16.323 Principles of Optimal Control
Spring 2008

16.323 Lecture 2

Nonlinear Optimization

- Constrained nonlinear optimization
- Lagrange multipliers
- Penalty/barrier functions also often used, but will not be discussed here.

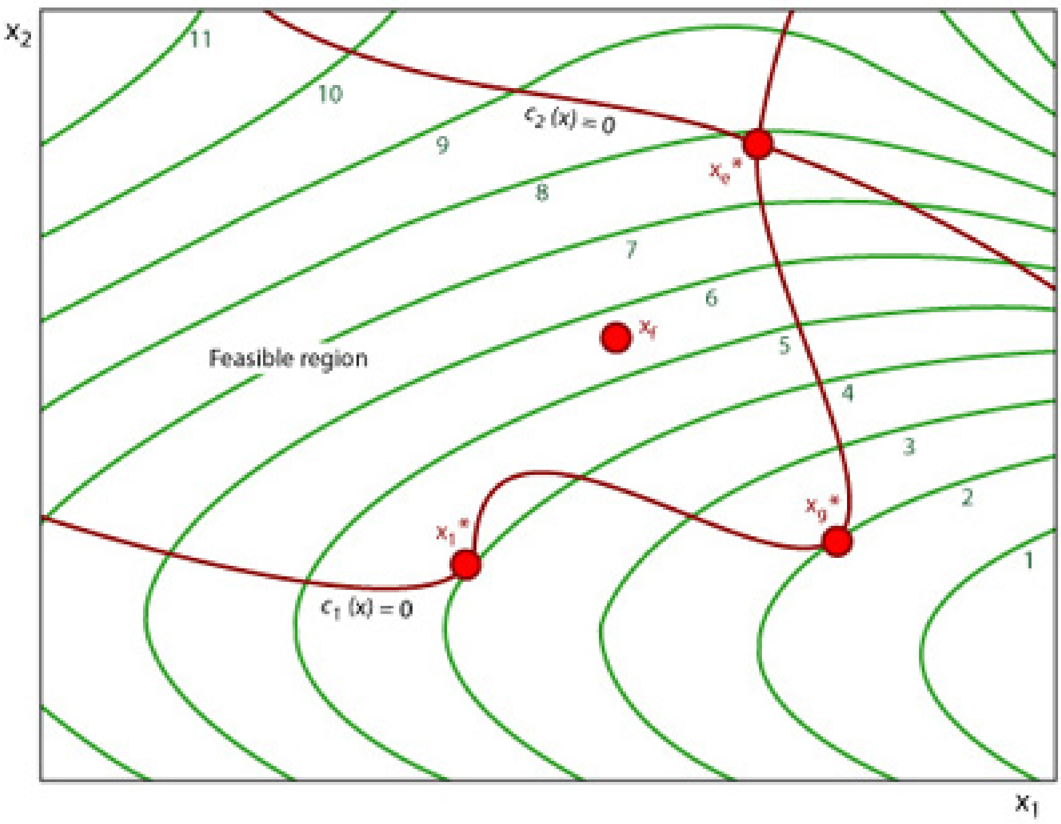


Figure by MIT OpenCourseWare.

Constrained Optimization

- Consider a problem with the next level of complexity: optimization with equality constraints

$$\min_{\mathbf{y}} F(\mathbf{y})$$
$$\text{such that } \mathbf{f}(\mathbf{y}) = 0$$

a vector of n constraints

- To simplify the notation, assume that the p-state vector $\mathbf{y}$ can be separated into a decision m-vector $\mathbf{u}$ and a state n-vector $\mathbf{x}$ related to the decision variables through the constraints. Problem now becomes:

$$\min_{\mathbf{u}} F(\mathbf{x}, \mathbf{u})$$
$$\text{such that } \mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$$

  – Assume that $p > n$ otherwise the problem is completely specified by the constraints (or over specified).

- One solution approach is **direct substitution**, which involves
  – Solving for $\mathbf{x}$ in terms of $\mathbf{u}$ using $\mathbf{f}$
  – Substituting this expression into F and solving for $\mathbf{u}$ using an unconstrained optimization.
  – Works best if $\mathbf{f}$ is linear (assumption is that not both of $\mathbf{f}$ and F are linear.)

- Example: minimize $F = x_1^2 + x_2^2$ subject to the constraint that $x_1 + x_2 + 2 = 0$
  - Clearly the unconstrained minimum is at $x_1 = x_2 = 0$
  - Substitution in this case gives equivalent problems:

$$\min_{x_2} \tilde{F}_2 = (-2 - x_2)^2 + x_2^2$$

  or

$$\min_{x_1} \tilde{F}_1 = x_1^2 + (-2 - x_1)^2$$

  for which the solution $(\partial \tilde{F}_2 / \partial x_2 = 0)$ is $x_1 = x_2 = -1$

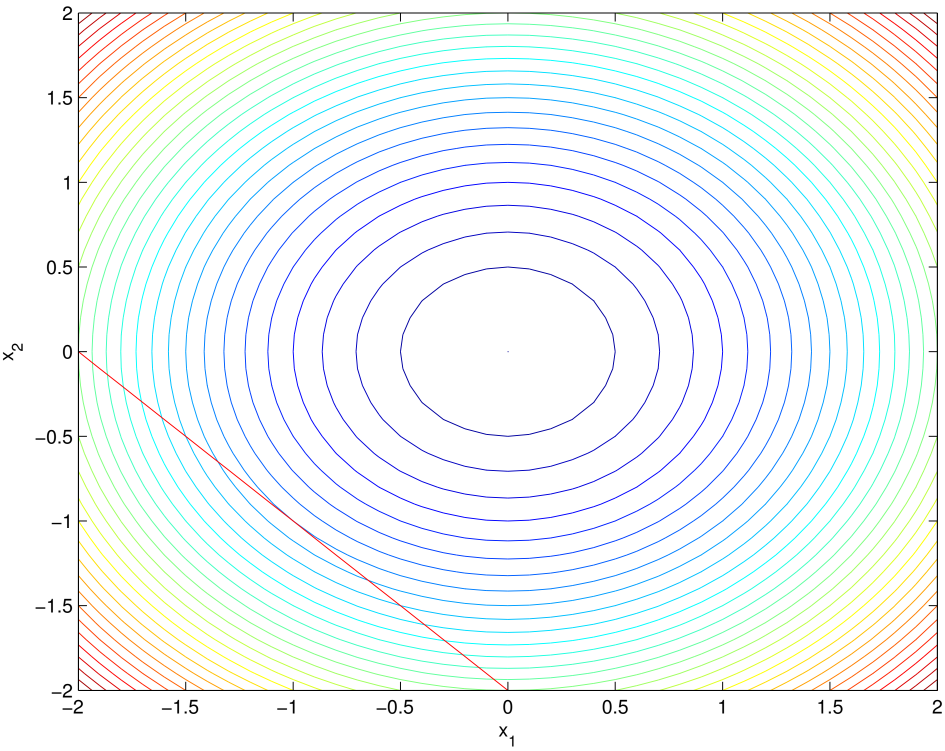


Figure 2.8: Simple function minimization with constraint.

