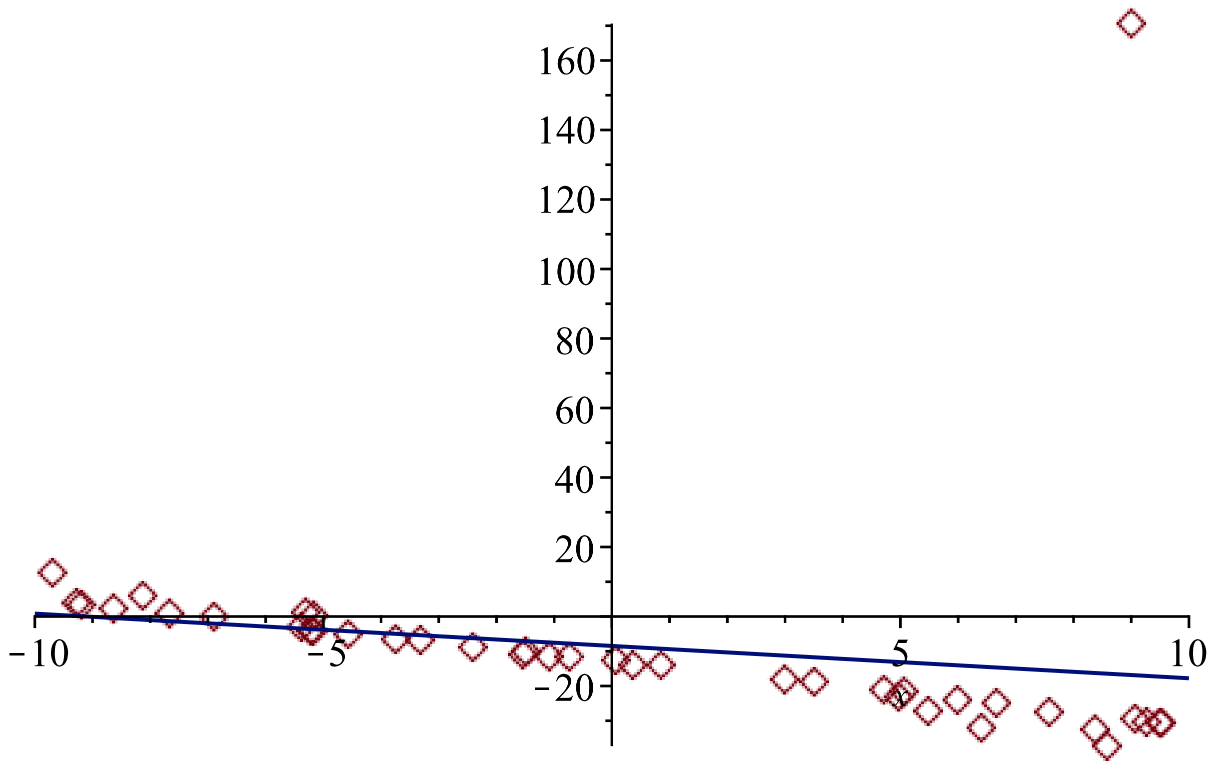
2019-09-26

ExecuteFromWeb

```
> ExecuteFromWeb("http://www.math.stonybrook.edu/~scott/mat331.fall19/daily/extras/bad_line.txt");
loaded 40 points into lpts.
> with(CurveFitting):
> bline:=LeastSquares(lpts,x)
```

