QUADRATIC MAJORIZATION OF A QUARTIC

JAN DE LEEUW

ABSTRACT. This is a didactic note, in which we show the precise behaviour of quadratic majorization algorithms. We also compare these algorithms with Newton's method.

1. INTRODUCTION

The problem we study in this note is minimization of the univariate quartic

$$f(x) = a + bx + cx^2 + dx^3 + ex^4 \tag{1}$$

using various iterative quadratic approximation algorithms.

It is, of course, not necessary at all to use iterative algorithms in this case. We can use simple algebraic methods. These gives both more information than iterative algorithms and they give the information faster. The purpose of this note is thus merely to illustrate with a simple example some concepts and results from the theory of majorization algorithms.

For ease of reference we give the derivatives of (1). The first four are

$$f'(x) = b + 2cx + 3dx^2 + 4ex^3, \tag{2a}$$

$$f''(x) = 2c + 6dx + 12ex^2, \tag{2b}$$

$$f'''(x) = 6d + 24ex, \tag{2c}$$

$$f''''(x) = 24e. \tag{2d}$$

Date: March 20, 2007.

In our actual computations we shall limit our attention to quartics with $b = 0$. This implies that the first derivative vanishes at zero. By assuming, in addition, that $c > 0$ we guarantee that f has a local minimum at zero. If $c = 0$ then f has a saddle point at zero. To ensure that the second derivative has an upper bound, we finally assume $e < 0$.

2. Uniform Quadratic Majorization

2.1. General. In uniform quadratic majorization, or UQM, we find a K such that the quadratic

$$g(x|y) = f(y) + f'(y)(x-y) + \frac{1}{2}K(x-y)^2 \tag{3}$$

satisfies

$$f(x) \leq g(x|y) \tag{4}$$

for all x and y. This is usually done by finding an upper bound for the second derivative. In fact, for twice-differentiable functions, (4) implies that $K \geq f''(y)$.

For fixed y we say that the quadratic $g(x|y)$ *majorizes* f at y. Considered as a function of two variables x and y we call $g(x|y)$ a *majorization scheme*. The *algorithmic map* $A()$ of the UQM algorithm is the **argmin** of the majorization function. Thus

$$A(y) = \mathop{\mathbf{argmin}}_{x}\, g(x|y) = y - \frac{1}{K}f'(y),$$

and the corresponding algorithm is $x^{(k+1)} = A(x^{(k)})$, where we suppose, of course, that the minimum is attained. By Zangwill [1969] this gives a globally convergent algorithm, by Ostrowski [1966] the rate of convergence is the modulus of the derivative of the algorithmic map at the point of attraction. Observe that

$$A'(y) = 1 - \frac{1}{K}f''(y)$$

2.2. **Quartic.** In the quartic case, with $e < 0$, the second derivative is maximized at the point where the third derivative vanishes, i.e. at

$$x = -\frac{1}{4}\frac{d}{e}.$$

It follows that the upper bound we look for is

$$f''(x) \leq K = 2c - \frac{3}{4}\frac{d^2}{e} = 2c + \frac{3}{4}\frac{d^2}{|e|} > 0.$$

The algorithmic map of UQM is the cubic

$$A(x) = x\left[\frac{3d^2 - 12d|e|x + 16e^2x^2}{8c|e| + 3d^2}\right].$$

Thus

$$\frac{|A(x) - 0|}{|x - 0|} = \left|\frac{3d^2 - 12d|e|x + 16e^2x^2}{8c|e| + 3d^2}\right|,$$

and

$$\lim_{x \to 0} \frac{|A(x) - 0|}{|x - 0|} = \frac{3d^2}{8c|e| + 3d^2}.$$

Also the derivative of the algorithmic map is the quadratic

$$A'(x) = \frac{3d^2 - 24d|e|x + 48e^2x^2}{8c|e| + 3d^2}.$$

value of the second derivative at zero is $2c$, and thus the linear convergence rate is

$$\lambda(0) = \frac{\frac{3}{4}\frac{d^2}{|e|}}{2c + \frac{3}{4}\frac{d^2}{|e|}}$$

We see that $d = 0$ leads to superlinear convergence. If $c = 0$, i.e. if the function has a saddle point at zero, then we have sublinear convergence.