- **Bottom line**: substitution works well for linear constraints, but process hard to generalize for larger systems/nonlinear constraints.

Lagrange Multipliers

- Need a more general strategy - using Lagrange multipliers.
- Since $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$, we can adjoin it to the cost with constants

$$\boldsymbol{\lambda}^T = \left[\begin{array}{ccc} \lambda_1 & \ldots & \lambda_n \end{array} \right]$$

  without changing the function value along the constraint to create **Lagrangian** function

$$L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = F(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u})$$

- Given values of $\mathbf{x}$ and $\mathbf{u}$ for which $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$, consider differential changes to the Lagrangian from differential changes to $\mathbf{x}$ and $\mathbf{u}$:

$$dL = \frac{\partial L}{\partial \mathbf{x}} d\mathbf{x} + \frac{\partial L}{\partial \mathbf{u}} d\mathbf{u}$$

  where $\frac{\partial L}{\partial \mathbf{u}} = \left[\begin{array}{ccc} \frac{\partial L}{\partial u_1} & \cdots & \frac{\partial L}{\partial u_m} \end{array} \right]$ (row vector)

- Since $\mathbf{u}$ are the decision variables it is convenient to choose $\boldsymbol{\lambda}$ so that

$$\frac{\partial L}{\partial \mathbf{x}} \triangleq \frac{\partial F}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \equiv 0 \tag{2.1}$$

$$\Rightarrow \boldsymbol{\lambda}^T = -\frac{\partial F}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \tag{2.2}$$

- To proceed, must determine what changes are possible to the **cost** keeping the equality constraint satisfied.
  - Changes to $\mathbf{x}$ and $\mathbf{u}$ are such that $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$, then

$$d\mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} d\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} d\mathbf{u} \equiv 0 \tag{2.3}$$

$$\Rightarrow d\mathbf{x} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} d\mathbf{u} \tag{2.4}$$

- Then the allowable cost variations are

$$
\begin{aligned}
dF &= \frac{\partial F}{\partial \mathbf{x}} d\mathbf{x} + \frac{\partial F}{\partial \mathbf{u}} d\mathbf{u} && (2.5)\\
&= \left(-\frac{\partial F}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial F}{\partial \mathbf{u}} \right) d\mathbf{u} \\
&= \left(\frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right) d\mathbf{u} && (2.6)\\
&\equiv \frac{\partial L}{\partial \mathbf{u}} d\mathbf{u} && (2.7)
\end{aligned}
$$

- So the gradient of the cost F with respect to $\mathbf{u}$ while keeping the constraint $\mathbf{f}(\mathbf{x}, \mathbf{u}) = 0$ is just

$$\frac{\partial L}{\partial \mathbf{u}}$$

and we need this gradient to be zero to have a stationary point so that $dF = 0 \ \forall \ d\mathbf{u} \neq 0$.

- Thus the necessary conditions for a stationary value of F are

$$
\begin{aligned}
\frac{\partial L}{\partial \mathbf{x}} &= 0 && (2.8)\\
\frac{\partial L}{\partial \mathbf{u}} &= 0 && (2.9)\\
\frac{\partial L}{\partial \boldsymbol{\lambda}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) = 0 && (2.10)
\end{aligned}
$$

which are $2n + m$ equations in $2n + m$ unknowns.

- Note that Eqs. 2.8–2.10 can be written compactly as

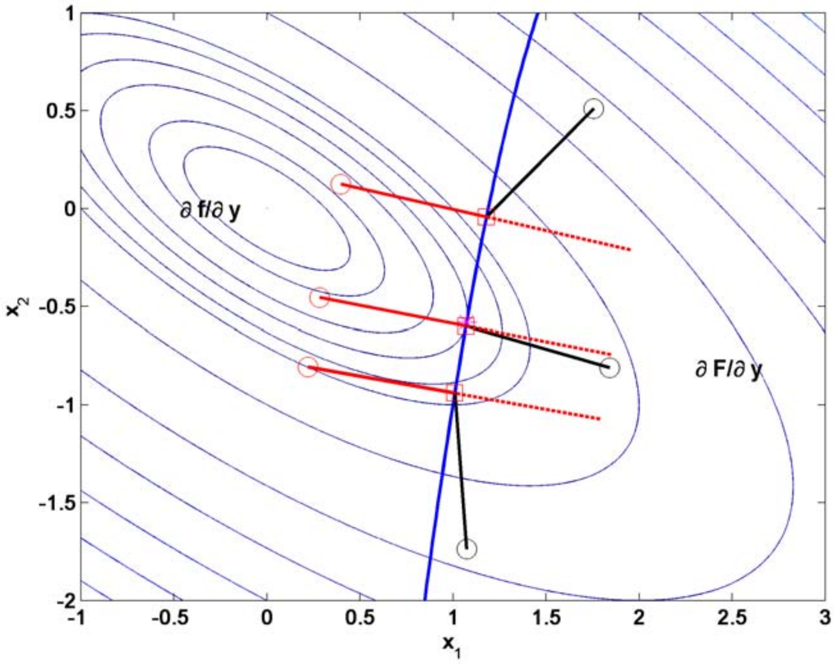
$$
\begin{aligned}
\frac{\partial L}{\partial \mathbf{y}} &= 0 && (2.11)\\
\frac{\partial L}{\partial \boldsymbol{\lambda}} &= 0 && (2.12)
\end{aligned}
$$

  – The solutions of which give the stationary points.

Intuition

- Can develop the intuition that the constrained solution will be a **point of tangency** of the constant cost curves and the constraint function
  – No further improvements possible while satisfying the constraints.

- Equivalent to saying that the gradient of the cost ftn (normal to the constant cost curve) $\partial F/\partial \mathbf{y}$ [black lines] must lie in the space spanned by the constraint gradients $\partial \mathbf{f}/\partial \mathbf{y}$ [red lines]
  – Means cost cannot be improved without violating the constraints.
  – In 2D case, this corresponds to $\partial F/\partial \mathbf{y}$ being collinear to $\partial f/\partial \mathbf{y}$



- **Note:** If this were not true, then it would be possible to take $d\mathbf{y}$ in the negative of the direction of the component of the cost gradient orthogonal to the constraint gradient, thereby reducing the cost and still satisfying the constraint.
  – Can see that at the points on the constraint above and blow the optimal value of x_2

