

# 第5回 課題

1. 正整数  $n$  の  $m$ 乗根 をNewton法によって求める関数 `kadai1(m, n)` を作れ. 「整数  $n$  の  $m$ 乗根」とは $m$ 乗して整数 $n$ に等しくなる実数のこと、方程式  $f(x) = x^m - n = 0$  の実数根となっている.  $n$ が偶数の場合、正負2つの答えが存在するが、正の解を返すようにせよ.

2. 連立方程式  $x^2 + y^2 = 1, ax + by = 1$  の2つ解  $(x_1, y_1), (x_2, y_2)$ をNewton法で計算して配列`[[x1,y1], [x2, y2]]`として返す関数 `kadai2(a, b)` を作れ. 解がない場合には`null`を返すこと. この問題の場合、 $a^2 + b^2 < 1$  のとき解を持たず、その他の場合、かならず解をもつ (要確認). 2次方程式の解の公式を使っても構わないが、平方根を計算する関数`Math.sqrt`は用いずにNewton法で解くこと.



# 第5回課題 解答例 (1)



1. の問題については純粹にニュートン法を実行すればよいが、正負の部分を気にする必要がある。まず方程式は、

$$f(x) = x^m - n = 0$$

であり、この方程式を解くための反復公式は、以下のようになる。

$$x' = x - \frac{f(x)}{f'(x)} = x - \frac{x^m - n}{mx^{m-1}}$$

これをプログラムとして表現すれば、つぎのスライドのようなる。

## 第5回課題 解答例 (2)

```
function kadai1(m, n){
  function expr(a, n){
    if (n == 0) return 1.0
    else {
      var x = expr(a, Math.floor(n / 2))
      if (n % 2 == 0) return x * x
      else return x * x * a
    }
  }
  function f(x){ return expr(x, m) - n;}
  function fd(x){ return expr(x, m - 1) * m;}
  var x = 1.0
  var eps = 1.0e-6
  while (true){
    var v = f(x);
    if (Math.abs(v) < eps) return x
    x = Math.abs(x - v / fd(x))
  }
}
```

