IS609 Week 6 HW

Charley Ferrari

Monday, September 28, 2015

Page 251, Question 2. Nutritional Requirements

A rancher has determined that the minimum weekly nutritional requirements for an average sized horse include 40lb of protein, 20lb of carbohydrates, and 45lb of roughage. These are obtained from the following sources in varying amounts at the prices indicated:

	Protein (lb)	Carbs (lb)	Roughage (lb)	Cost
Hay (per bale)	0.5	2.0	5.0	\$1.80
Oats (per sack)	1.0	4.0	2.0	\$3.50
Feeding Blocks (per block)	2.0	0.5	1.0	\$0.40
High Protein Conc. (per sack)	6.0	1.0	2.5	\$1.00
Requirements per horse (per week)	40.0	20.0	45.0	

Relationships:

Protein = (0.5 × Hay) + (1 × Oats) + (2 × FB) + (6 × HPC)

Carbs = (92 × Hay) + (4 × Oats) + (0.5 × FB) + (1 × HPC)

Roughage = (5 × Hay) + (2 × Oats) + (1 × FB) + (2.5 × HPC)

Cost = (1.8 × Hay) + (3.5 × Oats) + (0.4 × FB) + (1 × HPC)

Constraints:

Protein > 40

Carbs > 20

Roughage > 45

Optimization:

Minimize Cost

Page 264, Question 6

Minimize 10x + 35y subject to:

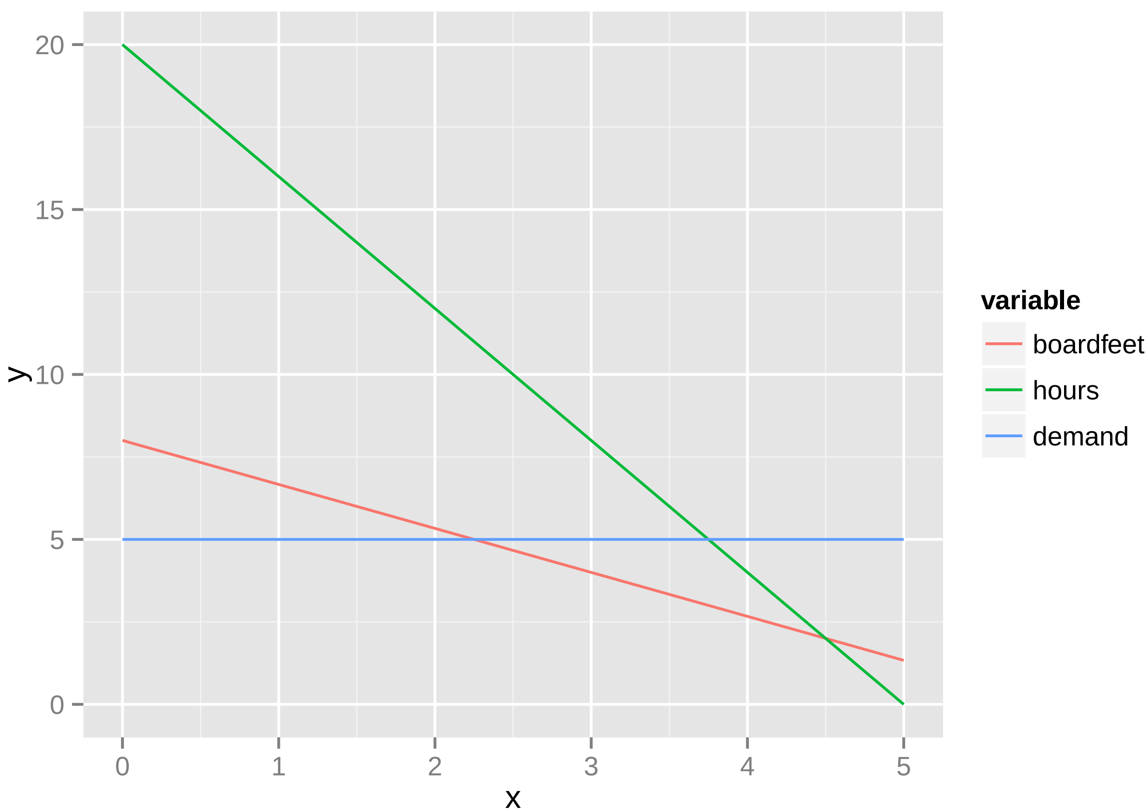
(Board feet of lumber, hours of carpentry, demand, nonnegativity)

