

數學教育의 電算化에 對한 理論과 實際*

한 재 영**

1. 서론

이 논문의 목적은 컴퓨터 언어로 표현된 대수적인 수식으로 그래픽을 다단계 연출하는 Animation 기법들을 연구하여 수학교육 현장에서 직접 활용할 수 있는 여러 가지 모델을 제시하는데 있다. Mathematica Version 3.0의 기능 중에 애니메이션의 원리를 수학교과과정의 전산화 프로그램에 원용하여 수학교육 방법을 개선하고자 하였다. 원주는 회전각을 매개변수로 하는 중심에서 일정한 거리를 갖는 점의 자취이다. 이 원의 중심이 회전각을 매개변수로 하는 곡선이나 직선을 따라 움직이면 원주상의 점은 매우 복잡한 경로를 따라 다단계 그래픽을 출력하게 되며 이런 점들이 만드는 자취를 이용하여 수식에 의한 특수한 그래프를 출력하고자 한다. 수학적 애니메이션은 한 개의 그래프를 그리는데 영화의 화면과 같이 단계별로 여러 조각의 그림을 출력하여 그를 합성하는 그래픽 기법으로 수학적 결론에 도달하는 그래프를 여러 단계로 나누어 출력함으로써 움직이는 화면을 학습자에게 보여 줄 수 있다. 또한 이 논문을 통해서 수학은 완성된 것이 아니라 그 과정을 학습하는 움직이는 학문이라는 관념을 심어주어 창의적이고 생동감 있는 학습을 진행하는데 기여하고

자 한다.

수학교과과정 그 자체를 전산 프로그램화 하여 PC만으로 학습자 스스로 수학을 보편적 학문으로 받아들이게 하는데 이 논문의 주된 목표이다. 이런 유형의 연구는 수학교육의 새로운 형태를 창출하는 계기를 마련하게 될 것으로 기대되며 여기에 언급된 것들은 수식에 의한 애니메이션의 기본 기법에 대한 입출력으로 이런 입력기법들은 학습단원의 성격이나 학습 목표의 설정에 따라 수학교과내용을 전산화하는데 긴요한 입출력 기법으로 사용될 수 있다. 수학적 애니메이션은 컴퓨터의 용량이 허용되는 한에서 다단계출력이 효과적이거나 여기서는 한글 편집을 쉽게 하기 위하여 몇 단계의 구분으로 입력하여 한글로 옮겼다. 이런 기법들이 구체적으로 학습현장에서 어떻게 활용되는가 하는 것은 앞으로 여러 학술지에 발표될 본인의 논문을 참고하기 바란다. 새로운 소프트웨어를 이용하여 수학교과 자체를 전산화하고 전산프로그램에 의하여 수업을 진행하여 학습자의 흥미를 유발하고 살아있는 수학교육을 실현하려는 시도가 구미각국에서는 이미 완성단계에 있다. 이 논문에서는 다단계, 단계별, 다층별 그래픽 연출기법을 개발하여 학습현장에 이용하는데 유용한 입출력의 프로그램을 다루려 하며 입력에 사용된 그래픽 언어를 읽어보면 출력하려는 그래픽의 목표를 쉽게 파악할 수

* 이 논문은 '98년도 한국대학교육협의회 대학 교수 국내교류 연구비 지원에 의한 것임'

** 충북대학교

있다. 또한 입력된 프로그램의 주된 목적도 간략히 기술하였으며 이 논문에서 사용된 명령어와 수치의 변화를 관찰하여 유사한 수학문제의 해결에 직접 적용할 수 있기를 기대한다. 이 원고는 Mathematica 3.0의 입출력을 수식은 Text, 그래픽은 Bitmap으로 Copy하여 한글로 옮긴 것이다. 명령어들의 적절한 결합으로 보다 효율적인 바탕그림을 만드는데 많은 노력이 요구되며 이런 기법들을 지면의 한정 때문에 작은 글씨로 제시했다. C-언어로 표현된 프로그램을 재편집하는 과정에서 명령어가 분리되는 일이 없도록 유의해야 했다. 컴퓨터 화면에는 역동하는 채색된 그림이 다단계 다층 다면으로 연출되지만 한글로 편집된 이 논문에서는 그런 탁월한 동력화된 전산기능을 연출할 수 없었다. 본지의 편집실이나 투고자 본인이 소지하고 있는 Mathematica 디스켓 원본을 독자 스스로 입출력하여 보면 이 논문에서 제시하고 있는 애니메이션 기법을 익히고 각자가 필요한 학습프로그램을 작성하는데 유용하리라 본다.

II. 다단계 다중화면의 동시 연출 기법

반경 r, 중심이 (x, y)인 원주상의 동점 (x + r sin t, y + r cos t)가 이루는 자취를 편의상 Han으로 정의하여 수학교육에 긴요한 여러 가지 그래프를 출력한다. 변수 x, y, r, t를 얻고자하는 그림의 상황에 따라 적절히 조절함으로써 학습의 목표를 선명하게 제시하고 그 결과를 극명하게 얻을 수 있는 다양한 그래픽의 연출이 가능하다.

정의 1-1 Han[x_, y_, r_, t_] := {Circle[{x, y}, r], Line[{x, y}, {x, y}+r{Sin[t], Cos[t]}], PointSize[0.05], Point[{x+r Sin[t], y+r Cos[t]}]}

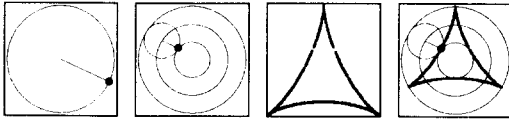
위 정의에서 출력되는 곡선의 굵기는 점의 크기 r로 조정하고, 회전각 t로 회전의 정도를 결정한다. 구체적인 그림을 출력하려면 그림의 스타일을 지칭하는 컴퓨터언어와 실행프로그램의 설정이 뒤따른다.

