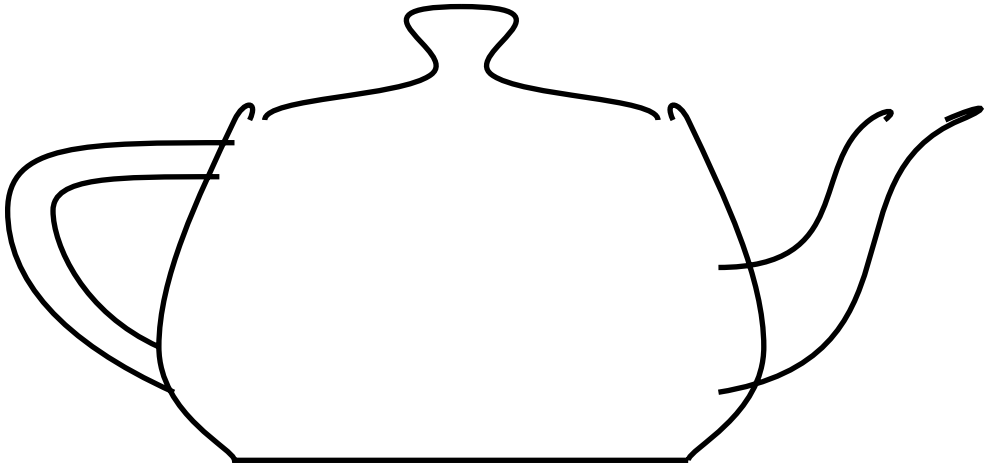
Guía del libro Mathematical Illustrations

(Bill Casselman)

(Versión preliminar 0.86)



Equipo docente

Enero 2018

Índice general

Capítulo 1

Getting started in PostScript

El lenguaje PostScript y su intérprete. PostScript es un lenguaje interpretado, esto es, se ejecuta sentencia a sentencia sin previa compilación desde un programa llamado *intérprete*. Esto hace que los errores tanto sintácticos como semánticos aparezcan en tiempo de ejecución. El interprete que utilizaremos en nuestro caso, es de libre distribución y se llama *Ghostscript*.

El lenguaje se diseñó explícitamente para la generación de gráficos y por tanto incluye toda una serie de primitivas para el dibujo plano (2D): segmento rectilíneo, círculo, cuerdas, Curvas Bézier, etc.

Sobre los ejercicios

A lo largo de las secciones del libro, Casselman va proponiendo ejercicios para afianzar el concepto que acaba de explicar. Algunos son de pura aplicación de casos particulares y otros requieren de cierto trabajo extra como buscar información en otro lado.

Los únicos ejercicios que debéis hacer son los que os he puesto en las prácticas. Están sacados de los ejercicios del libro. He intentado que hagáis solo aquellos que son más fáciles de hacer. Así que cuando leáis el libro no os detengáis en los ejercicios. Ya volveréis a ellos cuando hagáis las prácticas.

1.1. Simple Drawing

La filosofía de *PostScript* es la siguiente:

El intérprete espera las ordenes desde una línea de prompt GS>. La salida de los gráficos los genera en una pantalla gráfica blanca que representa una hoja en blanco y cuyo origen de coordenadas esta situado en la esquina inferior-izquierda.

La unidad de medida por defecto es el punto Adobe. El punto Adobe tiene una medida exacta respecto las pulgadas inglesas, inch (se expresan con ") que es esta:

1 Adobe point = $\frac{1}{72}$ inch.

En centímetros la relación es esta:

1" (inch) $\simeq$ 2,54 cm (de forma aproximada)

Las dimensiones de la hoja dependen en que formato trabaja nuestro ordenador lo característico suele ser:

- Letter (8.5" $\times$ 11" o 612 $\times$ 792 Adobe points o 21.59 cm $\times$ 27.94 cm)
- A4

Los parámetros por defecto en cuanto a dimensiones pueden ser cambiados mediante un cambio de escala y un cambio de origen de coordenadas.

Un *cambio de escala* es cambiar la dimensión de la unidad de medida. En *PostScript* se realizan mediante el comando *scale* y con dos parámetros de entrada el factor de escala para el eje X y el Y. Siempre tendremos cuidado en hacer cambios de escala homogéneos, esto es la misma escala en los dos ejes.

Ejemplos:

- Si estamos en la unidad punto Adobe y queremos trabajar en pulgadas haremos:*72 72 scale*. Esto significa que una unidad (en los dos ejes), a partir de ahora, es 72 veces más grande que la unidad anterior.
- Si estamos trabajando en cm y queremos trabajar en dm haremos *10 10 scale*

- Para evitar errores de escala se suele utilizar la instrucción *dup* que duplica el último valor de la pila: *72 dup scale*

Dibujar figuras mediante *PostScript* es como si tuviéramos un lápiz en la mano y vamos dándole ordenes de desplazamiento por la página sin levantar el lápiz.

Estos trazos que se hacen 'sin levantar el lápiz' y que constituyen por así decirlo una parte básica del dibujo es lo que se llama un camino o *path*. Para dibujar (definir) un *path* lo primero que debemos hacer es abrir el *path*(camino) **newpath**, luego posicionar el lápiz, ***x y moveto***, y a continuación realizar movimientos sencillos o complejos de dibujo(dibujar rectas, curvas, etc) si es un trazo cerrado podemos cerrarlo de forma automática con con la instrucción *closepath* y finalmente cerramos el *path* con las instrucciones ***stroke*** o ***fill***.

Nota: La instrucción *closepath* cierra el trazo mediante una línea recta desde la ultima posición a la posición de inicio. No es el final de definición de un camino (path). El final de la definición de un camino es mediante las instrucciones stroke y fill.

newpath ... ... stroke

newpath ... ... fill

1.2. Simple Cooordinate changes

Un cambio de escala conlleva siempre a que redimensionamos también el valor de la anchura del trazo de dibujo a la nueva unidad. Como la pantalla es la misma un trazo de grosor 1 cm o 1 pulgada es muy grande, por eso debemos volver al grosor inicial de 1 Adobe point. Esto nos garantiza siempre un trazo fino pero siempre visible desde nuestra nueva escala.

Por ejemplo si hemos hecho un cambio de escala a pulgadas:

72 72 scale

Devolveremos el grosor a como estaba 1 Adobe point = 1/ 72 pulgadas.

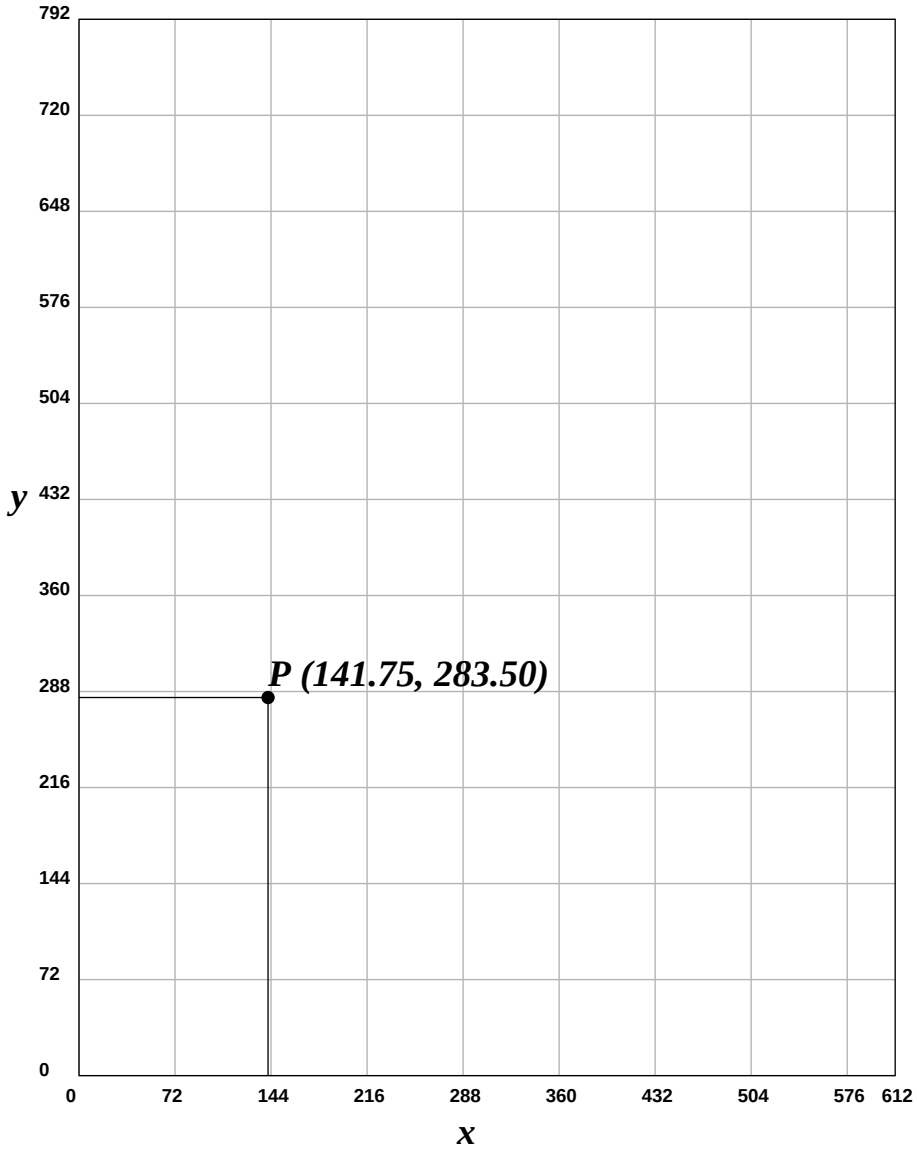
1 72 div setlinewidth

1.3. Coordinate frames

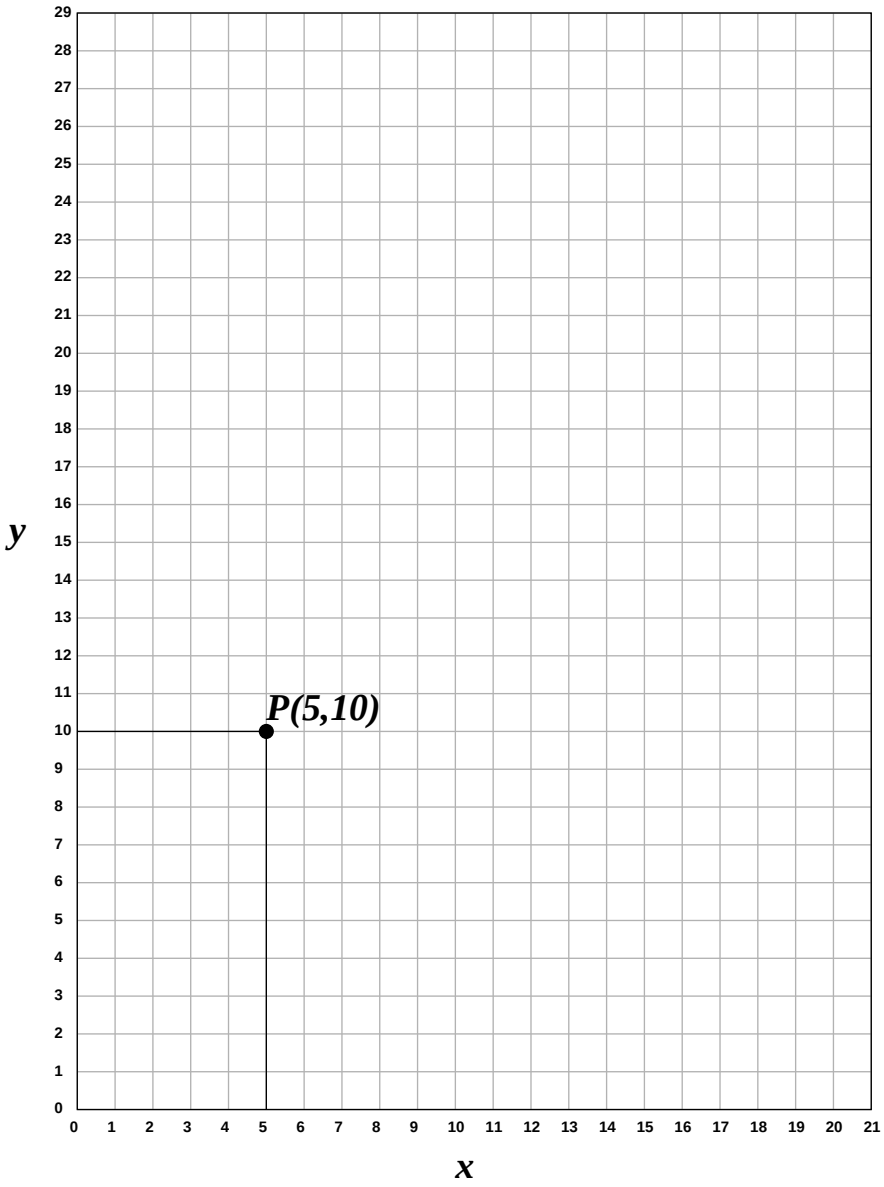
Los marcos de coordenadas o de referencia son nuestro punto de partida de medición para los puntos del plano. Debemos pensar siempre en que nuestra hoja (página) esta firme o pegada a una 'mesa' y que sobre ella ponemos nuestras reglas: ejes X, Y (casi siempre perpendiculares entre si: sistemas ortonormales) con un origen y una determinada unidad de medida.

Se cambia de sistema de coordenadas cada vez que hacemos un cambio de escala, una traslación del origen de coordenadas o una rotación de todos los ejes.

Cuando se inicializa el programa *PostScript* , crea un marco de referencia inicial para la página. Tiene el origen en la esquina izquierda inferior y la unidad de medida es el **punto Adobe**. Esta es la página inicial con su marco de referencia y unidad de medida. Hemos señalado el punto P de coordenadas (141,75 , 285,50).