$$8x + 6y \leq 48$$

$$4x + y \leq 20$$
$$y \geq 5$$
$$x, y \geq 0$$

I'll solve all of these for y and graph them out in ggplot, to get an idea of my feasible zone:

```
x <- seq(0,5,by=0.1)

data <- data.frame(x = x, boardfeet = (48-8*x)/6, hours = 20-4*x, demand = 5)

library(reshape2)

data <- melt(data, id.vars="x", value.name="y")

library(ggplot2)

ggplot(data, aes(x=x, y=y, color=variable)) + geom_line()
```

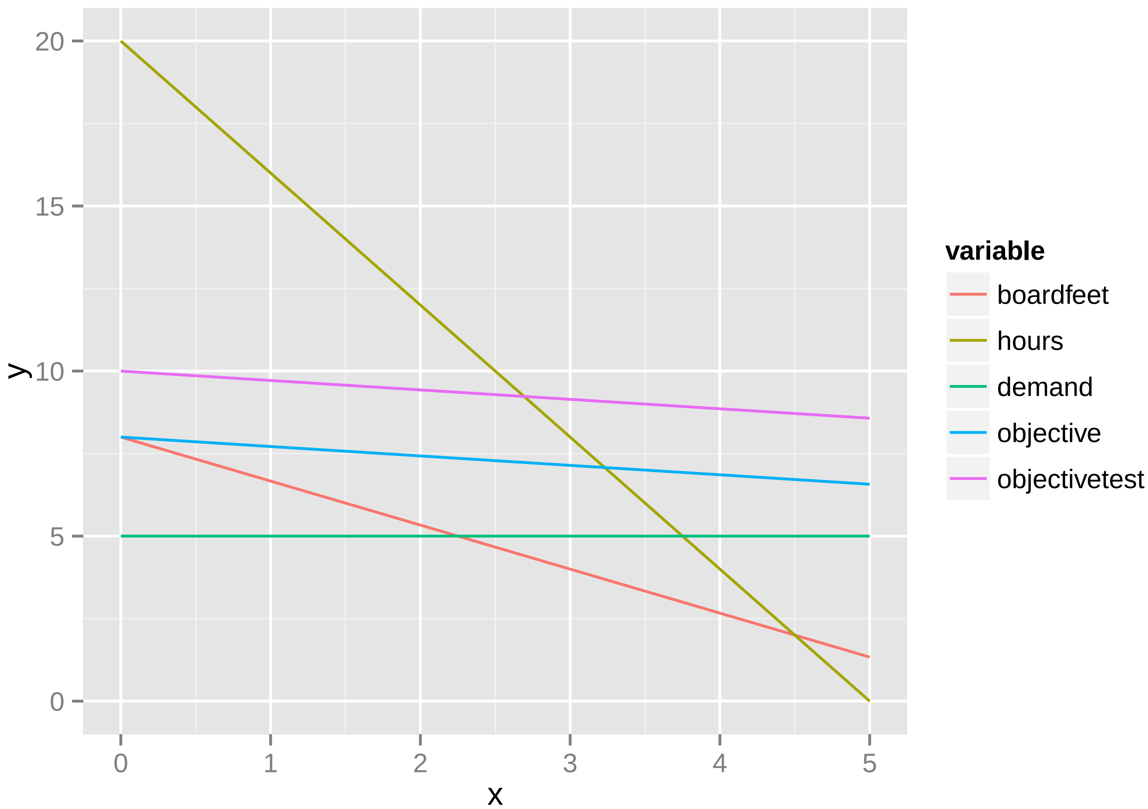


My feasible zone is below the red line (board-feet of lumber), below the green line (hours of carpentry), above the blue line (demand), and above 0. So, the feasible zone is the triangle defined by y=5, x=0, and the boardfeet line 8x + 6y = 48. This means that the slope of the objective function will determine which point, (0,8) or (2.25,5) the optimization occurs. If the slope is more negative than the boardfeet line ($-\frac{4}{3}$), then the optimized point will occur at (2.25, 5). Otherwise, it will occur at (0,8).

