Notación y repaso

Si estás ahí sentado después de haber pasado por un proceso de admisión con vistas a ser ingeniero, seguro que has oído hablar de matrices y vectores. No es necesario que recuerdes demasiado ya que la teoría se construirá desde el principio para que tenga mayor generalidad. Aquí vamos a ver una brevísima y somera introducción sobre todo para fijar la notación.

0.1. Matrices

Una *matriz* $m \times n$ es solo una tabla de números con m filas y n columnas ($\mathsf{altura} \times \mathsf{base}$). En este curso los números serán reales la mayor parte del tiempo pero los números complejos no solo existen en ingeniería sino que son importantes, por tanto no nos olvidaremos del todo de ellos. Por cierto, en algunas aplicaciones es interesante considerar matrices más exóticas, por ejemplo de bits sujetos a operaciones especiales. Denotaremos con $\mathcal{M}_{m\times n}(\mathbb{R})$ al conjunto de todas las matrices reales $m \times n$. Si uno permite números complejos, escribiremos $\mathcal{M}_{m\times n}(\mathbb{C})$ y $\mathcal{M}_{m\times n}$ si no tenemos interés en distinguir los dos casos (aunque a partir del segundo capítulo preferiremos $\mathcal{M}_{m\times n}(K)$ a $\mathcal{M}_{m\times n}$). Las *matrices cuadradas*, con $m = n$, son más importantes que el resto y se suele abreviar $m \times n$ por m en la notación. Por ejemplo $\mathcal{M}_2(\mathbb{R})$ son las matrices reales 2×2. Como $\mathbb{R} \subset \mathbb{C}$, se tiene $\mathcal{M}_{m\times n}(\mathbb{R}) \subset \mathcal{M}_{m\times n}(\mathbb{C})$.

Las matrices se suelen denotar con letras mayúsculas y sus *elementos*, los números que la integran, con la letra minúscula correspondiente, indicando con subíndices la fila y la columna, en este orden. Por ejemplo

$$A = \begin{pmatrix} 2 & 3 & 5 \\ 7 & 11 & 13 \end{pmatrix} \quad \Longrightarrow \quad a_{12} = 3, \quad a_{22} = 11, \quad a_{23} = 13.$$

Muchas veces se escribe $A = (a_{ij})$, o $A = (a_{ij})_{i,j=1}^{2,3}$ si se quiere indicar el número de filas y columnas.

La matriz que tiene todos sus elementos cero se conoce como *matriz nula* y a veces se denota con O. Entre las matrices cuadradas destacan las *diagonales* cuyo nombre se explica por sí solo: son las matrices D con $d_{ij} = 0$ cuando $i \neq j$. Si además $d_{ii} = 1$ para cada i, se dice que tenemos la *matriz identidad*, denotada por I (o por I_n si se quiere indicar su tamaño).

Una operación sobre las matrices que a este nivel parecerá arbitraria[1] es la *trasposición* que consiste en intercambiar filas y columnas. De esta forma convierte matrices de $\mathcal{M}_{m\times n}$ en matrices de $\mathcal{M}_{n\times m}$. Se suele indicar con el superíndice t. Una cosa curiosa es que para matrices complejas la operación relevante, en relación con la nota a pie de página que quizá acabas de leer, no es la trasposición sino trasponer y después conjugar. Los físicos lo suelen indicar con el superíndice † (*dagger*, daga en inglés) y como se parece mucho a la t la usaré en las notas aunque sea muy poco frecuente en matemáticas. Por supuesto, para matrices de $\mathcal{M}_{m\times n}(\mathbb{R})$ da igual aplicar t que †. Un par de ejemplos son:

$$A = \begin{pmatrix} 2 & 6 \\ 1 & 1 \\ 4 & 6 \end{pmatrix} \qquad \Rightarrow \qquad A^t = A^\dagger = \begin{pmatrix} 2 & 1 & 4 \\ 6 & 1 & 6 \end{pmatrix}$$

y

$$B = \begin{pmatrix} 1+i & 2i \\ -i & 1 \end{pmatrix} \qquad \Rightarrow \qquad B^t = \begin{pmatrix} 1+i & -i \\ 2i & 1 \end{pmatrix}, \; B^\dagger = \begin{pmatrix} 1-i & i \\ -2i & 1 \end{pmatrix}$$

donde $i = \sqrt{-1}$.

