J. of the Chosun Natural Science
Vol. 3, No. 4 (2010) pp. 241 – 245

어떤 최대최소문제들에 대한 컴퓨터대수체계(CAS)의 이용

김선홍†

How to Use CAS to Solve Certain Minmax Problems

Seon-Hong Kim†

Abstract

In this paper, we study how to use CAS to solve certain minmax problems. More specifically, we reduce some to the problems about the packing of certain convex sets in the plane, and use Recognize algorithm to obtain good hypotheses for results. This method can be widely used to solve same types of mathematical problems.

Key words : CAS, Mathematica, Minmax Problems, Recognize

1. 서 론

1826년에 Abel은 “5차 이상의 방정식은 일반적으로 대수적으로 풀 수 없다”라는 사실을 발표하였고, 이로 인하여 자연스럽게 다항식의 근을 찾기 위한 수치해석적인 방법(Numerical method)을 찾는 연구들이 시작되었다. 20세기에 들어서는 컴퓨터의 도입과 더불어 새로운 많은 수치해석적인 방법들이 발견되었는데, 이와 더불어 1980년대 초부터는 컴퓨터가 소형화되고 수학 또는 수학교육에서 문제해결력이 강조되면서 컴퓨터를 이용한 수치계산, 시뮬레이션 등의 방법으로 컴퓨터를 문제해결에 활용하는 연구가 활발히 진행되었다.

컴퓨터 대수 체계(CAS)란 컴퓨터를 이용하여 수학에 관련된 문제를 기호로 처리하여 연산을 수학하는 시스템으로, 수치계산과 대수적 연산을 수행하고 필요한 경우에 연산의 결과를 그래픽으로 처리하는 소프트웨어를 통칭한다. CAS는 1960년경에 시작되어 현재는 수학을 비롯한 제반 과학과 공학의 연구에 널리 사용되어지고 있으며, 대표적인 상업 프로그램은 Mathematica와 Maple등이 있다. 이 논문에서는 극값의 문제들(extremal problems)과 관련된 어떤 수학문제의 풀이에 CAS가 도움을 주는 예들을 소개한다. 여기서 소개할 두 가지 방법은 이 논문에서 언급할 문제들뿐만이 아니라 다양한 극값문제들에 유용하기에 그 중요성이 있다. 논문에서 사용할 CAS는 Mathematica 5.1이며 기본적인 명령어는 설명 없이 사용되어질 것이다. 현재 최신의 Mathematica는 Mathematica 8이며 적어도 버전 5.1이상에서는 이 논문에서 사용될 명령어들이 동일함을 밝혀둔다.

평면에서의 극값의 문제들은 역사적으로 많이 있어 왔다. 한 예로, 저자는 [3]에서 평면에서의 단위벡터들에 대한 다음의 최대최소문제(minmax problem)를 해결하였다.

정리 1 정수 $N \geq 2$에 대하여

$$f(N) := \min_{a_i \text{real}} \max\left\{\left|\sum_{n=1}^{N} e^{ia_n}\right|, \left|\sum_{n=1}^{N} e^{iNa_n}\right|\right\}$$

라 하면, $f(N)$은 다음의 값을 갖는다.

(1) $f(2) = 1$

(2) $f(2) = 0$, $N \geq 4$

(3) $f(3)=\sqrt{2k+3} = 0.769292\cdots$, 여기서 k는 방정식 $1+2+(1+x+x^2+x^3)=0$의 유일한 실근이다.

위 정리의 (1)과 (2)의 증명은 [3]에 상세히 설명되어져 있다. 는 일반성을 잃지 않고 다음과 같이 나타낼 수 있다.

*본 연구는 숙명여자대학교 2010학년도 교내연구비 지원에 의해 수행되었음
숙명여자대학교 수학과 (Department of Mathematics, Sookmyung Women's University)

†Corresponding author: shkim17@sookmyung.ac.kr
(Received : December 9, 2010, Revised : December 20, 2010, Accepted : December 23, 2010)

