Overview of Matlab Optimization Toolboxes

Natasha Sharma

CPS 5195 Seminar, October 28 2014

What we covered in the last session

- How to boil an egg in a microwave by optimally supplying heat? (Cast as an optimal control problem).
- Use of Symbolic Toolbox in Matlab to avoid human error in tedious calculations.
- Some examples
- ...ran out of time!

What we covered in the last session

- How to boil an egg in a microwave by optimally supplying heat? (Cast as an optimal control problem).
- Use of Symbolic Toolbox in Matlab to avoid human error in tedious calculations.
- Some examples
- ...ran out of time!

What we covered in the last session

- How to boil an egg in a microwave by optimally supplying heat? (Cast as an optimal control problem).
- Use of Symbolic Toolbox in Matlab to avoid human error in tedious calculations.
- Some examples
- ...ran out of time!

What we covered in the last session

- How to boil an egg in a microwave by optimally supplying heat? (Cast as an optimal control problem).
- Use of Symbolic Toolbox in Matlab to avoid human error in tedious calculations.
- Some examples
- ...ran out of time!

What we covered in the last session

- How to boil an egg in a microwave by optimally supplying heat? (Cast as an optimal control problem).
- Use of Symbolic Toolbox in Matlab to avoid human error in tedious calculations.
- Some examples
- ...ran out of time!

Motivation for today's talk

- Finish what we started...(Practical session on Matlab optimization toolbox)
- Since the first and second derivative test fails for transcendental functions, we need to come up with better way to find x^* satisfying $\nabla f(x^*) = 0$.
- We present a constrained and unconstrained minimization problem and list the steps to be followed to solve them using the toolbox.

Motivation for today's talk

- Finish what we started...(Practical session on Matlab optimization toolbox)
- Since the first and second derivative test fails for transcendental functions, we need to come up with better way to find x^* satisfying $\nabla f(x^*) = 0$.
- We present a constrained and unconstrained minimization problem and list the steps to be followed to solve them using the toolbox.

Motivation for today's talk

- Finish what we started...(Practical session on Matlab optimization toolbox)
- Since the first and second derivative test fails for transcendental functions, we need to come up with better way to find x^* satisfying $\nabla f(x^*) = 0$.
- We present a constrained and unconstrained minimization problem and list the steps to be followed to solve them using the toolbox.

Unconstrained minimization

Consider the objective function $f(x, y)$ to be :

$$\min\ f(x,y) = x\exp(-x^2 - y^2)\ +\ (x^2 + y^2)/20$$

With the help of Matlab, we can plot the function to get an idea of where its minimizer lies!
This can be done with the use of ezsurfc(.,.) function.

Minimizing an unconstrained function f

fminunc

```
f = @(x,y) x.*exp(-x.^2 -y.^2) +...
    (x.^2 + y.^2)/20;
ezsurfc(f,[-2,2]);
fun =@(x)f(x(1),x(2));
x0  =[-.5;0];
options = optimoptions('fminunc', ...
          'Algorithm','quasi-newton');
options.Display = 'iter';
[x,fval,exitflag,output] =...
fminunc(fun,x0,options);
```

Algorithm: Quasi-Newton

Before knowing what a quasi-Newton method is, we should understand what a Newton's method is. The Newton's method for finding the zeros of a function g and is given by:

$$x_{n+1} = x_n - g'(x_n)^{-1} g(x_n)$$

For our two dimensional problem, we want to find x^* for which $\nabla f(x^*) = 0$ holds.
Thus, $g(x_n) = \nabla$f and $g'(x_n) = \nabla^2$ f
The quasi-Newton algorithm is employed when we cannot compute the Hessian ∇^2f and only consider an approximation of it.

Minimizing f with constraints

Consider the objective function $f(x,y)$ to be :

$$\min\ f(x,y) = x\exp(-x^2 - y^2)\ +\ (x^2 + y^2)/20$$

subject to the constraint:

$$xy/2\ +\ (x+2)^2 + (y-2)^2/2 \le 2$$

The constraint above can be written as

$$g(x,y) = xy/2\ +\ (x+2)^2 + (y-2)^2/2 - 2$$

Minimizing f with constraints

fmincon

```
f = @(x,y) x.*exp(-x.^2-y.^2)+(x.^2+y.^2)/20
g = @(x,y) x.*y/2+(x+2).^2+(y-2).^2/2-2;
ezplot(g,[-6,0,-1,7])
hold on;
ezcontour(f,[-6,0,-1,7])
plot(-.9727,.4685,'ro');
legend('constraint','fcontours','minimum');
hold off;
options = optimoptions('fmincon', ...
'Algorithm','interior-point',...
'Display','iter');
```

fmincon

```
options = optimoptions('fmincon',...
'Algorithm','interior-point',...
'Display','iter');
gfun = @(x) deal(g(x(1),x(2)),[]);
[x,fval,exitflag,output] = ...
fmincon(fun,x0,[],[],[],[],[],[]...,
gfun,options);
```

Interior Point Method

Solve minimize f(x)
subject to

$$c_i(x) \geq 0 \quad i = 1, 2, ...m. \qquad (*)$$

- Barrier Function $B(x, \mu)$

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^{m} ln(c_i(x))$$

- Minimizer of (*) is equivalent to the minimizer of $B(x, \mu)$ as $\mu \to 0$.

Interior Point Method (continued)

- Introduce the Lagrange multiplier $\lambda \in \mathbb{R}^m$ and complementarity condition $\lambda_i \; c_i = \mu, \quad i = 1, 2, ..., m$.
- In view of the Lagrange multiplier λ_i and the definition of $B(.,.)$ we have

$$\nabla B = \nabla f - (\sum_{i=1}^{m} \lambda_i \partial_j c_i)_{j=1}^{m}.$$

- Thus, our task is to find the minimizer of $B(.,.)$ via a Newton or quasi-Newton method.