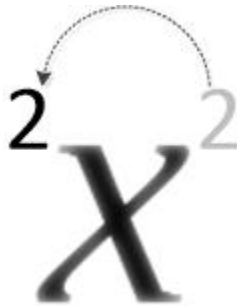


2장. 기본함수: 파워함수



파워함수

파워함수의 역함수

파워함수의 접하는 직선 함수

선, 직선조각, 반직선

벡터

베이스스(기저, Basis)

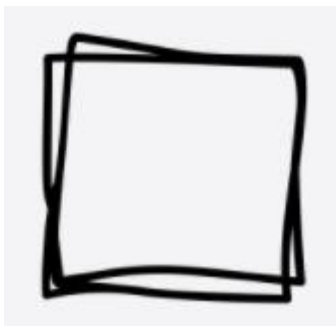
선형조합(Linear Combination)

복잡한 수(Complex Number)

이야기의 흐름

- 기본함수 → 벡터의 내적과 외적 → 복소수의 곱셈 → 아름다운 공식 → 수학의 완벽성과 과학의 한계
- 변환 → 우리 인식의 한계 → 시간의 상대성 → 컴퓨팅의 미래 → 창조, 구속, 타락 → 복음

(위 내용을 한 장의 그림으로 그려서 매 장(chapter)의 앞 부분에 현재의 위치를 설명하고 요약한 후에, 본 내용을 시작할 것, 20220414)





파워함수(Power Function)

고등학교 수학에서 배우는 모든 함수(function)는 **기본함수** (*elementary function*)의 범주에 속합니다. 실제로 컴퓨터 프로그래밍을 해 보면, 일반적인 프로그래밍에 사용되는 대부분의 함수들은 기본함수로 표현할 수 있습니다.

놀랍게도 기본함수는 모두 여섯 종류(본함수 세 개 + 역함수 세 개)입니다. 그리고 우리의 첫 번째 목표는 기본함수와 그들의 관계를 이해하는 것입니다. 기본 함수는 다음과 같이 (그림2-1)로 나타낼 수 있습니다.

	원래함수	역함수 (inverse function)
파워함수(power function)	(1)	(1)의 역함수
지수함수(exponential function)	(2)	(2)의 역함수
투영된 선분길이함수	(3)	(3)의 역함수

(그림2-1) 기본함수표: 이 인벤토리를 구성하는 모든 항목을 수집하면, 우리는 첫 번째 목표를 이룬 것입니다.

(그림2-1)을 보면 기본함수의 형태는 모두 세 가지이며, 나머지 세 가지는 기본 형태의 역함수로 구성됩니다. 우리는 (그림2-1)의 비워진 부분을 계속 채워나갈 것입니다. (그림2-1)은 우주의 비밀을 풀어나가는 게임에서 주인공이 갖춘 인벤토리(inventory)라고 생각하면 됩니다. 우리가 인벤토리의 비워진 곳을 모두 원하는 아

이템으로 채웠을 때, 이 게임은 끝나게 될 것입니다.

우리가 제일 먼저 살펴볼 함수는 **파워함수(멱함수, 누승함수, power function)**입니다. 파워함수는 지수 형태의 함수 몸체에서 밑(base)이 독립변수이고 지수 부분이 **상수(constant)**인 함수를 말합니다.

우리는 이미 1장에서 **제곱함수(square function)**을 살펴보았는데, 제곱함수는 파워함수의 간단한 예입니다.

$$f(x) = x \times x = x^2 \text{ (식2-1)}$$

(식2-1)은 독립변수 x 를 두 번 곱하는 일을 합니다. 이제 x 를 세 번 곱하는 일을 하는 함수를 다음과 같이 정의할 수 있습니다.

$$f(x) = x \times x \times x = x^3 \text{ (식2-2)}$$

x 를 n 번 곱하는 함수를 다음과 같이 정의할 수 있습니다.

$$f(x) = x^n \text{ (식2-3)}$$

기본함수의 덧셈에 의한 조합 역시 기본함수이므로, 아래 (식2-4) 역시 기본함수입니다.

$$f(x) = 2x^3 + 3x^2 \text{ (식2-4)}$$

(식2-4)를 보면 $2x^3$ 처럼 곱셈(나눗셈)으로만 구성된 덩어리를 **항(term)**이라고 합니다. 항에서 상수부분을 **계수(coefficient)**라고 합니다. $2x^3$ 항의 계수는 2입니다. (식2-4)는 두개의 항으로 구성된 함수입니다.

$$f(x) = 2x^3 + 3x^2$$

$\underbrace{\hspace{1.5cm}}$
항(term)

 $\underbrace{\hspace{1.5cm}}$
항(term)

 \uparrow
계수(coefficient)

(그림2-1b) $f(x)$ 는 2개의 항으로 이루어져있습니다. 항을 이루는 구성 요소에서 상수인 부분을 계수(coefficient)라고 합니다. $2x^3$ 항의 계수는 2입니다.

이 책에서는 설명을 간단하게하기 위해, 기본함수를 설명할 때, 두개 이상의 항으로 구성된 함수는 다루지 않겠습니다. 이제 우리는 기본 함수표의 첫 번째 함수를 찾았습니다. 그래서 (그림2-2)를 얻었습니다. (그림2-2)에서 x 는 변수(variable)이고 n 은 상수(constant)입니다.

	원래함수	역함수 (inverse function)
파워함수(power function)	x^n	(1)의 역함수
지수함수(exponential function)	(2)	(2)의 역함수
투영된 선분길이함수	(3)	(3)의 역함수

(그림2-2) 기본함수표: 파워함수를 찾았습니다.

파워함수의 역함수

우리는 1장에서 함수가 정의되면 (1) 함수를 그리고, (2) 함수의 역함수를 찾고, (3) 함수의 접하는 직선의 함수를 찾는 것을 이해할 수 있어야 한다고 했습니다.

이제 x^n 의 역함수를 다음과 같이 정의할 수 있습니다.

$${}_n x \text{ (식2-5)}$$

(식2-5)는 "***n*번 곱해서 x 가 되는 수**"를 의미합니다. 예를 들면 ${}^4 16$ 은 4번 곱해서 16이 되는 수이므로, 2를 의미합니다. (${}^4 16$ 의 실제 수학 표현은 ${}^4 \sqrt{16}$ 입니다)

$${}^4 16 = {}^4 (2^4) = 2 \text{ (식2-6)}$$

$${}^2 4 = 2$$

$${}^2 9 = 3$$

$${}^3 8 = 2$$

$${}^3 27 = 3$$

파워 함수의 역함수를 정의하다 보면, 실수(real number)로는 답을 구할 수 없는 이상한 상황이 발생합니다. (식2-7)을 만족하는 실수는 존재하지 않습니다.

$${}^2 (-1) \text{ (식2-7)}$$

$$\square \times \square = -1$$

(그림2-3) $^2(-1)$ 은 제곱하면 -1 이 되는 이상한 수를 의미합니다. 표준 수학기호 $\sqrt{\quad}$ 를 사용하여 나타내면 $\sqrt[2]{-1} = \sqrt{-1}$ 이 됩니다.

(식2-7)은 **제공해서 -1이 되는 수**를 의미합니다. 실수에는 이러한 수가 없으므로, 우리는 새로운 수에 대한 개념을 정의할 필요를 느낍니다. 일단 이상한 수를 무시하고 기본함수표의 역함수 부분을 채우면 (그림2-3b)와 같습니다.

	원래함수	역함수 (inverse function)
파워함수 (power function)	x^n	$^n x$
지수함수 (exponential function)	(2)	(2)의 역함수
투영된 선분길이함수	(3)	(3)의 역함수

(그림2-3b) 기본함수표: 파워함수의 역함수 $^n x$ 를 추가했습니다.



파워함수의 접하는 직선 함수

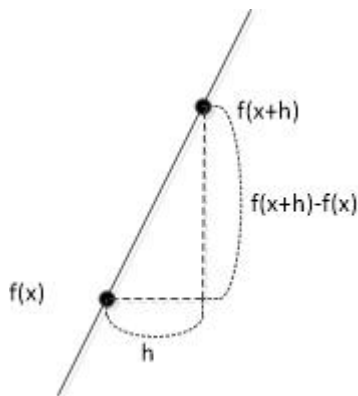
함수가 주어지면 역함수를 찾는 것뿐만 아니라, 접하는 직선을 정의하는 함수를 찾는 것이 중요하다고 했습니다. 우리는 1장에서 이미 x^2 에 대해서는 접하는 직선을 정의하는 함수를 찾았습니다. 그것을 (식2-8)과 같이 f^{slope} 로 나타내었습니다.

$$f(x) = x^2$$

$$f^{slope} = 2x^{2-1} = 2x^1 = 2x \quad (\text{식2-8})$$

비슷한 방식으로 x^n 에 대해서 "접하는 직선을 정의하는 함수"를 찾을 수 있습니다. $f(x) = x^n$ (식2-3)에 대한 접하는 직선을 정의하는 함수는 (식2-9)와 같습니다.

$$f^{slope} = nx^{n-1} \quad (\text{식2-9})$$



(그림2-4) 임의의 함수 $f(x)$ 에 대해서 접하는 직선의 기울기

$f^{slope} = nx^{n-1}$ (식2-9)를 구하기 위해서, 임의의 함수 $f(x)$ 에 대해서 접하는 직선의 기울기를 구하는 과정을 살펴봅시다. (그림 2-4)는 $f(x)$ 에 대해서 x 의 변화량 h 가 무한대로 작아질 때, $(f(x+h)-f(x))/h$ 를 구하는데, 이것은 접하는 직선의 기울기입니다. 왜냐하면 분모(denominator) $h = (x+h)-x$ 는 x 의 변화량을 의미하고, 분자(numerator)는 y 의 변화량을 의미하기 때문입니다. (식2-9)를 찾기 위해서 우리는 "무한에 대한 직관"을 사용하여, 무한대로 작아지는 h 에 대해서 다음과 같은 (식2-10)을 풀어야 합니다.

$$\text{무한대로 작아지는 } h \text{에 대해 } \frac{f(x+h)-f(x)}{h} \quad (\text{식2-10})$$

(식2-10)은 임의의 값 x 에서 직선의 기울기를 구한다는 의미입니다. "무한대로 작아지는 h 에 대해"를 수학적인 기호를 사용하여 다음과 같이 (식2-11)처럼 \lim 으로 나타내도록 합니다.

$$\infty_{h \rightarrow 0} \quad (\text{식2-10b})$$

$$\lim_{h \rightarrow 0} \quad (\text{식2-11})$$

$\lim_{h \rightarrow 0}$ (식2-11)은 "**리미트 h 가 0으로 갈 때**" 영어로는 "limit when h approaches 0"라고 읽습니다.

