The Matlab Optimization Toolbox may be used to solve linear and nonlinear optimization problems that come in two flavors: constrained and unconstrained. In this class, we will principally deal with constrained optimization problems. For this reason, this tutorial will focus on the use of the \texttt{fmincon} command and the general structure of the constrained optimization problem. Refer to the Optimization Toolbox Manual in the Matlab Help Desk for information on other optimization programs.

1. **Program Structure**

For a constrained optimization problem, the complete Matlab code consists of three \texttt{.m} files: 1) the main program, 2) the objective function, and 3) the constraint function.

1.1 **Main program**

The functions of the main program are: 1) to initialize the independent variables in the optimization problem, 2) to set the optimization options, and 3) to call the optimization function, \texttt{fmincon}, 4) to process the output of the optimization analysis (ie. graph the results). The initialization of the independent variables is an important step, especially in nonlinear optimization problems where local minima/maxima may exist. For nonlinear problems, it is always a good idea to run your optimization code a few times, each with a different set of initial conditions. Note whether your output values are dependent on the initial conditions and for goodness sake, MAKE SURE THAT YOUR ANSWERS MAKE SENSE (no negative length values, please).

The \texttt{options} expression in Matlab allows you a great deal of control over the specifics of the optimization process (e.g. number of iterations in the analysis, solution tolerance, etc.). For the purposes of this class, we will use the general options statement given below. If you’re curious about other parameters in the options structure, type \texttt{help optimset}.

```matlab
options = optimset('Largescale','off');
```

The bread and butter of the constrained optimization program is the \texttt{fmincon} call statement. There are quite a few components of this statement, so we’ll examine it with an example:

```matlab
[x,fval] = fmincon('profit',x0,[],[],[],[],[],[],'confun',options)
```

This statement determines the value of the x \texttt{vector} that minimizes the variable, fval. The objective function is ‘profit’, and the ‘confun’ function contains the constraint equations. The initial guesses for x are contained in the x0 vector. The six sets of brackets [] are simply placeholders for parameters that we are not specifying during the optimization.
analysis. Matlab assumes default values for these parameters if they are not specified. Type \texttt{help fmincon} for more info on this.

\section*{1.2 Objective function}

The objective function is the function that you wish to \textit{minimize}. If you want to maximize the function, simply multiply the expression by $-1$. The structure of the objective function .m file is identical to any other user-defined function (see the Matlab Basics Tutorial if you forget how to do this).

\section*{1.3 Constraint function}

The constraint function takes as input the independent vector variable, $x$, and outputs two vectors corresponding to the inequality and equality constraint functions. Inequality constraint functions must always be expressed as \textit{less than or equal to zero}. For example, if you have the expression, $A > B$, you would write it as $B - A < 0$. Equality constraints should be set equal to zero (e.g. $A = B$ becomes $A - B = 0$).

\section*{2. Example Code}

The constrained minimization problem is to determine the values of $A$, $B$, and $C$ that \textit{maximize} the following expression: $f = -1000 + 5A + 3BC - B + 2C$. The constraints are as follows:

1. $2A = B + C$
2. $A + B + C = 1000$
3. $A \geq 0, B \geq 0, C \geq 0$

The program code looks like this:

\textbf{The main code(``opti.m'')}:
\begin{verbatim}
clear all
x0=[0,0,0];
options=optimset('Largescale','off');
[x,fval]=fmincon('profit',x0,[],[],[],[],[],[],'confun',options)
\end{verbatim}

\textbf{The objective function (``profit.m'')}:
\begin{verbatim}
function f=profit(x)
f=-(1000+5*x(1)+3*x(2)*x(3)-x(2)+2*x(3))
\end{verbatim}

\textbf{The constraint function (``confun.m'')}:
\begin{verbatim}
function [cin,ceq]=confun(x)
cin=[-x(1);-x(2);-x(3)];
ceq=[x(1)+x(2)+x(3)-1000;2*x(1)-(x(2)+x(3))];
\end{verbatim}

You execute this code by typing `opti' at the command prompt.