

## Apêndice A: equações e gráficos

---

### Equações

Escrever equações matemáticas não é uma tarefa fácil em nenhum sistema. Mas graças à enorme quantidade de pessoas que usam e desenvolvem ferramentas para gerar conteúdos para a internet, existem diversas soluções para os mais variados gostos.

A maneira mais rápida e direta de se produzir equações em HTML é ser criativo com os recursos dos próprios elementos HTML e declarações de estilo. Isto em geral funciona para equações pequenas e simples, mas mesmo assim com resultados nem sempre muito satisfatórios. O exemplo a seguir mostra algumas possibilidades.

### Listagem:

```
<i>V</i> =  
(4/3) &pi;<i>R</i><sup>3</sup>
```

$$V = (4/3)\pi R^3$$

```
<i>r</i> =  
exp[<i>a</i> / (<i>b</i> <sup>_</sup>  
<i>c</i>) <sup>1/2</sup>]
```

$$r = \exp[a / (b - c)^{1/2}]$$

```
<b>τ</b> =  
<b>r</b> &times; <b>F</b>
```

$$\boldsymbol{\tau} = \mathbf{r} \times \mathbf{F}$$

```
<b>a</b> =  
<i>d</i><sup>2</sup><b>x</b>/  
<i>d</i><sup>2</sup><i>t</i><sup>2</sup> =  
<i>a</i><sub>x</sub> <b>i</b> +  
<i>a</i><sub>y</sub> <b>j</b> +  
<i>a</i><sub>z</sub> <b>k</b>
```

$$\mathbf{a} = d^2\mathbf{x}/dt^2 = a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k}$$

```
<i>I</i> = (1/2) <i>M</i>  
(<i>R</i><sub>1</sub><sup>2</sup> +  
<i>R</i><sub>2</sub><sup>2</sup>)
```

$$I = (1/2)M(R_1^2 + R_2^2)$$

```
<font size="+2">&int;</font>  
<i>x</i><sup>2</sup> <i>d</i><i>x</i> =
```

$$\int x^2 dx = x^3/3$$

`<i>x</i><sup>3</sup>/3`

Note a presença de vários caracteres especiais escritos com códigos que começam com "&" e terminam com um ";". No apêndice, você vai encontrar uma listagem com dezenas desses caracteres. Note, em particular, que o sinal de multiplicação, ×, não é um "x" e o sinal de menos, −, não é um hífen, "-".

<code>&amp;pi;</code> → π	<code>&amp;mu;</code> → μ	<code>&amp;omega;</code> → ω
<code>&amp;times;</code> → ×	<code>&amp;minus;</code> → −	<code>&amp;int;</code> → ∫

### Sua vez.. (0-NaN)

Estude a sintaxe dos exemplos anteriores e construa sequências de texto com marcadores HTML de modo a produzir as equações abaixo:

$$x(t) = A \text{sen}(\omega t) + B \text{cos}(\omega t)$$
$$y = v_t [ t - (m/k)(1 - e^{-(k/m)t}) ]$$

É trabalhoso, prolixo e maçante escrever equações desta maneira. Além disso, equações mais complexas podem não ficar claras ou não serem apresentadas na forma que usualmente utilizamos. Considere, por exemplo, a equação a seguir que, além de difícil de especificar, produz uma equação de baixíssima qualidade editorial.

### Listagem:

```
<i>&rho;</i> = <i>&rho;</i><sub>r</sub>
[ (<i>D</i><sub>b</sub><sup>3</sup>/3 -
<i>H</i><sub>r</sub> <i>D</i><sub>h</sub><sup>2</sup>/2) /
(<i>D</i><sub>b</sub><sup>3</sup>/3 -
<i>H</i><sub>h</sub><sup>2</sup>/2) ]
```

### Resultado:

$$\rho = \rho_r [ (D_b^3/3 - H_r D_h^2/2) / (D_b^3/3 - H D_h^2/2) ]$$

O comitê internacional que propõe as especificações para a internet (W3C, ou *World wide web consortium*) desenvolveu, para isso, o MathML. O MathML é um poderoso esquema de marcação dos elementos de uma equação que, além de proporcionar uma formatação de qualidade gráfica elevada, permite marcar cada número e símbolo em uma equação individualmente, permitindo com que a eles sejam atribuídas declarações de estilo ou que sejam manipulados utilizando JavaScript. Isso, entretanto, também tem um preço: a sintaxe da marcação é extremamente prolixa e trabalhosa, deixando ainda o código-fonte muito difícil de compreender.