$$bline := -8.47217789980814 - 0.930101594702111\, x \quad (1)$$

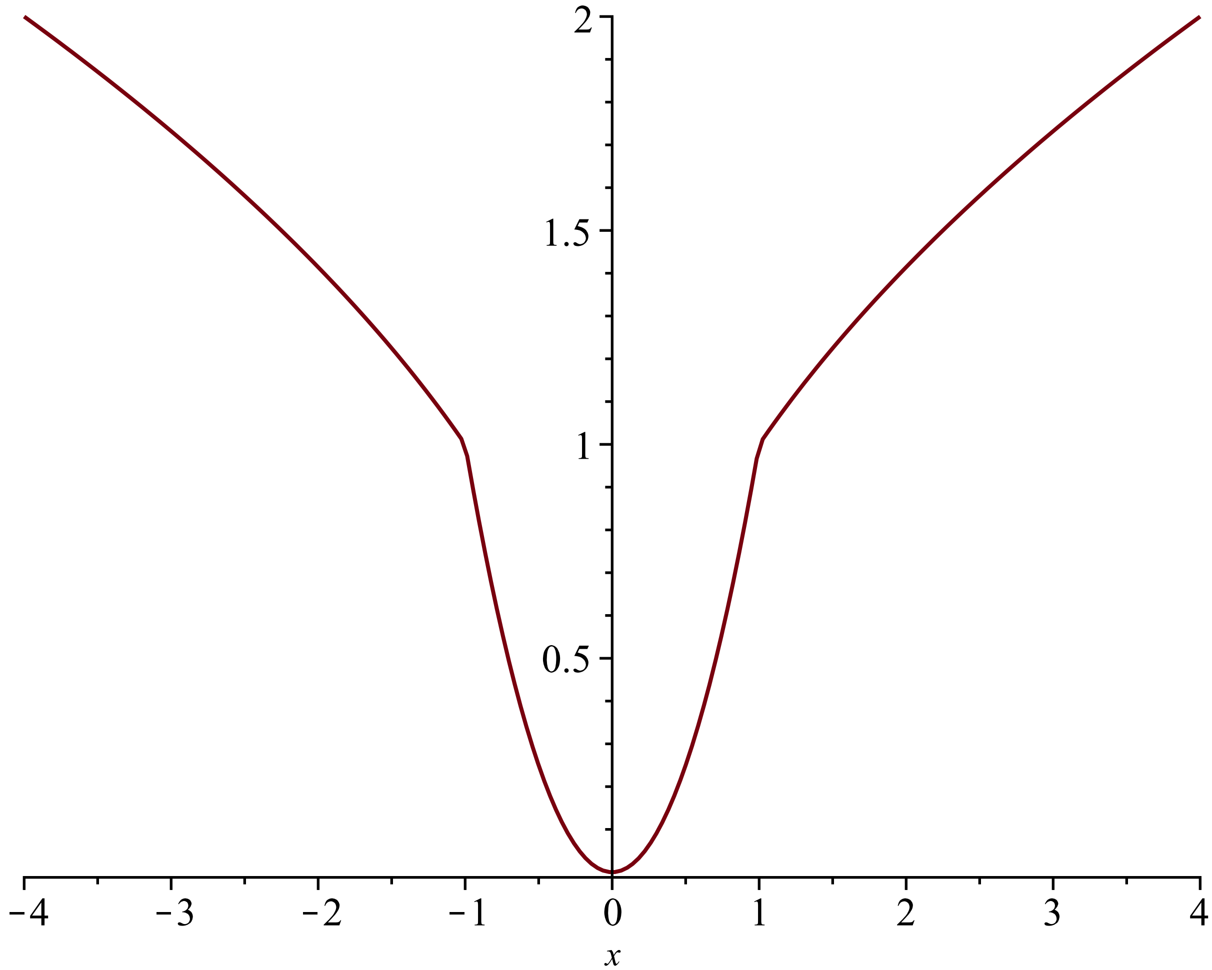
> $plot([lpts, bline], x = -10..10, style = [point, line], symbolsize = 20)$



> $obj := x \rightarrow piecewise(x < -1, \mathrm{sqrt}(|x|), x < 1, x^2, \mathrm{sqrt}(x))$

$$obj := x \mapsto \begin{cases} \sqrt{|x|} & x < -1 \\ x^2 & x < 1 \\ \sqrt{x} & otherwise \end{cases} \quad (2)$$

