第5回課題 解答例

- 係数がa, b, c, d で与えられた 3次方程式の根を求める．そのためにまず，2次方程式の根を求める関数をつくる.
- さらにその前にバイナリサーチを行うプログラムを作る．ただし，fは与えられた関数である.

```
function bsearch(l, r, f){
    var EPS = 1.0e-10
    var L = f(l)
    while (Math.abs(r - l) > EPS){
        m = (l + r) / 2
        if (f(m) * L > 0) l = m
        else r = m
    }
    return r
}
```

2次方程式の根を求める

まず，2次方程式の根を求める関数 solver2(a, b, c)を定義する． まず判別式を計算して，負であれば解なしということで[]を返す．そうでなければ，2つの区間について2分法で解く．

```
function solver2(a, b, c){
    function f(x){ return (a * x + b) * x + c }
    var d = b * b - 4 * a * c
    var A = Math.sqrt(a * a + b * b + c * c) / a
    if (d < 0) return []
    else{
        var mmm = -b / a / 2
        return [bsearch(-A, mmm, f), bsearch(mmm, A, f)]
    }
}
```

3次方程式の根を求める

- 2次方程式を解く関数を用いて3次方程式を解く．導関数f'(x)の根をsolver2で求めてそれの結果により分岐する.

```
function solver3(a, b, c, d){
    function f(x){ return ((a * x + b) * x + c) * x + d }
    var A = Math.sqrt(a * a + b * b + c * c + d * d) / a
    var ddiff = solver2(3 * a, 2 * b, c)
    if (ddiff == []){
        return [bsearch(-A, A, f)]
    } else {
        var m1 = ddiff[0]
        var m2 = ddiff[1]
        if (f(m1) * f(m2) < 0)
           return [bsearch(-A, m1, f), bsearch(m1, m2, f),
                   bsearch(m2, A, f)]
        else
           return [bsearch(-A, A, f)]
    }
}
```

プログラム全体（１）

- 全体を関数 kadai でくるむ.

```
function kadai(a, b, c, d){
    function bsearch(l, r, f){
        var EPS = 1.0e-10
        var L = f(l)
        while (Math.abs(r - l) > EPS){
            m = (l + r) / 2
            if (f(m) * L > 0) l = m
            else r = m
        }
        return r
    }

    function solver2(a, b, c){
        function f(x){ return (a * x + b) * x + c }
        var d = b * b - 4 * a * c
        var A = Math.sqrt(a * a + b * b + c * c) / a
        if (d < 0) return []
        else{
            var mmm = -b / a / 2
            return [bsearch(-A, mmm, f), bsearch(mmm, A, f)]
        }
    }
```