3. Overrelaxation

4. Best Quadratic Majorization

We can try to improve the quadratic majorization by computing

$$
\begin{aligned}
K(y) &= \max_x \frac{f(x) - f(y) - f'(y)(x-y)}{\frac{1}{2}(x-y)^2} = \\
&= \max_x f''(y) + \frac{1}{3}f'''(y)(x-y) + \frac{1}{12}f''''(y)(x-y)^2.
\end{aligned}
$$

The maximum is attained at

$$
x = y - 2\frac{f'''(y)}{f''''(y)},
$$

and it is equal to

$$
K(y) = f''(y) - \frac{1}{3}\frac{[f'''(y)]^2}{f''''(y)}.
$$

The BQM algorithm is

$$
x^{(\nu+1)} = x^{(\nu)} - \frac{f'(x^{(\nu)})}{K(x^{(\nu)})}
$$

and the linear convergence speed at x_∞ is

$$
\lambda(x_\infty) = 1 - \frac{f''(x_\infty)}{K(x_\infty)}.
$$

In our quartic examples

$$
\lambda(0) = \frac{\frac{1}{2}\frac{d^2}{|e|}}{2c + \frac{1}{2}\frac{d^2}{|e|}}
$$

5. Local Majorization

6. Newton's Method

For comparison purposes, we also look at Newton's method. In our example

$$\begin{aligned}
x^{(\nu+1)} = x^{(\nu)} - \frac{f'(x^{(\nu)})}{f''(x^{(\nu)})} &= \\
&= \frac{3d(x^{(\nu)})^2 + 8e(x^{(\nu)})^3}{2c + 6d(x^{(\nu)}) + 12e(x^{(\nu)})^2} = \\
&= x^{(\nu)}\left[\frac{3d(x^{(\nu)}) + 8e(x^{(\nu)})^2}{2c + 6d(x^{(\nu)}) + 12e(x^{(\nu)})^2}\right] = \\
&= (x^{(\nu)})^2\left[\frac{3d + 8e(x^{(\nu)})}{2c + 6d(x^{(\nu)}) + 12e(x^{(\nu)})^2}\right].
\end{aligned}$$

This shows clearly what happens near zero. If $c > 0$ we have

$$x^{(\nu+1)} \approx \frac{3d}{2c}(x^{(\nu)})^2.$$

If $c = 0$ and $d \neq 0$ we have

$$x^{(\nu+1)} \approx \frac{1}{2}x^{(\nu)}.$$

And if $c = d = 0$ we have (exactly)

$$x^{(\nu+1)} = \frac{3}{4}x^{(\nu)}.$$

Thus Newton converges quadratically to a local minimum and linearly to a saddlepoint at zero.

7. Numerical Example

Consider, for example,

$$f(x) = 1 + 5x^2 - 4x^3 - \frac{1}{4}x^4.$$

The function has only two real roots, at -17.1658919 and 1.2944681. The function is plotted in Figure 1, and the first and second derivative are in Figure 2. There are local maxima at -12.78233 and

0.78233, and a local minimum at 0. The value of the second derivative at zero is 10. The maximum of the second derivative is attained at -4, where its value is 58.

7.1. UQM.

The linear convergence rate of UQM at zero is $1-10/58 = 0.8275862$. To illustrate the algorithm we have plotted uniform quadratic majorizations at $y = -15$, $y = 0$, and $y = 4$.