정의 1-2 CLP = {Circle[{x, y}, r], Line[{x, y}, {x, y}+r{Sin[t], Cos[t]}], PointSize [0.07], Point[{x + Sin[t], y + r Cos[t]}], PlotStyle->{{Dashing[{0.002}], Thickness[{0.5}], RGBColor[1,0,0]}, {Dashing[{0.001}, 0.002]}, Thickness[{0.1}], RGBColor[0, 1, 0]}, {Dashing[{0.003}], Thickness[{0.1}], RGBColor[0,0,1]}}

바탕화면의 연출기법 1-1 mc2=CLP/.{x->2 Sin[-t/2],y->2 Cos[-t/2],r->1}; mc2/t->0; Show[Graphics[%], AspectRatio -> Automatic]; Show[Graphics[{Hue[0.5], Rectangle[{-1,1},{1,3}]}], AspectRatio->Automatic]; Show[%,%,AspectRatio-> Automatic]

출력된 그림의 기하학적 의미를 파악하기 위해서 바탕화면의 구성이 중요한 요소이다. 애니메이션에 의한 그림의 다단계출력을 돕는 밑그림의 프로그래밍의 구체적인 예는 다음과 같은 것이 유용하게 쓰인다. 정지된 화면인 밑그림을 그리는데 그래픽에 관한 여러 가지 기교를 동원하는 작업이 필요하다.

바탕화면의 연출기법 1-2 mc2=CLP/.{x->2 Sin[-t/2], y->2 Cos[-t/2],r->1}; mc2/t->2; FullForm[%]; Show[Graphics[%], AspectRatio->Automatic]; Show[Graphics[{{Line[{-2 Sin[1], 2 Cos[1]}, {-2 Sin[1]+Sin[2], 2 Cos[1]+Cos[2]}]}, {Dashing[{0.05}], RGBColor[0,1,1]}, {PointSize[0.05*], Point[{-2 Sin[1]+Sin[2], 2 Cos[1]+Cos[2]}]}], AspectRatio->Automatic, Background->Graylevel[0]}; Show[%,%}]

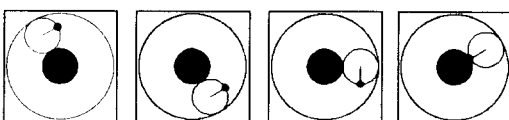


명령어 **Animate**는 그래픽의 다단계연속출력을 지시하는 컴퓨터언어로 완성된 그림을 그리는데 여러 단계를 극명하게 보여주는 이점이 있다. 매개변수 t 의 구간을 몇 등분하느냐에 따라 연계된 그림이 화면으로 다단계로 연결되어 출력된다.

절리 1-1 `Animate[Show[Graphics[]], {t, 0, a, a/n}]`에서 n 은 다단계연출의 화면 개수를 지정한다. n 이 클수록 화면의 완성에 이르는 과정이 그래픽으로 섬세하게 나타난다. 컴퓨터의 용량이나 출력성능의 한계로 개인용 컴퓨터에서는 출력이 제한된다. 또 이들을 인쇄하려면 너무 많은 시간과 종이가 소비된다. 성능이 배가된 PC나 멀티미디어 프로젝트에서는 백여 단계의 출력이 가능함으로 학습시간에는 상당수의 다단계 연속 화상도 출력할 수 있다. 다만 출력된 결과 모두를 저장하는 것은 컴퓨터 용량을 너무 많이 잠식함으로 Kernel의 Delete All Output을 시행한 다음 저장한다.

다단계출력의 기본 1-3

```
Animate[Show[Graphics[{{mc2, RGBColor[0,0,1]},
Disk[{0,0},1], Hue[{0.5}], PlotStyle-> {Dashing[{0.001}],
RGBColor[1,0,0], Circle[{0,0},2]}, {Dashing[{0.001,0.002}],
RGBColor[0,1,0], Circle[{0,0},3]}, {PointSize[0.05],
newpts2}], AspectRatio->Automatic], {t,0,4Pi,4Pi/n}]
```

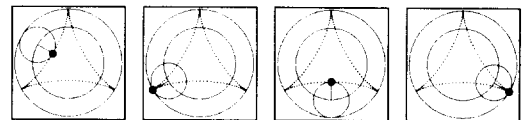


여기서 $n=4$ 로 두면 검은 점 위를 미끄러지면서 밖의 원을 따라서 회전하는 작은 원의 움직임이 울동적으로 출력된다. 그림을 아름답고 선명하게 보기 위해서 색조, 점의 크기, 점의 개수, 점선의 간격 등을 적당하게 조정한다. 아래 그림은 Mathematica에서 비트맵으로 복사하여 한글로 옮긴 것이다. 그림의 크기는 Mathematica에서 조정하여 한글로 옮기거나 그와 반대로 작업한 것이다. 하나의 화면에 두개 이상의 그래픽이 동시 출력되는 화면의 구성은 여러 가지의 자료가 그리는 그래픽을 다 단계동시 연출한다.

절리 1-2 `Animate[{{Show[Graphics[]], ... , {Show[Graphics[]], {etc.}}, {t, 0, x, x/n}]`은 동일화면에 다중동작을 연출하는 프로그램이다.

