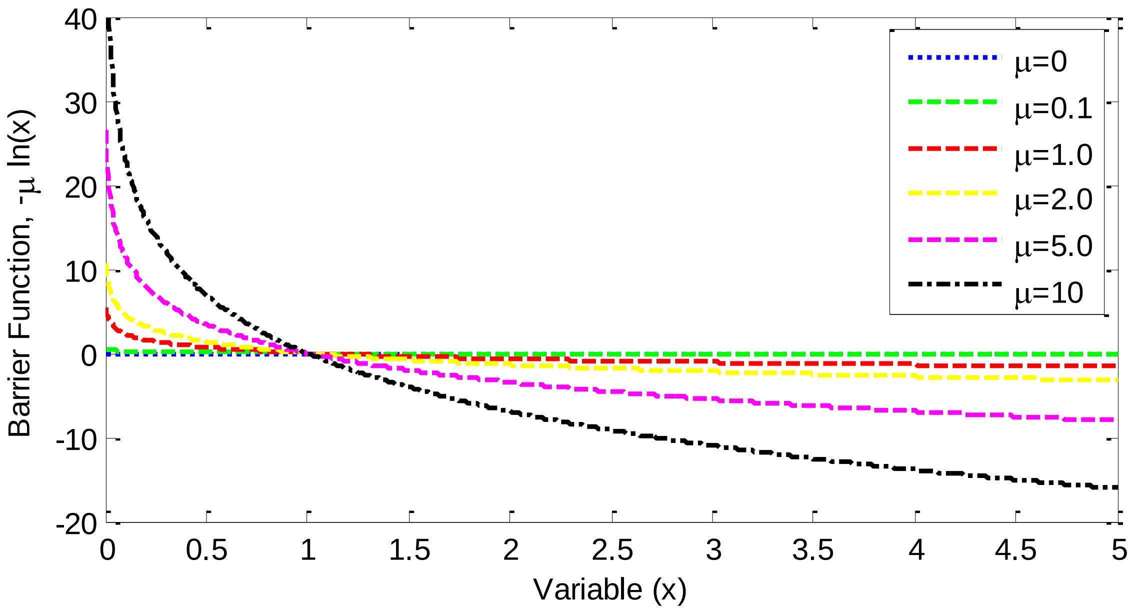
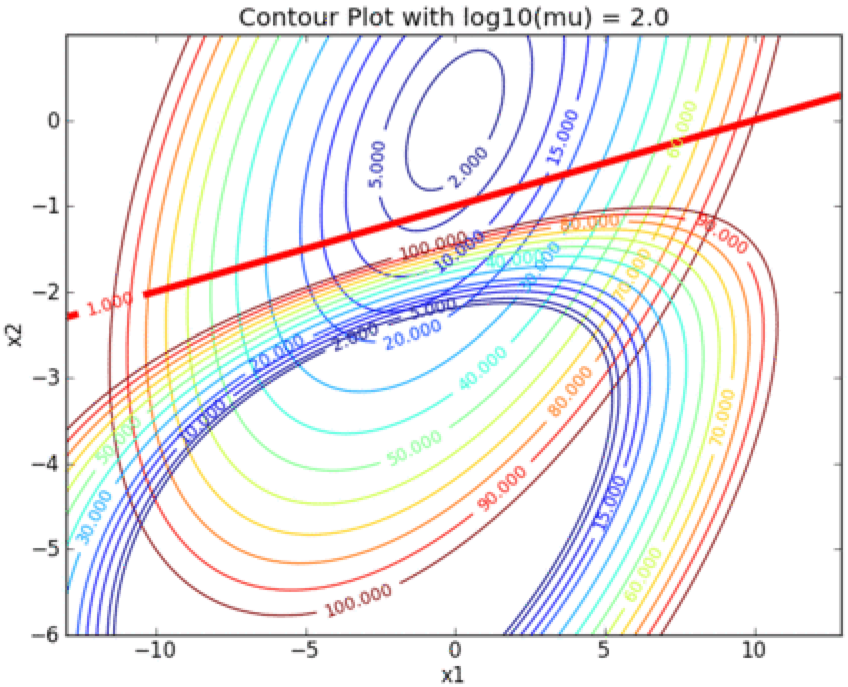
# Interior Point Methods

## ME575 – Optimization Methods

John Hedengren



# Overview

- Nonlinear constrained minimization
- Slack variables
- Barrier function
- Newton’s method for KKT conditions
- Interior Point Method (IPM)
- Tutorial Examples

# Nonlinear Constrained Optimization

$$
\begin{aligned}
\min_{x \in R^n} \quad & f(x) \\
s.t. \quad & c(x) = 0 \\
& x \geq 0
\end{aligned}
$$

- Linear, quadratic, or general nonlinear objective and constraints
- Convex optimization, local solution possible for non-convex problems
- Convert maximization by minimizing negative of the objective
- Convert general inequalities to simple inequalities with slack variables

# Slack Variables

$$\min_{x \in R^n} \quad f(x)$$
$$s.t. \quad g(x) \geq b$$
$$h(x) = 0$$

$$\Rightarrow$$

$$\min_{x \in R^n} \quad f(x)$$
$$s.t. \quad c(x) = 0$$
$$x \geq 0$$

- Complete worksheet on slack variables

$$\min_{x} \quad x_1 x_4 (x_1 + x_2 + x_3) + x_3$$
$$s.t. \quad x_1 x_2 x_3 x_4 \geq 25$$
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 = 40$$

- Additional slack variable tutorial:
  - http://apmonitor.com/me575/index.php/Main/SlackVariables