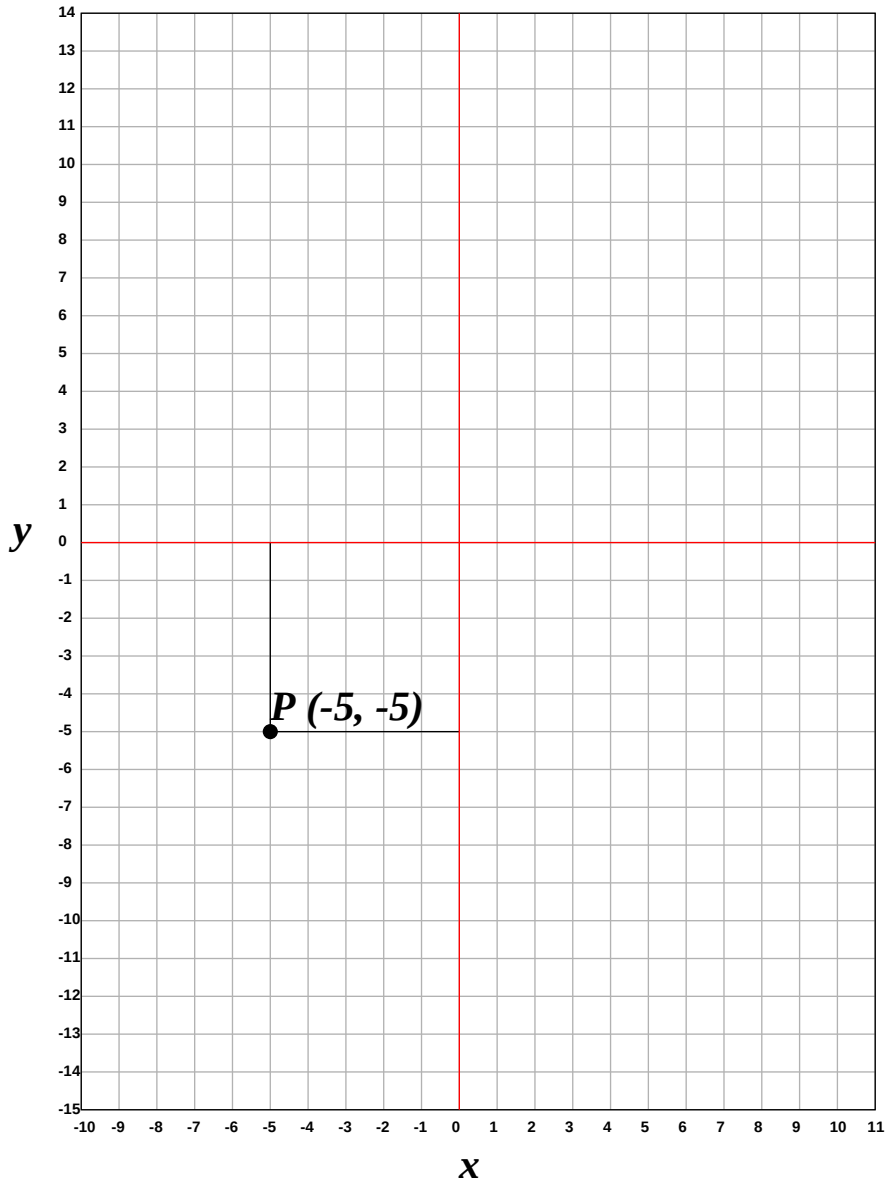


Si hacemos un cambio de escala y pasamos a cm mediante la instrucción *28.35 28.35 scale*, tenemos una página con el siguiente marco de referencia:



El punto P, que está situado en la misma posición, tiene ahora unas nuevas coordenadas, $P = (5, 10)$.

Si a continuación trasladamos el origen de coordenadas mediante la la instrucción: *10 15 translate*, tenemos un nuevo marco de referencia y las nuevas coordenadas del punto son: $P = (-5, -5)$.



Hay que tener mucho cuidado con las rotaciones y las traslaciones de ejes, es fácil perder el origen de coordenadas. Veremos consejos y formas de gestionar los cambios de coordenadas y de recuperar o volver a los marcos de coordenadas por defecto.

1.4. Doing arithmetic in PostScript

En esta sección se explica las principales funciones matemáticas de *PostScript* y como se utilizan.

1.5. Errors

Escribir en *PostScript* al principio es una fuente continua de errores.

Veremos algunos trucos para ver como podemos 'depurar' nuestro código. Por ejemplo, para ver que línea se está ejecutando, pondremos los típicos comentarios de escritura de mensajes para que los saque por la consola de salida, la instrucción: *(punto control 1) =* escribe en la consola el string: **punto control 1**.

Podemos hacer también lo que nos recomienda Casselman, que es llamar a un gestor de errores llamado *ehandler*. Para ello debemos conseguirlo y ponerlo siempre en nuestro directorio de los programas fuentes y llamarlo en la primera línea del programa: *(ehandler.ps) run*. El gestor de errores nos reduce la salida de errores enormemente.

1.6. Working with files and viewers GhostView or GSView

Trabajar con el interprete directamente no es lo más acertado ni lo natural. Lo que se hace es escribir las instrucciones de *PostScript* en un archivo y guardarlo con un nombre con extensión .ps. Este archivo, el programa fuente, se le pasa entero al intérprete. Esto se puede hacer desde los programas diseñados para ello como GSView o GhostView.

De forma alternativa también podéis utilizar el WinEdt tal como os he indicado en la primera práctica de la guía del curso.

Interesante e importante es que siempre vuestros archivos .ps empiecen con el siguiente encabezado:

%!PS-Adobe-2.0

y terminen con un *showpage*

1.7. Some fine points

Algunos consejos para mejorar la calidad de las líneas. En general lo mejor es no tocar la variable *linecap* y utilizar siempre que se pueda el *closepath*.

1.8. A trick for eliminating redundancy

Un buen truco para evitar escritura redundante. Se trata de utilizar los comandos *gsave* y *grestore* de forma inteligente.

PostScript guarda una serie de parámetros de dibujo en un conjunto de variables trasparentes para nosotros. Al conjunto de valores de estas variables en cada momento recibe el nombre de *estado gráfico (graphics state)*. Entre ellas figura el grosor de línea, la escala, etc, y también todos los caminos abiertos que están por cerrar.

Valiéndonos de ellos conseguimos dibujar y rellenar un mismo camino sin necesidad de volver a pintarlo.

La instrucción ***gsave*** guarda una copia de los valores del *estado gráfico* en ese instante. Y ***grestore*** restaura el *estado gráfico* con los valores almacenados cuando hicimos el último *gsave*. Por eso siempre se utilizan por parejas. Es como unas llaves que se abren y cierran.

Cuando estudiemos las funciones y procedimientos veremos como podemos evitar esta práctica no muy recomendable.

Volveremos a estos comandos más adelante y los utilizaremos de forma correcta.

Capítulo 2

Getting started in PostScript

Cuando se estudia Informática Gráfica, es imprescindible el conocimiento básico de la geometría de coordenadas, ahora bien, nunca debemos perder 'de vista' su significado geométrico. Por ello, es importante que repasemos los principales conceptos geométricos y las propiedades geométricas que se dan en el plano (y más adelante en el espacio).

Una vez sepamos los principios básicos de la geometría y su representación en coordenadas, tampoco hace falta ser ningún experto en el área. Tan solo tendremos a nuestro alcance un libro de geometría analítica con un buen resumen de las fórmulas analíticas de las figuras y propiedades geométricas: rectas, distancias entre puntos y rectas, intersección de rectas, circunferencias y sus intersecciones, la tangentes, etc, etc,....

La tendencia en matemáticas de presentar resultados de forma algebraica y su demostración también algebraica ha hecho mucho daño a la capacidad de observar las figuras geométricas por lo que son en verdad: dibujos geométricos.

La introducción de coordenadas en la geometría significó una gran revolución y se entendió por fin porque algunos problemas geométricos no pueden resolverse con solo las herramientas de regla (línea recta) y compás (círculo).

Las coordenadas en geometría han hecho posible que máquinas como los ordenadores puedan tratar sobre figuras geométricas mediante sus ecuaciones y encontrar sus intersecciones y al mismo tiempo visualizarlas.

2.1. Points and vectors

Los dos objetos básicos sobre lo que se sustenta toda la geometría de coordenadas son los **puntos** y los **vectores**.

Un *punto* representa una posición, y solo tiene esa cualidad. El plano está formado de puntos. Y esta es la imagen que debemos tener del punto.

Los *vectores* son la entidades que representan una magnitud con dirección. Representan la distancia relativa entre dos puntos pero además señalan un sentido de recorrido para esta magnitud. Por eso los representamos mediante una flecha.

Debemos tener claro lo siguiente: **vectores** y **puntos** tienen existencia propia independientemente que haya un sistema de coordenadas o no. El punto es como una mancha negra en una hoja (plano) y está ahí haya o no sistema de coordenadas desde el cual le podamos asignar unas coordenadas o medidas desde un origen.

Dado un sistema de coordenadas, esto es, un origen y una unidad de medida, a cada punto del plano podemos asignar dos coordenadas, por ejemplo: $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$. Y dados estos dos puntos existe un único vector que va desde el punto P al Q y cuyas coordenadas son $x_Q - x_P$ y $y_Q - y_P$, estas diferencias describen la **posición relativa** de los dos puntos.

Distinguir entre puntos y vectores cuando manejamos sus coordenadas seguiremos la siguiente notación:

- Punto de coordenada $x, y \Leftrightarrow (x, y)$
- Vectore de coordenadas $x, y \Leftrightarrow [x, y]$

Dados dos puntos, $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$ existe un único vector $\mathbf{v}\,(\overrightarrow{PQ})$ que va de P a Q. Este vector como sabemos tiene coordenadas

$$[x_Q - x_P, y_Q - y_P]$$

y por tanto podemos expresarlo como:

$$\mathbf{v} = Q - P$$

Teniendo siempre presente que no es más que un recurso o abuso de notación para describir a un vector. Ya que los puntos por si mismos no se pueden sumar ni restar, ya que esto no tiene sentido. De la expresión anterior y abusando de la notación expresamos también:

$$Q = P + \overrightarrow{PQ} = P + \mathbf{v}$$

como el punto Q que surge al desplazarnos desde el punto P hasta el final del vector **v**.

Sin embargo, si que existe un álgebra de vectores: los vectores se pueden sumar, restar y multiplicar por un número real (se suele llamar escalar para enfatizar que no se trata de un vector). La suma de vectores se realiza con la conocida ley del paralelogramo.

2.2. Areas of parallelograms

El concepto de área, no siempre fácil de definir ni darle una definición rigurosa en matemáticas, juega un papel importante en geometría.

Para nosotros nos bastará con la noción intuitiva de área como 'región' plana que se puede medir (por comparación respecto a otra área tomada como unidad). Como resultado de esta medida podemos asociar un número al área.

Recordemos que en geometría se dice que dos figuras son **congruentes o iguales** si tienen la misma forma y dimensiones exactas.

Ahora bien, en el plano existen figuras que puedan tener la misma área pero adopten distinta forma. Se dice entonces que las figuras son **equivalentes**.

Fueron los griegos quienes aplicaron el siguiente principio[1]: Si a (dos figuras de) áreas iguales (o sea equivalentes) se les añade (o quita) áreas iguales continuamos teniendo dos figuras equivalentes (de igual área).

Del principio anterior podemos obtener el siguiente argumento (Euclides):

[1]Principio: argumento o razonamiento que es de por sí muy claro y lo damos como válido

Dos regiones (figuras) tienen la misma área si pueden descomponerse en pequeñas partes que sean congruentes.

El primer resultado geométrico que nos encontraremos será ver que:

- Un paralelogramo tiene la misma área que un rectángulo de igual base y altura.

Para no alargar el curso nos quedaremos con el razonamiento de la primera demostración, que solo sirve para paralelogramos que no están muy inclinados. La inclinación es lo que se conoce como *skew*.

Podéis saltaros las demostraciones que siguen.

La transformación de un rectángulo en un paralelogramo de igual base y altura se llama: **shear**.

2.3. Lengths

El teorema de de pitágoras es la herramienta básica para el calculo de distancias entre dos puntos.

El teorema de pitágoras es de hecho un resultado geométrico que trata de áreas. y de hecho es así como lo pensaron los antiguos griegos.

2.4. Vector projections

Esta parte es importante y nos será de gran utilidad. Las proyecciones hacen posible descomponer un vector (de dirección arbitraria) en un par de vectores (de direcciones perpendiculares entre sí) y uno de ellos estará sobre una dirección 'deseada' o prefijada.

Un concepto importante en geometría (vectorial) es el de **longitud con signo**. Este concepto esta ligado a las proyecciones y en breve veremos que representa.

Casselman, en esta sección, va a deducir la fórmula analítica de la proyección perpendicular, y de paso también deduce la fórmula para el cos de

un ángulo entre dos vectores en función de sus componentes, todo ello argumentando de forma geométrica ('visual').

Lo que yo voy a presentar es lo mismo simplificando la exposición y suponiendo que ya sabemos la siguiente fórmula del coseno del ángulo α entre dos vectores, $\mathbf{u} = (x, y)$ y $\mathbf{v} = (a, b)$:

$$\cos(\alpha) = \frac{x\ a + y\ b}{\|u\|\ \|v\|}$$

A la cantidad $x\ a + y\ b$, que depende de los vectores $\mathbf{u} = (x, y)$ y $\mathbf{v} = (a, b)$ y que aparece con mucha frecuencia cuando trabajamos en coordenadas cartesianas (marco de referencia ortogonal) se le llama producto escalar de u por v: $u \cdot v = x\ a + y\ b$. La expresión anterior la podemos escribir como:

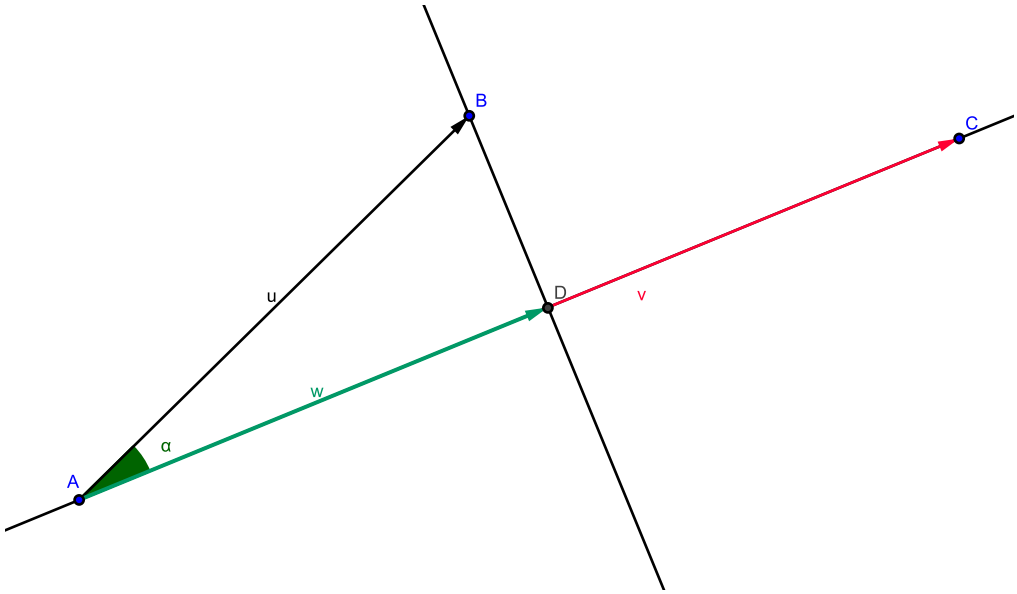
$$\cos(\alpha) = \frac{x\ a + y\ b}{\|u\|\ \|v\|} = \frac{u \cdot v}{\|u\|\ \|v\|}\,.$$

Antes que nada, recordemos que dado un vector $\mathbf{v}$, siempre podemos encontrar un vector unitario en la misma dirección, a saber:

$$\mathbf{v_u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

Y por tanto cualquier vector w que tenga la misma dirección que v se podrá escribir como $w = s\,\frac{v}{\|v\|}$, siendo s un escalar (un número, no un vector).