다단계 다중화면 연출 기법 1-4

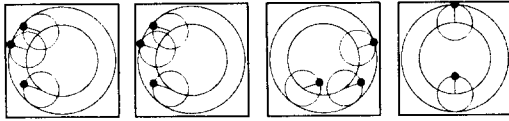
```
Animate[Show[Graphics[{{mc2,Circle[{0,0},2],Circle[{0,0},3],PointSize[0.015],newpts2,PlotStyle->{{Dashing[{0.002}],Thickness[{0.1}],RGBColor[1,0,0]}, {Dashing[{0.001,0.002}],Thickness[{0.1}],RGBColor[0,1,0]}, {Dashing[{0.003}],Thickness[{0.1}],RGBColor[0,0,1]}}], AspectRatio->Automatic], {t,0,4Pi,4 Pi/6}]
```



다단계 동시출력 연출 기본 1-5

```
mc4=CLP/.{x->2 Sin[-t/4],y->2 Cos[-t/4],r->1};
mc8=CLP/.{x->2 Sin[-t/8], y->2 Cos[-t/8],r->1};
Animate[{{Show[Graphics[{{mc2, mc4, mc8,
RGBColor[0,0,1], {Dashing[{0.001}], RGBColor[1,0,0],
Circle[{0,0},2]}, {Dashing[{0.001,0.002}], RGBColor[0,1,0],
Circle[{0,0},3]}, {PointSize[0.007], newpts2}}],
AspectRatio->Automatic}], {Show[Graphics[{{mc2, mc4,
```

```
mc8, RGBColor[0,0,1]], {Dashing[{0.001}],
RGBColor[1,0,0], Circle[{0,0},2]], {Dashing[{0.001,0.002}],
RGBColor[0,1,0], Circle[{0,0},3]], {PointSize[0.007],
newpts4}}], AspectRatio->Automatic]], {t,0,8Pi,8Pi/4}]
```

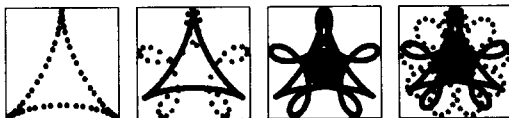


여러 개의 그래프를 한 화면에서 연계하여 다단계로 연출하는 기법은 다양한 그림을 한붓그리는 작업으로 그래픽의 중요한 기능이다.

절리 1-3 연계할 그래프를 구성하는 Table을 출력하고 이들을 순서대로 프로그램 하면 계산의 시차가 그래픽의 순서를 결정한다. 이의 프로그램의 기본형은 `Animate[Show[Graphics[Table[{hpts1, hpts2, etc.}], {n, 0, a, a/n}]]`이다.

다단계 동시출력 기본 1-6

```
hpts2=Table[{PointSize[0.05],Point[{-2 Sin[t/2]+Sin[t],2
Cos[t/2]+Cos[t]}], {t,0,20Pi,0.2}];
hpts4=Table[{PointSize[0.05],Point[{-2 Sin[t/4]+Sin[t],2
Cos[t/4]+Cos[t]}], {t,0,40Pi,0.4}]; Animate[Show
[Graphics[Table[{PointSize[0.05], hpts2, hpts4, Hue[{0.1}],
Disk[{0,0},1], Point[{-2 Sin[t/8]+Sin[t], 2 Cos[t/8]+Cos[t]}],
{t,0, 0.5 n,0.5}], AspectRatio->Automatic],
{n,0,32Pi,32Pi/1}]
```



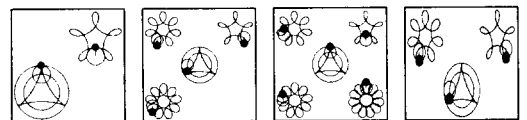
절리 1-4 원점과 연계된 여러 장소에 다양한 그림을 동시화면으로 다단계 연출하려면 하나의 고정 점에서 일정한 규칙으로 배열된 점을 만들어 애니메이션 한다. 이런 수식의 프로

그램은 `mc[n_, t_, a_, b_] := CLP[x, y, t, r]/.{x->a+2Sin[-t/n], y->b+2Cos[-t/n], r->1}`의 형태이다.

다음은 원점을 중심으로 대칭적인 4개의 그림을 순차적으로 출력한 것을 한 가지 형태로 보여준 것이다. 이런 방법은 수식에 의한 그래프의 출력에 긴요한 다중화면 연계화상 연출의 기본으로 한 고정 점에 Pivot하는 기법이다.

다단계 동시출력의 기법 1-7

```
jtable2=jtable[[1]]/.{a->0, b->0};jtable4=jtable[[2]]/.{a->6,
b->6}; jtable6=jtable[[3]]/.{a->6,
b->6};jtable8=jtable[[4]]/.{a->-6, b->-6}; jpts2=
Table[jtable[[1,4]], {t,0,4Pi,0.01}]/.{a->0,b->0};
jpts4=Table[jtable[[2,4]], {t,0,8Pi,0.02}]/.{a->6,b->6};
jpts6=Table[jtable[[3,4]], {t,0,12Pi,0.03}]/.{a->-6, b->-6};
jpts8=Table[jtable[[4,4]], {t,0,16Pi,0.04}]/.{a->-6, b->-6};
jpts10=Table[jtable[[5,4]], {t,0,20Pi,0.05}]/.{a->6, b->-6};
Animate[Show[Graphics[{jtable2, jtable4, jtable6, jtable8,
jtable10, Circle[{0,0},2], Circle[{0,0},3], Dashing[{0.002}],
PointSize[0.005], jpts10, jpts8, jpts6,jpts4, jpts2}],
AspectRatio->Automatic], {t,0,4Pi,4 Pi/4}]
```

