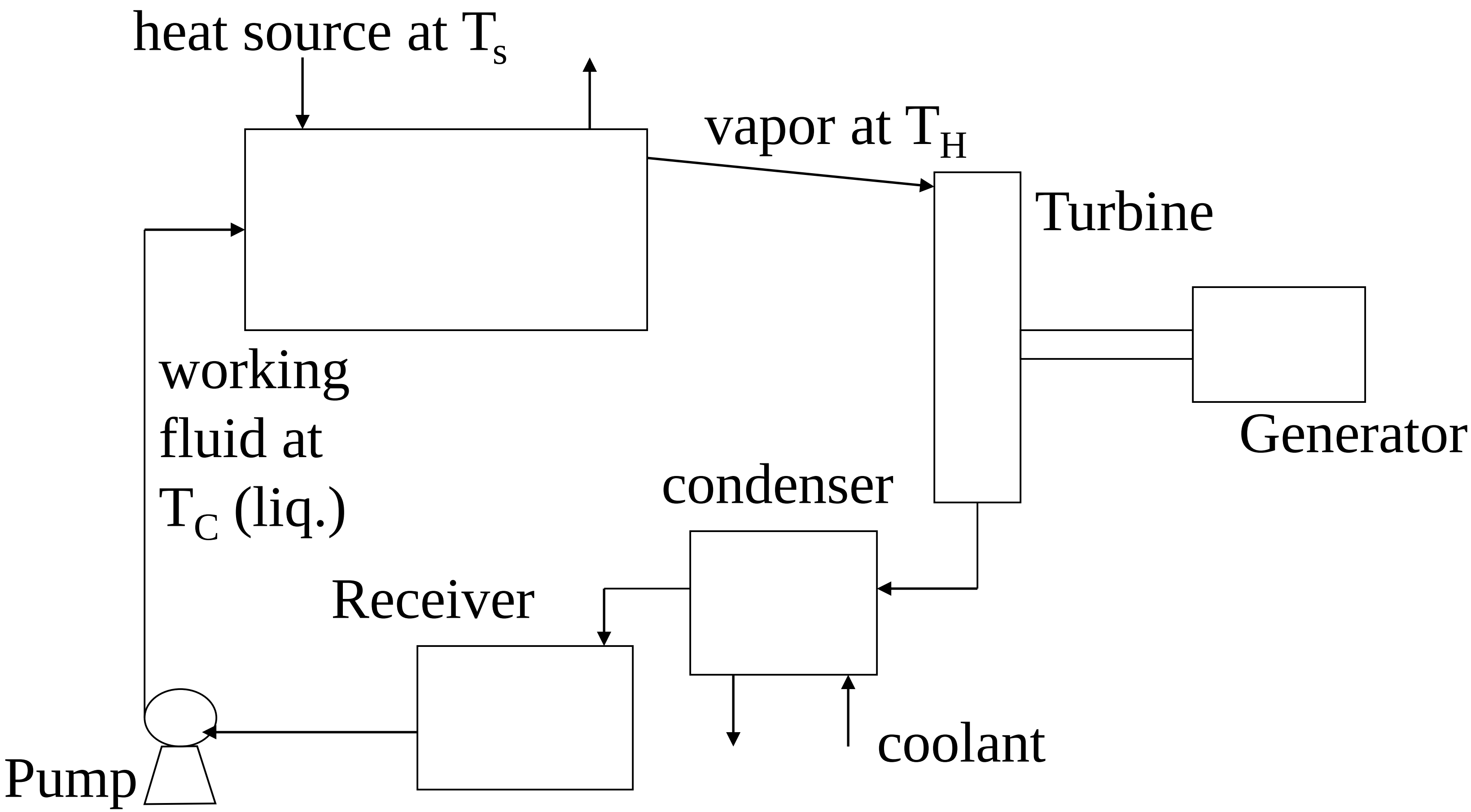
Example: Recovery of Waste Heat from a Chemical Plant

- It is well-known that money can be saved by using hot waste streams in heat exchangers to heat up process fluids prior to disposal

- This brings up a number of questions:
  - How many heat exchangers should we use?
  - How large should each exchanger be?
  - What type of a cycle should I use to recover the heat?

Heat Recovery Cycle

Let's look at a simple one-exchanger recovery cycle



Heat Recovery Cycle, cont.

- If cooling water is used as the condensing fluid, then T_C is known. T_S is also fixed.

- Thus we really have a one-variable problem: Find T_H which minimizes costs (operating and capital) but returns as much energy as possible from the heat source.

Problem Formulation

- How much work can we get out of the heat source (ideally)?

$$P_{\text{Turbine}} = Q\left[\frac{T_S - T_C}{T_S}\right]$$

where P is the power, and Q is the energy of the heat source in BTU/hr

- However, we use energy to condense the working fluid:

$$P_{\text{Cond}} = Q\left[\frac{T_H - T_C}{T_H}\right]$$

Problem Formulation, cont.

- So the total power output is the difference of these two:

$$\Delta P = Q\left[\frac{T_C}{T_H} - \frac{T_C}{T_S}\right]$$

- What is the cost to operate the cycle?

$$\text{Op. Cost} = C_H \eta\, yQ\left[\frac{T_C}{T_H} - \frac{T_C}{T_S}\right]$$

where C_H is the pumping cost, η is the efficiency (normally 70%), and y is the no. of hrs/year operating

- This is minimized when $T_H = T_S$. Is that possible? No!

Problem Formulation, cont.

- We need the capital cost of the exchanger:

$$\text{Cap. Cost} = \frac{C_A Q r}{U(T_S - T_H)}$$

where C_A is the cost/area, r is the annualization factor, and U is the heat transfer coefficient

- Now, how do we formulate the problem?

Add the capital and operating cost equations, and **minimize**

Problem Solution

- The objective function becomes:

$$f(T_H) = C_H \eta\, yQ\left[\frac{T_C}{T_H} - \frac{T_C}{T_S}\right] + \frac{C_A Qr}{U(T_S - T_H)}$$

- For a low order function like this, solve analytically:

$$f'(T_H) = C_H \eta\, yQ\left[-\frac{T_C}{T_H^2}\right] + \frac{C_A Qr}{U(T_S - T_H)^2} = 0$$

- That's just a quadratic. How do we know which solution is best?
- Usually, one is non-physical ($T_H > T_S$)

Problem Solution, cont.

- The solution is

$$T_H = T_S\left(\frac{\alpha_1 - \sqrt{\alpha_1\,\alpha_2}}{\alpha_1 - \alpha_2}\right)$$

$$\alpha_1 = C_H\,\eta\, y T_C U \qquad \alpha_2 = C_A r$$

- Note this is the physical solution only, and that the Q's cancel out.