El problema de la proyección perpendicular es el siguiente: dado un vector $\mathbf{u}$, y una dirección marcada por otro vector $\mathbf{v}$, queremos encontrar el vector resultante de proyectar de forma perpendicular $\mathbf{u}$ sobre $\mathbf{v}$. La situación la describe bien el dibujo:



El vector **w** es la proyección (perpendicular) de **u** en **v**.

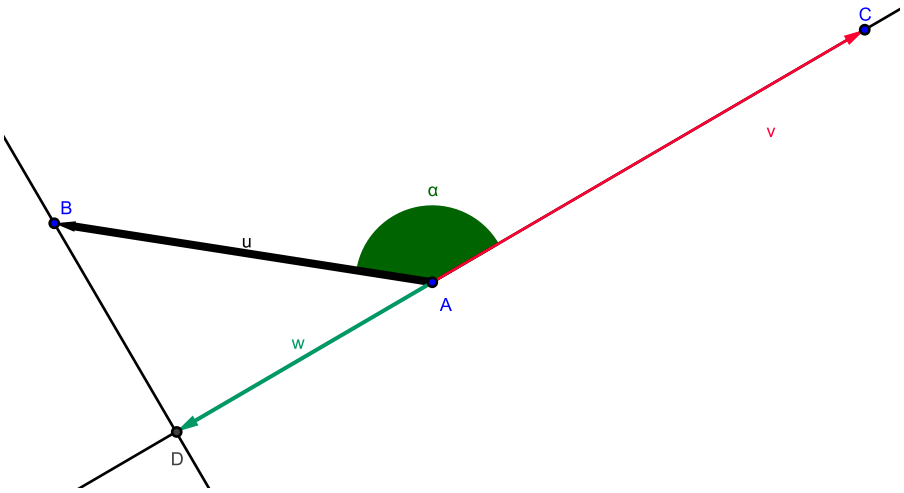
El vector **w** tiene, en el caso que representa la imagen anterior, la misma dirección que **v**. Por tanto lo podemos representar como:

$$w = s\,\frac{v}{\|v\|}$$

Se puede deducir claramente que $s = \|w\|$. Y con un poco de trigonometría:

$$s = \|u\|\cos(\alpha) = \|w\|$$

Ahora bien, si en cambio la posición del vector u respecto al vector v sobre el que queremos proyectar es esta:



Ahora también tendremos:

$$w = s\,\frac{v}{\|v\|}$$

pero en este caso el vector w tiene la dirección contraria a v, por tanto tendremos que $s = -\|w\|$:

$$w = -\|w\|\,\frac{v}{\|v\|}$$

También en este caso tenemos:

$$s = \|u\|\cos(\alpha) = \|u\|(-\cos(\pi - \alpha)) = -(\|u\|\cos(\pi - \alpha)) = -\|w\|$$

ya que $\alpha > \frac{\pi}{2}$, tenemos, $cos(\alpha) = -\cos(\pi - \alpha)$.

A la cantidad **s** se la llama **longitud con signo**. En valor absoluto equivale a la longitud de w y es **positiva** cuando la proyección coincide con la dirección v, y **negativa** cuando se proyecta en sentido contrario. Queda perfectamente determinada mediante:

$$s = \|u\|\cos(\alpha)$$

Ahora bien si ponemos $\cos(\alpha)$ en función de u y v tenemos:

$$s = \|u\|\cos(\alpha) = \|u\|\frac{u \cdot v}{\|u\|\,\|v\|} = \frac{a\ x + b\ y}{\|v\|}$$

Y ahora podemos expresar el vector w como:

$$w = s\,\frac{v}{\|v\|} = \left(\frac{x\ a + y\ b}{\|v\|}\right)\ \frac{v}{\|v\|} = \left(\frac{x\ a + y\ b}{\|v\|^2}\right)\ [a, b]$$

Con que hayáis entendido esto me basta. El que quiera revisar el razonamiento de Casselman lo puede intentar ahora sabiendo por donde van los 'tiros'.

De todo esto, lo que a nosotros nos interesa es saber la fórmula, ya que serán precisamente las fórmulas las que implementaremos en los algoritmos.

2.5. Rotations

De esta sección, lo importante a tener en cuenta es que aparecen las matrices. En Informática Gráfica los vectores se representan mediante vectores fila y se multiplican a la matrices por la derecha. Además se ajusta perfectamente a la sintaxis de *PostScript* .

2.6. The cosine rule

Esta fórmula, que generaliza el teorema de pitágoras y la hace válida para cualquier triángulo, es muy útil a la hora de resolver triángulos. No hace falta que estudies ninguna prueba.

2.7. Dot products in higher dimensions

No nos debe preocupar, solo nos moveremos en 2D y 3D.

2.8. Lines

Esta sección si que es importante. Tenemos que tener claro que una **recta** son los puntos que satisfacen la ecuación:

$$Ax + By + C = 0$$

Por tanto una recta particular r vendrá definida perfectamente cuando conozcamos los tres parámetros, **A, B y C**. De hecho estos tres parámetros los podemos asociar como las coordenadas[2] de la recta r, y se representan como $[A, B, C]$.

Sin embargo existe cierta ambigüedad, ya que cualquiera coordenada $[sA, sB, sC]$ representa a la misma recta, por tanto, estas coordenadas que determinan a una recta, y todos sus múltiplos por un escalar, se llaman **coordenadas homogéneas**.

Lo que viene a continuación es muy importante para nuestro curso, y es ver que podemos darle un significado geométrico a nuestras coordenadas homogéneas de la recta.

Es imprescindible estudiar toda esta parte hasta que esté bien entendida. Si no comprendéis algo, por favor preguntad lo en los foros.

[2]Entendidas como parámetros que sirven para determinar o caracterizar a un determinado objeto o concepto geométrico, en este caso la recta. No son coordenadas del sistema de referencia del plano.

Capítulo 3

Variables And Procedures

3.1. Varaibles in *PostScript*

En *PostScript* la definición de variables y su asignación de valores se hace todo en la misma instrucción:

```
/s 3 def
```

Esto tiene el inconveniente de que cuando asignamos un nuevo valor debemos declararla de nuevo.

Las variables son visibles desde el momento de su declaración a todo el código que viene a continuación.

3.2. Procedures in *PostScript*

3.3. Keeping Track Of Where You Are

Una buena idea para no perder el origen de coordenadas ni su orientación es utilizar las instrucciones *gsave* y *grestore* siempre en pareja.

gsave Guarda el **estado gráfico actual**, que es una serie de parámetros relacionados con la salida gráfica. Entre ellos está: origen coordenadas, grosor línea, el color, caminos abiertos, etc, ...

3.4. Passing Argumens To Procedures

Aunque PS permita la 'encapsulación' de código mediante un nombre (procedimiento), para poder llamarlo cuantas veces queramos, no es un lenguaje con notación funcional. Por ello, los argumentos de entrada se simulan mediante la definición de variables dentro del procedimiento, pero cuyos valores iniciales los colocamos en la pila justo antes de hacer la llamada al procedimiento.

Para que la instrucción de asignación funcione correctamente, hay que escribir `/s 3 def` en ese orden. Pero si ya tenemos en la pila el valor deseado, 3, para hacer la asignación correcta debemos:

- poner en la pila: `/s`
- intercambiar los dos últimos valores de la pila con: `exch`
- ahora ya tenemos en la pila `/s 3` antes de la instrucción `def`.

Ejemplo:

```
% definición procedimiento dibujo cuadrado
% argumento entrada lado del cuadrado. Ejemplo llamada:   3 cuadrat

/cuadrat {
 /s exch def
newpath
   0 0 moveto
   s 0 rlineto
   0 s rlineto
  s neg 0 rlineto
  closepath
} def

% Programa:
3 cuadrat stroke
```

Pila	Instrucciones
/s 3	exch

Pila	Instrucciones
3 /s	def

Si tenemos varios parámetros de entrada, vamos asignando variables a estos valores en orden inverso (según están en la pila) a sus correspondientes variables.

3.5. Procedures as functions

Las funciones son procedimientos que devuelven un o varios valores. En *PostScript* los valores devueltos desde cualquier función son aquellos que dejamos en la pila.

Importante recordar aquí, hacer buen uso del ´sangrado' de líneas con el fin de separar bien las instrucciones para tener código claro y de fácil lectura. Y sobre todo, comentar el código continuamente.

3.6. A final improvement

Un buen consejo: cuando tengamos un procedimiento para realizar una figura, no llamar a las instrucciones de dibujo: `stroke` (trazo), `fill` (relleno), `clip` (contorno) dentro del procedimiento, sino desde fuera, así siempre podemos utilizar una figura y decidir como visualizarla.

Capítulo 4

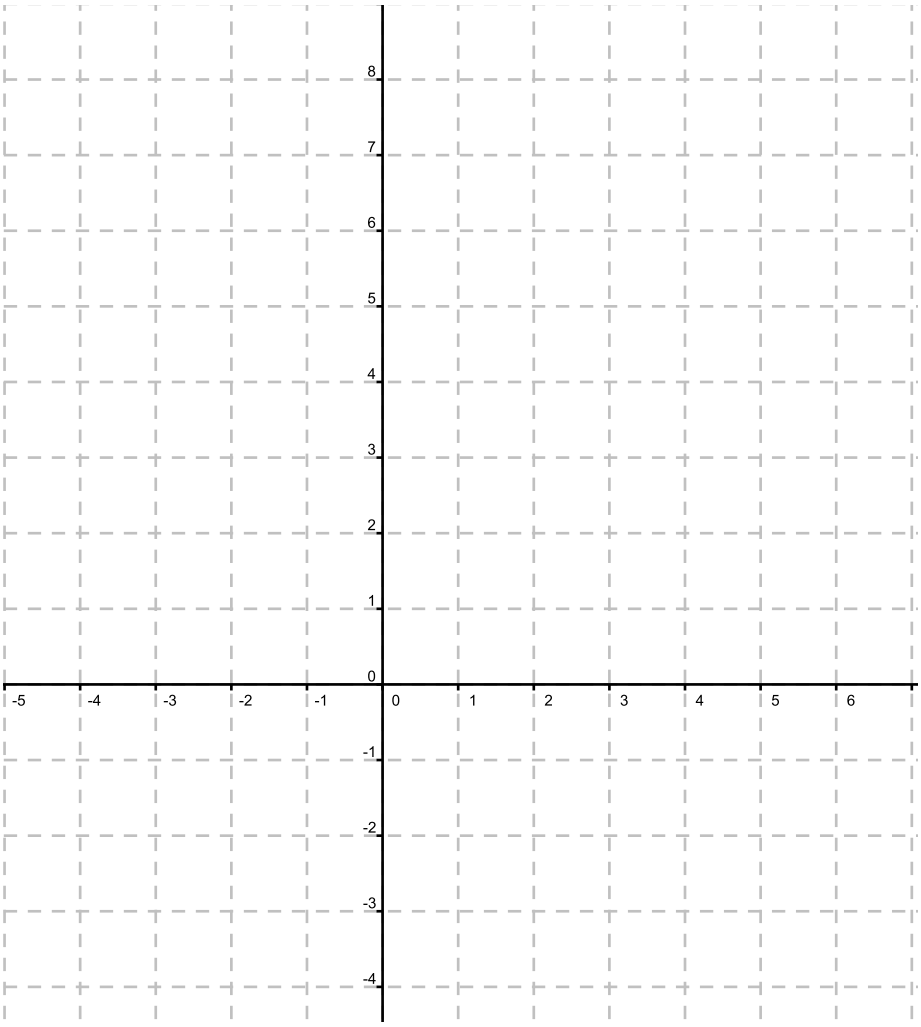
Coordinates and conditionals

Este capítulo es muy importante. En él repasaremos el concepto clave en geometría de coordenadas: el sistema de coordenadas o marco de referencia. Un sistema de coordenadas es el 'marco' o entorno que sirve para fijar un punto destacado llamado **origen** y unas líneas no paralelas, llamadas **ejes** que sirven para fijar la unidad de medida y determinar un punto en el plano (o espacio). Las **coordenadas** del punto son los puntos de corte de las paralelas a los ejes que pasan por dicho punto respecto a los propios ejes.

En general y para efectos prácticos en la gran mayoría de casos, trabajaremos con **sistemas de referencia rectangulares**, que son aquellos que tienen sus ejes perpendiculares entre si y la misma medida común en ambos ejes.

Los sistemas de referencia no rectangulares surgen en entorno específicos como es la representación en el plano de parte de una esfera. Por ejemplo cuando se estudia el trazado de mapas y planos.

Un sistema de referencia debe ser visualizado como una rejilla que cubre todo el plano.



Debemos tener presente que un **punto** representa una 'posición' en la hoja, y que es independiente del sistema de referencia. Solo en presencia de un sistema de referencia le podemos asignar unas coordenadas.