7.2. BQM.

7.3. Newton.

8. Code

8.1. Plots.

These are

```
f<-polynomial(c(1,0,5,-4,-.25))
x<-seq(-20,5,length=100)
g<-deriv(f)
h<-deriv(g)
u<-deriv(h)
m<-as.character(print(f))
pdf("plotf.pdf")
plot(x,predict(f,x),type="l",col="RED",ylab="f",main=m)
abline(h=0)
dev.off()
pdf("plotg.pdf")
plot(x,predict(g,x),type="l",col="BLUE",ylab="f'",main=m)
abline(h=0)
abline(v=solve(g)[1])
abline(v=solve(g)[2])
abline(v=solve(g)[3])
dev.off()
pdf("ploth.pdf")
```

```
plot(x,predict(h,x),type="l",col="GREEN",ylab="f''",main=m)
abline(h=0)
abline(v=solve(h)[1])
abline(v=solve(h)[2])
dev.off()
pdf("plotu.pdf")
plot(x,predict(u,x),type="l",col="GREEN",ylab="f'''",main=m)
abline(h=0)
abline(v=solve(u))
dev.off()
pdf("plotm.pdf")
x<-seq(-.5,1.5,length=100)
plot(x,predict(f,x),type="l",col="RED",ylab="f",main=m)
dev.off()
```

```
require("polynom")
a<-1
b<-0
c<-5
d<--4
e<--1/4
f<-polynomial(c(a,b,c,d,e))
g<-deriv(f)
h<-deriv(g)
u<-deriv(h)
v<-deriv(u)
m<-as.character(print(f))
kmax<-(2*c)-(3*d^2)/(4*e)
x<-seq(-20,5,length=100)
#
pdf("uqmmaj.pdf")
plot(x,predict(f,x),type="l",col="RED",ylab="f",main=m)
```

```
y<-−15
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kmax*(x−y)^2
points(x,ff,type="l",col="BLUE")
y<-0
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kmax*(x−y)^2
points(x,ff,type="l",col="BLUE")
y<-4
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kmax*(x−y)^2
points(x,ff,type="l",col="BLUE")
dev.off()
#
pdf("bqmmaj.pdf")
plot(x,predict(f,x),type="l",col="RED",ylab="f",main=m)
y<-−15
kopt<-predict(h,y)−(predict(u,y)^2)/(3*predict(v,y))
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kopt*(x−y)^2
points(x,ff,type="l",col="BLUE")
y<-0
kopt<-predict(h,y)−(predict(u,y)^2)/(3*predict(v,y))
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kopt*(x−y)^2
points(x,ff,type="l",col="BLUE")
y<-4
kopt<-predict(h,y)−(predict(u,y)^2)/(3*predict(v,y))
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*kopt*(x−y)^2
points(x,ff,type="l",col="BLUE")
dev.off()
#
pdf("newmaj.pdf")
plot(x,predict(f,x),type="l",col="RED",ylab="f",main=m)
y<-−15
knew<-predict(h,y)
ff<-predict(f,y)+predict(g,y)*(x−y)+0.5*knew*(x−y)^2
points(x,ff,type="l",col="BLUE")
y<-0
```

```
knew<-predict(h,y)
ff<-predict(f,y)+predict(g,y)*(x-y)+0.5*knew*(x-y)^2
points(x,ff,type="l",col="BLUE")
y<-4
knew<-predict(h,y)
ff<-predict(f,y)+predict(g,y)*(x-y)+0.5*knew*(x-y)^2
points(x,ff,type="l",col="BLUE")
dev.off()
#
```

```
require("polynom")
a<-1
b<-0
c<-5
d<--4
e<--1/4
f<-polynomial(c(a,b,c,d,e))
g<-deriv(f)
h<-deriv(g)
u<-deriv(h)
v<-deriv(u)
m<-as.character(print(f))
kmax<-(2*c)-(3*d^2)/(4*e)
#
x<-seq(-20,5,length=100)
uqm<-x-predict(g,x)/kmax
pdf("uqm1.pdf")
plot(x,uqm,type="l",col="RED",ylab="update")
abline(0,1)
dev.off()
#
x<-seq(-.5,1.5,length=100)
```

```
uqm<-x-predict(g,x)/kmax
pdf("uqm2.pdf")
plot(x,uqm,type="l",col="RED",ylab="update")
abline(0,1)
dev.off()
#
x<-seq(-20,5,length=100)
kopt<-predict(h,x)-(predict(u,x)^2)/(3*predict(v,x))
bqm<-x-predict(g,x)/kopt
pdf("bqm1.pdf")
plot(x,bqm,type="l",col="RED",ylab="update")
abline(0,1)
dev.off()
#
x<-seq(-.5,1.5,length=100)
kopt<-predict(h,x)-(predict(u,x)^2)/(3*predict(v,x))
bqm<-x-predict(g,x)/kopt
pdf("bqm2.pdf")
plot(x,bqm,type="l",col="RED",ylab="update")
abline(0,1)
dev.off()
#
x<-seq(-20,5,length=100)
newt<-x-predict(g,x)/predict(h,x)
pdf("newt1.pdf")
plot(x,newt,type="l",col="RED",ylab="update")
abline(0,1)
dev.off()
#
x<-seq(-.5,1.5,length=100)
newt<-x-predict(g,x)/predict(h,x)
pdf("newt2.pdf")
plot(x,newt,type="l",col="RED",ylab="update")
abline(0,1)
```

```
dev.off()
```

References

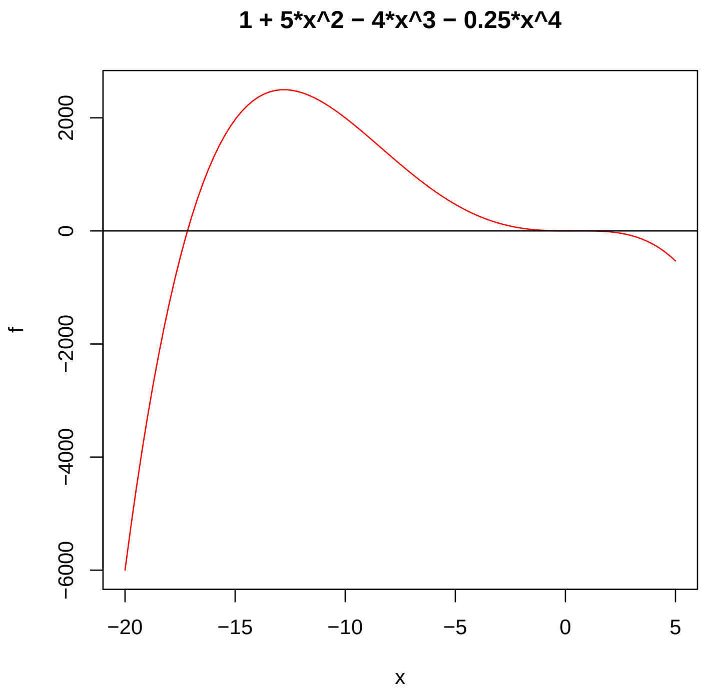
A. M. Ostrowski. *Solution of Equations and Systems of Equations*. Academic Press, New York, N.Y., 1966.