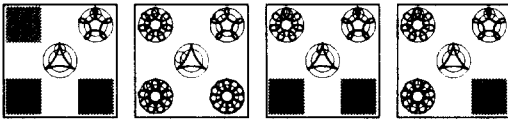


절리 1-5 고정 점에 연계되지 않는 여러 장소에 다양한 그림을 동시화면으로 다단계 연출하는 기본형은 사각형배열

```
Animate[Show[Graphics[{{Rectangle[{a, b}, {c, d}],
{Rectangle[{a', b'}, {c', d'}], {Rectangle[{a'', b''}, {c'',
d''}], {Rectangle[{a''', b'''}, {c''', d'''}],
Rectangle[{a''', b''''}, {c''', d''''}]}],
AspectRatio->Automatic], {t, 0, e, e/n}]
```

다단계 동시출력 기법 1-8

```
sunen=Show[Graphics[{jtablen,Circle[{0,0},2],Circle[{0,0},3],Dashing[{0.001}],PointSize[0.05],jpts2}],AspectRatio->Automatic];
Animate[Show[Graphics[{Hue[0.9],Rectangle[{-3,-3},{3,3},sune2]},(Hue[0.7],Rectangle[{3,3},{9,9},sune4]),(Hue[0.5],Rectangle[{-3,3},{-9,9},sune6]),(Hue[0.3],Rectangle[{-3,-3},{-9,-9},sune8]),(Hue[0.1],Rectangle[{3,-3},{9,-9},sune10])}],AspectRatio->Automatic],{t,0,8Pi,8 Pi/10}]
```

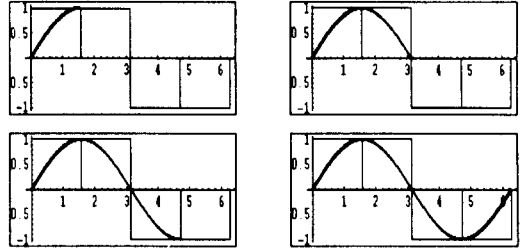


정리 1-6 주기함수의 구간을 몇 단계로 나누어 애니메이션 하면 그래픽의 성격을 관찰하기에 편리하다. 변수의 변역 $\{x, a, b, (b-a)/n\}$ 의 정수 n 은 구분동작을 나타낸다.

다음은 주기함수 $\sin x$ 를 함수의 증감을 식별하기 쉽도록 연출한 것이다. 주기와 그의 분할을 적절히 조절하면 기초함수의 영역별 변화를 극명하게 보여줄 수 있다. 이런 작업은 사인 곡선의 여러 가지 특성을 학습하는데 큰 효과를 얻을 수 있는 전산화 학습의 실체이다.

다단계 동시출력의 기본 1-9

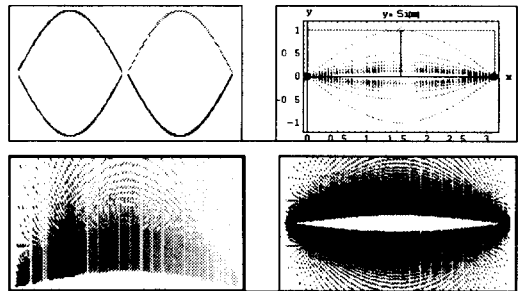
```
HanG[x_,y_,r_,t_]:=Circle[{x,y},r],Line[{x,y},{x,y}+r{Sin[t], Cos[t]}], Point[{x+r Sin[t],y+rCos[t]}];
Animate[Show[Graphics[Table[{PointSize[0.01],RGBColor[0,0,0],Line[{Pi/2,0},{Pi/2,1]}],Line[{0,0},{Pi,0},{Pi,1},{0,1}],Line[{3Pi/2,0},{3Pi/2,-1}],Line[{Pi,0},{Pi,-1},{2Pi,-1},{2Pi,0}],HanG[x,y,r,t]/{x->t/4,y->Sin[t/4],r->0.0001}],{t,0,0.1n,0.1}],Axes->True,AspectRatio->Automatic],{n,0,800Pi,800Pi/4}]
```



사인 곡선은 모든 곡선의 근원이 되는 것으로 이 곡선이 갖는 기하학적 조형미를 연출하여 학습시간에 학생들에게 보여주면 수학에 대한 새로운 흥미를 갖게 된다. $HanG[x, y, r, t] / \{x \rightarrow \pi + t/m, y \rightarrow -\sin[\pi + t/n], r \rightarrow a\}$ 에 m, n, a 의 값을 적절히 조정하면 사인곡선의 아름다운 조형미를 출력하게 된다. 다음의 출력 영상은 사인곡선의 조형미를 여러 측면에서 연출하여 학습시간에 보여줌으로써 학습자들이 수학에 많은 흥미를 유발한 대표적인 작품들이다.

사인곡선의 조형미 연출 기법 1-10

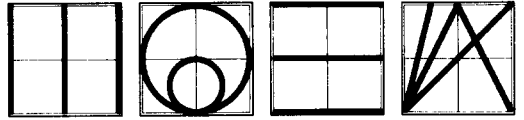
```
Animate[Show[Graphics[Table[{PointSize[0.01],Hue[2/t],HanG[x,y,r,t]/{x->t/4,y->1/mSin[t/4],r->0.001}],{m,0,10,0.1},{t,0,0.1n,0.1}],Frame->True,FrameTicks->False,AspectRatio->Automatic],{n,0,40Pi,40Pi/1}]
```



III. 학습용 그래픽 구성 기법

수학의 기본을 이루는 도형으로 원과 삼각

형, 사각형의 그래픽 연출 기법을 개발하고자 한다. 다른 도형은 이들의 적절한 조합으로 볼 수 있으므로 문제의 성격에 따라 그래픽 기법을 원용한다.



정의 2-1 그래픽의 효능을 제고하기 위하여 한글의 자모를 형성하는 기본 프로그램을 다음과 같이 형성한다.