Um exemplo pode deixar isso mais claro. A equação para a distribuição angular da função de onda de um elétron no estado com momento angular  $\ell = 1$  e projeção  $m_\ell = 0$  é dada por:

$$\Theta_{10}(\theta) = \frac{3}{4} \pi \cos \theta$$

que, em MathML, é especificada da seguinte maneira:

```
<math display="block">
  <msub>
    <mi mathvariant="normal">&#x0398;</mi>
    <mrow>
      <mn>10</mn>
    </mrow>
  </msub>
  <mo stretchy="false">(</mo>
  <mi>&theta</mi>
  <mo stretchy="false">)</mo>
  <mo>=</mo>
  <msqrt>
    <mfrac>
      <mn>3</mn>
      <mrow>
        <mn>4</mn>
        <mi>&#x03C0;</mi>
      </mrow>
    </mfrac>
  </msqrt>
  <mi>cos</mi>
  <mo>&#x2061;</mo>
  <mi>&#x03B8;</mi>
</math>
```

Parece absurdo ter que fazer isso para especificar uma simples equação mas, acredite, dentro da filosofia geral de marcação dos elementos de um documento segundo sua funcionalidade, isso tem suas razões e usos. Se não fosse assim, tanta gente inteligente e bem paga não perderia tempo desenvolvendo isso.

Entretanto, na maioria das vezes deseja-se simplesmente escrever uma equação sem atribuir qualquer marcação especial ou funcionalidade a seus elementos. Devido à importância e versatilidade da internet hoje, diversas entidades produziram aplicativos e *plug-ins* que podem ser utilizados *online* e *offline* para descrever e renderizar equações das maneiras tradicionalmente utilizadas por usuários de diferentes áreas. Neste texto optamos por apresentar uma estratégia que faz uso de um *plug-in* gratuito, o `MathJax.js`, que permite ao usuário digitar suas equações como se estivesse utilizando o `LATEX`, e que automaticamente disponibiliza a codificação da equação também em MathML.

O exemplo a seguir mostra como montar um documento que incorpora o *plug-in* do `MathJax.js` previamente baixado no computador do usuário e colocado na pasta `c:/MathJax` e como codificar a equação acima utilizando a sintaxe do `LATEX`.

```
<html>
<head>
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    extensions: ["tex2jax.js"],
```

```

    jax: ["input/TeX","output/HTML-CSS"],
    tex2jax: {inlineMath: [["$","$"],["\\(","\\)"]}
  });
</script>
<script src="file:///c:/MathJax/MathJax.js"></script>
</head>
<body>

\begin{equation}
\rho = \left(
\frac{\frac{1}{3}D_b^3 - \frac{1}{2}D_h^2 H_r}{
\frac{1}{3}D_b^3 - \frac{1}{2}D_h^2 H}
\right)
\rho_r
\end{equation}

</body>
</html>

```

Ao carregar o documento, o navegador deverá apresentar somente algo como:

$$\rho = \left( \frac{\frac{1}{3}D_b^3 - \frac{1}{2}D_h^2 H_r}{\frac{1}{3}D_b^3 - \frac{1}{2}D_h^2 H} \right) \rho_r$$

No exemplo, foi assumido que o MathJax está na pasta `c:/MathJax` do computador do usuário, mas é possível especificar outros endereços, como a página do próprio MathJax ([cdn.mathjax.org](http://cdn.mathjax.org)) na internet.

O L<sup>A</sup>T<sub>E</sub>X vem sendo há décadas uma das principais plataformas utilizadas na edição de textos técnicos e científicos de excelente qualidade editorial. Suas possibilidades são muitas e inúmeros livros, tutoriais e exemplos sobre seu uso podem ser encontrados na internet, assim como sobre o MathML e o MathJax.