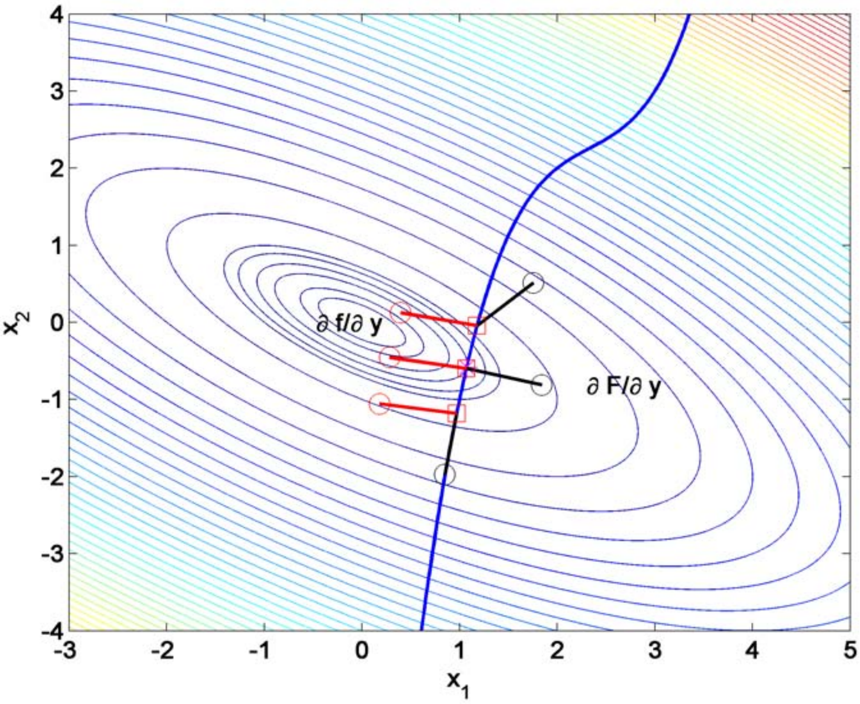


Figure 2.9: Minimization with equality constraints: shows that function and cost gradients are nearly collinear near optimal point and clearly not far away.

$$f(x_1, x_2) = x_2 - ((x_1)^3 - (x_1)^2 + (x_1) + 2) = 0 \text{ and } F = \frac{1}{2}\mathbf{x}^T \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \mathbf{x}$$

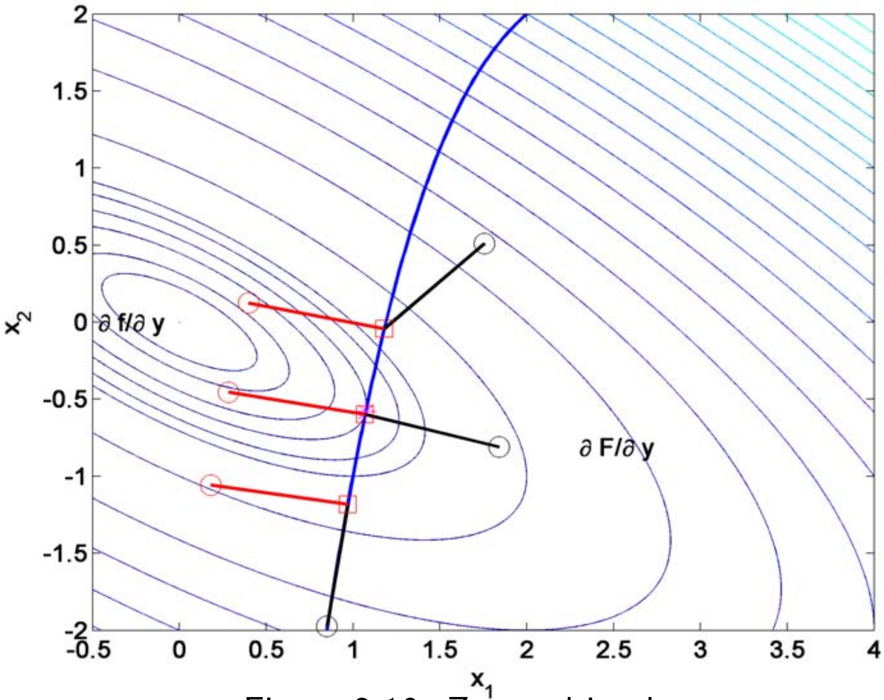
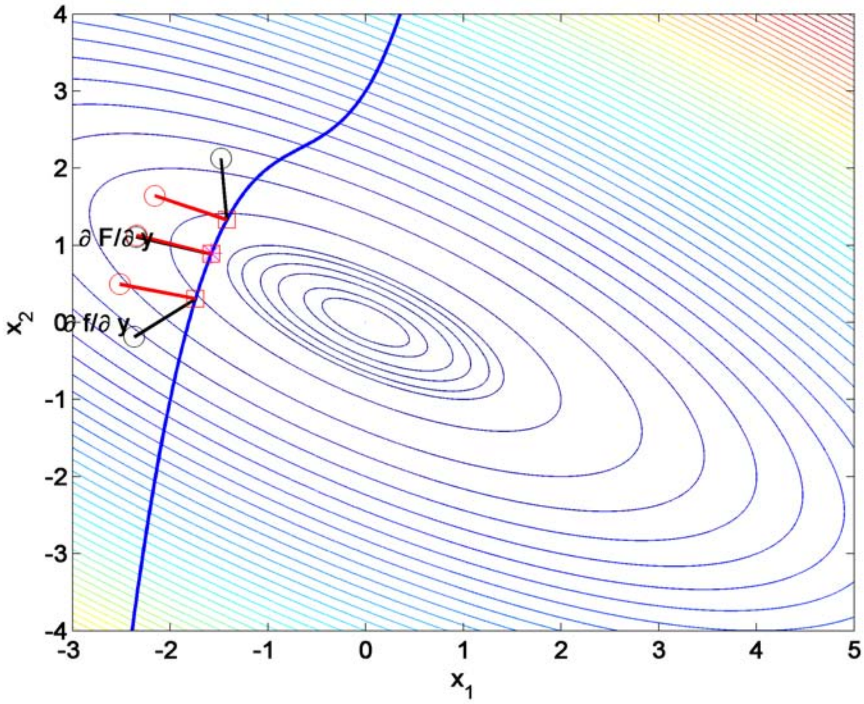


Figure 2.10: Zoomed in plot.



$f(x_1, x_2) = x_2 - ((x_1 - 2)^3 - (x_1 - 2)^2 + (x_1 - 2) + 2) = 0$ and $F = \frac{1}{2}\mathbf{x}^T \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \mathbf{x}$