$f(3)=\min_{a,b\text{real}}\max\{|1+e^{ia}+e^{jb}|,|1+e^{i3a}+e^{j3b}|\}$

먼저 $f(3)$은 $|1+e^{ia}+e^{ib}|=|1+e^{i3a}+e^{i3b}|$을 만족하는 실수 a, b로부터 얻어짐을 추측할 수 있으며, 이에 대한 증명은 어렵지 않다[3]. 그런데

$$s(a,\ b):=|1+e^{ia}+e^{ib}|^2$$
$$2(\cos a+\cos b+\cos(a-b))+3,$$
$$t(a,\ b):=|1+e^{i3a}+e^{i3b}|^2$$
$$2(\cos 3a+\cos 3b+\cos 3(a-b))+3,$$

이므로 $f(3)$가 구해지기 위해서는

$$s(a,\ b)=t(a,\ b)$$

를 만족하는 a, b를 구해야한다. 따라서 $f(3)$를 구하는 문제는 곡선들

$$\cos a+\cos b+\cos(a-b)=k$$
$$\cos 3a+\cos 3b+\cos 3(a-b)=k$$

이 만나는 점 $(a,\ b,\ k)$를 찾는 문제로 바뀌어 질 수 있다. 여기서 $s(a,b)=t(a,b)\geq 0$이기 때문에 $t\geq -1.5$이다.

이 논문의 일차적인 목적은 [3]에서 고려되지 않았던 위의 $(a,\ b,\ k)$를 찾기 위한 CAS(특히 Mathematica)의 활용이다. 이런 종류의 문제는 다양하며, 결과를 얻기 위해서는 잘 다듬어진 가설이 필요하다. 이를 위해 CAS의 도움을 받으면 CAS이 없던 과거와 비교해 훨씬 쉽게 문제의 해결에 접근할 수 있다. 물론, 알맞은 가설이 정립된 후에는 엄밀한 수학적인 증명이 요구되어진다. 이 논문의 2절에서는 먼저, 증명이 될 가설을 세우기 위해 Mathematica의 패키지 중의 하나인 Number Theory라는 디렉토리의 안에 있는 Recognize의 활용을 볼 것이다. 그리고 3절에서는 라그랑지 승수법(Lagrage multiplier method)과 Gröbner 기저를 이용한 풀이로서의 Mathematica code를 제공할 것이다. 그러나 라그랑지 승수법이 이 문제에 대해서는 풀이에 적합하나 다른 여러 유사한 문제들에는 적용하기 힘듦을 상기시키며, 우리의 Recognize를 이용한, 가설을 세우기 위한 절차가 중요한 의미를 가짐을 강조할 것이다.

2. Recognize 패키지를 이용한 방법

Mathematica에서 Recognize 패키지안에 있는 명령어

Recognize[x, n, t]

는 실수 x를 근사근(an approximate zero)으로 갖는 차수가 n이하인 t를 변수로 갖는 정수계수 다항식을 구해준다. 따라서 이 패키지를 이용하면, 실수 x가 대수적인 수(algebraic number)일 때 그것을 근으로 갖는 가장 작은 차수의 정수계수 다항식을 구할 수 있다. 예를 들면,

```
<< NumberTheory`Recognize`
NSolve[3 x^3 - 2 x + 5 == 0]
        {{x → -1.37174}, {x → 0.68587 - 0.862893 i},
         {x → 0.68587 + 0.862893 i}}
sol = First[x /. %]
        -1.37174
Recognize[sol, 3, x]
        5 - 2 x + 3 x^3
```

위에서 sol = -1.37174...가 그 자체로 보았을 때 대수적인 수인지는 분명하지 않지만 (실제로는 $3x^3$-$2t$+5의 근 이었으므로 대수적인 수), Recognize 명령어는 그 수를 근사로 갖는 가장 작은 차수의 정수계수다항식을 찾아준다. 실제로 sol로부터 -5+2x-3x^3을 찾아주었는데, 이 다항식은 처음에 NSolve명령어로 근사 근을 구했던 다항식과 부호만이 다를 뿐이다. 이 Recognize 명령어는 많은 수학문제에서 어떤 유의미한 결과를 얻기 위한 도구로서 사용이 가능하다. 예를 들면 어떤 근사로 나타내어진 수가 대수적인 수임을 알고 있고 그 수를 정확히 표현할 필요가 있을 때, 이 명령어를 통하여 그 대수적인 수를 근으로 갖는 가장 작은 차수의 정수계수 다항식을 구하여 그 수가 이 다항식의 근임을 보여주면 된다.