The slope of the line 10x + 35y = c is $-\frac{10}{35}$, or $-\frac{2}{7}$. This is not more negative than the boardfeet slope, so the optimum will occur at (0,8). Lets graph this out.

If the line 10x + 35y = c crosses through (0,8), then 35*8 = c, so c = 280. Lets add this to our lines:

```
x <- seq(0,5,by=0.1)

data <- data.frame(x = x, boardfeet = (48-8*x)/6, hours = 20-4*x, demand = 5,
                   objective = (280-10*x)/35, objectivetest = (350-10*x)/35)

data <- melt(data, id.vars="x", value.name="y")

ggplot(data, aes(x=x, y=y, color=variable)) + geom_line()
```



Here I graphed the objective along with a test, with c a bit higher (at 350). You can see if it's a bit higher, it's outside of the feasible zone.

Page 268, Question 6 (Algebraic Solution)

This time we're going to rewrite the equations in question 6

Maximize 10x + 35y subject to:

$$8x + 6y + c1 = 48$$

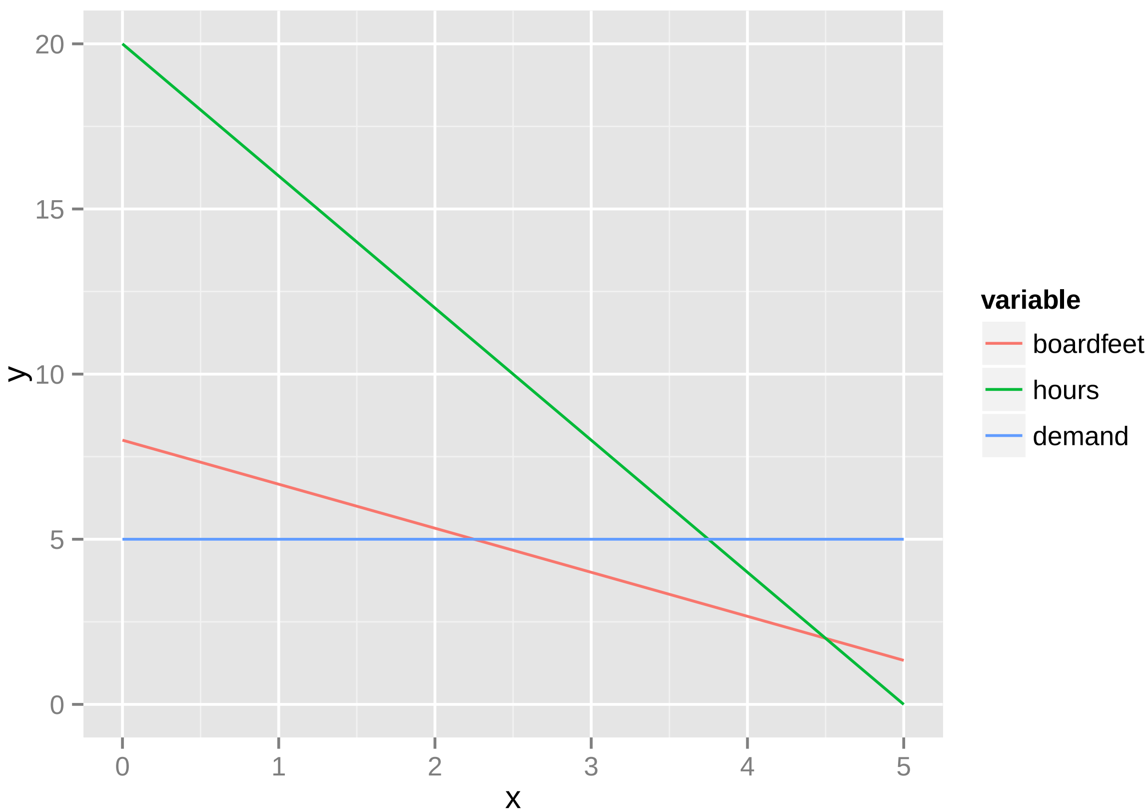
$$4x + y + c2 = 20$$

$$y + c3 = 5$$

Here we won't consider the case of setting y = 0, because we already know $y \geq 5$. So, we'll be solving this by setting four variables to 0 two at a time: x, c1, c2, and c3.

