Principios del modo matemático

Composición de textos científicos

22 de septiembre de 2023

1 Dos tipos de fórmulas

Las fórmulas en un texto matemático pueden aparecer dentro de una línea *inline* o resaltadas y centradas *displayed*. Las primeras en LaTeX se encierran entre `$` y las segundas entre `\[` y `\]`. Por ejemplo la fórmula $x + y = 2$ se ha escrito como `$x+y=2$` mientras que `\[ x+y=2 \]` (que se suele escribir en tres líneas en el código fuente para que la fórmula sea más visible) daría lugar a

$$x + y = 2$$

Los espacios en las fórmulas matemáticas son en general irrelevantes, así que `\[ x +y = 2 \]` daría el mismo resultado. Cabe preguntarse en el primer caso por qué no tecleamos la fórmula como texto, sin `$`, el resultado sería entonces x+y=2 que es más feo y posiblemente incoherente con el tipo de letra de otras fórmulas que involucren x e y. Esto se aplica también a fórmulas mínimas que solo contienen un carácter. Seguro que te resultaría raro leer en un libro algo de la forma:

Despejamos la x de

$$2x - 4 = 0.$$

Lo correcto habría sido escribir "`Despejamos la $x$ de`" para que se preserve el tipo de letra.

Si examinas fuentes escritas por otras personas, a veces verás `$$...$$` en lugar de `\[...\]` porque en TeX, que es la base de LaTeX, es así. A pesar de que funciona y es más fácil de teclear, es preferible evitarlo porque en ciertas situaciones muy particulares no gestiona bien centrados y espacios (para complicar más las cosas, en principio `$...$` también es TeX y su reflejo en LaTeX debería ser `\(...\)`, pero prácticamente nadie lo usa y, hasta donde yo sé, en este caso no hay ninguna diferencia).

Una de las peculiaridades de las expresiones matemáticas es que a menudo contienen elementos que no respetan la altura natural de la línea. Los casos más simples son los subíndices y superíndices que en LaTeX se indican respectivamente con `_` y `^`. Cuando estos tengan más de un carácter o comando, los agruparemos con llaves. Por ejemplo, $E = mc^2$ es `$E=mc^2$` y

$E = mc^{1+1}$ es `$E=mc^{1+1}$`. Esto lleva a la duda de cómo teclear una llave si queremos que aparezca de verdad, basta poner `\` delante. Así tenemos

$$\{x_n\} \ \leftrightarrow\ \texttt{\textbackslash\{x\_n\textbackslash\}} \qquad \text{y} \qquad \{x_n\}_{n=1}^{\infty} \ \leftrightarrow\ \texttt{\textbackslash\{x\_n\textbackslash\}\_\{n=1\}\^{}\textbackslash infty}$$

Además hemos aprendido que el comando `\infty` da lugar a ∞.

La regla general es que en LaTeX los símbolos matemáticos se reemplazan por sus nombres en inglés o por abreviaturas suyas precedidos de la barra hacia atrás `\` y, en general, esta misma filosofía se aplica a los comandos LaTeX aunque no correspondan a símbolos matemáticos (como vimos con `\LaTeX` y `\today`). Es también un hecho general que los espacios tras un comando no se tienen en cuenta, por ello se usa `{\LaTeX}` o `\LaTeX{}` si queremos aislar el resultado del comando `\LaTeX` de la siguiente palabra.

Ejemplos bastante comunes de comandos matemáticos son los que dan lugar a las letras griegas, para las que se usan el nombre completo. Si se desean en mayúsculas y no existen en el alfabeto latino, se pone la primera letra en mayúsculas. Así

```
\[
 \alpha^\beta,
 \qquad
 \Omega^{\lambda^2},
 \qquad
 \theta(\pi)
\]
```

da lugar a

$$\alpha^\beta, \qquad \Omega^{\lambda^2}, \qquad \theta(\pi)$$

y además hemos aprendido que `\qquad` introduce cierta separación entre fórmulas. En la línea de lo explicado, no existe `\Alpha` porque la letra α mayúscula es similar a la del alfabeto latino. Las letras griegas ϵ, θ, ρ y ϕ admiten variantes tipográficas usadas históricamente. Se indican precediendo el nombre con `var`. Por ejemplo, con `\varepsilon` se obtiene ε, que es la tipografía preferida en libros de cálculo.

Las funciones trigonométricas básicas son

`\sin`, `\cos`, `\tan`, etc.

Si utilizamos el paquete de idioma `babel` con la opción `spanish` tendremos la traducción `\sen`. Más adelante en el curso veremos cómo definir `\sen` u otros comandos nosotros mismos. Otras funciones son

`\log`, `\exp`, `\min`, `\max`, `\inf`, `\sup`

y seguro que puedes imaginar más ejemplos y comprobarlos en un manual.