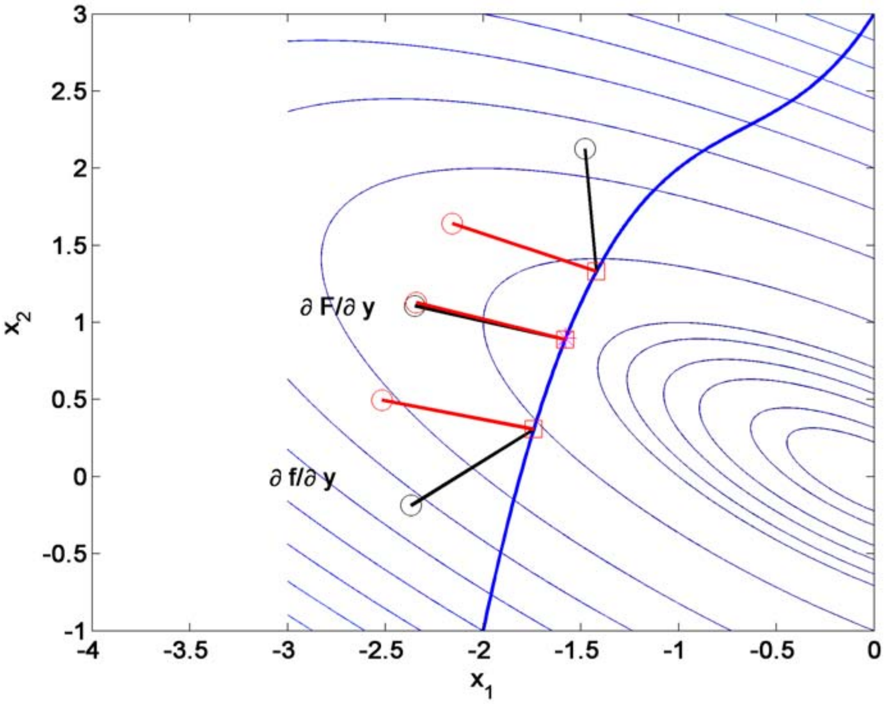


Figure 2.11: Change constraint - note that the cost and constraint gradients are collinear, but now aligned

- Generalize this intuition of being "collinear" to larger state dimensions to notion that the cost gradient **must lie in the space spanned** by the constraint gradients.
  - Equivalent to saying that it is possible to express the cost gradient as a linear combination of the constraint gradients
  - Again, if this was not the case, then improvements can be made to the cost without violating the constraints.

- So that at a constrained minimum, there must exist constants such that the cost gradient satisfies:

$$\frac{\partial F}{\partial \mathbf{y}} = -\lambda_1 \frac{\partial f_1}{\partial \mathbf{y}} - \lambda_2 \frac{\partial f_2}{\partial \mathbf{y}} - \cdots - \lambda_n \frac{\partial f_n}{\partial \mathbf{y}} \tag{2.13}$$

$$= -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \tag{2.14}$$

or equivalently that

$$\frac{\partial F}{\partial \mathbf{y}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}} = 0$$

which is, of course, the same as Eq. 2.11.

Constrained Example

- Minimize $F(x_1, x_2) = x_1^2 + x_2^2$ subject to $f(x_1, x_2) = x_1 + x_2 + 2 = 0$
  - Form the Lagrangian

$$L \triangleq F(x_1, x_2) + \lambda f(x_1, x_2) = x_1^2 + x_2^2 + \lambda(x_1 + x_2 + 2)$$

  - Where λ is the Lagrange multiplier

- The solution approach without constraints is to find the stationary point of $F(x_1, x_2)$ ($\partial F/\partial x_1 = \partial F/\partial x_2 = 0$)
  - With constraints we find the stationary points of L

$$\mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \frac{\partial L}{\partial \mathbf{y}} = 0, \quad \frac{\partial L}{\partial \lambda} = 0$$

which gives

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 2x_1 + \lambda = 0 \\ \frac{\partial L}{\partial x_2} &= 2x_2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= x_1 + x_2 + 2 = 0 \end{aligned}$$

- This gives 3 equations in 3 unknowns, solve to find $x_1^\star = x_2^\star = -1$

- The key point here is that due to the constraint, the selection of x_1 and x_2 during the minimization are not independent
  - The Lagrange multiplier captures this dependency.

- Difficulty can be solving the resulting equations for the optimal points (can be ugly nonlinear equations)

Inequality Constraints

- Now consider the problem

$$\min_{\mathbf{y}} F(\mathbf{y}) \tag{2.15}$$

$$\text{such that } \mathbf{f}(\mathbf{y}) \leq 0 \tag{2.16}$$

  – Assume that there are n constraints, but do not need to constrain n with respect to the state dimension p since not all inequality constraints will limit a degree of freedom of the solution.

- Have similar picture as before, but now not all constraints are active

  – Black line at top is inactive since $x_1 + x_2 - 1 < 0$ at the optimal value $\mathbf{x} = [1 \ \ -0.60] \Rightarrow$ it does not limit a degree of freedom in the problem.

  – Blue constraint is active, cost lower to the left, but $f_1 > 0$ there

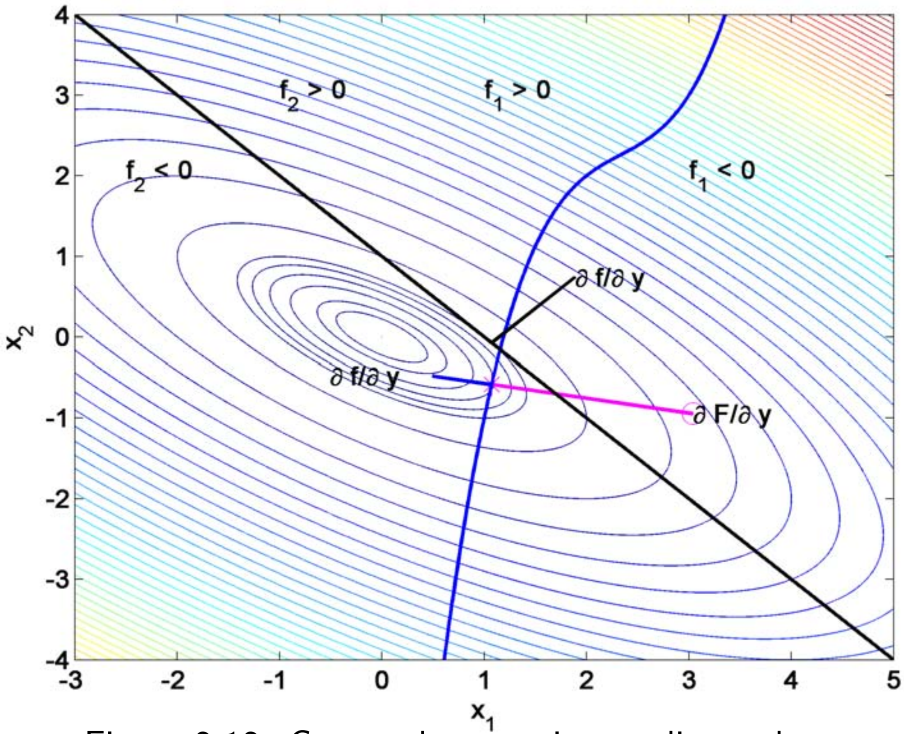
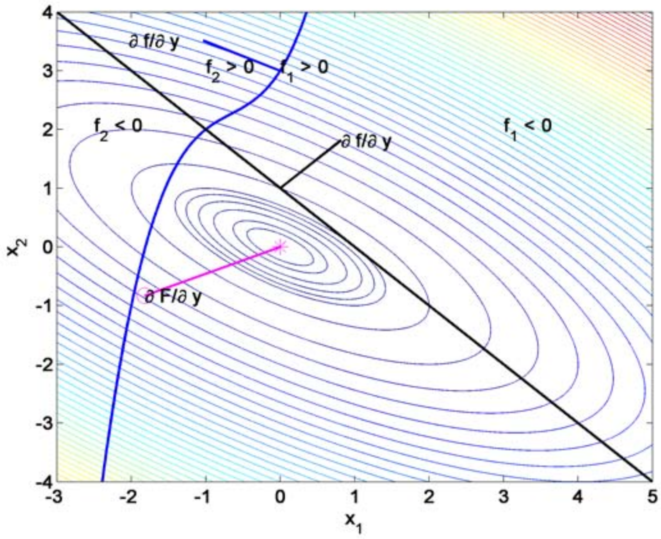
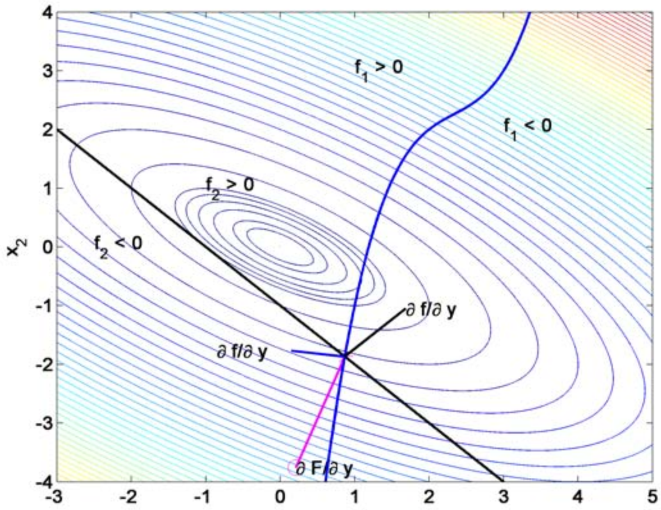