- There are lots of other examples of unconstrained optimization in chemical engineering – just remember to formulate carefully

Steepest Descent

- Goal: minimize $f(\bar{x})$
  – unconstrained only

  – must be differentiable over the interval of interest

Search Directions

- Idea: look for a series of points for which $f(\overline{x}^{k+1}) < f(\overline{x}^{k})$

- This will eventually find a local minimum, or get stuck in a saddle point

- So what direction do we have to move in to get from $\overline{x}^{k}$ to $\overline{x}^{k+1}$?

Search Directions, cont.

- The key is to show that any direction $\bar{s}$ which satisfies

$$\nabla^T f(\bar{x})\bar{s} < 0$$

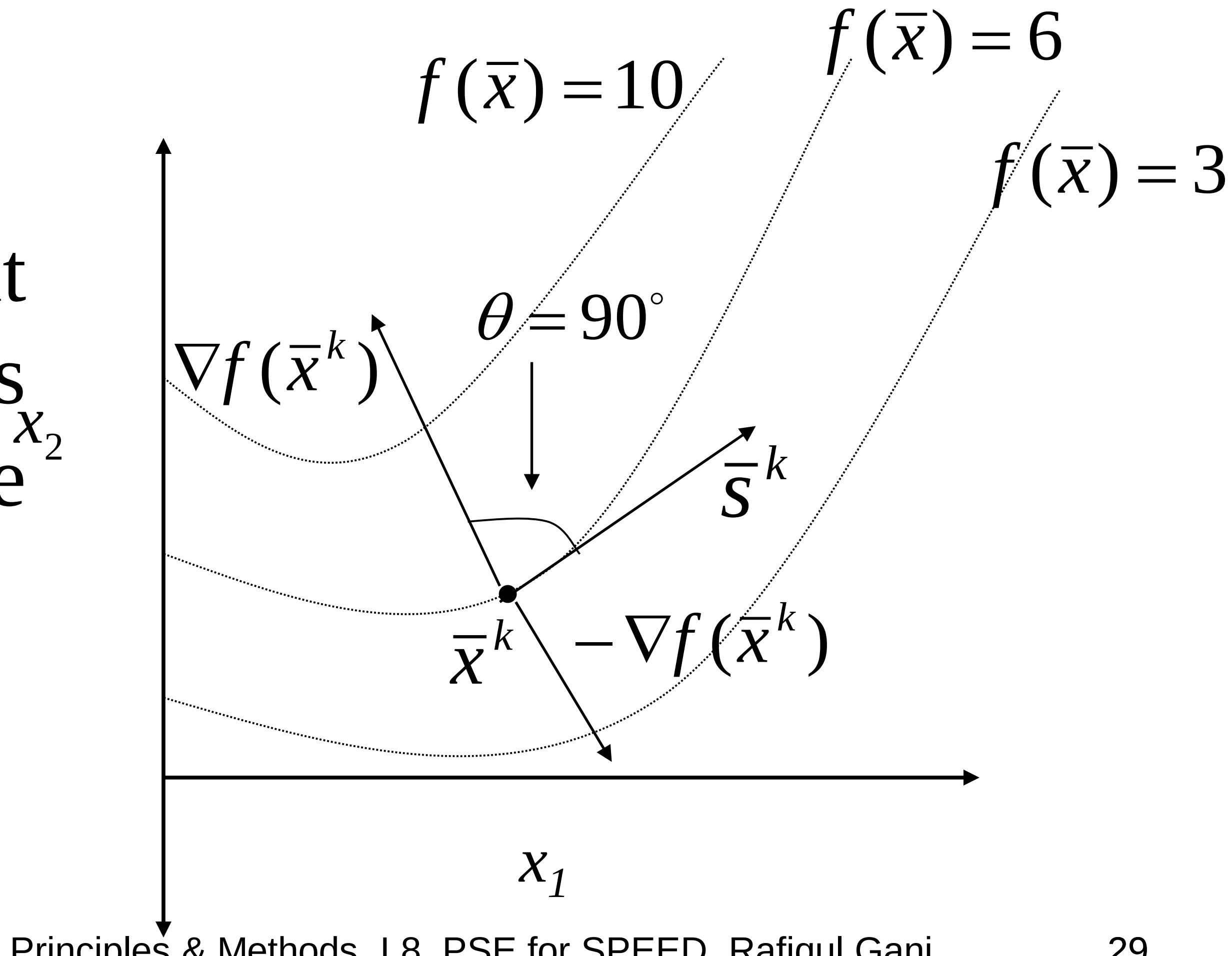
will lead to an improvement in $\bar{x}$

- Note this is vector-vector multiplication, between the gradient of the function and the search direction

A 2-D Example

- Note $\nabla f(\bar{x}^k)$ is the direction of the greatest increase in *f* :

Note the gradient vector is always orthogonal to the contour



Two Major Issues

- So, based on local information, we see that

$$\bar{s}^k = -\nabla f(\bar{x}^k)$$

is the search direction which will get us to the minimum the fastest

- But how far should we go in the direction $\bar{s}^k$ before recomputing $\nabla f(\bar{x}^{k+1})$ and getting a new $\bar{s}^{k+1}$?

Steepest Descent: algorithm

- We need a way to get from one point to the next, called an <u>iteration formula</u>

- For Steepest Descent, we get

$$\bar{x}^{k+1} = \bar{x}^{k} + \Delta\bar{x}^{k} = \bar{x}^{k} + \alpha^{k}\bar{s}^{k}$$

$$= \bar{x}^{k} - \alpha^{k}\nabla f(\bar{x}^{k})$$

- α^k is a scalar defining how far in the search direction to move

Steepest Descent: line searches

- How do we get α^k ?
- The logical way is to perform a one-D line search
  - find the minimum value of *f* along a line through the current point in the search direction
  - this finds the optimum in the fewest steps, but is time consuming
- In general, finding the optimal α^k is not needed

Example: Steepest Descent

- Let's use a function where the contours are easy to visualize:

$$\min\ f(\bar{x}) = x_1^2 + x_2^2$$

- Start at the point (2,2)

$$x^{(1)} = [2,2] - \alpha^{k} \nabla f(\bar{x}^{k})$$

- So first find the gradient:

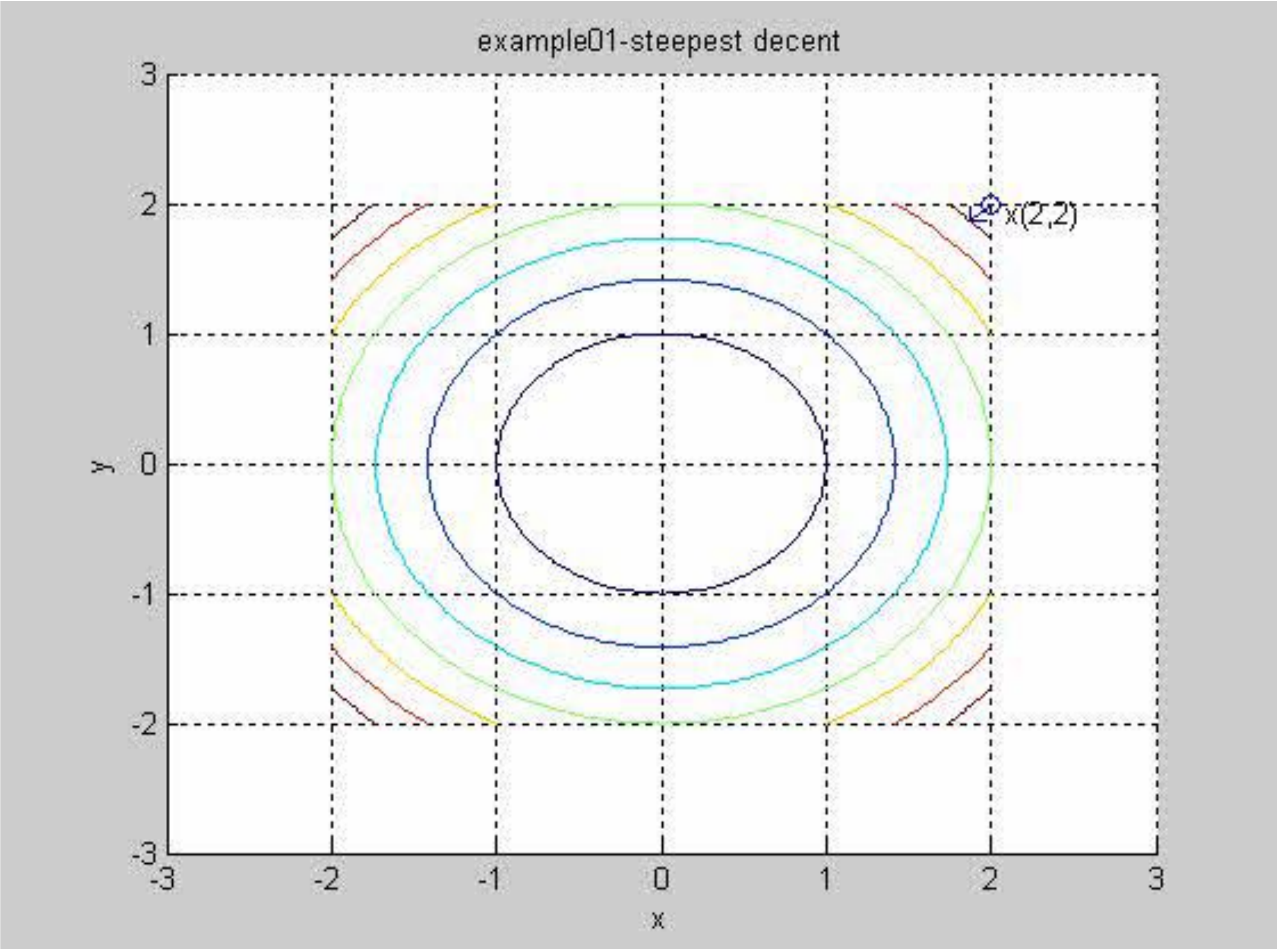
$$\nabla f(\bar{x}) = [2x_1, 2x_2] \quad \text{so} \quad \nabla f(\bar{x}^0) = [4,4]$$

Example: Steepest Descent

- So our search direction is $\bar{s}^k = -\begin{bmatrix} 4 \\ 4 \end{bmatrix}$

- Let's just set $\alpha = 0.1$

- Since all contours are concentric, the search direction will not change for any iteration.


example01-steepest decent
x(2,2)
x
y
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

Example: Steepest Descent

- What would happen if $\alpha = 0.5$?

$$x^{(1)} = [2,2] - 0.5[4,4] = [0,0]$$

- We would reach the optimum in one step!

- To compute this, find the optimal α:

$$\frac{d}{d\alpha}\left[f(\bar{x}^k + \alpha^k \bar{s}^k)\right] = 0$$

- Solve to see $\alpha = 0.5$

Nonlinear Programming

- Abbreviated NLP
  - nonlinear constraints and/or a nonlinear objective function
  - no integer/binary variables
- An important relaxation for solving MINLP's
- What are the major challenges here?
  - Locally optimal solutions may exist
  - Solution found can therefore be initial guess-dependent
  - Solutions do not necessarily lie at extreme points

Example: Lagrange Multipliers

- Let's look at a different example:

$$\begin{aligned} \min \quad & f(\bar{x}) = x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 1 = 0 \end{aligned}$$

- Note the only nonlinearity is in the constraint – still an NLP

The Lagrangian Function

- If we define the Lagrangian Function as:

$$L(\bar{x}, \lambda) = f(\bar{x}) + \lambda\, h(\bar{x})$$

- We can then write

$$\nabla L(\bar{x}, \lambda)\,\big|_{\bar{x}^*, \lambda^*} = 0$$

which is a necessary condition for optimality, along with $h(\bar{x}) = 0$ for feasibility

The Lagrangian Function: Example

- Let's write this for our example:

$$\min \qquad f(\bar{x}) = x_1 + x_2$$

$$\text{s.t.} \qquad h(\bar{x}) = x_1^2 + x_2^2 - 1 = 0$$

$$L(\bar{x}, \lambda) = f(\bar{x}) + \lambda\, h(\bar{x}) = x_1 + x_2 + \lambda\,(x_1^2 + x_2^2 - 1)$$

- Take first partials, and include the constraint:

$$1 + 2\lambda\, x_1 = 0$$

$$1 + 2\lambda\, x_2 = 0$$

$$x_1^2 + x_2^2 - 1 = 0$$

Three equations, three variables: solve for the optimal x's and λ

The Lagrangian Function: Example

- The solution (found numerically) is

$$x_1 = \pm 0.707$$
$$x_2 = \pm 0.707$$
$$\lambda = \pm 0.707$$

- The Lagrange Multiplier tells us how sensitive the objective function is to changes in the constraint *h*, much like a marginal cost.

