

## ① 予備知識

2 階微分可能であり、2 階導関数が連続な (すなわち  $C^2$  級) 関数  $f(x)$  の場合、次の極値判定法が使えます。

- (1)  $f'(a) = 0$  かつ  $f''(a) > 0$  ならば、関数  $f(x)$  は  $x = a$  において極小値  $f(a)$  をとる。
- (2)  $f'(a) = 0$  かつ  $f''(a) < 0$  ならば、関数  $f(x)$  は  $x = a$  において極大値  $f(a)$  をとる。

この判定法は「2 階微分テスト」と呼ばれています。残念ながら 2 階微分係数が 0 になってしまう場合は、この判定法では極値をとるか否か判定できません。

## ② バッチ処理

$x$  を変数とする関数を引数にとり、極値を返す関数を定義してみたいと思います。

```

1  rmmpt(f) := block([fx, fxx, s, t,                                     - rmmpt.mac -
2      Lz: [], Lp: [], Ln : [], realonly: true ],
3      fx: diff(f, x),
4      fxx: diff(f, x, 2),
5      s: algsys([fx], [x]),
6      if length(s) = 0 then
7          return("Critical Point Not Found")
8      else (
9          for i: 1 thru length(s) do (
10             t: subst(s[i], fxx),
11             if t = 0 then
12                 Lz: cons(s[i][1], Lz)
13             elseif t > 0 then
14                 Lp: cons(subst(s[i], [x, f]), Lp)
15             else
16                 Ln: cons(subst(s[i], [x, f]), Ln)
17         ),
18         if length(Lz) # 0 then (
19             print("UNKNOWN:"),
20             for i in Lz do disp(i)
21         ),
22         if length(Lp) # 0 then (
23             print("Relative Minimum Point:"),

```

```

24         for i in Lp do disp(i)
25     ),
26     if length(Ln) # 0 then (
27         print("Relative Maximum Point:"),
28         for i in Ln do disp(i)
29     )
30 )
31 );

```

2行目のLz、Lp、Lnは、それぞれ2階微分係数が0の点、正の点、負の点を格納するリストです。また、1階微分係数が0になる $x$ を求める際、求解の範囲を実数に制限するため、変数 `realonly` に `true` を代入しています。5行目で方程式  $f'(x) = 0$  を解いています。

6行目と7行目は、方程式  $f'(x) = 0$  の解を求めることが出来なかった場合の処理です。解が見つかった場合は、各解毎に2階微分係数を調べます(9行目のforループ)。

12行目は、2階微分係数が0となる(すなわち極値判定不能な) $x$ をリストLzに蓄積しています。14行目は、2階微分係数が正の(すなわち極小となる)点をリストLpに蓄積しています。16行目は、2階微分係数が負の(すなわち極大となる)点をリストLnに蓄積しています。

18行目以降は、結果を出力する処理です。

```

(%i1) load("rmmpt.mac");
(%o1)                                     rmmpt.mac
(%i2) rmmpt(3*x^4 - x^2 + 1);
Relative Minimum Point:
          1      11
[- -----, --]
      sqrt(6)  12

          1      11
[-----, --]
      sqrt(6)  12

Relative Maximum Point:
          [0, 1]

(%o2)                                     done

```