먼저

$$s(3\pi/10,\ \pi)=t(3\pi/10,\ \pi)=1$$

이므로, 우리가 구해야 할 k는 $k\leq -1$을 만족한다. 이제,

$$F_1(a,b)\cos a+\cos b+\cos(a-b),$$
$$F_2(a,b)\cos 3a+\cos 3b+\cos 3(a-b),$$
$$K_1(a,b)=F_1(a,b)-k,\ K_2(a,b)=F_2(a,b)-k$$

라 놓고 인 경우에 각각 Mathematica를 이용하여 곡선 을 한 평면위에 동시에 그려보면 다음과 같다.

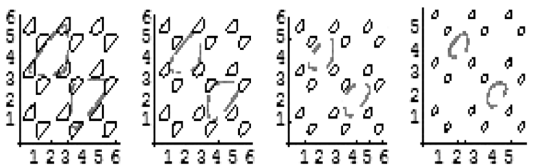

그림 1 $K_1(a,b)=0$와 $K_2(a,b)=0$
Fig. 1. $K_1(a,b)=0$ and $K_2(a,b)=0$

그림 1의 Mathematica code는 길어서 부록에서 소개한다(위 왼쪽: k = -1.1, 위 오른쪽: k = -1.2, 아래 왼쪽: k = -1.3, 아래 오른쪽: k = -1.4). 그림 1에서 가로의 축은 a축, 세로의 축은 b축이다. 그림 1의 곡선들은 $b = a$와 $b = -a+2\pi$에 대칭인 삼각형과 비슷한 모양을 한 닫힌 볼록 곡선(convex closed curve)들로 보이며, 첫 번째부터 세 번째까지의 그림을 비교해 보아, $-1.3 < k < -1.1$인 어떤 에 대하여 두곡선

$$K_1(a,b) = 0,\ K_2(a,b) = 0$$

이 접한다는 것(tangential intersection)을 짐작할 수 있고, 그러한 k가 $f(3)$를 결정짓게 된다는 것을 알 수 있다. 한편 k가 -1에서 -3으로 감소할 때 그림 1의 볼록 곡선들은 줄어들다(shrink)가 마침내는 사라져버린다는 것을 추측할 수 있다. k=-1.2인 경우에 $b = -a+2\pi$, $2.5<a<3$에서 두 곡선

$$F_1(a,b) = k,\ F_2(a,b) = k$$

이 접하는 것처럼 보인다. 실제로 이러한 추측은 우리가 여기서 다루는 함수들이 삼각함수들의 합이기 때문에 개연성이 있다. 실제로 Mathematica를 이용하여 계산하여보면,

```
trigpoly[n_, a_, b_] = Cos[n a] + Cos[n b] + Cos[n (a - b)];
cv = FindRoot[trigpoly[1, a, -a + 2 π] ==
     trigpoly[3, a, -a + 2 π], {a, 2.5, 3},
   WorkingPrecision → 50];
pac = a /. cv
        2.656530942331075843905293852589412841761761\
         5020719
```

이므로, $2.5 < a < 3$이며

$$F_1(a,-a+2\pi) = F_2(a,-a+2\pi)$$

을 만족하는 근사 값 a는 2.6565…임을 알 수 있다. 이 a=2.6565…를 $F_1(a,-a+2\pi)$에 대입한 값이 우리가 구해야할 k인 것처럼 보인다. 그런데 a=2.6565…을 수학적으로 어떻게 표현할 수 있을까? 이를 위한 방법이 위에서 소개하였던 Recognoze명령어를 사용하여 2.6565…를 대수적인 수로 인식하고, 이 수를 근으로 갖는 가장 작은 차수의 정수계수 다항식을 찾아보는 것이다. 실제로,

```
trigpoly[1, a, -a + 2 π] /. a → pac
Recognize[%, 15, x] // Factor
        -1.204094636854992018802881451088002676572\
          0690691393

        (-1 + x) (3 + 2 x + 2 x^2 + 2 x^3)
```

위의 계산으로부터 a=2.6565…일 때, $F_1(a,-a+2\pi)$은 방정식 $3+2x+2x^2+2x^3=0$의 실근임을 추측할 수 있다. 이 방정식은 비교적 간단한 3차 방정식이지만, 아래의 가설 3과 모습을 맞추기 위해 cos$a$를 근으로 갖는 가장 작은 차수의 정수계수 다항식을 Recognize명령어를 통해 다시 찾으면 다음과 같다.

```
Recognize[Cos[pac], 15, x] // Factor
        1 - 2 x + 4 x^3
```

따라서 위의 모든 추측들을 종합하면 아래의 가설을 만들 수 있다.