Lagrange Multipliers for Inequalities

- For inequality constraints, Lagrange multipliers are denoted by u
- Note that if we set $u = 0$ for inactive constraints, then we can put all of the constraints into this format without knowing which are active
- This is a statement of <u>complementary slackness</u> – if the constraint is not satisfied exactly, the multiplier is zero

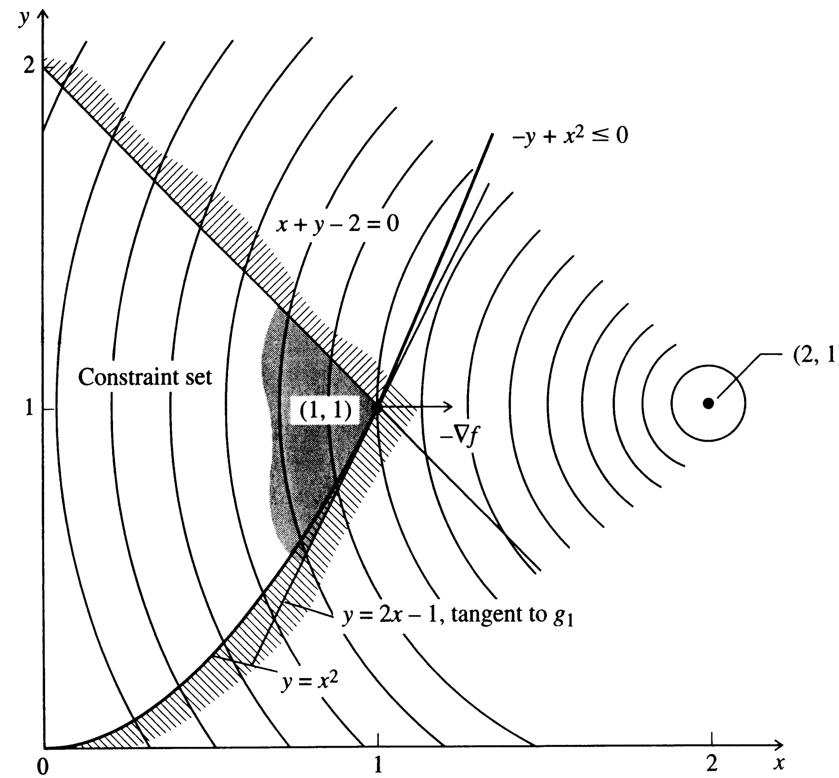
Example: Inequality Constrained NLP

min $\quad f(x, y) = (x-2)^2 + (y-1)^2$

s.t. $\quad g_1(x, y) = -y + x^2 \leq 0$

$g_2(x, y) = y + x \leq 2$

$g_3(y) = y \geq 0$



• Note the optimum of the constrained problem is [1,1]

• First two constraints are active, third is inactive at [1,1]

Logic of an Optimum Point

- Think of an optimum in the following way:

  At any local optimum, no small feasible change in the values of the variables will improve the value of the objective function

- This logic allows us to write conditions required for a point to be locally optimal. Writing such conditions converts a constrained NLP problem into a nonlinear equation solving problem

Feasible Descent Directions

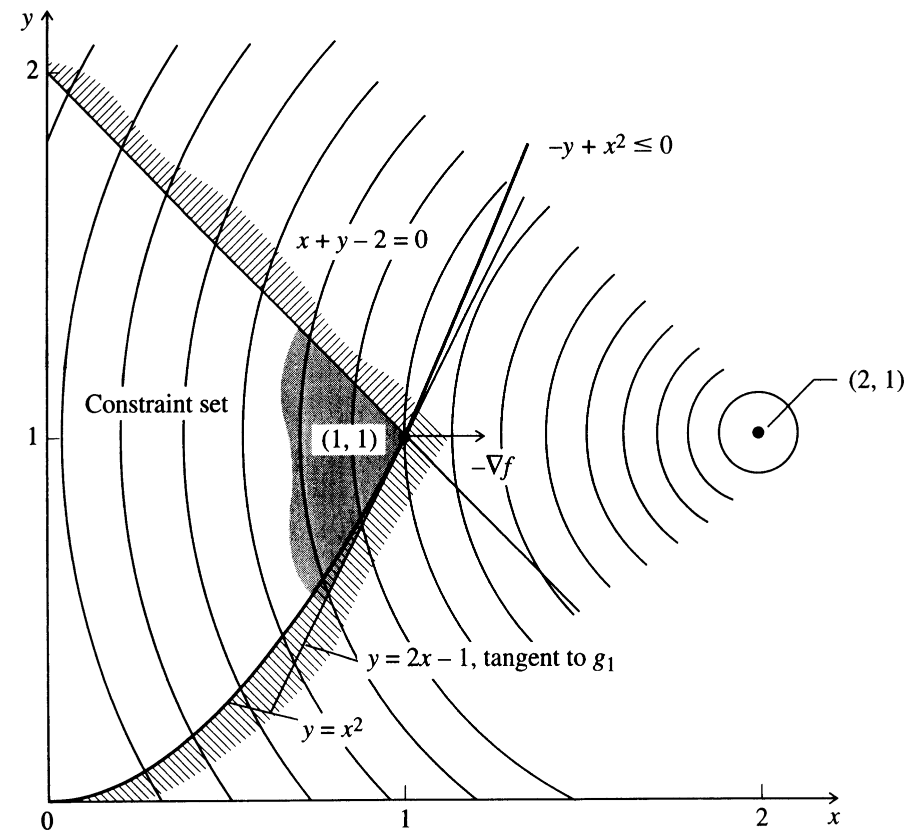
$$\min \quad f(x,y) = (x-2)^2 + (y-1)^2$$

$$\text{s.t.} \quad g_1(x,y) = -y + x^2 \leq 0$$

$$g_2(x,y) = y + x \leq 2$$

$$g_3(y) = y \geq 0$$

- Since none of the feasible search directions are within 90° of – grad *f*, we must be at an optimal point



The KKT Conditions, Algebraically

- If we convert this geometric logic to a set of algebraic equations, we can solve them to find our optimum point

- This idea was worked on independently by Karush and Kuhn & Tucker in the 1960’s

KKT Conditions – General Form

- For the general NLP problem:

$$\begin{aligned} \min \quad & f(\bar{x}) \\ \text{s.t.} \quad & h_i(\bar{x}) = b_i \\ & g_j(\bar{x}) \le c_j \end{aligned}$$

we can write the KKT conditions as

$$\nabla_{\bar{x}} L(\bar{x}^*, \bar{\lambda}^*, \bar{u}^*) = 0$$

$$\bar{u} \ge 0, \quad u_j[g_j(\bar{x}^*) - c_j] = 0$$

$$h_i(\bar{x}) = b_i, \quad g_j(\bar{x}) \le c_j$$

- The solutions to these equations are all the extreme points: minima, maxima, saddle points