Figure 2.12: Cost and constraint gradients shown





With $x_1 + x_2 - 1 \le 0$, both constraints are active

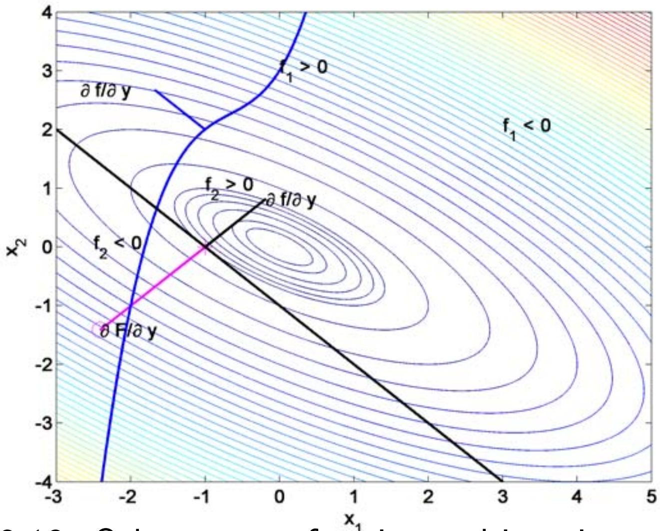


Figure 2.13: Other cases of active and inactive constraints

- Intuition in this case is that at the minimum, the cost gradient must lie in the space spanned by the **active** constraints - so split as:

$$\frac{\partial F}{\partial \mathbf{y}} = -\sum_{\substack{i \\ \text{active}}} \lambda_i \frac{\partial f_i}{\partial \mathbf{y}} - \sum_{\substack{j \\ \text{inactive}}} \lambda_j \frac{\partial f_j}{\partial \mathbf{y}} \tag{2.17}$$

  – And if the constraint is inactive, then can set $\lambda_j = 0$

- With equality constraints, needed the cost and function gradients to be collinear, but they could be in any orientation.

- For inequality constraints, need an additional constraint that is related to the allowable changes in the state.
  – Must restrict condition 2.17 so that the cost gradient points in the direction of the "allowable side" of the constraint ($f < 0$).
  $\Rightarrow$ Cost cannot be reduced without violating constraint.
  $\Rightarrow$ Cost and function gradients must point in opposite directions.
  – Given 2.17, require that $\lambda_i \geq 0$ for active constraints

- **Summary:** Active constraints, $\lambda_i \geq 0$, and Inactive ones $\lambda_j = 0$

- Given this, we can define the same Lagrangian as before $L = F + \boldsymbol{\lambda}^T \mathbf{f}$, and the necessary conditions for optimality are

$$\frac{\partial L}{\partial \mathbf{y}} = 0 \tag{2.18}$$

$$\lambda_i \frac{\partial L}{\partial \lambda_i} = 0 \ \forall i \tag{2.19}$$

  where the second property applies to all constraints

  – Active ones have $\lambda_i \geq 0$ and satisfy $\frac{\partial L}{\partial \lambda_i} = f_i = 0$

  – Inactive ones have $\lambda_i = 0$ and satisfy $\frac{\partial L}{\partial \lambda_i} = f_i < 0$.

- Equations 2.18 and 2.19 are the "essence" of the Kuhn-Tucker theorem in nonlinear programming - more precise statements available with more careful specification of the constraints properties.

  – Must also be careful in specifying the second order conditions for a stationary point to be a minimum - see Bryson and Ho, sections 1.3 and 1.7.

- Note that there is an implicit assumption here of **regularity** – that the active constraint gradients are linearly independent – for the $\lambda's$ to be well defined.

  – Avoids redundancy

Cost Sensitivity

- Often find that the constraints in the problem are picked somewhat arbitrarily - some flexibility in the limits.
  - Thus it would be good to establish the extent to which those choices impact the solution.

- Note that at the solution point,
$$\frac{\partial L}{\partial \mathbf{y}} = 0 \Rightarrow \frac{\partial F}{\partial \mathbf{y}} = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$$
If the state changes by $\Delta \mathbf{y}$, would expect change in the
$$\text{Cost} \qquad \Delta F = \frac{\partial F}{\partial \mathbf{y}} \Delta \mathbf{y}$$
$$\text{Constraint} \qquad \Delta \mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Delta \mathbf{y}$$
So then we have that
$$\Delta F = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \Delta \mathbf{y} = -\boldsymbol{\lambda}^T \Delta \mathbf{f}$$
$$\Rightarrow \qquad \frac{dF}{d\mathbf{f}} = -\boldsymbol{\lambda}^T$$
  - **Sensitivity of the cost to changes in the constraint function is given by the Lagrange Multipliers.**

- For active constraints $\boldsymbol{\lambda} \geq 0$, so expect that $dF/d\mathbf{f} \leq 0$
  - Makes sense because if it is active, then allowing $\mathbf{f}$ to increase will move the constraint boundary in the direction of reducing F
  - Correctly predicts that inactive constraints will not have an impact.