Lets look at the graph again:

```
x <- seq(0,5,by=0.1)

data <- data.frame(x = x, boardfeet = (48-8*x)/6, hours = 20-4*x, demand = 5)

data <- melt(data, id.vars="x", value.name="y")

ggplot(data, aes(x=x, y=y, color=variable)) + geom_line()
```



c1, c2 = 0

$$8x + 6y = 48$$
$$4x + y = 20$$
$$y - c3 = 5$$

This correspondes to the point (4.5, 2), with c3 = -3, x = 4.5, and y = 2. This is not feasible because c3 is negative. This corresponds to the intersection of the red and green lines.

c1, c3 = 0

$$8x + 6y = 48$$

$$4x + y + c2 = 20$$
$$y = 5$$

This corresponds to the point (2.25, 5), with c2 = 6, x=2.25, and y = 5. This is a feasible point, and on the graph it corresponds to the intersection of the blue and red lines.

c2, c3 = 0

$$8x + 6y + c1 = 48$$
$$4x + y = 20$$
$$y = 5$$

This corresponds with the values c1 = -12, x = 3.75, and y = 5. We know this isn't a feasible point because c1 is negative, and the three slack variables we picked need to be nonnegative. This point is (3.75, 5), and corresponds to the intersection of the green and blue lines.

c1, x = 0

$$6y = 48$$
$$y + c2 = 20$$
$$y - c3 = 5$$

This corresponds with the values c2 = 12, c3 = 3, and y = 8. This is feasible, and corresponds with the point (0,8), which is feasible. This is at the y-intercept of the red line.

c2, x = 0

$$6y + c1 = 48$$
$$y = 20$$
$$y - c3 = 5$$

This corresponds with the values c1 = -71, c3 = 15, and y = 20. This is not feasible since c1 is negative, and corresponds to the point (0,20). This is the y-intercept of the green line.

c3, x = 0

$$6y + c1 = 48$$
$$y + c2 = 20$$
$$y = 5$$

This corresponds with the values c1 = 18, c2 = 15, and y = 5. This corresponds to the point (0,5), and is feasible. This corresponds with the y-intercept of the blue line.

Solving algebraically, we have three feasible points: (0,5), (0,8), and (2.25, 5). Lets plug these back into the objective function:

x	y	10x + 35y
0	5	175
0	8	280
2.25	5	197.5

This gives us the same answer we had before, the point that maximizes this function is at (0,8).

Page 278, Question 6 (Simplex Solution)

See the write-ups below:

x	y	C_1	C_2	C_3	z	RHS	Ratio
8	6	1	6	0	0	48	$48/6 = 8$
4	1	0	1	0	0	20	$20/1 = 20$
0	-1	0	0	1	0	-5	$-5/-1 = 5$
-10	-35	0	0	0	1	0	

Dependent variables: $\{C_1, C_2, C_3, z\}$
Independent variables: $\{x=0, y=0\}$
extreme point: $(x,y)=(0,5)$
Value of objective: $z = 175$

Optimality test: entering variable is y
Feasibility test: Ratio suggests that C_3 is the exiting variable
Pivot: divide the row containing the exiting variable by the coefficient of the entering variable in this row. Then, eliminate the entering variable y from the remaining rows

$y - C_3 = 5, \quad y = C_3 + 5$

$8x + 6C_3 + 30 + C_1 = 48$	$8x + C_1 + 6C_3 = 18$
$4x + C_3 + 5 + C_2 = 20$	$4x + C_2 + C_3 = 15$
$-10x - 35C_3 - 175 + Z = 0$	$-10x - 35C_3 + Z = 175$

x	y	C_1	C_2	C_3	Z	RHS	Ratio
8	0	1	0	6	0	18	$18/6 = 3$
4	0	0	1	1	0	15	$15/1 = 15$
0	1	0	0	-1	0	5	$5/-1 = -5$
-10	0	0	0	-35	1	175	