그러면 (식2-10)은 다음과 같이 (식2-12)로 적을 수 있습니다.

$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (\text{식2-12})$$

$f(x) = x^n$ 을 (식2-12)에 대입하면 다음과 같이 (식2-13)을 얻습니다.

$$\lim_{h \rightarrow 0} \frac{(x+h)^n - (x)^n}{h} \quad (\text{식2-13})$$

(식2-13)은 **이항정리(binomial theorem)**를 이용하면 nx^{n-1} 이 되는 것을 알 수 있는데, 결과를 유도하는 과정은 별도로 설명하지는 않습니다.(시그마(sigma, Σ) 표기법, 조합(combination, C_k^n)등의 추가적인 개념 설명이 필요하기 때문입니다)

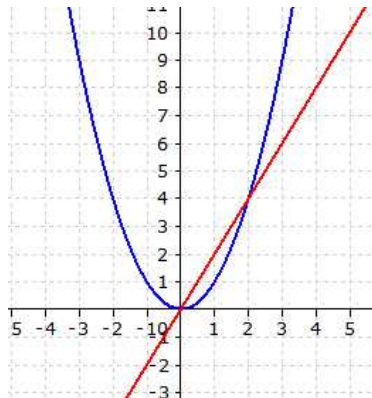
$$\lim_{h \rightarrow 0} \frac{(x+h)^n - (x)^n}{h} = nx^{n-1}$$

이제 기본함수표에 "접하는 직선을 정의하는 함수"와 이상한 수 (weird number)를 같이 표시해 보도록 하겠습니다. 그것은 (그림 2-4)와 같습니다.

	원함수	접하는 직선	역함수	이상한 수
파워함수	x^n	nx^{n-1}	${}_n x$	${}^2(-1)$
지수함수				
투영된 선분길이 함수				

(그림2-4) 기본함수표: 파워함수에 해당하는 줄을 모두 완성했습니다.

원함수와 접하는 직선 함수를 함께 그래프로 그리는 것은 종종 도움이 됩니다. x^2 과 대응하는 접하는 직선 $2x$ 를 그리면 (그림 2-4b)와 같습니다.



(그림2-4b) x^2 과 접하는 직선 $2x$ 의 그래프입니다. x 가 1일 때 $2x = 2 \times 1 = 2$ 인데, x^2 그래프가 $x = 1$ 일 때 접하는 접선의 기울기가 2라는 의미입니다.

이제 우리는 기본함수표에서 파워함수 부분을 모두 완성했습니다. 다음 함수를 알아보기 전에, 제곱하면 -1 이 되는, 이상한 수 $^2(-1)$ 을 이해할 필요가 있습니다. 그러기 위해서는 1차원에 머물러 있는 수의 개념을 보다 높은 차원으로 확장해야 합니다.

표준수학

미분연산자

$f^{slope}(x)$ 의 표준 수학기호는 $f'(x)$ 이며 “에프 프라임 오브 x (*f prime of x*)”라고 읽습니다. **미분 연산자(differentiation operator)**를 도입하면 (식2-13b)처럼 적을 수 있습니다.

$$f'(x) = \frac{df(x)}{dx} \quad (\text{식2-13b})$$

(식2-13b)가 의미하는 것은 x 값 위치에서 무한히 작게 x 값이 증가했을 때, $f(x)$ 가 변화된 정도에 대한 비율입니다. $(x, f(x))$ 에서 접하는 직선의 기울기를 구한다는 의미입니다.

$$\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{(x+h) - x} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (\text{식2-13c})$$

미분 연산자를 (식2-13b)와 같이 정의하는 것은 매우 유용합니다. 다음과 같은 식을 고려해 봅시다.

$$t = f(x) \quad (\text{식2-13d})$$

$$y = g(t) \quad (\text{식2-13e})$$

$t = f(x)$ (식2-13d)는 x 에 대한 t 의 함수입니다. t 함수의 접하는 직선의 기울기를 다음과 같이 구할 수 있습니다.

$$\frac{dt}{dx} \quad (\text{식2-13f})$$

$y = g(t)$ (식2-13e)는 t 에 대한 y 의 함수입니다. y 함수의 접하는 직선의 기울기를 다음과 같이 구할 수 있습니다.

$$\frac{dy}{dt} \quad (\text{식2-13g})$$

이제 입력 x 에 대해 y 를 구하는 합성함수(composite function) $g \cdot f(x)$ 를 고려해 봅시다.

$$\begin{aligned} y &= g(t) & (\text{식2-13h}) \\ y &= g(f(x)) \\ y &= g \cdot f(x) \end{aligned}$$

(식2-13h)가 정의한 함수에 대해 접하는 직선의 기울기를 구하는 함수는 다음과 같이 구할 수 있습니다.

$$\frac{dy}{dx} \quad (\text{식2-13i})$$

미분연산자의 **분자(numerator)와 분모(denominator)**를 순수한 실수라고 가정하면 $\frac{dy}{dx}$ 를 아래 (식2-13j)와 같이 나타낼 수 있습니다.

$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx} \quad (\text{식2-13j})$$

(식2-13j)에서 보면 분모와 분자에 사용된 dt 가 수(number)를 다

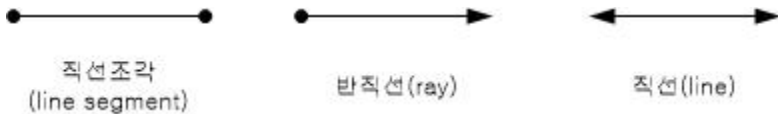
루듯이 **약분(simplifying fraction)**된 것을 알 수 있습니다. 이러한 특징 때문에 (식2-13b)와 같이 미분 연산자를 정의하는 것입니다. (식2-13j)는 **체인 룰(chain rule, 합성함수의 미분)**로 알려진 미분 규칙인데 다양한 공학 분야에서 매우 중요하고 빈번하게 사용합니다.



선, 직선조각, 반직선 (line, line segment, ray)

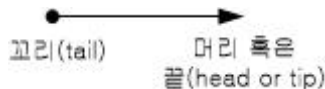
이상한 수 $^2(-1)$ 를 이해하기 위해 우리는 **벡터(vector)**를 이해하고, 벡터의 **선형독립(linearly independence)**과 **스팬(span)**의 개념을 이해해야 합니다. 벡터의 개념을 이해하기 위해 먼저 필요한 용어를 정의하도록 하겠습니다.

선에 대해서 양끝점이 정해지지 않은 선을 **직선(line)**이라고 합니다. 직선의 한쪽 끝점이 정해지면 **반직선(광선, ray)**이라고 하며, 양 끝점이 모두 정해지면 **직선 조각(라인 세그먼트, line segment)**이라고 합니다.



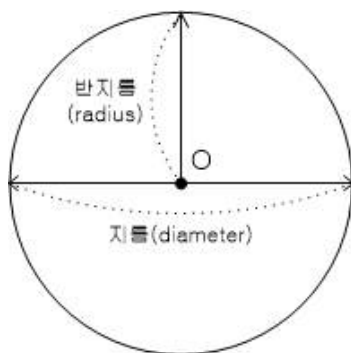
(그림2-5) 직선 조각, 반직선과 직선

직선 조각의 경우 양 끝점을 시작점(begin point), 끝점(end point)이라고 하며, 반직선의 경우 시작하는 끝점을 **테일(꼬리, tail)**이라고 하고, 반직선이 진행하는 방향의 끝점을 **헤드(머리, head)**라고 합니다. 헤드는 화살표의 머리(head)가 진행하는 방향을 의미합니다.



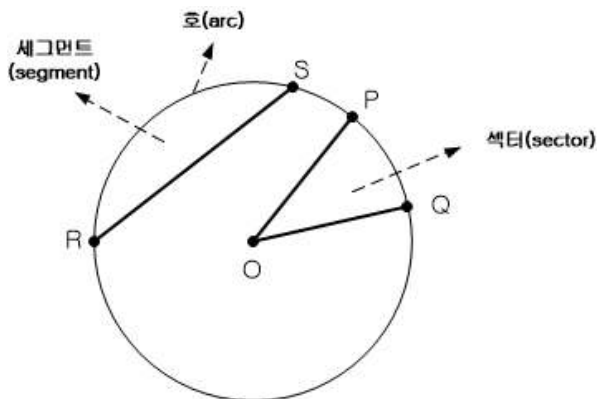
(그림2-6) 반직선의 테일과 헤드: 반직선의 시작점을 테일(tail)이라고 하며, 방향이 진행하는 끝점을 헤드(head)라고 합니다.

이제 원(circle)과 관련된 용어를 살펴보도록 하겠습니다.



(그림2-7) 원의 지름과 반지름

원의 중심 O를 지나는 직선이 원의 두 점과 만나서 이루는 반 직선의 길이를 **지름(diameter)**이라고 합니다. 지름의 절반 길이를 **반지름(radius)**이라고 합니다.

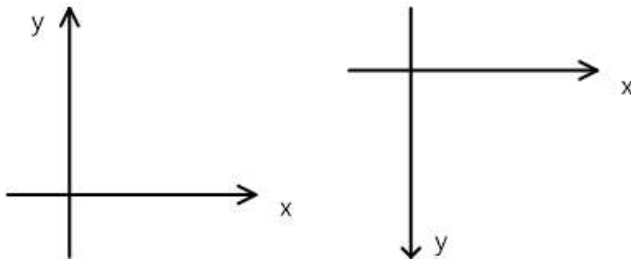


(그림2-8) 섹터(부채꼴, sector), 세그먼트와 원호(아크, arc)

(그림2-8)의 원의 지름에서 시작하는 두 개의 반직선이 원과 만나는 점을 각각 P와 Q라고 하면, O,P,Q가 이루는 원의 일부를 **섹터(부채꼴, sector)**라고 합니다.