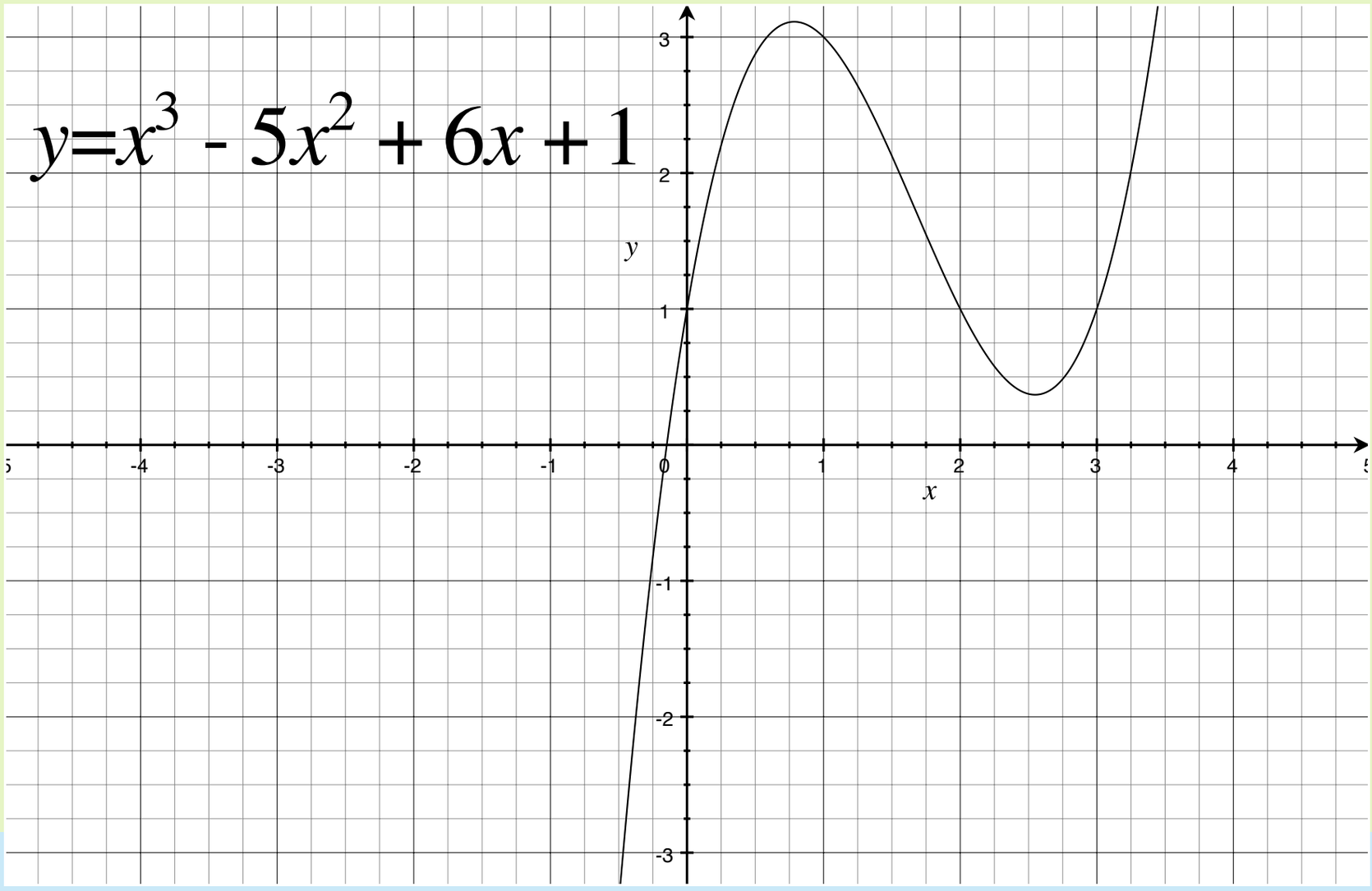
プログラム全体（2）

```
function solver3(a, b, c, d){
    function f(x){ return ((a * x + b) * x + c) * x + d }
    var A = Math.sqrt(a * a + b * b + c * c + d * d) / a
    var ddiff = solver2(3 * a, 2 * b, c)
    if (ddiff == []){
        return [bsearch(-A, A, f)]
    } else {
        var m1 = ddiff[0]
        var m2 = ddiff[1]
        if (f(m1) * f(m2) < 0)
            return [bsearch(-A, m1, f), bsearch(m1, m2, f),
                    bsearch(m2, A, f)]
        else
            return [bsearch(-A, A, f)]
    }
}
return solver3(a, b, c, d)
}
```

実行例 (1)

- 3次方程式を解いてみる.

