

# Matematică

Una din forțele cele mai motivatoare pentru Donald Knuth când a început să dezvolte sistemul TeX inițial a fost să creeze ceva care să permită construcția simplă a formulelor matematice și să arate bine în documentul final. Faptul că el a reușit a fost cel mai probabil motivul pentru care TeX (și mai târziu, LaTeX) a devenit atât de popular în comunitatea științifică. Indiferent de trecutul acestora, redactarea formulelor matematice este unul din punctele forte ale sistemului LaTeX. Este de asemenea un subiect vast datorită omniprezenței notațiilor matematice.

Dacă scrieți un document care necesită numai câteva formule matematice simple, atunci puteți folosi în general *plain* LaTeX: vă va pune la dispoziție cele mai multe instrumente de care aveți nevoie. Totuși, dacă scrieți un document științific ce conține formule numeroase și complicate, se recomandă să folosiți pachetul `amsmath` sau `mathtools` (care include automat pachetul `amsmath`), care introduc mai multe comenzi noi, ce oferă mai multe facilități decât *plain* LaTeX. Pentru a include aceste pachete, scrieți:

```
\usepackage{amsmath}
```

sau

```
\usepackage{mathtools}
```

în preambulul documentului.

## Mediile matematice

LaTeX trebuie să știe dinainte că textul ce urmează conține elemente matematice. Asta deoarece LaTeX formatează notațiile matematice diferit de textul normal. De aceea, în acest scop au fost dezvoltate medii speciale. Ele pot fi împărțite în două categorii în funcție de modul lor de prezentare:

- *text* — formulele din modul `text` sunt afișate *in-line*, adică în corpul de text în care sunt declarate. Spre exemplu, pot spune că  $a + a = 2a$  în cadrul acestei fraze.

- *displayed* — formulele în modul de afișare (*displayed*) sunt separate de corpul principal de text.

Întrucât formulele matematice necesită medii speciale, există evident nume corespunzătoare de medii folosite în modul obișnuit. Spre deosebire de cele mai multe alte medii, totuși, oferă câteva scurtături utile pentru a declara formulele. Tabelul următor rezumă toate acestea:

Tip	Mediu	Scurtătură	Scurtătură
		LaTeX	TeX
Text	<code>\begin{math}...\end{math}</code>	<code>\(...\)</code>	<code>\$...\$</code>
Displayed	<code>\begin{displaymath}...\end{displaymath}</code> sau	<code>\[...\]</code>	<code>\$\$...\$\$</code>
	<code>\begin{equation*}...\end{equation*}</code> <a href="#">[1]</a>		

**Notă:** Utilizarea `$$...$$` ar trebui evitată, întrucât poate cauza probleme, mai ales în cazul macrocomenzilor AMS-LaTeX. În plus, dacă apare vreo problemă, s-ar putea ca mesajele de eroare să nu vă fie de niciun ajutor.

Mai există un mediu ce poate fi folosit pentru tipul *displayed*: `equation`. Diferența dintre acesta și `displaymath` este că `equation` adaugă numere secvențiale la ecuații.

Dacă tipărești textul în mod normal, se spune că ești în *modul text*, în timp ce dacă scrii în unul din mediile matematice, se zice că ești în *modul matematic*, care prezintă câteva diferențe în comparație cu *modul text*:

1. Cele mai multe spații și capete de linie nu au nicio semnificație specială, de vreme ce toate spațiile fie sunt derivate logic din expresii matematice, fie trebuiesc specificate cu comenzi speciale precum `\quad`
2. Liniile libere nu sunt permise. Numai un paragraf per formulă.
3. Fiecare literă este considerată numele unei variabile și va fi tipărită ca atare. Dacă vrei să tipărești text normal într-o formulă (font și spațiere normale), atunci trebuie să introduceți textul folosind [comenzi specifice](#).

## Inserarea de cod matematic scris în modul "displayed" în blocuri de text

Pentru ca unii operatori, cum sunt `\lim` sau `\sum`, să fie afișați corect într-un mediu matematic (ca  $\$.....\$$ ), poate fi convenabil câteodată să adăugați comanda `\displaystyle` în interiorul mediului. Dacă faceți asta, linia s-ar putea să devină mai înaltă, însă va face ca exponenții și indecșii să fie afișați corect pentru unii operatori matematici.

## Simboluri

Matematica are o mulțime de simboluri! Dacă există un aspect dificil legat de matematica implementată în LaTeX, acesta este să încercați să rețineți comenzile matematice. Sunt desigur un set de simboluri ce pot fi accesate direct de la tastatură:

+ - = ! / ( ) [ ] < > | ' :

Pe lângă cele listate mai sus, trebuie să dați comenzi specifice pentru a genera simbolurile dorite. Și mai sunt o *mulțime* de litere din greaca veche, simboluri pentru mulțimi și relaționale, săgeți, operatori binari, etc. Spre exemplu:

$\begin{array}{l} \backslash \\ \text{forall } x \text{ \in } X, \text{ \quad } \backslash \text{exists } y \backslash \text{leq} \\ \backslash \text{epsilon} \\ \backslash \end{array}$	$\forall x \in X, \quad \exists y \leq \epsilon$
---	--

Din fericire, există un program ce poate simplifica în mare măsură căutarea comenzii pentru un simbol specific. Uitați-vă peste *Detexify* în secțiunea [legături externe](#) de mai jos. De asemenea, programele cu interfață grafică pentru LaTeX, cum sunt TexMaker sau Kile, au meniuri și butoane care permit utilizatorului să găsească ușor simbolurile căutate.