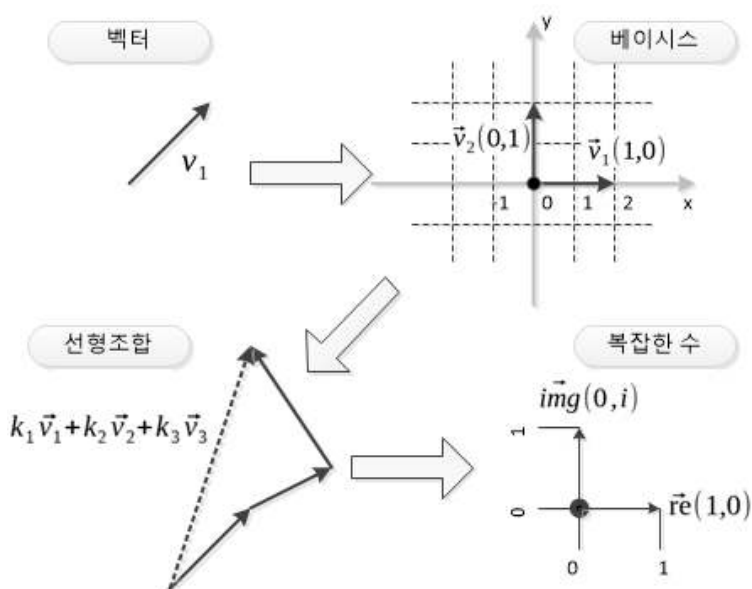
임의의 직선이 원을 가로지르면, 만나는 두 점을 R과 S라고 하면, R과 S가 잘라내는 원의 일부를 **세그먼트(segment)**라고 합니다. 그리고 세그먼트 혹은 섹터를 이루는 원 둘레의 곡선 일부를 **아크(원호, arc)**라고 합니다.

우리는 2차원에 대해서 다음 (그림2-9)와 같은 축을 사용할 수 있습니다.



(그림2-9) 2차원 축: 왼쪽의 2차원 축은 데카르트 좌표계 (Cartesian coordinate system)로 알려진 대표적인 축입니다. 오른쪽의 2차원 축은 y의 방향이 바뀐 것을 알 수 있습니다.

현재 우리의 목표는 이항수 -1 을 이해하는 것인데, 이항수 -1 과 일반적인 수(real number)를 모두 표현할 수 있는 새로운 수 체계를 정의하려는 것입니다. 이러한 수 체계를 정하려면, 수를 나타내는 방법, 수에 대한 연산을 정의해야 합니다. 수를 나타내고 수에 대한 연산을 정의하기 위해서 우리는 벡터(vector)를 사용할 것인데, 그러기 위해서는 (그림2-10)과 같은 절차로 진행될 내용을 이해해야 합니다.



(그림2-10) 벡터 → 베이스스 → 선형조합 → 새로운 수

이제부터 차례대로 벡터 → 베이스스 → 선형조합을 설명할 예정입니다. 약간 어려울 수도 있는데, 한 번 읽어보고 막힌다면, 계속해서 읽어서 이해한 후에 진행할 것을 권고합니다. 왜냐하면 이 부분이 "변환", "로렌츠 변환"을 이해하기 위한 필수적인 단계이기 때문입니다.

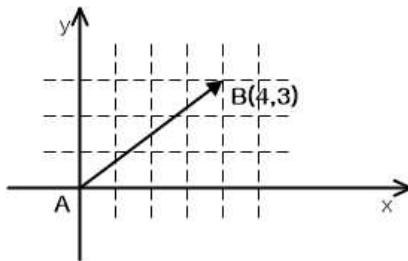
"벡터"에서는 크기와 방향을 나타내는 방법에 대해서 배웁니다. "베이스스"에서는 좌표축이 아니라, 벡터를 이용해서 좌표를 나타내는 방법을 배웁니다. "선형조합"에서는 벡터 덧셈의 기하학적인 의미를 이해합니다. 그리고 마침내 "복잡한 수"를 정의합니다.

벡터(Vector)

실수(real number)와는 다르게 **벡터(vector)**는 **크기(magnitude)**와 **방향(direction)**을 가집니다. 크기만을 가지는 값을 **스칼라(스케일러, scalar)**라고 하는데, 한글로 번역된 "스칼라"라는 발음이, 학자를 의미하는 Scholar를 연상시켜서 필자는 원래 영어 발음인 "**스케일러(scalar)**"라고 부르는 것을 좋아합니다. 스케일(scale, 크기)에서 파생된 scalar라는 단어는 "크기와 관련된 양"를 의미합니다.

예를 들어 자동차가 **속력(speed)** 100km/h로 달리고 있을 때 속력은 스칼라 값입니다. 속력은 방향을 고려하지 않습니다. 자동차가 부산에서 대구 방향으로 **속도(velocity)** 100km/h로 달린다면 이는 벡터입니다. 속도의 크기는 100이며 방향은 부산에서 대구를 나타냅니다.

시작점(initial point) A와 끝점(terminal point) B로 표현되는 벡터 v 는 $\vec{v} = \overrightarrow{AB}$ 로 나타냅니다. 변수나 상수로 사용하는, 일반 영문자 v 와 구분하기 위해 화살표(\rightarrow)를 v 의 위에 명시해서 \vec{v} 처럼 나타냅니다.



(그림2-11) 벡터의 표현: 벡터 $\vec{v} = \overrightarrow{AB}$ 는 $(4,3)$ 혹은 $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ 이라고 나타냅니다.

벡터 $\vec{v} = \overrightarrow{AB}$ 는 **행벡터(row vector)** $(4,3)$ 혹은 **열벡터(column vector)** $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ 으로 나타냅니다. $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ 표현에서 대괄호 대신에 괄호를 사용해서 $\begin{pmatrix} 4 \\ 3 \end{pmatrix}$ 으로 나타낼 수도 있습니다. 일반적으로 $(4, 3)$ 보다는 $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ 을 선호합니다. 왜냐하면 $(4, 3)$ 이라는 표현은 점(point)을 표현하는 것인지, 벡터를 표현하는 것인지 모호함이 발생할 수 있기 때문입니다.

이 책에서는 $(4, 3)$ 과 $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ 을 혼용해서 사용할 것입니다. 그리고 점과의 구분이 명확히 필요한 경우 $(4, 3)$ 이 아니라, **벡터 $(4, 3)$** 혹은 $\vec{v}(4,3)$ 으로 표현할 것입니다. 그렇게 하는 이유는 벡터 $(4, 3)$ 은 문장의 일부로 사용할 때 높이가 다르지 않아 자연스럽게 문장의 일부로 표현되기 때문입니다.

수(number)가 주어지면, 수에 대한 연산을 정의할 수 있습니다. 기본적인 사칙연산은 더하기, 빼기, 곱하기, 나누기 이며 각각을 다음과 같은 기호로 표시합니다.

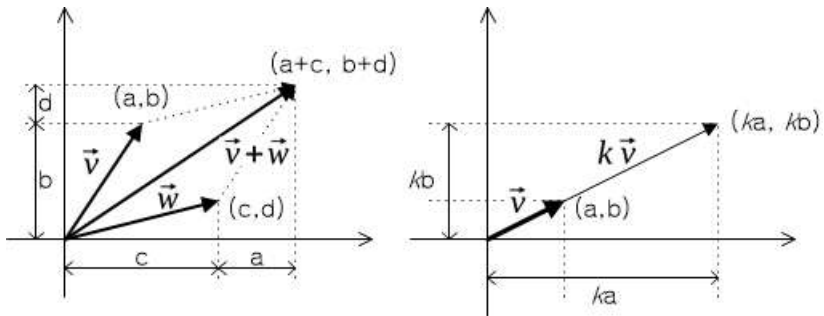
$$+, -, \times, /$$

위의 **사칙연산자(basic four arithmetic operator)**는 **피연산자(operand)**로 수(number)를 가집니다. 마찬가지로 벡터가 주어지면, 벡터에 대한 연산을 (식2-14)와 같이 정의할 수 있습니다. 벡터의 곱셈은 (식2-15)와 같이 다양한 목적을 위하여 정의할 수 있는데, 이 부분은 “변환”을 설명할 때 다루도록 하겠습니다.

$$\vec{v} + \vec{w}, \vec{v} - \vec{w}, \vec{v} \times \vec{w}, \vec{v} / \vec{w} \text{ (식2-14)}$$

$$\vec{v} \cdot \vec{w}, \vec{v} \times \vec{w}, \vec{v} \otimes \vec{w} \text{ (식2-15)}$$

벡터에는 여러 가지 연산이 있지만, 가장 간단한 연산은 **덧셈** (vector addition)과 **스칼라 곱셈**(scalar vector multiplication)입니다.



(그림2-12) 벡터의 덧셈: 벡터의 합은 각 요소(component)를 합한 것입니다. 벡터와 스칼라 k 의 곱은 벡터의 각 요소에 k 를 곱한 것입니다.

벡터 $\vec{v} = (a, b)$ 와 $\vec{w} = (c, d)$ 가 주어졌을 때, $\vec{v} + \vec{w}$ 의 의미는 \vec{v} 만큼 이동한 후에, \vec{v} 의 끝점에서 \vec{w} 만큼 이동하라는 의미입니다. 이것은 위 (그림2-12)에서 보듯이 다음과 같이 나타낼 수 있습니다.

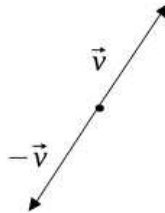
$$\vec{v} + \vec{w} = (a, b) + (c, d) = (a + c, b + d) \text{ (식2-16)}$$

벡터 $\vec{v} = (a, b)$ 와 임의의 실수값 k 에 대해서 곱셈을 정의할 수

있습니다. 그것은 \vec{v} 방향은 유지한 채로 크기를 k 배 변경시킨다는 의미로 다음과 같이 정의합니다.