内部関数exprは累乗  $a^n$  を計算するプログラムである。

# 第5回課題 解答例 (3)

2. については、グラフを書くと以下のようなになる。

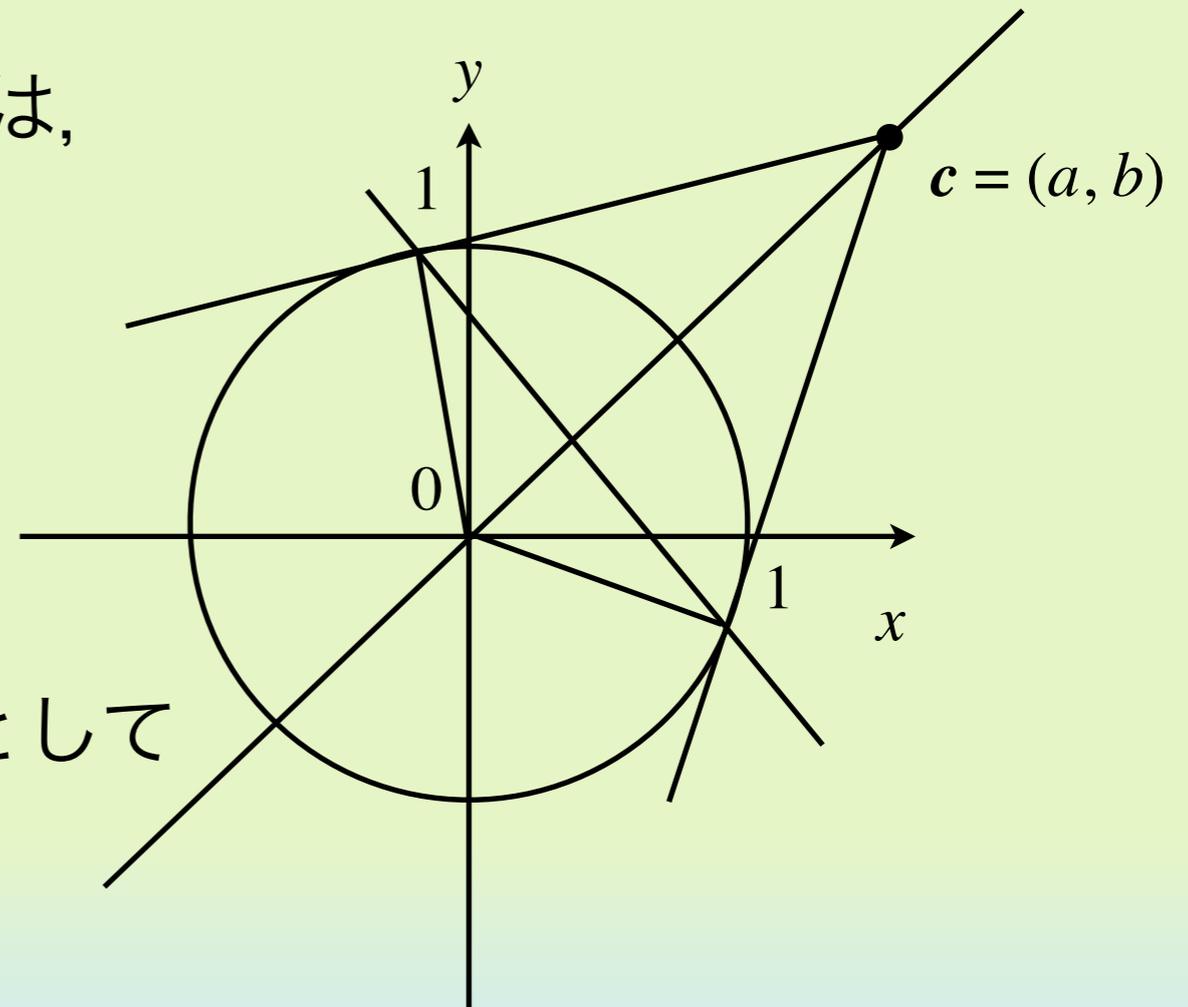
$$x^2 + y^2 = 1$$

$$ax + by = 1$$

この場合、与えられた直線と円の交点は、点cからこの円に引いた2本の接線と円の交点となる。

$$x = \frac{1 - by}{a}$$

とおく、ただし、 $a = 0$ の場合は例外として別途考える。このとき、



## 第5回課題 解答例 (4)

$$y^2 - \frac{2b}{a^2 + b^2}y + \frac{1 - a^2}{a^2 + b^2} = 0$$

が得られる。この方程式をNewton法で解く。解が2つあることは分かっているなので、2次曲線の軸の右側と左側から出発して、それぞれの解を得る。軸は  $y = b / (a^2 + b^2)$  である。

また、 $a = 0$  の場合には、 $y = 1/b$  と決定するので、方程式  $x^2 + 1/b^2 - 1 = 0$  を解けば良い。この方程式の2つの解は絶対値が同じで正負が反対になっているので、片方を求めて、もう片方をその符号を変える。

# 第5回課題 解答例 (5)

プログラムは以下のとおり.

```
function kadai2(a, b){
  var m = a * a + b * b
  var eps = 1.0e-6
  function f(y){ return y * y - 2 * b / m * y + (1 - a * a) / m }
  function fd(y){ return 2 * y - 2 * b / m }
  function g(x){ return x * x + 1 / b / b - 1 }
  function gd(x){ return 2 * x }
  function iter(){
    while (Math.abs(f(y)) > eps){
      y = y - f(y) / fd(y)
    }
    return y
  }
  if (a === 0){
    x = 100
    while (Math.abs(g(x) > eps)){
      x = x - g(x) / gd(x)
    }
    return [[x, 1 / b], [-x, 1 / b]]
  } else {
    var y = a * b / m - 100
    var y1 = iter()
    y = a * b / m + 100
    var y2 = iter()
    return [[(1 - b * y1) / a, y1], [(1 - b * y2) / a, y2]]
  }
}
```