가설 2 cos a_0가 $4x^3-2x+1=0$을 만족하는 a_0=2.6565…에 대하여,

$$K_1(a_0,-a_0+2\pi) = K_2(a_0,-a_0+2\pi) = 0$$
$$K_1(a_0,\ 2a_0) = K_2(a_0,\ 2a_0) = 0$$

이다. 여기서, k = -1.2040…는 $3+2x+2x^2+2x^3=0$의 실근이다.

위의 가설 2를 CAS의 도움 없이 만들기는 매우 힘들다. 그러나 좋은 CAS가 보급된 현재에 적절한 CAS의 이용은 위의 예에서 보듯이 문제해결을 더욱 쉽게 해준다. 한편 위의 가설2는 [3]에서 resultant를 이용하여 수학적으로 엄밀하게 증명되었다.

또한, 그림 1에서 k = -1.2인 경우에, 곡선들의 접점들 중의 다른 두개는 각각, 어떤 상수 τ, σ에 대하여 $b = \tau$, $b = a + \sigma$위에 놓여져 있는 것처럼 보인다. 이러한 추측을 바탕으로 가설 2를 만들 때와 같은 아이디어를 사용하여 아래의 가설 3를 만들어 보겠다.

```
cw =
  FindRoot[
   (trigpoly[1, a, a + σ] == trigpoly[3, a, a + σ]) /.
    a → pac, {σ, 0.5, 1.5}, WorkingPrecision → 40];
cx = FindRoot[(trigpoly[1, a, τ] == trigpoly[3, a, τ]) /.
    a → pac, {τ, 5.3, 5.7}, WorkingPrecision → 40];
qac = σ /. cw
qbc = τ /. cx
Recognize[Cos[qac], 15, x] // Factor
Recognize[Cos[qbc], 15, x] // Factor
        0.970123422517434789114699061380180084870\
        8

        5.313061884662151687810587705178825683524

        1 - 2 x^2 - 2 x^3

        1 - 2 x^2 - 2 x^3
```

참고로 위의 계산에서 σ(0.970123…)를 표현할 때, σ의 다항식표현은 아래의

$$-12-49x+112x^2-\cdots+73x^{14}+109x^{15}$$

과 같이 부정확하다. 그러나 cosσ인 경우에는 3차인 다항식

$$1-2x^2-2x^3$$

으로 간단히 표현되므로 σ대신에 cosσ를 사용한다. 인 경우도 마찬가지다.

```
Recognize[qac, 15, x] // Factor
```

$$-12-49x+112x^2-25x^3-55x^4+19x^5+146x^6+100x^7-27x^8+16x^9-21x^{10}-184x^{11}-168x^{12}-55x^{13}+73x^{14}+109x^{15}$$

또한 곡선 $K_1(a,b)$, $K_2(a,b)$의 내부(interior)는 $K_1(a,b)<0$, $K_2(a,b)<0$을 만족하는 점 (a,b)들의 집합임을 확인할 수 있다. 따라서,

가설 3 cosσ와 cosτ가 $2x^3+2x^2-1=0$을 만족하는 $\sigma=0.9701\cdots$, $\tau=5.3130\cdots$에 대하여, 직선 $b=\tau$, $b=a+\sigma$ 위에 있는 모든 점 (a,b)에서는

$$K_1(a,b)\geq 0,\ K_2(a,b)\geq 0$$

이다. 여기서, $k=-1.2040\cdots$는 $3+2x+2x^2+2x^3=0$의 실근이다.

가설 3도 참으로서 [3]에서 중명되어졌다. 위의 두 (증명 되어진) 가설을 이용하여 아래의 정리 4를 증명할 수 있다[3]. 따라서

$$f(3)\sqrt{2k+3}=0.769292\cdots$$

을 얻는다.

정리 4 만일 $k=-1.204094\cdots$는 방정식

$$3+2x+2x^2+2x^3=0$$

의 실근이라면, 두 집합

$$\{(a,b)\in[0,2\pi]^2:\ \cos a+\cos b+\cos(a-b)\leq k\}$$
$$\{(a,b)\in[0,2\pi]^2:\ \cos 3a+\cos 3b+\cos 3(a-b)\leq k\}$$

은 법 2π에 대하여 기껏해야 유한개의 점들, 집합에서 등호가 성립하는 점들, (a,b)에서 만난다.