## Litere din greaca veche

În matematică se folosesc deseori litere din greaca veche, iar acestea sunt foarte ușor de tipărit în *modul matematic*. Tot ce trebuie să faceți este să scrieți explicit denumirea literei grecești respective în engleză după un backslash: dacă prima literă a denumirii este mică, veți obține litera grecească mică, iar dacă prima (și numai prima) literă este mare, veți obține litera grecească mare. Notați că unele litere grecești mari arată ca cele latinești, așa că nu sunt puse la dispoziție de LaTeX (spre exemplu, *Alpha* și *Beta* mari sunt pur și simplu "A", respectiv "B"). *Epsilon*, *theta*, *phi*, *pi*, *rho*

și *sigma* mici au două versiuni diferite în LaTeX. *Varianta* alternativă are "var" înaintea denumirii literei:

`\[`  
`\alph`  
a,  
`\beta`,  
`\gam`  
ma,  
`\delta`  
  
,  
`\epsil`  
on,  
`\vare`  
psilon  
,  
`\zeta`,  
`\eta`,  
`\theta`  
  
,  
`\vartheta`  
eta,  
`\iota`,  
`\kappa`  
a,  
`\lamb`  
da,  
`\mu`,  
`\nu`,  
`\xi`,  
`\pi`,  
`\varpi`  
, `\rho`,  
`\varrh`  
o,  
`\sigm`  
a,  
`\varsig`  
gma,  
`\tau`,  
`\upsil`  
on,

$\alpha, \beta, \gamma, \delta, \epsilon, \varepsilon, \zeta, \eta, \theta, \vartheta, \iota, \kappa, \lambda, \mu, \nu, \xi, \pi, \varpi, \rho$

`\phi,`  
`\varphi`  
`hi,`  
`\chi,`  
`\psi,`  
`\omega`  
`ga,`  
`\Gamma`  
`mma,`  
`\Delta`  
`,`  
`\Theta`  
`a,`  
`\Phi,`  
`\Omega`  
`ga`  
`\]`

## Operatori

Un operator este o funcție scrisă sub forma unui cuvânt, spre exemplu: funcții trigonometrice (sin, cos, tan), logaritmi și funcții exponențiale (log, exp). LaTeX are multe dintre acestea definite sub formă de comenzi:

`\[`  
`\cos(2\theta) = \cos^2 \theta`  
`- \sin^2 \theta`  
`\]`

$$\cos(2\theta) = \cos^2 \theta - \sin^2 \theta$$

Pentru anumiți operatori de genul [limitelor](#), indicele este așezat sub operator:

`\[`  
`\lim_{x \to \infty} \exp(-x) = 0`  
`\]`

$$\lim_{x \rightarrow \infty} \exp(-x) = 0$$

Pentru [operatorul modular](#) sunt două comenzi: `\bmod` și `\pmod`:

`\[`  
`a \bmod b`  
`\]`

$$a \bmod b$$

$$\left[ \begin{array}{l} x \equiv a \pmod{b} \\ \end{array} \right] \quad \left| \quad x \equiv a \pmod{b} \right.$$

Pentru a utiliza operatori care nu sunt predefiniți, de genul [argmax](#), vezi [Operatori personalizați](#).

## Puteri și indici

Puterile și indicii sunt echivalente cu conceptele similare din modul text. Semnul de omisiune (^) se folosește pentru a scrie sub formă de exponent, iar semnul de subliniere (\_) are rolul de a scrie ca indice. Dacă se ridică la putere sau se scrie sub formă de indice mai mult de un caracter, acestea ar trebui grupate cu ajutorul unor acolade: ({} și {}).

$$\left[ \begin{array}{l} k_{n+1} = n^2 + k_n^2 - \\ k_{n-1} \end{array} \right] \quad \left| \quad k_{n+1} = n^2 + k_n^2 - k_{n-1} \right.$$

Un underscore (\_) poate fi utilizat cu o bară verticală (|) pentru a denota evaluarea cu ajutorul unei notații de indici în matematică:

$$\left[ \begin{array}{l} f(n) = n^5 + 4n^2 + 2 \\ \_{}_{n=17} \end{array} \right] \quad \left| \quad f(n) = n^5 + 4n^2 + 2|_{n=17} \right.$$

## Fracții și binomi

Puteți crea o fracție în LaTeX cu ajutorul comenzii `\frac{numărător}{numitor}`. De asemenea, [coeficientul binomial](#) (*choose function*) poate fi scris cu ajutorul comenzii `\binom`<sup>[1]</sup>:

$$\left[ \begin{array}{l} \frac{n!}{k!(n-k)!} = \binom{n}{k} \\ \end{array} \right] \quad \left| \quad \frac{n!}{k!(n-k)!} = \binom{n}{k} \right.$$

Puteți folosi de asemenea comanda `\choose` fără pachetul `amsmath`:

$$\left[ \frac{n!}{k!(n-k)!} = {n \choose k} \right] \quad \left| \quad \frac{n!}{k!(n-k)!} = \binom{n}{k} \right.$$

Puteți scrie fracții în cadrul altor fracții:

$$\left[ \frac{\frac{1}{x} + \frac{1}{y}}{y-z} \right] \quad \left| \quad \frac{\frac{1}{x} + \frac{1}{y}}{y-z} \right.$$

Notați că atunci când apare în cadrul altei fracții, sau încorporată în text,  $\frac{a}{b}$ , o fracție este în mod vizibil mai mică decât în formatul matematic de afișare (*displayed*). Comenzile `\frac` și `\dfrac`<sup>[1]</sup> forțază utilizarea stilurilor respective (în mod similar, comenzile `\tbinom` și `\dbinom` fac același lucru pentru coeficienții binomiali).