Alternative Derivation of Cost Sensitivity

- Revise the constraints so that they are of the form $\mathbf{f} \leq \mathbf{c}$, where $\mathbf{c} \geq 0$ is a constant that is nominally 0.
  - The constraints can be rewritten as $\bar{\mathbf{f}} = \mathbf{f} - \mathbf{c} \leq 0$, which means

$$\frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{y}} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$$

    and assuming the $\bar{\mathbf{f}}$ constraint remains active as we change $\mathbf{c}$

$$\frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{c}} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{c}} - I = 0$$

- Note that at the solution point,

$$\frac{\partial L}{\partial \mathbf{y}} = 0 \Rightarrow \frac{\partial F}{\partial \mathbf{y}} = -\boldsymbol{\lambda}^T \frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{y}} = -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$$

- To study cost sensitivity, must compute $\frac{\partial F}{\partial \mathbf{c}}$. To proceed, note that

$$\begin{aligned}
\frac{\partial F}{\partial \mathbf{c}} &= \frac{\partial F}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{c}} \\
&= -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{c}} \\
&= -\boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{c}} \\
&= -\boldsymbol{\lambda}^T
\end{aligned}$$

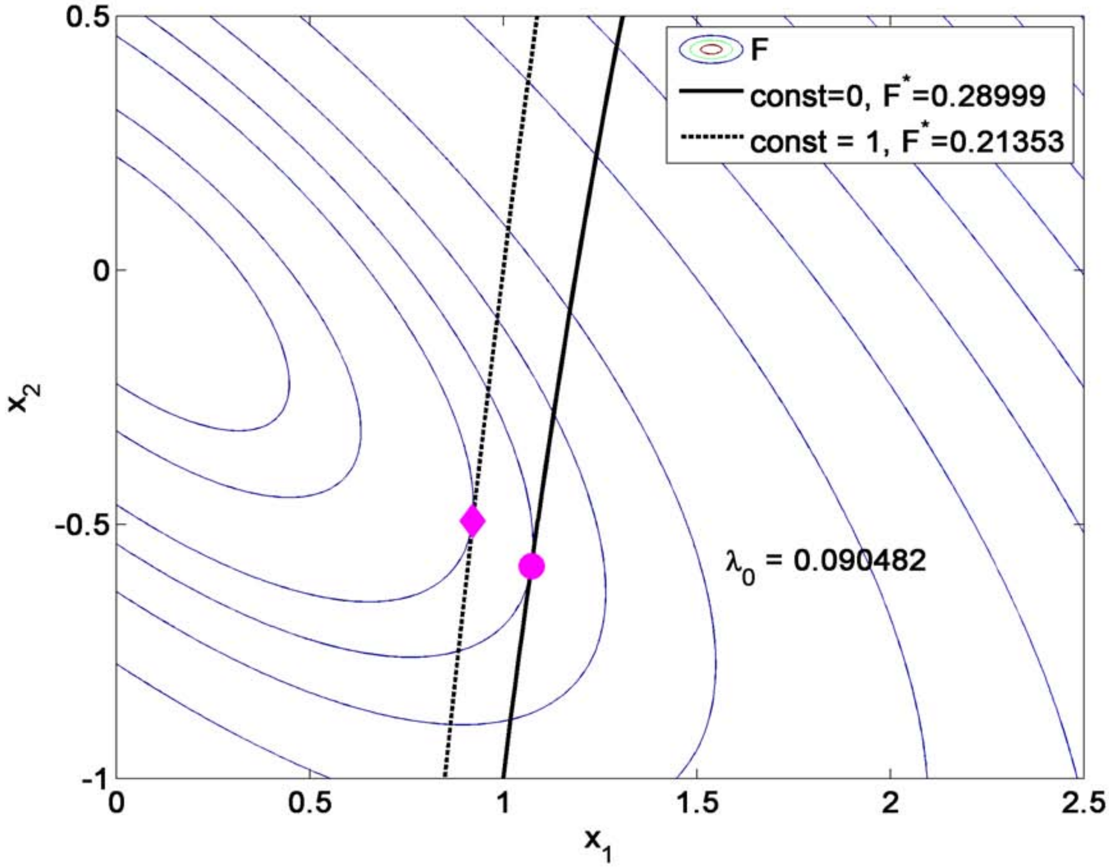


Figure 2.14: Shows that changes to the constraint impact cost in a way that can be predicted from the Lagrange Multiplier.

Simple Constrained Example

- Consider case $F = x_1^2 + x_1x_2 + x_2^2$ and $x_2 \geq 1$, $x_1 + x_2 \leq 3$
- Form Lagrangian

$$L = x_1^2 + x_1x_2 + x_2^2 + \lambda_1(1 - x_2) + \lambda_2(x_1 + x_2 - 3)$$

- Form necessary conditions:

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 2x_1 + x_2 + \lambda_2 = 0 \\ \frac{\partial L}{\partial x_2} &= x_1 + 2x_2 - \lambda_1 + \lambda_2 = 0 \\ \lambda_1 \frac{\partial L}{\partial \lambda_1} &= \lambda_1(1 - x_2) = 0 \\ \lambda_2 \frac{\partial L}{\partial \lambda_2} &= \lambda_2(x_1 + x_2 - 3) = 0 \end{aligned}$$

- Now consider the various options:
  - Assume $\lambda_1 = \lambda_2 = 0$ both inactive

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 2x_1 + x_2 = 0 \\ \frac{\partial L}{\partial x_2} &= x_1 + 2x_2 = 0 \end{aligned}$$

    gives solution $x_1 = x_2 = 0$ as expected, but does not satisfy all the constraints
  - Assume $\lambda_1 = 0$ (inactive), $\lambda_2 \geq 0$ (active)

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= 2x_1 + x_2 + \lambda_2 = 0 \\ \frac{\partial L}{\partial x_2} &= x_1 + 2x_2 + \lambda_2 = 0 \\ \lambda_2 \frac{\partial L}{\partial \lambda_2} &= \lambda_2(x_1 + x_2 - 3) = 0 \end{aligned}$$

    which gives solution $x_1 = x_2 = 3/2$, which satisfies the constraints, but $F = 6.75$ and $\lambda_2 = -9/2$

– Assume $\lambda_1 \geq 0$ (active), $\lambda_2 = 0$ (inactive)

$$
\begin{aligned}
\frac{\partial L}{\partial x_1} &= 2x_1 + x_2 = 0 \\
\frac{\partial L}{\partial x_2} &= x_1 + 2x_2 - \lambda_1 = 0 \\
\lambda_1 \frac{\partial L}{\partial \lambda_1} &= \lambda_1(1 - x_2) = 0
\end{aligned}
$$

gives solution $x_1 = -1/2$, $x_2 = 1$, $\lambda_1 = 3/2$ which satisfies the constraints, and $F = 0.75$