W. I. Zangwill. *Nonlinear Programming: a Unified Approach*. Prentice-Hall, Englewood-Cliffs, N.J., 1969.

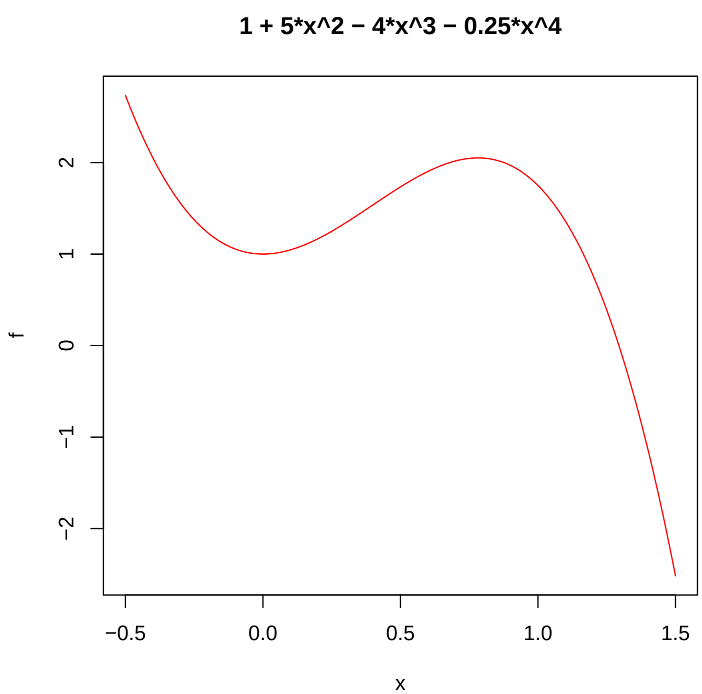
Department of Statistics, University of California, Los Angeles, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu

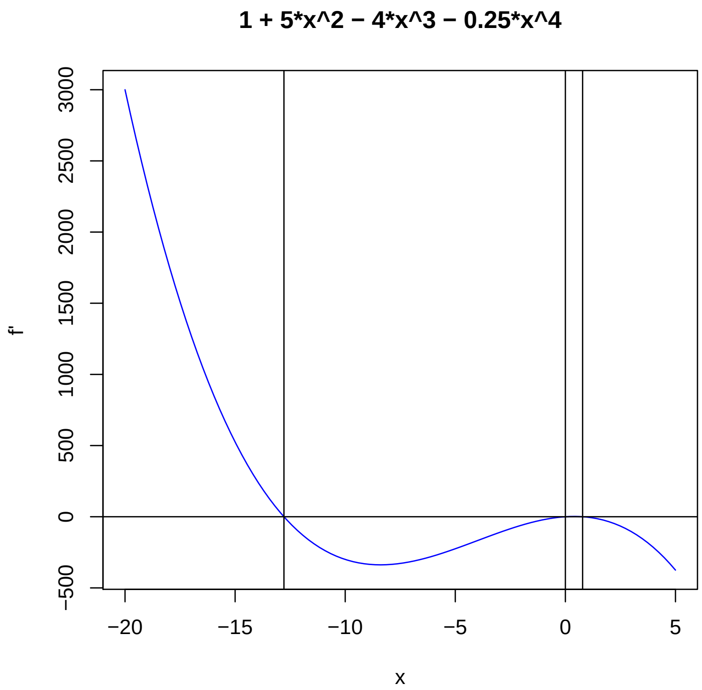


(a) Global.

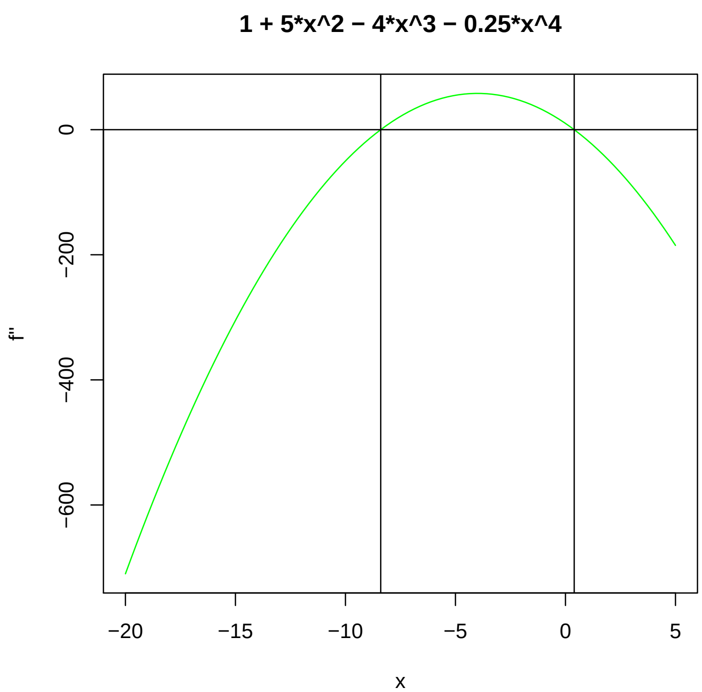


(b) Local.