(1) $\wedge 1 = \text{Han}[t/4, t/2, 0.001, t]$, $\wedge 11 = \text{Han}[t/4, t/4, 0.001, t]$, $\wedge 2 = \text{Han}[Pi-t/4, -(Pi-t/4)+2Pi, 0.001, t]$, $\wedge 111 = \text{Han}[Pi-t/4, -(Pi-t/4)+Pi, 0.001, t]$, $\varepsilon 1 = \text{Han}[t/2, Pi, 1/1000, t][[4]]$

(2) $\varepsilon 2 = \text{Han}[t/2, Pi/2, 1/1000, t][[4]]$, $\varepsilon 3 = \text{Han}[t/2, 0, 1/1000, t][[4]]$, $\varepsilon 11 = \text{Han}[t/4, Pi, 1/1000, t][[4]]$, $\varepsilon 21 = \text{Han}[t/4, Pi/2, 1/1000, t][[4]]$, $\varepsilon 31 = \text{Han}[t/4, 0, 1/1000, t][[4]]$, $\varepsilon 12 = \text{Han}[Pi-t/4, Pi, 1/1000, t][[4]]$, $\varepsilon 22 = \text{Han}[Pi-t/4, Pi/2, 1/1000, t][[4]]$, $\varepsilon 32 = \text{Han}[Pi-t/4, 0, 1/1000, t][[4]]$

(3) $\varepsilon 1 = \text{Han}[0, t/2, 1/1000, t][[4]]$, $\varepsilon 2 = \text{Han}[Pi/2, t/2, 1/1000, t][[4]]$, $\varepsilon 3 = \text{Han}[Pi, t/2, 1/1000, t][[4]]$, $\varepsilon 11 = \text{Han}[0, t/4, 1/1000, t][[4]]$, $\varepsilon 21 = \text{Han}[Pi/2, t/4, 1/1000, t][[4]]$, $\varepsilon 31 = \text{Han}[Pi, t/4, 1/1000, t][[4]]$, $\varepsilon 12 = \text{Han}[0, Pi-t/4, 1/1000, t][[4]]$, $\varepsilon 22 = \text{Han}[Pi/2, Pi-t/4, 1/1000, t][[4]]$, $\varepsilon 32 = \text{Han}[Pi, Pi-t/4, 1/1000, t][[4]]$

(4) $\varepsilon 1 = \text{Han}[Pi/3, t/2, 1/1000, t][[4]]$, $\varepsilon 2 = \text{Han}[2Pi/3, t/2, 1/1000, t][[4]]$

(5) $\circ = \text{Han}[(Pi/4-0.01) \text{Cos}[t/2]+Pi/2, (Pi/4-0.01) \text{Sin}[t/2]+Pi/4, 0.001, t]$

이 정의에 주어진 각각은 이 논문에서 연출하려는 그래픽의 기본형이 된다. 이들 중 가장 기본이 되는 그림은 다음과 같은 간단한 직선이나 원 등을 출력한다. 엄밀한 의미에서 수학의 대부분의 그래프는 이들의 결합으로 이루어지며, 이들의 연출 기법을 원용하면 다른 그래프도 근사적으로 그릴 수 있다.

정리 2-1 기본도형을 그리는 순서는 입력된 자료를 컴퓨터가 인식하고 출력하는 시차에 따라 그려진다. 입력 프로그램 $\varepsilon 1 = \text{Han}[0, Pi-t/2, 1/1000, t][[4]]$, $\varepsilon 1 = \text{Han}[s/2, Pi, 1/1000, s][[4]]$, $\varepsilon 3 = \text{Han}[Pi, Pi-u/2, 1/1000, u][[4]]$, $\varepsilon 3 = \text{Han}[v/2, 0, 1/1000, v][[4]]$ 는 사각형을 한글 맞춤법 순서로 그리는 직선들의 조합이다.

기본도형의 연출 기법 2-1

```
Animate[{{Show[Graphics[Table[{PointSize[0.05],
\varepsilon 1 = Han[0, Pi-t/2, 1/1000, t][[4]], \varepsilon 1 = Han[s/2,
Pi, 1/1000, s][[4]], \varepsilon 3 = Han[Pi, Pi-u/2, 1/1000, u][[4]], \varepsilon 3 = Han[v/2,
0, 1/1000, v][[4]], {t, 0.1, 0.1n, 0.1}],
AspectRatio->Automatic]}, {Show[Graphics[Table[{PointSize[0.05],
\varepsilon 1 = Han[0, Pi-t/2, 1/1000, t][[4]], \varepsilon 1 = Han[s/2, Pi, 1/1000, s][[4]],
\varepsilon 3 = Han[Pi, Pi-u/2, 1/1000, u][[4]], \varepsilon 3 = Han[v/2, 0, 1/1000, v][[4]],
{s, 0.1, 0.1n, 0.1}], AspectRatio->Automatic]}, {Show[Graphics[Table[{PointSize[0.05],
\varepsilon 1 = Han[0, Pi-t/2, 1/1000, t][[4]], \varepsilon 1 = Han[s/2, Pi, 1/1000, s][[4]],
\varepsilon 3 = Han[Pi, Pi-u/2, 1/1000, u][[4]], \varepsilon 3 = Han[v/2, 0, 1/1000, v][[4]],
{u, 0.1, 0.1n, 0.1}], AspectRatio->Automatic]}, {Show[Graphics[Table[{PointSize[0.05],
\varepsilon 1 = Han[0, Pi-t/2, 1/1000, t][[4]], \varepsilon 1 = Han[s/2, Pi, 1/1000, s][[4]],
\varepsilon 3 = Han[Pi, Pi-u/2, 1/1000, u][[4]], \varepsilon 3 = Han[v/2, 0, 1/1000, v][[4]],
{v, 0.1, 0.1n, 0.1}], AspectRatio->Automatic]}], {n, 0, 20, 20, 20, 20, 4}]}
```