Un alt mod de a scrie fracții constă în utilizarea comenzii `\over` fără pachetului `amsmath`:

$$\left[ {n! \over k!(n-k)!} = {n \choose k} \right] \quad \left| \quad \frac{n!}{k!(n-k)!} = \binom{n}{k} \right.$$

Pentru fracții relativ simple, poate fi mai estetic să folosiți [puteri și indici](#):

$$\left[ ^3/_7 \right] \quad \left| \quad 3/7 \right.$$

Dacă îi folosiți în tot documentul, se recomandă utilizarea pachetului `xfrac`. Acest pachet oferă comanda `\sfrac` pentru a crea fracții înclinate. Utilizare:

Pune  $\frac{1}{2}$   
căni de zahăr, \dots  
\\

$3 \times \frac{1}{2} =$   
 $1 \frac{1}{2}$   
\\

Pune  $1 \frac{1}{2}$  căni  
de zahăr, \dots  
\\

$3 \times 1 \frac{1}{2} = 1 \frac{1}{2}$   
 $1 \frac{1}{2}$   
\\

Pune  $\frac{1}{2}$  căni de zahăr, ...

Pune  $1 \frac{1}{2}$  căni de zahăr, ..

Altfel, pachetul `nicefrac` oferă comanda `\nicefrac`, a cărei utilizare este similară cu `\frac`.

## Fracții continue

Fracțiile continue ar trebui să fie scrise cu ajutorul comenzii `\cfrac`<sup>[1]</sup>:

```
\begin{equation}
x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + a_4}}}
\end{equation}
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

## Rădăcini

Comanda `\sqrt` creează un radical în jurul unei expresii. Aceasta acceptă un argument opțional specificat în paranteze drepte (`[ ]`) pentru a schimba gradul radicalului:

```
\\
\sqrt{\frac{a}{b}}
\\
\\
```

$$\sqrt{\frac{a}{b}}$$

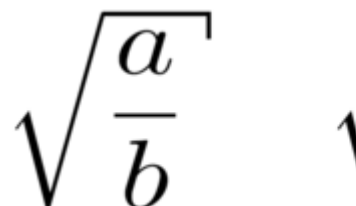
$$\sqrt[n]{1 + x + x^2 + x^3 + \dots}$$



```
\sqrt[n]{1+x+x^2+x^3+\ldots}
\]
```

Unii oameni preferă să scrie radicalul ca "închizându-se" peste conținut. Această metodă scoate în evidență ce este sub radical. Acest stil de redactare în mod normal nu este folosit când scrieți la calculator, dar dacă încă mai vreți să schimbați formatul radicalului, LaTeX vă oferă această posibilitate. Adăugați pur și simplu codul următor în preambulul documentului:

```
% Definiție nouă a radicalului:
% \sqrt este redenumit \oldsqrt
\let\oldsqrt\sqrt
% definește noul \sqrt în funcție
de cel vechi
\def\sqrt{\mathpalette\DHLhksqrt}
\def\DHLhksqrt#1#2{
\setbox0=\hbox{${#1}\oldsqrt{#2}\,}
$\dimen0=\ht0
\advance\dimen0-0.2\ht0
\setbox2=\hbox{\vrule height\ht0
depth-\dimen0}
{\box0\lower0.4pt\box2}}
```



Stilul nou este la stânga,  
cel vechi la dreapta

Acest cod TeX mai întâi redenumeste comanda `\sqrt` ca fiind `\oldsqrt`, apoi redefiște `\sqrt` în funcție de comanda veche, adăugând ceva în plus. Noul radical poate fi văzut în imaginea din dreapta, în comparație cu cel vechi. Din păcate acest cod nu merge cu un grad superior al rădăcinii: dacă încercați să scrieți  $\sqrt[b]{a}$  sub forma `\sqrt[b]{a}` după ce ați adăugat codul de mai sus, veți obține un rezultat greșit ( $[b]$  înăuntrul radicalului). Cu alte cuvinte, puteți redefini radicalul în felul acesta numai dacă nu aveți de gând să folosiți argumentul opțional al comenzii în întregul document.

## Sume și integrale

Comenzile `\sum` și `\int` inserează simbolurile pentru sumă, respectiv integrală, cu limite specificate de semnul de omisiune (`^`) și underscore (`_`):

$$\begin{array}{|l} \backslash\text{sum}_{i=1}^{10} t_i \\ \backslash \\ \backslash\text{int}_0^{\infty} e^{-x} dx \\ \backslash \end{array} \quad \left| \quad \begin{array}{l} \sum_{i=1}^{10} t_i \\ \\ \int_0^{\infty} e^{-x} dx \end{array}$$

Sunt multe alte comenzi "mari" ce operează într-o manieră similară:

<code>\sum</code>	$\Sigma$	<code>\prod</code>	$\amalg$	<code>\coprod</code>	$\amalg$
<code>\bigoplus</code>	$\bigoplus$	<code>\bigotimes</code>	$\bigotimes$	<code>\bigodot</code>	$\bigodot$
<code>\bigcup</code>	$\bigcup$	<code>\bigcap</code>	$\bigcap$	<code>\biguplus</code>	$\biguplus$
<code>\bigsqcup</code>	$\bigsqcup$	<code>\bigvee</code>	$\bigvee$	<code>\bigwedge</code>	$\bigwedge$
<code>\int</code>	$\int$	<code>\oint</code>	$\oint$	<code>\iint</code> <sup>[1]</sup>	$\iint$
<code>\iiint</code> <sup>[1]</sup>	$\iiint$	<code>\iiiiint</code> <sup>[1]</sup>	$\iiiiint$	<code>\idotsint</code> <sup>[1]</sup>	$\int \cdots \int$

Pentru mai multe simboluri legate de integrale, inclusiv cele care nu sunt incluse de fontul implicit Computer Modern, încercați pachetul `esint`.

Comanda `\substack`<sup>[1]</sup> permite utilizarea `\\` pentru a scrie limitele pe mai multe linii:

$$\begin{array}{|l} \backslash\text{sum}_{\substack{0 < i < m \\ 0 < j < n \\ \}} P(i,j) \\ \backslash \end{array} \quad \left| \quad \sum_{\substack{0 < i < m \\ 0 < j < n}} P(i,j)$$

Dacă vreți ca limitele unei integrale să fie specificate deasupra și sub simbolul integralei (ca în cazul sumei), folosiți comanda `\limits`:

$\int_a^b$		$\int_a^b$
------------	--	------------

Totuși, dacă vrei ca acest stil să se aplice la TOATE integralele din document, este de preferat să specificeți opțiunea `intlimits` la încărcarea pachetului `amsmath`:

```
\usepackage[intlimits]{amsmath}
```

Exponenții și indicii folosiți în alte contexte, ca și alți parametri din pachetul `amsmath` legați de aceștia sunt descriși în capitolul [Matematică avansată](#).


Pentru integrale mai mari, puteți utiliza declarații personale, sau puteți folosi pachetul `bigints`<sup>[2]</sup>.

## Paranteze drepte, acolade și delimitatori

În capitolul [Matematică avansată](#) se descrie cum puteți folosi acolade în ecuații ce se întind pe mai multe linii.

Utilizarea delimitatorilor de genul parantezelor drepte capătă o importanță deosebită când aveți de-a face cu ecuații ceva mai complicate. Fără ei, formulele pot deveni ambigue. De asemenea, tipuri speciale de structuri matematice, cum sunt matricile, se bazează în mod tipic pe delimitatori pentru a fi cuprinse într-un tot unitar.

LaTeX vă pune la dispoziție o serie de delimitatori:

$\begin{aligned} & \left( \right) \left[ \right] \left\{ \right\} \left\  \right\  \left\  \right\  \left\  \right\  \\ & \angle \rangle \lfloor \rfloor \lceil \rceil \end{aligned}$		
---	--	--

## Determinarea automată a dimensiunilor

Foarte des componentele formulelor matematice vor avea dimensiuni diferite, caz în care delimitatorii ce înconjoară expresiile ar trebui să varieze în mod corespunzător. Acest lucru se poate face automat cu ajutorul

comenzilor `\left` și `\right`. Oricare dintre delimitatorii menționați mai devreme poate fi folosit în combinație cu acestea:

$$\left[ \left( \frac{x^2}{y^3} \right) \right]$$

Dacă este nevoie doar de un delimitator într-o parte a unei expresii, atunci în partea cealaltă trebuie să folosiți un delimitator invizibil, denotat de un punct, după cum se vede în exemplul următor:

$$\left[ \left( \frac{x^2}{y^3} + \frac{x^3}{y^2} + \dots \right) \right]$$

## Determinarea manuală a dimensiunilor

În anumite cazuri, dimensiunea produsă automat de comenzile `\left` și `\right` poate să nu fie cea dorită, sau poate vreți pur și simplu un control mai fin asupra mărimii delimitatorilor. În cazul acesta, puteți utiliza modificatorii `\big`, `\Big`, `\bigg` și `\Bigg`:

$$\left[ \left( \big( \Big( \bigg( \Bigg( \right. \right. \right. \right. \right.$$

## Matrici și vectori

O matrice simplă poate fi creată cu ajutorul mediului `matrix`<sup>[1]</sup>: la fel ca pentru alte structuri tabelare, intrările sunt specificate de linii, coloanele fiind separate cu un ampersand (&), trecerea la o linie nouă făcându-se cu un dublu backslash (`\\`)

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

`\]` `|`

Pentru a specifica alinierea coloanelor în tabel, folosiți versiunea stelată<sup>[3]</sup>:

<pre> \begin{matrix} -1 &amp; 3 \\ 2 &amp; -4 \end{matrix} = \begin{matrix*}[r] -1 &amp; 3 \\ 2 &amp; -4 \end{matrix*} </pre>	$\begin{matrix} -1 & 3 \\ 2 & -4 \end{matrix} = \begin{matrix} -1 & 3 \\ 2 & -4 \end{matrix}$
---	---

Alinierea implicită este `c`, însă poate fi orice tip de coloană valid în mediul `array`.

Totuși, matricile sunt de obicei încadrate de delimitatori de un anumit tip, și deși puteți folosi comenzile `\left` și `\right`, există multe alte medii predefinite ce includ automat delimitatori:

Mediu	Delimitator	Note
<code>pmatrix</code> <sup>[1]</sup>	<code>()</code>	implicit centreează coloanele
<code>pmatrix*</code> <sup>[3]</sup>	<code>()</code>	permite specificarea alinierii coloanelor în parametrul opțional
<code>bmatrix</code> <sup>[1]</sup>	<code>. </code>	implicit centreează coloanele
<code>bmatrix*</code> <sup>[3]</sup>	<code>. </code>	permite specificarea alinierii coloanelor în parametrul opțional
<code>Bmatrix</code> <sup>[1]</sup>	<code>{}</code>	implicit centreează coloanele
<code>Bmatrix*</code> <sup>[3]</sup>	<code>{}</code>	permite specificarea alinierii coloanelor în parametrul opțional
<code>vmatrix</code> <sup>[1]</sup>	<code> </code>	implicit centreează coloanele
<code>vmatrix*</code> <sup>[3]</sup>	<code> </code>	permite specificarea alinierii coloanelor în parametrul opțional
<code>Vmatrix</code> <sup>[1]</sup>	<code>  </code>	implicit centreează coloanele
<code>Vmatrix*</code> <sup>[3]</sup>	<code>  </code>	permite specificarea alinierii coloanelor în

parametrul opțional

Când scrieți matrici de dimensiuni arbitrare, se obișnuiește să folosiți triplete de puncte ([puncte de suspensie](#)) orizontale, verticale și diagonale pentru a umple anumite coloane și linii. Acestea pot fi specificate cu ajutorul comenzilor `\cdots`, `\vdots`, respectiv `\ddots`:

$$\begin{array}{l} \left[ \right. \\ A_{\{m,n\}} = \\ \begin{array}{l} \backslash\text{begin}\{\text{pmatrix}\} \\ a_{\{1,1\}} \& \\ a_{\{1,2\}} \& \cdots \\ \& a_{\{1,n\}} \\ \backslash\backslash \\ a_{\{2,1\}} \& \\ a_{\{2,2\}} \& \cdots \\ \& a_{\{2,n\}} \\ \backslash\vdots \& \backslash\vdots \\ \& \backslash\ddots \& \\ \backslash\vdots \backslash\backslash \\ a_{\{m,1\}} \& \\ a_{\{m,2\}} \& \cdots \\ \& a_{\{m,n\}} \\ \backslash\text{end}\{\text{pmatrix}\} \\ \backslash\backslash \end{array} \\ \left. \right] \end{array} \quad A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

În unele cazuri, poate veți dori să aveți un control mai fin asupra alinierii la nivel de coloană, sau să inserați linii între coloane sau rânduri. Puteți face asta cu ajutorul mediului `array`, care este în esență o versiune în modul matematic a mediului `tabular` și care necesită specificarea în avans a coloanelor:

$$\begin{array}{l} \left[ \right. \\ \begin{array}{l} \backslash\text{begin}\{\text{array}\}\{\text{c|c}\} \\ 1 \& 2 \\ \backslash\text{hline} \\ 3 \& 4 \\ \backslash\text{end}\{\text{array}\} \\ \backslash\backslash \end{array} \\ \left. \right] \end{array} \quad \begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}$$

Poate ați observat că mediile din clasa matriceală AMS nu lasă destul spațiu vertical când se folosesc împreună cu fracții, ducând la rezultate similare cu:

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

Pentru a rezolva această problemă, adăugați un spațiu vertical adițional cu parametrul opțional al comenzii `\|`:

<pre> \  M = \begin{bmatrix}   \frac{5}{6} &amp; \frac{1}{6} &amp; 0 \\   \frac{5}{6} &amp; 0 &amp; \frac{1}{6} \\   0 &amp; \frac{5}{6} &amp; \frac{1}{6} \end{bmatrix} \  </pre>	$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$
--	---

Dacă vă trebuie "margini" sau "indecși" la matrice, *plain* TeX oferă macroul `\bordermatrix`:

<pre> \  M = \bordermatrix{~ &amp; x &amp; y \cr   A &amp; 1 &amp; 0 \cr   B &amp; 0 &amp; 1 \cr} \  </pre>	$M = \begin{matrix} & x & y \\ A & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ B & \end{matrix}$
---	--

## Matrici inserate în text

Pentru a insera o matrice mică și a nu spori spațierea liniei ce o conține, folosiți mediul `smallmatrix`:

<p>O matrice inserată în text trebuie să fie mai mică</p> <pre> <math>\bigl(\begin{smallmatrix} a &amp; b \\ c &amp; d \end{smallmatrix}\bigr)</math> </pre>	<p>O matrice inserată în text trebuie să fie mai mică <math>\left(\begin{smallmatrix} a &amp; b \\ c &amp; d \end{smallmatrix}\right)</math> pentru a nu crește spațierea unei porțiuni de text</p>
--	---