Lo mismo podemos decir de los **vectores**. Estos son objetos con identidad propia y tienen su propia ubicación en la hoja (zona visible del plano). Solo en la presencia de un sistema de coordenadas, le podemos asignar unas coordenadas. Evidentemente las dos coordenadas de un vector $\mathbf{v} = [x, y]$ por si solas no determinan un vector ya que faltaría indicar su punto de inicio. Por tanto estas dos coordenadas determinan un vector sin indicar su posición, o sea que determinan una dirección particular en el plano, y por su puesto, una magnitud concreta, su módulo (distancia relativa entre dos puntos).

Matrices

Otro concepto matemático importante que debemos recordar es el de **matriz**. Veremos que representan y como se opera con ellas.

Las matrices surgen de un campo bien concreto: los sistemas de ecuaciones (lineales). Veremos que las matrices es una 'notación' que sirven para manejar como un todo los coeficientes de estos sistemas de ecuaciones.

Ecuaciones lineales

Empecemos por algo sencillo, la ecuación lineal:

$$y = a \cdot x$$

Esta ecuación en general nos dice como varía y cuando x va cambiando. En nuestro entorno geométrico, supondremos siempre que la ecuación relaciones dos números cualesquiera (reales).

Ahora bien, si y solo si, el coeficiente **a** no es cero, podemos 'despejar' x :

$$x = \frac{1}{a} \cdot y$$

Ahora vamos a expresar esto último con el lenguaje del álgebra, que suena todo muy rimbombante:

Si el coeficiente a es distinto de cero, entonces podemos encontrar el número $\frac{1}{a}$, que hace posible despejar x.

En notación algebraica, se llama **inverso** de un número a y se representa por $\mathbf{a^{-1}}$, a cualquier número que cumpla:

$$a \cdot a^{-1} = 1$$

En el caso de lo números (reales) todo número que no sea cero, $a \neq 0$ tiene inverso, a saber $a^{-1} = \frac{1}{a}$, además es único.

Este lenguaje nos será útil a continuación para estudiar la matrices.

Sistema lineal de dos ecuaciones

Veamos ahora un caso un poco más complejo, los sistemas de 2 ecuaciones con 2 incógnitas:

$$\begin{aligned} x_2 &= a \cdot x_1 + c \cdot y_1 \\ y_2 &= b \cdot x_1 + d \cdot y_1 \end{aligned}$$

Un ejemplo donde obtenemos un sistema como este es cuando se produce un cambio (lineal) de un sistema de coordenadas del plano a otro sistema de coordenadas, un punto $P = (x_1, y_1)$ pasa a tener nuevas coordenadas $P = (x_2, y_2)$.

Podemos escribir las coordenadas x_2 e y_2 en función de las antiguas. Y obtenemos un sistema de ecuaciones como el de arriba.

Lo que determina una ecuación o un sistema de ecuaciones es el conjunto de coeficientes. Las variables x e y son meros 'espacios' vacíos donde rellenaremos de números reales.

Una forma de visualizar el sistema entero con los coeficientes como una entidad sola es agruparlos en una 'matriz' o tabla.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Podemos ahora agrupar también las incógnitas y las variables dependientes como un todo.

$$[x_2, y_2] = [x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

El expresar un sistema de ecuaciones mediante esta notación fue uno de los grandes aciertos de la matemática del siglo XIX. No solo por el aspecto visual, sino como veremos, esta notación sirve por si sola para manejar los sistemas de ecuaciones. Hemos conseguido desacoplar las incógnitas de los coeficientes. Ahora solo nos queda deducir algunas reglas para manejarlos a partir del comportamiento de la ecuaciones que representan.

A partir de aquí podemos ir nombrando los objetos obtenidos y obtener una propiedad.

Matriz 2×2 (2 filas x 2 columnas):

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Matriz 1×2 (1 fila x 2 columnas), ejemplos:

$$[x_1, y_1] \;\; [x_2, y_2]$$

Podemos ahora definir la operación siguiente como la **multiplicación** de la matriz $V(1 \times 2)$ por la matriz $M(2 \times 2)$ se obtiene una matriz $V'(1 \times 2)$ mediante el siguiente mecanismo:

$$[x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [x_1 \cdot a + y_1 \cdot c \,,\, x_1 \cdot b + y_1 \cdot d]$$

Cuando un factor es común a todos los coeficientes se suele representar como una multiplicación de un número por una matriz:

$$\begin{bmatrix} \alpha \cdot a & \alpha \cdot b \\ \alpha \cdot c & \alpha \cdot d \end{bmatrix} = \alpha \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Resolución de un sistema lineal de dos ecuaciones

Vamos ahora a resolver el sistema lineal (homogéneo) de ecuaciones.

$$\begin{aligned} x_2 &= a \cdot x_1 + c \cdot y_1 \\ y_2 &= b \cdot x_1 + d \cdot y_1 \end{aligned}$$

Si despejamos,

$$\begin{aligned} x_1 &= \frac{d \cdot x_2 - c \cdot y_2}{ad - bc} \\ y_1 &= \frac{b \cdot x_2 - a \cdot y_2}{bc - ad} \end{aligned}$$

Si llamamos $\Delta = ad - bc$, podemos reescribir el sistema como:

$$\begin{aligned} x_1 &= \frac{d}{\Delta} \cdot x_2 - \frac{c}{\Delta} \cdot y_2 \\ y_1 &= -\frac{b}{\Delta} \cdot x_2 + \frac{a}{\Delta} \cdot y_2 \end{aligned}$$

Expresemos ahora el sistema en notación matricial, como sabemos:

$$[x_1, y_1] = [x_2, y_2] \cdot \begin{bmatrix} \frac{d}{\Delta} & -\frac{b}{\Delta} \\ -\frac{c}{\Delta} & \frac{a}{\Delta} \end{bmatrix}$$

Si comparamos con el sistema original:

$$[x_2, y_2] = [x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Podemos hacer la siguiente asignación:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} \frac{d}{\Delta} & -\frac{b}{\Delta} \\ -\frac{c}{\Delta} & \frac{a}{\Delta} \end{bmatrix}$$

A la cantidad

$$\Delta = ad - bc$$

se la llama **determinante** de la matriz $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Dada una matriz $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, su determinante Δ se suele expresar como

$$\det(M) = |M| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = \Delta = ad - bc$$

Aquí vemos lo útil que resulta esta notación matricial a la hora de manejar sistema de ecuaciones. El sistema ha quedado 'compacto' y su natación hace que sea fácil su manejo como un todo.

$$[x_2, y_2] = [x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Lo representamos como

$$[x_2, y_2] = [x_1, y_1] \cdot M$$

siendo $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Si despejamos,

$$[x_1, y_1] = [x_2, y_2] \cdot M^{-1}$$

La matrices no son más que el conjunto de los coeficientes y las operaciones son las matrices, incluida el inverso de una matriz, responden a la manipulación de dichos coeficientes cuando manipulamos y resolvemos el sistema.

Esta álgebra matricial, como se ve en los libros de álgebra, está bien definida en el sentido que las operaciones son coherentes.

Composición seguidas de dos sistemas de ecuaciones

Vamos a suponer ahora que hacemos dos cambios de coordenadas seguidos. Un punto $P = (x_1, y_1)$ pasa a $P = (x_2, y_2)$ en el primer cambio de coordenadas y luego pasa a tener las siguientes coordenadas $P = (x_3, y_3)$ en el segundo cambio de coordenadas.

Primer cambio de coordenadas tenemos:

$$\begin{aligned} x_2 &= a \cdot x_1 + c \cdot y_1 \\ y_2 &= b \cdot x_1 + d \cdot y_1 \end{aligned}$$

Para el segundo cambio:

$$\begin{aligned} x_3 &= A \cdot x_2 + C \cdot y_2 \\ y_3 &= B \cdot x_2 + D \cdot y_2 \end{aligned}$$

Si expresamos los dos cambios seguidos como uno solo: expresamos las incógnitas x_3 e y_3 en función de x_1 e y_1. Después de un poco de álgebra,

sustituyendo las x_2 e y_2 del segundo sistema por sus valores del primero tenemos:

$$\begin{aligned} x_3 &= (Aa + Cb) \cdot x_1 + (Ac + Cd) \cdot y_1 \\ y_3 &= (Ba + Db) \cdot x_1 + (Bc + Dd) \cdot y_1 \end{aligned}$$

Vamos a expresar lo anterior mediante la notación matricial que hemos visto. Para el primer cambio tenemos:

$$[x_2, y_2] = [x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4.1}$$

Par el segundo podemos expresarlo:

$$[x_3, y_3] = [x_2, y_2] \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{4.2}$$

A continuación ponemos el último sistema obtenido por combinación de los dos primeros, tenemos:

$$[x_3, y_3] = [x_1, y_1] \cdot \begin{bmatrix} (Aa + Cb) & (Ba + Db) \\ (Ac + Cd) & (Bc + Dd) \end{bmatrix} \tag{4.3}$$

Fijémonos ahora que mediante la notación matricial en 4.1 y 4.2 tenemos desacopladas las incógnitas. Vamos, por tanto, a sustituir en este segunda ecuación 4.2 la matriz $[x_2, y_2]$ por su equivalente del primer sistema, a ver que pasa:

$$[x_3, y_3] = [x_1, y_1] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Si comparamos las dos últimas expresiones podemos obtener una regla para operar con las matrices de coeficientes cuando queremos expresar una composición dos cambios consecutivos de sistemas de ecuaciones: a esta operación la llamaremos **multiplicación** de dos matrices ambas de 2×2:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} aA + bC & aB + bD \\ cA + dC & cB + dD \end{bmatrix}$$

Usemos esto último para comprobar que el álgebra vectorial es consistente. ¿Que pasa cuando tenemos una matriz M y la multiplicamos por su inversa M^{-1}?

$$M \cdot M^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} \frac{d}{\Delta} & -\frac{b}{\Delta} \\ -\frac{c}{\Delta} & \frac{a}{\Delta} \end{bmatrix} = \begin{bmatrix} \frac{ad-bc}{\Delta} & \frac{-ab+ab}{\Delta} \\ \frac{cd-cd}{\Delta} & \frac{-bc+ad}{\Delta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Donde $\Delta = \det(M) = ad - bc$.

¿Existe siempre la inversa de una matriz?

Si $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, entonces sabemos que su inversa es:

$$M^{-1} = \frac{1}{\Delta} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Solo viendo esta matriz podemos afirmar que si $\Delta = \det(M) = 0$, entonces no podremos construir M^{-1}. Por tanto podemos decir que: existirá la inversa de una matriz si al menos su determinante es distinto de cero.

Cambios de sistemas de coordenadas

Los cambios de sistema de coordenadas, si son lineales, corresponden a un sistema lineal afín de coordenadas. Un sistema afín tiene una parte homogénea y una parte de traslación.

$$\begin{aligned} x_\bullet &= a \cdot x + c \cdot y + e \\ y_\bullet &= b \cdot x + d \cdot y + f \end{aligned}$$

Este sistema también se puede representar mediante notación matricial:

$$[x_\bullet, y_\bullet] = [x, y] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} + [e, f] \tag{4.4}$$

Sistemas de 3 ecuaciones y matrices de orden 3

Lo mismo podríamos hacer con los sistemas lineales de 3 ecuaciones con tres incógnitas, y deducir las matrices 3×3 y sus reglas de operación. No lo vamos ha hacer aquí, en cualquier manual o libro de matemáticas podemos encontrarlo.

4.1. Coordinates

Para que *PostScript* pueda dibujar correctamente en cualquier dispositivo trabaja internamente con tres sistemas de coordenadas:

Físico Es el que tiene asociado al dispositivo (gráfico) físico de salida. Nos da información de donde está situado el origen de coordenadas del dispositivo físico, así como de la resolución en función de punto Adobe. Será trasparente para nosotros, aunque lo utilizaremos algunas veces para calcular matrices asociadas a cambios de coordenadas.

Página Este será para nosotros el sistema de coordenadas básico que utiliza Adobe para trabajar en la página. Tiene el origen en la esquina inferior izquierda y utiliza como unidad el *punto Adobe*.

Usuario Es el sistema de coordenadas que se está utilizando en cada momento. Al inicializar *PostScript* el sistema de coordenadas de página y usuario coinciden.

4.2. How *PostScript* stores coordinates transformations

Los 6 coeficientes: a, b, c, d ,e f, que determinan un cambio de sistema de referencia (transformación afín) *PostScript* los almacena como un array de 6 elementos: $[a, b, c, d, e, f]$ con ese orden.

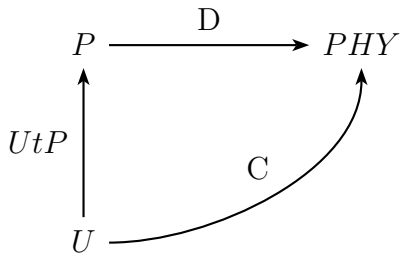
$$[x_\bullet, y_\bullet] = [x, y] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} + [e, f] \tag{4.5}$$

Lo práctico, cuando se trabaja con cambios de coordenadas, es nombrar a los dos marcos de referencia y también a la propia transformación (afín).

Sea un sistema de coordenadas inicial que llamaremos SC, donde cada punto se identifica mediante sus coordenadas (x, y). Hacemos ahora un cambio de sistema de coordenadas y pasamos al marco de referencia $SC_\bullet$, donde para diferenciarlas las coordenadas de un punto las denotaremos por $(x_\bullet, y_\bullet)$. A la trasformación afín la llamaremos, por ejemplo, T. Todo esto se expresa gráficamente mediante el diagrama:

$$SC \xrightarrow{T} SC_\bullet$$

El diagrama de la página 4 del libro muestra la relación entre los tres sistemas de referencia que utiliza *PostScript* .



4.3. Picturing the coordinate system

4.4. Moving into three dimensions

Esta sección es muy importante. Se presenta un 'truco' para manejar los sistemas afines como si fueran homogéneos. Esto es muy importante a la hora de describir los cambios de coordenadas tanto a nivel de notación como de cómputo.

El 'truco' consiste en asociar los puntos del plano (x, y) a los del espacio $(x, y, 1)$.