FIGURE 1. Quartic Function.



(a) First.



(b) Second.

FIGURE 2. Derivatives.

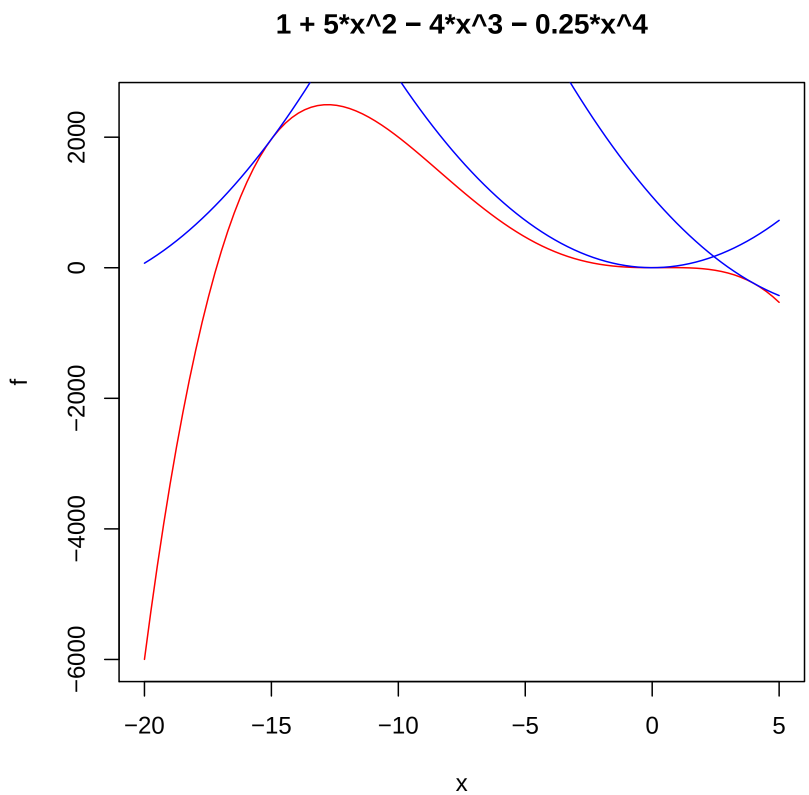


FIGURE 3. Uniform quadratic majorization at (-15,0,4).

