# Más sobre el modo matemático

Composición de textos científicos

25 de septiembre de 2020

## 1 Recuperándose de los errores

Cuando uno comienza a usar LaTeX va muy lento en gran medida porque las posibilidades de cometer errores son grandes. Los editores actuales con sus colores y autocompletado ayudan bastante pero es habitual al principio quedarse atascado porque el compilador se queja o se niega a producir un fichero de salida. Estos errores difíciles se producen casi siempre en las fórmulas porque es en ellas donde aparecen más comandos y la sintaxis es más intrincada.

Comencemos provocando un error con `\[ \frac{x+1}{y-1 \]` donde hemos olvidado la llave final. El editor (en este caso Kile) avisa

```
[LaTeX] finished with exit code 1
guide03.tex:0:File ended while scanning use \frac
```

que es bastante informativo. Aquí `guide03.tex` es el nombre del fichero y lo de `exit code 1` es un mensaje de error típico que salta cuando se ha llegado al final del documento sin terminar una tarea.

Aparte de los errores propiamente dichos hay también avisos. El más común en la redacción de un trabajo largo es `Overfull` y con menos frecuencia `Underfull`. En rigor con la jerga al uso más que un aviso habría que decir que es una `badbox`. El corazón TeX de LaTeX tiene un poderoso algoritmo para distribuir las palabras en párrafos y habitualmente dispondremos en nuestro sistema de un diccionario que da indicaciones sobre la división de palabras en sílabas pero hay veces en que la tarea de que las palabras quepan en una línea se vuelve imposible para LaTeX e indica una cantidad en puntos[1] en que ha excedido el espacio permitido. El problema suele aparecer cuando incluimos fórmulas en línea que siempre hay que evitar dividir. De todas formas vamos a forzarlo con texto escribiendo

```
\begin{verbatim}
 Estoy forzando a que respete la línea de la fuente y esta es muy larga
\end{verbatim}
```

Con el formato de este documento, en Kile el mensaje que aparece es:

[1] Un punto es cerca de un tercio de milímetro.

```
./guide03.tex:61:Overfull \hbox (48.15683pt too wide)
in paragraph
```

Lo que ha ocurrido es que el entorno[2] `verbatim` obliga a LaTeX a respetar las líneas de la fuente, lo que es imposible sin sobrepasar el margen.

Sobre todo cuando tenemos una fórmula con varias líneas, una estrategia útil para localizar los errores más escondidos es usar el símbolo `%` para comentar una línea y que no tenga efecto. Por ejemplo

```
\[
 %%% Este es un comentario
 x+
 %y}+
 z
\]
```

muestra $x + z$ y no la $y$ que está en una línea que daría el error "`Extra }, or forgotten $`" por la llave de sobra.

Al compilar se genera un fichero `.log` que tiene mucha información. La gran mayoría es irrelevante pero puede dar ciertos detalles finos de los errores avisos y *overfulls*. Vaya por delante que esos detalles finos normalmente no son demasiado útiles para los principiantes.

En mi caso al procesar con LaTeX (no PDFLaTeX) una versión casi vacía de este fichero obtuve en el `.log` como primera línea

```
This is pdfTeX, Version 3.1415926-2.5-1.40.14
(TeX Live 2013/TeX Live for SUSE Linux)
```

que da información sobre la versión de TeX empleada[3]. Después de muchas líneas de difícil interpretación se llega a un resumen de la memoria usada[4] La última línea fue

```
Output written on guide03.dvi (1 page, 1488 bytes).
```

lo que indica que se ha generado un fichero `dvi` que se puede visualizar con la herramienta que tengamos (okular, visor de Texworks...), habitualmente integrada en nuestro editor. Con PDFLaTeX pasó a ser

```
Output written on guide03.pdf (1 page, 113204 bytes).
```

[2] Un *entorno* es algo que empieza con un `begin` y acaba con un `end`.

[3] D. Knuth tuvo la humorística idea de nombrar las versiones con cifras de $\pi$.

[4] Ahora resulta increíble, pero en los primeros tiempos recuerdo verme forzado a escribir párrafos no muy largos para que TeX no lanzara el mensaje fulminante `If you really absolutely need more capacity, you can ask a wizard to enlarge me`.

seguido de cierta estadística acerca de las propiedades del PDF.

