Christopher Lum
lum@uw.edu

Lecture 07c
Solving Systems of Equations Using the Optimization Penalty Method

The YouTube video entitled ‘Solving Systems of Equations Using the Optimization Penalty Method’ that covers this lecture is located at https://youtu.be/rx2vUzjuDc0.

Outline

-System of Equations
    -Fully Constrained
    -Over Constrained
-Inequalities

System of Equations

Fully Constrained

Consider a system of 3 equations

$$x_1 x_2 = 3$$
$$x_1 + x_2 = -3 x_3$$
$$\frac{x_1^2}{x_3} = -4$$

We can write these in standard form of

$$f_1(x) = x_1 x_2 - 3 = 0 \qquad \textbf{(Eq.1.1)}$$
$$f_2(x) = x_1 + x_2 + 3 x_3 = 0 \qquad \textbf{(Eq.1.2)}$$
$$f_3(x) = \frac{x_1^2}{x_3} + 4 = 0 \qquad \textbf{(Eq.1.3)}$$

```
f1[x1_, x2_, x3_] = x1 x2 - 3;
f2[x1_, x2_, x3_] = x1 + x2 + 3 x3;
f3[x1_, x2_, x3_] = x1^2 / x3 + 4;
```

This is a system of 3 equations and 3 unknowns. In this case, because the equations are somewhat simple, we can attempt to find solutions by solving them simultaneously.

Solving Eq.1.1 for x_1 yields

$$x_1 = \frac{3}{x_2} \qquad \textbf{\textit{(Eq.1.A)}}$$

Substituting Eq.1.A into Eq.1.2 yields

$$x_1 + x_2 + 3\,x_3 = 0$$

$$\frac{3}{x_2} + x_2 + 3\,x_3 = 0$$

$$\frac{3+x_2^2}{x_2} + 3\,x_3 = 0$$

$$3\,x_3 = -\frac{3+x_2^2}{x_2}$$

$$x_3 = -\frac{3+x_2^2}{3\,x_2} \qquad \textbf{\textit{(Eq.1.B)}}$$

Substituting both Eq.1.A and Eq.1.B into Eq1.3 yields

$$\frac{x_1^2}{x_3} + 4 = 0$$

$$\frac{\left(\frac{3}{x_2}\right)^2}{\left(-\frac{3+x_2^2}{3\,x_2}\right)} + 4 = 0$$

$$\frac{\left(\frac{3}{x_2}\right)^2}{\left(-\frac{3+x_2^2}{3\,x_2}\right)} = -4$$

$$\left(\frac{3}{x_2}\right)^2 = 4\left(\frac{3+x_2^2}{3\,x_2}\right)$$

$$\frac{9}{x_2^2} = 4\left(\frac{3+x_2^2}{3\,x_2}\right)$$

$$\frac{9}{4} = \left(\frac{3+x_2^2}{3\,x_2}\right)x_2^2$$

$\frac{9}{4} = \left(\frac{3+x_2^2}{3}\right) x_2$

$\frac{27}{4} = \left(3 + x_2^2\right) x_2$

$\frac{27}{4} = 3\, x_2 + x_2^3$

```
Solve[27/4 == 3 x2 + x2^3, x2] // N
```

```
{{x2 → 1.37792}, {x2 → -0.688962 + 2.10333 i}, {x2 → -0.688962 - 2.10333 i}}
```

Taking the real solution as one possible solution, we have

$x_2 = 1.37792$

```
x2solutionA = 1.37792;
```

Now back substituting into Eq.1.A and Eq.1.B yields solutions for x_1 and x_3

```
x1solutionA = 3 / x2solutionA
x2solutionA
x3solutionA = -(3 + x2solutionA^2)/(3 x2solutionA)
```

```
2.17719

1.37792

-1.18504
```

We can verify with Mathematica

```
temp = Solve[{f1[x1, x2, x3] == 0, f2[x1, x2, x3] == 0, f3[x1, x2, x3] == 0}, {x1, x2, x3}] //
   MatrixForm // N
```

$$\begin{pmatrix} x1 \to -0.421928 + 1.2881\, i & x2 \to -0.688962 - 2.10333\, i & x3 \to 0.370297 + 0.271743\, i \\ x1 \to -0.421928 - 1.2881\, i & x2 \to -0.688962 + 2.10333\, i & x3 \to 0.370297 - 0.271743\, i \\ x1 \to 2.17719 & x2 \to 1.37792 & x3 \to -1.18504 \end{pmatrix}$$

Alternatively, we can formulate as an optimization problem

$(\wp_C)$ minimize $f_0(x) = \text{constant}$ ***(Eq.2)***

such that $f_i(x) = 0$ $\quad i = 1, 2, 3$ (3 equality constraints)

So in this case, the cost function is a constant and this “optimization” problem is merely a problem of finding the feasible set.