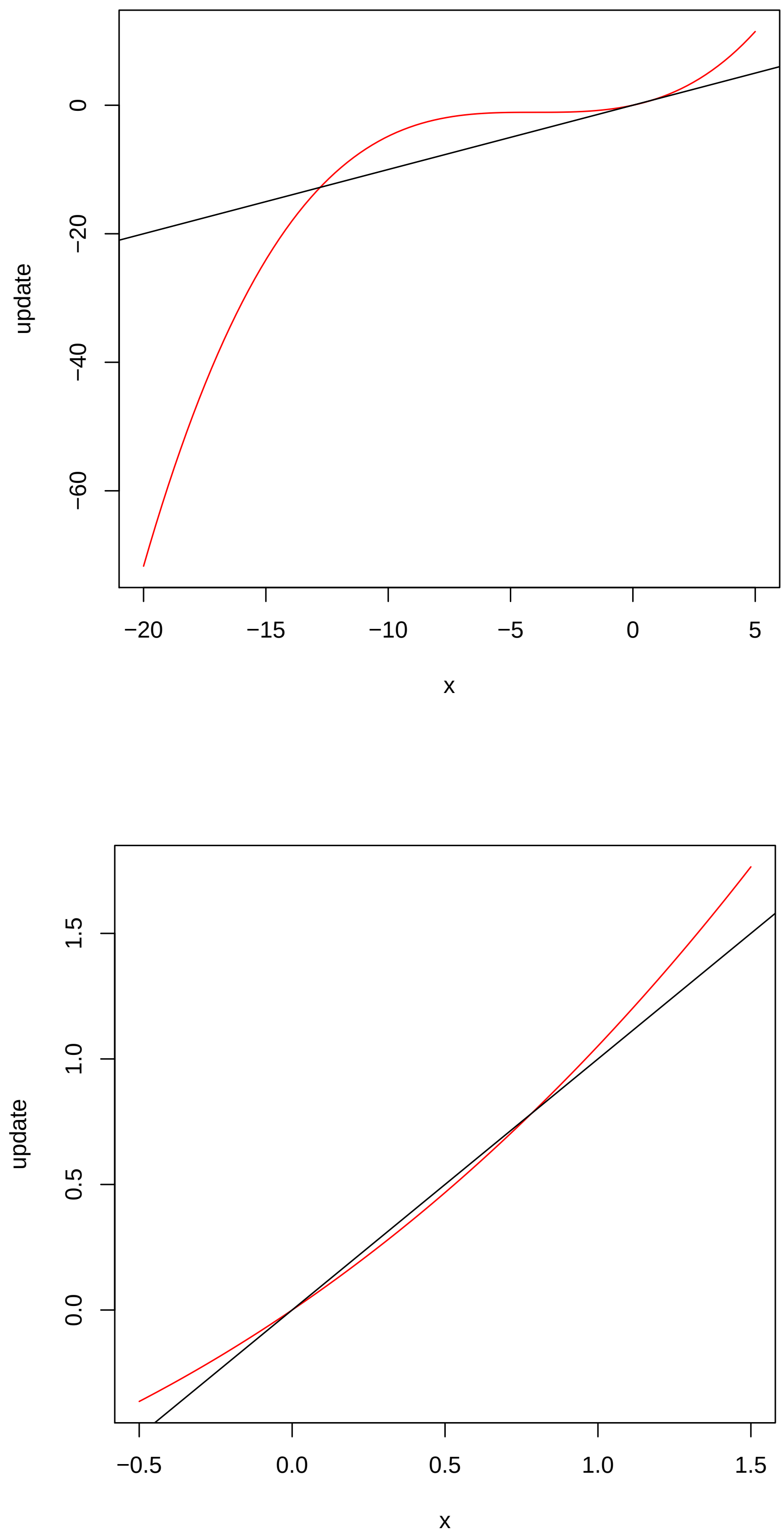


FIGURE 4. UQM Algorithmic Map (top), Zooomed in (bottom)

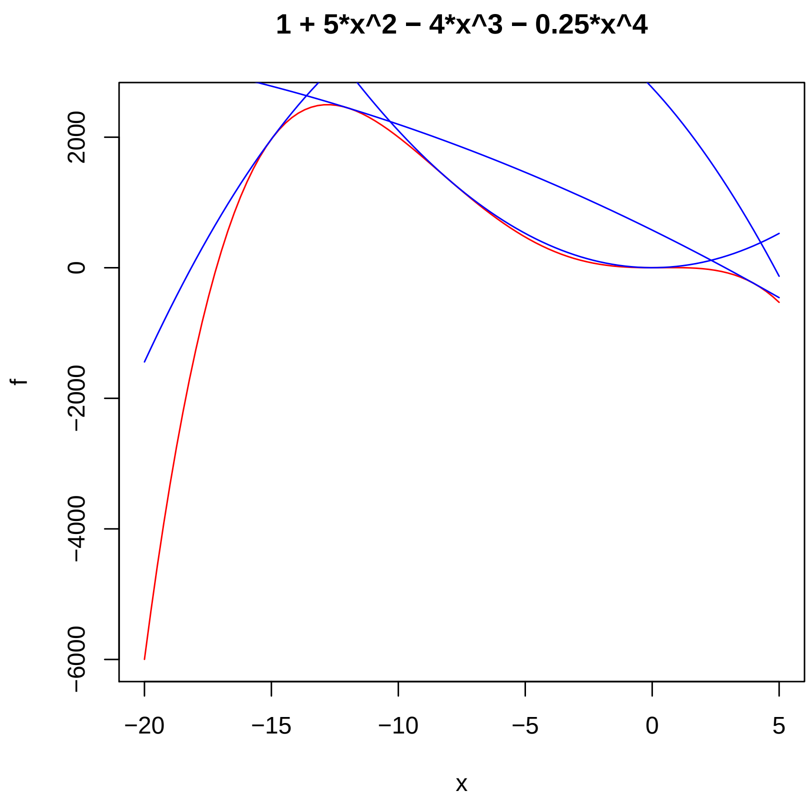


FIGURE 5. Best quadratic majorization at (-15,0,4).