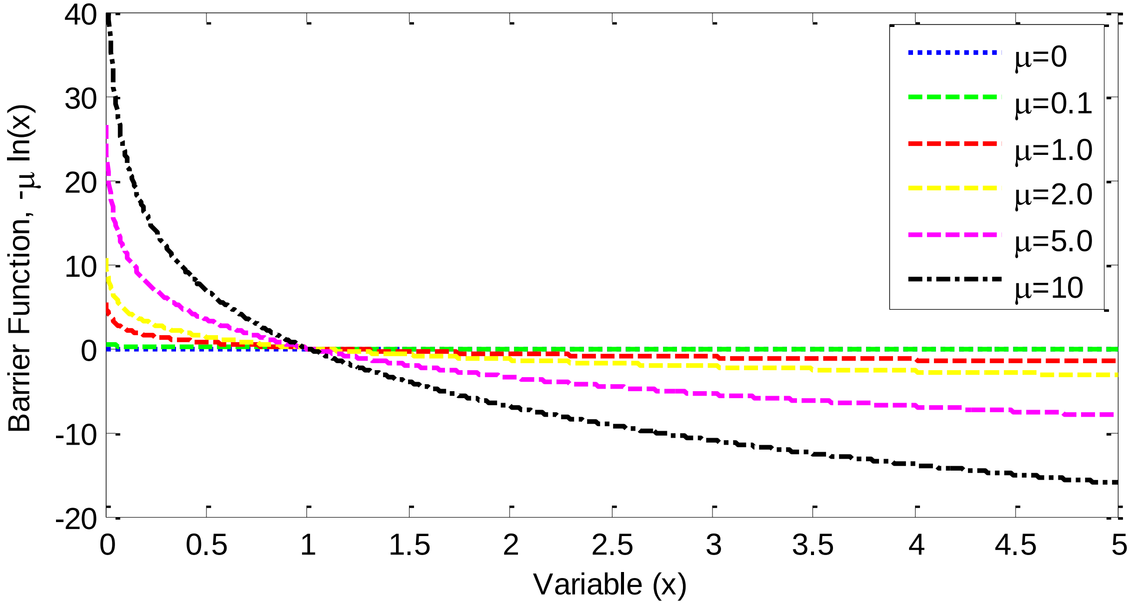
# Barrier Function

$$\min_{x \in R^n} \quad f(x) \qquad \qquad \min_{x \in R^n} \quad f(x) - \mu \sum_{i=1}^{n} \ln(x_i)$$

$$s.t. \quad c(x) = 0 \qquad \qquad s.t. \quad c(x) = 0$$

$$x \geq 0$$

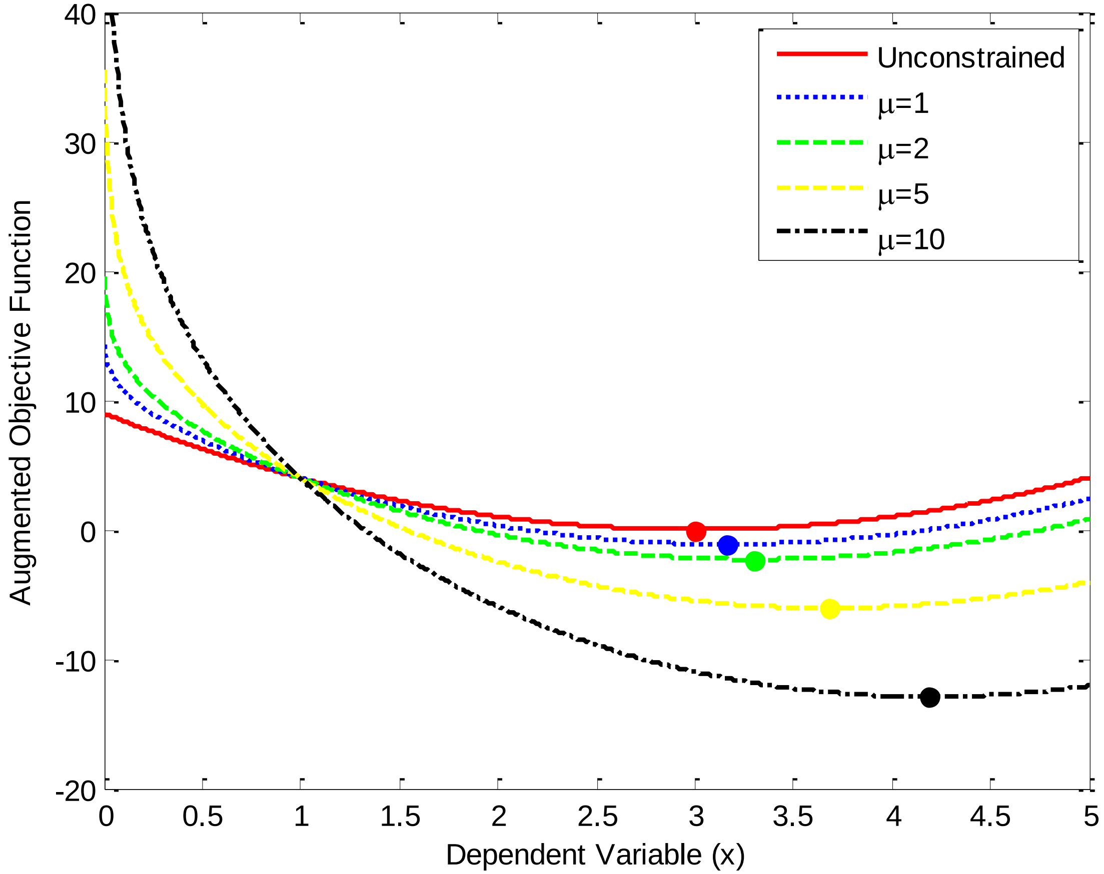
- Natural log barrier term for inequality constraints
- The term $\ln(x_i)$ is undefined for $x_i < 0$
- Search points only in interior feasible space