Del sistema afín:

$$[x_\bullet, y_\bullet] = [x, y] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} + [e, f]$$

pasamos a un sistema homogéneo:

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \cdot \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix} \tag{4.6}$$

El ejemplo de la recta es especialmente esclarecedor. Debemos tener presente que los cambios de coordenadas son solo eso, cambios en el origen y la unidad de medida. Los objetos geométricos restan donde estaban. En el nuevo sistema de referencia, tendrán desde luego, unas nuevas coordenadas[1]. Toda la elegancia y utilidad de la notación matricial se muestra en este ejemplo.

Recordar también la siguiente propiedad, que resulta muy útil si debemos implementarla para calcular inversas de este tipo de matrices:

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}^{-1} = \begin{bmatrix} A & 0 \\ v & 1 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -vA^{-1} & 1 \end{bmatrix}$$

[1]No estamos aquí hablando de movimientos de objetos. Los movimientos de objetos tendrá importancia cuando trabajemos en 3D. Matemáticamente los movimientos rígidos son también transformaciones afines

Donde, como ya sabemos:

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\Delta} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

4.4.1. La recta en coordenadas homogéneas

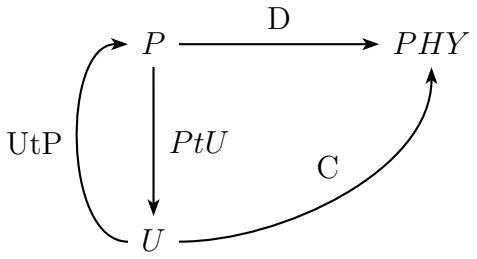
Otro de los beneficios de trabajar en coordenadas generalizadas es que se pueden manipular las rectas a través de sus coordenadas homogéneas: $[A, B, C]$.

Así, por ejemplo, la ecuación de una recta $Ax + By + C = 0$ puede escribirse como el producto escalar:

$$[x\ y\ z] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$$

Ahora vamos resolver el siguiente problema: tenemos una recta dibujada en la hoja, y queremos saber cuales serían sus coordenadas homogéneas en el sistema de Página (que es el sistema por defecto de *PostScript*). Se trata del problema de la práctica 2.

Por un lado, sabemos como se relacionan los sistemas de Usuario y de Página:



Cualquier punto se transformará mediante:

$$[x_p\ y_p\ 1] = [x_u\ y_u\ 1]UtP$$

Por tanto,

$$[x_u\ y_u\ 1] = [x_p\ y_p\ 1]UtP^{-1} = [x_p\ y_p\ 1]PtU$$

La recta en nuestro sistema de Usuario tendrá de ecuación:

$$[x_u\ y_u\ z_u] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$$

Sustituyendo $[x_u\ y_u\ z_u]$ por $[x_p\ y_p\ 1]PtU$

$$[x_p\ y_p\ 1]\ \underbrace{PtU \begin{bmatrix} A \\ B \\ C \end{bmatrix}}_{R} = 0$$

$$R = \begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix}$$

R es un vector $(3 \times 3)(3 \times 1) = 3 \times 1$. Y podemos decir que son las coordenadas homogéneas de una recta que en el sistema de Página generan exactamente los mismos puntos que la recta original.

$$[x_p\ y_p\ z_p] \begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix} = 0$$

O lo que es lo mismo, es la recta $A_p x + B_p y + C_p = 0$.

Resumiendo, para encontrar las coordenadas homogéneas de una recta que dibujara exactamente los mismos puntos que la recta original, debemos hacer:

$$\begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix} = PtU \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

¿Por qué se tiene que pasar a coordenadas de Página para resolver el problema de los puntos de corte con la página?

Porque el sistema de Página, es siempre el mismo y es con el que empieza a trabajar *PostScript* . En este *sistema de Página* las esquinas de una hoja (A4 o Letter) están perfectamente determinadas. Sin embargo, el sistema de Usuario puede haber cambiado tanto que ahora los márgenes de página no sabremos a priori sin cálculos donde estarán.

Como hacerlo en *PostScript*

Las matrices *UtP* y *PtU* no existen en *PostScript* de forma primitiva, pero podemos calcularlas como ya sabemos a partir de las matrices **C**(current) y **D**(default) (Mirar el diagrama de arriba).

$$UtP = C \cdot D^{-1}$$

$$PtU = D \cdot C^{-1}$$

Ahora podemos definir *PtU* en *PostScript* y la llamamos con la variable `T`,

```
/T
  matrix defaultmatrix  %D
  matrix currentmatrix  matrix invertmatrix   %C-1
  matrix concatmatrix   % D·C-1
 def
```

entonces La transformación $[x_u\ y_u\ 1] = [x_p\ y_p\ 1]PtU$ en *PostScript* se escribe como:

```
x_p y_p T transform
```

Pasos en *PostScript*

1. Tenemos nuestra recta $Ax+By+C = 0$ en el sistema actual de Usuario.

2. Calculamos $[A_p\ B_p\ C_p]$ a partir de

$$\begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix} = UtP \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Recordemos que las matrices de cambio de coordenadas como

$$UtP = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

en *PostScript* equivalen a un vector de longitud 6:$[a\,b\,c\,d\,e\,f]$.

$$\begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix} \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

$$\begin{bmatrix} A_p \\ B_p \\ C_p \end{bmatrix} = \begin{bmatrix} Aa + Bb \\ Ac + Bd \\ Ae + Bf + C \end{bmatrix}$$

La variable `T` será un vector de la forma $[a\,b\,c\,d\,e\,f]$. Por tanto para construir el vector de arribar en *PostScript* sería:

```
[
 A T 0 get mul B T 1 get mul add

 A T 2 get mul B T 3 get mul add

 A T 4 get mul B T 5 get mul add C add
]
```

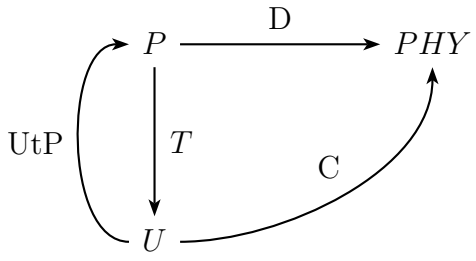
3. Calculamos puntos de corte en el *sistema Página* de la página con la recta de coeficientes $[A_p\ B_p\ C_p]$. Esto se hace con la llamada `segment-page`

4. Pasamos los puntos de corte (que estan en coord. Pagina) a coordenadas Usuario como lo hemos visto anteriormente: `x_p y_p T transform`

5. Dibujamos la recta.

4.5. How coordinate changes are made

Los tres sistema de referencia básicos de *PostScript* quedan determinados por sus matrices de transformación:



4.5.1. Cambios de sistemas de referencia

Tenemos que entender bien como se describen los cambios de sistema de referencia y así evitarnos errores y confusiones en el futuro.

El problema general es el siguiente:

De un sistema de referencia 1 $(x, y) \implies$ Sistema de referencia 2 $(x_\bullet, y_\bullet)$.

De todos los cambios posibles estamos interesados en aquellos que:

- Cambie el origen de coordenadas
- Cambie la escala en que medimos.

Estas transformaciones se llaman **AFINES**. Y básicamente son las traslaciones, rotaciones y los cambios de escala.

Importante: La transformación T que representa el cambio de coordenadas, hay que saber siempre como está actuando (en que dirección actua). Por eso cualquier cambio de coordenadas lo representamos mediante:

$$S \xrightarrow{T} S_\bullet$$

$$(x, y) \rightsquigarrow (x_\bullet, y_\bullet)$$

$$S_\bullet \xrightarrow{T^{-1}} S$$

$$(x_\bullet, y_\bullet) \rightsquigarrow (x, y)$$

Significa que la transformación actua en el sentido de la flecha, actuando en las coordenadas del sistema inicial (a la izquierda) para obtener la nuevas coordenadas:

$$(x_\bullet, y_\bullet) = T[(x, y)]$$

$$(x, y) = T^{-1}[(x_\bullet, y_\bullet)]$$

Si T es una **transformación afín**, entonces sabemos que será de la forma:

$$\begin{aligned} x_\bullet &= a \cdot x + c \cdot y + e \\ y_\bullet &= b \cdot x + d \cdot y + f \end{aligned}$$

O en forma matricial:

$$[x_\bullet, y_\bullet] = [x, y] \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} + [e, f]$$

O embebido en $3D$:

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \cdot \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

Que representa una ecuación lineal. Donde la matriz representa el coeficiente de proporcionalidad.

Podemos hacer la siguiente identificación

$$T = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \cdot T$$

Ya que el cambio afín queda totalmente determinado por esta matriz.

Resumiendo: representaremos los cambios de coordenadas mediante las siguientes notaciones:

$$(x, y) \xrightarrow{T} (x_\bullet, y_\bullet)$$
$$[x_\bullet, y_\bullet] = (x, y)M + (e, f)$$
$$[x_\bullet, y_\bullet, 1] = [x, y, 1]T$$

Si la matriz T es **no degenerada** ($\Delta \neq 0$) (como se espera de un cambio 'normal' de coordenadas)[2] entonces dicha matriz tiene inversa: T^{-1}

El cambio inverso de coordenadas lo representaremos como es natural:

$$[x, y, 1] = [x_\bullet, y_\bullet, 1]T^{-1}$$

Ejemplos:

1) Tenemos el siguiente cambio de coordenadas:

$$(X, Y) \xrightarrow{T} (X_\bullet, Y_\bullet)$$

determinado por las siguientes ecuaciones:

$$\begin{aligned} x_\bullet &= 2x + 1 \\ y_\bullet &= y - 1 \end{aligned}$$

Por tanto, podemos expresar T por medio la ecuación:

$$[x_\bullet, y_\bullet] = [x, y] \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} + [1, -1]$$

O bien,

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

[2]En contra de aquellos cambios que degeneran puntos del plano (distintos de cero) en el cero

2) Tenemos el siguiente cambio de coordenadas denotado por:

$$(X_\bullet, Y_\bullet) \xrightarrow{M} (X', Y')$$

y determinado por:

$$\begin{aligned} x' &= -y_\bullet \\ y' &= x_\bullet + 2 \end{aligned}$$

$$[x', y'] = [x_\bullet, y_\bullet] \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} + [0, 2]$$

O bien,

$$[x', y', 1] = [x_\bullet, y_\bullet, 1] \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

3)

$$SC(x, y) \xrightarrow{T} SC_\bullet(x_\bullet, y_\bullet)$$

Definido por:

$$\begin{aligned} x_\bullet &= 2x \\ y_\bullet &= 2y \end{aligned}$$

Por tanto, podemos expresar T por medio la ecuación:

$$[x_\bullet, y_\bullet] = [x, y] \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Significado geométrico: Un punto P que tenga coordenadas en $SC \equiv [1, 1]$ pasa a tener coordenadas en $SC_\bullet \equiv [2, 2]$

[imagen]

Podemos ver que:
1 *unidad* de $SC_\bullet$ equivale a $1/2$ *unidad* de SC

Hemos reducido la escala ORIGINAL a $1/2$. En *PostScript* un cambio así se da cuando escribimos:

```
0.5 0.5 scale
```

4) Tenemos un sistema de referencia $[X;Y)$ inicial. Queremos hacer un cambio de escala: aumentar por 72. Esto es, en el nuevo sistema de referencia la *unidad* es 72 veces más grande que la inicial. Esto se expresa mediante $\times 72$ o en *PostScript* : `72 72 scale` . Veamos como expresamos este cambio:

$$(X,Y) \xrightarrow{T} (X_\bullet, Y_\bullet)$$

$[x_\bullet, y_\bullet] = [x,y]T$.

$$[x_\bullet, y_\bullet] = [x,y] \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} + [0,0]$$

O bien,

$$[x_\bullet, y_\bullet, 1] = [x,y,1] \begin{bmatrix} a & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x_\bullet = a \cdot x$$
$$y_\bullet = d \cdot y$$

Un punto P de coordenadas $[72,72]$ en SC pasa a tener coordenadas $[1,1]$ en $SC_\bullet$. Sustituyendo esto en el sistema anterior, obtenemos

$$1 = a \cdot 72 \qquad a = \frac{1}{72}$$
$$1 = d \cdot 72 \qquad d = \frac{1}{72}$$

Por tanto, la transformación será:

$$\begin{aligned} x_\bullet &= \frac{1}{72} \cdot x \\ y_\bullet &= \frac{1}{72} \cdot y \end{aligned}$$

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \underbrace{\begin{bmatrix} \frac{1}{72} & 0 & 0 \\ 0 & \frac{1}{72} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{T}$$

5) De forma general, vamos a caracterizar la transformación T cuando representa un cambio de escala genérico. Por ejemplo si hacemos `A B scale` produciremos una transformación de escala no uniforme, ya que el cambio no es el mismo para los dos ejes.

$$SC[x, y] \xrightarrow{T} SC_\bullet[x_\bullet, y_\bullet]$$

$$[x_\bullet, y_\bullet, 1] = [x, y, 1]T$$

Un cambio de escala significa que cambiamos la escala original y pasamos ahora a otra unidad para medir.

$$\begin{aligned} \text{1 unidad de } X_\bullet &= \text{A unidades de } X \\ \text{1 unidad de } Y_\bullet &= \text{B unidades de } Y \end{aligned}$$

$$\begin{aligned} x &= Ax_\bullet & x_\bullet &= \frac{1}{A}x \\ y &= By_\bullet & y_\bullet &= \frac{1}{B}y \end{aligned}$$

$$[x_\bullet, y_\bullet] = [x, y] \begin{bmatrix} \frac{1}{A} & 0 \\ 0 & \frac{1}{B} \end{bmatrix}$$

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \begin{bmatrix} \frac{1}{A} & 0 & 0 \\ 0 & \frac{1}{B} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Podemos ahora expresar este mismo cambio de escala visto desde la otra dirección: pasar de las coordenadas $SC_\bullet$ finales para obtener la iniciales:

$$SC_\bullet \xrightarrow{M} SC$$

Donde $M = T^{-1}$

$$[x, y, 1] = [x_\bullet, y_\bullet, 1]T^{-1}$$

$$T^{-1} = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[x, y, 1] = [x_\bullet, y_\bullet, 1] \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

¿Cómo se calcula T^{-1} ?

a) Del enunciado del problema.

b) Del sistema:

$$\begin{aligned} x_\bullet &= ax + cy + e \\ y_\bullet &= bx + dy + f \end{aligned}$$

Despejando x e y:

$$\begin{aligned} x &= a'x_\bullet + c'y_\bullet + e' \\ y &= b'x_\bullet + d'y_\bullet + f' \end{aligned}$$

Obtenemos:

$$T^{-1} = \begin{bmatrix} a' & b' & 0 \\ c' & d' & 0 \\ e' & f' & 1 \end{bmatrix}$$

c) Calculando directamente T^{-1}

$$T^{-1} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}^{-1} = \begin{bmatrix} d/\Delta & -c/\Delta & 0 \\ -b/\Delta & a/\Delta & 0 \\ & -v \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} & 1 \end{bmatrix}$$

Donde,

$$-v \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = -[e, f] \begin{bmatrix} d/\Delta & -c\Delta \\ -b/\Delta & a/\Delta \end{bmatrix} = \left[\frac{fb - ed}{\Delta}, \frac{ce - fa}{\Delta} \right]$$

y $\Delta = \det(T) = (ad - cb)$.