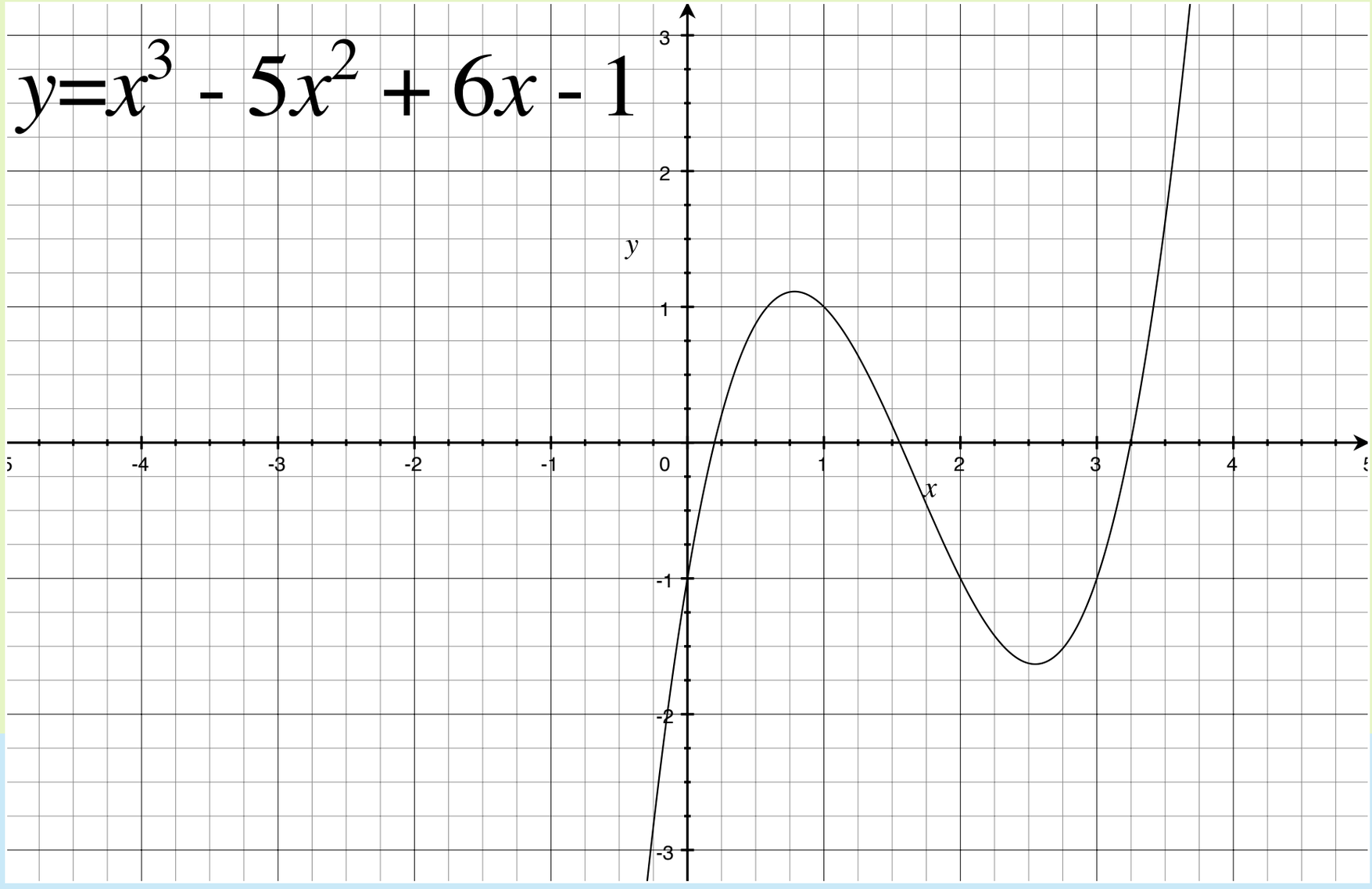
```
solver3(1, -5, 6, 1) = [-0.14789903567952864]
```



実行例 (2)

- もう一つ3次方程式を解いてみる.

```
solver3(1, -5, 6, -1) = [ 0.19806226421081438, 1.554958132096282
2, 3.246979603774226 ]
```



実行例 (3)

- 念のため2次方程式も解いてみる.

```
solver2(1, -5, 6) = [2.0000000000017919, 3.0000000000671654]
```