```
\end{smallmatrix}
\biggr)$
pentru a nu mări
spațierea unei
porțiuni de text.
```

## Adăugarea de text în ecuații

Mediul matematic diferă de mediul text prin reprezentarea textului. Iată un exemplu în care încercăm să reprezentăm text în mediul matematic:

```
\[
 50
mere
\times
100
mere =
foarte
multe
mere^
2
\]
```

$$50\text{mere} \times 100\text{mere} = \text{foartemultemere}^2$$

Sunt două probleme demne de menționat aici: nu sunt spații între cuvinte sau numere, iar literele sunt scrise cu caractere italice și mai spațiate decât ar fi normal. Ambele probleme sunt simple artefacte ale modului matematic, care apar în orice expresie matematică: spațiile sunt ignorate (LaTeX spațiază codul matematic potrivit cu propriile sale reguli) și fiecare caracter este un element separat (astfel că nu sunt așezate așa de apropiat ca în modul text).

Sunt mai multe moduri în care putem adăuga text în mod corespunzător în mediul matematic. Soluția cea mai întâlnită este să scrieți textul ca argument al comenzii `\text{...}`<sup>[1]</sup> (o comandă similară este `\mbox{...}`, deși aceasta generează probleme cu indicii și are un nume mai puțin descriptiv). Haideți să vedem ce se întâmplă când adaptăm codul ecuației de mai sus în acest fel:

```
\[
 50
\text{mere}
\times
\]
```

$$50\text{mere} \times 100\text{mere} = \text{foarte multe mer}$$



```

100
\text{mere
}
=
\text{foart
e multe
mere}^2
\]

```

Textul arată mai bine. Cu toate acestea, nu sunt spații între numere și cuvinte. Din păcate, trebuie să adăugați astfel de spații manual. Sunt mai multe căi prin care puteți adăuga spații între elementele matematice, însă de dragul simplității puteți adăuga literal caracterul spațiu în corpul de `\text` afectat (chiar înainte de text):

```

\[
50 \text{
mere}
\times 100
\text{
mere}
=
\text{foart
e multe
mere}^2
\]

```

**50 mere × 100 mere = foarte multe me**

## Text formatat

Comanda `\text` este acceptabilă în cele mai multe situații. Totuși, există o alternativă care oferă ceva mai multă flexibilitate. Poate vă aduceți aminte de [comenzile de formatare a fonturilor](#), cum sunt `\textrm`, `\textit`, `\textbf`, etc. Aceste comenzi formatează argumentul în mod corespunzător, spre exemplu, `\textbf{text aldin}` dă **text aldin**. Ele sunt la fel de valide într-un mediu matematic pentru a include text. Avantajul suplimentar aici este că vă permit un control mai bun asupra formătărilor fontului comparativ cu textul standard obținut cu comanda `\text`.