## Gráficos

Textos técnicos e científicos também frequentemente incluem gráficos e, novamente, a internet está repleta de *plug-ins* para auxiliar na produção deles. Uma busca vai lhe revelar uma constelação de possibilidades, das gratuitas às comerciais, das mais simples às extremamente sofisticadas. Todas funcionam de modo semelhante, apenas variando na complexidade com que são especificados os seus recursos (mas não se iluda, algumas pedem um elevado grau de dedicação para obter algum resultado!).

Neste documento utilizamos uma das mais simples, o Flot ([www.flotcharts.org](http://www.flotcharts.org)), para ilustrar como produzir gráficos "cotidianos" rapidamente. O exemplo a seguir mostra como fazer o gráfico da equação para a distribuição angular discutida na seção anterior.

## Listagem:

```

<html>

<head>
<script src="file:///c:/flot/jquery.js"></script>
<script src="file:///c:/flot/jquery.flot.js"></script>
</head>

<body>

<div id='grT10' style='width:300;height:200;margin:auto'></div>

<script>

function T10(x) {
    return Math.sqrt(3/4/Math.PI)*Math.cos(x);
}

var ptsT10 = [];
var ptsT10sqr = [];

for (var theta=0;theta<=Math.PI;theta=theta+Math.PI/36) {
    ptsT10.push([theta,T10(theta)]);
    ptsT10sqr.push([theta,T10(theta)*T10(theta)]);
}

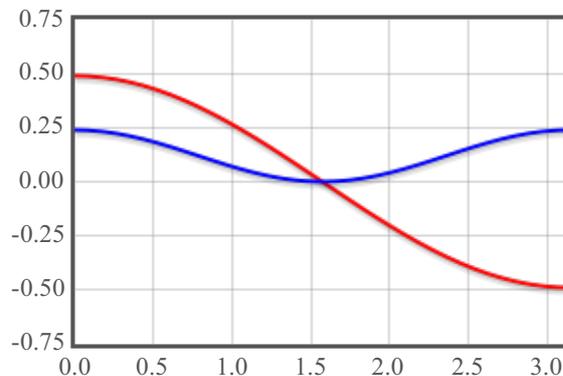
$.plot("#grT10",[ptsT10,ptsT10sqr],{ colors:["red","blue"] });

</script>

</body>
</html>

```

### Resultado:



O exemplo se inicia com a inclusão, no cabeçalho (`head`) de dois arquivos previamente baixados no computador do usuário e colocados no diretório `c:/flot` (aqui também poderia ser feita uma referência ao site do Flot na internet).

O corpo do documento contém essencialmente dois blocos. No primeiro, apenas a definição da `div` que vai receber o gráfico. É obrigatória a atribuição de um `id` para a divisão, que será utilizado pela função que efetivamente desenha o gráfico, e de uma largura (`width`) e altura (`height`) em uma declaração de estilo (`style`).

No segundo bloco está o script que define a função de interesse. Em seguida, são declaradas duas matrizes (vetores), uma para armazenar os pontos para o gráfico da função

de onda e outra para armazenar o gráfico do seu quadrado. O laço `for` utiliza o método `.push(...)` para adicionar a cada elemento das matrizes anteriormente declaradas uma matriz com apenas dois elementos que representam as coordenadas dos pontos do gráfico.

Na última linha do script é chamada a função `$.plot` do Flot, que recebe três parâmetros: o "endereço" da divisão onde o gráfico deve ser desenhado, uma matriz com a(s) matriz(es) de pontos a serem graficados e um objeto com as opções do gráfico. Este último parâmetro (o objeto com as opções do gráfico) pode ser definido previamente como uma variável bastante complexa, incluindo especificações sobre legendas, cores, tipos de gráficos (linhas, pontos, barras etc.), extremos e precisão das escalas, marcas de grade etc. Como usual, o leitor interessado pode encontrar detalhes e informações nos sites dos fornecedores dos *plug-ins* que optar utilizar.

Note que mesmo que se queira graficar apenas um conjunto de dados, é preciso colocar a matriz que contém os pontos dentro de outra matriz. Isto é, se o objetivo fosse graficar apenas a função de onda, a instrução seria:

```
$.plot("#grT10", [ptsT10]);
```

Note também que não foi passado o terceiro parâmetro com as opções do gráfico, o que faz com que as opções padrão sejam utilizadas.