정리 2-2 원의 출력 프로그램, $\text{Han}[(Pi/2-0.01) \text{Cos}[Pi/2+t/2]+Pi/2, (Pi/2-0.01) \text{Sin}[Pi/2+t/2]+Pi/2, 0.001, t]$, $\text{Han}[(Pi/2-0.01) \text{Cos}[3Pi/2+s/2]+Pi/2, (Pi/2-0.01) \text{Sin}[3Pi/2+s/2]+Pi/2, 0.001, s]$ 과 “s”의 입력 $\wedge 2 = \text{Han}[Pi/2-t/4, Pi-t/2, 0.001, t]$, $\wedge 1 = \text{Han}[Pi/2+s/4, Pi-s/2, 0.001, s]$ 를 조합하면 네모, 원, 삼각형이 한 화면에 순서대로 정교하게 출력된다.

기본 도형의 연출 기법 2-2

```

Animate[{{Show[Graphics[Table[{PointSize[0.05],RGBColor[0.500008, 0, 0.062501], t1=Han[0,Pi-t/2,1/1000,t][[4]],
t2=Han[s/2,Pi,1/1000,s][[4]], t3=Han[Pi,Pi-u/2,1/1000,u][[4]],
t4=Han[v/2,0,1/1000,v][[4]],{t,0.1,0.1n,0.1}],AspectRatio->Automatic]},
{Show[Graphics[Table[{PointSize[0.05],RGBColor[0.488289, 0.00781262,
0.17969],Line[{{0,0},{0,Pi},{Pi,Pi},{Pi,0},{0,0},{Pi/2,0},{Pi/2,
Pi},{0,Pi},{0,Pi/2},{Pi,Pi/2}],Han[(Pi/2-0.01)
Cos[Pi/2+t/2]+Pi/2,(Pi/2-0.01)
Sin[Pi/2+t/2]+Pi/2,0.001,t],Han[(Pi/2-0.01)
Cos[3Pi/2+s/2]+Pi/2,(Pi/2-0.01) Sin[3Pi/2+s/2]+Pi/2,0.001,s]],
{t,0.1,0.1n,0.1}], AspectRatio->Automatic]},
{Show[Graphics[Table[{PointSize[0.05],RGBColor[0.996109,
0, 0.500008], Line[{{0,0},{0,Pi},{Pi,Pi},
{Pi,0},{0,0},{Pi/2,0},{Pi/2,Pi},{0,Pi},{0,Pi/2},{Pi,Pi/2}],^2=
Han[Pi/2-t/4,Pi-t/2,0.001,t],^1= Han[Pi/2+s/4,Pi-s/2,0.001,s]],
{t,0.1,0.1n,0.1}],AspectRatio->Automatic}}],AspectRatio
->Automatic}]],{n,0,20Pi,20Pi/1}]

```



정리 2-3 도형의 회전은 아름다운 그림을 연출하는데 중요한 역할을 한다. 모음 “□”과 자음 “ㄴ”을 이루는 프로그램

```

HanR[Cos[[Theta]], Sin[[Theta]], x, y, r, t]/
[[Theta] ->t, x->a, y->b, r->1/1000], Han[Pi/2,
Pi-t/2, 1/1000, t][[4]], t3=Han[s/2, 0,1/1000,s][[4]],
HanR[Cos[[Theta]], Sin[[Theta]],x, y, r, s]/
[[Theta]->0, x->s/2, y->0, r->1/1000] 등은 여러
가지 모양의 회전체가 이루는 개형을 연출한다.

```

도형의 회전 기법 2-2

```

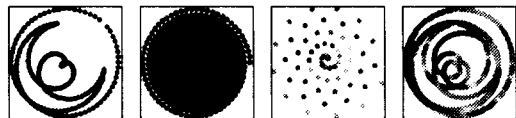
Animate[{{Show[Graphics[Table[{PointSize[0.05],
Hue[7/t],HanR[Cos[[Theta]], Sin[[Theta]],

```

```

x,y,r,t]/.{{Theta}->t,x->Pi/2,y->Pi-t/2,r->1/1000},HanR[Cos[[
Theta]],Sin[[Theta]],x,y,r,s]/.{{Theta}->0,x->s/2,y->0,r->1/100
0}],{t,0.1,0.1n,0.1}],AspectRatio->Automatic}},{Show[Graphi
cs[Table[{PointSize[0.05],Hue[1/t],HanR[Cos[[Theta]],Sin[[Th
eta]],x,y,r,t]/.{{Theta}->0,x->Pi/2,y->Pi-t/2,r->1/1000},HanR[C
os[[Theta]],Sin[[Theta]],x,y,r,s]/
{{Theta}->-s,x->s/2,y->0,r->1/1000}],{s,0.1,0.1n,0.1}],As
pectRatio->Automatic}]],{n,0,200Pi,200Pi/1}]

```



정리 2-3 그림을 그리고 그 것을 바로 지우는 작업은 프로그램의 색조를 조정하여 연출한다. Hue Color나 RGBColor에 이어 GrayLevel[0], RGBColor[1,1,1]을 시행하면 원래의 그래프가 지워지는 효과를 얻는다. 여기에 다른 색을 주고, 다시 지우는 과정을 반복하면 여러 색조의 그래픽을 연계하여 화면에 나타난다. 프로그램 Animate [{{Show[Graphics[Table[{RGBColor[A, B, C]}, ... {Show[Graphics[Table[{RGBColor[1, 1, 1]}]]]}은 RGBColor[A, B, C]의 그림을 그린 다음 흰색으로 다시 출력하여 화면에 아무 것도 없는 효과를 연출한다.