```

\[
50
\textrm{

```

**50 mere × 100 mere = foarte multe m**

```

mere}
\times 100
\textbf{
mere} =
\textit{foar
te multe
mere}^2
\]

```

## Formatarea simbolurilor matematice

Deci putem formata text, ce ziceți de formatarea matematicii? Sunt mai multe comenzi de formatare foarte similare cu cele pentru fonturi folosite mai devreme, cu excepția faptului că sunt menite specific pentru textul din modul matematic<sup>[4]</sup>:

Comandă LaTeX	Exemplu	Descriere	Utilizare
<code>\mathnormal{...}</code>	<i>ABCDEFabcdef123456</i>	fontul matematic implicit	cele mai multe notații matematice
<code>\mathrm{...}</code>	ABCDEFabcdef123456	fontul implicit sau normal, neînclinat	unități de măsură, funcții cu o lungime de un cuvânt
<code>\mathit{...}</code>	<i>ABCDEFabcdef123456</i>	font italic	
<code>\mathbf{...}</code>	<b>ABCDEFabcdef123456</b>	font aldin	vectori
<code>\mathsf{...}</code>	ABCDEFabcdef123456	<a href="#">Sans-serif</a> <a href="#">Font</a> <a href="#">Monospac</a> <a href="#">e (cu</a> <a href="#">lățime</a> <a href="#">fixă)</a>	
<code>\mathtt{...}</code>	ABCDEFabcdef123456		
<code>\mathcal{...}</code>	<i>ABCDEF</i> $\rightarrow \llcorner \lrcorner \{ \infty \in \exists \Delta \nabla$	Caligrafic (numai cu	Folosit adesea

	litere mari)	pentru scheme și categorii
<code>\mathfrak{...}</code> <sup>[5]</sup> <b>A B C D E F a b c d e f 1 2 3 4 5 6</b>	<a href="#">Fraktur</a>	Font aproape standard pentru algebre Lie Folosit pentru a denota mulțimi speciale (de exemplu, numere reale)
<code>\mathbb{...}</code> <sup>[4]</sup> <b>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</b>	<a href="#">Blackboard d bold</a>	
<code>\mathscr{...}</code> <sup>[6]</sup>	<a href="#">Script</a>	

Comenzile de formatare matematică pot include întreaga ecuație, și nu doar elementele de text: ele formatează numai litere, numere și literele grecești mari, ignorând restul sintaxei matematice.

Pentru a îngroșa literele grecești mici sau alte simboluri, folosiți comanda `\boldsymbol{...}`<sup>[1]</sup>; aceasta merge numai dacă există o versiune aldină (*bold*) a simbolului din fontul curent. Ca ultimă soluție, există comanda `\pmb{...}`<sup>[1]</sup> (*poor mans bold*): tipărește mai multe versiuni ale caracterului, puțin diferite unele de altele:

$\begin{array}{l} \backslash \\ \boldsymbol{\beta} = \\ (\beta_1, \beta_2, \dots, \beta_n) \\ \backslash \end{array}$	$\beta = (\beta_1, \beta_2, \dots, \beta_n)$
---	--

Pentru a modifica mărimea fonturilor în modul matematic, vezi [Modificarea dimensiunii fontului](#).

## Accente

Ce faceți când ați terminat simbolurile și fonturile? Pasul următor constă în a folosi accente:

a'	$\hat{a}$	a''	$\bar{a}$	a'''	$\overline{aaa}$	a''''	$\check{a}$	a''	$\tilde{a}$
<code>\hat{a}</code>	<code>\hat{a}</code>	<code>\bar{a}</code>	<code>\bar{a}</code>	<code>\overline{aaa}</code>	<code>\overline{aaa}</code>	<code>\check{a}</code>	<code>\check{a}</code>	<code>\tilde{a}</code>	<code>\tilde{a}</code>
<code>\grave{a}</code>	<code>\grave{a}</code>	<code>\acute{a}</code>	<code>\acute{a}</code>	<code>\breve{a}</code>	<code>\breve{a}</code>	<code>\vec{a}</code>	<code>\vec{a}</code>		
<code>\dot{a}</code>	<code>\dot{a}</code>	<code>\ddot{a}</code>	<code>\ddot{a}</code>	<code>\dddota</code>	<code>\dddota</code>	<code>\ddddota</code>	<code>\ddddota</code>		
<code>\not{a}</code>	<code>\not{a}</code>	<code>\mathring{a}</code>	<code>\mathring{a}</code>	<code>\widehat{AAA}</code>	<code>\widehat{AAA}</code>	<code>\widetilde{AAA}</code>	<code>\widetilde{AAA}</code>		

## Semnele plus și minus

LaTeX conferă semnelor + și – două semnificații posibile. Cea mai întâlnită este de operator binar. Când două elemente matematice apar în ambele părți ale semnului, se presupune că este operator binar, și, ca atare, alocă un anumit spațiu în fiecare parte a semnului. Cealaltă semnificație este de desemnare a semnului unui număr (pozitiv sau negativ). În acest sens, în matematică se folosește mai mult semnul – (pentru a desemna numere negative), întrucât un număr se presupune că este pozitiv dacă nu are semn. Semnul unui număr trebuie în mod normal să apară lângă număr (fără spațiu). Dacă puneți un semn + sau – fără nimic înaintea lui, dar vreți totuși să fie tratat ca un operator binar, puteți adăuga un caracter *invizibil* înaintea operatorului cu ajutorul `{}`. Acest lucru poate fi util dacă scrieți formule care se întind pe mai multe linii, iar o linie nouă ar putea începe cu = sau +, spre exemplu, și atunci puteți rezolva unele probleme de aliniere prin adăugarea caracterului invizibil acolo unde este necesar.