6) Estamos en un sistema de referencia $SR(X, Y)$ y queremos pasar a un sistema $SR(X_\bullet, Y_\bullet)$ con el origen desplazado a $[5, 4]$ respecto del original SR.

[IMagen]

Por ejemplo un punto $P \equiv [9, 7]$ en SR tendrá unas coordenadas $[4, 3]$ en $SR_\bullet$.

Como siempre expresemos este cambio mediante la transformación:

$$(X, Y) \xrightarrow{T} (X_\bullet, Y_\bullet)$$

$$\left.\begin{array}{c} x_\bullet = x - 5 \\ y_\bullet = y - 4 \end{array}\right\} \tag{4.7}$$

Por tanto, podemos expresar T por medio la ecuación:

$$[x_\bullet, y_\bullet] = [x, y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + [-5, -4]$$

$$[x_\bullet, y_\bullet, 1] = [x, y, 1] \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -4 & 1 \end{bmatrix}}_{T}$$

Si queremos expresar la transformación inversa, podemos hacerlo, por ejemplo, desde del sistema 4.7 despejando x e y.

$$\left.\begin{array}{c} x = x_\bullet + 5 \\ y = y_\bullet + 4 \end{array}\right\} \Rightarrow$$

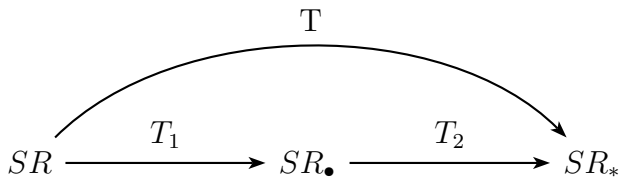
$$[x, y, 1] = [x_\bullet, y_\bullet, 1] \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 4 & 1 \end{bmatrix}}_{T^{-1}}$$

7) Supongamos ahora que estamos interesados en realizar los siguientes cambios:

   a) Cambio de escala $x72$

b) Traslación a $[2, 2]$

Podemos representar estos dos cambios mediante el siguiente diagrama:



$$[x, y] \rightsquigarrow [x_\bullet, y_\bullet] \rightsquigarrow [x_*, y_*]$$
$$[x, y] \rightsquigarrow [x_*, y_*]$$

$$\left.\begin{array}{c} [x_\bullet, y_\bullet, 1] = [x, y, 1]\, T_1 \\ [x_*, y_*, 1] = [x_\bullet, y_\bullet, 1]\, T_2 \end{array}\right\} \Rightarrow [x_*, y_*, 1] = [x, y, 1] T_1 T_2$$

$$T = T_1 \cdot T_2$$

Sabemos por los ejemplos anteriores que:

$$T_1 = \begin{bmatrix} \frac{1}{72} & 0 & 0 \\ 0 & \frac{1}{72} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

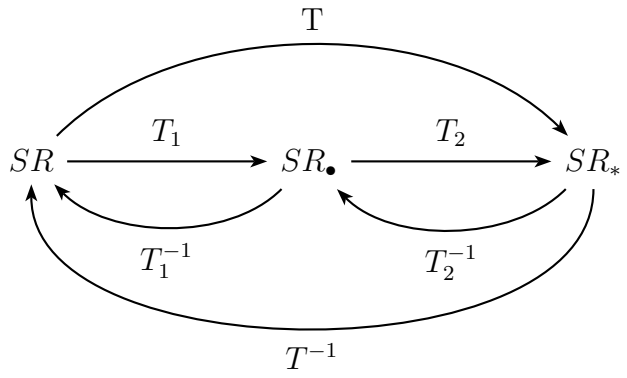
$$T_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -2 & 1 \end{bmatrix}$$

$$T = T_1 \cdot T_2 = \begin{bmatrix} \frac{1}{72} & 0 & 0 \\ 0 & \frac{1}{72} & 0 \\ -2 & -2 & 1 \end{bmatrix}$$

¿Como calculamos T^{-1}? Queremos encontrar la transformación que relaciona $[x, y]$ con $[x_*, y_*]$.

$$[x, y, 1] = [x_*, y_*, 1] T^{-1}$$

De la figura anterior:

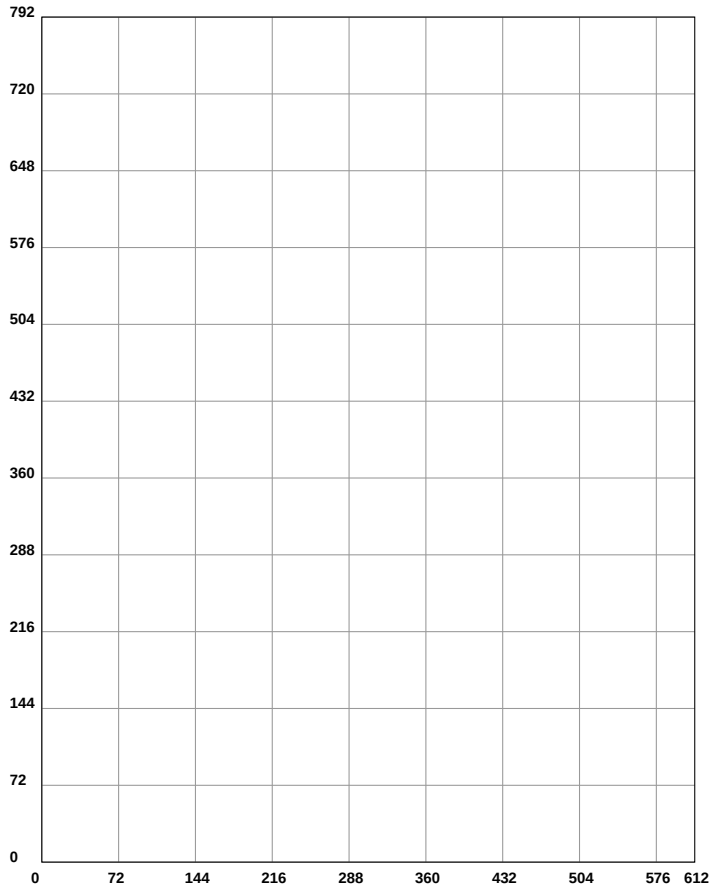


Por tanto :

$$T^{-1} = T_2^{-1} \cdot T_1^{-1}$$

Aquí vemos lo importante que es representar los cambios de coordenadas mediante un sencillo diagrama indicando claramente la dirección del nombre de la transformación.

8) Sea P el sistema inicial de coordenadas de página de *PostScript* . Como sabemos el origen se encuentra situado en la esquina inferior izquierda, y la unidad es el pto. Adobe.



Supongamos ahora que queremos trabajar en pulgadas. También sabemos que:

$$1 \text{ inch } = 72 \textit{Adobe} \text{ point}$$

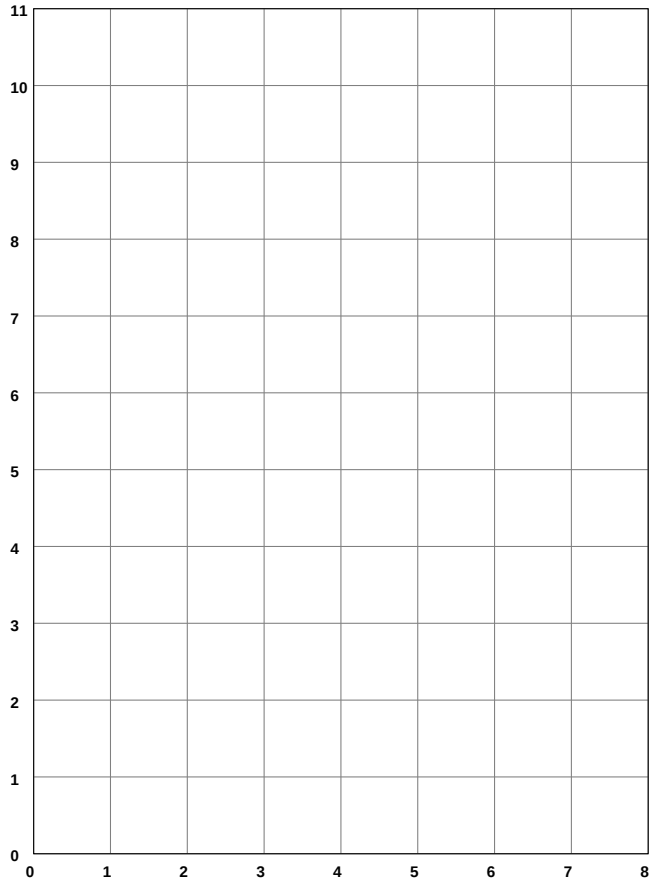
y por tanto debemos hacer un cambio de escala $x72$. Que en *PostScript* `72 dup scale`.

$$P \text{ (Page)} \xrightarrow{PtU} U \text{ (User)}$$

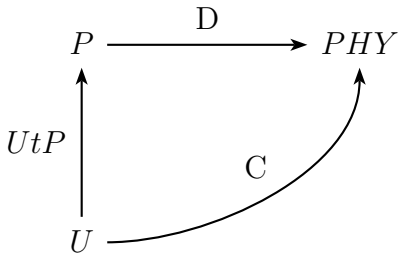
$$[x, y] \rightsquigarrow [x_u, y_u]$$

$$\begin{aligned} x_u &= \frac{1}{72} \cdot x \\ y_u &= \frac{1}{72} \cdot y \end{aligned}$$

$$[x_u, y_u, 1] = [x, y, 1]PtU$$



9) Los tres sistemas de coordenadas de *PostScript*



$$[x_{ph}, y_{ph}, 1] = [x_u, y_u, 1]\, C$$
$$[x_{ph}, y_{ph}, 1] = [x_p, y_p, 1]\, D$$
$$[x_p, y_p, 1] = [x_u, y_u, 1]\, UtP$$

C ≡ CurrentMatrix, o también denotada por CTM (Current Transform Matrix) es una matriz primitiva en *PostScript*

D ≡ DefaultMatrix es una matriz primitiva en *PostScript* y es constante. UtP ≡ User to Page Matrix, no es primitiva en *PostScript* . ¿Como se calcula?

$$UtP = C \cdot D^{-1}$$

Algunas relaciones del diagrama anterior:

a) $PHY \xrightarrow{D^{-1}} P$

b) $PHY \xrightarrow{C^{-1}} U$

c) $P \xrightarrow{UtP^{-1}} U$ Donde $UtP^{-1} = D \cdot C^{-1} = PtU$.

PostScript y las matrices cambio de coordenadas

Com ya hemos indicado anteriormente, las transformaciones afines en $2D$ son básicamente las transformaciones básicas [normales) entre sistemas de coordenadas: traslaciones, rotaciones y cambios de escala. También sabemos que estas transformaciones afines se pueden caracterizar por medio de una transformación lineal, con el recurso trabajar en el plano $Z = 1$ en 3D.

$$SR \xrightarrow{T} SR_{\bullet}$$

$$(x, y) \rightsquigarrow (x_{\bullet}, y_{\bullet})$$

$$[x_{\bullet}, y_{\bullet}, 1] = [x, y, 1]T \text{ donde } T = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

Por tanto las matrices afines quedan determinadas por solo 6 parámetros. Y de hecho *PostScript* guarda esta información mediante un array de dimensión de longitud 6: $T \equiv$ `[a b c d e f]`.

PostScript tiene una serie de funciones para el manejo de sus matrices de cambio de coordenadas. Casi todas ellas van precedidas de la palabra `matrix` que representa a la matriz unidad $I =$ `[1 0 0 1 0 0]`. Esto se hizo

por motivos de seguridad para que no "corromper" las matrices primitivas *C currentmatrix* y *D defaultmatrix*.

Ejemplos:

1. Como hacer una llamada en *PostScript* a la matriz D (Default). D representa el cambio del sistema de Página al Físico.