Si repetimos el error de la falta de llave en `\frac{x+1}{y-1` abriendo el `.log` con nuestro editor de texto preferido, incluso con el que usemos para LaTeX, veremos casi al final algo como

```
Runaway argument?
{y-1 \] \par \par \par \par \end {document}
\par \par \section \ETC.
! File ended while scanning use of \frac .
<inserted text>
                \par
<*> guide03.tex

I suspect you have forgotten a `}', causing me
to read past where you wanted me to stop.
I'll try to recover; but if the error is serious,
you'd better type `E' or `X' now and fix your file.

! Emergency stop.
<*> guide03.tex

*** (job aborted, no legal \end found)
```

El ``I suspect...'' es bastante ilustrativo. El resto no tanto y hace referencia a comandos de TeX como `\par`, para cambiar de párrafo, o `\end`, para terminar un fichero, que no son familiares a la mayor parte de los usuarios de LaTeX.

Al hilo del fichero `.log` quizá te haya llamado la atención que hay otros ficheros que se crean al compilar. Siempre tendremos un `.aux` que contiene información sobre las referencias. Dependiendo de algunas herramientas que usemos veremos quizá un `.toc`, un `.bbl`, un `.out` y otros más, en Texmaker seguramente también veas uno con `synctex` en la extensión. Excepto el último, que es para la comunicación con el editor para búsquedas inversas, el resto son ficheros auxiliares donde LaTeX almacena información auxiliar. Por ejemplo, el `.toc` es para la t*able* o*f* c*ontents*. Como regla, todos se pueden borrar y el único precio a pagar es que quizá haya que compilar más de una vez para que se recupere esa información temporal. Con Kile, uno no se da cuenta de ello porque sabe cuántas veces debe compilar y lo hace sin preguntar[5]. Usando este editor, la única diferencia que notaremos si borramos esos ficheros auxiliares es que al compilar tarda más. Esto es solo apreciable para documentos muy grandes.

[5]Estrictamente, he visto excepciones fuera de `article` trabajando con `beamer`.

## 2 Delimitadores

Con este palabro, me refiero a los símbolos que usamos para indicar bloques en una fórmula. En breve son los paréntesis y cosas parecidas que se abren y cierran. Los cuatro grupos más empleados son (, ); {, }; |, | y ‖, ‖. Los paréntesis y las barras simples verticales se teclean directamente. Ya sabemos que las llaves se indican con `\{` y `\}`. Para las dobles barras se emplea `\|`. Algo menos comunes son: [, ]; ⟨; ⟩ y más especializados aún ⌈; ⌉ y ⌊; ⌋.

Los que se obtienen directamente con el carácter de una tecla, quizá precedidos de `\` están resumidos en esta tabla:

| Código | `(` | `)` | `\{` | `\}` | `\|` | `\\|` | `[` | `]` |
|---|---|---|---|---|---|---|---|---|
| Resultado | ( | ) | { | } | \| | ‖ | [ | ] |

Mientras que los que tienen nombre son:

| Código | `\langle` | `\rangle` | `\lceil` | `\rceil` | `\lfloor` | `\rfloor` |
|---|---|---|---|---|---|---|
| Resultado | ⟨ | ⟩ | ⌈ | ⌉ | ⌊ | ⌋ |

Como cabe suponer la `l` y la `r` indican *left* y *right*.

Lo que distingue a los delimitadores es la posibilidad de escalarlos en función de las fórmulas que encierren. Hay dos maneras de proceder, una manual y otra automática. La primera consiste en preceder los delimitadores con uno de los siguientes operadores:

```
\big     \Big     \bigg     \Bigg
```

que están ordenados de forma creciente en el tamaño que inducen sobre el delimitador. Por ejemplo

```
\[
 \bigg\{
 \Big(
 2\cdot
 \big\lceil x_1+x_2\big\rceil
 \Big)
 \int f
 \bigg\}.
\]
```

da lugar a:

$$\left\{\left(2\cdot\big\lceil x_1+x_2\big\rceil\right)\int f\right\}.$$

En esta forma manual no hay necesidad de que los delimitadores estén compensados porque el tamaño lo elegimos nosotros y LaTeX no necesita saber dónde comienza o acaba la fórmula para obedecernos.

Por otro lado, la forma automática deja a LaTeX que decida el tamaño de los delimitadores en función de la fórmula que encierren y por ello es importante que estén compensados. Para ello se precede el inicial por `\left` y el final por `\right`. Por ejemplo,

```
\[
 \left(x+y\right)\cdot
 \left(
 \frac{x+\frac{z}{t+\frac 12}}{y+\sqrt{\frac{a}{b}+1}}
 \right).
\]
```

da lugar a

$$\left(x+y\right)\cdot\left(\frac{x+\frac{z}{t+\frac 12}}{y+\sqrt{\frac{a}{b}+1}}\right).$$

El primer `\left` y el primer `\right` son innecesarios, el resultado sería el mismo escribiendo simplemente `(x+y)` porque esos paréntesis no necesitan ser escalados ya que encierran algo de la altura habitual de una línea.