> $plot(obj(x), x = -4..4)$



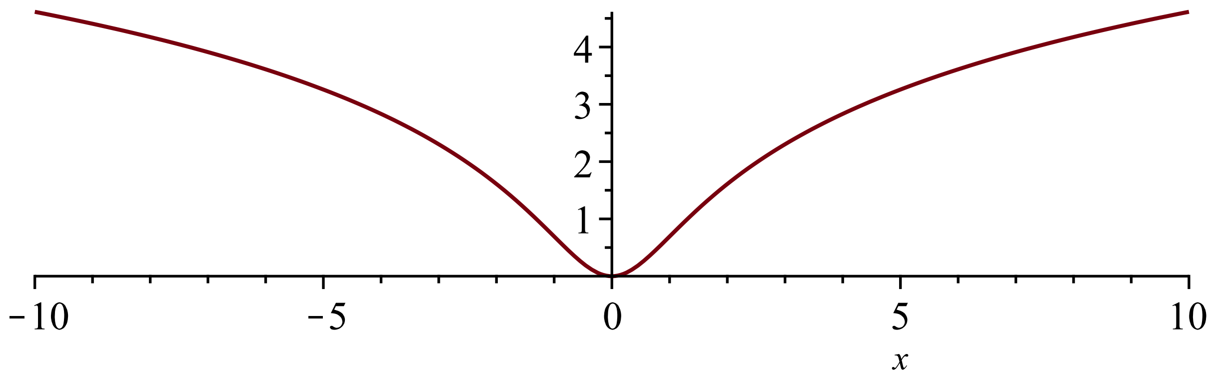
> *series*(ln(1 + *x*), *x*)

$$x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 + \mathrm{O}(x^6) \quad \textbf{(3)}$$

> *series*(ln(1 + *x*^2), *x*, 8)

$$x^2 - \frac{1}{2}x^4 + \frac{1}{3}x^6 + \mathrm{O}(x^8) \quad \textbf{(4)}$$

> *plot*(ln(1 + *x*^2), *x* =- 10 ..10, *scaling* = *constrained*)



Let's build a "least squares like process" using ln(1+dist^2) instead of dist^2

> $obj := dist \rightarrow \ln(1 + dist^2)$

$$obj := dist \mapsto \ln(1 + dist^2) \quad \textbf{(5)}$$

p - [x,y]. What is the contribution to the function to minimize given by p? p[1]=x, p[2]=y

> $err := (p, m, b) \rightarrow obj(\ (m \cdot p[1] + b)\ - p[2])$

$$err := (p, m, b) \mapsto obj(m\, p_1 + b - p_2) \quad \textbf{(6)}$$

> $err([2, 3], m, b)$

$$\ln(1 + (b + 2\, m - 3)^2) \quad \textbf{(7)}$$

> $err(lpts[1], -5, 1)$

$$5.964260646 \quad \textbf{(8)}$$

> $err(lpts[2], m, 1)$

$$\ln(1 + (-3.748081610\, m + 7.522580699)^2) \quad \textbf{(9)}$$

> $E := (lpts, m, b) \rightarrow add(\ err(\ lpts[i], m, b), i = 1..nops(lpts))$:

Warning, `i` is implicitly declared local to procedure `E`

> $E(lpts, -2, -5)$

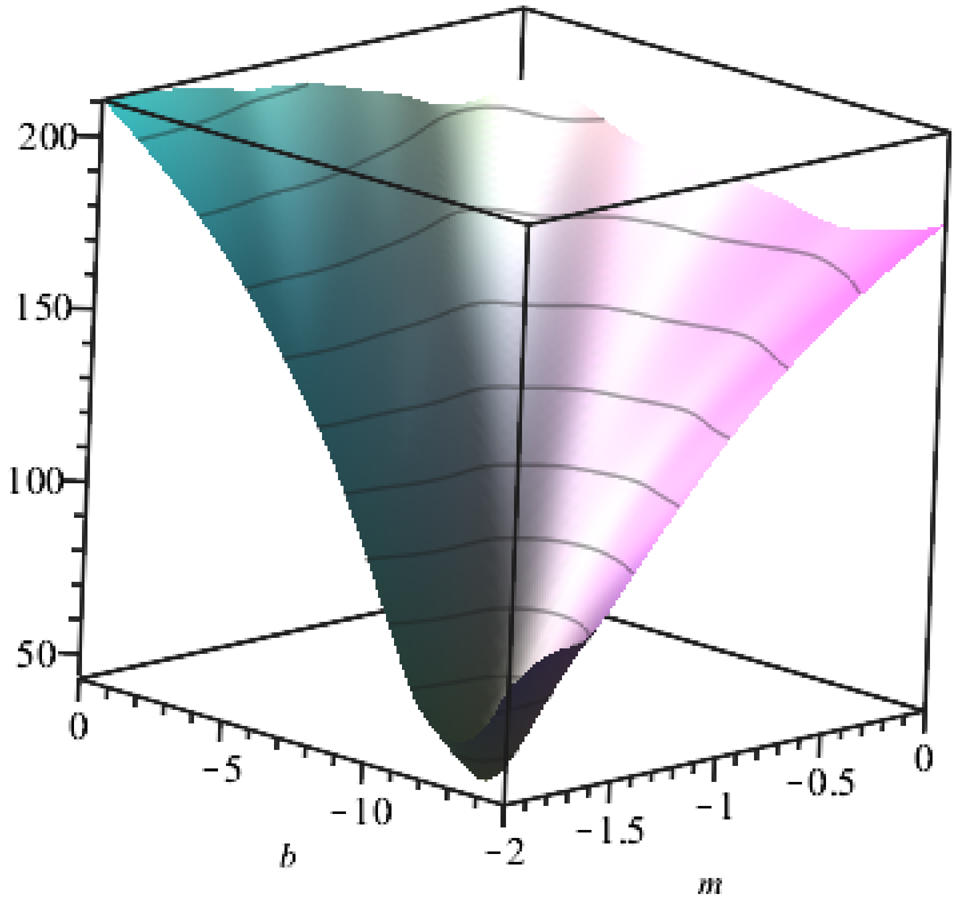
$$171.3977370 \quad \textbf{(10)}$$

> $E(lpts, -1.5, -5)$

(11)