The Second-Order KKT Conditions

- For constrained optimization, we can classify extreme points by looking at the second derivatives of the Lagrangian at that point:

$$\nabla^2_{\bar{x}} L \quad \text{(an } m \text{ x } m \text{ matrix, the Hessian)}$$

positive semi-definite	local minimum
negative semi-definite	local maximum
indefinite	saddle point

Proving Convexity

- Evaluating those second-order KKT conditions is equivalent to testing the convexity of the Lagrangian function

- A procedure to do this is as follows:
  - Construct the Hessian matrix $H(x)$
  - Compute its eigenvalues, check their signs
  - Refer to the chart to judge convexity

The Hessian Matrix

- It’s just the matrix of second partial derivatives:

$$H(\bar{x}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_3} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \dfrac{\partial^2 f}{\partial x_2 \partial x_3} \\ \dfrac{\partial^2 f}{\partial x_3 \partial x_1} & \dfrac{\partial^2 f}{\partial x_3 \partial x_2} & \dfrac{\partial^2 f}{\partial x_3^2} \end{bmatrix}$$

Eigenvalues and Positive Definiteness

- Here's how we relate the signs of the eigenvalues to convexity:

$f(x)$	$H(x)$	All Eigenvalues
strictly convex	positive definite	>0
convex	positive semidefinite	≥ 0
neither	indefinite	some ≥ 0, some ≤ 0
concave	negative semidefinite	≤ 0
strictly concave	negative definite	<0

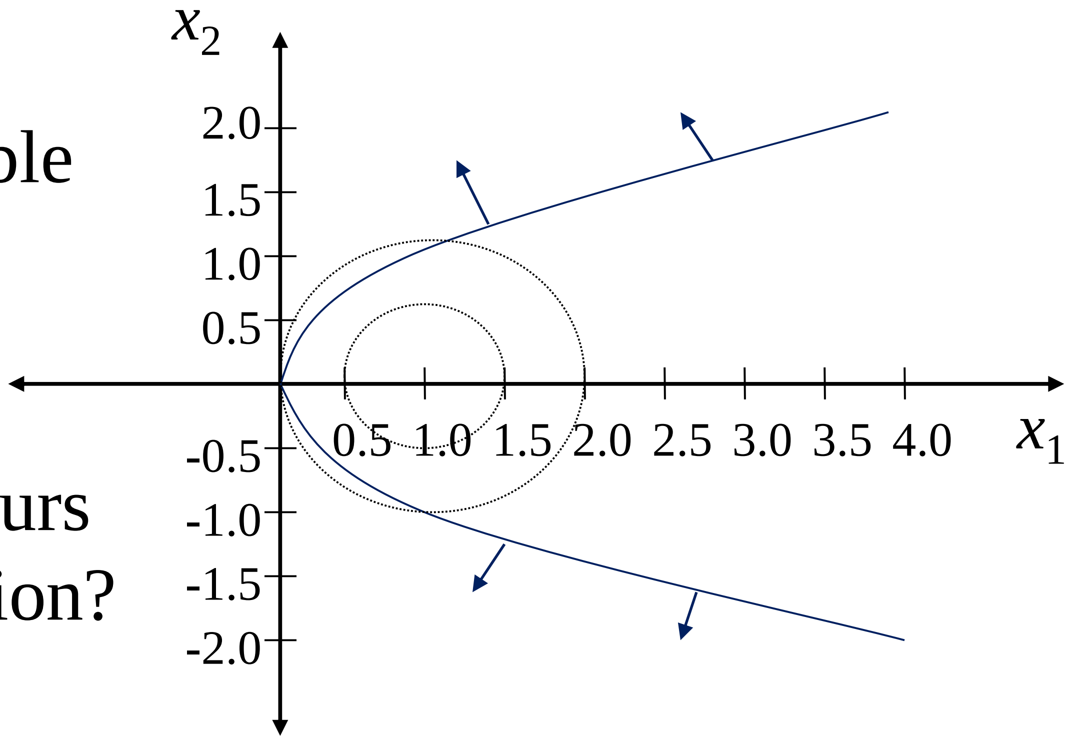
Example

$$\min \qquad f(\bar{x}) = (x_1 - 1)^2 + x_2^2$$

$$\text{s.t.} \qquad g_1(\bar{x}) = x_1 - x_2^2 \le 0$$

- What does the feasible region look like?

- Where are the contours of the objective function?



Example, continued

$$\min \quad f(\bar{x}) = (x_1 - 1)^2 + x_2^2$$

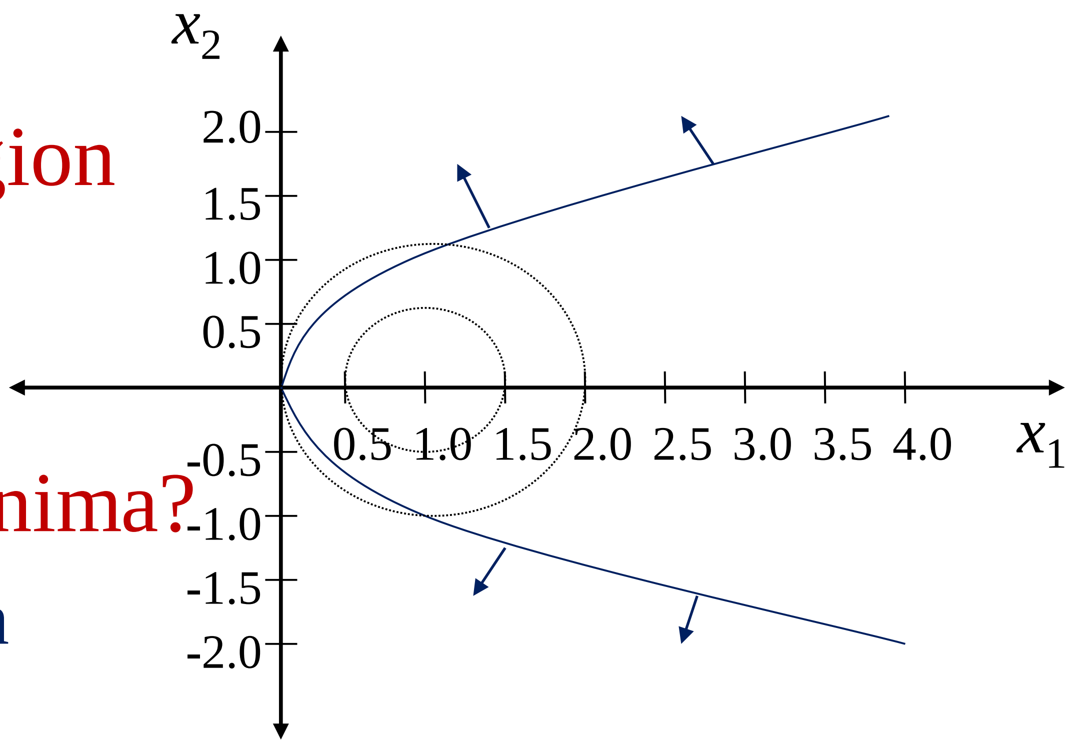
$$\text{s.t.} \quad g_1(\bar{x}) = x_1 - x_2^2 \le 0$$