Recall from our discussion on ‘Converting Constrained Optimization to Unconstrained Optimization Using the Penalty Method’, we could convert this to an alternate, approximate optimization problem of (for simplicity, we choose $f_0(x) = 0$)

$$(\wp_{\text{approx}}) \quad \underset{x \in \mathbb{R}^3}{\text{minimize}} \; \hat{f}_0(x) = \alpha_1 f_1(x)^2 + \alpha_2 f_2(x)^2 + \alpha_3 f_3(x)^2 \qquad \textbf{(Eq.3)}$$

We see that the minimum cost function value should be $\hat{f}_0(x^*) = 0$ which occurs when all 3 equations constraints are satisfied (therefore meaning that all 3 equations in Eq.1.1 - Eq.1.3 are solved simultaneously).

Starting from an initial guess of $\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ yields

```
xstar =

   1.0e-03 *

  -0.0000
  -0.1712
   0.0571


f0 =

  250.0000
```

This clearly has not converged to the correct solution as $f_0 \neq 0$. We can rerun the algorithm using this x^* as the initial guess

```
xstar =

    -0.0155
  -193.5125
    64.5093


f0 =

  160.0003
```

This has improved the cost but it is still too high. Repeating with this x^* as the initial guess does not improve the cost, and in fact the solution begins to converge to an unrealistic number (show Matlab sim).

The problem is that we have fallen into a local minimum due to our initial guess being too far from the actual optimal solution. If we instead initialize with an initial guess of $\begin{pmatrix} 1 & 1 & -1 \end{pmatrix}^T$, we obtain

```
xstar =

    2.1772
    1.3779
   -1.1850


f0 =

   2.0732e-19
```

Which is a correct answer. Note that this method only finds a single solution. If there are multiple solutions, you may need to start at another initial guess vector to coax the optimizer to find this different solution.

Over Constrained

It is interesting to note that we can over constrain the problem and still use the optimization penalty method to attempt to find a solution that is a compromise. For example, if we add a 4th constraint

$$f_4(x) = x_1 - x_2 = 0 \qquad \textbf{\textit{(Eq.1.4)}}$$

```
f4[x1_, x2_, x3_] = x1 - x2;
```

Note that this implies that $x_1 = x_2$ which is inconsistent with the previous solution of $x_1 = 2.1772$ and $x_2 = 1.3779$. Therefore, the problem has no solution when attempting to solve all 4 equations simultaneously

```
temp = Solve[{f1[x1, x2, x3] == 0, f2[x1, x2, x3] == 0,
    f3[x1, x2, x3] == 0, f4[x1, x2, x3] == 0}, {x1, x2, x3}] // MatrixForm
```

```
{}
```

Note if we drop a constraint, we can possibly solve the system of equations. For example, after dropping Eq.1.1, we have

```
temp = Solve[{f2[x1, x2, x3] == 0, f3[x1, x2, x3] == 0, f4[x1, x2, x3] == 0}, {x1, x2, x3}] //
   MatrixForm // N
```

```
( x1 → 2.66667  x2 → 2.66667  x3 → -1.77778 )
```

So to summarize

Eq.1.1, Eq.1.2, Eq.1.3, ~~Eq.1.4~~	$\Rightarrow$	$x^* = (\,2.18 \quad 1.38 \quad -1.19\,)$	(scenario 1)
~~Eq.1.1~~, Eq.1.2, Eq.1.3, Eq.1.4	$\Rightarrow$	$x^* = (\,2.67 \quad 2.67 \quad -1.78\,)$	(scenario 2)

However, even in the over constrained case we can still formulate the approximate optimization problem as

$$(\wp_{\text{approx}}) \quad \underset{x \in \mathbb{R}^3}{\text{minimize}}\ \hat{f}_0(x) = \alpha_1 f_1(x)^2 + \alpha_2 f_2(x)^2 + \alpha_3 f_3(x)^2 + \alpha_4 f_4(x)^2 \qquad \textbf{\textit{(Eq.4)}}$$

Depending on the values of penalty parameters, α_i, we can influence the optimizer solution