De nuevo surge la pregunta de para qué necesitamos escribirlo de esta forma si podríamos teclearlo como texto. Seguramente lo siguiente responda a la pregunta:

`$cos 0 + tan(0+0)$`	$\to$	$cos 0 + tan(0+0)$	Mal
`cos 0 + tan(0+0)`	$\to$	cos 0 + tan(0+0)	Mal
`$\cos 0 + \tan(0+0)$`	$\to$	$\cos 0 + \tan(0+0)$	Bien

De acuerdo, en la del centro quizá habría que decir que está regular. Su problema es que, siendo un poco exigente, el espaciamiento no es el adecuado.

2 Fracciones y raíces

Hay algunos comandos LaTeX que necesitan argumentos. Los más comunes son las fracciones. Su formato es `\frac{}{}` donde entre las primeras llaves se pone el numerador denominador y entre las segundas el denominador. Omitir las llaves solo funciona con números de un dígito. La mayor parte de los editores completan el comando para que sea más fácil de usar. LaTeX toma decisiones "inteligentes" acerca del tamaño de la fracción. Por ejemplo, con `E=\frac{3}{4} mc^2` obtenemos $E = \frac{3}{4}mc^2$ en una fórmula *in-line*, pero como *displayed* sería

$$E = \frac{3}{4}mc^2$$

(por cierto, aunque suene a broma, esta fórmula apareció en la física del siglo XX para la energía del electrón antes de la llegada de la relatividad).

También las fracciones de fracciones siguen normas sensatas acerca del tamaño:

$$\frac{x+\frac{1}{2}}{y} \quad \leftrightarrow \quad \texttt{\textbackslash frac\{x+\textbackslash frac 12\}\{y\}}$$

Más adelante veremos como cambiar esto a nuestro gusto.

Las raíces cuadradas responden a `\sqrt{...}` y se generalizan a raíces de índice `n` mediante `\sqrt[n]{...}`. Por ejemplo

$$\sqrt{\frac{1+x}{1-x}} \quad \leftrightarrow \quad \texttt{\textbackslash sqrt\{\textbackslash frac\{1+x\}\{1-x\}\}}, \qquad \sqrt[3]{1+2^4} \quad \leftrightarrow \quad \texttt{\textbackslash sqrt[3]\{1+2\^{}4\}}$$

Yo prefiero leer en un libro $(1+x)^{1/5}$ que $\sqrt[5]{1+x}$ o $(1+x)^{\frac{1}{5}}$. La estructura de `\sqrt[n]{...}` no es arbitraria. En LaTeX un posible parámetro opcional se escribe entre corchetes. Buscando la analogía con algunos temas de programación (Python, C++) que probablemente te suenen, el comando `\sqrt` funciona como si asignara por defecto contenido vacío a lo que está encima del "paraguas" mientras que un parámetro opcional sobreescribe el valor por defecto. Tardaremos un tiempo en ver dentro del curso alguna otra cosa en LaTeX que admita un parámetro opcional.

3 Operadores con límites

Algunas de las funciones que hemos visto y algunos operadores (comandos) admiten límites. Como regla general los superiores se indican como si fueran superíndices y los inferiores como subíndices. Estos límites se colocan automáticamente dependiendo de si la fórmula es *in-line* o *displayed.* El primer ejemplo básico son los límites, en sentido matemático, que como cabía esperar se indican con `\lim` y, también como cabía esperar, suelen incorporar `\to` para indicar el "tiende a". Por ejemplo

$$\lim_{x\to 0^+} x^{\tan x}=1$$

se obtiene con

```
\[
 \lim_{x\to 0^+} x^{\tan x}=1
\]
```

y en una línea veríamos $\lim_{x\to 0^+} x^{\tan x}=1$.

Algo similar se puede decir de máximo, mínimo, supremo e ínfimo. Por ejemplo, ¿sabrías dar una fuente para las siguientes fórmulas?

$$\inf_{x>0}\frac{\cos x}{1+x^2} \qquad \text{y} \qquad \max_{n>1}\{\exp(10n-n^2)\}.$$

Los operadores llamados "grandes" admiten también límites superiores. Los más utilizados son

`\sum`, `\prod`, `\bigcup` y `\bigcap`

que sin sus límites dan lugar a

$$\sum, \qquad \prod, \qquad \bigcup \qquad \text{y} \qquad \bigcap.$$

Por ejemplo, la famosa identidad del problema de Basilea y una de las leyes de Morgan,

$$\sum_{n=1}^\infty \frac{1}{n^2}=\frac{\pi^2}{6} \qquad \text{y} \qquad \bigcup_{j\in J} A_j^c = \big(\bigcap_{j\in J} A_j\big)^c,$$

se obtendrían con el código

```
\sum_{n=1}^\infty
\frac{1}{n^2}
=
\frac{\pi^2}{6}
```

y

```
\bigcup_{j\in J}A_j^c
=
\big(
\bigcap_{j\in J}A_j
\big)^c
```

Más adelante veremos los diferentes delimitadores, pero el último ejemplo ya muestra que `\big(` y `\big)` producen paréntesis un poco más grandes de lo normal.