En algunas situaciones se necesita escribir un delimitador sin compensar que de todas formas guarde el tamaño de una porción de fórmula. La manera de conseguirlo es abrirlo o cerrarlo con `\left.` o `\right.` (el punto es importante) que actúan como si allí estuviera el delimitador pero sin mostrarlo. Un pequeño reto es conseguir de esta forma

$$\left|\frac{1}{2}\right\rangle$$

donde los delimitadores `|` y `\rangle` guardan el tamaño de la fracción pero el primero no se cierra y el segundo no se abre. La fuente empleada para esta fórmula fue

```
\[
 \left.
 \left|
 \frac 12
 \right.
 \right\rangle
\]
```

El `\left.` es el que sustituye al inexistente `\left\langle` mientras que `\right.` hace las veces de `\right|` sin mostrar la barra.

# 3 Vectores y matrices

Los vectores en álgebra lineal se representan habitualmente poniendo una flechita sobre la letra que los nombra. En LaTeX se utiliza `\vec{...}`. Por ejemplo, la relación entre productos vectoriales y escalares del llamado triple producto vectorial:

$$\vec{a} \times (\vec{b} \times \vec{c}) = (\vec{a} \cdot \vec{c})\vec{b} - (\vec{a} \cdot \vec{b})\vec{b}$$

se escribiría como

```
\vec{a}\times (\vec{b}\times \vec{c})
=
(\vec{a}\cdot \vec{c})\vec{b}
-
(\vec{a}\cdot \vec{b})\vec{b}
```

Hay algún pequeño defecto con los vectores relativo a espaciamientos (que quizá ya alguien haya advertido en los paréntesis anteriores). Por ejemplo cuando queremos poner un apóstrofo tras un vector llamado $p$ deberíamos escribir `$\vec{p}'$` pero esto da lugar a $\vec{p}'$ que es muy feo, sin embargo $\vec{a}'$ no da problemas. La mejor manera de evitar esto es definir un nuevo comando pero eso por ahora está fuera de nuestros conocimientos. Aparte de este defecto, hay que tener en cuenta que la flecha no se ajusta al nombre del vector por ello debemos teclear `$\vec{p}_0$` y evitar `$\vec{p_0}$` pues los resultados son respectivamente $\vec{p}_0$ y $\vec{p_0}$. De igual forma no es aconsejable (ni habitual) que el nombre de un vector tenga más de un carácter. Si uno quiere que la flecha se ajuste, debe usar `$\overrightarrow$`. Por ejemplo el vector de $A$ a $B$ se indicaría con `$\overrightarrow{AB}$` que produce $\overrightarrow{AB}$. Hay otras soluciones basadas en paquetes especiales.

Para las matrices se suele utilizar el entorno `matrix`. La separación entre los elementos de una fila de una matriz se indica con `&` y la separación entre filas con `\\` así qeu en un primer intento de escribir la matriz identidad $2 \times 2$ usaríamos

```
\[
 \begin{matrix}
  1 & 0
  \\
  0&1
 \end{matrix}
\]
```

donde los espacios se han puesto un poco al azar para mostrar que son irrelevantes. El resultado no es el teníamos en mente:

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

Los paréntesis que nos faltan se pueden añadir como delimitadores, concretamente

```
\[
 \left( \begin{matrix} 1 & 0
  \\   0&1 \end{matrix} \right)
\]
```

(de nuevo se ha jugado al azar con espacios y líneas) daría lugar al resultado deseado

$$\left( \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \right)$$

y obviamente si nos gusta más que las matrices parezcan con corchetes bastaría cambiar `(` y `)` por `[` y `]`. La barra nos da una forma de escribir determinantes.

```
\[ \left|
\begin{matrix} 2&1\\ 5&3\end{matrix}
\right|=1 \]
```

produce $\left| \begin{matrix} 2 & 1 \\ 5 & 3 \end{matrix} \right| = 1$

Como las matrices se escriben prácticamente siempre con sus delimitadores, en uno de los paquetes matemáticos habituales hay definiciones de entornos que ya los incluyen. Lo nombres de estos entornos y el tipo de delimitador que introducen son:

| `pmatrix` | `bmatrix` | `Bmatrix` | `vmatrix` | `Vmatrix` |
|---|---|---|---|---|
| paréntesis | corchetes | llaves | barras | dobles barras |