Dependent variables: $\{y, C_1, C_2, z\}$
Independent variables: $\{C_3 = 0, x = 0\}$
Optimality Test: entering variable is C_3
Feasibility Test: exiting variable is C_1
Pivot: divide the row containing the exiting variable by the coefficient of the entering variable, Then, eliminate the entering variable in all remaining rows

$\frac{4}{3}x + \frac{1}{6}C_1 + C_3 = 3$

$4x + C_2 + 3 - \frac{4}{3}x - \frac{1}{6}C_1 = 15$

$Y - 3 + \frac{4}{3}x + \frac{1}{6}C_1 = 5$

$-10x - 105 + \frac{140}{3}x + \frac{35}{6}C_1 + Z = 175$

$C_3 = 3 - \frac{4}{3}x - \frac{1}{6}C_1$

$\frac{8}{3}x - \frac{1}{6}C_1 + C_2 = 12$

$\frac{4}{3}x + Y + \frac{1}{6}C_1 = 8$

$\frac{110}{3}x + \frac{35}{6}C_1 + Z = 280$

X	Y	C_1	C_2	C_3	Z	RHS
4/3	0	1/6	0	1	0	3
8/3	0	-1/6	1	0	0	12
4/3	1	1/6	0	0	0	8
110/3	0	35/6	0	0	1	280

Dependent Variables: $\{Y, C_2, C_3, Z\}$

Independent Variables: $\{X=0, C_1=0\}$

There are no more negative coefficients in the Tableau, so This is our solution. Lets plug in $C_1=0$ to get our Y

$8x + 6y + C_1 = 48 \quad x=0,\ C_1=0$

$6y = 48$

$y = 8$

So, we have our point: $(x=0, y=8)$ and the profit at this point: \$280

Page 284, Question 1

For the example problem in this section, determine the sensitivity of the optimal solution to a change in c_2 using the objective function $25x_1 + c_2x_2$

In the example program we can build on the fact that we know that the extreme point (12,15) remains optimal as long as the slope of the objective function is less than $-\frac{2}{3}$ but greater than $\frac{5}{4}$. With c_2 variable, the slope is $-\frac{25}{c_2}$. So, we end up with the inequality:

$$-\frac{5}{4} \leq -\frac{25}{c_2} \leq -\frac{2}{3}$$

Or

$$20 \leq c_2 \leq \frac{75}{2}$$

c_2 is the profit per bookcase. If this profit drops below 20, the carpenter should produce only tables. If the profit becomes greater than \$32.50 per bookcase, the carpenter should produce only bookcases. If the profit remains between these two values, than the carpenter should stick to the current optimal value of 12 tables and 15 bookcases.

Page 295, Question 3

Use the Curve Fitting Criterion to minimize the sum of the absolute deviations for the following models and data set:

a. $y = ax$

b. $y = ax^2$

c. $y = ax^3$

First lets define our data:

```
x <- c(7,14,21,28,35,42)
y <- c(8,41,133,250,280,297)

data <- data.frame(x=x, y=y, x2=x^2, x3=x^3)
```

for each part of this question, I can define a cost function. The cost function is simply:

$$f(c) = \sum |y - ax^k|$$

Where k is the power we're raising it to, 1 in a, 2 in b, and 3 in c.