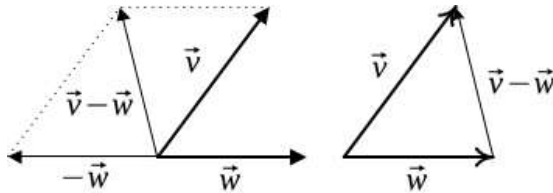
$$k\vec{v} = k(a, b) = (ka, kb) \quad (\text{식2-17})$$

k 가 -1이면 \vec{v} 의 **음 벡터(negative vector)**를 이해할 수 있습니다. 음 벡터는 크기는 유지한 채로 방향이 반대가 되는 벡터입니다. (그림2-13)은 \vec{v} 와 $-\vec{v}$ 의 관계를 보여줍니다.



(그림2-13) 음(negative) 벡터

$\vec{v} = (a, b)$ 일 때 $-\vec{v} = -1(a, b) = (-1a, -1b) = (-a, -b)$ 입니다.

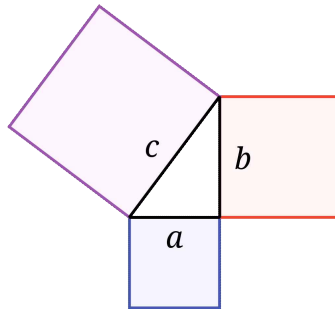


(그림2-14) 벡터의 뺄셈(subtraction): 벡터의 뺄셈 $\vec{v} - \vec{w}$ 는 \vec{v} 에 $-\vec{w}$ 를 더한 것과 같습니다

벡터 덧셈의 정의에 의해서, $\vec{v} - \vec{w} = (a - c, b - d)$ 입니다.

벡터 $\vec{v} = (a, b)$ 의 **길이(length, norm)**는 $|\vec{v}|$ 로 나타내며, 다음 (식2-18)과 같습니다. (식2-18)은 **피타고라스의 정리(Pythagorean theorem)**에 의해서 유도 가능합니다.

$$|\vec{v}|^2 = a^2 + b^2 = \sqrt{(a^2 + b^2)} \quad (\text{식2-18})$$



(그림2-14b) 직각삼각형의 빗변(hypotenuse), 반대 변(opposite), 인접한 변(adjacent)을 c , b , a 라고 하면 $c^2 = a^2 + b^2$ 가 성립합니다.

피타고라스의 정리는 (그림2-14b)처럼 직각 삼각형(right triangle)의 빗변(hypotenuse), 반대 변(opposite), 인접한 변(adjacent)을 c , b , a 라고 했을 때, 각 변이 이루는 정사각형(square)의 넓이의 관계를 나타냅니다. (그림2-14b)에서 관계는 다음 (식2-18b)와 같습니다.

$$c^2 = a^2 + b^2 \quad (\text{식2-18b})$$

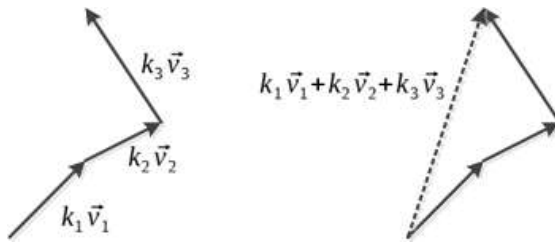
벡터의 방향 성분만을 고려하기 위해서 $|\vec{v}|$ 가 1이 되게 \vec{v} 를 바꿀 수 있는데, 이러한 과정을 **정규화(노멀라이즈, normalize)**라고

합니다. 임의의 수 n 을 n 으로 나누면 $n/n = 1$ 이 됩니다. 마찬가지로, 벡터 \vec{v} 를 정규화하기 위해서는 (식2-19)처럼, 벡터의 각 성분을 $|\vec{v}|$ 로 나누어 주어야 합니다. 정규화된 벡터는 화살표 대신에 **햇(^, hat)**기호를 사용해서, \hat{v} ("브이 햇"이라고 읽습니다)이라고 나타냅니다.

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|} = \frac{(a, b)}{|\vec{v}|} = \left(\frac{a}{|\vec{v}|}, \frac{b}{|\vec{v}|} \right) \quad (\text{식2-19})$$

$|\vec{v}|$ 와 정규화에 대한 추가적인 설명은 마지막 기본함수를 설명할 때 하려고 합니다.

주어진 수식을 직관적으로 이해하는 것은 많은 도움이 됩니다. 벡터의 덧셈과 스칼라 곱셈을 직관적으로 어떻게 이해해야 할까요?



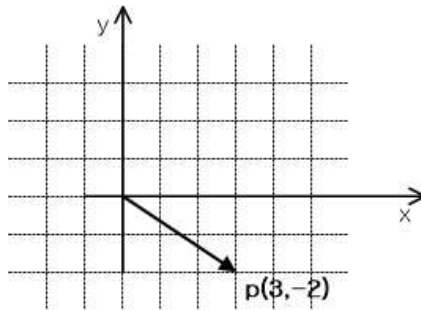
(그림2-14c) 벡터의 덧셈의 의미

(그림2-14c)를 보면 벡터의 덧셈은 벡터가 이루는 직선(line)의 끝점을 이어서 새로운 직선을 만드는 과정이라고 생각할 수 있습니다. 이것을 **선들의 조합(linear combination)**이라고 부르도록 하겠습니다.

■ 베이스스(기저, Basis)

우리는 이제 **베이스스(기저, basis)**와 **선형 조합(linear combination)**에 대해서 이해할 준비가 되었습니다.

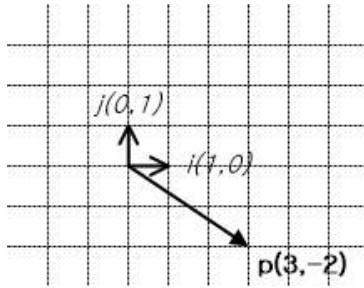
데카르트 표준 좌표계에 벡터 $\vec{p}(3, -2)$ 를 나타내어 봅시다. 그러면 $\vec{p}(3, -2)$ 는 아래 (그림2-15)와 같이 나타낼 수 있습니다.



(그림2-15) 벡터 $\vec{p}(3, -2)$

위 그림에서 x 와 y 는 벡터가 아니라 좌표축입니다. $\vec{p}(3, -2)$ 벡터의 3이 의미하는 것은 x -축을 따라 3단위(unit) 만큼, y -축을 따라 -2단위 만큼 이동한 위치가 벡터의 끝(tip)이 된다는 의미입니다.

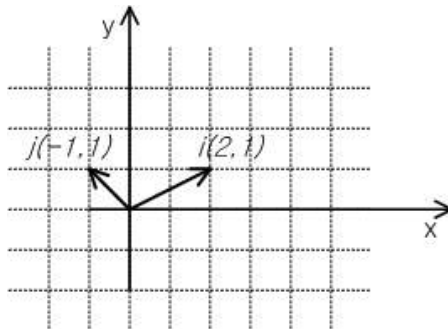
이제 축(axis) 대신에 x -축을 따라 단위 길이가 1인 벡터 $\hat{i}(1,0)$ 와, y -축을 따라 단위 길이가 1인 벡터 $\hat{j}(0,1)$ 를 생각해 볼 수 있습니다. 이것을 각각 \hat{i} , \hat{j} 으로 나타낸다고 합시다. 그러면 이제 벡터 $\vec{p}(3, -2)$ 를 \hat{i} 와 \hat{j} 의 덧셈으로 나타내는 것이 가능합니다.



(그림2-16) 베이스 벡터 $i(1,0)$ 와 $j(0,1)$

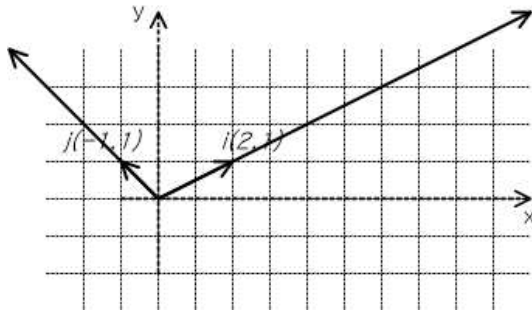
$$3\hat{i} - 2\hat{j} = 3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (\text{식2-20})$$

$3\hat{i}$ 는 스칼라 값 3과 벡터 $\hat{i}(1,0)$ 의 곱셈을 의미합니다. $-2\hat{j}$ 는 스칼라 값 -2와 벡터 $\hat{j}(0,1)$ 의 곱셈을 의미합니다. \hat{i}, \hat{j} 와 벡터의 덧셈을 이용하면 2차원 평면의 모든 벡터를 나타내는 것이 가능합니다. 이렇게 주어진 좌표계의 각 축을 구성하는 기본적인 벡터를 **베이스(basis, 기저)**라고 합니다. 이제 $\hat{i}(1,0)$ 과 $\hat{j}(0,1)$ 이 아니라 다른 베이스 벡터를 고려해 볼 수 있습니다. 아래 (그림2-17)을 봅시다.



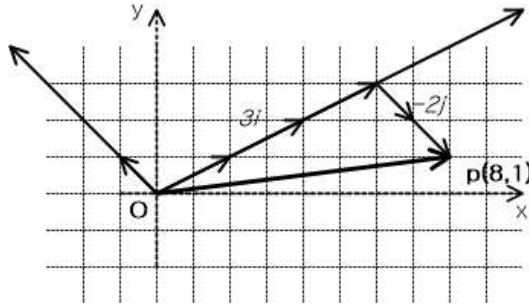
(그림2-17) 베이스스 벡터 $i(2,1)$ 와 $j(-1,1)$

위 (그림2-17)에서는 벡터 $i(2,1)$ 과 벡터 $j(1,-1)$ 을 축에 대응하는 베이스스 벡터로 사용하려고 합니다. 이 벡터들은 이제 서로 직교하지도 않고, 단위 길이를 가지지 않습니다. 이러한 $i(2,1)$, $j(1,-1)$ 가 나타내는 좌표축은 아래 (그림2-18)과 같습니다.



(그림2-18) 베이스스 벡터 $i(2,1)$ 와 벡터 $j(-1,0)$ 가 이루는 좌표계

이제 벡터 $i(2,1)$ 와 벡터 $j(1,-1)$ 가 이루는 좌표계에서 $(3,-2)$ 는 어떤 의미를 가지는 것일까요? 그것은 벡터 $i(2,1)$ 을 따라 3 단위만큼, 벡터 $j(1,-1)$ 을 따라 -2 단위만큼 움직인 위치를 나타냅니다. 그것을 식으로 나타내면 $3\vec{i} - 2\vec{j}$ 를 의미합니다.