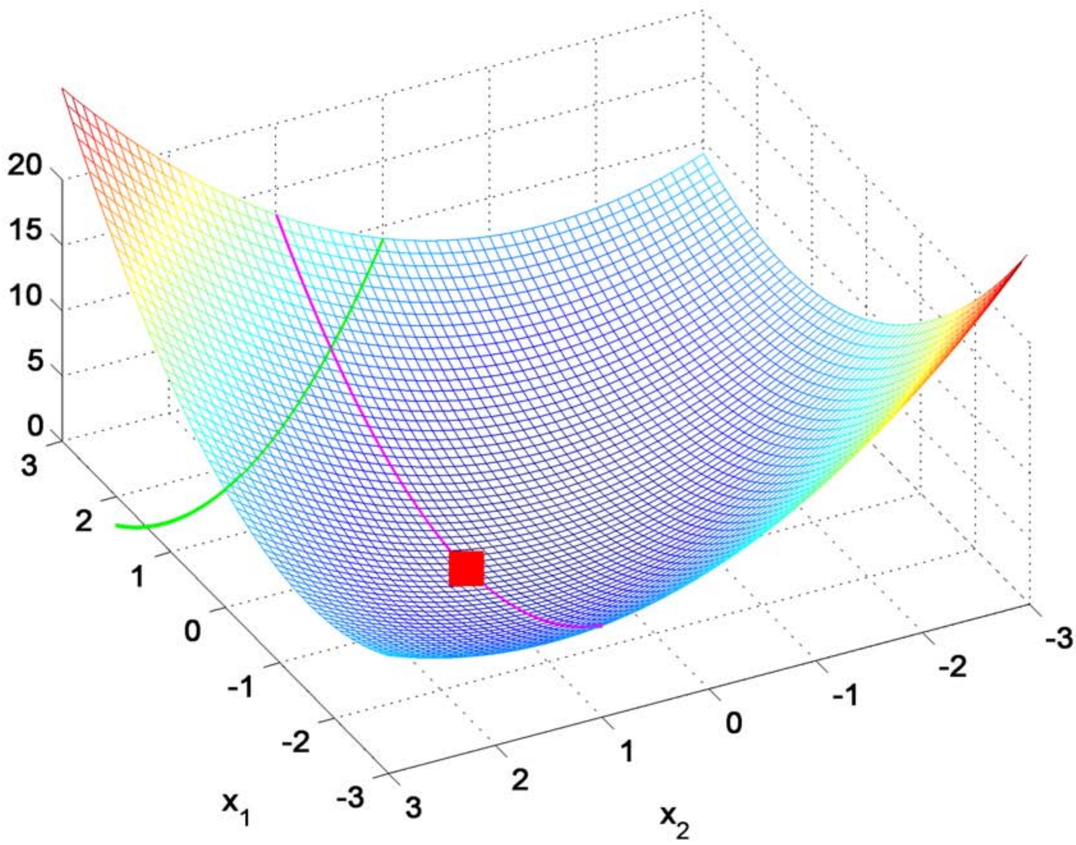


Figure 2.15: Simple example

Code to generate Figure 2.12

```
%
% 16.323 Spr 2008
% Plot of cost ftns and constraints

clear all;close all;
set(0, 'DefaultAxesFontSize', 14, 'DefaultAxesFontWeight','demi')
set(0, 'DefaultTextFontSize', 14, 'DefaultTextFontWeight','demi')

global g G f

F=[];g=[0;0];G=[1 1;1 2];

testcase=0
if testcase
    f=inline('(1*(x1+1).^3-1*(x1+1).^2+1*(x1+1)+2)');
    dfdx=inline('(3*1*(x1+1).^2-2*1*(x1+1)+1)');
else
    f=inline('(1*(x1-2).^3-1*(x1-2).^2+1*(x1-2)+2)');
    dfdx=inline('(3*1*(x1-2).^2-2*1*(x1-2)+1)');
end

x1=-3:.01:5;x2=-4:.01:4;
for ii=1:length(x1);
    for jj=1:length(x2);
        X=[x1(ii) x2(jj)]';
        F(ii,jj)=g'*X+X'*G*X/2;
    end;
end;
figure(1);clf
contour(x1,x2,F',[min(min(F)) .05 .1 .2 .29 .4 .5 1:1:max(max(F))]);
xlabel('x_1' )
ylabel('x_2' )
hold on;
plot(x1,f(x1),'LineWidth',2);

% X=FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB,NONLCON,OPTIONS)
xx=fmincon('meshf',[0;0],[],[],[],[],[],[],'meshc');
hold on
plot(xx(1),xx(2),'m*','MarkerSize',12)
axis([-3 5 -4 4]);

Jx=[];
[kk,II1]=min(abs(x1-xx(1)))
[kk,II2]=min(abs(x1-1.1*xx(1)))
[kk,II3]=min(abs(x1-0.9*xx(1)))
ll=[II1 II2 II3];
gam=.8; % line scaling
for ii=1:length(ll)
    X=[x1(ll(ii));f(x1(ll(ii)))]
    Jx(ii,:)=(g+G*X)';
    X2=X+Jx(ii,:)'*gam/norm(Jx(ii,:));

    Nx1=X(1);
    df=[-dfdx(Nx1);1];     % x_2=f(x_1) ==> x_2 - f(x_1) < =0

    X3=[Nx1;f(Nx1)];
    X4=X3+df*gam/norm(df);

    plot(X2(1),X2(2),'ko','MarkerSize',12)
    plot(X(1),X(2),'ks','MarkerSize',12)
    plot([X(1);X2(1)],[X(2);X2(2)],'k-','LineWidth',2)
    plot(X4(1),X4(2),'ro','MarkerSize',12)
    plot(X3(1),X3(2),'rs','MarkerSize',12)
    plot([X4(1);X3(1)],[X4(2);X3(2)],'r-','LineWidth',2)
    if ii==1;
        text([1.25*X2(1)],[X2(2)],'\partial F/\partial y' )
```

```
        text([X4(1)-.75],[0*X4(2)],'\partial f/\partial y' )
    end
end
hold off

%%%%%%%%%%%%%%%%%%%%%%%%

f2=inline('-1*x1-1');global f2
df2dx=inline('-1*ones(size(x))');

figure(3);gam=2;
contour(x1,x2,F',[min(min(F)) .05 .1 .2 .3 .4 .5 1:1:max(max(F))]);
xlabel('x_1' );ylabel('x_2'  )

xx=fmincon('meshf',[0;0],[],[],[],[],[],[],'meshc2');
hold on
Jx=(g+G*xx)';
X2=xx+Jx'*gam/norm(Jx);
plot(xx(1),xx(2),'m*','MarkerSize',12)
plot(X2(1),X2(2),'mo','MarkerSize',12);
plot([xx(1);X2(1)],[xx(2);X2(2)],'m-','LineWidth',2)
text([X2(1)],[X2(2)],'\partial F/\partial y')
hold off

hold on;
plot(x1,f(x1),'LineWidth',2);
text(-1,1,'f_2 > 0')
text(-2.5,0,'f_2 < 0')
plot(x1,f2(x1),'k-','LineWidth',2);
text(3,2,'f_1 < 0')
if testcase
    text(0,3,'f_1 > 0')
else
    text(1,3,'f_1 > 0')
end

dd=[xx(1) 0 xx(1)]';
X=[dd f(dd)];
df=[-dfdx(dd) 1*ones(size(dd))];
X2=X+gam*df/norm(df);
for ii=3
    plot([X(ii,1);X2(ii,1)],[X(ii,2);X2(ii,2)],'LineWidth',2)
    text([X2(ii,1)-1],[X2(ii,2)],'\partial f/\partial y')
end
X=[dd f2(dd)];
df2=[-df2dx(dd) 1*ones(size(dd))];
X2=X+gam*df2/norm(df2);
%for ii=1:length(X)
for ii=1
    plot([X(ii,1);X2(ii,1)],[X(ii,2);X2(ii,2)],'k','LineWidth',2)
    text([X2(ii,1)],[X2(ii,2)],'\partial f/\partial y')
end
hold off

%%%%%%%%%%%%%%%%%%%%%%%

f2=inline('-1*x1+1');global f2
df2dx=inline('-1*ones(size(x))');

figure(4);clf;gam=2;
contour(x1,x2,F',[min(min(F)) .05 .1 .2 .3 .4 .5 1:1:max(max(F))]);
xlabel('x_1');ylabel('x_2')

xx=fmincon('meshf',[1;-1],[],[],[],[],[],[],'meshc2');
hold on
Jx=(g+G*xx)';
X2=xx+Jx'*gam/norm(Jx);
plot(xx(1),xx(2),'m*','MarkerSize',12)
plot(X2(1),X2(2),'mo','MarkerSize',12);
plot([xx(1);X2(1)],[xx(2);X2(2)],'m-','LineWidth',2)