# Barrier Function Example 1

$$\min_{x \in R} \quad (x-3)^2$$
$$s.t. \quad x \geq 0$$

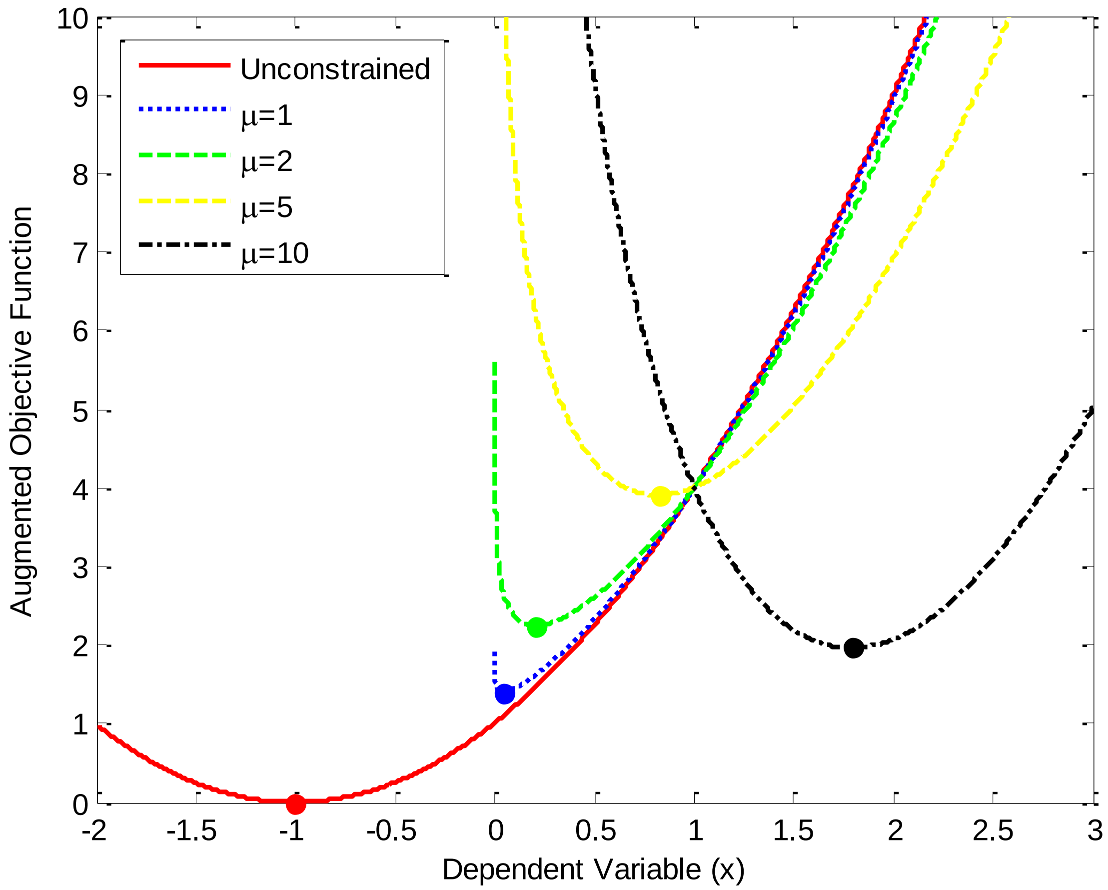
$$\min_{x \in R} \quad (x-3)^2 - \mu \ln(x)$$



# Barrier Function Example 2

$$\min_{x \in R} \quad (x+1)^2$$
$$s.t. \quad x \geq 0$$

$$\min_{x \in R} \quad (x+1)^2 - \mu \ln(x)$$



# How to Solve a Barrier Problem: Step 1

- KKT Conditions for Barrier Problem

$$\min_{x \in R^n} \quad f(x) - \mu \sum_{i=1}^{n} \ln(x_i) \qquad s.t. \quad c(x) = 0 \quad \Rightarrow \quad \nabla f(x) + \nabla c(x)\lambda - \mu \sum_{i=1}^{n} \frac{1}{x_i} = 0, \quad c(x) = 0$$

- Define $z_i = \frac{1}{x_i}$ and solve modified version of KKT conditions

$$\begin{aligned} \nabla f(x) + \nabla c(x)\lambda - z &= 0 \\ c(x) &= 0 \\ XZe - \mu e &= 0 \end{aligned}$$

# How to Solve a Barrier Problem: Step 2

- Find KKT solution with Newton-Raphson method

$$\begin{gathered}\nabla f(x) + \nabla c(x)\lambda - z = 0 \\ c(x) = 0 \\ XZe - \mu e = 0\end{gathered} \quad \Rightarrow \quad \begin{bmatrix} W_k & \nabla c(x_k) & -I \\ \nabla c(x_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k - z_k \\ c(x_k) \\ X_k Z_k e - \mu_j e \end{pmatrix}$$

- Where:

$$W_k = \nabla_{xx}^2 L(x_k, \lambda_k, z_k) = \nabla_{xx}^2 \left( f(x_k) + c(x_k)^T \lambda_k - z_k \right) \qquad Z_k = \begin{bmatrix} z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & z_n \end{bmatrix} \quad X_k = \begin{bmatrix} x_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & x_n \end{bmatrix}$$

# Find a Search Direction and Step Size: Example 3

$$\min_{x \in R^1} \quad f = y^2$$
$$s.t. \quad 2y \le 9$$
$$y \ge 0$$

$starting\ at\ y = 3,$
$\mu = 10, \lambda = 1, \alpha = 0.5$

$$\begin{bmatrix} W_k & \nabla c(x_k) & -I \\ \nabla c(x_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k - z_k \\ c(x_k) \\ X_k Z_k e - \mu_j e \end{pmatrix}$$

$$W_k = \nabla_{xx}^2 L(x_k, \lambda_k, z_k) = \nabla_{xx}^2 \left( f(x_k) + c(x_k)^T \lambda_k - z_k \right)$$

# How to Solve a Barrier Problem: Step 2

- Find KKT solution with Newton-Raphson method

$$\begin{gathered}\nabla f(x) + \nabla c(x)\lambda - z = 0 \\ c(x) = 0 \\ XZe - \mu e = 0\end{gathered} \quad \Rightarrow \quad \begin{bmatrix} W_k & \nabla c(x_k) & -I \\ \nabla c(x_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k - z_k \\ c(x_k) \\ X_k Z_k e - \mu_j e \end{pmatrix}$$

- Where:

$$W_k = \nabla_{xx}^2 L(x_k, \lambda_k, z_k) = \nabla_{xx}^2 \left( f(x_k) + c(x_k)^T \lambda_k - z_k \right) \qquad Z_k = \begin{bmatrix} z_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & z_n \end{bmatrix} \quad X_k = \begin{bmatrix} x_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & x_n \end{bmatrix}$$

- Rearrange into symmetric linear system

$$\begin{bmatrix} W_k + \Sigma_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{pmatrix} \qquad \Sigma_k = X_k^{-1} Z_k$$