La suma de dos matrices $A, B \in \mathcal{M}_{m\times n}$ se lleva a cabo de la manera obvia sumando elemento a elemento. La multiplicación de dos matrices solo se define si el número de columnas de la primera coincide con el número de filas de la segunda. Si $A \in \mathcal{M}_{m\times n}$ y $B \in \mathcal{M}_{n\times l}$ entonces $AB \in \mathcal{M}_{m\times l}$. El elemento ij del producto se calcula con la fórmula $\sum_k a_{ik}b_{kj}$. Esto equivale a decir que se hace una operación como la del producto escalar habitual de la fila i de A por la columna j de B y el resultado da lugar al elemento ij. Por ejemplo:

$$\begin{pmatrix} 1 & 0 & 1 \\ 2 & -1 & 1 \end{pmatrix} + \begin{pmatrix} -2 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 1 \\ 5 & 0 & 2 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 & 1 \\ 2 & -1 & 1 \end{pmatrix} \begin{pmatrix} -2 & 3 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -2 & 4 \\ -5 & 6 \end{pmatrix}.$$

Te preguntarás cómo a alguien en su sano juicio se le puede ocurrir que esta es una manera sensata de multiplicar tablas. La respuesta está relacionada con la composición de funciones ya que en álgebra lineal las matrices más que tablas a secas, son tablas de coeficientes de funciones de cierto tipo. Volveremos sobre ello en un futuro capítulo. Ahora si quieres apaciguar un poco tu curiosidad, considera

$$\begin{matrix} x_1 = a_{11}y_1 + a_{12}y_2 \\ x_2 = a_{21}y_1 + a_{22}y_2 \end{matrix} \qquad \text{y} \qquad \begin{matrix} y_1 = b_{11}z_1 + b_{12}z_2 \\ y_2 = b_{21}z_1 + b_{22}z_2 \end{matrix}$$

y expresa x_1 y x_2 en términos de z_1 y z_2 sustituyendo. Obtendrás que los coeficientes vienen dados por el producto de matrices $A = (a_{ij})_{i,j=1}^2$ y $B = (b_{ij})_{i,j=1}^2$.

Con respecto a la suma y el producto la trasposición cumple

$$(A+B)^t = A^t + B^t, \qquad \text{y} \qquad (AB)^t = B^t A^t.$$

[1] Si tienes curiosidad, aquí va el *spoiler*: la gracia de esta operación es que tiene que ver con los productos escalares, incluso en un contexto más general del que conoces.

Lo mismo es cierto reemplazando t por $\dagger$. El cambio de orden en la segunda fórmula no es ninguna tontería porque si por ejemplo $A, B \in \mathcal{M}_n$ es muy inusual que AB y BA coincidan.

Dicho sea de paso, las matrices O e I son claramente elementos neutros de la suma y de la multiplicación de matrices cuadradas. Es decir, $A + O = O + A = A$ y $AI = IA = A$.

0.2. Vectores

Hasta ahora para ti un *vector* era una lista ordenada de dos o tres números, sus *coordenadas*. En este curso los vectores serán algo más general, así un polinomio o una señal podrán ser considerados como vectores en un espacio adecuado. Antes de meternos en esas finuras, es importante que pienses que los vectores de toda la vida, los que conoces, pueden tener más de tres coordenadas si se trabaja en un espacio de más dimensiones[2]. Entonces representamos un vector de los de siempre por una lista de n números y en álgebra lineal es conveniente que estos números estén escritos en vertical, es decir, como una matriz $n \times 1$, lo cual es un fastidio al escribir un libro o unos apuntes porque las líneas van en horizontal. Un truco que usaré cuando sea necesario es el símbolo de trasposición. El conjunto de todos los vectores formados por n coordenadas se denomina $\mathbb{R}^n$ o $\mathbb{C}^n$, dependiendo de si queremos considerar números reales o complejos. Como ya sabes, habitualmente se indica que algo es un vector escribiendo una flecha sobre su nombre. Así son ejemplos de vectores de toda la vida

$$\vec{v} = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 7 \end{pmatrix} = (2,3,5,7)^t \in \mathbb{R}^4 \quad \text{y} \quad \vec{w} = \begin{pmatrix} 1+i \\ i \\ 0 \\ 1 \\ 9 \end{pmatrix} \in \mathbb{C}^5.$$

El vector con todas sus coordenadas cero, llamado *vector nulo*, se suele indicar mediante el símbolo $\vec{0}$.

La manía tan rara de que los vectores ahora estén en vertical proviene de que será conveniente multiplicar matrices por vectores y que resulte otro vector. Concretamente, si consideramos un vector $\vec{v}$ como una matrix $n \times 1$ entonces para $A \in \mathcal{M}_{m\times n}$ se tendrá $A\vec{v} \in \mathcal{M}_{m\times 1}$, es decir, es un vector de m coordenadas. En álgebra lineal la razón de ser de las matrices es que indican estas funciones que pasan vectores a vectores.