순서 변조 기법 2-3

```

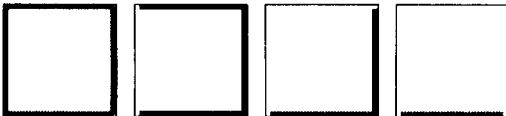
Animate[{{Show[Graphics[Table[{PointSize[0.05],
RGBColor[0.500008,0,0.062501],
t1=Han[0,Pi-t/2,1/1000,t][[4]], t2=Han[s/2,Pi,1/1000,s][[4]],
t3=Han[Pi,Pi-u/2,1/1000,u][[4]], t4=Han[v/2,0,1/1000,v][[4]]
}],{t,0,0.01n,0.01}],AspectRatio->Automatic}},{Show[Graphics
[Table[{PointSize[0.05],RGBColor[0.0742199,
0.457038,0.0429694],
t1=Han[0,Pi-t/2,1/1000,t][[4]], t2=Han[s/2,Pi, 1/1000,
s][[4]], t3=Han[Pi,Pi-u/2,1/1000,u][[4]], t4=Han[v/2,0,1/1000,
v][[4]], {s,0,0.01n,0.01}],AspectRatio->Automatic}]],

```

```

>Show[Graphics[Table[{PointSize[0.05],RGBColor[0.214847,0.22266,0.281254],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {0,0.01n,0.01}],
  AspectRatio->Automatic]],
  Show[Graphics[Table[{PointSize[0.05],RGBColor[0.417975,0.47657,0.0195315],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {v,0,0.01n,0.01}],
  AspectRatio->Automatic]],
  Show[Graphics[Table[{PointSize[0.05],RGBColor[1,1,1],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {t,0,0.01n,0.01}],
  AspectRatio->Automatic]],
  Show[Graphics[Table[{PointSize[0.05],RGBColor[1,1,1],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {s,0,0.01n,0.01}],
  AspectRatio->Automatic]],
  Show[Graphics[Table[{PointSize[0.05],RGBColor[1,1,1],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {u,0,0.01n,0.01}],
  AspectRatio->Automatic]],
  Show[Graphics[Table[{PointSize[0.05],RGBColor[1,1,1],
  1=Han[0,Pi-t/2,1/1000,t][[4]],
  2=Han[s/2,Pi,1/1000,s][[4]],
  3=Han[Pi,Pi-u/2,1/1000,u][[4]],
  4=Han[v/2,0,1/1000,v][[4]],
  {v,0,0.01n,0.01}],
  AspectRatio->Automatic]],
  {n,0,200Pi,200Pi/1}]]

```



IV. 수학 교과에 적용되는 그래픽의 실제

학습시간에 직접적으로 이용할 수 있는 컴퓨터 그래픽 중에서 사인 곡선과 함수의 증감을 연출한다.

사인 곡선 기법 3-1

```

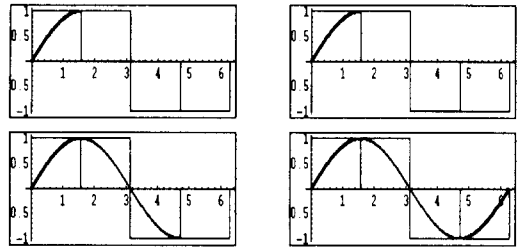
Animate[Show[Graphics[Table[{PointSize[0.01],RGBColor[0,

```

```

0,0], Line[{{Pi/2,0}, {Pi/2,1}}],
Line[{{0,0},{Pi,0},{Pi,1},{0,1}}],Line[{{3Pi/2,0},{3Pi/2,-1}}],
Line[{{Pi,0},{Pi,-1},{2Pi,-1},{2Pi,0}}],HanG[x,y,r,t]
/.{x->t/4,y->Sin[t/4],r->0.0001},{t,0,0.01n,0.01}],
Axes->True, AspectRatio->Automatic],{n,0,800Pi,800Pi/4}]

```

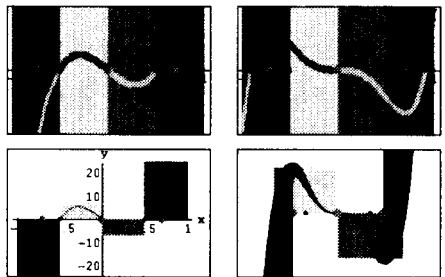


곡선 증감 연출 기법 3-2

```

Plot[g''[x],{x,-1,1},AxesLabel->{"x","y"},PlotStyle->{Thickness[0.01], Hue[1]}, Prolog ->
{{RGBColor[1,0,0],Rectangle[{-1,-25},{-0.5,0}],{RGBColor[1,0,0],Rectangle[{-0.5,0}, {0,7}]},
{RGBColor[0,1,0],Rectangle[{0,-7},{0.5,0}],{RGBColor[0,1,0],Rectangle[{0.5,0}, {1,25}]}}],
Epilog->{{PointSize[0.03], Hue[0.7], Point[{0,0}],Point[{-1/2,0}],Point[{1/2,0}],
{PointSize[0.05],Hue[0.1],Point[{0,0}],Point[{-1/(Sqrt[2]),0}],Point[{1/(Sqrt[2]),0}]}}]