(그림2-19) 새로운 축에서의 $(3, -2)$ 의 의미

이제 \vec{i} 와 \vec{j} 대신에 좌표축 단위를 의미하는 실제 값을 넣어서 계산해 보겠습니다.

$$3\vec{i} - 2\vec{j} = 3 \begin{bmatrix} 2 \\ 1 \end{bmatrix} - 2 \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \end{bmatrix} \quad (\text{식2-21})$$

그러면 $(8,1)$ 이라는 값을 얻는데, 이 좌표는 베이스스 $i(1,0)$ 과 $j(0,1)$ 을 사용하는 표준좌표계에서 벡터를 의미합니다.

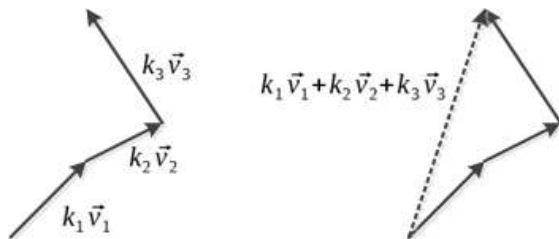
우리가 $i(1,0)$, $j(0,1)$ 을 사용하든, $i(2,1)$, $j(-1,1)$ 을 사용하든 2차원 상의 모든 벡터를 나타낼 수 있습니다. 그러면 벡터를 하나만 사용해서 2차원의 모든 벡터를 나타낼 수 있을까요? 아니면 벡터를 세 개 사용하는 것은 어떤가요? 이것을 알아보기 위해 좀 더 형식적인 수학적 정의를 이해할 필요가 있습니다.

■ 선형 조합(Linear Combination)

벡터 \vec{w} 가 다음과 같이 표현된다고 합시다.

$$\vec{w} = k_1 \vec{v}_1 + k_2 \vec{v}_2 + \dots + k_r \vec{v}_r \quad (\text{식2-22})$$

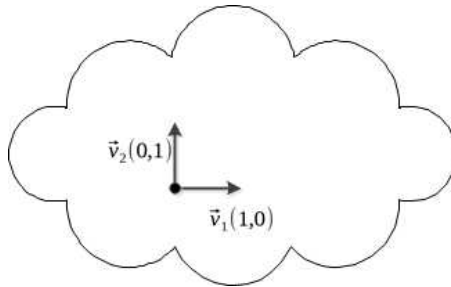
여기서, k_1, k_2, \dots, k_r 은 모두 스칼라 값입니다. 이 때, \vec{w} 를 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r$ 의 **선형 조합(linear combination)**이라고 합니다. 선형 조합의 의미는 \vec{w} 가 다른 벡터의 조합(선들의 조합)에 의해 표현될 수 있다는 의미입니다. 가능한 모든 벡터가 존재하는 공간을 **벡터 공간(vector space)**이라고 하는데, 2차원에서 벡터 공간은 평면을 의미하고, 3차원에서 벡터 공간은 3차원 공간을 의미합니다.



(그림2-20) 선형조합은 벡터가 나타내는 선(line)들의 조합(combination)을 의미합니다.

임의의 벡터 $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r$ 에 대해, $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r$ 의 선형 조합이 모든 벡터 공간상의 벡터를 표현할 수 있다면, 이 벡터들이 벡터 공간을 **스팬(span)**한다고 합니다. 2차원 평면의 예를 들면,

$(1,0)$, $(0,1)$ 벡터의 선형 조합 $k_1(1,0) + k_2(0,1)$ 은 임의의 k_n 값들을 대입함에 따라 모든 평면상의 점들을 표현할 수 있으므로, 2차원 평면 벡터 공간을 스팬합니다.

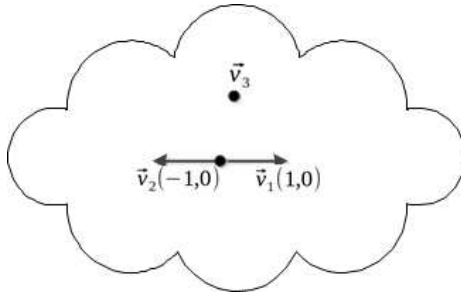


(그림2-20b) $v_1(1,0)$ 과 $v_2(0,1)$ 의 선형 조합은 2차원 평면의 모든 벡터를 스팬합니다.

우리는 2차원 평면을 스팬하는 무수히 많은 벡터들을 선택할 수 있습니다. 하지만, $k_1(1,0) + k_2(-1,0)$ 은 벡터 $(1,0)$ 과 벡터 $(-1,0)$ 이 동일한 일직선상에 위치하므로, 어떤 k_n 값을 대입하더라도 직선을 벗어난 평면상의 다른 점들을 표현할 수 없습니다. (그림2-20c)에서 $\vec{v}_1(1,0)$ 과 $\vec{v}_2(-1,0)$ 의 선형조합은 \vec{v}_3 을 나타낼 수 없습니다.

$k_1(1,0) + k_2(-1,0)$ 은 2차원 벡터 공간을 스팬하지 않습니다.

짐작하듯이 우리가 2차원 평면상에서 점의 위치를 표현하기 위해 좌표축을 고를 때에는 2차원 평면을 스팬하는 좌표축 벡터(베이스스)를 선택해야 합니다. 예를 들면 우리는 2차원 평면의 벡터를 표현하기 위해 벡터 $(1,0)$, $(0,2)$ 를 선택하는 것은 타당하지만, 벡터 $(1,0)$, $(-1,0)$ 을 선택하는 것은 타당하지 않습니다.



(그림2-20c) $\vec{v}_1(1,0)$ 과 $\vec{v}_2(-1,0)$ 의 선형 조합은 2차원 평면의 모든 벡터를 스팬하지 못합니다. $\vec{v}_1(1,0)$ 과 $\vec{v}_2(-1,0)$ 이 모두 1차원 직선 위에 존재하기 때문입니다.

$S = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r\}$ 가 벡터들의 집합 일 때, 아래의 벡터 방정식 (식2-23)을 고려해 봅시다.

$$k_1\vec{v}_1 + k_2\vec{v}_2 + \dots + k_r\vec{v}_r = \vec{0} \quad (\text{식2-23})$$

위 (식-23)은 최소한 $k_1 = 0, k_2 = 0, \dots, k_r = 0$ 의 **당연한 해 (trivial solution)**를 가집니다. 이 해의 의미는 벡터의 종류에 상관 없이 모든 k 가 0이라면 방정식이 성립한다는 의미입니다. 만약 이 해가 유일한 해라면 S 를 **선형 독립(linearly independent)** 집합이라고 합니다. 만약 다른 해가 존재한다면 **선형 종속(linearly dependent)** 집합이라고 합니다. 선형 독립의 의미는 S 에 속한 특정한 벡터가 다른 벡터들의 선형 조합에 의해 표현 불가능함을 의미합니다. 예를 들면, $S = \{ (0,1), (1,0), (0,2) \}$ 을 고려해 봅시다.

$$-2(0,1) + 0(1,0) + 1(0,2) = \vec{0} \quad (\text{식2-23b})$$

(식2-23b)는 $k_1 = 0, k_2 = 0, k_3 = 0$ 을 해로 가집니다. 하지만, 이외에도 $k_1 = -2, k_2 = 0, k_3 = 1$ 의 해를 가지므로, 선형 종속입니다.

$$(0,2) = 2(0,1) + 0(1,0) \text{ (식2-23c)}$$

(식2-23c)에서 보듯이 S 의 원소 $(0,2)$ 벡터는 다른 벡터의 선형 조합(linear combination)으로 표현이 가능합니다. 그러므로 S 는 선형종속(linearly dependent)입니다. 즉, 선형 독립인 벡터의 집합은 쓸데없는 벡터가 없어야 합니다. 2차원 평면의 예를 들면, 우리는 2차원 벡터 공간을 표현하기 위해 적절한 2개의 축만을 사용하면 됩니다. 만약 3개의 축을 사용한다면 필요 없는 여분의 축을 사용하는 결과 즉 선형 종속이 되는 것입니다.

이제 차원(dimension)과 차원상의 위치를 표현하기 위해 사용해야 하는 **축 벡터(axis vector)**를 수학적으로 정의할 준비가 되었습니다.

벡터 공간 V 에 대해 $S = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_r\}$ 이 V 의 **유한 집합(finite set)**이고 아래의 두 조건을 만족하면 S 를 **베이시스(기저, basis)**라고 합니다.

- 1) S 는 선형 독립이다.
- 2) S 는 V 를 스패н(span)한다.

베이시스의 의미는 벡터 공간을 모두 표현할 수 있는 최소한의 벡터 집합이라는 의미입니다. 우리는 어떤 r -차원의 벡터 공간에 대해서도 쉽게 한 베이시스를 다음과 같이 선택할 수 있습니다.

$$\vec{v}_1 = (1, 0, \dots, 0), \vec{v}_2 = (0, 1, 0, \dots, 0), \dots, \vec{v}_r = (0, \dots, 0, 1)$$

우리는 이러한 베이스를 **표준 베이스(standard basis)**라고 합니다. 그리고 베이스의 벡터의 개수를 **차원(dimension)**이라고 합니다.

$$S = \{(1,0,0), (0,1,0), (0,0,1)\} \text{ (식2-23d)}$$

(식2-23d)에 표시된 벡터 집합은 3차원에 대한 표준 베이스입니다. 벡터 집합의 모든 벡터 쌍이 서로 직각(orthogonal)이라면, 우리는 그 집합을 **직교 집합(orthogonal set)**이라고 합니다. 또한, 직교 집합의 모든 벡터의 길이가 1이라면 우리는 그 집합을 **정규 직교(올소노말, orthonormal)**라고 합니다. 예를 들면 아래의 벡터들은 정규직교입니다.

$$\vec{v}_1 = (0, 1, 0), \vec{v}_2 = \left(\frac{1}{2}, 0, \frac{1}{2}\right), \vec{v}_3 = \left(\frac{1}{2}, 0, -\frac{1}{2}\right)$$