¿Y cuál es la razón de ser de los vectores? Más claramente, ¿sirven para algo los vectores? A los matemáticos nos deja bastante chafados la pregunta común "¿y esto para qué sirve?" que surge después de que contemos nuestras historias, en parte

[2] Quizá digas "¿Y a mí qué me importan los espacios de dimensiones mayores si vivo en el mundo real, no como los matemáticos?". Como zasca radical, en ciertas situaciones es conveniente considerar que las señales manejadas en telecomunicaciones son vectores de un espacio de ¡infinitas dimensiones! cuyas coordenadas son los valores que toman para cada instante de tiempo.

porque pensamos mayoritariamente que la estética (la belleza) y no la utilidad es la principal justificación de las construcciones matemáticas y en parte porque no sabemos qué responder. Los vectores sirven para representar magnitudes, como las fuerzas en física, en las que no solo es importante el tamaño o intensidad sino también la dirección. Su introducción es relativamente tardía en la historia de las matemáticas, del siglo XIX, y el álgebra lineal no se asentó bien hasta principios del siglo XX, incluso a mediados de siglo no se impartía en todas las universidades, mientras que ahora forma parte de cualquier plan de estudios científico[3].

0.3. Usando el ordenador

Si estás cursando una ingeniería se da por hecho que no tienes alergia a los ordenadores. Entonces te alegrará saber que el álgebra lineal numérica está muy desarrollada. Aunque en este curso nadie te va a obligar a que uses el ordenador para cálculos con matrices, dedicaré esta sección a darte algunos rudimentos sobre cómo hacerlo. Seguro que en otras asignaturas no habrá opción, así que aprender lo básico ahora te puede ahorrar tiempo en el futuro.

De entre el *software* que se emplea en ingeniería con fines académicos el que ha alcanzado una mayor difusión es `matlab` cuyo nombre abrevia *matrix* y *laboratory*, lo que da una idea de que al menos originariamente, tiene casi 40 años, estaba orientado a cálculos con matrices. Actualmente su ámbito es más amplio, sobre todo cuando se instalan paquetes adicionales, sin embargo las matrices se siguen reflejando en la propia sintaxis y en mi experiencia (quizá sesgada) son los cálculos con ellas lo que mejor hace.

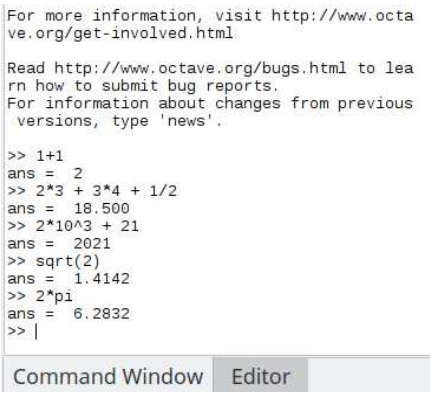
A pesar de que internamente esos cálculos se llevan a cabo con una adaptación de rutinas que hoy son de dominio público, `matlab` es *software* comercial propietario. Un punto a favor es que las licencias para estudiantes no son astronómicas y muchas universidades y departamentos tienen licencias generales. De todas formas, una alternativa libre y gratuita es (GNU) `octave`. Los comandos y programas básicos son intercambiables hasta el punto de que si no sales del ámbito básico te parecerá que el `octave` que estás usando en casa en tu portátil emula al `matlab` que usas cuando estás en el laboratorio de la universidad. Aquí no pasaremos de este ámbito básico y me referiré en lo sucesivo a `matlab/octave`.

Una vez instalado, al ejecutarlo aparecerá una interfaz de usuario con una consola. Podemos usarlo como una calculadora tecleando la cuenta que queramos hacer y pulsando la tecla `Intro`. La sintaxis para las matrices consiste en escribir los elementos entre `[` y `]` separados con espacios o comas en cada fila e indicando el cambio de fila con

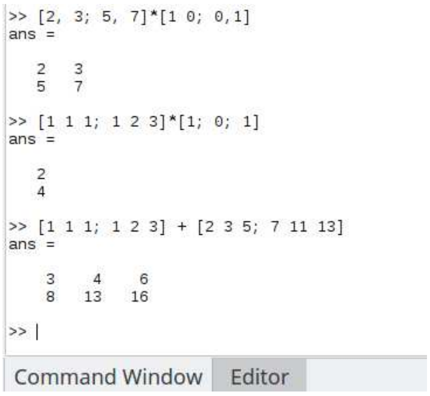
[3]La física cuántica fue en sus orígenes una mecánica matricial. W. Heisenberg no conocía la multiplicación de matrices y con la ayuda de M. Born, que había seguido cursos de matemáticas en los que aparecía, llegó a una formulación adecuada de su principio de incertidumbre. Desde el punto de vista actual llama la atención que alguien que sabía tanta física y matemáticas como Heisenberg no dominara un tema más que obligatorio en primero de grado hoy en día, pero eso fue en 1925 cuando pocos físicos lo hacían.

punto y coma. Por ejemplo, la matriz identidad 2×2 sería `[1 0; 0 1]` o `[1, 0; 0, 1]`. Como veremos, hay abreviaturas para matrices especiales.