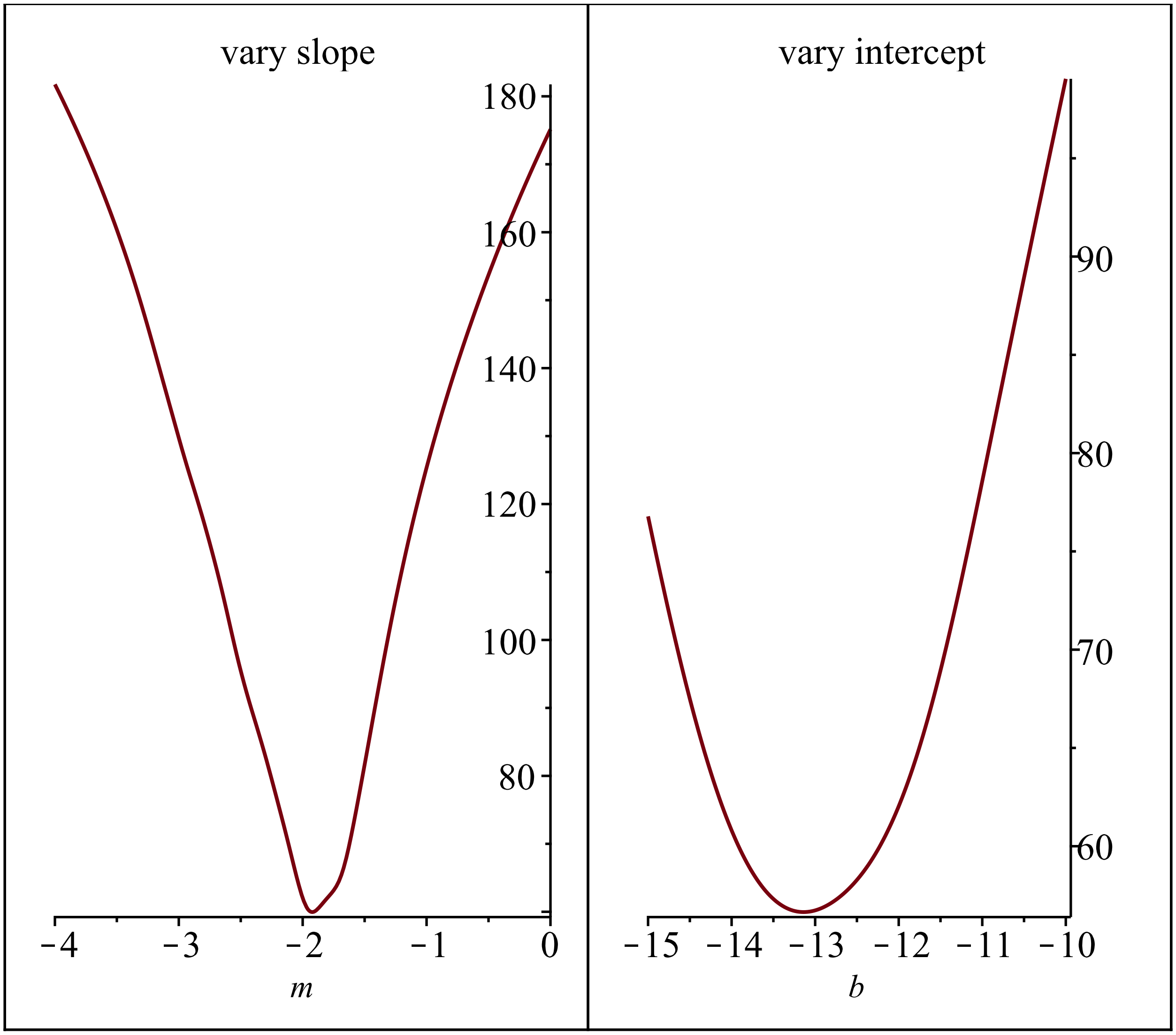
$$168.3525048 \qquad (11)$$

> $plot3d(E(lpts, m, b), m = -2 ..0, b = -15 ..0, style = patchcontour)$



> $with(plots):$

> $display(\ \langle plot(E(lpts, m, -12), m = -4 ..0, title = \text{"vary slope"}) \mid plot(E(lpts, -2, b), b = -15 ..-10, title = \text{"vary intercept"})\ \rangle\ )$



to find the min, take partials, set to 0, solve.