- Is the feasible region convex?

  No

- Where are the minima?

  Somewhere between [1,1] and [0,0]



Writing the KKT Conditions

$$\begin{aligned} \min \quad & f(\bar{x}) = (x_1 - 1)^2 + x_2^2 \\ \text{s.t.} \quad & g_1(\bar{x}) = x_1 - x_2^2 \leq 0 \end{aligned}$$

- Now write the KKT conditions:

$$L = (x_1 - 1)^2 + x_2^2 + u(x_1 - x_2^2)$$

$$\nabla_{\bar{x}} L = \left[2(x_1 - 1) + u \,,\, 2x_2 - 2ux_2\right]$$

- So we get:

$$2x_1 - 2 + u = 0 \qquad x_2(2 - 2u) = 0$$

$$x_1 - x_2^2 \leq 0 \qquad u(x_1 - x_2^2) = 0$$

$$u \geq 0$$

Solving the KKT Conditions

$$2x_1 - 2 + u = 0 \qquad x_2(2 - 2u) = 0$$

$$x_1 - x_2^2 \le 0 \qquad u(x_1 - x_2^2) = 0$$

$$u \ge 0$$

Can we solve this?

Yes, we have 3 equations, 3 variables.

Use the inequalities to check feasibility.

Solutions are:

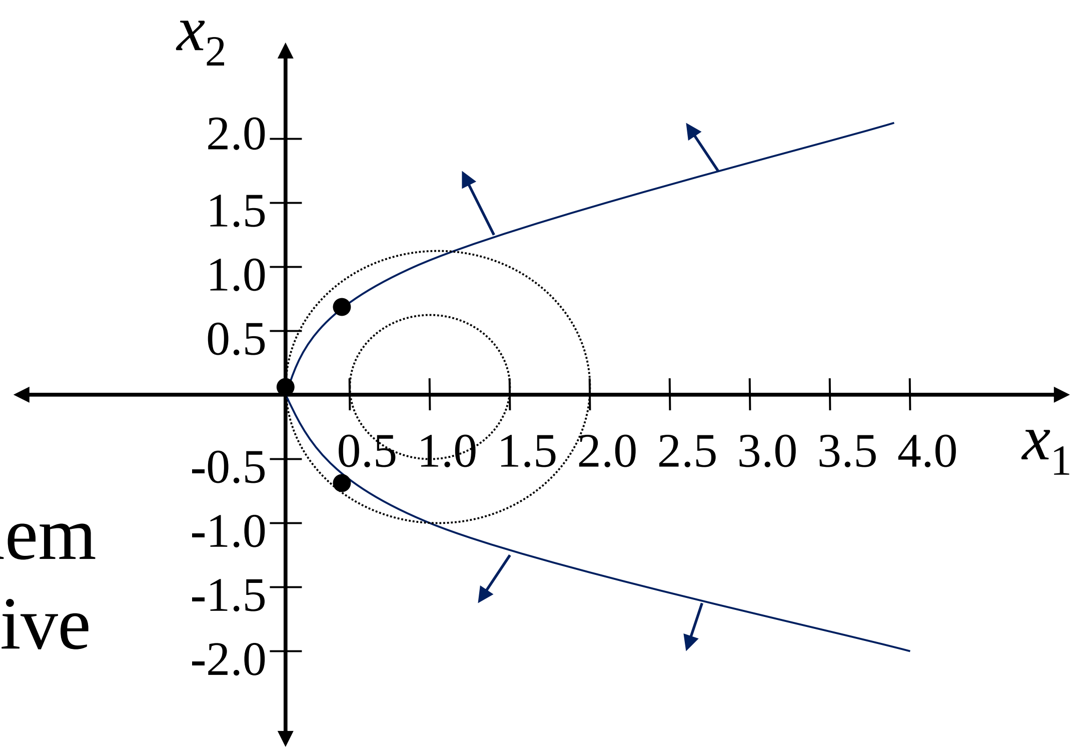
$$[x_1, x_2, u] = \begin{array}{l} [0,0,2] \ \text{feasible} \\ [1,0,0] \ \text{infeasible} \\ [0.5, \sqrt{0.5}, 1] \ \text{feasible} \\ [0.5, -\sqrt{0.5}, 1] \ \text{feasible} \end{array}$$

Example, continued

$$\min \qquad f(\bar{x}) = (x_1 - 1)^2 + x_2^2$$

$$\text{s.t.} \qquad g_1(\bar{x}) = x_1 - x_2^2 \leq 0$$

- So we have three extreme points.



- Now we classify them using second derivative information

Second-order KKT Conditions

$$\nabla_{\bar{x}} L = \left[2(x_1 - 1) + u \, , \, 2x_2 - 2ux_2 \right]$$

Write the second order KKT conditions:

$$\nabla_{\bar{x}}^2 L = \begin{bmatrix} 2 & 0 \\ 0 & 2 - 2u \end{bmatrix}$$

We get:

$$\text{At}\,[0,0,2]: \; \nabla_{\bar{x}}^2 L = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

$$\text{At}\,[0.5, \pm\sqrt{0.5}, 1]: \; \nabla_{\bar{x}}^2 L = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Finding the Eigenvalues

Now solve for the eigenvalues:

$$\text{At}\,[0,0,2]:\ \det[\lambda I - \nabla_{\bar{x}}^2 L] = (\lambda - 2)(\lambda + 2) = 0$$

$$\lambda = \pm 2 \quad \text{indefinite}$$

$$\text{At}\,[0.5, \pm\sqrt{0.5}, 1]:\ \det[\lambda I - \nabla_{\bar{x}}^2 L] = (\lambda - 2)(\lambda) = 0$$

$$\lambda = 0,2 \quad \text{positive semi-definite, minimum}$$

Successive Quadratic Programming (SQP)