3. 라그랑지 승수법을 이용한 방법

이 절에서는 두 곡선이 접하는 점을 알기 위해 라그랑지 승수법(Lagrage multiplier method)을 이용하기로 한다. 우리가 원하는 두 곡선이 접할 때 구해지기 때문에 라그랑지 관계

$$\nabla F_1(a,b)=\lambda\nabla F_2(a,b)$$

를 고려하며, 아래의 Mathematica code

```
Needs["Graphics`"]
grad[expr_, a_, b_] := {∂a expr, ∂b expr}
trigsubs = {Cos[a] → ca, Sin[a] → sa, Cos[b] → cb,
  Sin[b] → sb};
{e1, e2} = Map[TrigExpand[trigpoly[#, a, b]] &, {1, 3}];
polys =
 Flatten[{e1 - k, e2 - k, grad[e1, a, b] - λ grad[e2, a, b],
    Cos[a]^2 + Sin[a]^2 - 1, Cos[b]^2 + Sin[b]^2 - 1}] /.
  trigsubs
```

$$\{-k+c_a+c_b+c_a c_b+s_a s_b,\ -k+c_a^3+c_b^3+c_a^3 c_b^3-3c_a s_a^2-3c_a c_b^3 s_a^2+9c_a^2 c_b^2 s_a s_b-3c_b^2 s_a^3 s_b-3c_b s_a^2 s_b^2-3c_a^3 c_b s_b^2+9c_a c_b s_a^2 s_b^2-3c_a^2 s_a s_b^3+s_a^3 s_b^3,\ -s_a-c_b s_a+c_a s_b-\lambda(-9c_a^2 s_a-9c_a^2 c_b^3 s_a+3s_a^3+3c_b^3 s_a^3+9c_a^3 c_b^2 s_b-27c_a c_b^2 s_a^2 s_b+27c_a^2 c_b s_a s_b^2-9c_b s_a^3 s_b^2-3c_a^3 s_b^3+9c_a s_a^2 s_b^3),\ c_b s_a-s_b-c_a s_b-\lambda(9c_a^2 c_b^3 s_a-3c_b^3 s_a^3-9c_b^2 s_b-9c_a^3 c_b^2 s_b+27c_a c_b^2 s_a^2 s_b-27c_a^2 c_b s_a s_b^2+9c_b s_a^3 s_b^2+3s_b^3+3c_a^3 s_b^3-9c_a s_a^2 s_b^3),\ -1+c_a^2+s_a^2,\ -1+c_b^2+s_b^2\}$$

에서처럼, 삼각함수를 간단한 모양(예를 들어 cos$a$를 c_a로)으로 바꿔 쓰고, $c_a^2+s_a^2=1$등과 같은 잘 알려진 대수적인 관계를 보강한다. 그 다음에, k만을 제외한 모든 변수들을 소거하기 위해 Gröbner 기저를 이용한다. 아래 Mathematica code의 이론적 배경은[1] 또는 [2]에서 찾아질 수 있다.

```
kpoly =
 First[GroebnerBasis[polys, x, {λ, ca, cb, sa, sb},
    MonomialOrder → EliminationOrder]] // Factor
```

$$(-3+k)(-1+k)(1+k)(3+2k+2k^2+2k^3)$$

값 -1, 1, 3은 문제에서의 극값을 갖는 k가 아님을 쉽게 확인할 수 있다. 따라서 방정식

$$3+2k+2k^2+2k^3=0$$

의 실근 $k=-1.204094\cdots$가 $f(3)$를 결정짓는 상수가 된다. 따라서 $f(3)=\sqrt{2k+3}=0.769292\cdots$ 이다.