text([X2(1)],[X2(2)],'\partial F/\partial y')
hold off
```

```
hold on;
plot(x1,f(x1),'LineWidth',2);
text(-1,3,'f_2 > 0')
text(-2.5,2,'f_2 < 0')
plot(x1,f2(x1),'k-','LineWidth',2);
text(3,2,'f_1 < 0')
if testcase
    text(0,3,'f_1 > 0')
else
    text(1,3,'f_1 > 0')
end

dd=[xx(1) 0 xx(1)]';
X=[dd f(dd)];
df=[-dfdx(dd) 1*ones(size(dd))];
X2=X+gam*df/norm(df);
for ii=3
    plot([X(ii,1);X2(ii,1)],[X(ii,2);X2(ii,2)],'LineWidth',2)
    text([X2(ii,1)-1],[X2(ii,2)],'\partial f/\partial y')
end
X=[dd f2(dd)];
df2=[-df2dx(dd) 1*ones(size(dd))];
X2=X+gam*df2/norm(df2);
%for ii=1:length(X)
for ii=1
    plot([X(ii,1);X2(ii,1)],[X(ii,2);X2(ii,2)],'k','LineWidth',2)
    text([X2(ii,1)],[X2(ii,2)],'\partial f/\partial y')
end
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%

if testcase
    figure(1)
    print  -r300 -dpng mesh1b.png;%jpdf('mesh1b');
    axis([-4 0 -1 3]);
    print  -r300 -dpng mesh1c.png;%jpdf('mesh1c');
    figure(3)
    print  -r300 -dpng mesh2.png;%jpdf('mesh2');
    figure(4)
    print  -r300 -dpng mesh2a.png;%jpdf('mesh2a');
else
    figure(1)
    print  -r300 -dpng mesh1.png;%jpdf('mesh1');
    axis([-.5 4 -2 2]);
    print  -r300 -dpng mesh1a.png;%jpdf('mesh1a');
    figure(3)
    print  -r300 -dpng mesh4.png;%jpdf('mesh4');
    figure(4)
    print  -r300 -dpng mesh4a.png;%jpdf('mesh4a');
end

%
% sensitivity study
% line given by x_2=f(x_1), and the constraint is that x_2-f(x_1) <= 0
% changes are made to the constraint so that x_2-f(x_1) <= alp > 0
figure(5);clf
contour(x1,x2,F',[min(min(F)) .05 .1  .213 .29 .4 .6:.5:max(max(F))]);
xlabel('x_1')
ylabel('x_2')
hold on;
f=inline('(1*(x1-2).^3-1*(x1-2).^2+1*(x1-2)+2)');
dfdx=inline('(3*1*(x1-2).^2-2*1*(x1-2)+1)');
plot(x1,f(x1),'k-','LineWidth',2);
alp=1;
plot(x1,f(x1)+alp,'k--','LineWidth',2);

global alp
[xx1,temp,temp,temp,lam1]=fmincon('meshf',[0;0],[],[],[],[],[],[],'meshc3');
alp=0;
[xx0,temp,temp,temp,lam0]=fmincon('meshf',[0;0],[],[],[],[],[],[],'meshc3');
```

```
[meshf(xx0) lam0.ineqnonlin;meshf(xx1) lam1.ineqnonlin]

legend('F',['const=0, F^*=',num2str(meshf(xx0))],['const = 1, F^*=' ,num2str(meshf(xx1))])

hold on
plot(xx0(1),xx0(2),'mo','MarkerSize',12,'MarkerFaceColor','m')
plot(xx1(1),xx1(2),'md','MarkerSize',12,'MarkerFaceColor','m')

text(xx0(1)+.5,xx0(2),['\lambda_0 = ',num2str(lam0.ineqnonlin)])

axis([0 2.5 -1 .5])
print  -r300 -dpng mesh5;%jpdf('mesh5');
```

```
function F=meshf(X);

global g G

F=g'*X+X'*G*X/2;

end
```

```
function [c,ceq]=meshc(X);

global f

c=[];
%ceq=f(X(1))-X(2);
ceq=X(2)-f(X(1));

return
```

```
function [c,ceq]=meshc(X);

global f f2

%c=[f(X(1))-X(2);f2(X(1))-X(2)];
c=[X(2)-f(X(1));X(2)-f2(X(1))];
ceq=[];

return
```

Code for Simple Constrained Example

```
figure(1),clf
xx=[-3:.1:3]'; for ii=1:length(xx);for jj=1:length(xx); %
FF(ii,jj)= xx(ii)^2+xx(ii)*xx(jj)+xx(jj)^2;end;end;%
hh=mesh(xx,xx,FF);%
hold on;%

plot3(xx,ones(size(xx)),xx.^2+1+xx,'m-','LineWidth',2);%
plot3(xx,3-xx,xx.^2+(3-xx).^2+xx.*(3-xx),'g-','LineWidth',2);%

xlabel('x_1'); ylabel('x_2'); %
hold off; axis([-3 3 -3 3 0 20])%
hh=get(gcf,'children');%
set(hh,'View',[-109 74],'CameraPosition',[-26.5555 13.5307 151.881]);%

xx=fmincon('simplecaseF',[0;0],[],[],[],[],[],[],'simplecaseC');
hold on
plot3(xx(1),xx(2),xx(1).^2+xx(2).^2+xx(1).*xx(2),'rs','MarkerSize',20,'MarkerFace','r')
xx(1).^2+xx(2).^2+xx(1).*xx(2)

print -r300 -dpng simplecase.png;

```

```
function F=simplecaseF(X);

F=X(1)^2+X(1)*X(2)+X(2)^2;

return
```

```
function [c,ceq]=simplecaseC(X);

c=[1-X(2);X(1)+X(2)-3];
ceq=0;

return
```