```



함수의 구간 증감 연출 기법 3-4

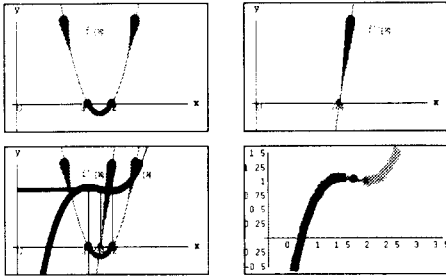
```

Animate[Show[Graphics[Table[{RGBColor[1,0,1],Text["O",{0.2,-0.1},{1,0}],{RGBColor[1,0,1],Text["7/4",{7/4+0.2,-0.1},
{1,0}],{PointSize[0.04],RGBColor[1,0,1],Point[{7/4,0}],{RGBColor[1,0,1],Text["7/4",{7/4+0.2,-0.1},
{1,0}],{RGBColor[1,0,0],PointSize[0.7],Text["f'[x]",{3,1.3},{1,0}],{RGBColor[0.1,0.7,0.1],Han[7t/24,f''[7t/24],0,

```



```
001,0.01,t]],{RGBColor[0.5,0.2,0.4],Han[7/4+7t/24,f'[7/4+7t/24],0.001,0.07,t]],{t,0,0.01n,0.01]],Axes->True,Ticks->False,PlotRange->{-0.5,1.5},AxesLabel->{"x","y"},AspectRatio->Automatic]],{n,0,600,600/1}]
```



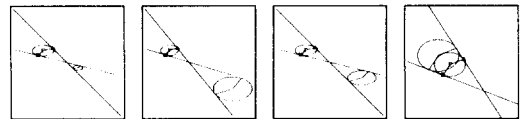
V. 결론

순수수학 이외의 제반 학문은 3차원 Euclid 공간을 활용하는 것으로 응용수학의 대부분은 그래픽의 효능을 제고함으로써 학습효과를 높일 수 있다. 탁월한 기능을 갖는 소프트웨어의 개발과 이용으로 수학학습공학이 발전하여 재래의 칠판 수업형태를 전산시스템으로 구성된 시청각교육으로 바꾸어 놓았으며 수학교과내용 자체를 전산화하여 순수이론은 수식, 그의 구현은 그래픽으로 재편하여 학습자에게 제시할 수 있게 되었다. 이런 노력은 주입식 수학학습 형태가 갖는 정체성을 탈피하여 눈에 보이는 수학, 움직이는 수학으로 형상화할 수 있는 장점이 있다. 이 논문은 수식에 의한 그래픽을 여러 가지 측면에서 출현하여 수학적 사실을 직관적으로 학습자에게 보여주는 C-언어에 의한 입출력을 소개한 것으로 다단계입출력은 출력이 가능한 범위에서 학습자가 수식의 의미를 기하학적으로 생각할 여유를 찾게 한다. 함수의 그래프가 갖는 형상을 개괄하는데 필요한 미분, 도함수, 누차도함수의 그래픽을 화면에 적절히 조화하여 수업하면 컴퓨터의 역할을 향

상시키고 수학의 역동성을 직시하게 할 수 있다. 여기에는 본인이 갖고 있는 “함수의 요철에 관한 그래픽”, “수학의 역동성”, “한글자모의 조형미”, “도형의 회전이 연출하는 아름다움” 등의 디스켓의 내용 중에서 현장 수학학습에 유용하게 활용될 수 있는 그래픽의 플루트와 Animation의 기법의 몇 가지 원리만을 제시한 것이다. 이런 노력으로 수학교육과정 그 자체를 전산화하여 PC로 학습자 개개인이 독자적으로 학습할 수 있는 프로그램의 개발과 보급에 현직 수학교육자들의 체계적 연구의 기초가 되길 기대한다. 예를 들어 각의 이등분선이 연출하는 컴퓨터 그래픽은 간단하면서도 수학 학습의 다양화를 실현시키는 전형적인 모델로 간주할 수 있다.

각의 이등분선 연출 기법

```
Animate[Show[Graphics[{{Circle[{-t,t},Abs[t]/Sqrt[5]],Line[{-5,5/2},{5,-5/2}],Line[{-5,10},{5,-10}],Line[{-6 t/5,(3 t)/5},{-t,t},{-(3t)/5,(6 t)/5}],Pont[{-t,t}],Point[{-6 t/5,(3 t)/5}],Point[{-3t/5,(6t)/5}],{Hue[0.5],Circle[{-Sqrt[5],Sqrt[5]},1],Line[{-3/Sqrt[5],6/Sqrt[5]},{-Sqrt[5],Sqrt[5]},{-6/Sqrt[5],3/Sqrt[5]}]},Text["P",{-Sqrt[5],Sqrt[5]},Text["A",{-3/Sqrt[5],6/Sqrt[5]},Text["B",{-6/Sqrt[5],3/Sqrt[5]}]}]}],AspectRatio->Automatic] ,{t,-5,5,2}]
```



참고 문헌

- Eugene W. Johnson, Brooks/Cole Publishing Company, 1995. Cincinnati, Washington, 1995.
- Marta L. Abell, James P. Braseton, Differential

Equation with Mathematica. Academic Press, 1977.

Thom Wickham, James, Mathematica Graphics, Springer-Verlag, 1944.

Stwphan Wolfram, The Mathematica(3rd,ed.), Cambridge Univ. Press, 1988.

Department of Mathematics Chungbuk National University CheongJu, 361-763, Korea

Computer programs and practical examples on education of mathematics

Han Jae Young

This paper is composed of computer programs and practical examples for the education of mathematics. The Animation package in the *Mathematica* standard packages is the Graphics Animation. Animations are the ways which produce a sequence of different pictures by rapid succession. The command Animate has a number of options such as Animate, SpinShow and

ShowAnimation. *Mathematica* is available for a large variety of practical education of mathematics. The method in this report are familiar with the general principles of operating the computer in *Mathematica* 3.0. In this paper there a set of useful examples of mathematical education and we discusses some of the advanced technique for the Input-Output systems.