4. 결 론

2절과 3절에서 다루어진 문제는 평면위의 곡선

$$\cos a+\cos b+\cos(a-b)=\underline{k}$$
$$\cos 3a+\cos 3b+\cos 3(a-b)=\underline{k}$$

들이 만나는 점 (a,b,k)를 찾는 것이었다. 이를 위하

여 2절에서는 CAS인 Mathematica를 이용하여 결과를 위한 가설을 세웠고, 3절에서는 라그랑지 승수법과 Gröbner 기저를 이용한 풀이를 제공하였다. 한편 2절에서 사용한 방법은 다른 문제인 단위원위의 세 벡터들의 거리들의 곱에 대한 최대최소문제

$$g(3):\max_{a_i \text{real}} min\{|(e^{ia}-e^{ib})(e^{ib}-e^{ic})(e^{ic}-e^{ia})|, \\ |(e^{i3a}-e^{i3b})(e^{i3b}-e^{i3c})(e^{i3c}-e^{i3a})|\}$$

의 풀이에 그대로 적용 된다 (과정 생략). 실제로 이 문제는 서론에서 $f(3)$를 구하기 위해 행했던 과정을 따르면, 평면위의 곡선들

$$-\sin a+\sin b+\sin(a-b)=\underline{k}$$
$$-\sin 3a+\sin 3b+\sin 3(a-b)=\underline{k}$$

이 만나는 점 (a, b, k)를 찾는 것으로 귀결됨을 확인할 수 있다. 이 문제는 2절에서 소개된 방법을 순서대로 이용하여 해결할 수 있으나 3절에서 소개된 방법의 적용은 매우 어렵다. 여기에 2절에서 소개하였던 방법의 중요성이 있다. 계산능력이 강한 개인용 컴퓨터와 강한 기호계산(symbolic computation)이 가능한 CAS가 보급된 오늘날 이들의 수학문제 풀이에 있어 적절한 활용은 문제해결을 한결 쉽게 해 준다. 실제로 2절에서 만들어 내었던 가설들은 CAS의 도움 없이 만들어지기 힘들다. 이 논문에서는 극값의 문제들과 관련된 $f(3)$, $g(3)$와 같은 종류에 문제의 풀이에 CAS를 이용하는 방법을 구체적으로 제시하였다. 이 방법은 이들과 유사한 문제에 적용할 수 있다는 것에 그 중요성이 있다.

참고문헌

[1] D. Cox, J. Little, and D. O'Shea, "Ideals, Varieties, and Algorithms", 2nd edition, Springer-Verlag, New York, 1996

[2] D. Cox, J. Little, and D. O'Shea, "Using Algrbraic Geometry", Springer-Verlag, New York, 1998

[3] S. H. Kim, "Planar packings and mappings related to certain minmax problems", Math. Inequal. Appl., Vol. 6, no 2, pp.351-374, 2003.

부록

다음은 <그림 1> 의 Mathematica code이다.

```
pl1 = ImplicitPlot[{trigpoly[1, a, b] == -1.1,
    trigpoly[3, a, b] == -1.1}, {a, 0, 2 π}, {b, 0, 2 π},
   PlotPoints → 200, AspectRatio → 1,
   PlotStyle →
    {{Thickness[0.012], GrayLevel[0.5],
      Dashing[{0.25, 0.1}]}, {Automatic}},
   DisplayFunction → Identity];
pl2 = ImplicitPlot[{trigpoly[1, a, b] == -1.2,
    trigpoly[3, a, b] == -1.2}, {a, 0, 2 π}, {b, 0, 2 π},
   PlotPoints → 200, AspectRatio → 1,
   PlotStyle →
    {{Thickness[0.012], GrayLevel[0.5],
      Dashing[{0.25, 0.1}]}, {Automatic}},
   DisplayFunction → Identity];
pl3 = ImplicitPlot[{trigpoly[1, a, b] == -1.3,
    trigpoly[3, a, b] == -1.3}, {a, 0, 2 π}, {b, 0, 2 π},
   PlotPoints → 200, AspectRatio → 1,
   PlotStyle →
    {{Thickness[0.012], GrayLevel[0.5],
      Dashing[{0.25, 0.1}]}, {Automatic}},
   DisplayFunction → Identity];
pl4 = ImplicitPlot[{trigpoly[1, a, b] == -1.4,
    trigpoly[3, a, b] == -1.4}, {a, 0, 2 π}, {b, 0, 2 π},
   PlotPoints → 200, AspectRatio → 1,
   PlotStyle →
    {{Thickness[0.012], GrayLevel[0.5],
      Dashing[{0.25, 0.1}]}, {Automatic}},
   DisplayFunction → Identity];
Show[GraphicsArray[{pl1, pl2, pl3, pl4}],
  DisplayFunction → $DisplayFunction];
```