> $solve(\ \{diff(E(lpts, m, b), m) = 0, diff(E(lpts, m, b), b) = 0\})$

takes too long for me. Let's try a numerical solution

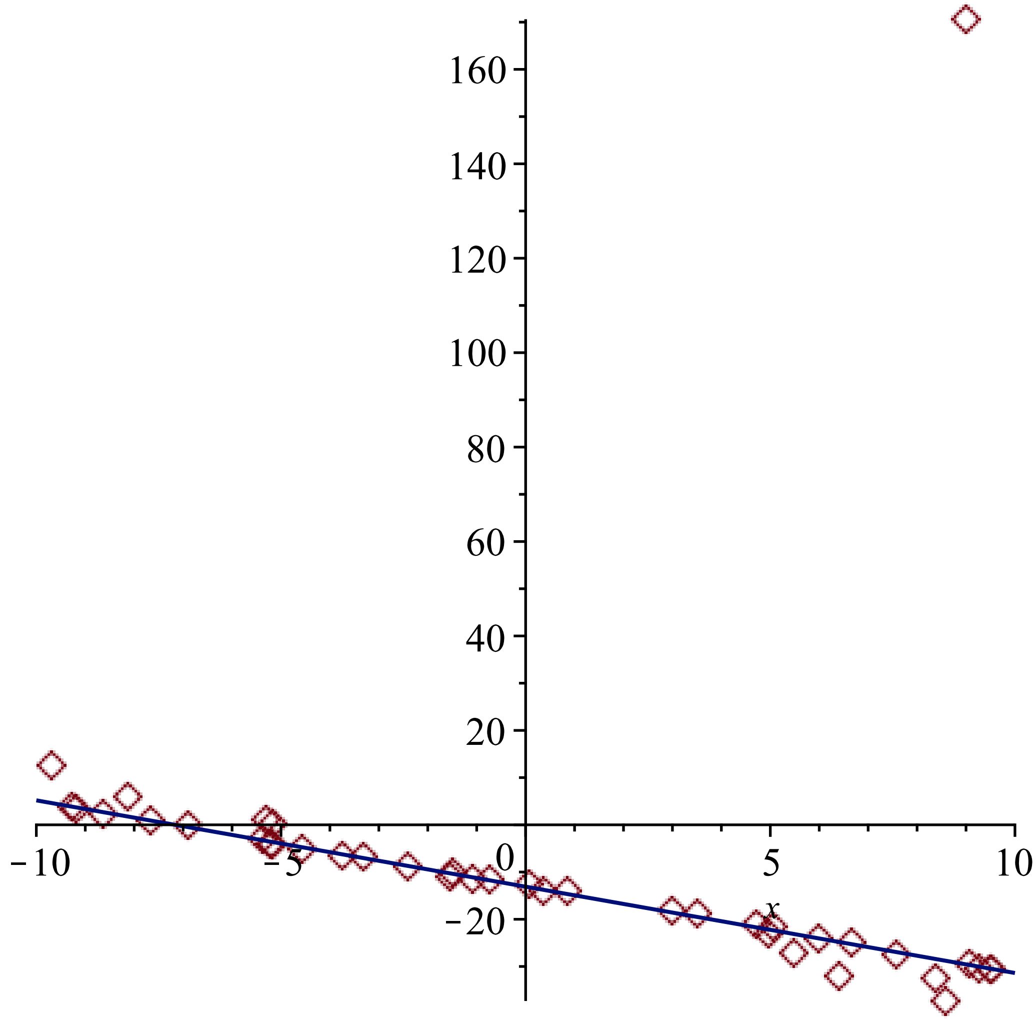
> $mb := fsolve(\ \{diff(E(lpts, m, b), m) = 0, diff(E(lpts, m, b), b) = 0\}, \{m = -4 ..0, b = -15 .. -12\})$

