

EXPRESSÕES MATEMÁTICA COM O $\mathcal{AMS-LAT}_{\text{E}}\text{X}$

Usando $\mathcal{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

RESUMO. Este é o único arquivo de atividade deste tópico, mas ele trata de vários recursos para a produção de expressões matemáticas. Muitos deles foram introduzidos pelo $\mathcal{AMS-LAT}_{\text{E}}\text{X}$, enquanto outros foram melhorados por ele. Qualquer que seja o caso, atualmente os melhores resultados você obterá com esses recursos. Por isso, acostume-se a utilizar os pacotes do $\mathcal{AMS-LAT}_{\text{E}}\text{X}$: o `amsmath` ao menos. A classe utilizada para produzir este artigo, `amsart`, também faz parte do $\mathcal{AMS-LAT}_{\text{E}}\text{X}$.

1. O AMBIENTE `GATHER`

O ambiente `gather` deve ser utilizado para agrupar (*gather*) várias expressões matemáticas centralizadas, cada qual numa linha:

$$\begin{aligned} (1) \quad & a_0x^0 + a_1x^1 + \cdots + a_nx^n, \\ (2) \quad & a_0x^0y^0 + a_1x^1y^1 + \cdots + a_nx^ny^n. \end{aligned}$$


Observe que as mudanças de linha (e, por conseguinte, de expressão) são comandadas por `\\`. E que cada expressão recebe um número, que pode ser acessado através de `\ref` ou `\eqref`, bastando para isso nomeá-las com `\label`.

Para remover a numeração de algumas, mas não de todas as expressões, acrescente o comando `\nonumber` ou `\notag` nas linhas, ou expressões, apropriadas (não se esqueça de remover os comandos `\labels` correspondentes):

$$\begin{aligned} (3) \quad & a_0x^0 + a_1x^1 + \cdots + a_nx^n, \\ & a_0x^0y^0 + a_1x^1y^1 + \cdots + a_nx^ny^n. \end{aligned}$$

Mas se *nenhuma* das expressões deve ser enumerada, simplesmente utilize `gather*` ao invés de `gather`:

$$\begin{aligned} & a_0x^0 + a_1x^1 + \cdots + a_nx^n, \\ & a_0x^0y^0 + a_1x^1y^1 + \cdots + a_nx^ny^n. \end{aligned}$$

 Um erro comum ao utilizar o ambiente `gather` é colocar `\\` na última linha. Isto produzirá uma linha vazia, mas com um número associado a ela. Outro erro é deixar uma linha em branco entre o parágrafo anterior e o ambiente `gather`, produzindo espaço vertical demais ali. Lembre-se: a expressão faz parte do texto!

Há ainda o ambiente `gathered`. Diferentemente de `gather` e `gather*`, que promovem automaticamente a transição entre os modos parágrafo e matemático, este novo ambiente só pode ser utilizado *no* modo matemático. Isto é, entre `\[` e `\]` ou num ambiente `equation`, por exemplo. Sua utilidade ficará mais clara quando conhecermos outros ambientes matemáticos.

2. O COMANDO `\tag`

Um recurso interessante introduzido pelo $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX (*American Mathematical Society*) é o comando `\tag`, que permite inserir um rótulo absoluto na expressão. Ele substitui a numeração sequencial e serve para produzir variações nela. Por exemplo, veja como foi construída a numeração da segunda expressão abaixo:

$$(4) \quad a^2 = b^2 + c^2 \quad \text{e}$$

$$(4') \quad a^2 = b^2 + c^2 - 2bc \cos \theta.$$

Existe ainda a versão com asterísco, `\tag*`, que *não* insere os parênteses. Experimente: troque `\tag` por `\tag*` na expressão acima.

Um exemplo mais complexo de utilidade de `\tag` é o seguinte: imagine que você queira produzir duas sequências de numeração de expressões matemáticas. Por exemplo, as expressões de uma sequência são numeradas assim: (i), (ii) etc; enquanto as da segunda, (I), (II) etc¹.