   ```
   matrix defaultmatrix
   ```

   Pone en la pila un array de 6 números. Esta "matriz" es constante y dependerá de la resolución en que trabaja *PostScript* . Supongamos por ejemplo que aparezca lo siguiente: `[1.66666 0 0 1.66666 0 0]`. ¿Que significa esto? Como

   $$D = \begin{bmatrix} 1{,}66666 & 0 & 0 \\ 0 & 1{,}66666 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   $$[x_\bullet, y_\bullet, 1] = [x, y, 1] \begin{bmatrix} 1{,}66666 & 0 & 0 \\ 0 & 1{,}66666 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   Podemos inferir que el origen la página está situado en la esquina inferior izquierda, y además

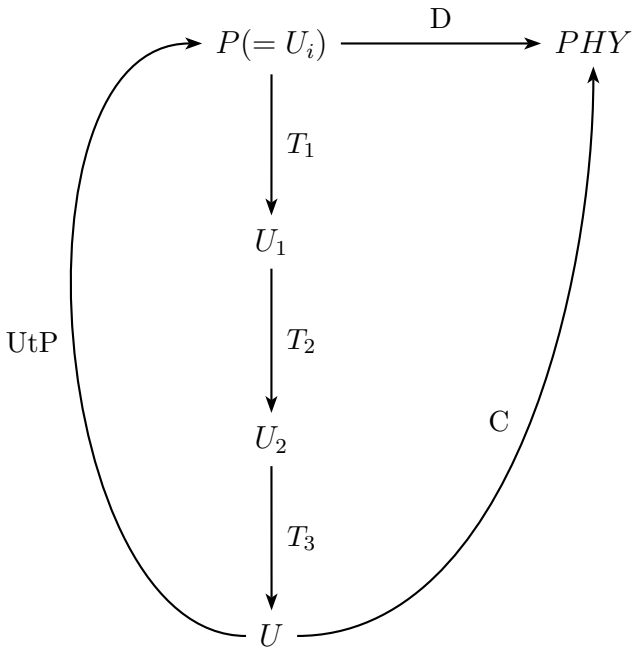
   $$1\text{unidad física (pixel)} \equiv \frac{5}{3} \text{ unidad Adobe.}$$

2. Como llamar a la matriz C (Current). Esta matriz representa el cambio de sistema de Usuario al físico.

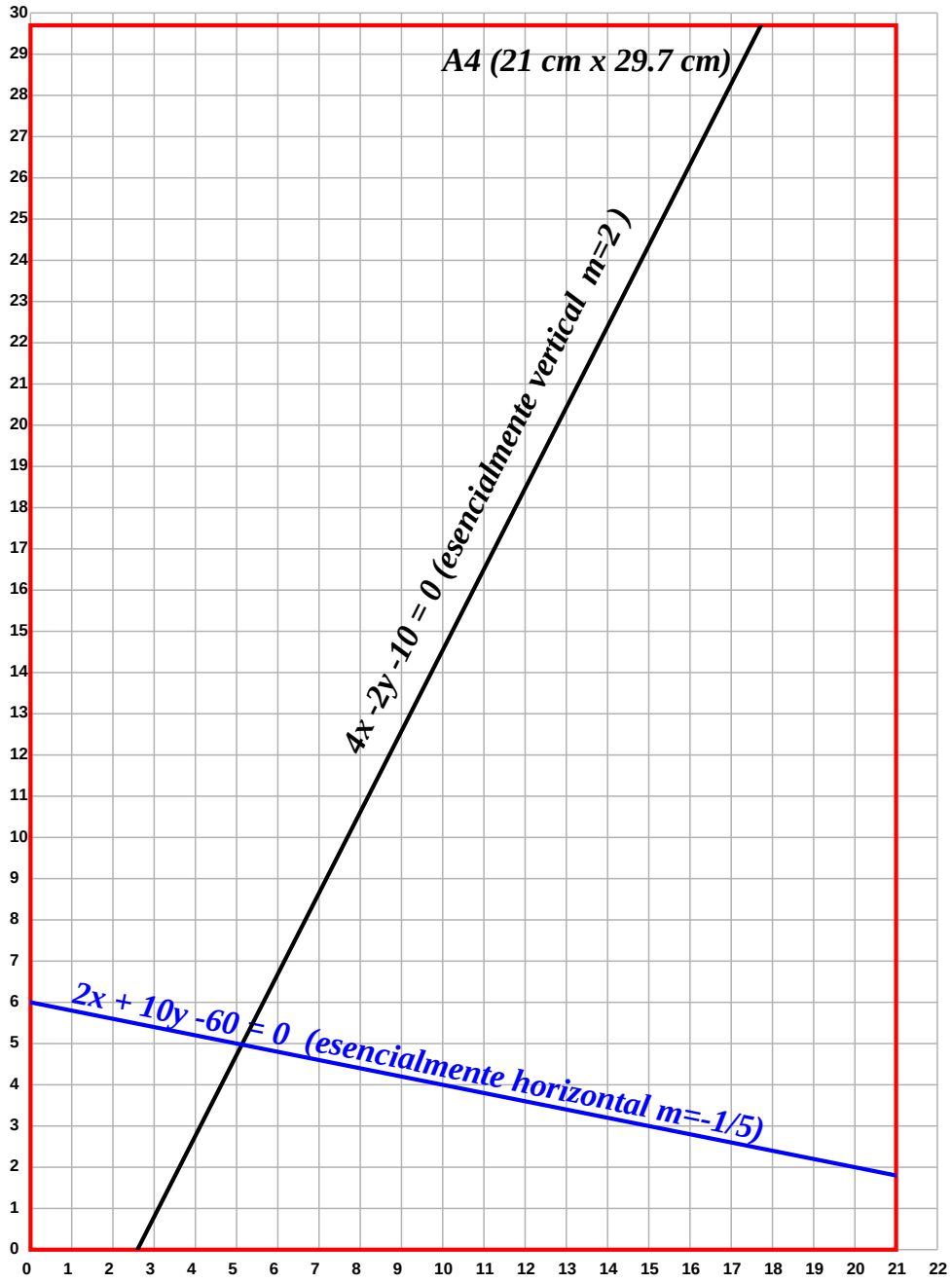
   ```
   matrix currenttmatrix
   ```

   Pone en la pila el array que representa a D en ese momento. Al **inicio** de la ejecución de *PostScript* (y mientras no cambiemos de coordenadas) el sistema de coordenadas de usuario.

3. A medida que el usuario va cambiando el sistema de coordenadas, todos estos cambios se guardan en la matriz C (Current) o también anomenada CTM (Current Transform Matrix).



4.6. Drawing infinite lines: conditionals in *PostScript*


A4 (21 cm x 29.7 cm)
4x -2y -10 = 0 (esencialmente vertical m=2)
2x + 10y -60 = 0 (esencialmente horizontal m=-1/5)

Capítulo 5

Transformations in 3D

Hasta ahora lo que hemos visto en el curso son las transformaciones (lineales) de **sistemas de coordenadas**. Esto es, hemos pasado de una forma de medir y orientación de los ejes (sistema de coordenadas o marco de referencia) a otra disposición de los ejes y de métrica (escala), otro sistema de coordenadas. Los objetos geométricos, como tales, **no** los hemos 'movido' del sitio.

Ahora, sin embargo, estamos interesados en el problema siguiente: tenemos un sistema de coordenadas, una figura (o composición de puntos) y queremos **moverla** a 'otro sitio'. Queremos cambiar las coordenadas de los puntos que forman la figura a otras distintas que determinarán su nueva posición. A este movimiento se le llama **transformación**.

Solo estamos interesados en aquellas transformaciones que permitan mover una determinada figura sin deformarla ni romperla. Estas transformaciones representan la manipulación de objetos sólidos o los movimientos en el plano de una figura rígida. Estas transformaciones se llaman **transformaciones rígidas** y se definen, como parece natural, como aquellas que conservan la distancias relativas entre puntos del objeto al que se le aplica la transformación.

En el plano estas transformaciones no las hemos estudiado por varios motivos, el principal es que en un curso introductorio no se puede ver todo en profundidad, segundo que el problema surge de forma natural al estudiar

las figuras sólidas. Las figuras sólidas deben ser vistas desde distintos ángulos para obtener la idea de su forma, además de poder 'alojarlas' y cambiarlas de sitio en un escenario. Las transformaciones en el plano se pueden ver como el caso particular de las transformaciones del espacio.

Las matemáticas involucradas serán: los sistemas de ecuaciones lineales y el álgebra vectorial en el espacio. Herramientas que ya han aparecido en el caso de los cambios de coordenadas en el plano.

5.1. Rigid transformations

Definición Una transformación se dice **rígida** si preserva las distancias relativas entre los puntos de un objeto. De esta definición se puede inferir:

- Preserva los ángulos. Esto nos acaba de confirmar que la definición es buena.
- La composición de transformaciones rígidas en rígida
- La inversa de una transformación rígida es rígida. Como no podía ser de otra forma.
- Una transformación rígida es del tipo afín. Esto significa que, si la transformación rígida la denotamos por $P \mapsto P_*$ esta se puede describir como $(x_*, y_*, z_*) = (x, y, z)A + v$ en términos de las componentes de los puntos. Como sabemos la aplicación afín tiene dos partes bien diferenciadas, la lineal, matriz A y el desplazamiento, vector v.

Esta última aserción la damos de forma gratuita sin 'demostrarla' o justificarla mucho.

También nos podríamos preguntar si todas las aplicaciones afines son transformaciones rígidas, esto es, si las aplicaciones afines conservan las distancias relativas. Desde luego la traslación del vector no altera las distancias, pero la matriz lineal puede ser, por ejemplo, de cambio de escala con lo cual ya no sería rígida la transformación. Por tanto concluimos que una aplicación afín es rígida si lo es su parte lineal.

Una traslación no ofrece ninguna dificultad en estudiarla. Por ello en el estudio de la transformaciones rígidas nos centraremos en la parte lineal de las aplicaciones afines. Repasaremos como se representan la aplicaciones lineales, como se manejan, que significa su matriz, etc.. y luego caracterizaremos a aquellas que sean rígidas.

5.2. Dot and cross products

En esta sección repasaremos las principales operaciones del álgebra vectorial y su significado geométrico.

Producto escalar (dot product) . Debemos recordar dos cosas:

$$u \cdot v = x_1x_2 + y_1y_2 + z_1z_2$$

$$\cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|}$$

Proyección paralela Es lo mismo que hemos visto para el caso 2D.

Volumen con signo Esta operación es interesante para ver como están orientadas las caras, volúmenes o vectores entre ellos. [**Optativo**] Para los que quieran saber por qué este determinante nos da el volumen con signo (o la superficie con signo en el plano) puede leer las páginas 185 y 186. Es un buen ejercicio para ver como las formulas y métodos algebraicos no surgen así sin más. Para entenderlo hay que recordar que el área de cuadrilátero de igual base entre paralelas permanece constante.

Producto vectorial (cross product) Esta operación es solo para vectores en el espacio.

Proyección perpendicular Lo mismo que ya sabíamos.

5.3. Linear transformations and matrices

Esta sección es importante. En ella repasamos los conceptos que ya sabemos de como representar aplicaciones lineales mediante matrices y veremos

el concepto clave que relaciona aplicación lineal y matriz: **el marco de referencia: frame**.

Esta sección debe ser estudiada al pie de la letra.

Un **marco de referencia (frame)** en un espacio de d dimensiones es un conjunto de d vectores linealmente independientes.

Un marco (de referencia) en un espacio de dimensión d determina un sistema de referencia, un sistema desde el cual se pueden asignar coordenadas a los puntos y vectores. Y a la inversa también se da, dado un sistema de coordenadas se puede sacar un marco de referencia (conjunto de vectores l. i.) de forma natural como veremos a continuación. En el espacio el marco formado por los vectores $\{e_1, e_2, e_3\}$, automáticamente tenemos un sistema de referencia ya que cualquier vector en el espacio se puede expresar como:

$$x = x_1 e_1 + x_2 e_2 + x_3 e_3$$

y por tanto los coeficientes (x_1, x_2, x_3) son las coordenadas de x. Las coordenadas de los vectores las representaremos mediante una matriz fila $[x_1, x_2, x_3]$.

Recíprocamente, dado un un sistema de coordenadas, los vectores

$$e_1 = [1, 0, 0], e_2 = [0, 1, 0], e_3 = [0, 0, 1]$$

forman un marco.

Aclarado esto vamos a aclarar algunos conceptos que deben quedar claro:

- Un **vector** es una entidad geométrica con significado intrínseco, esto es, por sí mismo: la posición relativa de dos puntos, o bien si tenemos una unidad de medida, representa algo que tiene dirección y magnitud. En presencia de un **marco** y solo en presencia de él se le puede asignar coordenadas.

Repetimos: los **vectores** son *esas flechas* que designan una magnitud con dirección. Se les puede asignar **coordenadas *solo con respecto a un marco de referencia***.

Definición: Un marco de referencia se dice **ortonormal** si los vectores que lo forman son de longitud la unidad y son mutuamente perpendiculares.

Por tanto como sabemos que dos vectores perpendiculares tienen su producto escalar igual a cero.

$$e_i \cdot e_j = \begin{cases} 1 \text{ si } i = 1 \\ 0 \text{ si } i \neq j \end{cases}$$

Si tenemos un vector v lo podemos expresar como la suma de sus proyecciones paralelas sobre los vectores del marco, por tanto en un marco ortonormal tendremos:

$$v = \sum c_i e_i \text{ , donde } c_i = e_i \cdot v$$

(Haced la comprobación)

Otro 'objeto' que representa una característica geométrica es la **función lineal**. La función lineal l asigna a cada vector un número con la propiedad siguiente que la hace decirse lineal:

$$l(au + bv) = al(u) + bl(v)$$

Si tenemos un vector expresado en función de los vectores de un marco, (e_1, e_2, e_3) $x = x_1e_1 + x_2e_2 + x_3e_3$, entonces $l(x)$ se puede expresar:

$$l(x) = l(x_1e_1 + x_2e_2 + x_3e_3) = x_1l(e_1) + x_2l(e_2) + x_3l(e_3) = l_1x_1 + l_2x_2 + l_3x_3$$

.

Una función lineal la representaremos mediante una matriz columna:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

Por tanto $l(x)$ lo expresamos mediante:

$$l(x) = [x_1\, x_2\, x_3\,] \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} .$$

- Una **función lineal** es una entidad geométrica con significado intrínseco, esto es, por sí misma: la asignación de un número a cualquier vector. En presencia de un **marco** y solo en presencia de él se le puede asignar coordenadas.

Definición: Una transformación lineal en una aplicación T que transforma vectores en vectores con la propiedad de ser lineal.

$$T(av + bw) = aT(v) + bT(w)$$

En presencia de un sistema de coordenadas $\{e_1, e_2, e_3\}$, podemos asociarle también coordenadas. Aunque en este caso, es una matriz de coordenadas.

Sea un vector x y una transformación lineal T. En presencia de un marco:

$$x = x_1e_2 + x_2e_2 + x_3e_3$$

$$T(x) = T(x_1e_2 + x_2e_2 + x_3e_3) = x_1T(e_1) + x_2T(e_2) + x_3T(e_3)$$

Ahora expresemos los vectores $T(e_i)$ en función de los vectores del marco, esto es: $T(e_i) = t_{i,j}e_j$. O de forma más explícita:

$$T(e_1) = t_{1,1}e_1 + t_{1,2}e_2 + t_{1,3}e_3$$

$$T(e_2) = t_{2,1}e_1 + t_{2,2}e_2 + t_{2,3}e_3$$

$$T(e_3) = t_{3,1}e_1 + t_{3,2}e_2 + t_{3,3}e_3$$

Por tanto,

$$T(x) = x_1T(e_1) + x_2T(e_2) + x_3T(e_3) = x_1(t_{1,1}e_1 + t_{1,2}e_2 + t_{1,3}e_3)+$$

$$+x_2(t_{2,1}e_1 + t_{2,2}e_2 + t_{2,3}e_3)+$$

$$+x_3(t_{3,1}e_1 + t_{3,2}e_2 + t_{3,3}e_3)$$

Agrupando,

$$T(x) = (x_1t_{1,1} + x_2t_{2,1} + x_3t_{3,1})e_1+$$

$$(x_1t_{1,2} + x_2t_{2,2} + x_3t_{3,2})e_2+$$

$$(x_1t_{1,3} + x_2t_{2,3} + x_3t_{3,3})e_3$$

De forma genérica, expresamos la coordenada j-ésima de $T(x)$ como: $(xT)_j = \sum_{i=1}^{3} x_i t_{i,j}$.