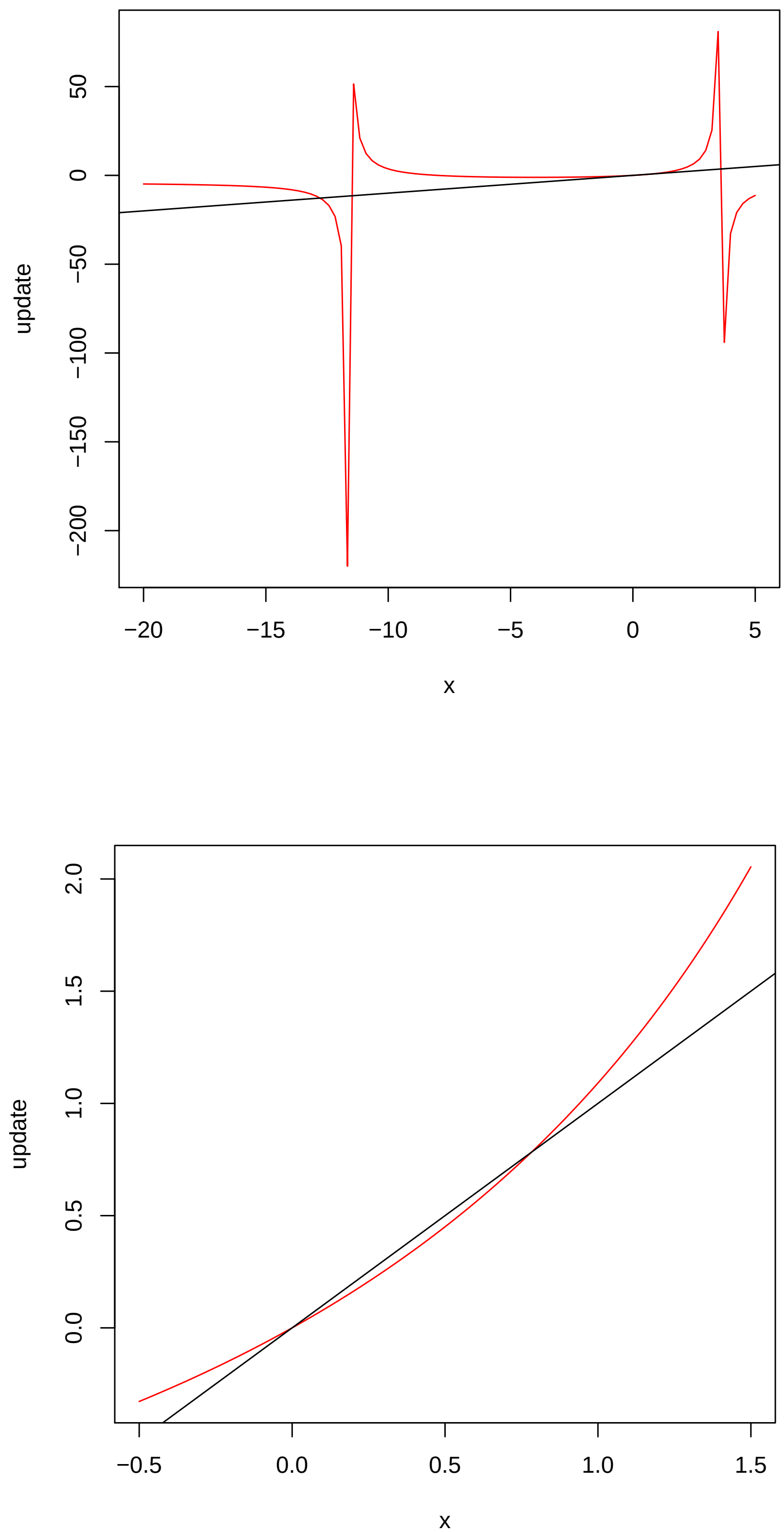


FIGURE 6. BQM Algorithmic Map (top), Zooomed in (bottom)