- For process flowsheet optimization, SQP has been shown to be one of the most efficient algorithms available, based on the number of function evaluations required
- An overview: To solve

$$\min f(\bar{x})$$

$$\text{subject to:} \quad \bar{g}(\bar{x}) \geq 0$$

$$\bar{h}(\bar{x}) = 0$$

At each iteration:

- Formulate a quadratic approximation for $f(x)$
- Linearize the constraints
- Solve this simplified QP to give a search direction
- Determine the step in the search direction
- Check for convergence

SQP: Theoretical Basis

- Consider an NLP with only equality constraints
- The KKT conditions may be written

$$\nabla_{\bar{x}} L = \nabla f(\bar{x}) + \sum_j \lambda_j \nabla h_j(\bar{x}) = 0 \quad \text{(i)}$$

$$\bar{h}(\bar{x}) = 0 \quad \text{(ii)}$$

- One way to solve these is by Newton's method. If you write down the iteration formula for that, you get

$$\begin{bmatrix} \nabla^2_{\bar{x}} L & J \\ J^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = -\begin{bmatrix} \Delta_{\bar{x}} L \\ \bar{h} \end{bmatrix}$$

where J is the Jacobian of the eq. constraints

SQP: Theoretical Basis

- The solution of that linear system can be written as an optimization problem (minimization of the error). Those equations define an optimization problem which is quadratic. For a general NLP, that problem looks like:

$$\min F(s) = \nabla f(\bar{x})\bar{s}^T + \frac{1}{2}\ \bar{s}^T \nabla^2_{\bar{x}} L(\bar{x}, \bar{\lambda}, \bar{u})\ \bar{s}$$

$$\text{subject to:}\quad g_j(\bar{x}) + \bar{s}^T \nabla g_j(\bar{x}) \geq 0$$

$$h_i(\bar{x}) + \bar{s}^T \nabla h_j(\bar{x}) = 0$$

- This may look scary, but it’s a known problem which can be solved quickly by modern optimization software.

TABLE 9.1 Basic SQP Algorithm

0. Guess x^0, set $B^0 = I$ (the identity matrix is a default choice). Evaluate $f(x^0)$, $g(x^0)$, and $h(x^0)$.
1. At x^i, evaluate $\nabla f(x^i)$, $\nabla g(x^i)$, $\nabla h(x^i)$. If $i > 0$, calculate s and y.
2. If $i > 0$ and $s^T y > 0$, update B^i using the BFGS formula (9.35).
3. Solve:

$$\min_{d} \quad \nabla f(x^i)^T d + 1/2\, d^T B^i d \tag{QP1}$$

$$\text{s.t.} \quad g(x^i) + \nabla g(x^i)^T d \le 0$$

$$h(x^i) + \nabla h(x^i)^T d = 0$$

4. If $\| d \|$ is less than a small tolerance or the Kuhn Tucker conditions (9.26) are within a small tolerance, stop.
5. Find a stepsize α so that $0 < \alpha \le 1$ and $P(x^i + \alpha d) < P(x^i)$. Each trial stepsize requires additional evaluation of $f(x)$, $g(x)$, and $h(x)$.
6. Set $x^{i+1} = x^i + \alpha d$, $i = i + 1$ and go to 1.

Try to understand all the steps by going through this example

EXAMPLE 9.4 Performance of SQP

To illustrate the performance of SQP, we consider the solution of the following small nonlinear program:

$$\begin{aligned} &\text{Min } x_2 \\ &s.t.\ -x_2 + 2(x_1)^2 - (x_1)^3 \le 0 \\ &-x_2 + 2(1-x_1)^2 - (1-x_1)^3 \le 0 \end{aligned} \tag{9.37}$$

The feasible region for Eq. (9.37) is shown in Figure 9.6a along with the countours of the objective function. From inspection we see that $x^* = [0.5, 0.375]$.

Starting from the origin ($x^0 = [0, 0]^T$) and with $B^0 = I$, we linearize the constraints and solve the following quadratic program:

$$\begin{aligned} &\text{Min } d_2 + 1/2\,(d_1{}^2 + d_2{}^2) \\ &s.t.\ d_2 \ge 0 \\ &d_1 + d_2 \ge 1 \end{aligned} \tag{9.38}$$

From the solution of Eq. (9.38) a search direction is obtained with $d = [1, 0]^T$ with multipliers $\mu_1 = 0$ and $\mu_2 = 1$. The contours of this quadratic function along with the linearized constraints in Eq. (9.38) are shown in Figure 9.6b for the first SQP iteration. A line search along d determines a stepsize of $\alpha = 0.5$ and the new point is $x^1 = [0.5, 0]^T$. Note that this point lies outside of the feasible region. Also, at this new point we see that from:

SQP: Example

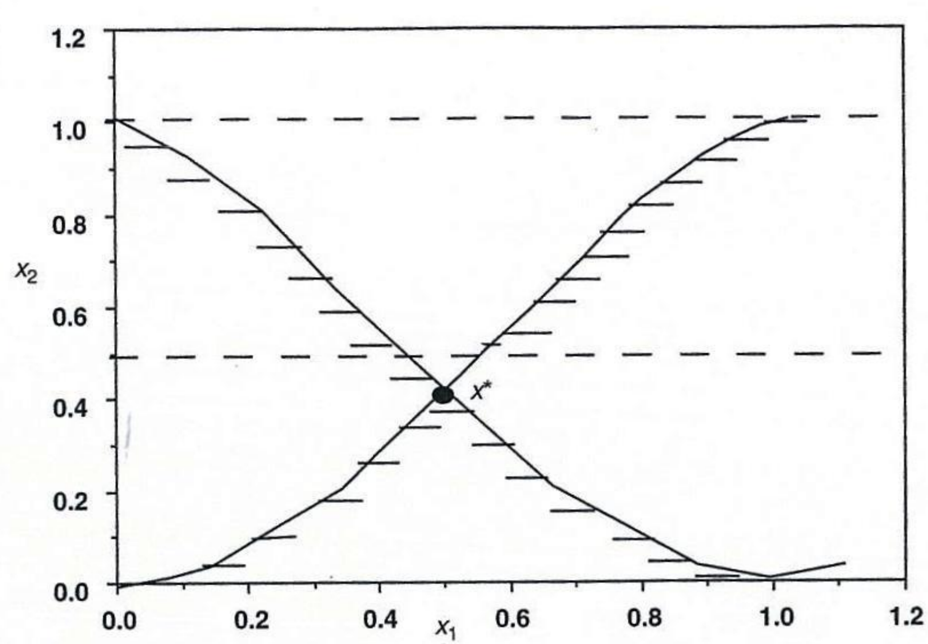
- Consider the following small NLP:

$$\min x_2$$

$$\text{subject to: } \; -x_2 + 2(x_1)^2 - (x_1)^3 \le 0$$

$$-x_2 + 2(1-x_1)^2 - (1-x_1)^3 \le 0$$

- The feasible region and optimal solution are shown here:



SQP: Example, cont.