There are a few problems with this method. We need to define the interval beforehand, which means we need to have an idea of where the desired a lies. Lets just solve using the lm function to find some approximations of where we should be looking (we won't expect them to match, since with lm we're minimizing the sum of the squared errors, but it should give us a ballpark goal.

```
summary(lm(y~x+0,data=data))
```

```
##
## Call:
## lm(formula = y ~ x + 0, data = data)
##
## Residuals:
##      1      2      3      4      5      6
## -43.33 -61.66 -20.99  44.68  23.35 -10.98
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   7.3328     0.6276   11.68 8.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.91 on 5 degrees of freedom
## Multiple R-squared:  0.9647, Adjusted R-squared:  0.9576
## F-statistic: 136.5 on 1 and 5 DF,  p-value: 8.071e-05
```

```
summary(lm(y~x2+0,data=data))
```

```
##
## Call:
## lm(formula = y ~ x2 + 0, data = data)
##
## Residuals:
##        1        2        3        4        5        6
##  -2.1367   0.4532  41.7697  87.8127  26.5824 -67.9213
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## x2  0.20687    0.02326   8.893 0.000299 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.37 on 5 degrees of freedom
## Multiple R-squared:  0.9405, Adjusted R-squared:  0.9286
## F-statistic: 79.08 on 1 and 5 DF,  p-value: 0.0002993
```

```r
summary(lm(y~x3+0,data=data))
```

```
##
## Call:
## lm(formula = y ~ x3 + 0, data = data)
##
## Residuals:
##       1       2       3       4       5       6
##   6.227  26.818  85.135 136.542  58.403 -85.919
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
## x3 0.0051684  0.0009735   5.309  0.00317 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.54 on 5 degrees of freedom
## Multiple R-squared:  0.8494, Adjusted R-squared:  0.8192
## F-statistic: 28.19 on 1 and 5 DF,  p-value: 0.003168
```

All of these are between [0,10], so I'll use that as my interval. I'm also setting a tolerance level as 0.01.

At the end of the while loop, the data frame solvetable will give me the values a and b, which give a range of possible values for a that is within my set tolerance.

a. $y = ax$

```r
psi <- ((1+sqrt(5))/2)-1

cost <- function(a, data){
  sum(abs(data$y - a*(data$x)))
}


a <- 0
```

```r
b <- 10
c1 <- a+(1-psi)*(b-a)
c2 <- a+psi*(b-a)
fc1 <- cost(c1,data)
fc2 <- cost(c2,data)

solvetable <- data.frame(a=a,b=b,c1=c1,c2=c2,fc1=fc1,fc2=fc2)

tolerance <- 0.01

t <- 1

while(t > tolerance){
  if(fc1<=fc2){
    b <- c2
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  } else{
    a <- c1
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  }
}

solvetable
```

```
##           a         b       c1       c2      fc1      fc2
## 1  0.000000 10.000000 3.819660 6.180340 509.9357 262.0643
## 2  3.819660 10.000000 6.180340 7.639320 262.0643 211.4257
## 3  6.180340 10.000000 7.639320 8.541020 211.4257 268.2328
## 4  6.180340  8.541020 7.082039 7.639320 199.7228 211.4257
## 5  6.180340  7.639320 6.737621 7.082039 220.5299 199.7228
## 6  6.737621  7.639320 7.082039 7.294902 199.7228 204.1929
## 7  6.737621  7.294902 6.950483 7.082039 207.1196 199.7228
## 8  6.950483  7.294902 7.082039 7.163346 199.7228 201.4303
## 9  6.950483  7.163346 7.031789 7.082039 201.9973 199.7228
## 10 7.031789  7.163346 7.082039 7.113096 199.7228 200.3750
## 11 7.031789  7.113096 7.062846 7.082039 200.0407 199.7228
## 12 7.062846  7.113096 7.082039 7.093902 199.7228 199.9719
## 13 7.062846  7.093902 7.074708 7.082039 199.5689 199.7228
## 14 7.062846  7.082039 7.070177 7.074708 199.5789 199.5689
## 15 7.070177  7.082039 7.074708 7.077508 199.5689 199.6277
## 16 7.070177  7.077508 7.072977 7.074708 199.5325 199.5689
```

Now lets just redefine our cost function and do the same thing for b and c:

b. $y = ax^2$

```
psi <- ((1+sqrt(5))/2)-1

cost <- function(a, data){
  sum(abs(data$y - a*(data$x^2)))
}


a <- 0
b <- 10
c1 <- a+(1-psi)*(b-a)
c2 <- a+psi*(b-a)
fc1 <- cost(c1,data)
fc2 <- cost(c2,data)

solvetable <- data.frame(a=a,b=b,c1=c1,c2=c2,fc1=fc1,fc2=fc2)

tolerance <- 0.01

t <- 1

while(t > tolerance){
  if(fc1<=fc2){
    b <- c2
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  } else{
    a <- c1
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  }
}

solvetable
```

```
##           a          b        c1        c2        fc1        fc2
## 1  0.0000000 10.0000000 3.8196601 6.1803399 16022.8644 26549.1356
## 2  0.0000000  6.1803399 2.3606798 3.8196601  9517.2711 16022.8644
## 3  0.0000000  3.8196601 1.4589803 2.3606798  5496.5933  9517.2711
## 4  0.0000000  2.3606798 0.9016994 1.4589803  3011.6778  5496.5933
## 5  0.0000000  1.4589803 0.5572809 0.9016994  1475.9155  3011.6778
## 6  0.0000000  0.9016994 0.3444185 0.5572809   526.7623  1475.9155
## 7  0.0000000  0.5572809 0.2128624 0.3444185   223.1277   526.7623
## 8  0.0000000  0.3444185 0.1315562 0.2128624   422.3910   223.1277
## 9  0.1315562  0.3444185 0.2128624 0.2631123   223.1277   285.5927
```

```
## 10 0.1315562  0.2631123 0.1818062 0.2128624   247.5555   223.1277
## 11 0.1818062  0.2631123 0.2128624 0.2320561   223.1277   223.2008
## 12 0.1818062  0.2320561 0.2009999 0.2128624   231.5670   223.1277
## 13 0.2009999  0.2320561 0.2128624 0.2201937   223.1277   219.8946
## 14 0.2128624  0.2320561 0.2201937 0.2247248   219.8946   217.8964
## 15 0.2201937  0.2320561 0.2247248 0.2275251   217.8964   216.6614
## 16 0.2247248  0.2320561 0.2275251 0.2292558   216.6614   217.5749
```

c. $y = ax^3$

```
psi <- ((1+sqrt(5))/2)-1

cost <- function(a, data){
  sum(abs(data$y - a*(data$x^3)))
}


a <- 0
b <- 10
c1 <- a+(1-psi)*(b-a)
c2 <- a+psi*(b-a)
fc1 <- cost(c1,data)
fc2 <- cost(c2,data)

solvetable <- data.frame(a=a,b=b,c1=c1,c2=c2,fc1=fc1,fc2=fc2)

tolerance <- 0.01

t <- 1

while(t > tolerance){
  if(fc1<=fc2){
    b <- c2
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  } else{
    a <- c1
    c1 <- a+(1-psi)*(b-a)
    c2 <- a+psi*(b-a)
    fc1 <- cost(c1,data)
    fc2 <- cost(c2,data)
    solvetable <- rbind(solvetable,c(a,b,c1,c2,fc1,fc2))
    t <- b-a
  }
}

solvetable
```

```
##     a             b           c1           c2          fc1           fc2
```

```
## 1  0 10.000000000 3.819660113 6.180339887 576764.2476 933847.7524
## 2  0  6.180339887 2.360679775 3.819660113 356074.5048 576764.2476
## 3  0  3.819660113 1.458980338 2.360679775 219680.7428 356074.5048
## 4  0  2.360679775 0.901699437 1.458980338 135384.7620 219680.7428
## 5  0  1.458980338 0.557280900 0.901699437  83286.9808 135384.7620
## 6  0  0.901699437 0.344418537 0.557280900  51088.7812  83286.9808
## 7  0  0.557280900 0.212862363 0.344418537  31189.1995  51088.7812
## 8  0  0.344418537 0.131556175 0.212862363  18890.5817  31189.1995
## 9  0  0.212862363 0.081306188 0.131556175  11289.6178  18890.5817
## 10 0  0.131556175 0.050249987 0.081306188   6591.9638  11289.6178
## 11 0  0.081306188 0.031056200 0.050249987   3688.6540   6591.9638
## 12 0  0.050249987 0.019193787 0.031056200   1897.1429   3688.6540
## 13 0  0.031056200 0.011862413 0.019193787    856.3900   1897.1429
## 14 0  0.019193787 0.007331374 0.011862413    461.0334    856.3900
## 15 0  0.011862413 0.004531039 0.007331374    401.0127    461.0334
## 16 0  0.007331374 0.002800336 0.004531039    585.4128    401.0127
```