우리는 어떤 차원에 대해 베이스를 선택할 때 여러 가지 이점 때문에 선형독립인 **정규직교 베이스(orthonormal basis)**를 선택합니다. **표준 베이스(standard basis)**는 항상 정규직교입니다. 그래서 우리는 2차원 평면과 3차원 공간에 대해서 표준 정규직교 베이스를 선택합니다. 3차원 공간의 경우 그것은 다음과 같습니다.

$$(1,0,0), (0,1,0), (0,0,1)$$

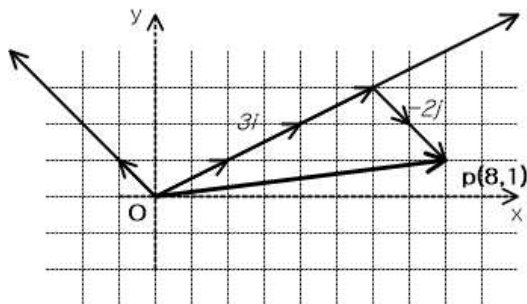
그리고 각각을 x-축, y-축, z-축 **단위 벡터(unit vector)**라 부릅니다. 우리는 x-축, y-축, z-축 단위 벡터의 선형 조합식을 아래와 같

이 구성할 수 있습니다.

$$k_1(1,0,0) + k_2(0,1,0) + k_3(0,0,1) \text{ (식2-24)}$$

(식2-24)에 의해서 3차원 공간상의 모든 점을 표현(span, 스패)할 수 있습니다. 그래서 우리는 3차원 공간의 물체의 위치를 기술하기 위해서 서로 직각인 길이가 1인 세 개의 벡터를 사용하는 것입니다.

벡터 $i(2,1)$, $j(-1,1)$ 을 베이스로 하는 좌표계상에 $(0,0)$ 을 시작점으로 하고, $(3,-2)$ 를 끝점으로 하는 벡터를 (그림2-21)에 나타내었습니다.



(그림2-21) 벡터 $i(2,1)$ 와 $j(-1,1)$ 를 베이스로 가지는 축에서의 벡터 $(3,-2)$ 의 의미

위의 그림에서 선 \overline{Op} 가 의미하는 벡터는 표준 좌표계에서 벡터 $(8,1)$ 을 의미합니다. 그것은 아래의 (식2-25)로 유도할 수 있습니다.

$$3\vec{i} - 2\vec{j} = 3\begin{bmatrix} 2 \\ 1 \end{bmatrix} - 2\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \end{bmatrix} \text{ (식2-25)}$$

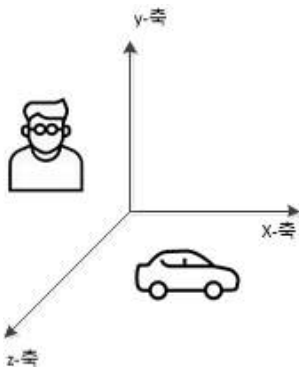
2차원에서 표준 베이스는 $i(1,0)$, $j(0,1)$ 입니다. 그러므로 (식 2-25)는 다음과 같이 표준 베이스를 사용하여 (식2-26)으로 쓸 수 있습니다.

$$8 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 1 \end{bmatrix} \quad (\text{식2-26})$$

우리는 공간을 정의하기 위해, 베이스를 정의하고, 베이스를 구성하는 벡터의 선형조합으로 공간을 정의할 수 있다는 것을 살펴보았습니다. 이렇게 정의된 공간에 어떤 인식 대상이 존재하면, 우리는 그 대상을 공간을 통해서 인식합니다. 그런데 우리는 공간 자체를 어떻게 인식할 수 있을까요?

■ 공간은 물리적 대상인가?

베이스스와 선형조합의 개념을 이해했으므로, 우리가 인지하는 3차원 공간 자체에 대해서 살펴봅시다. 우리가 자연스럽게 인식하는 공간은 3차원이므로, 3차원에 놓여있는 자동차의 위치를 기술하기 위해서는 세 개 요소를 가지는 베이스스를 사용합니다. 우리가 3차원 공간에 놓여 있는 자동차를 인식하는 과정을 생각해 봅시다.

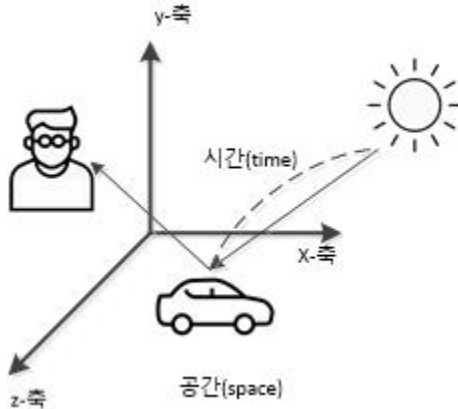


(그림2-22) 3차원 공간에 놓여 있는 자동차를 인식합니다.

우리는 공간 자체를 자동차를 인식하듯이 인식하지 않습니다. 우리는 공간을 통해서 자동차를 인식한다고 생각합니다. 하지만 본질은 그렇지 않습니다. 공간이 특별하기는 해도 그것은 물리적 대상입니다. 우리가 자동차를 인식하기 위해서는 (그림2-22)의 구성요소로는 충분하지 않습니다. 빛(light)과 시간(time)이 추가적으로 필요합니다.

자동차처럼 공간과 시간과 빛은 모두 만들어진 물리적 대상인

데, 우리 우주에서 만들어진 것은 모두 시작과 끝이 있습니다. 공간과 시간과 빛은 모두 언젠가 시작이 있었고, 미래 어느 시점에 끝이 있을 것입니다.



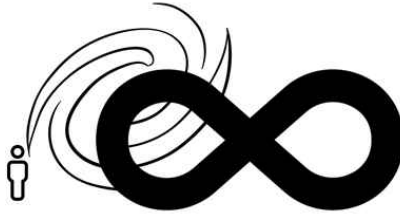
(그림2-23) 우리가 자동차를 인식하기 위해서는 공간(space), 시간(time)과 빛(light)이 필요합니다.

(그림2-23)은 사람이 자동차를 인식하는 과정을 보여줍니다. 우리는 어딘가에서 출발한 **빛 알갱이(광자, photon)**가 공간(space)을 거쳐서(time) 자동차의 표면에 닿아 반사되어서, 우리 눈에 들어온 광자의 양을 가지고 자동차를 인식합니다. **빛의 속도(speed of light)**는 일정하며 정보를 포함한 광자는 일정한 시간을 지나야 우리 눈에 도착합니다.

만약 우리 우주에 4차원 시공간(space-time)만 존재하면, 우리는 사물을 인식할 수 없습니다. **3차원 공간에 존재하는 물체를 1차원 시간을 통해 우리에게 정보를 전달해 줄 매개체인 빛(Light)이 있어야 하는 것입니다.** 만약 우주 제작방법에 대한 매뉴얼이 있다면, 그 매뉴얼의 첫 문장은 다음과 같이 시작해야 할 것입니다.

“태초에 무언가가 빛을 만들었다.” (주장2-27)

(주장2-27)의 빛은 우리가 아는 전등 빛이나 태양 빛이 아닙니다. 빛 그 자체가 되어야 합니다. 빛 그 자체가 만들어지면, 빛을 만들어 내는 태양이나 전등이 태양 빛이나 전등 빛을 만들어 낼 수 있습니다. 필자가 알기로는 (주장2-27)과 같이 시작하는 매뉴얼은 두 가지 인데, 하나는 **빅뱅이론(Big Bang Theory)**이고 하나는 **성경(Bible)**입니다. 백빙이론에서 무언가는 “**인류 원리(anthropic principle)**”이고, 성경에서 무언가는 “**하나님**”입니다.



(그림2-23b) 인류는 우리우주(our universe)에서 우리가 유일함을 관찰합니다. 그러므로 인류는 특별합니다. 하지만, 우주가 무한개라면, 인류는 우연히 존재하게 되었다는 개연성이 가능합니다.

어떤 분들은 빅뱅이론이 엉터리라는 주장을 하면서 “우연”은 복잡한 구조를 만들 수 없다고 주장합니다. 맞습니다. 하지만, 인류원리는 우연과는 다릅니다. 인류원리는 우연히 질서 있는 특별한 우주가 만들어졌다고 주장하지 않고, 특별한 우주에 살고 있는 우리가, 무한한 다른 우주가 우연임을 인식한다고 주장합니다. 우주의 개수가 무한개이므로 창조자를 가정하지 않고도 존재를 설명할 수 있게 됩니다. 필자의 주장은 둘 모두 가능성이 있다는 것입니다. 어떤 과학자가 빅뱅이론이 맞다 생각해서 창조를 믿지 않는다

면, 혹은 창조론이 맞다 생각해서 빅뱅은 무조건 틀렸다고 한다면, **정답일 가능성이 있는 한 후보를 정답의 가능성에서 제외시켜 버리는 잘못**을 범하는 것입니다.

우리 우주에서 우리가 무언가를 인식하는 과정에서 “변환”이 발생합니다. 필자는 6장 “변환”을 설명할 때, (그림2-23)에 숨어 있는 변환에 대해서 설명할 것입니다. 현대 물리학은 예전에 모순처럼 보였던 것들이 사실은 같은 것이었다는 결론을 내리는 경우가 있습니다. 예를 들면 **파동(wave)**과 **입자(particle)**가 그렇습니다. 그래서 지금은 파동과 광자를 구분하지 않고, **양자(quantum)**라고 합니다.

필자는 “로렌츠변환”을 설명할 때, 빅뱅이론과 성경의 주장이 하나의 현상에 대한 두 가지 해석이라고 주장할 것입니다. “변환”을 이해하면, 두 가지 독립적인 것처럼 보였던 것들이, 사실은 같은 하나임을 인정할 수 있는데, 꼭 이것이 물리적인 현상에 국한된 것은 아닙니다. 예를 들면 성경에서 모순처럼 보이는 “**행함**”과 “**믿음**”이 사실은 같은 대상에 붙여진 다른 이름인 것 같습니다. 그러므로 이것을 구분하려는 시도는 “본질적으로 불가능”한 것일 수 있습니다.

(식2-27) (식2-28) (식2-29) (식2-30)

(식2-31) (식2-32) (식2-33) (식2-34)

(식2-27)부터 (식2-34)는 편집상의 이유로 존재하지 않습니다.