- Solve for $d_k^z$ after the linear solution to $d_k^x$ and $d_k^\lambda$ with explicit solution

$$d_k^z = \mu_k X_k^{-1} e - z_k - \Sigma_k d_k^x$$

# Step Size ($\alpha$)

- Two objectives in evaluating progress
  - Minimize objective
  - Minimize constraint violations
- Two popular approaches
  - Decrease in merit function, $merit = f(x) + \nu \sum |c(x)|$
  - Filter methods
- Cut back step size until improvement

$$x_{k+1} = x_k + \alpha_k d_k^x$$

$$\lambda_{k+1} = \lambda_k + \alpha_k d_k^\lambda$$

$$z_{k+1} = z_k + \alpha_k d_k^z$$

# Convergence Criteria

- Convergence when KKT conditions are satisfied with a tolerance

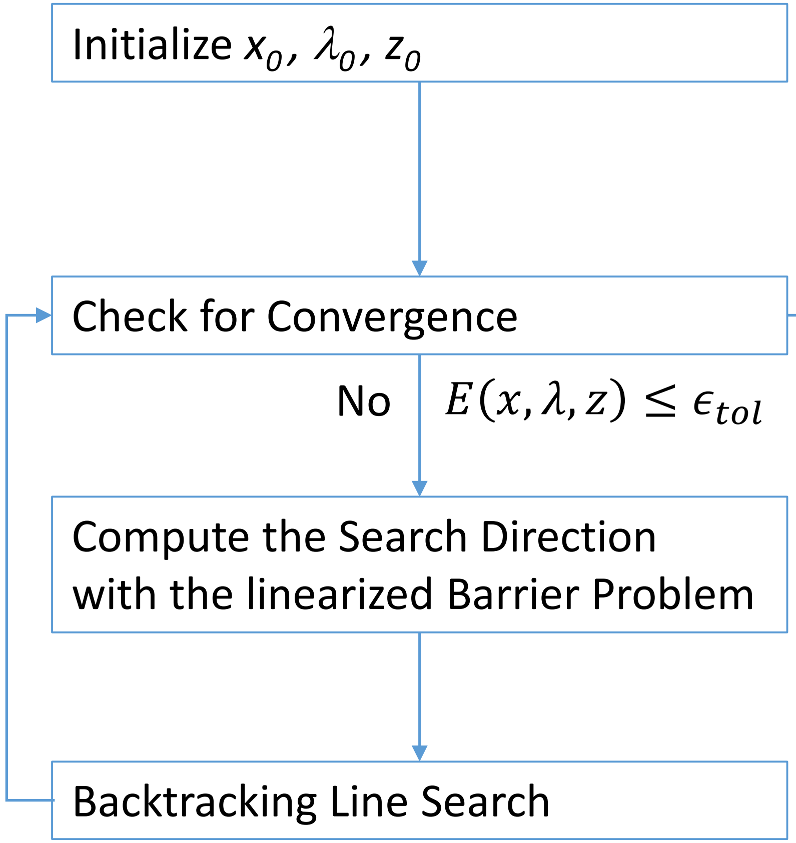
$$\max|\nabla f(x) + \nabla c(x)\lambda - z| \leq \epsilon_{tol}$$

$$\max|c(x)| \leq \epsilon_{tol}$$

$$\max|XZe - \mu e| \leq \epsilon_{tol}$$

- Tolerance for constraint violation may be more restrictive

# Interior Point Method Overview




$x_0 = feasible$

Initialize $\lambda$

$$z_0 = \frac{\mu}{x_0} \qquad \begin{bmatrix} I & \nabla c(x_0) \\ \nabla c(x_0)^T & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda_0 \end{bmatrix} = - \begin{bmatrix} \nabla f(x_0) - z_{L,0} - z_{U,0} \\ 0 \end{bmatrix}$$

Yes → Optimal Solution

$E(x, \lambda, z) \leq \epsilon_{tol}$

$$\Sigma_k = X_k^{-1} Z_k \qquad \begin{bmatrix} W_k + \Sigma_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^\lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + \nabla c(x_k)\lambda_k - Z_k \\ c(x_k) \end{bmatrix}$$

$$d_k^z = \mu X_k^{-1} e - z_k - \Sigma_k d_k^x$$

$$x_{k+1} = x_k + \alpha_k d_k^x$$

$$\lambda_{k+1} = \lambda_k + \alpha_k d_k^\lambda$$

$$z_{k+1} = z_k + \alpha_k d_k^z$$

Determine $\alpha$ by decrease in Merit Function or with Filter Methods

# Barrier Function Example 4

$$\min_{x \in R^2} \quad x_2(5 + x_1)$$

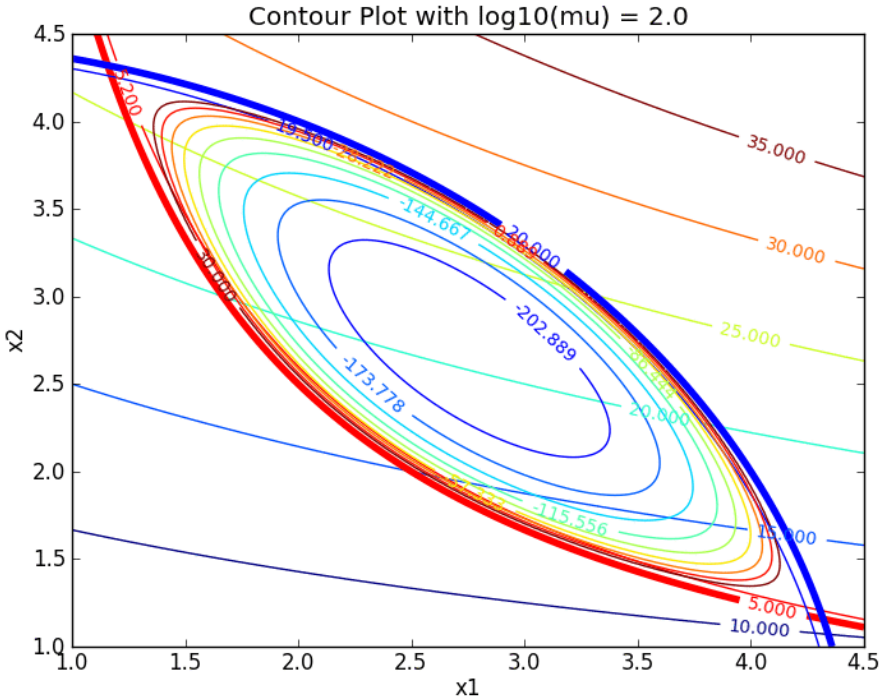
$$s.t. \quad \begin{aligned} x_1 x_2 &\geq 5 \\ x_1^2 + x_2^2 &\leq 20 \end{aligned}$$