A solução é simples: basta escrevermos `\tag{(i)}`, `\tag{(ii)}` etc num ambiente `equation*` para a primeira sequência, e `\tag{(I)}`, `\tag{(II)}` etc para a segunda. É claro que não faremos isso manualmente; usaremos contadores. Para isso, criamos um ambiente `eqroman` que será responsável por:

- (1) incrementar o contador `eqroman` (você pode dar o mesmo nome ao ambiente e ao contador),
- (2) iniciar e terminar o ambiente `equation*` e, finalmente,
- (3) inserir `\tag{\roman{eqroman}}`

Analogamente, definimos um ambiente `eqRoman` que incrementa o contador `eqRoman`, inicia e termina o ambiente `equation*`, e acrescenta o rótulo `\tag{\Roman{eqRoman}}` imediatamente antes de terminar o ambiente `equation*`. Esta é uma equação criada com `eqroman` (com “r” minúsculo):

$$(i) \quad a^2 = b^2 + c^2,$$

e esta, com o ambiente `eqRoman` (com “R” maiúsculo):

$$(I) \quad a^2 = b^2 + c^2 - 2bc \cos \theta.$$

¹Não me pergunte por que você faria isso!

Assim, a numeração torna-se automática e você pode inclusive atribuir rótulos (comando `\label`) a essas expressões e citá-las com `\ref` ou `\eqref`.

Talvez uma aplicação mais útil dos comandos `\tag` e `\tag*` seja a de você rerepresentar uma expressão qualquer, mantendo a numeração original. Por exemplo, o número da expressão

$$(2) \quad a_0x^0y^0 + a_1x^1y^1 + \cdots + a_nx^ny^n$$

é o mesmo da primeira exibição dela, na seção 1. Para fazer isso eu defini o rótulo absoluto da equação como sendo igual à numeração da equação 2 num ambiente `equation*`. O asterísco é importante para que não incrementemos o contador `equation`, prejudicando a contagem das expressões seguintes.

3. O AMBIENTE `MULTLINE`

Diferentemente do `gather`, o ambiente `multline` deve ser utilizado numa única expressão longa, que não cabe em uma linha. Por exemplo,

$$(5) \quad (a+b)^{10} = \sum_{i=0}^{10} \binom{10-i}{i} a^i b^{10-i} = a^{10}b^0 + \\ 10a^9b^1 + 45a^8b^2 + 100a^7b^3 + 190a^6b^4 + 252a^5b^5 + \\ 190a^4b^6 + 100a^3b^7 + 45a^2b^8 + 10a^1b^9 + a^0b^{10}$$

é uma equação longa demais para caber em uma linha apenas. Então, usamos `multline` para quebrá-la em várias linhas. E note que, novamente, a mudança de linha é devida ao `\\`, que pode ser colocado em qualquer lugar da equação. No exemplo acima (e provavelmente em todas as situações onde você utilizará `multline`) este comando foi posicionado por tentativa e erro.

Observe ainda que, como se trata de apenas uma equação, há somente um número (e `\label` e/ou `\tag`) associado a ela. Mas você pode suprimir a numeração utilizando a versão com asterísco do ambiente: `multline*`.



Um erro bastante comum ao utilizar o ambiente `multline` é errar a digitação do nome dele: é `multline`, com “t” mudo, não `multlíne`. Este ambiente não existe (mas você pode criá-lo, se quiser).

Observe a equação (5): a primeira linha aparece alinhada à esquerda e a última, à direita. E as linhas intermediárias estão centralizadas. Este é o comportamento padrão do ambiente `multline`, mas você pode modificá-lo. Primeiramente, o comprimento `\multlinegap`

define a endentação da primeira e da última linha... 1 cm neste exemplo:

$$(a+b)^{10} = \sum_{i=0}^{10} \binom{10-i}{i} a^i b^{10-i} = a^{10} b^0 + \\ 10a^9 b^1 + 45a^8 b^2 + 100a^7 b^3 + 190a^6 b^4 + 252a^5 b^5 + \\ 190a^4 b^6 + 100a^3 b^7 + 45a^2 b^8 + 10a^1 b^9 + a^0 b^{10}$$

⚠ Se você utilizar o ambiente `multline*` (com asterísco), o efeito da endentação será nulo na primeira linha da equação. Note também que utilizamos o ambiente `setlength` para alterar `\multlinegap` e, ao mesmo tempo, definir o escopo de validade desta alteração. É uma alternativa ao comando `\setlength`.