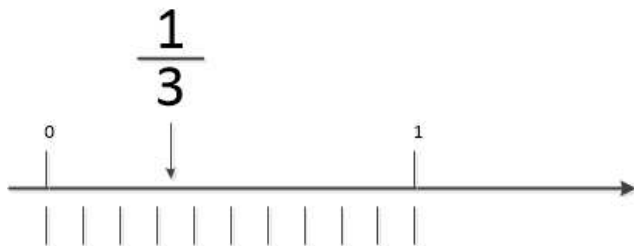
복잡한 수(complex number)

지금부터 이 절(section)에서 이야기하는 내용은 "**하나의 아름다운 수식**"을 이해하기 위한 첫 발자국입니다. "하나의 아름다운 수식"은 여섯 개의 기본함수의 의미를 모두 이해하고, 주어진 기본함수의 접하는 직선의 기울기를 구하는 과정을 이해한 후에야 "아름답다"고 느낄 수 있습니다.

이제 다시 파워함수의 성질을 분석하면서 발견한 이상한 수 $^2(-1)$ 를 자세하게 살펴보려고 합니다. 역사를 통해서 수의 개념은 발전해 왔는데, 초반에 모든 수는 수직선 위에 나타낼 수 있을 것이라고 생각했습니다. 즉 수는 1차원에만 존재한다고 생각했습니다.

먼저 우리는 수에 대한 고정관념을 버릴 필요가 있습니다. 우리는 0과 1을 포함한 **자연수(natural number)**, 음수를 포함한 **정수(integer)**, 두 정수의 비율로 표시할 수 있는 **이성적인 수(유리수, rational number, 유리(有理)는 이성이 있다는 뜻입니다)**, 실수(real number)이지만, **비이성적인 수(무리수, irrational number)**, 0보다 크고 1보다 작은 실수를 의미하는 **대시열(소수, decimal)**을 알고 있습니다.

첫 번째 버려야 할 고정관념은 모든 수를 수직선위에 표시할 수 있다고 생각하는 것입니다. 하지만, 위 수들 중 어떤 수는 수직선 위에 정확하게 나타낼 수 없습니다.



(그림2-24) $1/3$ 은 수직선 위에 정확하게 나타낼 수 없습니다.

어떤 수가 실제라는 것은 어떤 의미일까요? 1은 실제하는 수일까요? 어떤 수를 수직선 위에 나타낼 수 없다는 것은, 그 수를 실수형태로 적었을 때 정확한 값을 계산할 수 없다는 의미입니다. 현재의 어떤 컴퓨터도 $1/3$ 을 정확하게 계산할 수 없습니다. 지금 컴퓨터에서 계산기 프로그램을 사용하여 $1/3$ 을 계산해 보세요. 어떤 계산기도 $1/3$ 의 정확한 값을 출력하지 못하고 근사치만을 출력할 뿐입니다. 그러면 $1/3$ 을 단지 그것이 자연수의 비율로 표시된다고 해서, 실제하는 수라고 말할 수 있을까요? 마찬가지로 제곱하면 2가되는 수 $^22 = \sqrt[2]{2}$ 도 존재한다고 알고는 있지만, 그 정확한 값을 알지는 못합니다.

(그림2-24)처럼 수직선이 주어졌을 때, 0과 1같은 자연수, 0.5같은 실수는 수직선 위에 정확하게 나타낼 수 있지만, $1/3$ 은 수직선 위에 정확하게 나타낼 수 없습니다. 이러한 사실은 참으로 놀랍습니다. 우리 주위에는 존재는 알고 있지만, 정확하게 나타낼 수는 없는 수(number)가 존재합니다!

하지만, 우리는 사고실험(thought experiment)을 통해서 $1/3$ 을 정확하게 나타낼 수는 있습니다. $1/3 = 0.\dot{3}$ 인데 “무한에 대한 직관”을 사용하는 것입니다. 수직선의 0과 1사이의 길이는 유한하므로, 이것을 무한으로 나누는 것이 가능하다면, $1/3 = 0.\dot{3}$ 을 나타내

는 것이 가능해 집니다. 그러므로 무한에 대한 직관을 사용하면 $1/3$ 도 2_2 도 모두 실재하는 수입니다. 다만 $1/3$ 는 소수(decimal)에 규칙성이 있고, 2_2 는 소수에 규칙성이 없다는 것입니다.

우리 우주의 시공간(space-time)을 구성하는 하나의 차원이 “**한 방향으로만 진행하는**” 시간(time)이어서 우리는 시간에 제한되어 살고 있으므로, 무언가를 무한으로 반복하는 수학적 정의는 가능해도, 공학적 구현은 불가능합니다. 공학적 구현이 불가능함에도 불구하고, 무한에 대한 수학적 사고가 가능하다는 것은, 우리의 본질은 아마도 시간을 초월한 더 높은 차원에 존재하는 것 같습니다.

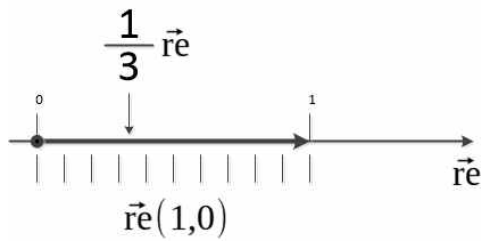
우리 우주는 아주 기묘합니다. 우리는 우주가 본질적으로 아날로그 형태일 것이라고 생각하지만, 실제 우리 우주는 컴퓨터처럼 디지털 우주입니다. 최소의 에너지 단위는 **플랑크 상수(Planck constant)**라고 알려진 엄청나게 작은 값 h 보다 작은 형태로 존재할 수 없습니다. 우주가 디지털 형태임에도 불구하고 $1/3 = 0.\dot{3}$ 과 같은 아날로그(무한대) 수를 인식하고 정의할 수 있는 이유는 무엇 일까요? 이 질문에 대한 가장 간단한 대답은 우리의 본질은 시간을 초월한, 무한대를 공학적으로 구현할 수 있는 높은 차원에 존재한다는 대답일 것입니다. 대답이 여러 개 존재할 때, 일반적으로 가장 간단한 대답이 정답입니다.

수에 대해서 두 번째 버려야 할 고정관념은 모든 수가 수직선위에 존재한다고 가정하는 것입니다. 하지만, 어떤 수들은 1차원 수직선을 벗어나 2차원 공간에 존재할 수 있습니다! 그것을 이해하기 위해 수(number)를 나타내기 위해 선형조합(linear combination)을 사용하도록 합시다.

실수(real number)를 의미하는 re 의 re 를 사용하여, 2차원 $\overrightarrow{re}(1,0)$ 벡터를 정의하면 모든 1차원 **실수공간(real number space)**

은 $\overrightarrow{re}(1,0)$ 의 선형조합으로 나타낼 수 있습니다. 그것은 (식2-35)와 같습니다.

$$k_1\overrightarrow{re} \text{ (식2-35)}$$



(그림2-25) 모든 1차원 수(실수 공간, real number space)는 $\overrightarrow{re}(1,0)$ 의 선형 조합으로 나타낼 수 있습니다.

예를 들면, 2는 $2\overrightarrow{re}$, $1/3$ 는 $(1/3)\overrightarrow{re}$ 로 나타냅니다. 이제 $^2(-1)$ 을 간단하게 표시하기 위해 (식2-36)처럼 심벌 i 를 사용하도록 하겠습니다.

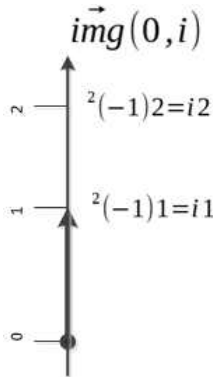
i

(그림2-25b) $^2(-1)$ 를 i 로 나타냅니다.

i 는 **상상의 수(허수, imaginary number)**를 의미합니다.

$$^2(-1) = ^2\sqrt{-1} = \sqrt{-1} = i \text{ (식2-36)}$$

그리고 $\overrightarrow{img(0,i)}$ 를 축으로 사용하는 1차원 세로축을 고려해 보면, 우리는 새로운 i 와 연관된 새로운 1차원 **허수공간(imaginary number space)**을 정의할 수 있습니다.



(그림2-26) 1차원 허수공간을 정의합니다.

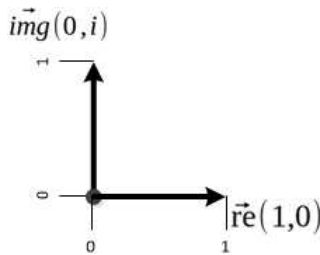
$\overrightarrow{img(0,i)}$ 를 베이스로 하는 1차원 허수공간에서 2는 $2i$ 를 의미하는데, i 가 변수처럼 보일 수 있으므로, $i2$ 라고 적기도합니다. (식2-37)처럼 $\overrightarrow{img(0,i)}$ 의 선형조합으로 모든 허수를 나타낼 수 있습니다.

$$k_2 \overrightarrow{img} \quad (\text{식2-37})$$

우리가 원래 알고 있던 1차원 실수와 허수를 모두 나타낼 수 있는 방법은 없을까요? 그것은 새로운 수를 $\overrightarrow{re}(1,0)$ 와 $\overrightarrow{img}(0,i)$ 의 선형조합으로 표현하는 것입니다. 즉 새로운 2차원 좌표계를 정의하고, 베이스로 \overrightarrow{re} 와 \overrightarrow{img} 를 사용하는 것입니다. 그러면 새로

운 수는 (식2-39)과 같은 선형조합으로 표현 가능합니다.

$$k_1 \overrightarrow{re} + k_2 \overrightarrow{img} \text{ (식2-39)}$$



(그림2-27) re 와 img 를 베이스로 하는 2차원 좌표계상의 점을 수로 간주하고 “복잡한 수(complex number)”라고 합니다.