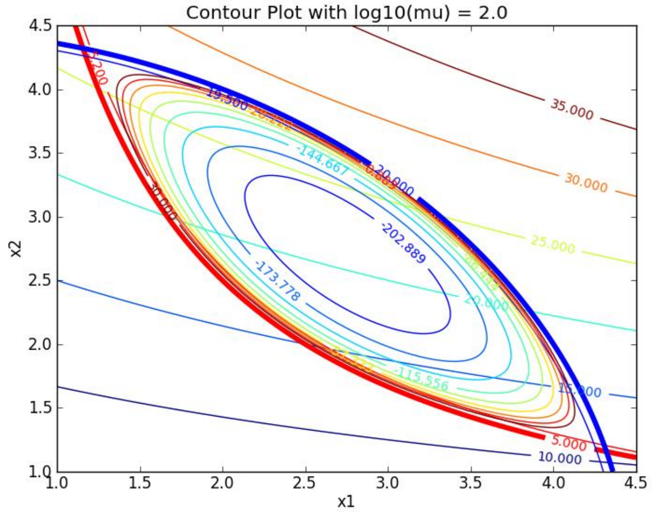
# Barrier Function: Example 4

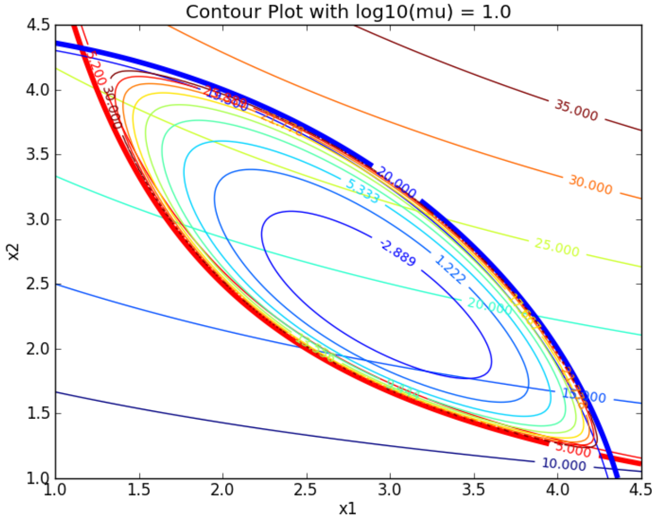
$$\min_{x \in R^2} \quad x_2(5 + x_1)$$

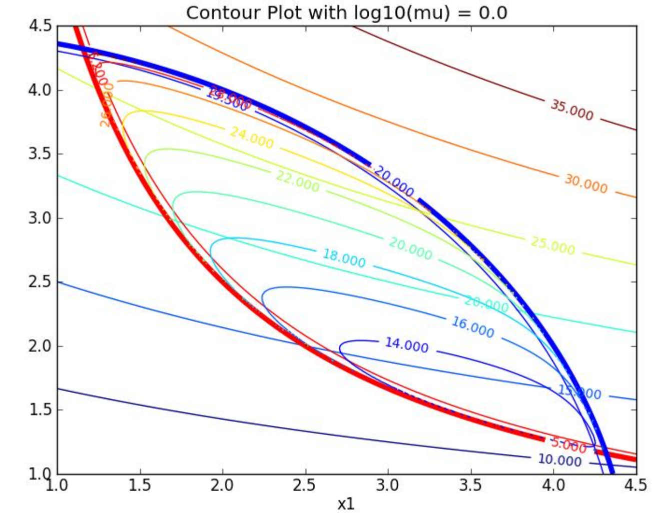
$$s.t. \quad x_1 x_2 \geq 5$$

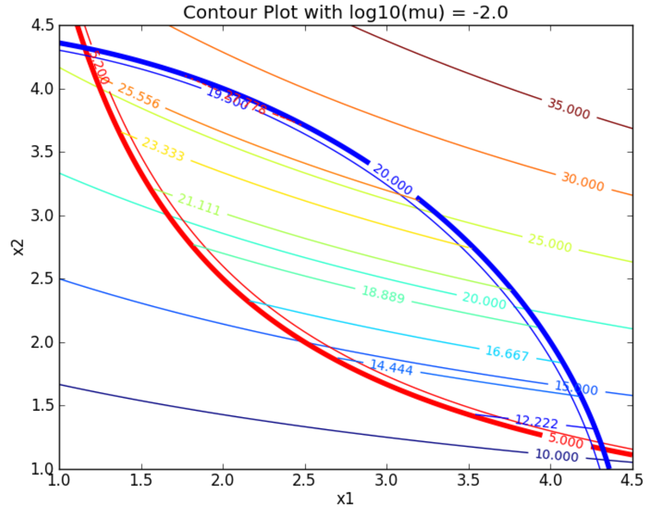
$$x_1^2 + x_2^2 \leq 20$$



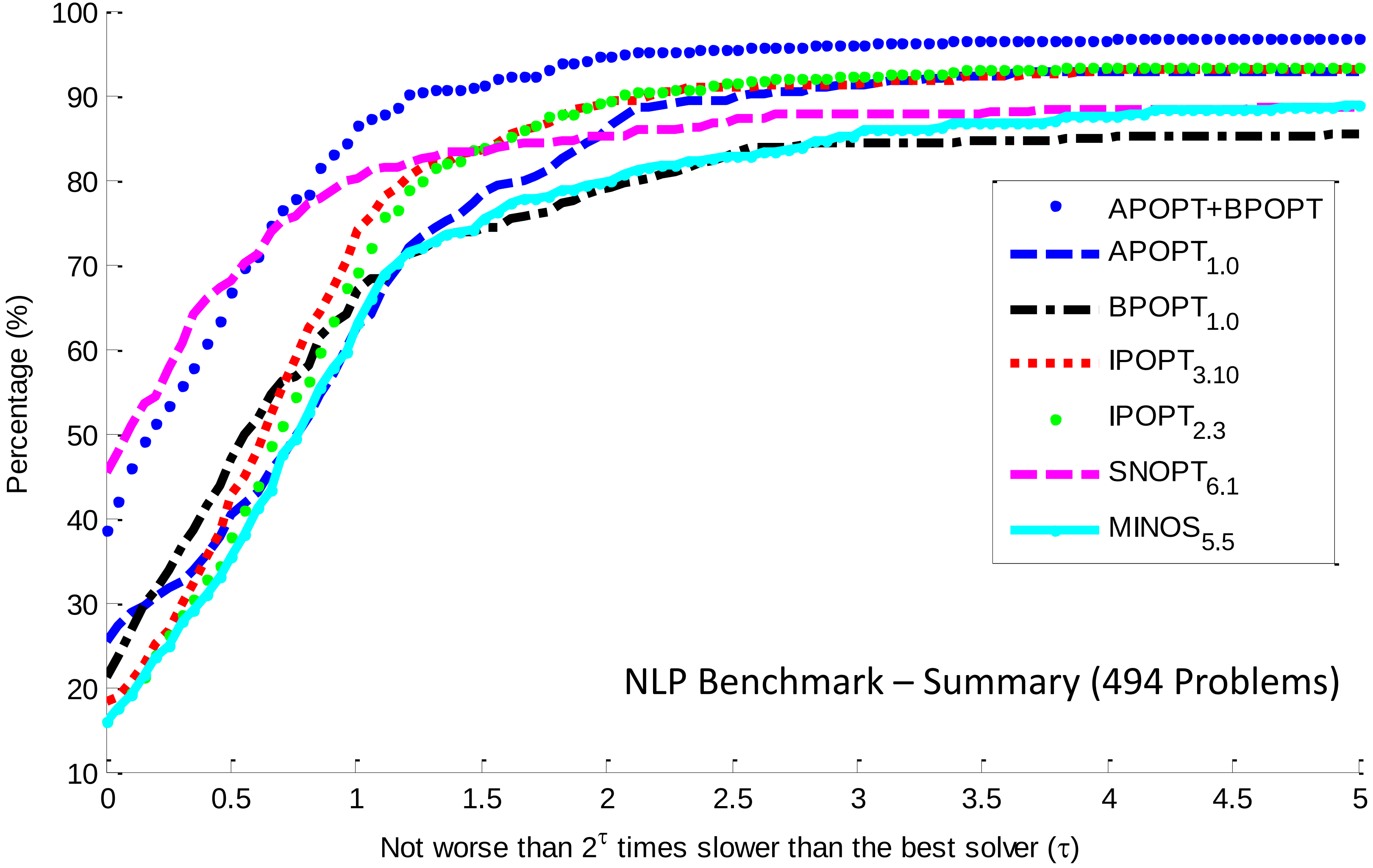



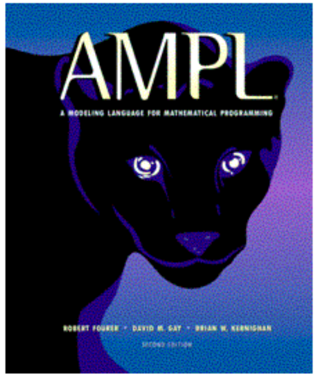



# Comparing Interior Point and Active Set Methods



# APOPT Solver

- AMPL / MATLAB / Python Interfaces from [http://apopt.com](http://apopt.com)




**AMPL®**

**A Modeling Language for Mathematical Programming**

**AMPL IDE Interface**
Version 1.0 available for download

**Webinar on AMPL Interfaces**
View the recording **made Tuesday, January 28**

APM Python Interface
Python gives users an open-source option for solving nonlinear programming problems with a growing community of users.

APM MATLAB Interface