## Controlarea spațierii orizontale

LaTeX este evident destul de bun la formatarea matematicii — acesta a fost unul din scopurile principale ale sistemului Tex de bază pe care l-a extins LaTeX. Totuși, nu este întotdeauna foarte bun la interpretarea cu precizie a formulele matematice. Trebuie să facă anumite prezumții în cazul expresiilor ambigue. Rezultatul tinde să fie o spațiere orizontală puțin inestetică. În asemenea situații, outputul este încă satisfăcător, și totuși, orice perfecționist va dori fără îndoială să-și *ajusteze* formulele pentru a asigura o spațiere corectă. Aceste ajustări sunt în general de mare finețe.

Sunt și alte situații în care LaTeX dă un output corespunzător, însă doriți să mai adăugați ceva spațiu, poate pentru a adăuga un anumit tip de comentariu. Spre exemplu, în exemplul următor, este preferabil să vă asigurați că există destul spațiu între elementele matematice și text.

<pre> \left[   f(n) = \left\{ \begin{array}{l}   1 \\   n/2 \quad &amp; \\   \quad &amp; \text{pentru } n \text{ par} \\   -(n+1)/2 \quad &amp; \\   \quad &amp; \text{pentru } n \text{ impar} \end{array} \right. \right. </pre>	$f(n) = \begin{cases} n/2 & \text{pentru } n \text{ par} \\ -(n+1)/2 & \text{pentru } n \text{ impar} \end{cases}$
--	--

(Notați că acest exemplu poate fi exprimat mai elegant cu ajutorul mediului `cases` oferit de pachetul `amsmath`, care este descris în capitolul [Matematică avansată](#)).

LaTeX are două comenzi ce pot fi folosite oriunde în document (nu doar într-un mediu matematic) pentru a insera un anumit spațiu orizontal. Acestea sunt `\quad` și `\qquad`.

Un `\quad` este un spațiu egal cu dimensiunea fontului curent. Astfel, dacă utilizați un font de 11pt, atunci spațiul furnizat de comanda `\quad` va fi tot de 11pt (orizontal, desigur). Comanda `\qquad` dă dublul acestei valori. După cum puteți vedea în codul din exemplul anterior, am folosit spații `\quad` pentru a separa elementele matematice de text.

Bine, să revenim la ajustarea de finețe menționată la începutul documentului. Un exemplu bun l-ar constitui afișarea integralei nedefinite de  $y$  în funcție de  $x$ :

$$\int y \, dx$$

Dacă vreți să încercați asta în LaTeX, puteți scrie:

$$\left[ \int y \, \mathrm{d}x \right] \quad \Bigg| \quad \int y \, dx$$

Totuși, asta nu dă rezultatul corect. LaTeX nu ține cont de spațiul liber lăsat în cod pentru a semnifica faptul că  $y$  și  $dx$  sunt entități independente. În schimb, LaTeX le grupează laolaltă. Comanda `\quad` ar adăuga prea mult spațiu aici — în asemenea situații ne trebuie spații spații mai mici, oferite de LaTeX prin intermediul următoarelor comenzi:

Comandă	Descriere	Dimensiune
<code>\,</code>	spațiu mic	3/18 dintr-un <i>quad</i>
<code>\:</code>	spațiu mediu	4/18 dintr-un <i>quad</i>
<code>\;</code>	spațiu mare	5/18 dintr-un <i>quad</i>
<code>\!</code>	spațiu negativ	-3/18 dintr-un <i>quad</i>

**Notă:** puteți folosi mai multe comenzi împreună sau aceeași comandă de mai multe ori în același loc pentru a obține mai mult spațiu, dacă este necesar.

Așadar, pentru a rectifica problema curentă:

$$\left[ \int y \, \, \, \mathrm{d}x \right] \quad \Bigg| \quad \int y \, dx$$

$$\left[ \int y \, : \, \mathrm{d}x \right] \quad \Bigg| \quad \int y \, dx$$

$$\left[ \int y \, ; \, \mathrm{d}x \right] \quad \Bigg| \quad \int y \, dx$$

Spațiul negativ poate părea nepotrivit, totuși, n-ar fi acolo dacă nu ar avea *vreun* rost! Considerați exemplul următor:

```

\left(
\begin{array}{c}
n \\
r
\end{array}
\right) = \frac{n!}{r!(n-r)!}

```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Expresia matriceală pentru reprezentarea coeficienților binomiali de mai sus are prea mult spațiu între paranteze și conținut. Această situație poate fi ușor corectată prin adăugarea de spații negative după paranteza stângă și înainte de paranteza dreaptă:

```

\left(\!
\begin{array}{c}
n \\
r
\end{array}
\!\right) = \frac{n!}{r!(n-r)!}

```

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

În orice caz, adăugarea manuală de spații ar trebui evitată oricând este posibil: face codul sursă mai complex și este împotriva principiilor de bază ale abordării *What You See is What You Mean* — *Vezi ceea ce vrei să spui*, specifică LaTeX. Calea cea mai bună de urmat constă în definirea unor comenzi care să aibă spațierea dorită. Apoi, când folosiți aceste comenzi, nu mai trebuie să adăugați nici un spațiu. Mai târziu, dacă vă răzgândeți în legătură cu lungimea spațiului orizontal, puteți ușor să-l schimbați, modificând comanda definită mai înainte. Să zicem că vreți ca *d*-ul din *dx*-ul unei integrale să fie scris cu font *roman* și să fie la o distanță mică de restul textului. Dacă vreți să scrieți o integrală cum este  $\int x \, \mathrm{d}x$ , atunci puteți defini o comandă de genul:

```
\newcommand{\dd}{\; \mathrm{d}}
```

în preambulul documentului. Am ales numele `\dd` pur și simplu pentru că amintește de "d"-ul pe care îl scriem și pentru că este ușor de tipărit. Astfel, codul pentru integrală devine  $\int x \, \mathrm{d}x$ . Acum, oricând trebuie să scrieți o integrală, este de ajuns să utilizați comanda `\dd` în loc să scrieți "d", și toate

integralele vor avea același stil. Dacă vă răzgândiți, trebuie doar să modificați definiția din preambul, și toate integralele vor fi modificate în mod corespunzător.

## Matematică avansată: pachetul AMS Math

Pachetul de matematică de la AMS — [Societatea Americană de Matematică \(\*American Mathematical Society\*\)](#) oferă foarte multe facilități, ce creează un nivel mai înalt de abstractizare față de limbajul matematic LaTeX; dacă îl folosiți, vă va face viața mai ușoară. Unele comenzi `amsmath` vor face alte comenzi *plain* LaTeX să pară învechite: pentru a păstra consistența în documentul final, ar fi bine să utilizați comenzi `amsmath` oricând este posibil. Dacă faceți așa, veți obține un output elegant fără să vă faceți griji în legătură cu alinierea sau alte detalii, codul sursă rămânând accesibil. Dacă vreți să utilizați acest pachet, trebuie să adăugați în preambul:

```
\usepackage{amsmath}
```

### Text și puncte în formule

`amsmath` definește de asemenea comanda `\dots`, care este o generalizare a comenzii existente `\ldots`. Puteți folosi `\dots` atât în modul text, cât și în modul matematic, iar LaTeX va înlocui comanda cu trei puncte "...", însă va decide în funcție de context dacă va pune punctele jos (`\ldots`) sau centrat (`\cdots`).

### Puncte

LaTeX vă pune la dispoziție mai multe comenzi pentru a insera puncte în formule. Acest lucru se poate dovedi deosebit de util dacă trebuie să tipăriți niște matrici mari și să omiteți unele elemente. Mai întâi de toate, iată principalele comenzi din LaTeX legate de puncte:

Cod	Output	Comentariu
<code>\dots</code>	<code>...</code>	Puncte generice, de folosit în text (și în afara formulelor). Gestionează automat spațiile libere de dinainte și de după puncte potrivit contextului; este o comandă de nivel mai înalt.
<code>\ldots</code>	<code>...</code>	Outputul este similar cu cel anterior, însă nu se gestionează automat spațiul liber; funcționează la un





$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>	$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>
$\neq$	<code>\neq</code>	$\smile$	<code>\smile</code>	$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\dot{=}$	<code>\doteq</code>
$\frown$	<code>\frown</code>	$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\propto$	<code>\propto</code>	$=$	<code>=</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$<$	<code>&lt;</code>	$>$	<code>&gt;</code>		

### Litere grecești

Simbol	Script
A și $\alpha$	<code>\Alpha și \alpha</code>
B și $\beta$	<code>\Beta și \beta</code>
$\Gamma$ și $\gamma$	<code>\Gamma și \gamma</code>
$\Delta$ și $\delta$	<code>\Delta și \delta</code>
E, $\epsilon$ and $\varepsilon$	<code>\Epsilon, \epsilon și \varepsilon</code>
Z și $\zeta$	<code>\Zeta și \zeta</code>
H și $\eta$	<code>\Eta și \eta</code>
$\Theta$ , $\theta$ and $\vartheta$	<code>\Theta, \theta și \vartheta</code>
I și $\iota$	<code>\Iota și \iota</code>
K și $\kappa$	<code>\Kappa și \kappa</code>
$\Lambda$ și $\lambda$	<code>\Lambda și \lambda</code>
M și $\mu$	<code>\Mu și \mu</code>
N și $\nu$	<code>\Nu și \nu</code>
$\Xi$ și $\xi$	<code>\Xi și \xi</code>
$\Pi$ , $\pi$ și $\varpi$	<code>\Pi, \pi and \varpi</code>
$\rho$ , $\rho$ și $\varrho$	<code>\Rho, \rho și \varrho</code>
$\Sigma$ , $\sigma$ și $\varsigma$	<code>\Sigma, \sigma și \varsigma</code>
T și $\tau$	<code>\Tau și \tau</code>
$\Upsilon$ și $\upsilon$	<code>\Upsilon și \upsilon</code>
$\Phi$ , $\phi$ și $\varphi$	<code>\Phi, \phi și \varphi</code>
$\chi$ și $\chi$	<code>\Chi și \chi</code>
$\Psi$ și $\psi$	<code>\Psi și \psi</code>
$\Omega$ și $\omega$	<code>\Omega și \omega</code>

### Operații binare

Si	S	Si	Sc	Si	Sc	Si	Sc
m	cr	m	ri	m	ri	m	ri
bo	ip	bo	pt	bo	pt	bo	pt
l	t	l	pt	l	pt	l	pt