For example, if we select α_i according to the following mapping

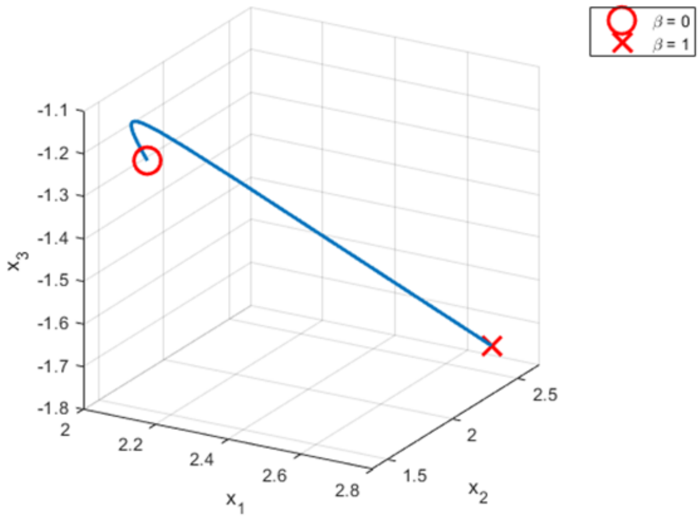
$$\alpha_1 = (1 - \beta)$$
$$\alpha_2 = 1$$
$$\alpha_3 = 1$$
$$\alpha_4 = \beta$$

We see that at $\beta = 0$, this becomes the scenario 1 and when $\beta = 1$, this becomes scenario 2

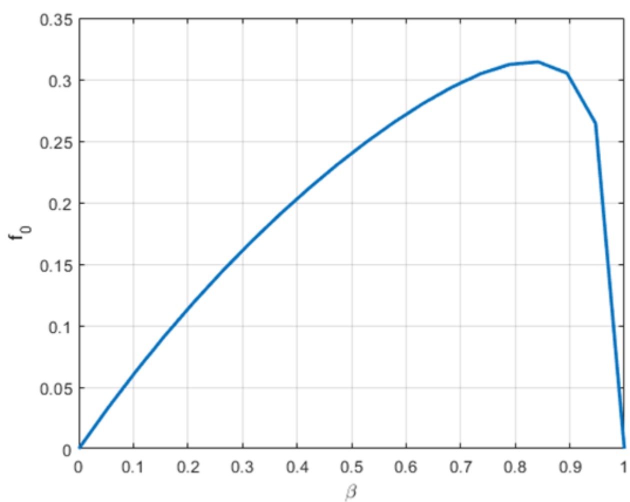
So we have

$$(\wp_{\text{approx}}) \quad \underset{x \in \mathbb{R}^3}{\text{minimize}}\ \hat{f}_0(x) = (1 - \beta)\ f_1(x)^2 + f_2(x)^2 + \alpha_3 f_3(x)^2 + \beta f_4(x)^2$$

Solving for various $\beta \in [0,\ 1]$ yields



With f_0 shown below



So at values of β between 0 and 1, the solution violates the constraints but the optimizer choses points attempt to minimize the amount of violation.

Inequalities

We can apply the penalty method to solve simultaneous equations that contain both equalities and inequalities. For example, consider the following 5 equations/relationships on 6 variables

$$x_1 x_5 - 3 = 0$$
$$x_1 + x_2 + 3 x_4 = 0$$
$$x_2^2 / x_6 + 4 = 0$$
$$x_1 + \cos(x_2) x_5 \leq 0$$

$x_5 + x_1 + x_3 + 7\,x_6 \leq 0$

We formulate the approximate optimization problem of

$$(\wp_{\text{approx}}) \quad \underset{x \in \mathbb{R}^3}{\text{minimize}}\ \hat{f}_0(x) = f_0(x) + \alpha_1\,f_1(x)^2 + \alpha_2\,f_2(x)^2 + \alpha_3\,f_3(x)^3 + \alpha_4\,\max(0,\,f_4(x))^2 + \alpha_5\,\max(0,\,f_5(x))^2$$

Using ‘fminsearch’, we obtain the solution (starting from x_{guess} = zeros(6, 1)

```
xstar =

   1.0e-03 *

    0.0157
   -0.0036
    0.0420
    0.1823
   -0.1083
   -0.0000


f0 =

    9.0000
```

Which has not yet converged. If we use this intermediate solution as the initial guess for the next call to fminsearch, we obtain

```
xstar =

   -0.4031
   -0.0000
   -2.1516
    0.1344
   -7.4424
   -0.0000


f0 =

   1.8520e-21
```

This is a solution as the cost function is near zero.

We can verify that this satisfies the constraints

```
f1 =

     0


f2 =

   5.5511e-17


f3 =

     0


f4 =

   -7.8455


f5 =

  -10.0292
```

Changing the initial guess vector will change the solution (but it will still satisfy the constraints). For example, with an $x_{\text{guess}} = (1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6)^T$, we obtain

```
xstar =

    1.4319
    2.3295
    4.7920
   -1.2538
    2.0952
   -1.3567


f0 =

   8.6670e-23
```

This still satisfies the constraints

```
f1 =

   1.8878e-12


f2 =

  -8.5283e-12


f3 =

   3.2210e-12


f4 =

   -0.0096


f5 =

   -1.1777
```