MATLAB provides a powerful mathematical scripting language to improve the capability of optimization solutions.

# IPOPT Solver

- Download Source Code from https://projects.coin-or.org/Ipopt

## Welcome to the Ipopt home page

*Note that these project webpages are based on Wiki, which allows webusers to modify the content to correct typos, add information, or share their experience and tips with other users. You are welcome to contribute to these project webpages. To edit these pages or submit a ticket you must first register and login.*

### Introduction

Ipopt (**I**nterior **P**oint **OPT**imizer, pronounced eye-pea-Opt) is a software package for large-scale nonlinear optimization. It is designed to find (local) solutions of mathematical optimization problems of the from

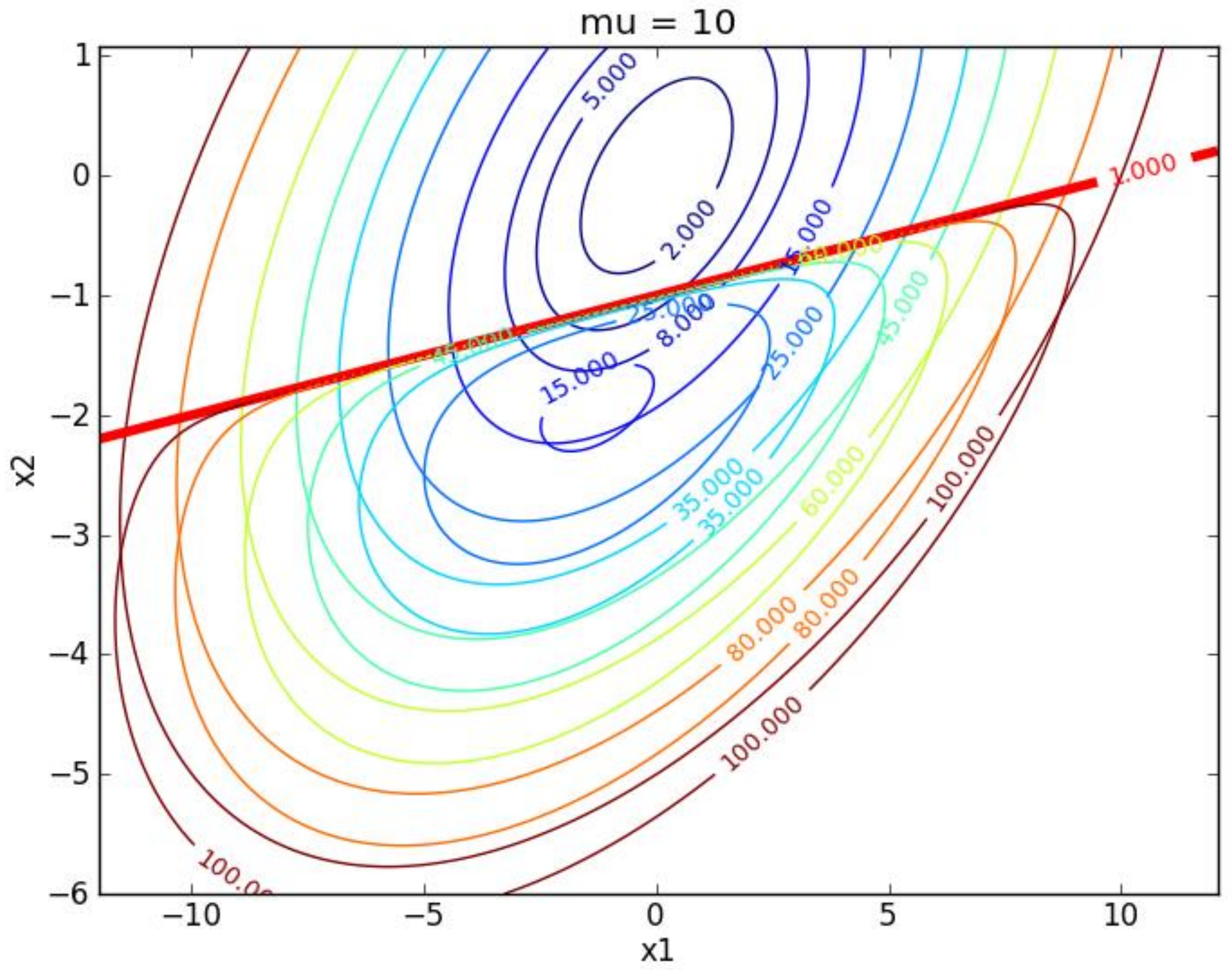
```
   min     f(x)
x in R^n

s.t.       g_L <= g(x) <= g_U
           x_L <=  x   <= x_U
```