E para as linhas intermediárias você pode utilizar os comandos `\shoveleft` e `\shoveright`, que assemelham-se aos `\raggedleft` e `\raggedright`, respectivamente². Neste exemplo,

$$(a+b)^{10} = \sum_{i=0}^{10} \binom{10-i}{i} a^i b^{10-i} = a^{10} b^0 + \\ 10a^9 b^1 + 45a^8 b^2 + 100a^7 b^3 + 190a^6 b^4 + 252a^5 b^5 + \\ 190a^4 b^6 + 100a^3 b^7 + 45a^2 b^8 + 10a^1 b^9 + a^0 b^{10},$$

a linha intermediária foi empurrada para a direita com `\shoveright`. Experimente utilizar `\shoveleft` e, simultaneamente, alterar o comprimento `\multlinegap`.

⚠ A presença do comando `\\` em expressões matemáticas com mais de uma linha acarreta um complicações extra, que costuma ocorrer especialmente em conjunção com os comandos `\left` e `\right`. Tenha em mente o seguinte: *não pode haver um `\\` entre um par de chaves ou entre um `\left` e um `\right`.*

Uma maneira de avaliar se você está cumprindo com esta regra consiste em colocar cada linha da sua expressão em ambientes matemáticos separados e compilar. Se tudo transcorrer sem erros, então tudo bem. Mas se o L^AT_EX informar o erro

Missing `\right`. inserted

é porque você desrespeitou esta regra.

Utilize a equação abaixo para explorar, com o professor e com seus colegas, esta questão. O objetivo é quebrar esta equação em duas, mudando de linha logo após a parcela $\partial B_x / \partial x$:

$$\mathbf{F} = p_m \left(\frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial x} + \frac{\partial B_z}{\partial x} \right)$$

²Atenção: `\shoveleft` e `\shoveright` são comandos ordinários, que agem sobre seus argumentos, enquanto `\raggedleft` e `\raggedright` são declarações.

4. O AMBIENTE `SUBEQUATIONS`

Ainda mais comum que numerar expressões como em (4') é escrever (4a), (4b) etc. Para esta situação existe o ambiente `subequations`:

$$(6a) \quad a^2 = b^2 + c^2$$

$$(6b) \quad a^2 = b^2 + c^2 - 2bc \cos \theta.$$

Com esta construção você pode citar o grupo de equações, como em (6), ou cada uma delas isoladamente: (6a) e (6b). Note que não precisamos digitar as letras “a”, “b” etc. Isto é feito automaticamente (com contadores, como você certamente imaginou).



Observe que `subequations` não é um ambiente que inicia/termina o modo matemático. Ele apenas define o conjunto de expressões que, por algum motivo, estão relacionadas. Isto significa que você pode colocar não apenas `gathers` dentro de `subequations`, mas qualquer outro ambiente matemático! Veja:

$$(7a) \quad a^2 = b^2 + c^2$$

$$(7b) \quad a^2 = b^2 + c^2 - 2bc \cos \theta.$$

$$(7c) \quad (a+b)^{10} = \sum_{i=0}^{10} \binom{10-i}{i} a^i b^{10-i} = a^{10} b^0 + 10a^9 b^1 + 45a^8 b^2 + 100a^7 b^3 + 190a^6 b^4 + 252a^5 b^5 + 190a^4 b^6 + 100a^3 b^7 + 45a^2 b^8 + 10a^1 b^9 + a^0 b^{10}$$

$$I = I_0 e^{-t/\tau_{RL}} + \frac{\mathcal{E}}{R} (1 - e^{-t/\tau_{RL}})$$

As duas primeiras equações foram criadas com `gather` e a terceira, com `multline`. A última equação foi construída com `\[` e `\]` e, por isso, sequer recebem um número. Trata-se de uma situação atípica, mas serve para ilustrar a utilização do ambiente `subequations`.