A la izquierda vemos algunos cálculos típicos de una calculadora. Tecleando `format long` obtenemos más de las cinco cifras significativas por defecto. A la derecha hay ejemplos de suma y producto de matrices.



Cálculos numéricos en la GUI de `octave`



Cálculos con matrices

En cuanto queramos hacer algo más que un cálculo breve, es mejor utilizar un fichero en vez de teclear en la consola. Por ejemplo, creemos con nuestro editor favorito (`matlab/octave` tiene uno incorporado si lo prefieres) el archivo `myprog.m`, donde la extensión `.m` es obligatoria, conteniendo:

```
1  % Mi primer programa
2  A = [1, 0, 1; 2, -1, 1];
3  B = [-2, 1, 0; 3, 1, 1];
4  % Calcula la suma de matrices
5  A+B
6  % La traspuesta de B es
7  B'
8  % El producto de A por la traspuesta de B es
9  A*B'
10 % Esto da A
11 A+zeros(2,3)
12 % y esto vuelve a dar A*B'
13 A*B'*eye(2)
```

Por cierto, indico los números de línea para hacer referencia a ellos y facilitar la copia, no hay que teclearlos. Al escribir en la consola del interfaz `myprog` se ejecutarán secuencialmente estas operaciones Es importante que hayamos arrancado `matlab/octave` en el mismo lugar donde está `myprog.m` o que hagamos que reconozca el *path*. Con las líneas 5 y 9 tendremos una comprobación del cálculo de suma y producto de matrices que se dio como ejemplo en la primera sección. Como es fácil adivinar, `%` es el prefijo que indica un comentario. El apóstrofo corresponde a † y por tanto traspone las matrices reales. Con `zeros(n)` se tiene la matriz nula $n\times n$ y en general `zeros(m,n)` da la de dimensiones $m\times n$, mientras que `eye(n)` es la matriz identidad $n\times n$. Un punto y coma tras de una línea no muestra el resultado de la operación. Si en las línea 2 suprimimos este punto y como final, en la salida veríamos `A =` y la matriz indicada.

En `octave` nos puede resultar cansino que cuando la salida sea un poco larga haya que pasar línea a línea. Para desactivarlo basta escribir en la consola `more off`.

Terminaré refiriéndome al *software* libre para cálculo simbólico `sagemath`. Si solo vas a utilizarlo para álgebra lineal quizá no merezca la pena que pases por su instalación, la cual puede ser dura si no tienes la distribución Linux adecuada. En ese caso todavía puedes ejecutar programas o cálculos cortos *online*[4] en `https://sagecell.sagemath.org/`. Está basado en `python` y si conoces este lenguaje de programación podrás utilizarlo para hacer bucles y otras cosas. El constructor de matrices admite diversas formas. Con `matrix(m,n,L)` tenemos una matriz (simbólica) $\mathtt{m} \times \mathtt{n}$ cuyos elementos vienen dados por la lista `L`, la cual puede venir dada por los elementos leídos por filas o ser una lista de listas, una por cada fila.

La traducción a `sagemath` de las líneas 1–9 del programa anterior sería:

```
# Mi primer programa
A = matrix(2,3, [1, 0, 1, 2, -1, 1])
B = matrix(2,3, [[-2, 1, 0], [3, 1, 1]])
# Calcula la suma de matrices
print(A+B)
# La traspuesta de B es
print(B.transpose())
# El producto de A por la traspuesta de B es
print(A*B.transpose())
# Esto da A
print(A+zero_matrix(2,3))
# y esto vuelve a dar A*B^t
print(A*B.transpose()*identity_matrix(2))
```

El prefijo para comentario `#` y el comando `print` para mostrar en pantalla[5], están tomados de `python`. Si usas `sagemath` en consola, es decir, sin un interfaz como la versión *online*, esto debe ir en un archivo con extensión `.sage`, digamos `myprog.sage`, que se ejecuta con `%runfile myprog.sage`.

El hecho de que las matrices construidas de la forma anterior sean simbólicas significa en la práctica que los cálculos son exactos y que admiten variables. Por ejemplo, con `matrix(2,2,[sqrt(2),x,0,1])^2` calcularíamos el cuadrado, la multiplicación por sí misma, de una matriz 2×2 que contiene la variable `x` y en esta operación no se sustituye $\sqrt{2}$ por una aproximación con cierto número de cifras decimales, `sagemath` sabe que $(\sqrt{2})^2$ es exactamente 2.

[4]Parece ser que esta posibilidad también existe con `octave` en `https://octave-online.net/` pero no tengo experiencia al respecto.

[5]Disculpa futura: Yo uso habitualmente `sagemath` bajo `python2` en vez de bajo `python3` por tanto es posible que se me escape algún `print algo` que debiera ser `print(algo)`.