where `f(x): R^n --> R` is the objective function, and `g(x): R^n --> R^m` are the constraint functions. The vectors `g_L` and `g_U` denote the lower and upper bounds on the constraints, and the vectors `x_L` and `x_U` are the bounds on the variables `x`. The functions `f(x)` and `g(x)` can be nonlinear and nonconvex, but should be twice continuously differentiable. Note that equality constraints can be formulated in the above formulation by setting the corresponding components of `g_L` and `g_U` to the same value.

Ipopt is part of the COIN-OR Initiative.

# Interior Point Homework Problem 1

$$\min_{x \in R^2} \quad x_1^2 - 2x_1x_2 + 4x_2^2$$
$$s.t. \quad 0.1x_1 - x_2 > 1$$



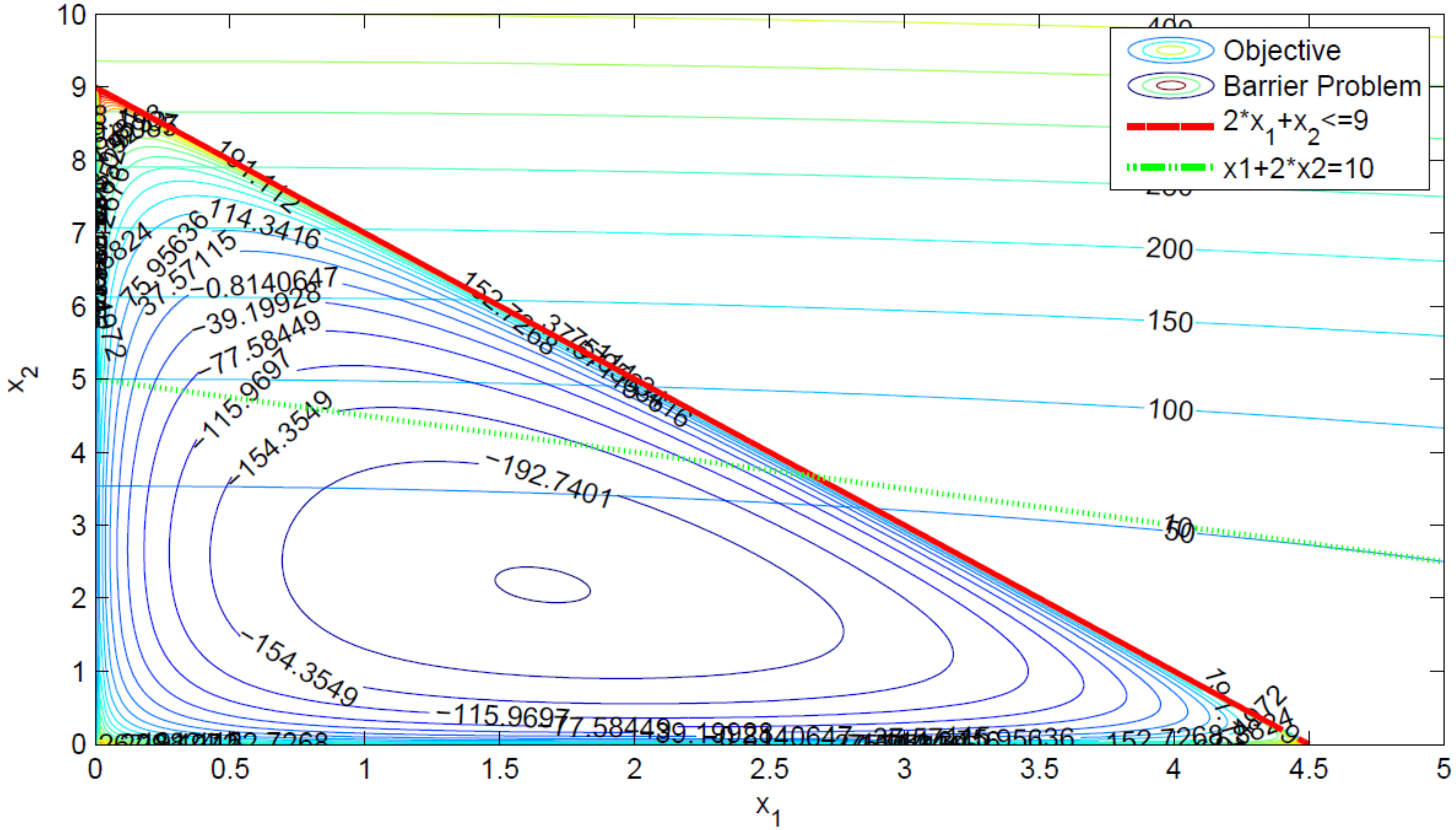
# Interior Point Homework Problem 2

$$\min_{x \in R^2} \quad x_1^2 + 2x_2^2$$

$$s.t. \quad 2x_1 + x_2 \leq 9$$

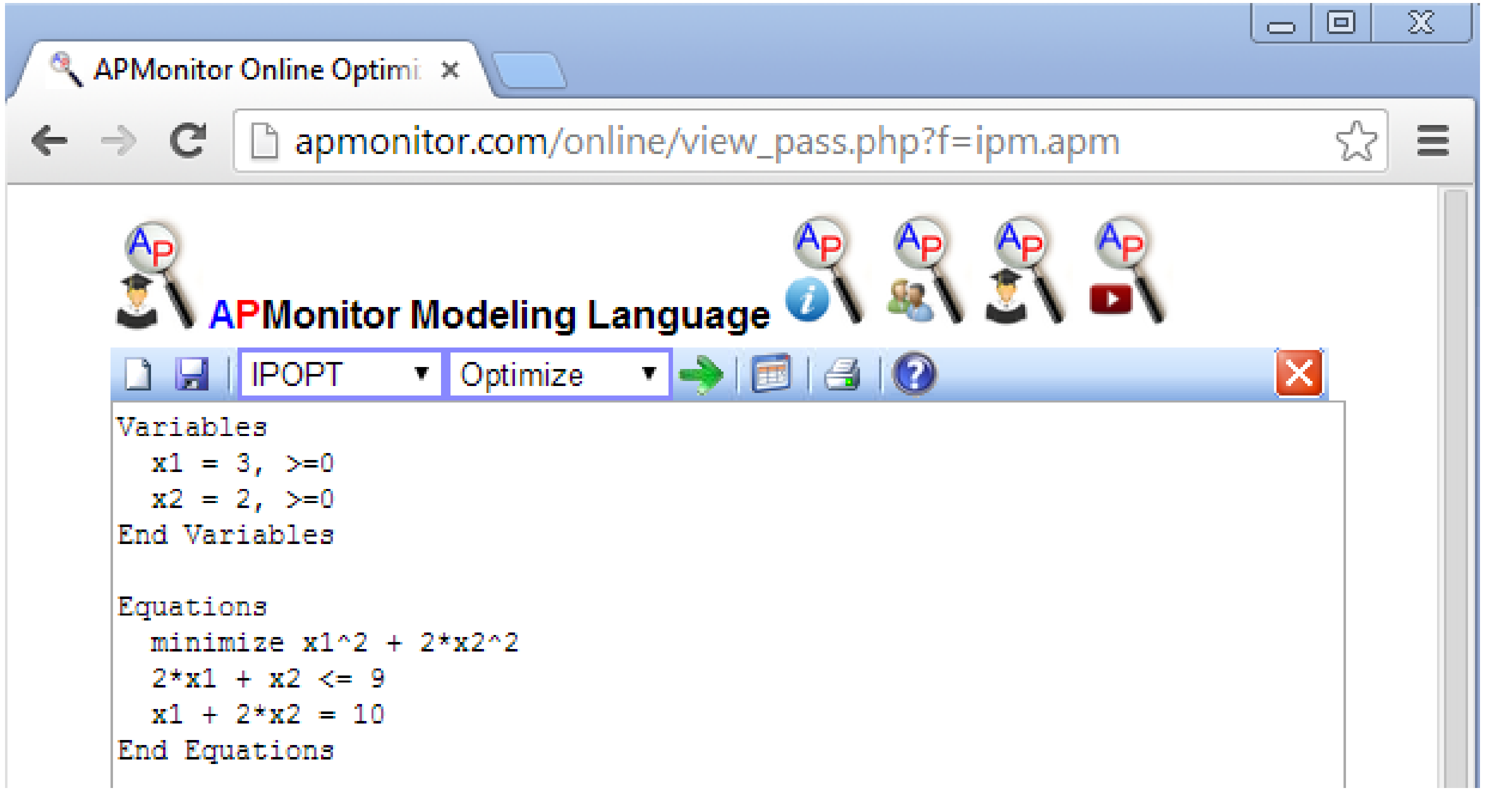
$$x_1 + 2x_2$$

$$x_1 > 0, x_2 > 0$$



# Homework Help: IPOPT Output

- Visit: http://apmonitor.com/online/view_pass.php?f=ipm.apm



# Homework Help: IPOPT Output

```
This is Ipopt version 3.10.1, running with linear solver ma27.

Number of nonzeros in equality constraint Jacobian...:        5
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        2

Total number of variables............................:        3
                     variables with only lower bounds:        3
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        2
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.7000000e+01 3.00e+00 2.11e+00   0.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  3.4257280e+01 9.99e-16 6.88e-01  -0.7 8.63e-01    -  5.90e-01 1.00e+00h  1
   2  3.4026513e+01 1.67e-16 1.65e-02  -1.9 1.58e-01    -  9.73e-01 1.00e+00f  1
   3  3.4000478e+01 4.51e-16 3.66e-04  -7.8 1.93e-02    -  9.85e-01 1.00e+00f  1
   4  3.4000000e+01 2.23e-15 4.71e-07  -9.6 3.59e-04    -  9.99e-01 1.00e+00f  1
```

# IPOPT Output Headings

- **iter**: The current iteration count.
- **objective**: The unscaled objective value at the current point.