5. O AMBIENTE `ALIGN`

Outro tipo de construção comum são equações alinhadas por algum símbolo. Por exemplo, as equações

$$(8) \quad x = \frac{a^2 - b^2}{a + b}$$

$$(9) \quad = \frac{(a - b)(a + b)}{a + b}$$

$$(10) \quad = a - b$$

estão todas alinhadas pelo símbolo de igual. Isto é feito com o ambiente `align` ou `align*` do pacote `amsmath`. `\[` promove a mudança de linha e `&` indica a posição do alinhamento que, no caso, é à esquerda do símbolo de igual.

Na verdade, o `&` tem dupla função: de marcar a posição do alinhamento e de inserir uma tabulação. É mais fácil compreender isto com um exemplo:

$$\begin{array}{ll} f(x) = ax^2 + bx + c & g(x) = ax^3 + bx^2 + cx + d \\ = a(x - x_1)(x - x_2) & = a(x - x_1)(x - x_2)(x - x_3). \end{array}$$

Considere apenas a primeira linha: o primeiro `&` marca o alinhamento em $f(x)$, o segundo marca uma tabulação, que separa $f(x)$ de $g(x)$, e o terceiro `&` marca o alinhamento em $g(x)$. A mesma análise vale para a segunda linha. Na verdade, os `&` de índice ímpar (o primeiro e o terceiro, no exemplo) têm função de alinhamento, enquanto os de índice par (segundo) tem função de tabulação, como nas tabelas. A quantidade de `&` numa mesma linha é limitada apenas pelo tamanho da linha.



O L^AT_EX define o ambiente `eqnarray`, cuja função é a mesma do `align`. No entanto, este último faz um trabalho bem melhor e, por conta disso, atualmente não é mais recomendado o uso do `eqnarray` ou `eqnarray*`. Quando você o vir por aí, substitua-o por `align` (os porquês de se evitar `eqnarray` podem ser vistos [aqui](#)).

6. MATRIZES

Da mesma forma que o ambiente `align` hoje em dia substitui o `eqnarray`, o pacote `amsmath` define o ambiente `matrix`, que deve substituir o `array`. Veja só: uma matriz...

$$\begin{array}{ll} \text{... com matrix:} & \text{... com array:} \\ \left\langle \begin{array}{cc} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{array} \right\rangle & \left\langle \begin{array}{cc} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{array} \right\rangle \end{array}$$

Note que, com `matrix`, você não precisa definir a quantidade de colunas da sua “tabela”; basta seguir adicionando `&` conforme a necessidade.

No caso particular de matrizes com parênteses, o ambiente `pmatrix` torna o trabalho ainda mais fácil, pois ele os insere automaticamente:

$$\left(\begin{array}{cc} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{array} \right)$$

7. EQUAÇÕES CONDICIONAIS

Para terminar, vejamos o ambiente `cases`, que permite a construção de equações com condições, como esta:

$$\theta(x) = \begin{cases} 0, & \text{se } x < 0 \\ \frac{1}{2}, & \text{se } x = 0 \\ 1, & \text{se } x > 0 \end{cases}$$

CONCLUSÃO

Como vimos, o pacote `amsmath` trouxe recursos que simplificam enormemente a produção de expressões matemáticas de alta qualidade. Fazer o que fizemos aqui sem elas seria muito trabalhoso com os recursos básicos do \LaTeX . Por isso, é útil conhecer algumas das inúmeras ferramentas que este pacote disponibiliza. O que vimos aqui é o básico, para não dizer o mínimo. Baixe o manual do \mathcal{AMS} - \LaTeX [aqui](#). Outro documento interessante é o `Math mode`, de Herbert Voß. Há também o livro “Math into \LaTeX ”, de George Grätzer.