$$mb := \{b = -13.09953223, m = -1.828089260\} \tag{12}$$

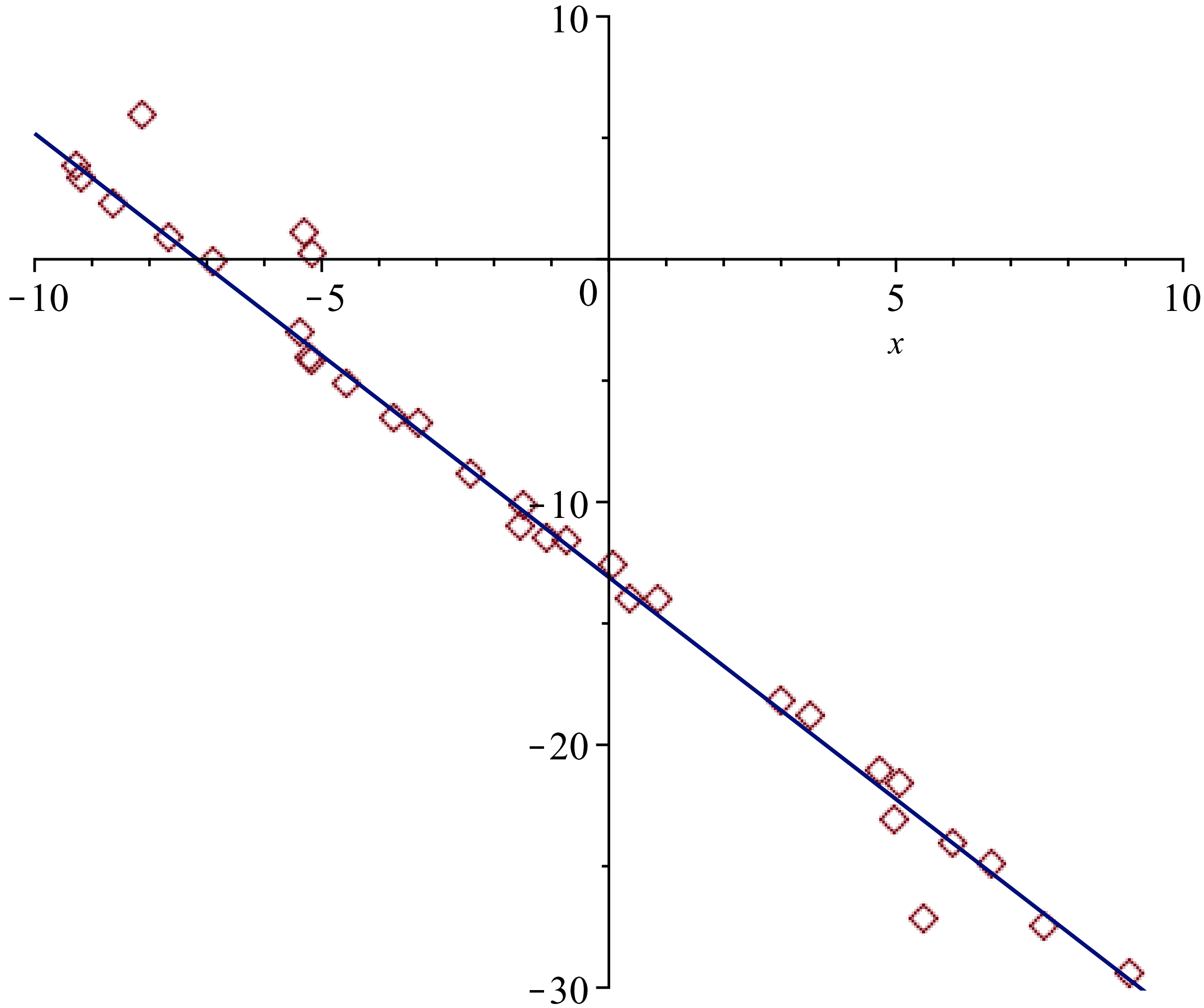
> $goodline := subs(mb, m \cdot x + b)$

$$goodline := -1.828089260\, x - 13.09953223 \tag{13}$$

> $plot([lpts, goodline], x = -10 ..10, style = [point, line], symbolsize = 20)$



> *plot*([*lpts*, *goodline*], *x* =− 10 ..10, *style* = [*point*, *line*], *symbolsize* = 20, *view* = [− 10 ..10, − 30 ..10])



Let's write all of this as a procedure.

```
> lnfit:=proc(pts, mguess, bguess)
     local obj,err, E,i,mb;
     obj:=x->ln(1+x^2);
     err:=(p,m,b)->obj( m*p[1]+b - p[2]);
     E:= (pts,m,b) -> add( err(pts[i],m,b), i=1..nops(pts));
     mb:=fsolve( {diff(E(pts,m,b), m)=0,  diff(E(pts,m,b),b)=0},
                 {m=mguess, b=bguess});
     return( subs(mb, m*x+b));
  end:
```

> $lnfit(lpts, -2..0, -15..-10)$

$$-1.828089260\,x - 13.09953223 \quad \mathbf{(14)}$$

>