(그림2-27)은 re 와 img 를 베이스로 하는 2차원 좌표계를 정의한 것입니다. 우리가 실수를 1차원 좌표계의 점(벡터)으로 생각했듯이, 새로운 수를 (그림2-27)이 나타내는 2차원 좌표계의 점(벡터)으로 나타내면, 기존의 실수와 새로운 허수를 모두 포함하는 수를 정의할 수 있습니다. re 와 img 를 베이스로 가지는 2차원 평면을 **복잡한 평면(complex plane, 복소평면)**이라고 하고, 복잡한 평면 위의 수를 re 와 img 의 선형조합으로 나타내고, 이 수를 “**복잡한 수(complex number, 복소수)**”라고 합니다.

(식2-39)를 벡터의 요소가 모두 나타나도록 표현해 보면 (식2-40)과 같습니다.

$$k_1 \overrightarrow{re} + k_2 \overrightarrow{img} = k_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + k_2 \begin{bmatrix} 0 \\ i \end{bmatrix} = \begin{bmatrix} k_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ k_2 i \end{bmatrix} \text{ (식2-40)}$$

(식2-40)을 보면 실수 부분의 합은 k_1 , 허수 부분의 합은 k_2 인 것을 알 수 있습니다. 그러면 이 복잡한 수를 선형조합을 이용하지 않고 (식2-41)처럼 식의 형태로 정의할 수 있습니다.

$$\begin{bmatrix} k_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ k_2 i \end{bmatrix} = k_1 + k_2 i \quad (\text{식2-41})$$

a 와 b 를 실수라고 가정하고, $^2(-1) = i$ 라고 하면, 우리는 새로운 수를 (식2-42)와 같이 정의할 수 있습니다.

$$a + bi \quad (\text{식2-42})$$

표준 용어인 복소수(複素數, complex number)의 소(素)는 '본디', '근본'을 의미합니다. 그런데 이 명칭은 **프라임 수(소수(素數), prime number, 1과 자신 외에는 약수를 가지지 않는 수)**의 한자와 같고, 1보다 작고 0보다 큰 수를 의미하는 **소수(小數, decimal)**와는 한글과 같아서 헷갈립니다. complex number의 적절한 번역 이름은 복합수(complex number)인 것 같지만, 복소수와 소수는 헷갈리지 않으므로, "복소수"라는 용어를 그대로 사용하기로 하겠습니다. 새로운 기호 i 를 사용하여 기본함수표를 채워보면 (그림 2-28)과 같습니다.

	원함수	접하는 직선	역함수	이상한 수
파워함수	x^n	nx^{n-1}	${}_n x$	i
지수함수				
투영된 선분길이 함수				

(그림2-28) 기본함수표: 파워함수 열(row)을 구성하는 네 개의 요소는 각각 x^n , nx^{n-1} , ${}_n x$, i 입니다.

필자는 고등학교에서 처음으로 허수 i 를 배웠을 때, 존재하지도 않는 수를 배워서 도대체 무슨 도움이 될까라고 생각했습니다. 하지만 i 는 3차원 공간에서의 자연스러운 회전을 나타내기 위해서, 신호분석을 위한 푸리에 변환(Fourier transform) 등 많은 곳에서 필수적으로 이용되는 실제 수(not an imaginary number)입니다. 그래서 어떤 수학자들은 허수(imaginary number)가 아니라, 2차원 공간에서 수직축에 해당하는 정보를 포함하는 수이므로, **측면수 (lateral number)**라고 이름을 붙이자고 주장하기도 합니다.

0 1 i  

(그림2-28b) 기본함수표의 파워함수를 완성하면서 우리는 추가로 기본수 i 를 찾았습니다. 이제 기본수는 0,1, i 입니다.

다음 종류의 기본함수를 알아보기 전에, 새로운 수 "복소수"를 정의했으므로, 덧셈과 곱셈 등 연산을 정의 할 수 있습니다. 복소수 $a + bi$ 는 복소평면 상의 벡터 $\vec{v}(a,b)$ 로 나타낼 수 있으므로, 복소수의 덧셈은 벡터의 덧셈과 같은 의미를 가집니다. 두 번째 복소수 $\vec{w}(c,d)$ 를 정의하면 복소수의 덧셈은 (식2-43)과 같이 정의합니다.

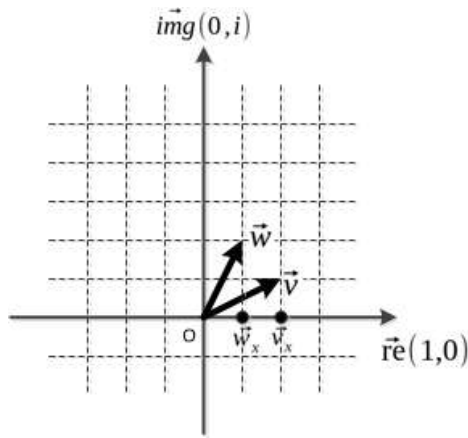
$$\vec{v} + \vec{w} = (a,b) + (c,d) = (a+c, b+d) = (a+c) + (b+d)i \quad (\text{식2-43})$$

복소수의 곱셈은 두 복소수의 곱을 **전개(expand out)**하여 결과를 얻을 수 있습니다. $\vec{v} = a + bi$ 와 $\vec{w} = c + di$ 에 대해, 두 복소수의 곱은 (식2-44)와 같이 구할 수 있습니다.

$$\begin{aligned} \vec{vw} & \quad (\text{식2-44}) \\ &= (a + bi)(c + di) \\ &= ac + adi + bci + bdi^2 \\ &= ac + adi + bci + bd(-1) \\ &= ac + adi + bci - bd \\ &= (ac - bd) + (ad + bc)i \end{aligned}$$

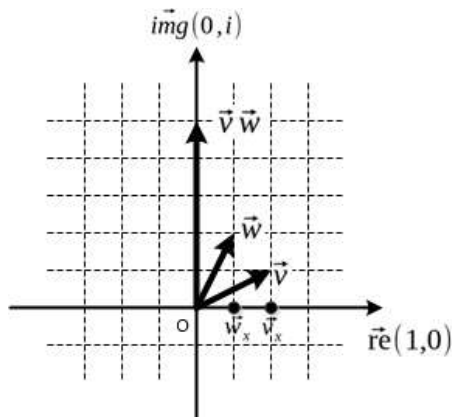
복소수의 곱셈은 우리가 이해하고자 하는 "**아름다운 수식**"의 핵심적인 부분입니다. 복소수의 곱셈의 의미를 이해하기 위해 두 복소수 $\vec{v}(2,1)$ 와 $\vec{w}(1,2)$ 를 곱해 봅시다.

$$\vec{vw} = (2 + i)(1 + 2i) = (2 - 2) + (4 + 1)i = (0, 5) \quad (\text{식2-45})$$



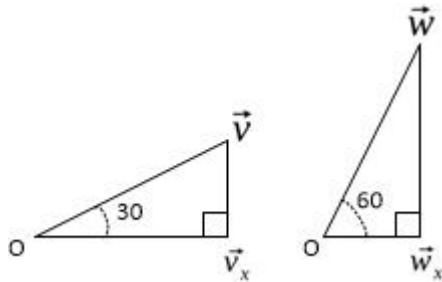
(그림2-29) 복소평면의 두 복소수 $\vec{v}(2,1)$ 와 $\vec{w}(1,2)$

(식2-45)에서 보듯이 $\vec{v}(2,1)$ 와 $\vec{w}(1,2)$ 를 곱하면 $(0,5)$ 를 얻습니다. $(0,5)$ 는 무엇을 의미하며 어떻게 해석해야 할까요? (그림2-30)에 복소수의 곱셈의 결과인 $\vec{vw} = (0,5)$ 를 표시하였습니다.



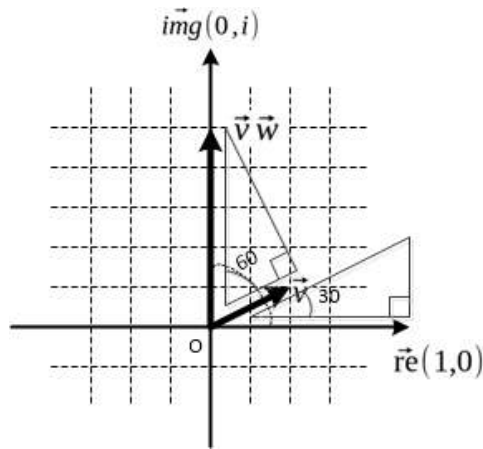
(그림2-30) $\vec{v}(2,1)$ 와 $\vec{w}(1,2)$ 의 곱셈의 결과 $\vec{vw} = (0,5)$ 입니다.

곱셈의 결과를 이해하기 위해, \vec{v} 의 끝점에서 \vec{re} 축에 투영한 점의 벡터를 \vec{v}_x 라고 하고 \vec{w} 의 끝점에서 \vec{re} 축에 투영한 점의 벡터를 \vec{w}_x 라고 합시다. 그러면 (그림2-31)과 같은 삼각형 $\triangle vOv_x$ 와 $\triangle wOw_x$ 의 내각을 생각해 볼 수 있습니다. $\triangle vOv_x$ 의 내각은 30도(degree), $\triangle wOw_x$ 의 내각은 60도입니다.



(그림2-31) 삼각형 $\triangle vOv_x$ 와 $\triangle wOw_x$ 의 내각은 각각 30, 60입니다.

\vec{v}, \vec{w} 및 곱셈의 결과인 \vec{vw} 벡터에 대해서 길이를 무시하면, 결과 벡터 \vec{vw} 는 "30도 + 60도 = 90도"가 되는 것을 알 수 있습니다.



(그림2-31) 길이를 무시하면 복소수의 곱셈의 결과는 회전을 나타냅니다. \vec{v} 와 \vec{re} 축이 이루는 각은 30입니다. \vec{vw} 는 벡터 \vec{v} 에서 \vec{w} 가 나타내는 각 60을 더한 위치의 벡터를 의미합니다.

임의의 두 복소수에 대해서도 이 성질이 성립하는데, 두 복소수의 곱셈은 회전과 관계된 것을 알 수 있습니다. 이 사실은 정말 중요합니다. 다시 한번 적어 보겠습니다.

"복소수의 곱셈은 각도(degree)의 덧셈을 나타냅니다."

복소수에 대해서 이야기할 것이 더 있습니다. 하지만 추가적인 사항은 아직 설명하지 못한 기본함수와 연관되어 있어서, 뒤로 설명을 미루도록 하겠습니다.

@