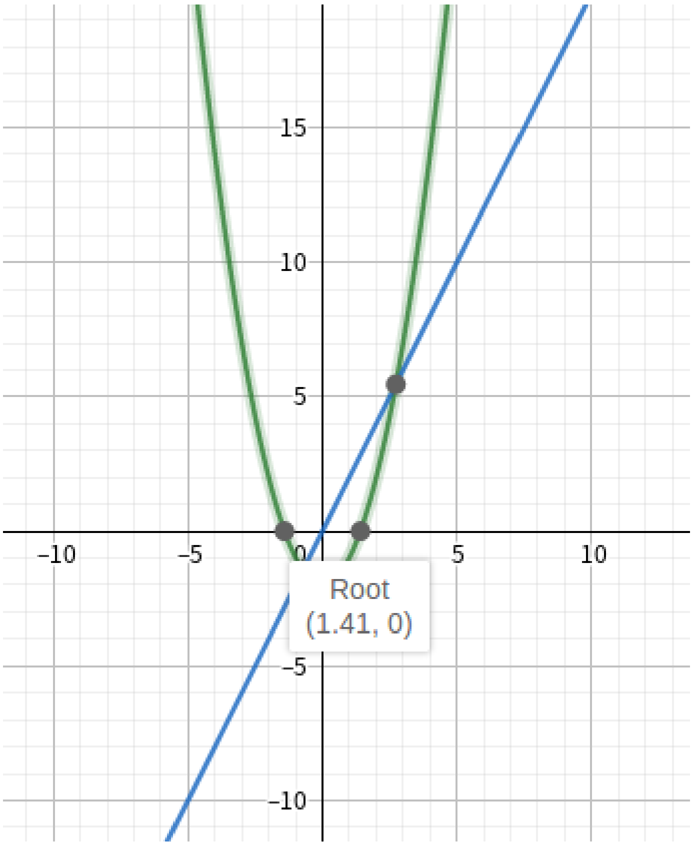
牛顿迭代法开某数的平方：

可以设：$f(x) = x^2 - n$

他的一阶导：$g(x) = 2x$

然后我们搞一条f(x)的切线l, 设l为: $y_1 - y_0 = k * (x_1 - x_0)$

当我们要求的点是开根号的点， 那么那个点要收束于相应精度的 $y_1 = 0$时，此时的$|x_1|$就是答案。



[绘图]

证明：

$y_1 - y_0 = k * (x_1 - x_0)$

当$y1$为0时：

$-y_0 = k(x_1 - x_0)$

$-\frac{y_0}{k} = x_1 - x_0$

又因为：

$y_0 = f(x_0)$, $k = g(x_0)$.

$$-\frac{f(x_0)}{g(x_0)} = x_1 - x_0$$

所以有：

$$x_1 = x_0 - \frac{f(x_0)}{g(x_0)}$$

所以下一次的迭代x_n只需要上一次的x_{n-1} ,那么每次计算都是一次迭代， 最终我们得到一个在相应精度内的近似值。

代码：

```
#define EPS 1e-7

double N(double x, double n) {return x * x - n;}
double N_p(double x) {return 2 * x;}

double newton(double a, double (*n)(double, double), double (*n_p)(double)) {
    double x = 1.0;
    while (fabs(n(x, a)) > EPS){
        x -= n(x, a)/n_p(x);
    }
    return x;
}
```