- **inf_pr**: The unscaled max constraint violation at the current point.
- **inf_du**: The max scaled dual infeasibility at the current point.
- **lg(mu)**: log10 of the value of the barrier parameter m
- **||d||**: The infinity norm (max) of the primal step
- **lg(rg)**: log10 of the value of the regularization term for the Hessian of the Lagrangian. Dash (-) indicates that no regularization was done.
- **alpha_du**: The stepsize for the dual variables
- **alpha_pr**: The stepsize for the primal variables
- **ls**: The number of backtracking line search steps

# Homework Help: IPOPT Output

```
                                   (scaled)                 (unscaled)
Objective...............:   3.4000000286420644e+01    3.4000000286420644e+01
Dual infeasibility......:   4.7086480847724488e-07    4.7086480847724488e-07
Constraint violation....:   2.2339985764063819e-15    2.2339985764063819e-15
Complementarity.........:   2.9975411542695386e-07    2.9975411542695386e-07
Overall NLP error.......:   4.7086480847724488e-07    4.7086480847724488e-07


Number of objective function evaluations             = 5
Number of objective gradient evaluations             = 5
Number of equality constraint evaluations            = 5
Number of inequality constraint evaluations          = 0
Number of equality constraint Jacobian evaluations   = 5
Number of inequality constraint Jacobian evaluations = 0
Number of Lagrangian Hessian evaluations             = 4
Total CPU secs in IPOPT (w/o function evaluations)   =      0.001
Total CPU secs in NLP function evaluations           =      0.001

EXIT: Optimal Solution Found.

 The solution was found.

 The final value of the objective function is    34.000002864206
```

# Homework Help

- Download BPOPT Solver (MATLAB version) from Interior Point Page
  - http://apmonitor.com/me575/index.php/Main/InteriorPointMethod
  - Homework Problem 1 = BPOPT problem 10
  - Homework Problem 2 = BPOPT problem 6

```
% load bpopt libraries
addpath('bpopt')

% Solve problem 1
prob = bp_create(10);   % create problem
sol = bpopt(prob);      % solve problem
```