Visualmente su aspecto es, respectivamente:

$$\left( \begin{matrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{matrix} \right), \quad \left[ \begin{matrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{matrix} \right], \quad \left\{ \begin{matrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{matrix} \right\}, \quad \left| \begin{matrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{matrix} \right|, \quad \left\| \begin{matrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{matrix} \right\|.$$

Por ejemplo, para la tercera matriz se ha usado

```
\begin{Bmatrix}
 1&1&2 \\ 3&5&8
\end{Bmatrix}
```

Los elementos de las matrices siempre aparecen centrados. Así

```
\begin{matrix}
 1 &       2&3&4 \\ 5&6&7&8 \\ 9&1000&11&12
\end{matrix}
```

genera

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 1000 & 11 & 12 \end{matrix}$$

Hay un entorno llamado `array` que es más primitivo que `matrix` y obliga a especificar el tipo de alineamiento que uno quiere: centrado, justificado a la derecha o justificado a la izquierda. Lo veremos cuando estudiemos con detenimiento las tablas porque tiene un formato similar.

En ocasiones a uno le gustaría tener una matriz de tamaño menor que el habitual, por ejemplo para mencionarla en una línea de texto (aunque esto es en general muy poco aconsejable). Existe un entorno llamado `smallmatrix` que tiene este efecto. Por ejemplo

```
\[
 \left(\begin{smallmatrix} 0&1\\ 1&0\end{smallmatrix}\right)^2
 =
 \begin{pmatrix} 1&0\\ 0&1 \end{pmatrix}.
\]
```

produce

$$\left(\begin{smallmatrix} 0&1\\ 1&0\end{smallmatrix}\right)^2 = \begin{pmatrix} 1&0\\ 0&1 \end{pmatrix}.$$

Hasta donde yo sé en los paquetes habituales no hay definidos entornos que incorporen los delimitadores a `smallmatrix` por lo que hay que usar `\left` y `\right` con este fin.

# 4 Flechas

En matemáticas se usan a menudo flechas horizontales para indicar por ejemplo implicaciones, límites y funciones. Las más básicas son

$\leftarrow$ `\leftarrow` $\rightarrow$ `\rightarrow` $\leftrightarrow$ `\leftrightarrow`

La segunda es la misma que el `\to` de los límites. La idea a tener en mente es que estas flechas se alargan si precedemos su nombre con `long` y se vuelven anchas como las habituales de implicación si ponemos en mayúsculas la primera letra. Por ejemplo

$$x \to y \Rightarrow 0 \leftarrow x - y \Longleftrightarrow f : A \longrightarrow B$$

se obtiene con

```
x\rightarrow y
\Rightarrow
0\leftarrow x-y
\Longleftrightarrow
f:A\longrightarrow B
```

Otra flecha a la derecha que se usa con cierta frecuencia es `\mapsto` con su variante de mayor longitud `\longmapsto` que son similares a lo que se obtendría con `\rightarrow` pero con una pequeña barra vertical a la izquierda. En matemáticas se emplea para indicar que estamos considerando la imagen por una función de un elemento, no de todo el dominio. Por ejemplo `a\mapsto b` se ve como $a \mapsto b$.

Hay también flechas verticales que tienen unos nombres igualmente intuitivos.

$\uparrow$ `\uparrow` $\downarrow$ `\downarrow` $\updownarrow$ `\updownarrow`

El efecto de poner la primera letra en mayúscula es como antes:

$\Uparrow$ `\Uparrow` $\Downarrow$ `\Downarrow` $\Updownarrow$ `\Updownarrow`

Existen también flechas en direcciones de los puntos cardinales intermedios (noreste, etc.) las cuales se indican con sus siglas en inglés, por ejemplo `ne` corresponde a *northeast*.

$\nearrow$ `\nearrow` $\searrow$ `\searrow` $\swarrow$ `\swarrow` $\nwarrow$ `\nwarrow`

Hay otras muchas flechas definidas en los paquetes matemáticos que ya se han mencionado[6] pero es poco práctico intentar memorizarlas porque son poco frecuentes. Recuérdese que los editores normalmente incorporan una ayuda con listas de símbolos donde podemos explorar símbolos raros. En mi caso en el apartado *Arrows* en la ayuda a la izquierda de Kile veo flechas como $\rightarrowtail$, $\circlearrowleft$, $\rightleftharpoons$, $\rightsquigarrow$ y otras muchas que creo no haber empleado nunca. Quien tenga curiosidad en una de estas cuatro en particular puede mirar la fuente de este documento. Si la curiosidad es mayor se debe ir a la documentación general de LaTeX provista por algún manual o a la específica de los paquetes matemáticos.

[6]Es medianamente común entre los matemáticos cargar también `latexsym` que añade una flecha más.