- If we start with an initial guess of (0,0), and linearize the constraints (Taylor series), we get the following initial QP to solve:

$$\min\ s_2 + \frac{1}{2}\left(s_1^2 + s_2^2\right)$$

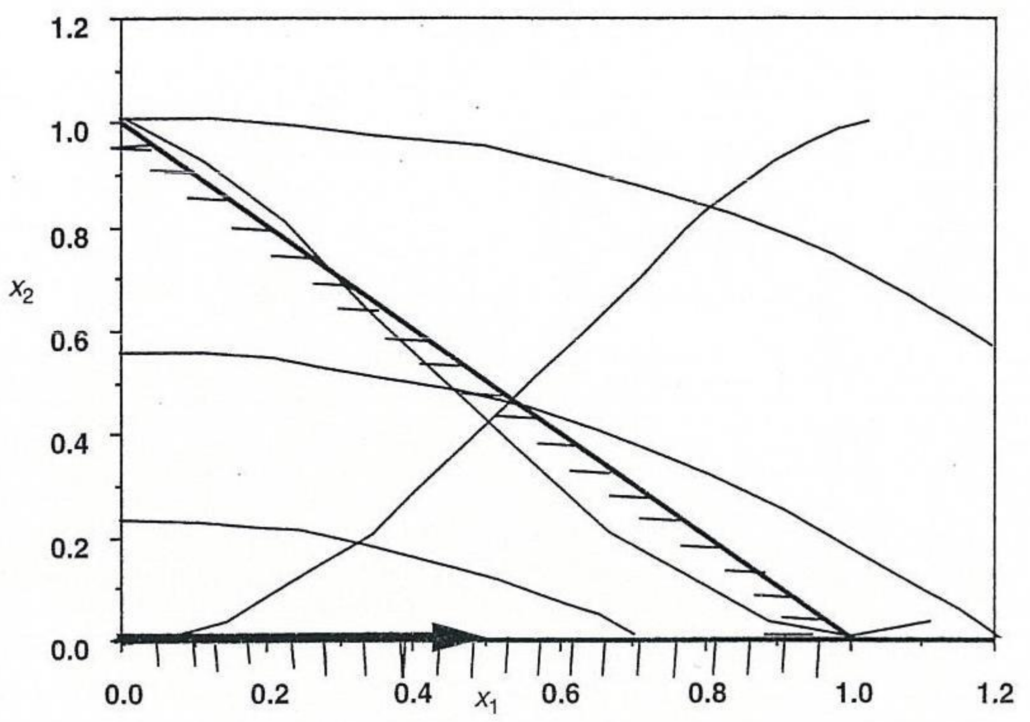
$$\text{subject to:}\quad s_2 \geq 0$$

$$s_1 + s_2 \geq 0$$

- Solving this QP gives a search direction of s=(1,0)
- If we do a 1-D line search to determine how far to move in that direction, we find α=0.5 and the new point is x_1=(0.5,0). Note the diagram on the next slide.

SQP: Example, cont.

- Note the diagram showing this step:



- Now we repeat the procedure, but start from the new point (0.5,0)

SQP: Example, cont.

- The second QP subproblem is

$$\min\ s_2 + \frac{1}{2}\left(s_1^2 + s_2^2\right)$$

$$\text{subject to}: \quad -1.25s_1 - s_2 + 0.375 \leq 0$$

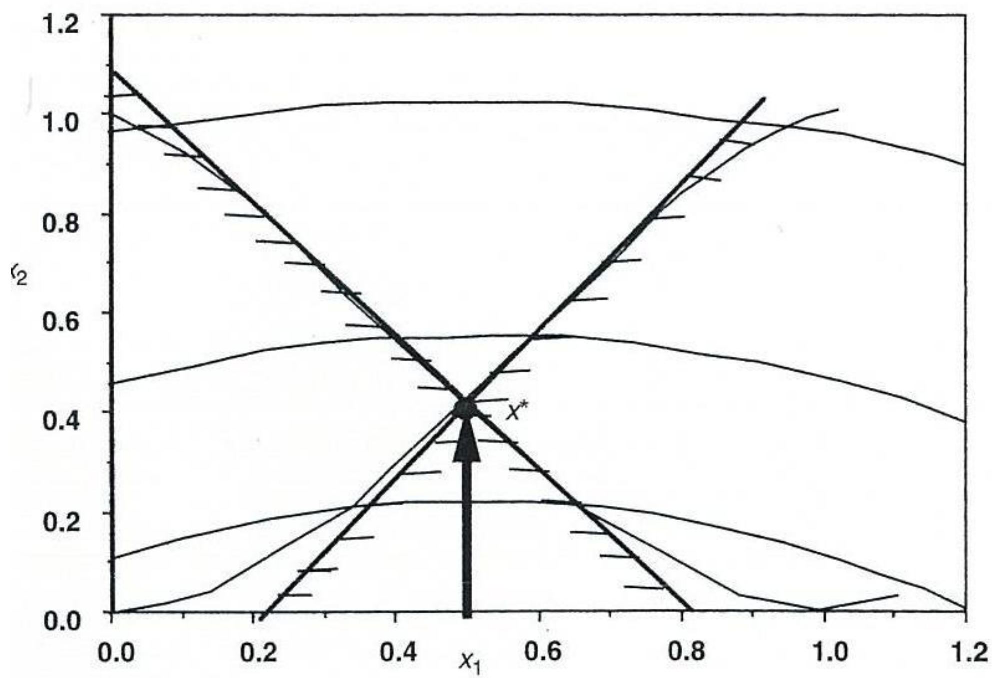
$$1.25s_1 - s_2 + 0.375 \leq 0$$

- Solving this QP gives a search direction of s=(0,0.375)

- If we do a 1-D line search to determine how far to move in that direction, we find α=1 and the new point is x_1=(0.5,0.375).

SQP: Solution

- SQP finds the solution in two steps for this NLP problem.



- One can also check the second-order conditions to ensure that this point is a local minimum