Ahora escribimos el vector $T(x)$ como un array de coordenadas:

$$T(x) = [x_1t_{1,1} + x_2t_{2,1} + x_3t_{3,1},\ x_1t_{1,2} + x_2t_{2,2} + x_3t_{3,2},\ x_1t_{1,3} + x_2t_{2,3} + x_3t_{3,3}]$$

La última expresión de $T(x)$ la podemos escribir de forma matricial:

$$T(x) = [x_1\, x_2\, x_3] \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix}$$

'Coordenadas' de T en presencia de $\{e_1, e_2, e_3\}$ es la matriz:

$$\begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix}$$

De forma correcta: matriz lineal asociada a T en presencia del sistema de coordenadas $\{e_1, e_2, e_3\}$.

Fijaros que esta matriz esta formada por filas que son las componentes del vector $T(e_i)$ en ese marco.

Otra vez, enfatizamos:

- Una **transformación lineal** es una entidad geométrica con significado intrínseco, esto es, por sí misma: la transformación de un vector a otro vector. En presencia de un **marco** (sistema de coordenadas) y solo en presencia de él se le puede asignar una matriz.

El determinante de una matriz M asociada a una transformación lineal, T, con respecto a un marco tiene un significado geométrico intrínseco: *Es el factor por el cual T escala todos los volúmenes.*

Aclaremos esto, si V es el volumen formado por tres vectores, v_1, v_2, v_3, después de aplicar T a los tres vectores el volumen que forman ahora los tres vectores, V' habrá cambiado en cierto factor respecto del anterior, $V' = kV$. Pues bien, el determinante de la matriz asociada a T es precisamente k. Lo podemos expresar como $\det(T) = k$.

¿Que pasa si $\det(T) = 1$, pues que en este caso la transformación no cambia la escala, o sea deja a todos los vectores con la misma magnitud (su módulo). De aquí ya podemos sacar una conclusión: si queremos estudiar las transformaciones rígidas, o sea que conserven la distancia relativa, serán

aquellas en las que no deberán cambiar los vectores de tamaño, y según lo que acabamos de decir, serán precisamente aquellas cuyo determinante valga 1.

El caso de $\det(T) = -1$ es un caso especial de transformación rígida: son las reflexiones. Que a nosotros no nos interesan de momento.

5.4. Changing coordinate systems

Ya sabemos que: **vectores, funciones lineales y transformaciones lineales** adquieren coordenadas en presencia de un marco. Ahora nos preguntamos ¿que pasa cuando cambiamos de marco de referencia?, ¿como cambian las coordenadas de los entes geométricos anteriores?

Lo primero que vamos a ver es como describir la relación entre dos marcos de referencia.

Esta descripción hará referencia solo a los vectores que forman los marcos de referencia. Para ello trabajaremos con matrices columna cuyos componentes son vectores y no coeficientes numéricos. Para enfatizar este hecho escribiré los vectores con una flecha arriba: $\vec{v}$.

Primero empecemos en el caso $2D$. Sean $\{\vec{e_1}, \vec{e_2}\}$ y $\{\vec{f_1}, \vec{f_2}\}$ dos marcos en $2D$. Construyamos los siguientes vectores columna:

$$e = \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \end{bmatrix} , f = \begin{bmatrix} \vec{f_1} \\ \vec{f_2} \end{bmatrix} .$$

Expresemos los vectores $\vec{f_1}$ y $\vec{f_2}$ en función del primer marco.

$$\vec{f_1} = f_{1,1}\vec{e_1} + f_{1,2}\vec{e_2}$$
$$\vec{f_2} = f_{2,1}\vec{e_1} + f_{2,2}\vec{e_2}$$

Que podemos expresarlo de forma matricial como:

$$\begin{bmatrix} \vec{f_1} \\ \vec{f_2} \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \end{bmatrix} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \end{bmatrix}$$

Que escribimos abreviadamente:

$$f = T_f^e e$$

Por tanto la matriz T_f^e relacione los dos marcos.

Nota: Esta matriz es conceptualmente distinta de la matriz de una transformación lineal. Expresa la forma en que se relacionan los dos marcos (sistemas de vectores independientes) entre sí.

Para el caso $3D$ y más general d-dimensional se obtienen las mismas fórmulas generalizando de forma natural.

$$f = T_f^e e$$

$$\begin{bmatrix} \vec{f_1} \\ \vec{f_2} \\ \vec{f_3} \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \\ \vec{e_3} \end{bmatrix}$$

5.4.1. Vectores

Vamos a expresar un vector $3D$[1] en el marco $\{\vec{e_1}, \vec{e_2}, \vec{e_3}\}$:

$$\vec{x} = x_1\vec{e_1} + x_2\vec{e_2} + x_3\vec{e_3} = x_e e = [x_1, x_2, x_3] \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \\ \vec{e_3} \end{bmatrix}$$

Podríamos igualmente expresar $\vec{x}$ en función del marco f:

$$\vec{x} = s_1\vec{f_1} + s_2\vec{f_2} + s_3\vec{f_3} = x_f f = [s_1, s_2, s_3] \begin{bmatrix} \vec{f_1} \\ \vec{f_2} \\ \vec{f_3} \end{bmatrix}$$

Como sabemos expresar $f = T_f^e e$ podemos escribir:

$$x = x_f f = x_f(T_f^e e) = (x_f T_f^e)e = x_e e$$

Expresado en forma matricial:

[1] en el caso d-dimensional es lo mismo

Sabemos que los marcos se relacionan por:

$$\begin{bmatrix} \vec{f_1} \\ \vec{f_2} \\ \vec{f_3} \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \\ \vec{e_3} \end{bmatrix}$$

Por tanto:

$$\vec{x} = s_1\vec{f_1}+s_2\vec{f_2}+s_3\vec{f_3} = x_f f = [s_1, s_2, s_3] \begin{bmatrix} \vec{f_1} \\ \vec{f_2} \\ \vec{f_3} \end{bmatrix} = [s_1, s_2, s_3] \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} \vec{e_1} \\ \vec{e_2} \\ \vec{e_3} \end{bmatrix}$$

Por tanto:

$$[x_1, x_2, x_3] = [s_1, s_2, s_3] \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix}$$

RESUMIENDO: Si e y f son dos marcos tal que $f = T_f^e e$ entonces para cualquier vector $\vec{x}$ sus coordenadas se relacionan de la siguiente forma:

$$x_e = x_f T_f^e \ .$$

5.4.2. Funciones lineales

Vemos ahora como se comportan las coordenadas de una función lineal en un cambio de marco.

Sean e y f dos marcos con $f = T_f^e e$. Sea l una función lineal. La cantidad $l(\vec{x})$ es una magnitud intrínseca: independiente del marco. por tanto expresada en los dos marcos da el mismo valor:

$$l(\vec{x}) = x_e l_e = x_f l_f$$

Como ya sabemos l_e es l en función de e:

$$l_e = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

$$l_i = l(e_i)$$

.

También sabemos como se relacionan las coordenadas de un vector $\vec{x}$ en los dos marcos: $x_e = x_f T_f^e$ (ver apartado anterior)

$$l(\vec{x}) = x_e l_e = x_f T_f^e l_e = x_f l_f$$

Por tanto:

$$l_f = T_f^e l_e$$

o si se quiere:

$$l_e = (T_f^e)^{-1} l_f$$

Por si alguien tiene problemas a la hora de visualizar lo que hemos hecho:

$$l(\vec{x}) = [x_1, x_2, x_3] \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = [s_1, s_2, s_3] \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

Los componentes del vector se relacionan como sabemos:

$$[x_1, x_2, x_3] = [s_1, s_2, s_3] \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix}$$

Así que:

$$l(\vec{x}) = [x_1, x_2, x_3] \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = [s_1, s_2, s_3] \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

Por tanto:

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$$

O bien:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix}^{-1} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

RECORDATORIO: Si e y f son dos marcos tal que $f = T_f^e e$ entonces para cualquier función lineal l,

$$l_e = (T_f^e)^{-1} l_f$$

5.4.3. Transformaciones lineales

Repasemos, una transformación lineal es una operación que transforma un vector en otro vector. Siempre consideramos los vectores situados en el origen. Por tanto cualquier transformación de un vector siempre dará otro vector situado también en el origen, ya que para toda transformación lineal, el origen se convierte en origen: $T(0) = 0$.

Sea como siempre e y f dos marcos de referencia tal que: $f = T_f^e e$.

La tansformación de un vector $\vec{x}$ mediante la transformación lineal A en el marco $e = \{e_1, e_2, e_3\}$ se puede expresar como ya sabemos:

$$(\vec{x}A)_e = x_e A_e = [x_1, x_2, x_3] \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix}$$

De forma similar:

$$(\vec{x}A)_f = x_f A_f = [y_1, y_2, y_3] \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix}$$

Por tanto, y sabiendo que para cualquier vector: $x_e = x_f T_f^e$:

$$\begin{aligned} x_e A_e &= (xA)_e \\ &= (xA)_f T_f^e \\ &= (x_f A_f) T_f^e \end{aligned}$$

$$= x_f(A_f T_f^e)$$

$$= (x_e(T_f^e)^{-1})(A_f T_f^e)$$

$$x_e((T_f^e)^{-1} A_f T_f^e)$$

Por tanto:

$$A_e = (T_f^e)^{-1} A_f T_f^e$$

5.5. Rigid linear transformation

En esta sección se estudia que aspectos tiene la matriz de una transformación rígida en cualquier marco de referencia.

Antes que nada repasaremos algunos hechos básicos sobre los marcos de referencia.

Sea $e = \{\vec{e_1}, \vec{e_2}, \vec{e_3}\}$ un marco de referencia en el espacio (3D), esto es, tres vectores linealmente independientes.

- Las coordenadas de estos vectores en su propio marco de referencia serán:

$$e_1 = (1, 0, 0)$$

$$e_2 = (0, 1, 0)$$

$$e_3 = (0, 0, 1)$$

- Su **módulo** será por tanto: $e_i \cdot e_i = |e_i|^2 = 1$, esto es $|e_i| = 1$.

- ¡¡ Cuidado !! Estas unidades son medidas respecto a los vectores de e. Además $\vec{e_i} \cdot \vec{e_j} = 0$ solo significa que los vectores están sobre las direcciones principales, y no que su ángulo sea 90^0.

Cuando es importante la unidad métrica y trabajar en espacios de la geometría del plano y espacio, debemos tener 3 ejes perpendiculares y la misma unidad en todos los ejes. Por ello lo **natural** es escoger el marco de

referencia (los tres vectores independientes) a partir del sistema de medida y definir el siguiente marco como natural:

$$e_1 = [1, 0, 0],\ e_2 = [0, 1, 0],\ e_3 = [0, 0, 1]$$

Esto es, definimos tres vectores unitarios en las 3 direcciones perpendiculares del espacio. De esta forma todas las unidades son homogéneas y las direcciones perpendiculares entre sí formando 90^o. Este marco de referencia importante se le llama **marco canónico de referencia**.

La condición de perpendicularidad: $e_i \cdot e_j = 0$ cobra ahora el sentido natural de ser 90^o. Además se dice que un marco cuyos vectores sean perpendiculares entre sí se llama **ortogonal**. Si además los vectores son todos unitarios recibe el nombre (rimbombante) de **ortonormal**.

Para nosotros siempre partiremos del marco canónico e, pues es la forma natural de trabajar. A partir de aquí podremos cambiar de marco: $f = T_f^e e$ y referir los vectores, funciones lineales o transformaciones lineales al nuevo marco. Cosa que ya sabemos.

Ahora nos interesa caracterizar aquellas transformaciones lineales que hemos llamado **rígidas** que tienen la propiedad que **conservan las distancias relativas**, esto se traduce que los vectores (distancia relativa) transformados por esta transformación lineal tendrán la misma longitud que los vectores originales.

Con todo esto ya podéis leer la sección 12.5 y llegar a las siguientes conclusiones:

- Una transformación lineal es rígida precisamente cuando las filas de su matriz de coordenadas forman un marco ortogonal
- Una matriz es ortogonal si y solo si $AA^t = I$
- El determinante de una matriz ortogonal es ± 1.

5.6. Orthogonal transformations in 2D

Solo debéis llegar al punto donde dice:

- Toda transformación lineal rígida en $2D$ es: o una rotación o una reflexión.

5.7. Orthogonal transformations in 3D

En el espacio caracterizar las transformaciones lineales rígidas es algo más complejo, pero se puede ver que todas se pueden describir como una **rotación axial** siempre que preserven la orientación.

Con esto y el dibujo de la primera parte que estudiéis es suficiente.

De las dos secciones que vienen a continuación, como no vamos a implementar nada ni hacer ninguna practica con ellos no hace falta que lo estudiéis.

5.8. Calculating the effect of an axial rotation

(No hace falta estudiarla) Aquí vemos como podemos aplicar una rotación axial a un vector mediante un procedimiento general e implementable. Se trata de construir la matriz de dicha transformación.

5.9. Finding the axis and angle

(No hace falta estudiarla) Aquí el problema es inverso al anterior. Dada una matriz que representa una transformación rígida en el espacio (que preserva la orientación) y debemos sacar y calcular cual ha sido el eje de rotación y el ángulo.

Bibliografía

[1] Bill Casselman, *Mathematical Illustrations, a manual of geometry and PostScript*. Cambridge University Press, (2005).

[2] D. Escudero, *Fundamentos de Informática Gráfica*. CEYSA, (2003).