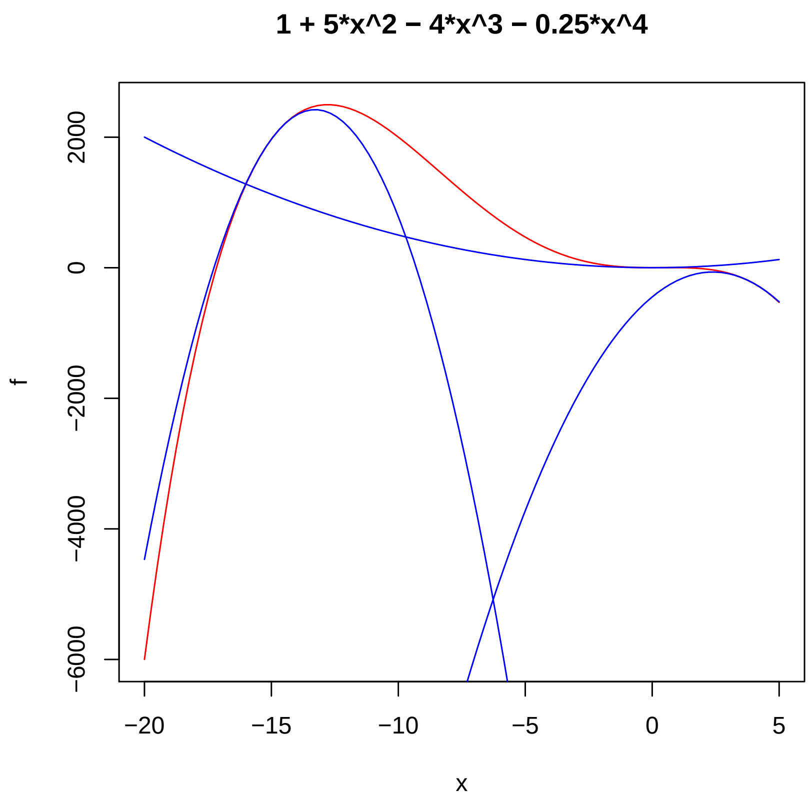
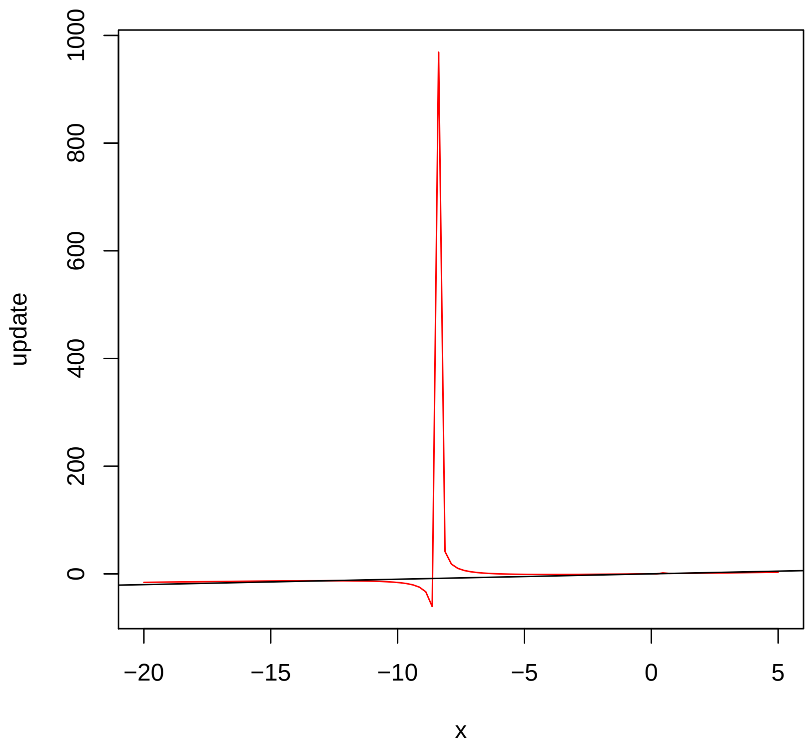


FIGURE 7. Newton quadratic approximation at (-15,0,4).



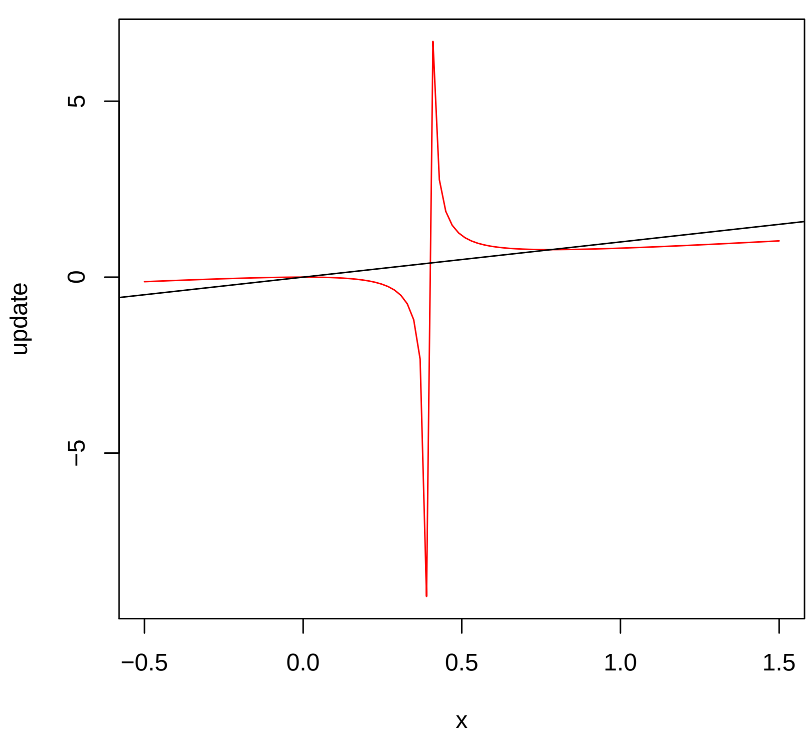


FIGURE 8. Newton Algorithmic Map (top), Zooomed in (bottom)