Las integrales siguen la filosofía anterior, así podríamos escribir la consecuencia del teorema fundamental del cálculo

$$\frac{d}{dx}\int_0^x f = f(x)$$

con

```
\frac{d}{dx}
\int_0^x f = f(x)
```

Hay un punto con las integrales sobre el que volveremos más adelante y es que cómo espaciar el diferencial. Por ejemplo, si uno es un poco exigente la fórmula

$$\int_0^1 x dx= \frac 12$$

que se obtendría con `\int_0^1 x dx= \frac 12` queda un poco fea porque x y dx están pegados. Casi todo el mundo se contenta con $x\,dx$ y algunos quieren ver $x\,\mathrm{d}x$, esto es, con la "d" recta. Si no deseas mirar ahora el código fuente de esta guía, espera a que avance un poco más el curso.

4 Operadores binarios y relaciones binarias

Aparte de cosas como `=`, `>` y alguna otra más que podemos teclear directamente con nuestro teclado los operadores que suelen relacionar dos términos de una expresión matemática son

$\le$	`\le`	$\ge$	`\ge`	$\cup$	`\cup`	$\cap$	`\cap`
$\in$	`\in`	$\equiv$	`\equiv`	$\subset$	`\subset`	$\supset$	`\supset`
$\times$	`\times`	$\cdot$	`\cdot`	$\vee$	`\vee`	$\wedge$	`\wedge`
$\sim$	`\sim`	$\simeq$	`\simeq`	$\approx$	`\approx`	$\circ$	`\circ`
$\pm$	`\pm`	$\setminus$	`\setminus`	$\mid$	`\mid`	$\perp$	`\perp`

Muchos de ellos se pueden negar al precederlos de `\not`. Así $\not\approx$ se obtienen con `\not\approx`. Están las abreviaturas particulares `\notin`, `\ne` y `\nmid` para $\notin$, $\ne$ y $\nmid$. La última es casi obligatoria porque `\not\mid` no da el resultado esperado. Con mucho, `\nmid` es el comando que más uso de los definidos en el paquete `amssymb`, lo que da una idea de que el resto no son muy comunes.

5 Otros símbolos

Ya conocemos símbolos como ∞ o las letras griegas. Los editores de LaTeX suelen ofrecer en algún apartado una lista de los símbolos más comunes u otros más extraños. Incluso los usuarios más experimentados de vez en cuando los dan vistazo. Dependiendo de los temas sobre los que solamos escribir, unos símbolos se volverán mas o menos comunes. Algunos bastante universales son

∂ `\partial` ∇ `\nabla` $\emptyset$ `\emptyset` $\forall$ `\forall` $\exists$ `\exists`

La negación del último tiene una abreviatura propia `\nexists` que da $\nexists$, porque `\not\exists\verb` no queda bien alineado. Otros preferirán emplear `\lnot\exists` ya que `\lnot` es la negación lógica $\neg$. Aunque el comando por defecto para el conjunto vacío es `\emptyset`, la verdad es que queda más bonito $\varnothing$ obtenido con `\varnothing`.

Personalmente, los siguientes cuatro símbolos que son variantes de letras los uso con cierta frecuencia, sobre todo los dos primeros (para indicar partes reales e imaginarias):

$\Re$ `\Re` $\Im$ `\Im` $\wp$ `\wp` ℓ `\ell` $\hbar$ `\hbar`

El último es una necesidad si a uno le gusta la física teórica.

Otros cinco símbolos que apenas uso, pero que a ti te pueden resultar interesantes, son:

$\aleph$ `\aleph` $\angle$ `\angle` $\square$ `\square` $\triangle$ `\triangle` $\heartsuit$ `\heartsuit`

Mi recomendación es que no pierdas el tiempo memorizando muchos símbolos. Al final aprenderás los que uses habitualmente y el resto los olvidarás.

Aunque la aplicación web `detexify` es poco más que un juguete, nos puede sacar de algún apuro si tenemos pulso y no tenemos ganas de buscar en la documentación. Basta dibujar en `https://detexify.kirelabs.org` el símbolo deseado e intentará darnos el código.